VLSI Implementation of a 2x2 MIMO-OFDM System on FPGA

M.Jasmin

Assistant Professor ECE Dept Bharath university, Chennai

Abstract--- *Multiple Input Multiple Output Orthogonal Frequency Division Multiplexing (MIMO-OFDM) technology is an attractive transmission technique for wireless communication systems with multiple antennas at transmitter and receiver. The core of this technology is that it divides one data stream to many. Hence, data rate, reliability and diversity can be increased along with the stability for multi- path signals. FPGA implementation is carried with good channel estimation method, efficient FFT/IFFT processor and better coding techniques. This work describes the efficient implementation of a Low-Power 64-point Pipeline FFT/IFFT processor adopting a single-path delay feedback style. The proposed architecture applies a reconfigurable complex multiplier and bit-parallel multipliers to achieve a ROM-less FFT/IFFT processor, thus consuming less power. Header-based channel estimation with maximum likelihood algorithm is chosen in consideration of hardware feature as well as communication theory for fast prototyping. The pipeline architecture here includes the simple logic of one adder and channel memories without redundancy. Thus reducing the complexity from O* (n^2) *to O (1), it saves 43 percent of the hardware resources and achieves a better performance in the architecture.*

Keywords--- *MIMO-OFDM, FFT/IFFT, Channel Estimation, FPGA*

I. INTRODUCTION

The growing demand for high system capacity, high transmission rate and broadband access motivates researches to search a new technology, resulting in MIMO- OFDM technique. The advantages of MIMO-OFDM are that it uses bandwidth more efficiently without increasing the transmission power, and combats the inter-symbol interference (ISI) effect and multi-path effect. OFDM is a promising approach for high delay spread channels or high data rate with low complexity of equalizers. Therefore MIMO-OFDM has become one of the most promising physical layer schemes in the fourth generation wireless communication systems.

The Fast Fourier Transform (FFT) was proposed by Cooley and Turkey to efficiently reduce the time complexity. For hardware implementation, various FFT processors have been proposed. These implementations can be mainly classified into memory-based and pipeline architecture styles. Memory-based architecture is widely adopted to design an FFT processor, also known as the single processing element (PE) approach. This deign style is usually composed of a main PE and several memory units, thus the hardware cost and the power consumption are both lower than the other architecture style. However, this kind of architecture style has long latency, low throughput, and cannot be parallelized. On the other hand, the Pipeline architecture style can get rid off the disadvantages of the foregoing style, at the cost of an acceptable hardware overhead. Generally, the pipeline FFT processors have two popular design types. One uses single- path delay feedback (SDF) pipeline architecture and the other uses multiple-path delay commutator (MDC) pipeline architecture. The single-path delay feedback (SDF) pipeline FFT is good in its requiring less memory space and its multiplication computation utilization being less than 50%, as well as its control unit being easy to design. Such implementations are advantageous to low- power design. Based on these reasons, the SDF pipeline FFT is adopted. The proposed architecture includes a reconfigurable complex constant multiplier and bit parallel complex multipliers instead of using ROM's to store twiddle factors.

In MIMO-OFDM systems, the information in the channel matrix is essential for decoding the transmitted message correctly. If the channel matrix is not estimated accurately, the channels cannot be fully decoupled at the receiver and the spatial streams become coupled. In this paper, channel estimator for MIMO-OFDM is implemented on FPGA. The system takes very low hardware resource utilization. The first requirement for reducing hardware resource is to select proper algorithm. Channel estimator uses header-based Maximum Likelihood (ML) method, in this approach special training symbols are transmitted before data packets are transmitted. Once training symbols are received and detected in receiver, the received data are modulated and preprocessed in frequency domain. Hence, the channel frequency response is obtained in frequency domain, which is helpful to reduce computational complexity. Channel estimation block requires a number of operations and occupy many hardware resources and should be implemented efficiently with consideration of hardware area and performance.

