Design of Memory-Based FFT Processor with Generalized Efficient Conflict-Free Address Schemes

K.MADHU[1], SHAIK MAHABOOB SUBAHAN[2]
[1]*P.G Student, Department of Electronics and Communication Engineering*
[2]*Assistant Professor, Dr.K.V. Subba Reddy Institute of Technology.*

Abstract- A generalized efficient conflict-free address scheme for arbitrary point memory-based fast Fourier transform (FFT) processor was presented. In the proposed scheme, a high radix decomposition method was utilized to reduce the computation levels and small radix connected multipath-delay-commutator butterfly units were adopted to eliminate the complexity of the computation engine as well. Several important functions of memory-based FFT processor were combined together, including the continuous-flow mode, variable computation size and conflict-free address scheme. Moreover, a prime factor algorithm was employed to decrease the multiplications and the twiddle factor storage when there exist prime factors in the decomposition. At last, a unified Winograd Fourier transform algorithm (WFTA) butterfly core was designed for the small 2, 3, 4, 5 point DFTs to reduce the computation complexity further. Simulation results show that, the power cost of FFT processor can only be 40.8 mW in work frequency 122.88 MHz, it is very suitable for the LTE system.

Index Terms- Continuous data flow, fast Fourier transform (FFT), generalized mixed radix (GMR), in-place, vector reverse.

I. INTRODUCTION

THE symmetrical recurrence division multiplex (OFDM) regulation method has been generally misused in correspondence frameworks, for example, 802.11, earthbound computerized video broadcasting (DVB-T), and earthbound advanced sight and sound telecom (DMB-T). Nonetheless, OFDM regulation includes the discrete Fourier change (DFT) that requirements generous calculation. Today, different FFT processors, for example, pipelined or memory-based designs, have been proposed for various applications. Be that as it may, for long-measure FFT processors, for example, the 2048-point FFT, the pipelined engineering would cost more region and power than the memory-based plan. Thus, memory based methodology has increased increasingly consideration as of late in FFT processor plans for long-estimate DFT applications. For the memory-based processor configuration, limiting the important memory estimate is successful for zone decrease since the memory costs a noteworthy piece of the processor. Then again, the FFT processor for the most part receives on-chip static irregular access memory (SRAM) rather than outside memory. The reason is the high-voltage I/O and the expansive capacitance in the printer-circuit-board (PCB) follow would expand control utilization for outside memory. Other than the power issue, utilizing outer memory additionally builds the PCB-level check cost for final result producers. Hence, it is a pattern to utilize the on-chip SRAM for FFT processors and to lead FFT streamlining for better framework level incorporation.

II. LITERATURE SURVEY

To limit the important memory measure, a set up approach [1] is taken for the two butterflies yield and I/O information. That is, the yield information of butterflies are composed back to their unique area amid the calculation time. In addition, for the I/O information, the new information x[n] would be placed in the area of the yield information X[n] of the past FFT image. Then again, for the memory-based processor, the high-radix structure would be taken to expand the throughput to meet continuous prerequisites. In any case, when both set up arrangement and highradix structure are received all the while, it would result in memory struggle issue if the information have not been tended to appropriately. As of not long ago, there have been a few distributions on this issue, the set up approach for multibank memory structure, to advance the processor plan [1]– [3]. Be that as it may, the methodologies specified in [1] and [2] work for settled radix-r. The methodology in [3] works for radix-2/4. Henceforth, every one of them couldn't be utilized for prime-number memory-based DFT processor outline, this still remains a test today. Note that a prime-number FFT has been proposed in late correspondence principles, e.g., the 3780-point DFT in Chinese direct TV (DMB-T) [4] and the 1536-point DFT in Third Generation Partnership Project Long-Term Evolution [5]. Despite the fact that, for the prime-number FFT calculation, the zero-cushioning technique can be taken to estimated it by processing some 2n-point FFT, it has poor flag to-quantization-commotion proportion execution than a prime-radix FFT. Consequently, to build up a methodology for a multibank set up tending to calculation for general size, a FFT is important. In this concise, a summed up blended radix (GMR) calculation is proposed to upgrade the memory-based

FFT processor plan. It underpins not just set up approach to limit the important memory estimate for the two butterflies yield and I/O information yet in addition multibank memory structure to build its most extreme throughput to fulfill more framework applications without memory strife. After the calculation is presented, we take the 3780-point FFT as an illustrative precedent. At last, a low-many-sided quality equipment usage of a file vector generator is likewise proposed for our calculation.

