Design and Verification of IEEE 802.11ah for IoT and M2M Applications

Rami Akeela
Electrical Engineering Department
Santa Clara University
Santa Clara, California 95053
Email: rakeela@scu.edu

Dr. Yacoub Elziq
JimzuTech
Saratoga, California
Email: elziq.ip@gmail.com

***Abstract*—In order to efficiently support the Machine-to-Machine (M2M) and Internet of Things (IoT) applications, a new amendment for the Wi-Fi standard known as IEEE 802.11ah is introduced. In 802.11ah (or Wi-Fi HaLow), several enhanced MAC features are added in order to provide scalability for a large number of stations, increase the range of operation, while at the same time reduce the energy consumption compared to the existing Wi-Fi standards. In this paper, a Verilog RTL implementation of the new standard is presented. It is suitable for IoT, M2M, V2V (Vehicle-to-Vehicle) applications, and smart grids that require long battery life and long range reliability. Software simulations using Xilinx Vivado are also provided to verify the design. The design is then synthesized using the same tool, and performance, power, and area are reported.**

Key words: Wi-Fi, 802.11ah, HaLow, long range, low power, IoT, M2M

I. Introduction

Advances in wireless technologies have altered consumers communication habits. Wireless technologies are an essential part of users' daily lives, and its effect will become even larger in the future. In a technical report, the World Wireless Research Forum (WWRF) has predicted that 7 trillion wireless devices for 7 billion people will be deployed by 2020 [1]. In a more recent report in 2014, Business Insider predicted that a total of 24 billion IoT devices will be connected to the internet by 2020 [2]. This will supplement a novel dimension to the world of information and communication technologies, where connectivity anytime and anywhere will now encompass everything, hence introducing the concept of the Internet of Things (IoT) [3]. The definition of "things" in the IoT world is very broad and comprises a variety of items. These items include personal objects we carry around such as smart phones, tablets and digital cameras. It also includes sensors and RFID. As a result, the IoT consists of heterogeneous sets of devices and heterogeneous communication strategies between these devices and dierent data servers. When these huge numbers of devices are connected to the Internet to form the IoT network, the first challenge is to adjust the basic connectivity and networking layers to handle the large numbers of end points [4]. There are wireless technologies that have been developed, such as ZigBee [5], for loT purposes due to different demanding requirements, one of those being high energy efficiency. Cellular technologies for example, are being scaled to serve this kind of traffic. Developing existing technologies to better meet the requirements of IoT is thus an important task. In many scenarios, the wireless node must operate without battery replacement for many years. Energy efficiency is therefore a very important constraint when designing an IoT network. The hardware, the link layer, the MAC layer, and all other higher layers should be jointly designed to minimize the total energy consumption [6]. Achieving an optimal joint design across all layers of the network protocol stack is quite challenging. Several existing proprietary solutions for Machine-to-Machine (M2M) and Wireless Sensor Networks (WSN) use cases are available in the market [7]. The IEEE 802.15 standard family has also addressed the low-power use case issue earlier. For example, Radio Frequency Identification (RFID) offers some of these functionalities. ZigBee and Bluetooth support other use cases. IEEE 802.11 is known to be a de-facto standard for wireless Local Area Networks (WLAN) and one of the most deployed solutions for enterprise architecture.

In this paper, we introduce a hardware implementation for 802.11ah. The design is based on Verilog, and it provides a good solution to any SoC to meet the needs of various IoT and M2M applications. In this discussion, we analyze the architecture of the standard, provide an RTL implementation, and then demonstrate its reliability through simulations and synthesis.

The paper is structured as follows: Section II discusses how this new standard is different from previous standards. Section III gives an overview on the system architecture of 802.11ah and discusses the implementation of the different blocks. In Section IV, the software simulations and silicon area are presented. Finally, the main conclusions are given and the future work is discussed in Section V.