This paper is organized as follows. In section II, the system block diagram of the MIMO-OFDM is shown. Section III, presents FFT/IFFT architecture for application in wireless communication systems. In section IV, channel estimation for MIMO-OFDM is described. In section V, Simulation results are shown. Finally, Conclusions in section VI.

II. SYSTEM BLOCK DIAGRAM

The architecture of the transmitter and receiver is illustrated in Fig.1 and Fig. 2 respectively. First of all, bit stream which has been scrambled and interleaved is separated into spatial streams by stream parser. Secondly, spatial streams are mapped into constellation. Thirdly, the points on the constellation are through the STBC encoder to transform the spatial streams to space-time streams. Fourthly, spatial mapper maps space-time streams into transmit chains. And lastly, the transmit chains are inserted pilot, IFFT modulated, added CP (Cyclic Prefix), then transmitted through the DUC, DAC and RF modules. The transmitted signals are received through RF modules, ADC, DDC and remove CP. The received chains are FFT modulated, pilot extraction, channel estimation and passed through the STBC decoder to transform to spatial streams from space-time stream. The spatial streams are demapped, interleaved, and descrambled to get the original bit stream.

A. Scrambler

A scrambler is a device that manipulates a data stream before transmitting. A scrambler can be placed just before an encoder, or it can be placed after the encoder, just before the modulation. The manipulations are reversed by a descrambler at the receiving side. Scrambling is accomplished by the addition of data to the original data or the changing of some important data of the original signal in order to make extraction of the original signal difficult.

B. Digital Up/Down Converter

The digital up converter (DUC) and the digital down converter (DDC) are important components of this system. The carrier modulation/demodulation is realized by mixer and multiplier with the frequency of 69MHz. The sample rate up/down is implemented through cascading the Cascaded Integrator Comb (CIC) filters, compensation filter and matched filter. CIC filter performs sample rate conversion by using only additions and subtractions not multipliers. But it has a passband droop shortcoming, so we need a corresponding compensation filter to make the passband flat. The matched filter is used to make the transmitter and the receiver paired up and it can improve the SNR by reducing the noise.

C. Cyclic Prefix and Preamble

The cyclic prefix refers to the prefixing of data with a repetition of the end. As a guard interval, it eliminates the intersymbol interference from the previous data. It allows the linear convolution of a frequency-selective multipath channel to be modelled as circular convolution, which in turn may be transformed to the frequency domain using FFT. This approach allows for simple frequency-domain processing, such as channel estimation and equalization. Preamble is used for synchronization and channel estimation at the receiver.

D. Space-Time Block Coding/Decoding

Alamouti’s transmit diversity scheme with two transmit antennas and two receive antennas are used in this paper. Alamouti’s scheme is a space-time block code and suitable when two transmit antennas and an arbitrary number of receive antennas are used. As the main and most important characteristic of Alamouti’s scheme is simple in the coding and decoding. Space–time blockin wireless communications to data stream across a number coding is a technique used transmit multiple copies of a of antennas and to exploit the various received versions of the data to improve the reliability of data-transfer.

E. Demapper

QAM demapper method is used to reduce the complexity of implementation and resources occupancy.

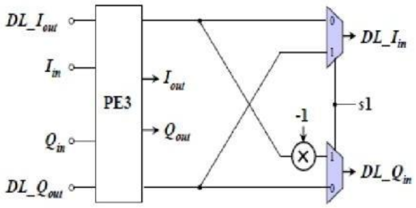
III. PROPOSED FFT/IFFT ARCHITECTURE

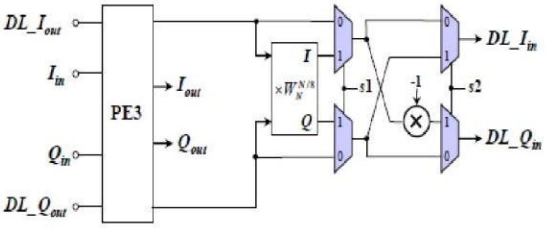
Traditional hardware implementation of FFT/IFFT processors usually employs a ROM to look up the wanted twiddle factors, and then word length complex multipliers to perform FFT computing. However, this introduces more hardware cost, thus a bit-parallel complex constant Multiplication scheme is used.