III. EXISTING METHOD

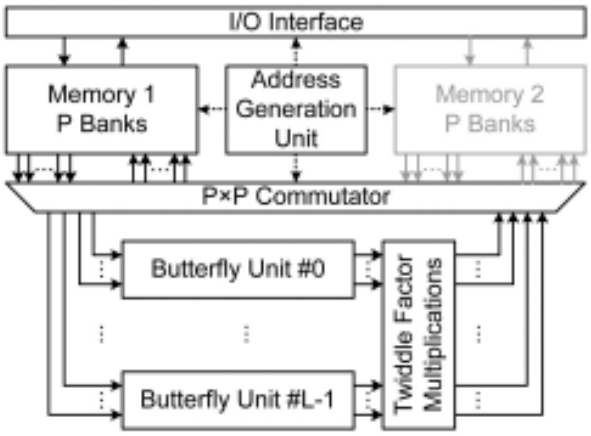

Fig.1: Architecture of a multiple-PE,

The meaning of a discrete Fourier change (DFT) is given by C(k) = N−1 n=0 x(n)Wnk N . In [15] and [16], the strategy for breaking down one-dimensional DFTs into multidimensional DFTs is illustrated. On the off chance that the length N is deteriorated into N1N2, the record mapping can be communicated as takes after:

n = K1n1 + K2n2N , n1, k1 = 0, 1,..., N1 − 1

k = K3k1 + K4k2N , n2, k2 = 0, 1,..., N2 − 1 (2) memory-based FFT processor [12].

where • signifies modulo-N activity. On the off chance that N1 and N2 are moderately prime, a Ruritanian guide for the info n and a Chinese leftover portion hypothesis delineate the yield k are utilized. In this manner

K1 = N2; K3 = N2N−1 2 , so N2N−1 2 N1 = 1 K2 = N1;

K4 = N1N−1 1 , so N1N−1 1 N2 = 1. (3)

This is the celebrated Cooley– Tukey calculation, which is likewise called the normal factor calculation (CFA). Both the PFA and the CFA decay the N-point DFT into N2 DFTs with N1-point and N1 DFTs with N2-point progressively. They both use an information list delineate calculation and a yield reorder outline regular yielding. The mapping strategy for the PFA is more entangled than that of the CFA. Be that as it may, there are twiddle factors Wn2k1 N between the two-level DFTs for the CFA yet not for the PFA. The mapping strategies for (3) and (5) can both be stretched out to measurements more noteworthy than two.

IV. PROPOSED FFT ARCHITECTURES

There are two different FFT modules in LTE system: one is the 2n mode (n = 7–11) FFT, and the other is the 35 different-length NSPP DFT, whose transform sizes range from 12 to 1296. The architectures of the 2n-point FFT and the NSPP DFT are similar. Only some details are different. The entire FFT processor architecture is shown in Fig. 2. It consists of the following main parts: an input and an output index vector generator, a computation address generator, three different memory bank groups, a PE unit applied with the HRSB scheme, some commutators between the memory and the PE, some prestored twiddle factors, and FFT size parameters, and an exponent scaling unit.

