Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

6th International Conference
on Electrical Engineering
ICEENG 2008

VHDL modelling of the wireless audio transceiver through IEEE802.3 and IEEE802.11

By

K. W. Chew * W. L. Pang* Florence Choong* F.Y. Chua*

Abstract:

The wireless communication is highly deployed due to it convenience of mobility. The wireless local area network, WLAN is dominated by IEEE802.11 standard. Wired local area network (LAN) is the network backbone of the WLAN. Almost all the new desktops' motherboard are equipped with the Ethernet connecter and all new notebooks are equipped will IEEE802.11 WLAN. This evoked us to develop a wireless audio transceiver that compatible with IEEE802.3 and IEEE802.11 protocols. The VHDL modelling is done to model transceiver that convert the audio signal to IEEE802.3 packets to be transmitted through IEEE802.3 and IEEE802.11. The VHDL is selected to model the transceiver and VHDL is defined in IEEE as a tool of creation of electronics system because it supports the development, verification, synthesis and testing of hardware design, the communication of hardware design data and the maintenance, modification and procurement of hardware [1]. The transceiver consists of two individual modules, the transmitter and the receiver. The original audio signal is segmented to UDP packets that compatible with the IEEE802.3 protocol. The audio packets are broadcasted out through wireless access point (AP). After the audio packets received by the wireless clients through IEEE802.11 and IEEE802.3, the receiver will convert the audio packets back to audio signal. By broadcasting the data packets, all receiving speakers will be able to receive the data packets and convert the digital bits into audio wave. This gives the user the flexibility to place as many speakers as they want. The Cyclic Redundancy Check algorithm is not included in the transceiver because audio steaming is real time application, there is always not enough time to retransmit the audio packets that are corrupted. The proposed transceiver achieves 12kbps and sufficient to support audio streaming.

Keywords:

VHDL Modeling, IEEE802.3, IEEE802.11 and Transceiver

* Faculty of Engineering, Multimedia University, Malaysia

1. Introduction:

The wireless communication is highly deployed due to it convenience of mobility. The wireless local area network, WLAN is dominated by IEEE802.11 standard. All the new notebooks are equipped with the IEEE802.11 b WLAN. It becomes one of the main focuses of the WLAN research. Wired network is dominated by IEEE802.3 protocol. The IEEE802.3 has high data rate, 10Mbps and almost all the new desktops' motherboard are equipped with Ethernet terminal to access the IEEE802.3 local area network (LAN). This give us an idea to utilize this two protocols for wireless audio transmission.

The audio source is converted to IEEE802.3 packets and the packets are transmitted through IEEE802.3 protocol to an IEEE802.11 Access Point (AP). The audio signal is broadcasted out through AP. The audio signal is received by the wireless receiver that form by IEEE802.11 wireless adapter, Ethernet and VHDL receiver that convert the packets back to the audio signal.

The remaining of this paper is organized as follows: Literature review IEEE802.3 is carried out in section 2. The simulation tools that have been used are discussed in section 3. Section 4 illustrated the operation behaviour of the IEEE802.3 transmitter and the IEEE802.3 receiver is discussed in section 5. Simulation analyses are discussed in section 6. The maximum transmission rate is calculated in section 7. Finally, a conclusion is presented in Section 8.

2. IEEE 802.3 Ethernet

IEEE 802.3 Ethernet is developed to send / receive data at 10Mbps. It is able to access the internet network by using a contention method known as Carrier Send Multiple Access with Collision Detection (CSMA/CD) [2, 3, 4, 5]. The IEEE 802.3 frame format is located in the data link layer of the OSI (Open System International) and it performs the role of fragmenting the large data into smaller packets before passing the raw bits into the physical layer. The frame format of the IEEE 802.3 is stated in Figure 1 and the properties in Table 1.

