Low utilization FPGA implementation of OFDM transceiver based on IEEE 802.11n standard

Ahmed El-Gohary[1], Mohamed Saad[1], Omar Mahmoud[1], Mohamed Thabet[1], Sayed Shaban[1], Mohannad Khaled[1], and Hassan Mostafa[1,2]

[1]Electronics and Communications Engineering Department, Cairo University, Giza 12613, Egypt.
[2]University of Science and technology, Nanotechnology and Nanoelectronics Program, Zewail City of Science and Technology, October Gardens, 6th of October, Giza 12578, Egypt.
ahmedhani123@yahoo.com, mohamedmsaad66@gmail.com, omar.mahmoud.yahya@gmail.com, mohamedthabet0000@yahoo.com, sayedshaban193@gmail.com, mohanedeece@gmail.com, hmostafa@waterloo.ca

***Abstract*—Orthogonal Frequency Division Multiplexing (OFDM) is considered as one of the most important techniques in the wireless communication field, it's used by many standards such as Wi-Fi standards 802,11a/n/ac. The basic principle of OFDM is transmitting data by dividing the stream into several parallel bit streams and modulating each of these data streams onto individual orthogonal carriers or subcarriers. This paper includes the design and the implementation of an OFDM transceiver based on 802.11n standard. All the modules are designed using Verilog hardware description language. The transceiver is implemented on FPGA, then tested and compared with a MATLAB reference model, achieving lower utilization than most of the pervious related work presented before.**

***Index Terms*—OFDM, FPGA, IFFT, FFT, Viterbi, IEEE 802.11n**

I. Introduction

OFDM technology, also known as orthogonal frequency division multiplexing, plays a significant role in modern telecommunications, including Wi-Fi standards like 802.11a/n/ac. It has also been chosen for the cellular telecommunications standard like LTE / LTE-A. OFDM has been used in many high data rate wireless systems because of the many advantages it provides, like immunity to selective fading, spectrum efficiency and resilient to ISI and more.

The basic principle of OFDM is to split a high-rate data stream into a number of lower rate streams that are transmitted simultaneously over a number of subcarriers. This is achieved by making all the subcarriers orthogonal to one another [1]. The work includes the implementation of main blocks of the OFDM system, including scrambler/de-scrambler, convolutional encoder and Viterbi decoder, interleaver/de-interleaver, pilots insertion, IFFT/FFT and guard interval insertion/removal [2].

Presented work is divided as follows: Section II presents the idea and implementation of each block, section III presents the verification methodology of the design and section IV concludes the results.

II. Blocks Design and Implementation

This section includes the description and implementation of each block in the transmitter and its inverse in the receiver. The general block diagram is shown in figure 1.

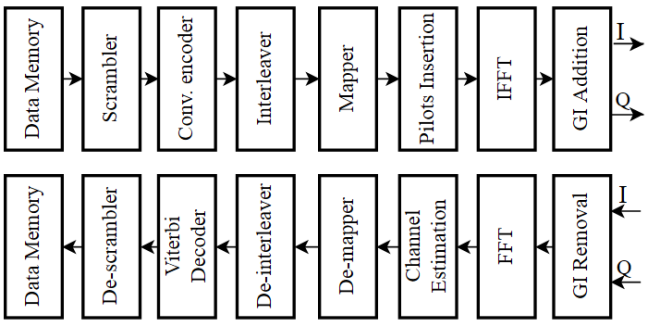

Figure 1 Base-band blocks for transmitter and receiver

A. Scrambler

The purpose of scrambler is to prevent a long sequence of 1s or 0s and to ensure the randomness of the data. This helps the timing recovery at the receiver. The implementation of scrambler consists of a shift register and XOR gates according to the polynomial function. The original data is being XORed with the generated pseudo-random sequence. The descrambler is exactly the same in order to retrieve the original data.

