VLSI IMPLEMENTATION AND PERFORMANCE ANALYSIS OF EFFICIENT MIXED-RADIX 8-2 FFT ALGORITHM WITH BIT REVERSAL FOR THE OUTPUT SEQUENCES.

M. MOHAMED ISMAIL
(Research scholar- JNTU)
Associate Prof./ECE Dept,
B.S.Abdur Rahman University
Formerely,B.S.A. Crescent
Engg.College
Chennai-48. India

Dr. M.J.S RANGACHAR
Professor/ ECE Dept
B.S.Abdur Rahman University
Formerely,B.S.A. Crescent
Engg.College
Chennai-48. India

Dr.Ch. D. V. PARADESI RAO
Prof. (Retd),
ECE Dep
JNTU,
Hyderabad-72

E-mail: m_md_ismail@yahoo.com

ABSTRACT

In recent years, as a result of advancing VLSI technology, OFDM has received a great deal of attention and been adopted in many new generation wideband data communication systems such as IEEE 802.11a, HiPerLAN/2,digital audio/video broadcasting (DAB/DVB-T) and asymmetric digital subscriber line (ADSL), very high speed digital subscriber line (VDSL) in wireless and wired communications, respectively. In these communication systems the different applications mentioned above have different demands in operation speed and length of FFT/IFFT. The modified Mixed Radix 8-2 Butterfly FFT with bit reversal for the output sequence derived by index decomposition technique is our proposed VLSI system architecture to design the prototype FFT/IFFT processor for OFDM systems.

In this paper the analysis of several FFT algorithms such as radix-2, radix-4, split radix and mixed radix 4-2 and proposed mixed radix8-2 were designed using VHDL and outputs are analysed. Mainly
the results show that the proposed processor architecture can save the area approximately 5% and power more than 40% when compared to basic radix 2 system, which may be attractive for many real-time systems. Reduction in area and power can further leads to less implementation cost. Also the proposed algorithm makes an offer the simple bit reversal mechanism which is only supported by a fixed radix FFT algorithm.

Key words: *OFDM, FFT/IFFT, VLSI, VHDL, Mixed Radix with bit reversal.*

1. INTRODUCTION

Fast Fourier transform (FFT) are widely used in different areas of applications such as communications, radars, imaging, etc. One of the major concerns for researchers is the enhancement of area reduction, power reduction and processing speed. Several methods of for computing FFT (and IFFT) are discussed in [Shousheng He and Mats Torkelson April. 1996, pp.776-780 and Shousheng. He and Mats Torkelson May. 1998, pp. 131-134]. These are basic algorithms for implementation of FFT and IFFT blocks. While there has been extensive research on the theoretical efficiency of these algorithms (traditionally algorithms have been compared based on their floating point operation counts), there has been little research to-date comparing algorithms on practical terms. The choice of the best algorithm for a given platform is still not easy because efficiency is intricately related to how an algorithm can be implemented on a given architecture.

In this paper, a mixed radix 8-2 butterfly structure with simple bit reversing for output sequences derived by index decomposition technique is presented. The new method to obtain the mixed radix butterfly structure with simple bit reversing for output sequence is established. Therefore the proposed mixed radix 8-2 offers an engineering insight of general mixed radix.

2. FAST FOURIER TRANSFORM

The Discrete Fourier Transfer (DFT) plays an important role in many applications of digital

signal processing including linear filtering, correlation analysis and spectrum analysis etc.
The DFT is defined as:

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{nk} \qquad k=0,1,...,N-1$$

(1) Where $W_N^{nk}=e^{-j\frac{2\pi}{N}nk}$ is the DFT coefficient.Evaluating the Equation (1) directly requires N complex multiplications and (N-1) complex additions for each value of the DFT. To compute all *N* values therefore requires a total of *N*^2 complex multiplications and *N*(N-1) complex additions. Since the amount of computation, and thus the computation time, is approximately proportional to *N*^2, it will cost a long computation time for large values of N. For this reason, it is very important to reduce the number of multiplications and additions. This algorithm is an efficient algorithm to compute the DFT [C. Sidney Burrus-1977 and Lihong Jia, Yonghong GAO, Jouni Isoaho, and Hannu Tenhunen-1998], which is called Fast Fourier Transform (FFT) algorithm or radix-2 FFT algorithm, and it reduce the computational complexity from (N^2) to (N $\log_2$(*N*).

3. MIXED RADIX 4-2

A mixed radix algorithm is a combination of different radix-r algorithms. That is, different stages in the FFT computation have different radices. For instance, a 64-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. When compared to split Radix[Daisuke Takahashi-2001], mixed radix system provides regular butterfly structure and it is easy for implementation.