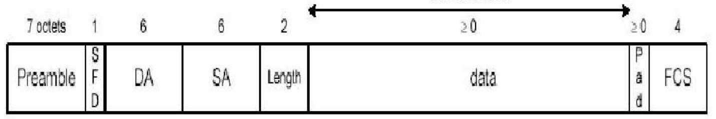
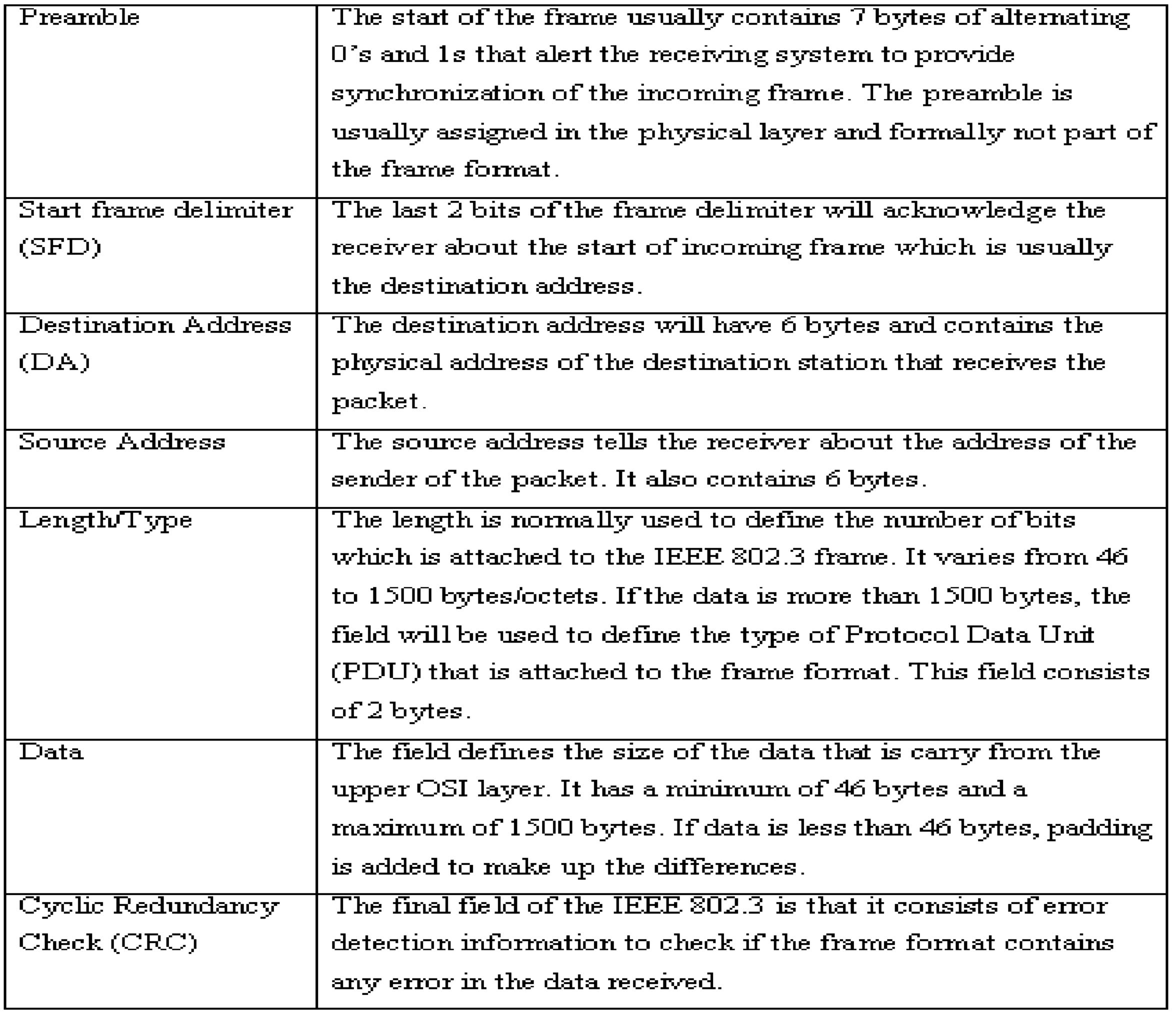