B. Encoder and Decoder

1) Convolutional Encoder

Encoding is used to add redundancy to the transmitted bits, redundancy adds extra information, and the extra information ensures that the data is received correctly. In 802.11n standard the convolutional encoder shall use the industry-standard generator polynomials, g0 = 133_8 and g1 = 171_8, of rate R = 1/2, as shown in figure 2. The bit denoted as "A" shall be output from the encoder before the bit denoted as "B".

978-1-7281-1184-1/19/$31.00 ©2019 IEEE

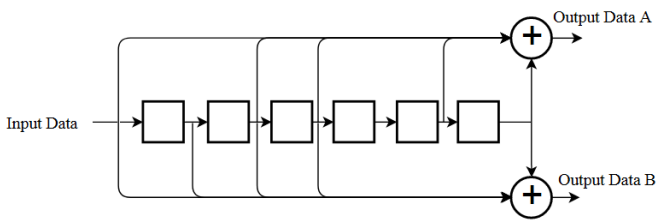

Figure 2 The convolutional encoder

2) Viterbi Decoder

Convolutional encoding and Viterbi decoding [3] are widely used in modern digital communication systems to achieve low bit error data transmission. Targeting high data throughput through evolutional wireless standards has required very high speed and low-power decoders. The represented receiver design uses the conventional Viterbi decoder with simple Add-Compare-Select (ACS) units. However, the main improvement in area and speed lies in the Trace-Back Unit (TBU). The conventional Viterbi decoding consists of three main stages; updating metrics, finding maximum metric and finally tracing back the best path of this maximum metric. The second stage can be explained as a traversal of data linked-list given the tail address and given that each node has address to the previous node. In the investigated designs in the literature review, it's noticed that the complete address is stored in the TBU memory, which is 6-bit in the IEEE 802.11n standard, then the message extraction is done using those 6-bit addresses. However, by looking to a simple trellis diagram as shown in figure 3, it could be seen that each point on the trellis has only two previous pointers, and those points could be calculated using the address of this point and a bit that selects between the two points.

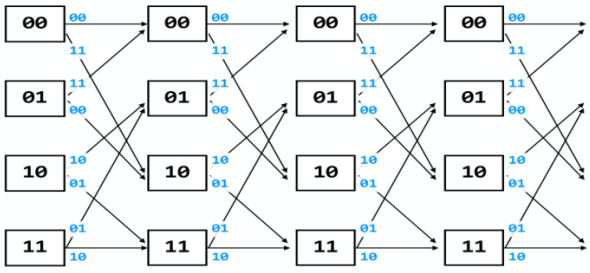

Figure 3 Simple trellis diagram for a 4-state encoder

In the proposed design, the tail address is only used as it is, which is the address of the winning path, plus a simple circuit which decides the next address. The first decoding stage thus stores only 1 bit that decides which previous address to calculate, so instead of storing 6 bits in each element, we only store 1 bit, which significantly enhances the maximum frequency and the utilized area.

C. Interleaver and De-Interleaver

Interleaving is a process which disperses the positions of the data bits before transmission so that the corrupted information can be recovered at the receiver by rearranging the data. Interleaver block size is corresponding to the number of bits in OFDM symbol, NCBPS. The Interleaver is defined by two permutations; the first permutation causes adjacent coded bits to be mapped onto non-adjacent subcarriers. The second one causes adjacent coded bits to be mapped alternately onto less and more significant bits of the constellation and, thereby, long runs of low reliability (LSB) bits are avoided. Figure 4 shows the hardware implementation. The De-Interleaver does the inverse operation using different equations. The implementation idea can be simplified to writing data onto memory and reading it again using different patterns. The data is being written in rows in a 2-dimensional memory, and read as columns from the same memory. To maintain constant throughput, the memory can be replicated to read from one and write into the other one simultaneously, and vice versa.

