Implementation of Energy Efficient Partial FFT Processor for Wireless Communication System

A. Vimala, S.Manikandan, M.Darani kumar, S. Charumathi, A. Priyadharshini

Abstract: The Processor which is widely used in the orthogonal frequency division multiple access (OFDMA) communication system is the Fast Fourier Transform. To improve the transmission performance in OFDMA system, resource allocation is implemented. In this context, we designed and found the partial cached Fast Fourier Transform Processor which satisfies the purpose for the distribution of allocation resources to the user of the OFDMA system. We designed 128 point partial cached FFT Processor. This paper presented by using energy efficient partial FFT processor for wireless communication systems.

Index Words: DIT-Decimation in Time, FFT-Fast Fourier transform

I. INTRODUCTION

The OFDM technique has been widely employed in nowadays wireless communication systems. Now it is going to be used as OFDMA in the future-generation mobile communication systems, such as IEEE802.16e/m and the Third-Generation Partnership Project Long-Term Evolution (3GPP-LTE) standards. The FFT processor fills maximum cost and computational power in the OFDM baseband processor. In this system, only partial subcarriers needed to be computed in the user side. Since there is rise in the data growing demands, Communication and Multimedia has requested need for formulating many new different devices into a greater speed and efficiency bandwidth network capacity. In wireless Communication system, the Multiple Input Multiple schemes have been learned and attention have been received by academic as well as industry. The principles of OFDMA modulation have been followed for many decades. Nowadays these techniques does not exist in books and laboratories of research institute and is not practiced in nowadays communications systems. These techniques are practiced in delivery systems of data through the mobile line, wireless networking systems and digital television and radio.

The major cost and computation process is occupied by the FFT Processor in the OFDM baseband processor. In this design, the efficiency of FFT Processor power consumption is reliable because the most of the works focused on the low cost and power. Orthogonal Frequency Division Multiplexing system is well known for robust nature stands opposite to frequency selective fading channel. In the OFDM systems the Fast Fourier Transform processors are used as the demodulation or modulation kernels. In different application of OFDM FFT processors are varied according to size. Nowadays, multimedia like videos, high voice quality have been supported by wireless communication systems .Henceforth, it causes enormous growth demand for high speed, and efficient communication technology. The OFDM is an effective modulation type for satisfying demands. FFT or IFFT is regarded as the core in the OFDMA system.

II. FFT ALGORITHM

FFT algorithms can be generally classified as fixed-radix, split-radix and mixed-radix. These two basic two FFT types are decimation in time and decimation in frequency. Both of these relay in the recursive decomposition of a N point transform successively transformed into smaller subsequences. Based on complexity there is no difference between these two algorithms. The main difference between those two algorithms is, DIT acquires its name from the process of arranging the computation into tiny transformations, the sequence x(n) which is a time sequence that is successively decomposes into tiny subsequences. The working of DIT begins with the reverse ordered input and it generates the normal order output. DIF: the smaller sub sequences are obtained by decomposing the sequence of discrete Fourier transform coefficient X(k). DIF begins with the normal order input and it produces the output in bit reversed order. In FFT the Signal transformation from Time-domain to Frequency-domain takes place. The basic Butterfly structure and digit reversed sorting diagrams are given as follows.

Revised Manuscript Received on February 05 2019.

A. Vimala, Assistant Professor, Department of ETE, Karpagam College of Engineering, Coimbatore, India

S.Manikandan, Assistant Professor, Department of ETE, Karpagam College of Engineering, Coimbatore, India,

M.Darani kumar, Assistant Professor, Department of ECE, Karpagam Academy of Higher Education, Coimbatore, India