Figure 1: IEEE 802.3 frame format

Table 1: IEEE 802.3 Frame Format Properties



Preamble	The start of the frame usually contains 7 bytes of alternating 0's and 1s that alert the receiving system to provide synchronization of the incoming frame. The preamble is usually assigned in the physical layer and formally not part of the frame format.
Start frame delimiter (SFD)	The last 2 bits of the frame delimiter will acknowledge the receiver about the start of incoming frame which is usually the destination address.
Destination Address (DA)	The destination address will have 6 bytes and contains the physical address of the destination station that receives the packet.
Source Address	The source address tells the receiver about the address of the sender of the packet. It also contains 6 bytes.
Length/Type	The length is normally used to define the number of bits which is attached to the IEEE 802.3 frame. It varies from 46 to 1500 bytes/octets. If the data is more than 1500 bytes, the field will be used to define the type of Protocol Data Unit (PDU) that is attached to the frame format. This field consists of 2 bytes.
Data	The field defines the size of the data that is carry from the upper OSI layer. It has a minimum of 46 bytes and a maximum of 1500 bytes. If data is less than 46 bytes, padding is added to make up the differences.
Cyclic Redundancy Check (CRC)	The final fie ld of the IEEE 802.3 is that it consists of error detection information to check if the frame format contains any error in the data received.

Before IEEE 802.3 frame is able to be transmitted, it needs to first receive an acknowledgement from other stations to ensure that the network is not busy [2]. Once the transmitting station understands that the network is free, it will transmit the IEEE 802.3 frame. If a collision happens during the transmission, the sending station will send a jamming signal to acknowledge all the other stations about the collision of the frame. The minimum data that must be delivered in the internet network is 46 bytes to ensure that the CSMA/CD operates correctly. If there is no minimum data set, the receiving stations will be confused about the right time to discard a frame

3.Simulation Tool

VHDL or VHSIC Hardware Description Language is the programming language that is used to program FPGA chips or specified integrated circuits. VHSIC stands for Very High Speed Integrated Circuit and is capable of processing logic gates in a very fast speed. Many people prefer using VHDL language for designing digital system because it enables the system to be modeled and verified before the design is converted into a real hardware circuitry [6]. Furthermore, VHDL is a language that allows logic gate processes to occur in a concurrent system, which is essential for pipe lining application. In addition to that, VHDL algorithm is also able to define how the hardware is configured unlike the microcontroller where the user has to determine the software algorithm based on the hardware.

Most of the FPGA manufacturers will also provide the users with VHDL compiler, simulator and synthesizer to perform their designs. Among the reputable VHDL compiler are ModelSim and Quartus 2. The user is also responsible for writing the test bench that will define the time that the signals should change states. By using VHDL to obtain a simulation, the user can minimized the error before implementing the design in a FPGA chip.

4. IEEE802.3 Transmitter Algorithm

The incoming bits from the external hardware will be transmitted into the FPGA as shown in Figure 2. An algorithm to sample the incoming bits at 125 us is written. The sampling time 125 us is chosen because the incoming bits, which are transmitted by the external hardware, is set to 125us. Thus, if the sampling time of the algorithm is not set to 125us, there will be no synchronization between the external hardware and the FPGA chip. After sampling each bit, a serial to parallel algorithm is written so that the bits that enter serially will be processed in parallel inside the FPGA chip.

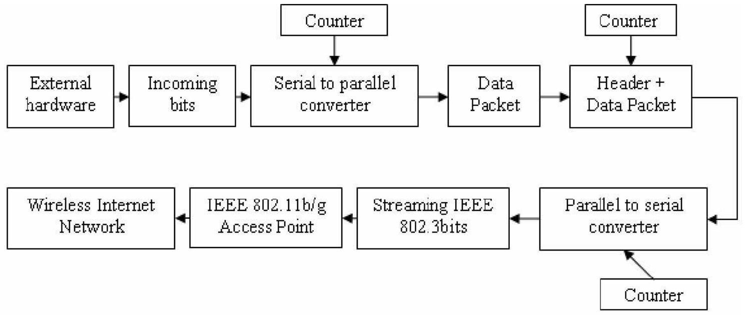

Counter
Counter
External hardware
Incoming bits
Serial to parallel converter
Data Packet
Header + Data Packet
Wireless Internet Network
IEEE 802.11b/g Access Point
Streaming IEEE 802.3bits
Parallel to serial converter
Counter

Figure 2: Block diagram of the VHDL transmitter

The number of bits that will be processed in parallel inside the FPGA chip will be 336 bits or 42 bytes. The minimum bytes that an IEEE 802.3 frame format can transmit are 42 bytes. If less than 42 bytes are sampled, the IEEE 802.11 b/g Access Point will fail to recognize the data packet as a valid IEEE 802.3 frame format and the data packet will be discarded. The padding algorithm is not written because it will cause more delay to the transmitting data. If the delay is too high, the FPGA chip may not keep up with the sampling data and discard all the incoming bits from the external hardware. Since introducing the padding algorithm will cause more problems, this algorithm is not implemented.