Besides, since the twiddle factors have a symmetric property, the complex multiplications used in FFT

$$W_N^k.(a+jb) = W_N^{k-\left(\frac{N}{4}\right)}.(b-ja),\ N/4 < k < N/2$$
$$W_N^k.(a+jb) = -W_N^{k-\left(\frac{N}{2}\right)}.(a+jb),\ N/2 < k < 3N/4$$
$$W_N^k.(a+jh) = -W_N^{k-\left(\frac{3N}{4}\right)}.(h-ja),\ 3N/4 < k < N$$

Arbitrary twiddle factor used in FFT can utilize these operation types to derive the wanted value, thus can significantly shorten the size of ROM used to store the twiddle factors. To further decrease the size of ROM, two extra operation types are included.





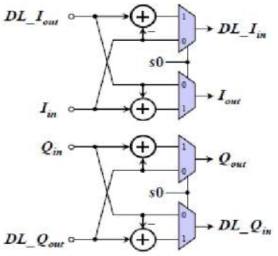


Figure 5: Circuit Diagram of PE2 Stage Figure 3: Radix-2 64 Point Pipeline FFT/IFFT

Type 4: $1 < k < N/4$ (4)

Type 5: (5)

$$W_N^k.(a+jb) = [W_N^{\left(\frac{N}{4}\right)-k}.(b+ja)]^*,$$
$$[W_N^{\left(\frac{N}{2}\right)-k}.(b+ja)]^*,\ N/4 < k < N/2$$

A radix-2 64-point pipeline FFT/IFFT processor with low power consumption, as shown in Fig.3. The proposed architecture is composed of three different types of processing elements (PEs), a complex constant multiplier, delay-line (DL) buffers (as shown by a rectangle with a number inside), and some extra processing units for computing IFFT. Here, the conjugate for extra processing units is easy to implement, which only takes the 2's complement of the imaginary part of a complex value. The divided-by-64 module can be substituted with a barrel shifter. In addition, for a complex constant multiplier in Fig.3, a novel reconfigurable complex constant multiplier is used to eliminate the twiddle-factor ROM. This new multiplication structure thus becomes the key component in reducing the chip area and power consumption of our proposed FFT/IFFT processor.

A. *Processing Elements*

Based on the radix-2 FFT algorithm, the three types of processing elements (PE3, PE2, and PE1) used in our design are illustrated in Fig. 4, Fig. 5, and Fig. 6, respectively. First,

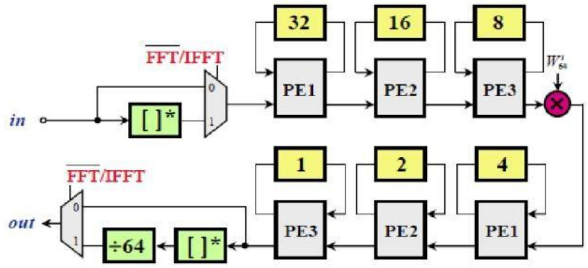


Figure 6: Circuit Diagram of PE1 Stage

The PE3 stage is used to implement a simple radix-2 butterfly structure only, and serves as the submodules of the PE2 and PE1 stages. In the figure, I_{in} and I_{out} are the real parts of the input and output data, respectively. Q_{in} and Q_{out} denote the image parts of the input and output data, respectively.

Similarly, DL_I_{in} and DL_I_{out} stand for the real parts of input and output of the DL buffers, and DL_Q_{in} and

DL_Qout are for the image parts, respectively.