4. MIXED-RADIX 4-2 FFT ALGORITHMS WITH BIT REVERSING

The Mixed-Radix 4-2 butterfly with simple bit reversing output sequences is induced by transforming a one-dimensional array into three-dimensional arrays uniquely. The necessary and sufficient conditions for the unique and one-to-one index mapping are proposed in the paper [C. Sidney Burrus-1977]. By using the Common Factor Algorithm (CFA) [[Shousheng He and Mats Torkelson April. 1996, pp.776-780, Shousheng. He and Mats Torkelson May. 1998, pp. 131-1341 and C. Sidney Burrus-1977] one-dimensional array can be mapped into three dimensional arrays.

The mixed-radix 4/2 [Byung G. Jo and Myung H. Sunwoo -2005] butterfly unit is shown in Figure 1. It uses both the radix-2^2 and the radix-2 algorithms can perform fast FFT computations and can process FFTs that are not power of four. The mixed-radix 4/2, which calculates four butterfly outputs based on X(0)~X(3). The butterfly unit [Gordon L. Demuth -1989 and Kyung L. Heo, Jae H. Baek, Myung H. Sunwoo, Byung G. Jo, and Byung S. Son,-2003] has three complex multipliers and eight complex adders. Four multiplexers represented by the solid box are used to select either the radix-4 calculation or the radix-2 calculation.

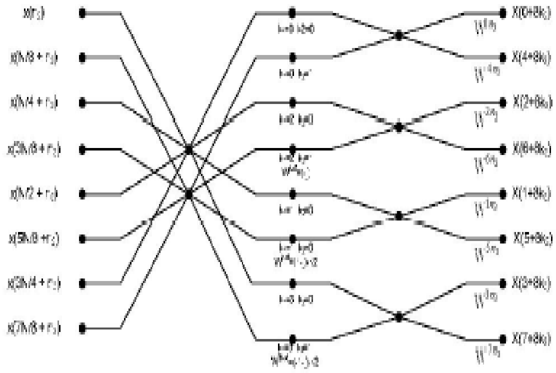

Fig 1: The basic butterfly for mixed-radix 4/2 DIF FFT algorithm.

5. MIXED-RADIX 8-2 FFT WITH BIT REVERSING OUTPUT SEQUENCES FOR 64 POINTS FFT

The basic butterfly for the proposed mixed radix 8-2 is shown in the Figure 2. The proposed Mixed-Radix 8-2 is composed of one radix-8 butterflies and four radix-2 butterflies [Young-jin Moon, Young-il Kim-2006]. The Mixed-Radix 4-2 butterfly with simple bit reversing output sequences is induced by transforming a one-dimensional array into three-dimensional arrays uniquely. The necessary and sufficient conditions for the unique and one-to-one index mapping are proposed in the paper [C. Sidney Burrus -1977]. By using the Common Factor Algorithm (CFA) [Shousheng He and Mats Torkelson April. 1996, pp.776-780, Shousheng. He and Mats Torkelson May. 1998, pp. 131-1341 and C. Sidney Burrus-1977], one-dimensional array can be mapped into three dimensional.

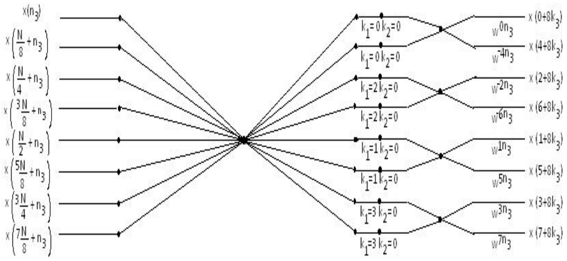

Fig 2: The basic butterfly for mixed-radix 8-2 FFT algorithm.

In order to verify the proposed scheme, 64-points FFT based on the proposed Mixed-Radix 8-2 butterfly with simple bit reversing for ordering the output sequences is exampled. As shown in the Figure 3, the block diagram for 64-points FFT is composed of total six-teen Mixed-Radix 8-2 Butterflies. In the first stage, the 64 point input sequences are divided by the 8 groups which correspond to n3=0, n3=1, n3=2, n3=3, n3=4, n3=5, n3=6, n3=7 respectively. Each group is input sequence for each Mixed-Radix 8-2 Butterfly. After the input sequences pass the first Mixed-Radix 8-2 Butterfly stage, the order of output value is expressed with small number below each butterfly output line as shown in the figure 3.