S. Charumathi, UG Student, Karpagam College of Engineering

A. Priyadharshini, UG Student, Karpagam College of Engineering, Coimbatore, India

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

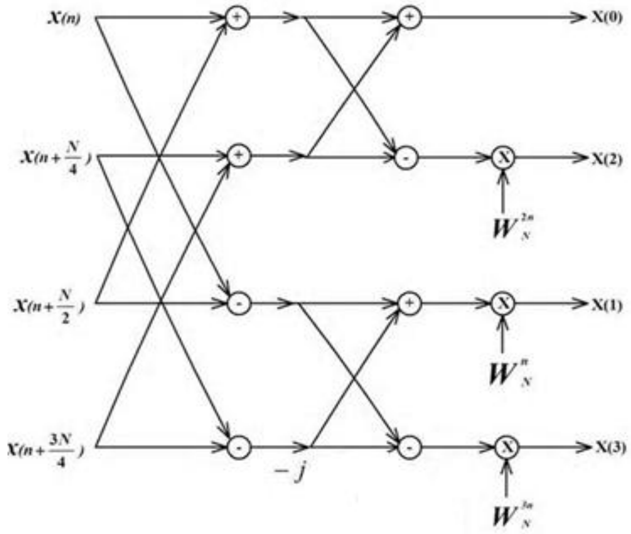


Fig. 1 The basic butterfly for radix-2 ^{2} DIF FFT algorithm

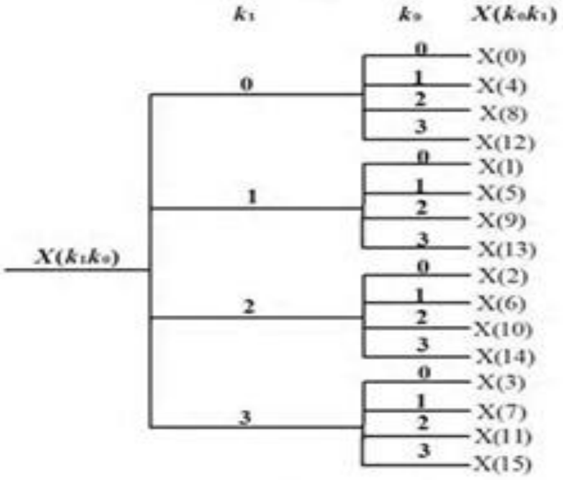


Fig. 2 Example Four diagram depicting digit-reversed sorting

III. THE PROPOSED ARCHITECTURE

The proposed system architecture used Radix 2^2 algorithm. It consists of RAM units, Butterfly units (BF1, BF2) and multiplier. The proposed architecture acts as a distributer and delivers the allocated resources to the user.

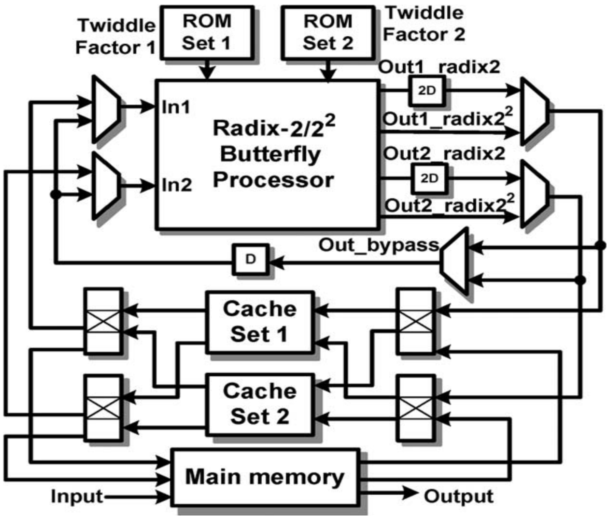


Fig. 3 Architecture of proposed cache FFT processor

This architecture is employed in the receiver side of the OFDMA communication system. In order to achieve the speed and the throughput of the processor the inputs of the architecture are given in multiple choices Proposed system of the architecture is shown in Fig.3.

IV. BUTTERFLY UNITS

The kernel of the FFT Processor is Butterfly units [4]. It consists of RAMs, Demux and Multiplexers. The input signal is fed to demux which next flows through adders and subtracters. MUX. gives the output. RAM is used to store the output of the subracters. Depending upon the select lines both the output and input lines are given. We have designed two types of similar Butterfly units. The random access memory (RAM) of the 1st Butterfly unit stores all of N/2th input data [7]. While, the Random Access Memory swaps the space in the buffer for the coming input data. When (N/2)+n th input data are given into the 1st Butterfly unit, x(n) input data which is stored in RAM are read and summed up with the new input data and next eliminated output data x[n]-x[N/2+N] is located in the place of the preceeding input data x(n). After the nth input operation, when the new input data are fed into the 1st Butterfly unit, the new incoming data are stored in the RAM and (x[n]-x[N/2+N]) RAM. Hence, 100% utilization rate can be achieved by the butterfly processor. To increase the latency for the dual streams , radix - 23 butterfly processor is used. Because for 100% utilization more number of delay registers must be inserted. The cost will be high if radix-4 and four streams are taken into use. Hence, cost and the throughput contribute to radix-2 / 22 dual-stream architecture.