II. IEEE 802.11ah Unique Features

Due to the diversity and complexity of the applications and environments in the IoT world, the next Wi-Fi standard must meet the following requirements in order to support M2M applications [8]:

- Up to 8,191 devices associated with an access point (AP)
- Carrier frequencies of sub-1 GHz
- Transmission range up to 1 km in outdoor areas
- Data rates of at least 100 Kbps

978-1-5090-4338-5/17/$31.00 ©2017 IEEE

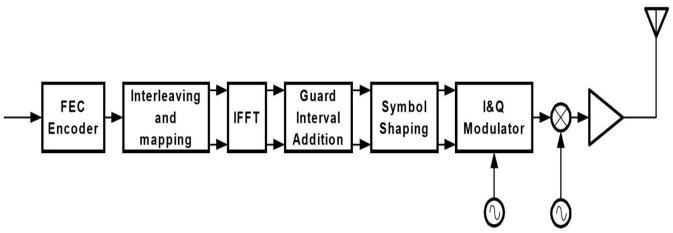

Fig. 1: 802.11ah Transmitter Block Diagram

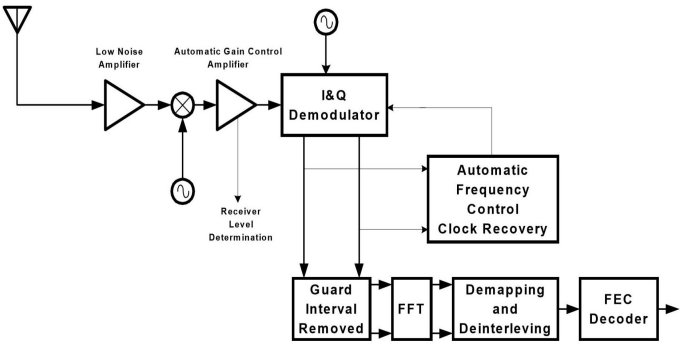

Fig. 2: 802.11ah Receiver Block Diagram

- Short and infrequent data transmissions
- Very low energy consumption
- Cost-effective solution

Current Wi-Fi standards, however, lack effective power-saving mechanisms and use unsuitable bands. IEEE 802.11ah is being designed as an amendment of the IEEE 802.11-2007 wireless networking standard. It has a new PHY and MAC design that operates in the sub-1 GHz license-exempt bands to provide extended range (1 Km radius) Wi-Fi networks, compared to the current 2.4 GHz and 5 GHz bands. It also benefits from lower energy consumption, supporting low cost battery-powered sensors operating without a power amplifier. The new standard also supports 4, 8, and 16 MHz bandwidths for higher-data rate applications.

Power consumption is reduced by adopting MAC protocols that yield lower power by using smaller frame formats, beaconless paging mode, sensor traffic priority. Power is further reduced by making the on-time for sensors even shorter, and that's due to the 100 Kbps minimum data rate. Another major enhancement is increasing the number of nodes connected to one access point to thousands. This is accomplished by optimizing the MAC by using efficient paging and scheduled transmissions [9]. In January 2016, the Wi-Fi Alliance announced an extension of 802.11ah to be called Wi-Fi HaLow [10].

III. System Architecture and Design

Figure 1 and Figure 2 illustrate the reference block diagram of the transmitter and receiver structures of the new standard [9]. The following blocks were designed and implemented:

Transmitter: $CRC \rightarrow Encoder \rightarrow Interleaver \rightarrow Scrambler$;

Receiver: $Descrambler \rightarrow Deinterleaver \rightarrow Decoder \rightarrow CRC$.

The FFT/IFFT blocks were not included in this implementation due to their complexity that usually requires a special synthesis tool. The Fast Fourier Transform (FFT) and the Inverse Fast Fourier Transform (IFFT) blocks, which are responsible for mapping digital data to/from analog signal, involve a computationally intensive set of vector and array matrix multiplications, and hence they were not included in this implementation.

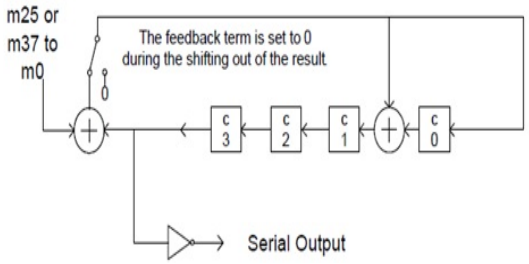

Fig. 3: 4-bit CRC Diagram

A. Cyclic Redundancy Check (CRC)

A cyclic redundancy check is an error-correcting code that uses cyclic codes (codes that shift bits of a codeword circularly) for error detection and correction (if necessary). A message or bit-stream sent with CRC enabled appends a check value to each specified block of data, which can be checked at the receiver.

