International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 4 (2012), pp. 513-520

http://www.irphouse.com

VLSI Implementation of OFDM using Efficient Mixed-Radix 8-2 FFT Algorithm with bit reversal for the output sequences.

M. Mohamed Ismail, Dr. M.J.S Rangachar and Dr.Ch. D. V. Paradesi Rao

(Research scholar- JNTU) Associate Prof./ECE Dept,
Formerely, B.S.A. Crescent Engg.College Chennai-48. India
Professor/ ECE Dept B.S.Abdur Rahman University
Formerely,B.S.A. Crescent Engg.College Chennai-48. India
Prof. (Retd), ECE Dept JNTU, Hyderabad
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.FFT and IFFT are the main arithmetic kernel in the OFDM system. This paper discusses the computational complexity of several Fast Fourier Transform (FFT) algorithms, analyzes the design procedure for implementing these algorithms into hardware, and then proposes one that applies to OFDM communication system. The Mixed Radix 8-2 Butterfly FFT with bit reversal for the output sequence derived by index decomposition technique is our proposed 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 the proposed mixed radix 8-2 were designed using VHDL and their performance are analysed. Then these FFTs are applied to OFDM system and it is implemented using VLSI design technique. The results show that the proposed processor architecture can save the area of approximately 7% and power more than 40%, which may be attractive for many real-time systems.

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

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 to meet real-time processing requirements and to reduce hardware complexity mainly with respect to area and power and to improve processing speed over the last decades. However pipeline algorithm may provide optimum solution for speed, but it occupies more area for different stages used in the pipeline also increase the implementation complexity. Several methods of for computing FFT (and IFFT) are discussed in [1], [2], [3]. 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 proposed mixed radix 8-2 offers an engineering insight of general mixed radix.

2. FAST FOURIER TRANSFORM- Radix-2, Radix-4 and Split Radix

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:

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{nk} \qquad k=0,1,\ldots,N-1 \qquad W_N^{nk}=e^{-j\frac{2\pi}{N}nk} \tag{1}$$

Where the X (k) and x (n) are frequency-domain sequences and time-domain sequence. 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], 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)*. There are two well-known types of decompositions called Decimation In Time (DIT) and Decimation In Frequency (DIF) FFT. Throughout this paper, we will handle the DIF type of decomposition.

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 FFT algorithm is capable of producing efficient hardware with structural regularity. It is more efficient than radix-2 FFT algorithm and applicable to all 2n-point FFT systems. It is therefore selected for implementation of the OFDM systems.

3. 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 [4]. By using the Common Factor Algorithm (CFA) [1], [2], [3], and [4].

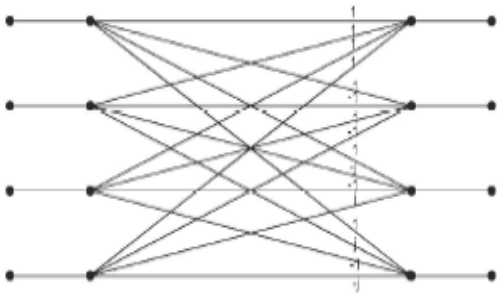

Fig 1a: Modified Radix-4-2 butterfly

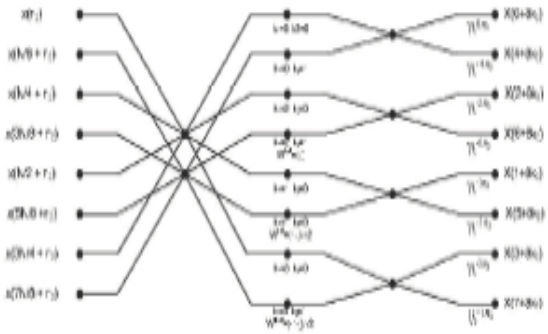

Fig 1b: Mixed Radix 4-2 butterfly

The modified mixed radix 4-2 butterfly structure for adapting simple bit revering output is shown in fig.1a.The signal flow graph (SFG) of the mixed radix 4-2 butterfly structure is shown fig.1b.

4. Mixed-Radix 8-2 FFT with Bit Reversing output sequences for 64 points FFT

The Signal Flow Graph (SFG) of the proposed butterfly structure is shown in the Figure 2.

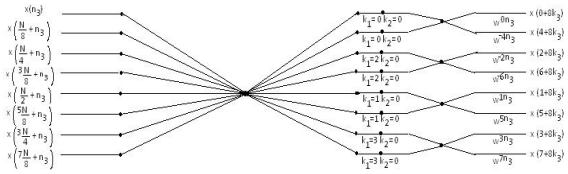

Fig 2: The basic butterfly for mixed-radix 8-2 FFT algorithm.

The proposed Mixed-Radix 8-2 is derived from Mixed-Radix 4-2 butterfly with simple bit reversal for the output sequence [12]. It is composed of one radix-8 butterflies and four radix-2 butterflies.