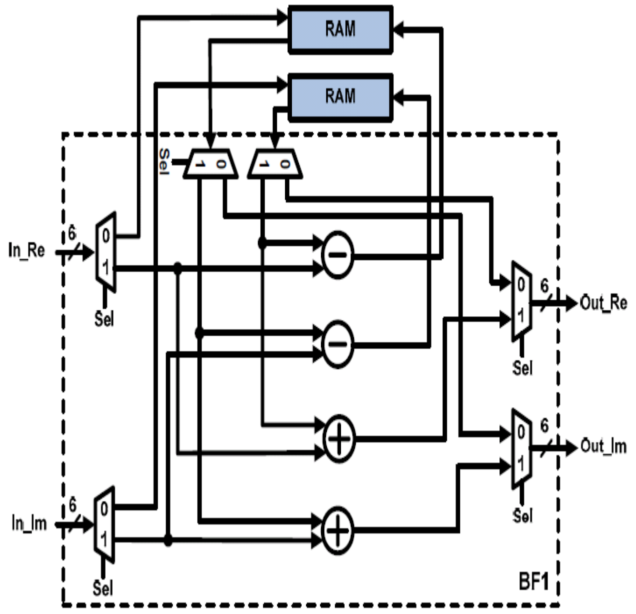


Fig. 4 Butterfly unit 1

The above figure shows the Butterfly unit 1. The Butterfly unit 2 is similar to that of the Butterfly unit ,but differs in the operation of 3N/4th .

The input is fed to the 2's complement and mux are added along in the Butterfly unit 2 through the input side. The Butterfly unit 2 is shown in Fig 5.

The process of combining the result of small discrete Fourier transforms to a larger Discrete Fourier Transform or viceversa have been done as a portion of the computation by the butterfly in Fast Fourier transform algorithms. Since the shape of the dataflow figure resembles butterfly shape, the name "butterfly" is used. In the Cooley – Tukey FFT algorithm the most commonly reffered term "butterfly" is used.It decomposes the Discrete Fourier Transform into smaller DFTs. The input sequence passing through the demux in the Butterfly unit is real and imaginary. The path of the output through the ram or adder and subtractor depends on the selection line input.

Multiplier

The requirement of word length is smaller than the full range. Corresponding LSB's are occupied with zeros based on the required resolution. Using 4-input AND and OR gates, dynamic range detector can be easily designed to create the power-aware signal by checking 4 MSB's of input signal. For the small-value mode, the final product is obtained by shifting the multiplier output right by 4 bits before the signal which given as input is moved left by four bits and occupies LSB's with corresponding zeros.
we sth