802.11ah uses $G(D) = D^4 + D^1 + 1$ as the generating polynomial. Thus, $crc(D) = c_0D^4 + c_1D^2 + c_2D^1 + c_3D^0$.

For the implementation, a finite state machine is used to control the output such that the output of the CRC is aligned to the end of the message. The shift register that performs the CRC-4 calculation is identical in the transmit and receive paths. In the transmission path, zeros are shifted into the shift register at the end of the bitstream to obtain the signature. In the reception path, the signature is appended to the message and the resulting signature should be all zeroes.The CRC-4 is implemented as a simple bit shift register on a rising edge clock. Figure 3 depicts the CRC hardware implementation.

B. Convolutional Encoder

The Convolutional encoder consists of shift registers and XOR gates. The number of shift registers is based on the constraint length and the generator polynomial equations. For the standard $1/2$ encoder, we have the constraint length of 7. This means we have 6 shift registers and one input for the generator polynomials.

The figure below is a rate 1/3 (m/n) encoder with constraint length K of 3. The Generator polynomials are $G_1 = (1, 1, 1)$, $G_2 = (0, 1, 1)$, and $G_3 = (1, 0, 1)$. Therefore, output bits are calculated (modulo 2) as follows:

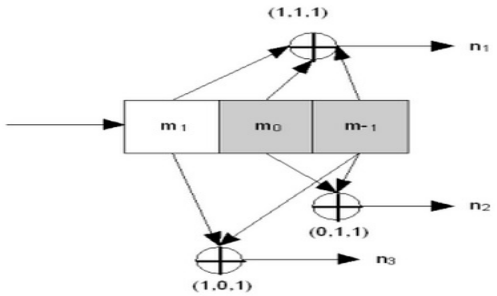

Fig. 4: Rate 1/3 non-recursive, non-systematic convolutional encoder with constraint length 3

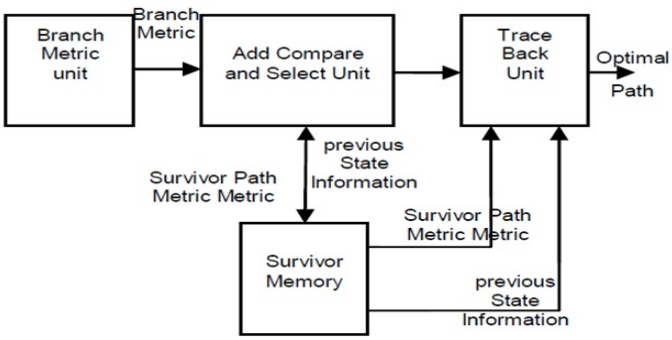

Fig. 5: Block Diagram of Viterbi Decoder

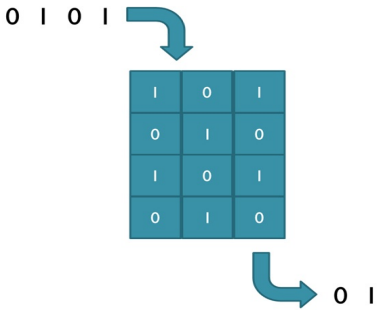

Fig. 6: Basic Interleaving Block Diagram

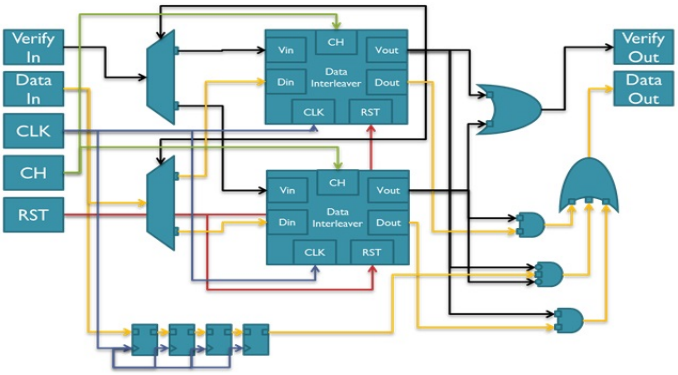

Fig. 7: Dual Interleaver RTL Diagram

$$n_1 = m_1 + m_0 + m_{-1}$$
$$n_2 = m_0 + m_{-1}$$
$$n_3 = m_1 + m_{-1}$$

C. Viterbi Decoder

The Viterbi decoder uses the Viterbi algorithm to estimate the most probable symbol sequence generated by the convolutional encoder on the transmission path. A Viterbi decoder generally consists of three major blocks, as depicted in Figure 5:

- Branch Metric Unit (BMU)
- Path Metric Unit (PMU)
- Trace-back Unit (TBU)

The implementation of the Viterbi decoder uses a statistical model for recollecting the original bitstream and is an intensive step in the receiver path, both in computation and in hardware. Thus, the code was generated using MathWorks MATLAB HDL code generator.

D. Interleaver/Deinterleaver

Interleaving is frequently used to improve the performance of forward error-correcting codes by creating a more uniform distribution of errors. It is widely used for burst error-correction, i.e. the errors occurred during the transmission are spread across multiple codewords. Therefore, errors within one codeword may be small enough to be corrected using some specific method (e.g., random error correction). There is a two-dimension m x N matrix for data storage, where m is the interleaving depth and N is the codeword length. The data is written column-by-column into the matrix and read out row-by-row. Figure 6 illustrates the process of the interleaving method. The main disadvantage of this method is that there will be an extra delay because deinterleaving can be started only after all the interleaved data is received. To solve the issue of a long delay, a different implementation is presented in Figure 7.

In Figure 7, one can notice that it is a dynamic interleaver, where the data fed can continuously be interleaved as we have two stages working in parallel. Vout from 1^{st} interleaver and 2^{nd} interleaver are being ORed and fed forward to produce a Verify out. For Data Out, either of Dout for 1^{st} or 2^{nd} is taken as only one interleaver is actively producing output. This Dataout is then fed forward to the next module. Every time an interleaver module is busy outputting data, the opposite interleaver can be used to receive data. This allows twice the throughput and continuous data processing compared to a single interleaver module.

In 802.11ah, the number of data sub-carriers is 48. The number of coded bits per symbol is the product of coded bits per carrier and number of carriers per symbol. For BPSK, the number of coded bits per carrier is 1; for QPSK, it is 2; and 4 and 6 for 16-QAM and 64-QAM respectively.

E. Scrambler/Descrambler

Scrambling is a type of "encryption" used in analog domain. It is accomplished by the addition of components to the original signal or the changing of some important component of the original signal so that to make it difficult for external