B. *Reconfigurable Complex Constant Multipliers*

Based on Equations (6)-(10), a reconfigurable low- complexity complex constant multiplier for computing is proposed, as shown in Fig. 7 and Fig. 8. This structure of this complex multiplier also adopts a cascaded scheme to achieve low-cost hardware. Here, the meaning of two input signals (I_{in} and I_{out}) and two output signals (Q_{in} and Q_{out}) are the same as the signals in the PE1 stage.

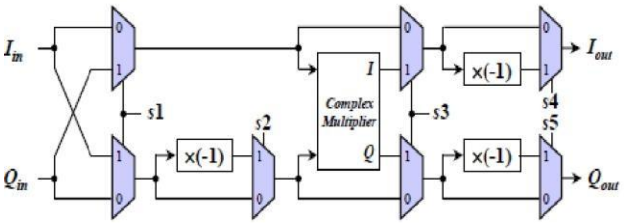
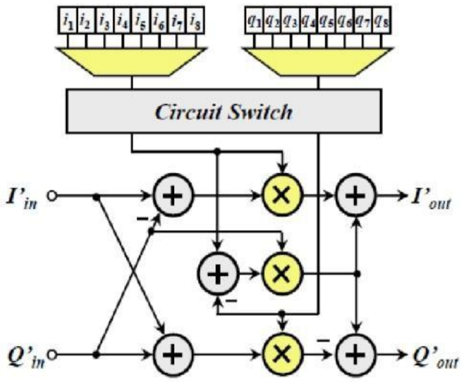


Figure 7: Reconfigurable Complex Constant Multiplier


i_1 i_2 i_3 i_4 i_5 i_6 i_7 i_8
q_1 q_2 q_3 q_4 q_5 q_6 q_7 q_8
Circuit Switch
I'_{in}
Q'_{in}
I'_{out}
Q'_{out}
–

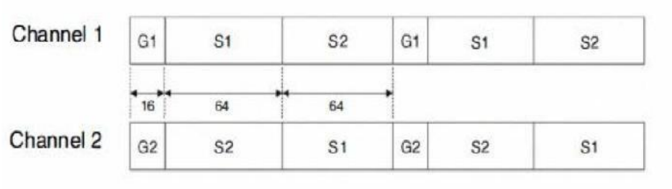


Figure 8: Complex Multiplier in Fig. 8, this circuit is responsible for the computation of multiplication by twiddle factor in Fig.7, which is also W_{64} an important circuit of FFT/IFFT processor. The word length multiplier used in Fig.8 adopts a low-error fixed-width booth multiplier for hardware cost reduction.

I. CHANNEL ESTIMATION FOR MIMO-OFDM

Header-based channel estimation with Maximum Likeli- hood algorithm is chosen to satisfy both the performance and less hardware resource utilization. Suppose X is a matrix of symbols being transmitted on a sub-carrier where the channel matrix is H. Then the received matrix of symbols R is given by

$$R=HX+W \qquad (6)$$

where W is the white Gaussian noise vector at receiver. If we know X and R, the ML estimate is given by equation (7)

Structure of channel estimation training symbols for 2 x 2 MIMO-OFDM is presented in Fig.9, where S1 and S2 indicate tone sequence 1 and tone sequence 2. Gl and G2 are the guard band of S1 and S2, respectively. Guard band is added for making cyclic extension of FFT symbols to avoid Inter Symbol Interference (ISI). S1, S2, G1, and G2 are repeated once again. Extra 3dB combining gain can be obtained by sending the same symbol two times. The number of arithmetic operation per the whole channel estimation symbols in Fig.9 is

$$(64 \text{ multiplications} + 64 \text{ additions}) \times 4 \qquad (12)$$

where 4 is the number of channels in 2 x 2 MIMO-OFDM. Finally, 160 MOPS can be obtained from equation (12). This algorithm can be extended to support N x N MIMO-OFDM in the same way as 2 x 2 MIMO-OFDM. That is, when X is unitary matrix in N-dimensions, Hij where i, j =0, 1, ... ,N can be obtained by using N sequences without decoding mixed channel.