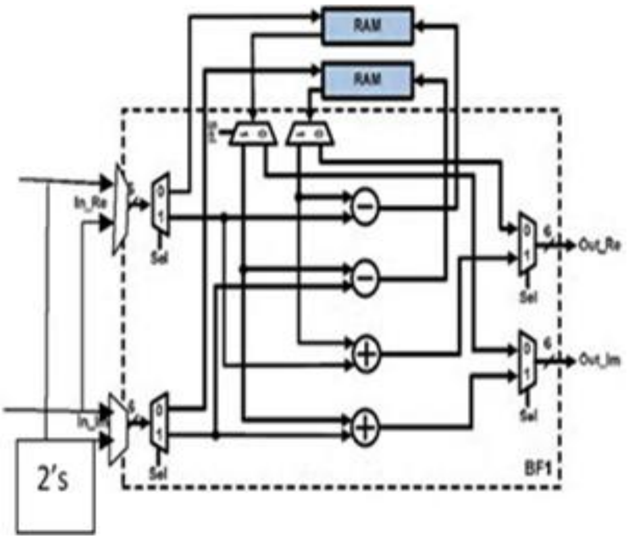


Fig. 5 Butterfly unit 2

Cache/ Memory Architecture

To enhance the transfer of data between the cache sets and the butterfly processor and between the cache sets and main memory we use two cache set. For the dual-stream processing, in one clock cycle the cache sets do performs 2 write operations and 2 read operations. To prevent the usage of the four port cache and reduce the control of complexity, the cache set is divided into two banks for even and odd addresses. To detect whether the data accessed in the c bank are correct and to change the access positions odd/even address detector are incorporated if necessary, as shown in Fig. 5. So, the computational time schedule can be carefully monitored to ensure that 1 even & 1 odd address is used for 2 read ports and 2 write ports. Like the architecture of cache, within one cycle the main memory needs to handle 2 read operations and 2 write operations. The defined architecture of cached-memory is same as the single memory architecture except that small cache memory lies between main memory and the processor .

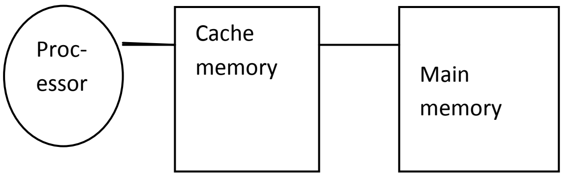


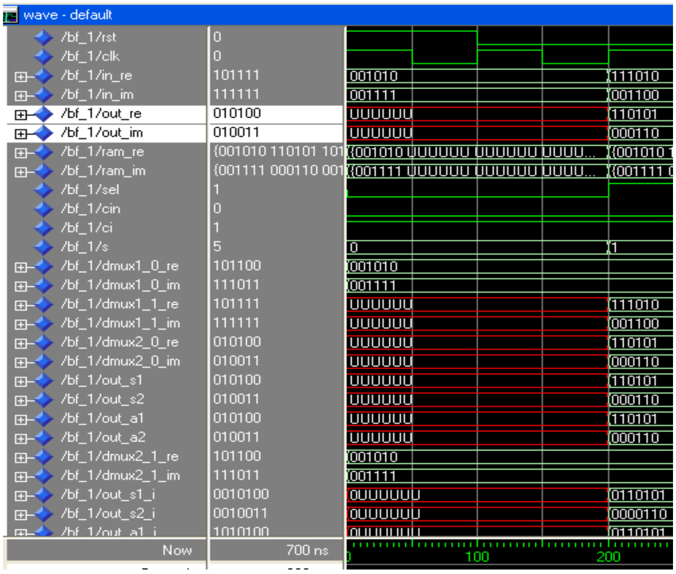
Fig. 6 Cached-memory block diagram

The above figure displays the heavily coupled processor cache pair and the N word main Memory. We already know that cache of data increases the bandwidth effectively to a memory only if the access of memory pattern shows enough locality. Even though all Fast Fourier Transform algorithms has bad locality, which clearly defines an algorithm that offer better locality over huge portions of the computation. In this context, global communication lies in the Fast Fourier Transform is grouped into a only few intermediate steps and accomplished through correct addressing while occupying and flushing the cache at ease. Since Fast Fourier Transform algorithm is deterministic, tags of cache are not necessary and correct cache operation is determined through the fixed, precalculated, cache address mapping. Since data flow is independent of data, data may be catched before from the main memory previously they are needed. The cached memory architecture gives 2 key advantages rather than other approaches. Especially, it yields:

- Increasing speed-Tiny memories are faster than huge ones,
- Efficiency in Increasing energy—Only tiny memories are required.