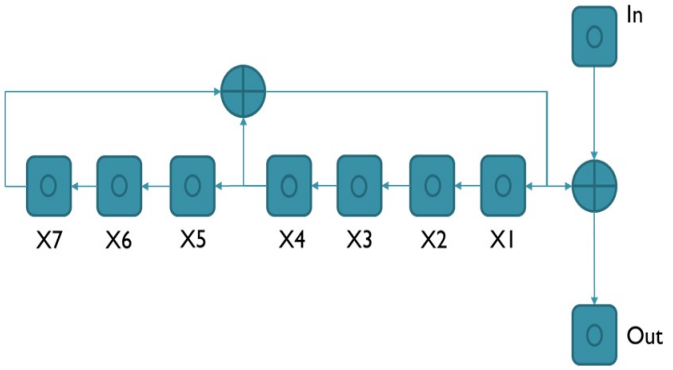

Fig. 8: Scrambler Block Diagram

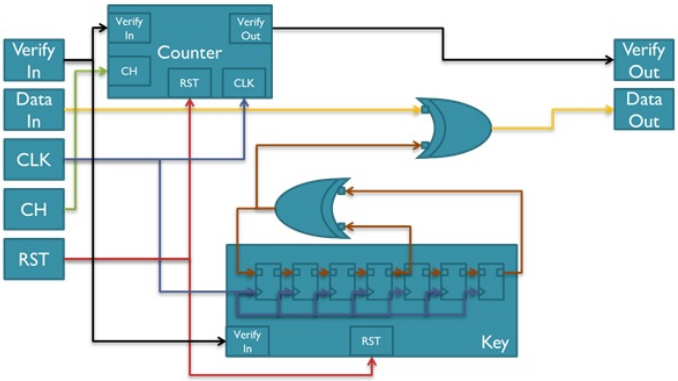

Fig. 9: Scrambler RTL Diagram

parties to extract the original signal. A scrambler is a device that transposes signals at the transmitter to make the message unintelligible at a receiver not equipped with an appropriately set descrambling device. It converts an input string into a seemingly random output string of the same length (e.g., by pseudo-randomly selecting bits to invert), thus avoiding long sequences of bits of the same value. The scrambler usually takes the form of shift registers with feedback connections, while the descrambler is a feedforward-connected shift register.

There are basically two types of scramblers. One is additive, also known as frame-synchronous scramblers, which uses modulo 2 operation (XOR). The other is called multiplicative, also known as self-synchronizing scramblers, which uses multiplication in the Z domain.

Every type of scrambler has its own generator polynomial, and generator polynomial used in 802.11ah is $S(x) = x^7 + x^4 + 1$. It will generate a repeated frame length of 127 bits. The XOR operation is adopted for the transmission. The Shift registers (length of 7) are used to achieve the operational goal. It first performs the XOR operation between the 4^{th} and 7^{th} shift registers, and the output of the operation will be passed to the 1^{st} shift register. In the meantime, the output will also be XOR'ed with the input bit. And the output of this operation represents the final output. The same process will repeat until the input bits are finished. Figure 8 shows the block diagram of the aforementioned additive scrambler, and Figure 9 illustrates the RTL implementation of it.

F. Controller

The controller module shown in Figure 10 receives the initial unmodified packet where it first detects 7 consecutive ones and the next following 8 bits depicts the size of data to be transmitted and received. We have a verify out signal which goes high as soon as the real data to be transmitted comes along or is identified.

G. Interfacing

Figure 11 shows all the basic necessary connections and leaves out all the miscellaneous ports. One should note that the diagram does not show the CRC modules, where they would be connected to the Encoder from the Transmitter side, and the Decoder from the Receiver side.

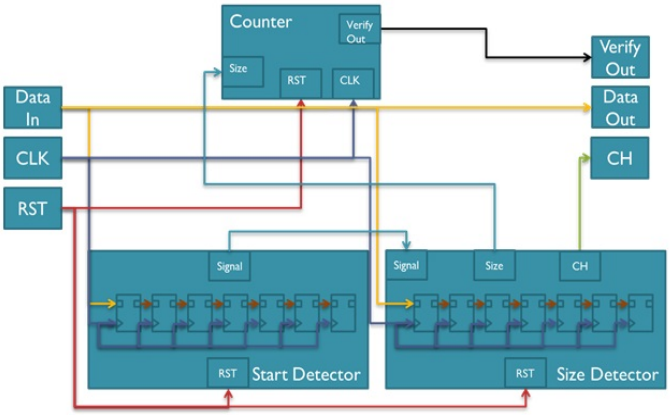

Fig. 10: Controller RTL Diagram

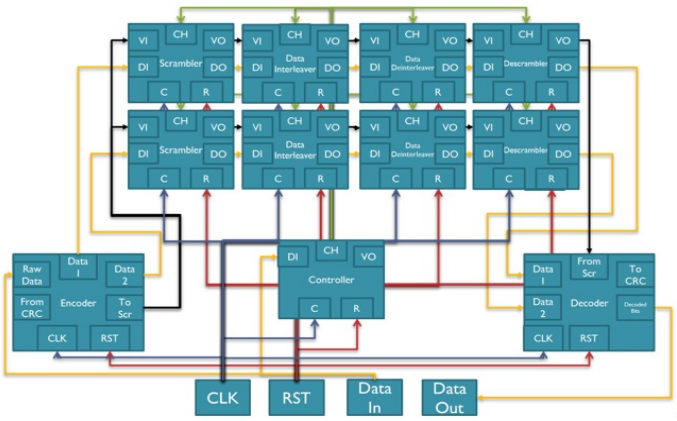

Fig. 11: Interfacing Diagram

IV. Verification and Performance Analysis

We started with a reference model for all the blocks, which was developed using the MathWorks MATLAB platform.