After sampling 42 bytes, an algorithm to attach an IEEE 802.3 header to the data is written. (Figure 2). The IEEE 802.3 header consists of a preamble, the destination address, the source address and length of data sent. The preamble is normally done in the physical layer instead of the MAC layer but it is included in the algorithm to demonstrate how digital bits are encoded and decoded by the FPGA chip. The destination address of the algorithm is selected to be FFFFFF because the data packets will be broadcasted to all network stations. By broadcasting the data packets, all receiving speakers will be able to receive the data packets and convert the digital bits into audio wave. This gives the user the flexibility to place as many speakers as they want inside the room. If the destination address is fixed to a predefined address, only a single receiving speaker is able to play the audio wave. The source address will inform other internet network stations the origin of the data packet. The length field of the IEEE802.3 frame format is set to “2AH” to represent that the data sent is 42 bytes. The header packet value of the IEEE 802.3 frame format is clearly defined at the start of the VHDL algorithm throughout the transmission since the header parameters is constant during the audio streaming process.

The Cyclic Redundancy Check algorithm is not included in the IEEE 802.3 because by sending streaming audio, there is always not enough time to retransmit the audio that is corrupted. If the receive audio data is corrupted, we have no choice but to discard the packet and hope that the incoming data packets will not face the same problem as well. If the corrupted audio file is permitted to be retransmitted, it will result in internet network congestions. This is the setback of transmitting audio in real time.

Once the header and the data combined to become an IEEE 802.3 frame format, the data needs to be converted into serial bits again. The purpose of converting the parallel data packet back into serial bits is to reduce the number pins used in the FPGA chip and the IEEE 802.3 network cable limitations. Thus, a parallel to serial algorithm is introduced to transform the data back into single bits again. The algorithm flow chart can be observed in Figure 3.