V. RESULT

Butterfly Unit 1



```
/bf_1/rst           0
/bf_1/clk           0
/bf_1/in_re         101111                  001010                                   111010
/bf_1/in_im         111111                  001111                                   001100
/bf_1/out_re        010100                  UUUUUU                                   110101
/bf_1/out_im        010011                  UUUUUU                                   000110
/bf_1/ram_re        {001010 110101 101      {001010 UUUUUU UUUUUU UUUU...            {001010 1
/bf_1/ram_im        {001111 000110 001      {001111 UUUUUU UUUUUU UUUU...            {001111 0
/bf_1/sel           1
/bf_1/cin           0
/bf_1/ci            1
/bf_1/s             5                       0                                        1
/bf_1/dmux1_0_re    101100                  001010
/bf_1/dmux1_0_im    111011                  001111
/bf_1/dmux1_1_re    101111                  UUUUUU                                   111010
/bf_1/dmux1_1_im    111111                  UUUUUU                                   001100
/bf_1/dmux2_0_re    010100                  UUUUUU                                   110101
/bf_1/dmux2_0_im    010011                  UUUUUU                                   000110
/bf_1/out_s1        010100                  UUUUUU                                   110101
/bf_1/out_s2        010011                  UUUUUU                                   000110
/bf_1/out_a1        010100                  UUUUUU                                   110101
/bf_1/out_a2        010011                  UUUUUU                                   000110
/bf_1/dmux2_1_re    101100                  001010
/bf_1/dmux2_1_im    111011                  001111
/bf_1/out_s1_i      0010100                 0UUUUUU                                  0110101
/bf_1/out_s2_i      0010011                 0UUUUUU                                  0000110
/bf_1/out_a1_i      1010100                 0UUUUUU                                  0110101
                Now 700 ns                  0                  100                   200
```

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VI. CONCLUSION

For OFDMA Communication system a 128 point partial cached FFT Processor is presented. The need for huge speed processing of data, power efficient and complexity of lower hardware can be determined and the power transitions can be observed which will be useful for understanding the performance of the system in real world implementation in this architecture. Approximately, power manipulation is done to find the amount of power required for implementation. The power dissipation ranges from 50-60%. Henceforth, this Fast Fourier Transform processor is necessarily efficient for OFDMA receiver realization.

REFERENCES

1. S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in Proc. ISSSE, 1998, pp. 257–262.
2. S. Lee and S. C. Park, "Modified SDF architecture for mixed DIF/DIT FFT," in Proc. IEEE Int. Conf. Circuits Syst., May 2007, pp. 2590–2593.
3. Chao-Ming Chen, Chien-Chang Hung and Yuan-Hao Huang, "An Energy-Efficient partial FFT processor for the OFDMA Communication system" in proc. IEEE trans. Corcuits and Systems, vol. 57 no. 2, pp. 136-140, Feb 2010.
4. C. T. Lin, Y. C. Yu, and L. D. Fan, "A low-power 64-point FFT/IFFT design for IEEE 802.11a WLAN application," in Proc. IEEE Int. Conf. Circuits Syst., May 2006, pp. 4523–4526.
5. R. Min, M. Bhardwaj, and A. Chandrakasan, "A partially operated FFT/IFFT processor for low complexity OFDM modulation and emodulation of WiBro in-car entertainment system," IEEE Trans. Consum. Electron., vol. 54, no. 2, pp. 431–436, May 2008.
6. C. P. Fan and G. A. Su, "A grouped fast Fourier transform algorithm design for selective transformed outputs," in Proc. IEEE APCCAS, 2006, pp. 1939–1942.
7. Hang Liu and Hanho Lee, "High speed four-parallel 64 point radix 2^4 MDF FFT / IFFT processor for MIMO-OFDMA system", in proc.IEEE Int Conf. Computers and Communication ,2008, pp. 1469-1472.
8. Bevan.M.Bass" A Low-Power, High-Performance, 1024-Point FFT Processor " in Proc. IEEE journ, Solid States and Circuits vol.34. no.3, pp 380-387, march 1999.
9. R. Min, M. Bhardwaj, and A. Chandrakasan, "Quantifying and enhancing power awareness of VLSI systems," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 9, no. 6, pp. 757–772, Dec. 2001.
10. T. Lenart and V. Owall, "Architectures for dynamic data scaling in2/4/8 K pipeline FFT cores," IEEE Trans. Very Large Scale Integr. (VLSI)Syst., vol. 14, no. 11, pp. 1286–1290, Nov. 2006.
11. C80216m-08_503, Motorola IEEE 802.16 m Downlink Resource Mapping, IEEE, May 2008.
12. 3GPP, R1-071091, Philips Resource-Block Mapping of Distributed Transmissions in E-UTRA Downlink, Feb. 2007.