For example, in the scrambler module, we first initialize the scrambler to be at the state $s = 1111\ 111$ and in order to verify the 127-bit repeated pattern, we inject a stream of all

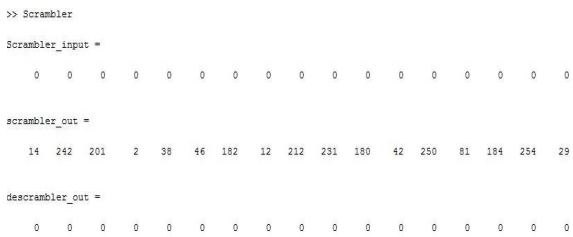

Fig. 12: Scrambler MATLAB Simulation

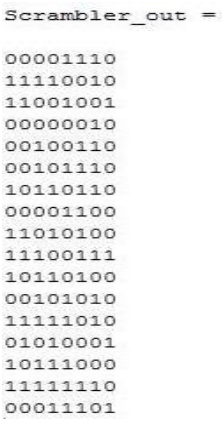

Fig. 13: Output of Scrambler in bits

0s to the system. As explained earlier, the new MSB (most significant bit) is the XOR'ed result of 4^{th} and 7^{th} bit. By shifting the s and replacing the new MSB, we can get the next state. In the meantime, the XOR operation of input and the new MSB is performed to generate the output of the scrambler. The descrambler is the same as the scrambler module. Inside the module, the input is passed to the scrambler first, and then goes through the descrambler. The input, scrambler output and the descrambler output are displayed and presented in Figure 12 and Figure 13. Figure 13 shows the binary representation of the scrambler output, and we can see that the pattern is exactly the same as we described before.

The algorithm was then captured, and system-level analysis was run using Xilinx Vivado 2015.4. The simulation was carried out to check if the design met the requirements of the 802.11ah standard. , and the waveforms are shown in Figure 14. It can be seen that $Data_{out}$ matches $Data_{in}$.

The design was then synthesized on the ZYNQ-7 ZC702 FPGA board. Figure 15 shows the schematic of the 802.11ah design. The hardware utilization summary is presented in Table I. As expected, the decoder module takes up the most silicon area because of the inefficient use of the automated Verilog code generation from MATLAB. The next biggest modules are the interleaver and deinterleaver. This is also expected because of the large use of latches. This can be further reduced by using memory modules.

The Total On-Chip Power is 0.718 W. Power analysis is reported in Table II.

TABLE I: Utilization Report

	Total	Used	%
LUTs	53200	6117	11.5
Flipflops	106400	6824	6.41

TABLE II: Dynamic Power Report

Signals	0.335 W	57%
Logic	0.25 W	42%
I/O	0.005 W	1%

V. Conclusion

In this work, the design of IEEE 802.11ah (HaLow) is presented. Results indicate that the design is suitable for FPGA and ASIC based applications. Contributions of this work will provide a good foundation for a wide variety of applications in IoT.

Future work may consider the implementation of 802.11ah on a board that provides programmablity, using the idea of partitioning the system into hardware and software. This method would be based on a set of criteria that would yield a better performance and lower power consumption.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] "WWRF (2009) Visions and research direction for the wireless world," Tech. Rep., 2009.

[2] J. Greenough. (2014) How the 'Internet of Things' will impact consumers, businesses, and governments in 2016 and beyond. [Online]. Available: http://www.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10

[3] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future internet research and experimentation: the fire initiative," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 89–92, 2007.

[4] M. Mafuta, M. Zennaro, A. Bagula, G. Ault, H. Gombachika, and T. Chadza, "Successful deployment of a wireless sensor network for precision agriculture in malawi," in *Networked Embedded Systems for Every Application (NESEA), 2012 IEEE 3rd International Conference on*. IEEE, 2012, pp. 1–7.

[5] B. Badihi Olyaei, "Modeling, performance evaluation and suitability study of zigbee technology for machine-to-machine communications applications," 2014.

[6] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 5, pp. 2349–2360, 2005.