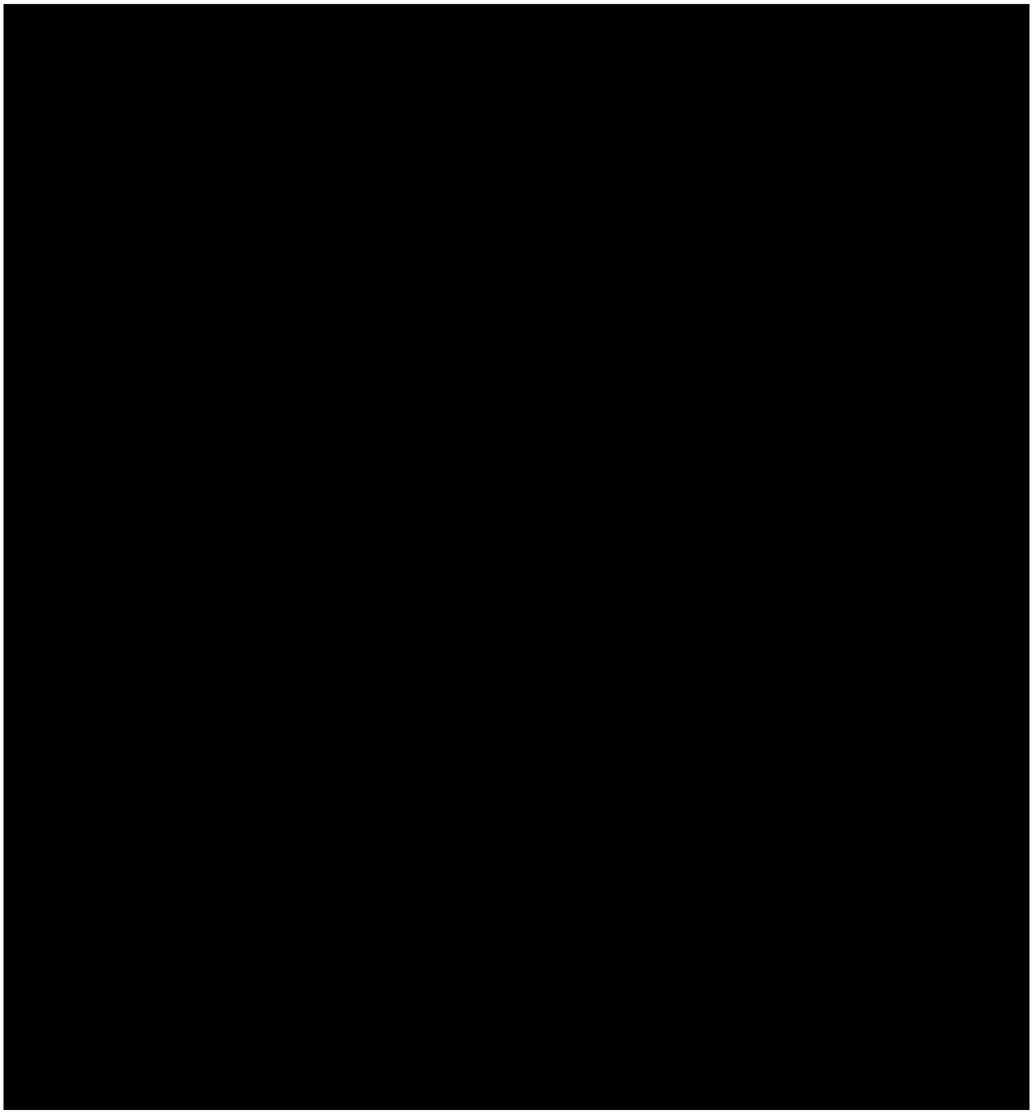


Figure 3: VHDL transmitting algorithm flow chart

5. IEEE 802.3 Receiver Algorithm

Figure 4 shows the block diagram of the receiver side. The IEEE 802.11b/g Client is always alerted for data packets in the internet network system. Once a data packet is detected, it will check for the headers of the IEEE 802.11b/g frame format and discard the corrupted frame format. If the frame format is valid, it will then removed the header and send the data to the IEEE 802.3 network to be processed. An assumption is made to assume that the header is removed at the IEEE 802.3 protocol instead of the IEEE 802.11b/g protocol for theoretical understanding.

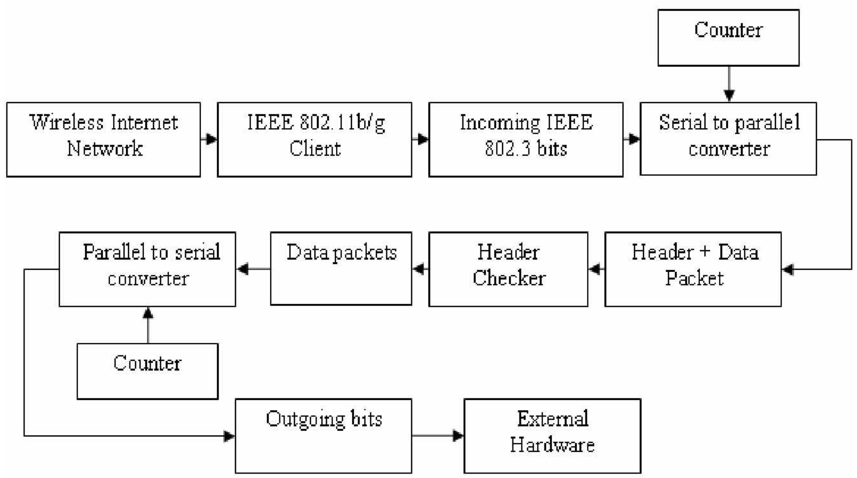


Figure 4: Block diagram of the VHDL receiver

Figure 5: VHDL receiving algorithm flow chart

If the IEEE 802.11 b/g Client detects a network packet in the wireless medium, it will capture the incoming bits to be processed at the FPGA. (Figure 4) The sampling time for each of the incoming bit is 82.03125 us. The sampling time at the receiver is much faster because the IEEE 802.3 frame format has a longer data bits to sample. The binary bits will be converted into a parallel packet by applying the serial to parallel algorithm. Once the binary bits are converted into a digital packet, it will be compared with the predefine header inside the FPGA. If the header of the digital packets is similar to the one at the transmitting side, the data packet will be processed in the following module. If the digital packets are not the same as the header of the transmitting, the digital packets will be discarded. The IEEE 802.11 b/g Client will continue to wait for other incoming binary bits.