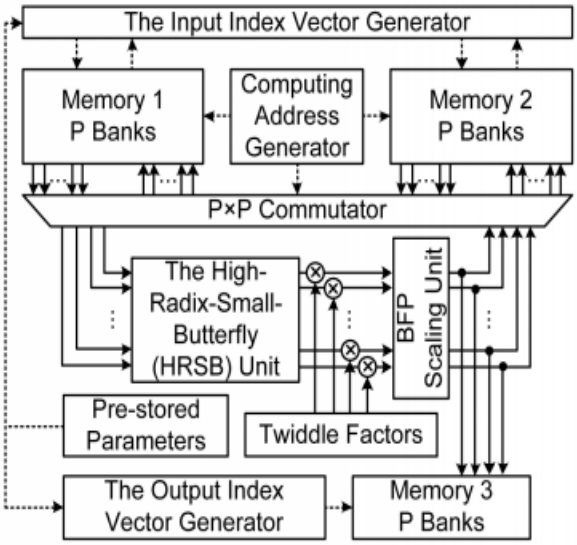

Fig.2: Block diagram of the proposed FFT processor architecture.

The dashed lines represent the control signals while the real lines denote the flowing data. The input index vector generator distributes the input data to different memory banks without data conflicts, and the output one reorders the output data to a natural sequence. The computation address generator obtains all the concurrent data of each cycle and stores back the intermediate results. Memory groups 1 and 2 are in a ping-pong mode to hold two continuous data symbols in input sampling, and memory group 3 is used to output the computed data in right order. Three memory groups only exist in NSPP FFTs. For SPP FFTs, as in-place strategy is available, only two memory groups are enough. The HRSB unit is the kernel processing engine. The commutators located between the memories and HRSB unit provide efficient data routing

mechanism which is controlled by the computation address generator. The twiddle factors and the FFT size parameters are all prestored in register files, which are used to configure the FFT working modes. The exponent scaling unit controls scaling operations for block floating-point, which can increase the signal-to-quantizationnoise ratio and reduce the memory storage.

The 2n-point FFT is operated in continuous-flow mode with the in-place strategy using only two parallel radix-2/22/23 MDC units. For instance, to execute a 128-point FFT, only one MDC unit is active in stage 0, and two radix-23 MDC units are activated in the remaining stages. The computation can be completed within 128 clock cycles. The input and output index vector generators in the 2n-point FFT are merged to one. The only difference is that the binary representation of the index for the input is in a forward manner, while it is in a reversed manner for the output. The architectures of the index vector generator and the radix-23 MDC unit are detailed in [12].

The length of the NSPP DFT can be generally expressed as N = 2p3q 5r , p = 2,..., 8; q = 1,..., 5; q = 1, 2. (22) Because the factors 2, 3, and 5 are mutually prime, [14] simply factorizes N into three different prime parts, namely, 2p, 3q, and 5r using the PFA, and applies the CFA within different parts. However, this factorization method cannot completely satisfy the continuous-flow mode. For example, the 972-point DFT is decomposed into 4 × 3 × 9 × 9, in [14]. The ninepoint DFTs are computed using a radix-3 delay element matrix within three cycles, and radix-4 and radix-3 DFTs can both be computed in one cycle. Therefore, the total computation cycle is 243 + 324 + 324 + 324 = 1215, which is more than the computation length. Moreover, it suffers a much more complicated data routing scheme.

In this paper, we present a new factorization method that will make the continuous-flow mode completely available and simultaneously ensure the effectiveness of the conflictfree address scheme. The DFT length N is first decomposed into three prime parts as in [14]. If the number of computation cycles is not available to support the continuous-flow mode, some trivial factors are combined together to make the decomposition more compact. The proposed new factorization method consists of the following steps.

i) Obtain each value of p, q, and r.
ii) If $p > 4$, 2p is decomposed into 24 and $2p-4$, except that 25 is composed of 23 and 22; otherwise, there is only one level for 2p. For 3q, if q = 5, there will be three levels, 32, 32, and 3; 32 and $3q-2$ for q = 3, 4; and one level for q = 1, 2. And only one level for 5r.
iii) Calculate the computation time based on the above decomposition. If the decomposition levels for 2p, 3q, and 5r are i, j, and k, the computation time of the complete decomposition is $T = i \cdot (N/4) + j \cdot (N/3) + k \cdot (N/5)$.
iv) If the computation time T is still larger than N, some trivial factors in each part will be combined. However, the 24, 32, and 52 factors should remain unchanged.
v) Calculate the new computation time. If it still does not satisfy the time restriction, step to iv) again.