General concept to implement channel estimation algorithm on hardware is to deploy each processing logic and memory onto hardware domain as shown in Fig.10. Each subcarrier delivers 1 bit signal as shown in equation (11), adder and multiplier required for equation (10) should become simpler. ROM Txl and Tx2 store sequence SI and S2, respectively. Switch block selects either ROM Txl or ROM Tx2 when either SI or S2 is received. RAM is needed to store intermediate channel coefficients which are calculated over maximum 4 frames. However, the baseline architecture still consists of O (n^2) calculation blocks and memories in Figure

Instead, propose a pipelined architecture which uses only one block instead of O (n^2) redundant blocks in Fig.11. 4µsec is taken for a pipeline stage period between RAM1 and RAM2 because a transmission period per one group of modulated data is 4 µsec.

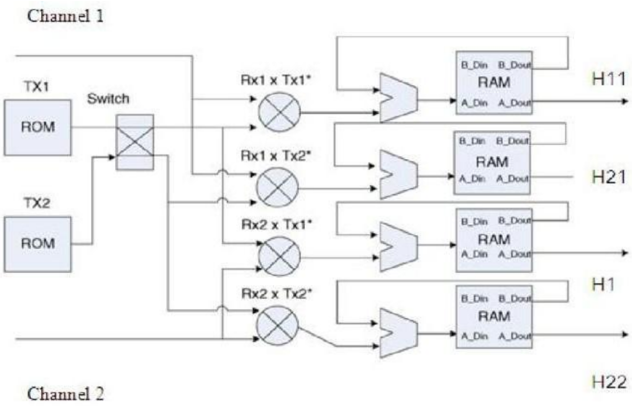


Figure 10: Architecture of Baseline Channel

VI. CONCLUSION

A flexible MIMO-OFDM system is presented in this paper. Some necessary transmitter and receiver algorithms, such as OFDM synchronization, channel estimation, encoding and decoding of STBC, modulation, demapper, scrambler, descrambler, interleaver, de-interleaver and other components are implemented in FPGA.

A novel ROM-less and low-power pipeline 64-point FFT/IFFT processor for MIMO-OFDM applications have been described to lower hardware cost and power consumption compared to other architectures. An efficient channel estimation block for 2 x 2 MIMO-OFDM implementation results in terms of resource utilization as well as performance. More channels can be supported by using our proposed architecture. Channel estimation block for 2 x 2 MIMOOFDM will be implemented and combined with 3 x 3 MIMO-OFDM transceiver proposed in [15] as a future work.

REFERENCES

[1] Sampath, H.,Talwar, S.,Tellado, J., "A fourth- generation MIMO-OFDM broadband wirelesssystem: design, performance, and field tria l results," Communications Magazine, IEEE , Volume: 40 , Issue: 9, 2002.

[2] Ventura, L.M., Nieto, X., Gregoire, J.P., " A Broadband Wireless MIMO-OFDM Demonstrator. Design and Measurement Results," Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on 11-14 Sept. 2006.

[3] S. He and M. Torkelson, "Designing Pipeline FFT Processor for OFDM (de)Modulation," in Proc. URSI Int. Symp. Signals, Systems, and Electronics, vol. 29, Oct.1998, pp. 257-262.

[4] H.L. Groginsky and G.A. Works, "A pipeline fast Fourier transform," IEEE Transactions on Computers, vol. C-19, no. 11, pp. 1015-1019, Nov. 1970.

[5] Y. Jung, H. Yoon, and J. Kim, "New efficient FFT algorithm and pipeline implementation results for OFDM/DMT applications," IEEE Transactions on Consumer Electronics, vol. 49, no. 1, pp. 14-20, Feb. 2003.