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 considered. As shown in the Figure 3, the block diagram for 64-points FFT is composed of total six-teen Mixed-Radix 8-2 Butterflies.

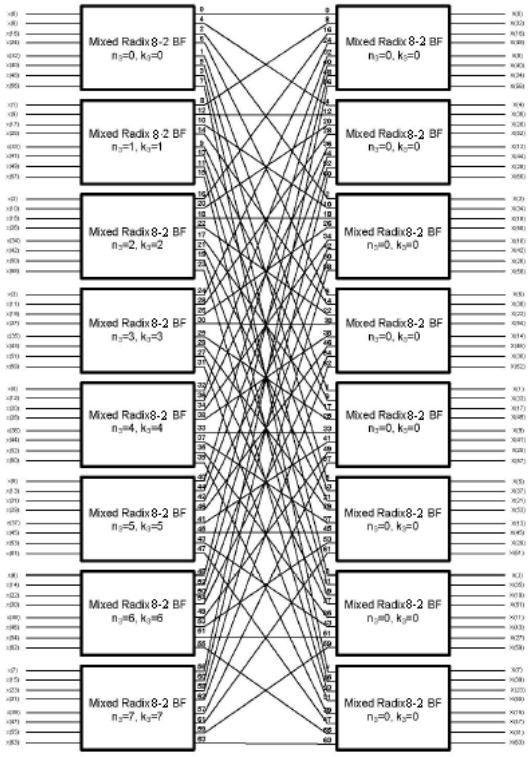

Fig.3: Proposed Mixed-Radix 8-2 Butterfly for 64 point FFT

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 in the figure 3.

From the SFG of Mixed–Radix 8-2 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) [12], 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[12]. After the each input group data passes the first radix-8 butterflies, the outputted data is multiplied by the special twiddle factors. Then, these output sequences are applied as input 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. which shows the SFG for 64 point FFT of the proposed mixed radix 8-2 FFT algorithm. 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 [10], [11] has two complex multipliers and eight complex adders.

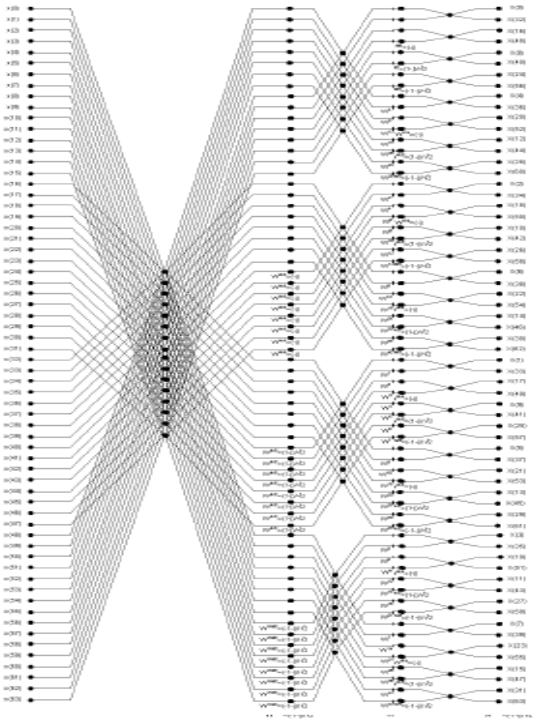

Fig.4: Mixed-Radix 8-2 SFG for 64 Points FFT

5. Implementation of Mixed-Radix 8-2 FFT in OFDM System

The Frequency Division Multiplexing (FDM) system had been able to use a set of subcarriers that were orthogonal to each other; a higher level of spectral efficiency could have been achieved. The guard bands that were necessary to allow individual demodulation of subcarriers in an FDM system would no longer be necessary. The use of orthogonal subcarriers would allow the subcarriers' spectra to overlap, thus increasing the spectral efficiency. As long as orthogonality is maintained, it is still possible to recover the individual subcarriers' signals despite their overlapping spectrums. If the dot product of two deterministic signals is equal to zero, these signals are said to be orthogonal to each other. Orthogonality can also be viewed from the standpoint of stochastic processes. If two random processes are uncorrelated, then they are orthogonal. Given the random nature of signals in a communications system, this probabilistic view of orthogonality provides an intuitive understanding of the implications of orthogonality in OFDM. Recall from signals and systems theory that the sinusoids of the FFT form an orthogonal basis set, and a signal in the vector space of the FFT can be represented as a linear combination of the orthogonal sinusoids.