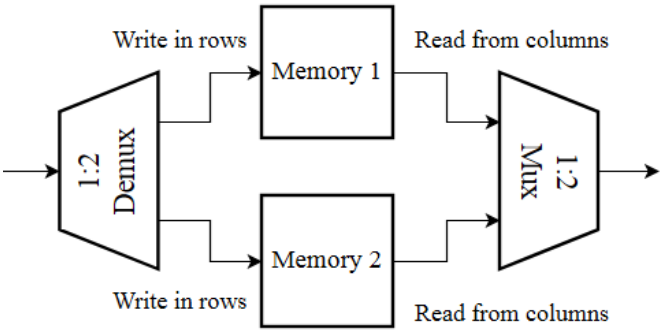

Figure 4 Interleaver hardware implementation

D. Mapper and De-Mapper

1) Mapper

The mapper can use BPSK, QPSK, 16-QAM and 64-QAM depending on the modulation scheme. The input encoded and interleaved bit stream shall be divided into groups of NBPSC (1, 2, 4, or 6) gray encoded bits then converted into complex numbers representing BPSK, QPSK, 16-QAM or 64-QAM symbols. Finally a weighting factor is used to normalize the energy according to the used scheme.

2) De-Mapper

This module have the inverse operation of the mapper module, In the transmitter's side each group of bits are represented in one symbol, now in the receiver's side this effect should be reversed, so each symbol should be converted back to its bits. In order to do that the constellation is divided into areas each area is bounded by threshold values. By comparing each symbol with the threshold values the corresponding bits can be obtained.

E. Pilots insertion and removal

Pilots are known values that have specific locations in the OFDM symbol. They are inserted to monitor the channel response variations with time through measuring their values in the receiver, while zeros are inserted to minimize adjacent channel interference as stated before. The OFDM symbol in the frequency domain thus consists of 108 data symbols, 6 pilots, and 14 zeros. The implementation is done using a MUX that selects between the mapper's output, zeros or the pilots according to the sub-carrier index. The MUX's output is connected to a 1-entry FIFO. The FIFO's writing clock is the bit-domain clock, and its reading clock is the sample-domain clock. The FIFO is necessary to prevent timing violations and functional errors that occur in clock-domain crossing.

F. IFFT/FFT

The frequency domain sub-carriers shall be transformed to time domain using 128-point Inverse Discrete Fourier Transform (IDFT) operation. To implement the operation in hardware, the Fast Fourier Transform (FFT) algorithm is used. The main idea behind the FFT algorithm is the decomposition of N-point DFT into 2 N/2-point DFTs, through (1), (2), and (3).

$$\boldsymbol{X(k)} = \sum_{n=0}^{N-1} \boldsymbol{x(n)} e^{\frac{-j2\pi nk}{N}} \qquad (1)$$

$$\boldsymbol{X(2k)} = \sum_{n=0}^{\frac{N}{2}-1} (\boldsymbol{x(n)} + \boldsymbol{x(n+N/2)}) e^{\frac{-j2\pi n(k)}{N/2}} \qquad (2)$$

$$\boldsymbol{X(2k+1)} = \sum_{n=0}^{\frac{N}{2}-1} ((\boldsymbol{x(n)} - \boldsymbol{x}\left(\boldsymbol{n}+\frac{N}{2}\right)) e^{\frac{-j2\pi n}{N}}) e^{\frac{-j2\pi n(k)}{N/2}} \qquad (3)$$

Equation (1) represents the DFT operation, while equations (2) and (3) can be derived easily from (1). Equation (2) states that the even frequency samples can be obtained from a simple addition followed by the N/2-point DFT of the result, and similarly, equation (3) states that the odd frequency samples can be obtained from subtraction, complex rotation, and N/2-point DFT of the results. This can be represented in the signal flow graph in figure 5.