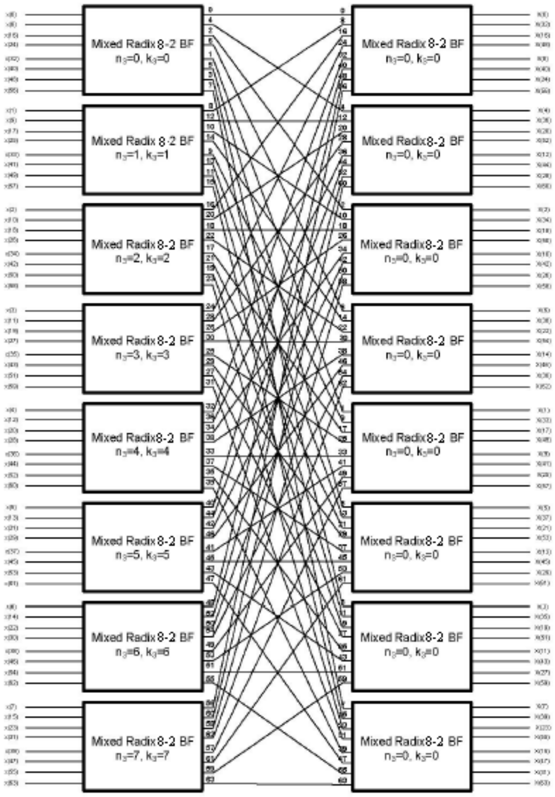
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=1, k3=1
Mixed Radix 8-2 BF n3=2, k3=2
Mixed Radix 8-2 BF n3=3, k3=3
Mixed Radix 8-2 BF n3=4, k3=4
Mixed Radix 8-2 BF n3=5, k3=5
Mixed Radix 8-2 BF n3=6, k3=6
Mixed Radix 8-2 BF n3=7, k3=7
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=0, k3=0
Mixed Radix 8-2 BF n3=0, k3=0

Fig.3: Proposed Mixed-Radix 8-2 Butterfly for 64 point FFT

The proposed Mixed-Radix 8-2 is composed of one radix-8 butterflies and four radix-2 butterflies. The figure 4 shows the SFG for 64 point FFT of the proposed mixed radix 8-2 FFT algorithm. In the first stage, the input data of one radix-8 butterflies which are expressed with the equation B4 (o, n3, kj) B4 (i, n3, k1), are grouped with the x (n3), x (N/4±n3), x(N/2±n3), x(3N/4±n3) and x(N/8±n3),x (3N/8±n3),x(5N/8±n3),x(7N/8±n3) respectively.
After the each input group data passes the first radix-8 butterflies, the outputted data is multiplied by the special twiddle factors. Then, these outputted sequences are inputted into the second stage which is composed of the radix-2 butterflies. After passing the second radix-2 butterflies, the outputted data are multiplied by the twiddle factors. These twiddle factors WQ (1+k) is the unique multiplier unit in the

proposed Mixed-Radix 8-2 Butterfly with simple bit reversing the output sequences. Finally, we can also show order of the output sequences in Fig.4 above. The order of the output sequence is 0,4,2,6,1,5,3 and 7 which are exactly same at the simple binary bit reversing of the pure radix butterfly structure. Consequently, proposed mixed radix 8-2 butterfly with simple bit reversing output sequence include one radix 8 butterflies, four radix 2 butterflies, one multiplier unit and additional shift unit for special twiddle factors. The proposed Mixed Radix 8-2 butterfly unit [Gordon L. Demuth,-1989 and Kyung L. Heo, Jae H. Baek, Myung H. Sunwoo, Byung G. Jo, and Byung S. Son-2003], has two complex multipliers and eight complex adders.