[7] D. Niyato, L. Xiao, and P. Wang, "Machine-to-machine communications for home energy management system in smart grid," *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 53–59, 2011.

[8] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "IEEE 802.11AH: the WiFi approach for M2M communications," *Wireless Communications, IEEE*, vol. 21, no. 6, pp. 144–152, December 2014.

[9] "IEEE 802.11ah Draft 3.0, Draft Standard for Information technology Telecommunications and information exchange between systems local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Sub 1 GHz License Exempt Operation," *IEEE Std. 1516-2000*, March 2015.

[10] M. Hamblen. (2016) Wi-Fi for the Internet of Things gets a name: Wi-Fi HaLow. [Online]. Available: http://www.computerworld.com/article/3018510/mobile-wireless/wi-fi-for-the-internet-of-things-gets-a-name-wi-fi-halow.html

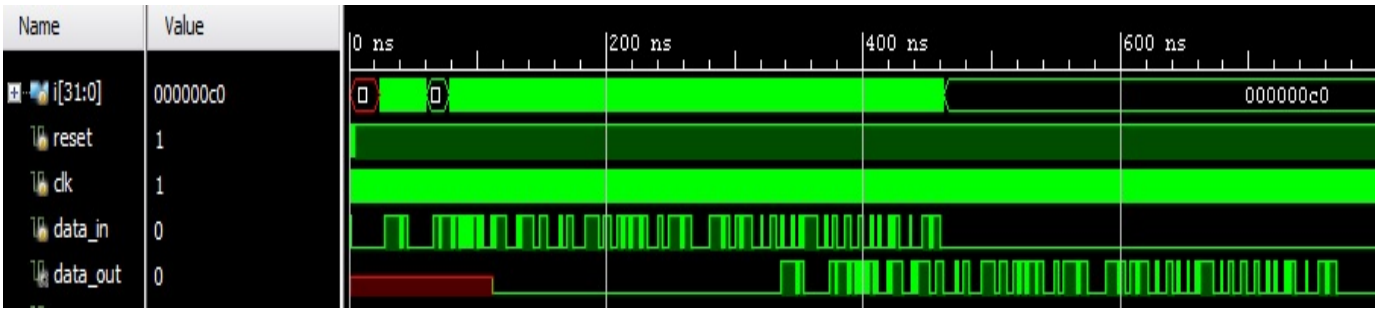

Fig. 14: Simulation Results

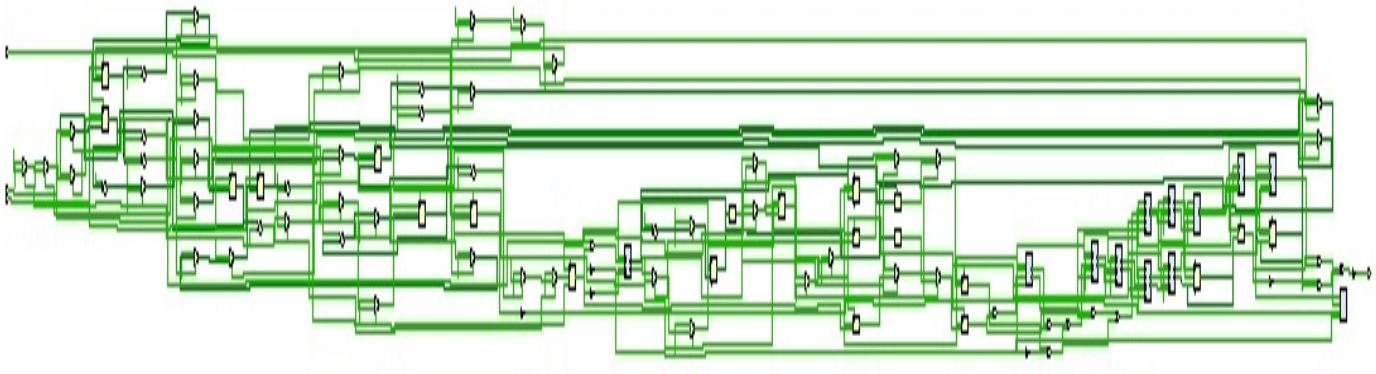

Fig. 15: Schematic of Proposed Design of 802.11ah