One view of the FFT is that the transform essentially correlates its input signal with each of the sinusoidal basis functions. If the input signal has some energy at a certain frequency, there will be a peak in the correlation of the input signal and the basis sinusoid that is at that corresponding frequency. This transform is used at the OFDM transmitter to map an input signal onto a set of orthogonal subcarriers, i.e., the orthogonal basis functions of the DFT. Similarly, the transform is used again at the OFDM receiver to process the received subcarriers.. The signals from the subcarriers are then combined to form an estimate of the source signal from the transmitter. The orthogonal and uncorrelated nature of the subcarriers is exploited in OFDM with powerful results. Since the basic functions of the FFT are uncorrelated, the correlation performed in the FFT for a given subcarrier only sees energy for that corresponding subcarrier. The energy from other subcarriers does not contribute because it is uncorrelated. This separation of signal energy is the reason that the OFDM subcarriers' spectrums can overlap without causing interference. The fig 5 shows that block diagram of OFDM transmitter and receiver.

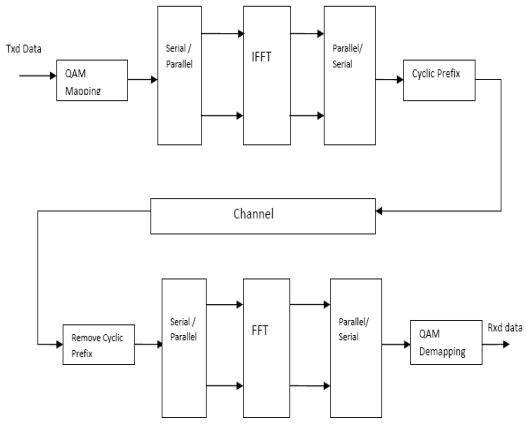

Fig.5 OFDM Transmitter and Receiver

The OFDM is implemented using VLSI design technique with various FFT architectures and the results are discussed in the next section.

6. RESULT

Employing the parametric nature of this core, the OFDM block is synthesized on one of Xilinx's Virtex-II Pro (2V6000ff1517) FPGAs with different configurations. The results of logic synthesis and power analysis for various 64 point FFT architectures and then FFT based orthogonal frequency division multiplexing (OFDM) using Radix-2, Radix-4, split Radix, mixed -radix 4-2 and the proposed mixed- radix 8-2 are presented in Table 1 & 2. We analyzed the 64-point FFT alone and then implemented the FFT in OFDM to compare the number of CLB slices, utilization factor and Power. The results shows that the proposed processor architecture can save the area approximately 7% and power more than 40% when compared to basic radix 2 system, which may be attractive for many real-time systems.

Table 1: Comparison of 64-point FFT Algorithm based on CLB Slices, Utilization factor and power.

64 point FFT	CLB Slices/7680	Utilization factor	Power in mW
Radix-2 FFT	851	11.1%	2188.75mW
Radix-4 FFT	765	9.96%	1799.50mW
Split Radix	835	10.8%	2152.45mW
Mixed Radix 4-2 FFT	750	9.77%	1872.88mW
Mixed Radix 8-2 FFT	596	7.76%	899.24mW

Table 2: Comparison of OFDM with 64-point FFT Algorithm based on CLB Slices, Utilization factor and power.

OFDM with 64 point FFT	CLB Slices/7680	Utilization factor	Power in mW
Radix-2 FFT	1213	15.8%	4685.60mW
Radix-4 FFT	1104	14.4%	3012.51mW
Split Radix	1196	15.6%	4492.40mW
Mixed Radix 4-2 FFT	1088	14.2%	3831.63mW
Mixed Radix 8-2 FFT	620	8.1%	2696.49mW

7. Conclusion:

In this paper, we designed an OFDM receiver with different FFT algorithms and they are implemented using VLSI design process. It was found during the algorithm design that many blocks need complex multipliers and adders and therefore special attention needs to be given to optimize these circuits and maximize reusability. In particular, the models have been applied to analyze the performance of mixed-radix FFT architectures used in OFDM. Actual hardware resource requirements were also presented and simulation results were given for the synthesized design. The 64-point

Mixed-Radix 8-2 FFT based OFDM architecture was found to have a good balance between its performance and its hardware requirements and is therefore suitable for use in OFDM systems.

8. References

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

[2] Shousheng He and Mats Torkelson, "Designing Pipeline FFT Processor for OFDM (de)Modulation", IEEE Signals, Systems, and Electronics, Sep. 1998, pp. 257-262.

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

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

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

[6] Martin Vetterli and Pierre Duhamel, "Split-Radix Algorithms for Length-ptmDFT's", IEEE Trans. Acoust, Speech, and Signal Processing, Vol. 37, No. 1, Jan. 1989, pp. 57-64.

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

[8] 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, pp. 499-506.

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

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

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

[12] Young-jin Moon, Young-il Kim, “A Mixed-Radix 4-2 Butterfly with Simple Bit Revering for Ordering the Output Sequences”, ISBN 89-551 9-1 29-4, Feb. 20-22, 2006 ICA0T2006.