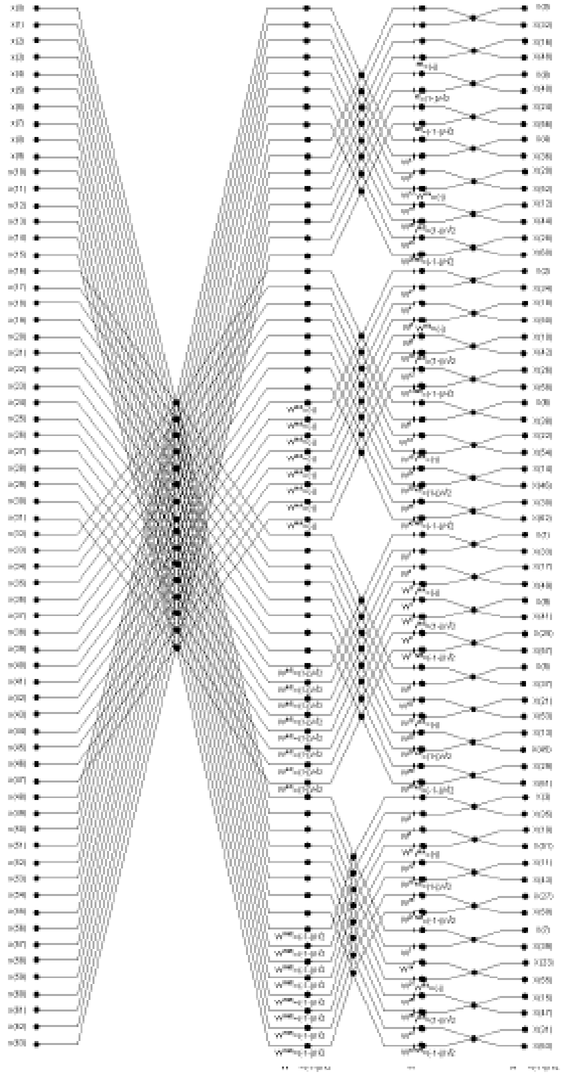

Fig.4: Mixed-Radix 8-2 SFG for 64 Points FFT

6. RESULT

Employing the parametric nature of this core, the FFT block is synthesized on one of Xilinx's Virtex-II Pro (2V6000ff1517) FPGAs with different configurations. The results of logic synthesis for 64 point FFT of Radix-2, Radix-4, split Radix, mixed radix 4-2 and Mixed Radix 8-2 are presented in Table 1. The 64-point FFT is chosen to compare the number of CLB slices (responsible to occupy an area in the FPGA). The results shows that the proposed processor architecture can save the area approximately 5% and power more than 40% when compared to basic radix 2 system, which may be attractive for many real-time systems. Reduction in area and power can further leads to less implementation cost.

64 point FFT	CLB Slices/7680	Utilization factor	Power in mW
Radix-2 FFT	851	11.1%	4685.60mW
Radix-4 FFT	765	9.96%	3012.51mW
Split Radix	835	10.8%	4492.40mW
Mixed Radix 4-2 FFT	750	9.77%	3831.63mW
Mixed Radix 8-2 FFT	596	7.76%	2696.49mW

Table 1: Comparison of FFT Algorithm based on CLB Slices, Device Utilization and Power.

7. CONCLUSION:

The FFT algorithms considered here include radix-2, radix-4, split-radix 2/4, mixed-radix 4/2 and mixed radix 8-2 which is the subject of this study. The hardware implementation for radix-2 FFT algorithm is the easiest but it is the least efficient. Split-radix 2/4 FFT algorithm is more efficient but its algorithm cannot produce regularity in hardware structure, thus not amenable to implementation. In contrast, mixed-radix 8-2 FFT algorithm is capable of producing hardware with structural regularity. It is more efficient than radix-2 FFT algorithm and applicable to all 2n-point FFT systems. This Algorithm makes an offer the simple bit reversing for ordering the output sequences which is only supported by a fixed-radix FFT algorithm. Moreover, the proposed technique makes an offer systematic viewpoint in the Mixed-Radix algorithm. The proposed Mixed Radix 8-2 butterfly unit has only two complex multipliers and eight complex adders. Reduction in area and power can further leads to less implementation cost. It is therefore can be selected for implementation of a OFDM based processor.

REFERENCES:

[1] Byung G. Jo and Myung H. Sunwoo,2005, "New Continuous-Flow Mixed-Radix (CFMR) FFT Processor Using Novel In-Place Strategy", IEEE Transactions on Circuits and Systems, Vol. 52, No. 5.

[2] Daisuke Takahashi,2001,"An Extended Split-Radix FFT Algorithm", IEEE Signal Processing Letters, Vol. 8, No. 5, pp. 145-147.

[3] Gordon L. Demuth,1989 "Algorithms for Defining Mixed Radix FFT Flow Graphs", IEEE Trans Acoust, Speech, and Signal Processing, Vol. 37, No. 9, pp. 1349-1358.

[4] Kyung L. Heo, Jae H. Baek, Myung H. Sunwoo, Byung G. Jo, and Byung S. Son,2003 "New In-Place Strategy for a Mixed-Radix FFT processor", IEEE SOC Conference, pp. 8 1-84.

[5] Lihong Jia, Yonghong GAO, Jouni Isoaho, and Hannu Tenhunen, 1998 "A New VLSI-Oriented FFT Algorithm and Implementation", IEEE ASIC Conf., pp. 337-341.

[6] Shousheng He and Mats Torkelson, 1996 "A New Approach to Pipeline FFT Processor", IEEE Parallel Processing Symposium, pp.776-780.

[7] Shousheng. He and Mats Torkelson,1998 "Design and Implementation of a 1024-point Pipeline FFT Processor", IEEE Custom Integrated Circuits Conference, pp. 131-134.

[8] C. Sidney Burrus, 1977 "Index Mapping for Multidimensional Formulation of the DFT and Convolution", IEEE Trans. Acoust., Speech, and Signal Processing, Vol. ASSP-25, pp. 239-242.

[9] Young-jin Moon, Young-il Kim Portable Internet system research team Electronics and Telecommunications Research Institute (ETRI) Daejeon, Korea”2006, A Mixed-Radix 4-2 Butterfly with Simple Bit Revering for Ordering the Output Sequences”, ICA0T2006,pp1774-1778.