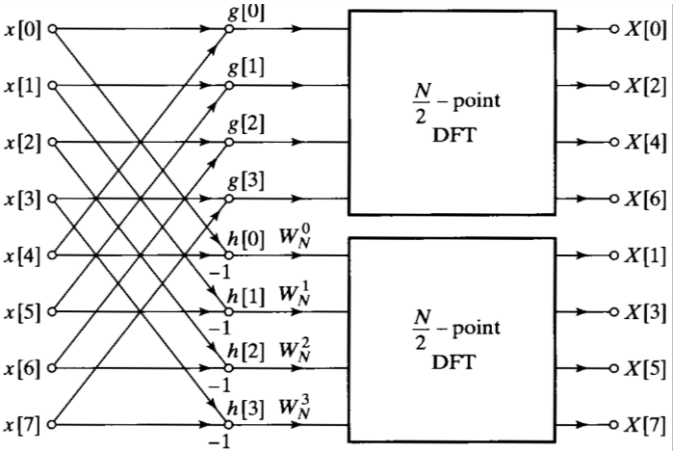

Figure 5 IFFT/FFT N/2 decomposition

For an 8-point DFT, the structure will consist of 3 stages; each one decomposes the DFT until the last stage is a 2-point DFT which is only addition and subtraction. Since the required FFT block is 128-point, then the hardware implementation will consist of 7 stages. Since the main goal is to have least possible utilization, Radix-2 Single-path Delay Feedback (R2SDF) architecture was chosen for this block, which is based on having the signals propagate in a single path that has a delayed feedback. [4]. Figure 6 shows the block diagram of a single stage.

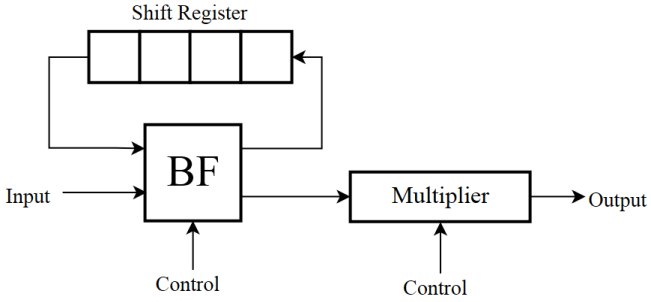

Figure 6 Single IFFT/FFT stage

The IFFT/FFT stage consists of a register, butter-fly and a multiplier. The register is used to add a delay for samples, since all operations are done on a sample and a delayed one. The butter-fly is used for addition/subtraction operations. Finally the multiplier is used for the complex exponential term, a rotational CORDIC [5] is used to minimize the area with acceptable errors.

G. GI addition and removal

Guard interval is provided before the data period given by IFFT so that the ISI occurs in the guard interval which can be removed afterwards and the data can be retrieved, this makes high level of robustness against multipath delay spread of OFDM system. The guard period gives time for multipath signals from the previous symbol to decay before the information from the current symbol is gathered.

Cyclic Prefix (CP) is the insertion of the last portion of the OFDM symbol inside the Guard Interval. The CP is inserted to extend the periods of the subcarrier sinusoids in time domain to prevent Inter-Carrier Interference (ICI), as shown in figure 7.

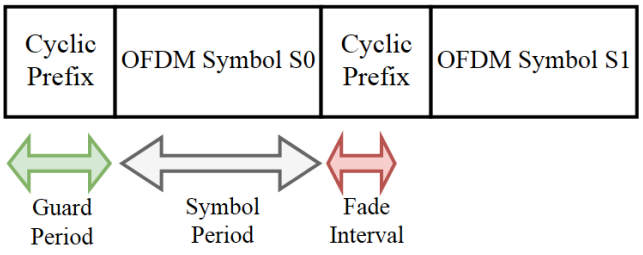

Figure 7 Cyclic prefix in OFDM symbols

For ensuring the guard insertion works with one symbol per clock cycle, the architecture uses two memory modules simultaneously. One module is used to write over, while the other is used to read from, and vice versa. The IFFT bit-reverses the data in the OFDM symbol, that's why guard insertion stores the data with bit-reversed address and reads in order with a simple counter. When the data is read, the read address starts from either 112 in case of short GI, or 96 in case of long GI. When the CP is outputted, a hold signal is outputted from the GI to the entire previous chain as feedback to stop the input stream until all the CP's samples are outputted from the transmitter.