V. SIMULATION RESULTS

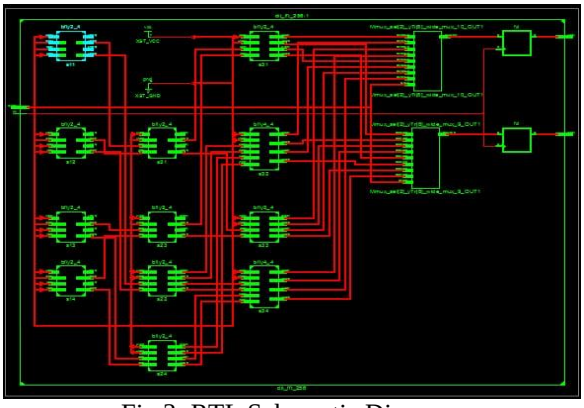

Fig.3: RTL Schematic Diagram

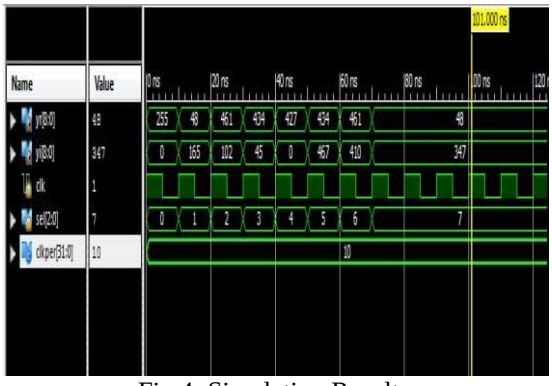

Fig.4: Simulation Results

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	153	63400	0%
Number of fully used LUT-FF pairs	0	153	0%
Number of bonded IOBs	22	210	10%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig.5: Design Summary

VI. CONCLUSION

In this brief, a GMR algorithm has been proposed to optimize the general-size memory-based FFT processor design. It supports the in-place policy for both butterflies output and I/O data to minimize the necessary memory size. Hence, only 2N words memory is required for any size FFT computation for real-time requirements. Furthermore, our proposal also supports the multibank addressing for a high-radix structure without memory conflict by reversing the decomposition order of the previous FFT symbol. Finally, a lowcomplexity index vector generator has been proposed for our algorithm. It only costs a few accumulators, making our proposal very suitable for multistandard and multimode OFDMbased applications.

VII. REFERENCES

[1]. L. G. Johnson, "Struggle free memory tending to for committed FFT equipment," IEEE Trans. Circuits Syst. II, Analog Digit. Flag Process., vol. 39, no. 5, pp. 312– 316, May 1992.
[2]. J. A. Hidalgo, J. Lopez, F. Arguello, and E. L. Zapata, "Zone proficient engineering for quick Fourier change," IEEE Trans. Circuits Syst. II, Analog Digit. Flag Process., vol. 46, no. 2, pp. 187– 193, Feb. 1999.
[3]. B. G. Jo and M. H. Sunwoo, "New constant stream blended radix (CFMR) FFT processor utilizing novel set up methodology," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 5, pp. 911– 919, May 2005.
[4]. Z.- X. Yang, Y.- P. Hu, C.- Y. Dish, and L. Yang, "Plan of a 3780-point IFFT processor for TDS-OFDM," IEEE Trans. Communicate., vol. 48, no. 1, pp. 57– 61, Mar. 2002.
[5]. 3GPP TS 36.201 V8.3.0 LTE Physical Layer—General Description, E-UTRA, Mar. 2009.
[6]. C. Burrus, "Record mappings for multidimensional plan of the DFT and convolution," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-25, no. 3, pp. 239– 242, Jun. 1977.
[7]. D. P. Kolba and T. W. Parks, "A prime factor FFT calculation utilizing fast convolution," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-25, no. 4, pp. 281– 294, Aug. 1977.
[8]. A. M. Despain, "quick Fourier change calculations equipment for usage," IEEE Trans. Comput., vol. C-28, no. 5, pp. 333– 341, May 1979.