[6] A.Wenzler and E. Luder, "New structures for complex multipliers and their noise analysis," in Proc. IEEE Int. Symp. on Circuits and Systems, May 1995, vol. 2, pp. 1432–1435.

[7] Wei Han, T. Arslan, A.T. Erdogan, M. Hasan, "A novel low power pipelined FFT based on subexpression sharing for wireless LAN applications," in Proc. IEEE Workshop on Signal Processing Systems,2004, pp. 83-88.

[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, pp. 499-506, July 2005.

[9] Koushik Maharatna, Eckhard Grass, and Ulrich Jagdhold, "A 64-Point fourier transform chip for high-speed wireless LAN application using OFDM," IEEE Journal of Solid-State Circuits, vol. 39, no. 3, pp. 484-493, Mar. 2004.

[10] Chu Yu, Yi-Ting Liao, Mao-Hsu Yen, Pao-Ann Hsiung, and Sao-Jie Chen, "A Novel Low-Power 64- point Pipelined FFT/IFFT Processor for OFDM Applications," in Proceeding IEEE Int'l Conference on Consumer Electronics. Jan. 2011, pp. 452-453.

[11] Yuan Chen, Yu-Wei Lin, and Chen-Yi Lee, "A Block Scaling FFT/IFFT Processor for WiMAX Applications," in Proceeding IEEE Asian Solid-state Circuits Conf., 2006, pp. 203-206.

[12] K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, “Design of low-error width modified Booth multiplier,” IEEE Trans. Very Large Scale Integration Systems, vol. 12, no. 5, pp. 522–531, May 2004.

[13] Praveen Bagadi and S. Das, "MIMO-OFDM Channel Estimation using Pilot Carries," International Journal of Computer Applications, vol. 2, pp. 81-88,May 2010.

[14] S.Tiiro, J. Y lioinas, M. MyllyHi, and M. Juntti, "Implementation of the least squares channel estimation algorithm for MIMO-OFDM systems," ITG Workshop on Smart Antennas, pp. 155-161,Feb. 2009.

[15] J. S. Park, T. Ogunfunmi, "FPGA implementation of the MIMO-OFDM Physical Layer using single FFT multiplexing," in Proceeding of IEEE International Symposium on circuits and Systems (ISCAS), 2010.

[16] S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” IEEE Journal on Selected Areas in Communication, vol. 16, pp. 1451-1458, Oct. 1998.

[17] Ian Griffiths,“FPGA Implementation Of MIMO Wireless Communications System”,University Of New Castle,Australia,1st November, 2005.

[18] Changchuan Yin, Jingyu Li, Xiaolin Hou and Guangxin Yue, “Pilot Aided LS Channel Estimation in MIMO-OFDM Systems,”8th International Conference on Signal Processing,vol.3, pp.(16-20), 2006.

[19] En Zhou, Xiaolin Hou, Jianping Chen, “FPGA Implementation and Experimental Performaces of a Novel Timing Synchronization Method in MIMO- OFDM Tesbed,” Proc. of 14th Asia-Pacific Conference on Communications, APCC 2008. Oct. 2008.

[20] Junior, L.H.M., Junior, R.R.S., Silveira, M,“ An FPGA implementation of Alamouti's transmit diversity technique applied to an OFDM system,” Antennas and Propagation Society International Symposium 2006, pp. 149 – 152.

[21] S. Yoshizawa and Y.Miyanaga, "VLSI Implementation of a 4x4 MIMO-OFDM transceiver with an 80-MHz channel bandwidth," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on, vol., no., pp. 1743-1746,24-27 May 2009.

[22] S. Haene, D. Perels, and A. Burg "A Real-Time 4-Stream MIMOOFDM Transceiver: System Design, FPGA Implementation, and Characterization," IEEE Journal on Selected Areas in Communications, vo1.26, no.6, pp.877-889,August 2008.