Concerning the Guard removal, it is the inverse of guard insertion, as it removes the cyclic prefix added in the transmitter. The idea is based on stopping the chain until the OFDM symbol comes at the input port, using a counter and a hold signal that stops the chain for number of cycles according to the guard interval duration.

III. Verification Methodology

The following points summarize the verification methodology and the whole project steps:

- High Level Model: The transmitter and receiver were modeled on a system level using MATLAB, and the models were verified by generating uniformly-distributed random bits, passing them to the transmitter model, receiving them through the receiver model, and ensuring that the received bits and the random input bits are identical, assuming ideal channel.
- Register Transfer Level: After the verification of the high-level model, it can then be used as a reference to the next step, so in this step each block was implemented using Verilog HDL, and the block's performance can be verified by comparison with the high-level model.
- Post-Synthesis Netlist: The RTL codes are verified to give correct results during behavioral simulation, so they're synthesized and the netlist is simulated to make sure the synthesized netlist gives the same results.
- FPGA Implementation: The design is implemented on the FPGA, and its output is monitored by the Integrated Logic Analyzer (ILA). The output of the design on the FPGA is verified by comparison with the behavioral simulations results.

IV. Results

The following tables are the utilization metrics on ZYNQ ZC702 Evaluation Board for the transmitter, receiver and synchronization. The used design frequency is 156 MHz. The utilization is much better than most of the previous related work [6, 7, 8].

TABLE I. TRANSMITTER UTILIZATION METRICS

Parameter	Used	%
LUT	11704	22
LUTRAM	1051	6
FF	10735	10
BRAM	8	6
BUFG	3	9

TABLE II. RECEIVER UTILIZATION METRICS

Parameter	Used	%
LUT	22568	42
LUTRAM	2821	16
FF	9679	9
BRAM	2.50	2
BUFG	3	9

TABLE III. SYNCHRONIZATION UTILIZATION METRICS

Parameter	Used	%
LUT	17130	32
LUTRAM	171	1
FF	19577	18
BRAM	1	1
DSP	42	19
BUFG	1	3

V. Acknowledgement

This work was partially funded by ONE Lab at Zewail City of Science and Technology and at Cairo University, NTRA, ITIDA, ASRT, and NSERC.

References

[1] Heiskala, Juha, and John Terry Ph D. OFDM wireless LANs: A theoretical and practical guide. Sams, 2001.

[2] C. Osama, Heba Magdy, H. Mahmoud, K. Khaled, M. Osama, Yara Hossam, S. Hosny, and H. Mostafa, "Wireless Communication Between ZYNQ 7000 FPGAs using USRP", IEEE International Conference on Modern Circuits and Systems Technology (MOCAST 2019), Thessaloniki, Greece, In Press.

[3] Forney, G. David. "The viterbi algorithm." Proceedings of the IEEE 61.3 (1973): 268-278

[4] C. Yu, M.-H. Yen, P.-A. Hsiung, and S.-J. Chen, "A low-power 64-point pipeline FFT/IFFT processor for OFDM applications," IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 40–40, 2011.

[5] James W. Cooley, Peter A. W. Lewis, and Peter W. Welch, "Historical notes on the fast Fourier transform," Proc. IEEE, vol. 55 (no. 10), p. 1675–1677 (1967).

[6] Haene, Simon, David Perels, and Andreas Burg. "A real-time 4-stream MIMO-OFDM transceiver: system design, FPGA implementation, and characterization." IEEE Journal on Selected Areas in Communications 26.6 (2008).

[7] Manavi, Farzad. Implementation of OFDM modem for the physical layer of IEEE 802.11 a standard based on Xilinx Virtex-II FPGA. Diss. Concordia University, 2004

[8] S. Hosny, E. Elnader, M. Gamal, A. Hussien, A. Hussein, and H. Mostafa, "A Software Defined Radio Transceiver Based on Dynamic Partial Reconfiguration", IEEE International Conference on Next Generation Circuits and Systems (NGCAS 2018), Malta, pp. 158-161, 2018.