After checking the validity of the data, the data packet will be converted into serial bits again. The rationale of such actions is to reduce the amount of pins for the FPGA and the limitation of the speaker wiring connections. Therefore, in order to convert the data back to serial form, there is no choice but to use a parallel to serial converter. The time for the parallel to serial converted is set to 125 us to synchronize the sampling time of the FPGA and the external speaker. The receiving algorithm flow chart is shown in Figure 5.

6 <u>IEEE 802.3 Frame Format Simulation Analysis</u>

The data packet will combine the header to form a complete IEEE 802.3 frame format. (Figure 6) The combination of the data packet from the *serial packet* (Figure 6 Green Box) and the header is performed in the *output* module (Figure 6 Red Box). In order to combine the data packet and the header, an *enable* is sent to acknowledge the *output* the right moment to combine the 2 data. The *enable* signal is shown under the *clear* and is represented by the green box. Figure 7 shows the simulation of a complete IEEE 802.3 frame format.

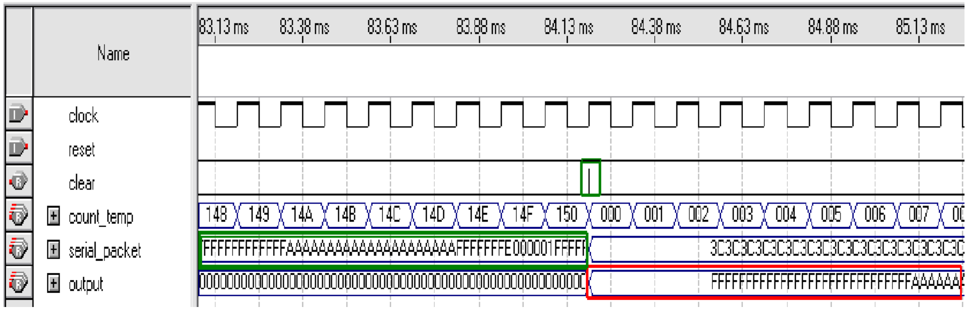


Figure 6: Combination of the header and the data packets

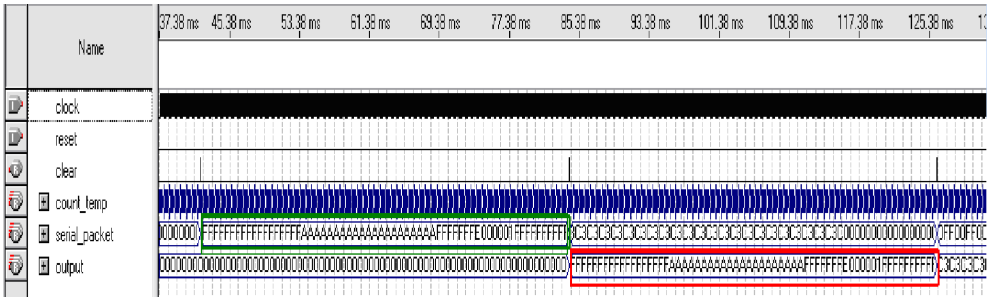


Figure 7: Complete IEEE 802.3 frame format simulation

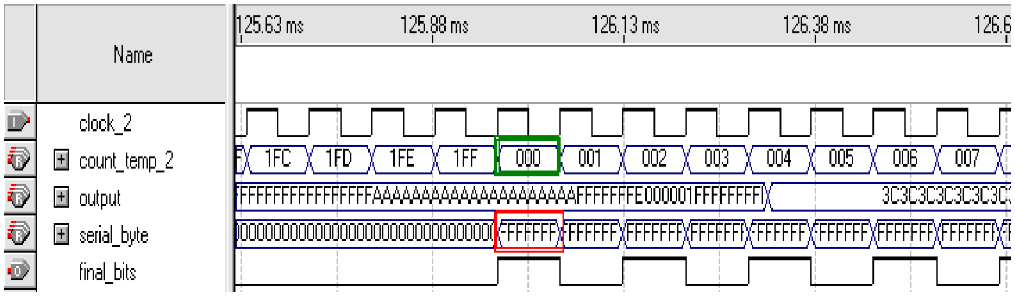


Figure 8: Conversion of IEEE 802.3 frames into serial bits

Once we obtained the complete IEEE 802.3 frame format, the IEEE 802.3 frame is converted into serial bits again. The data needed to be converted into a serial format because the IEEE 802.3 network cable can only support the transmitted data with a maximum of 2 pin. Thus, a parallel to serial command is executed by the FPGA in *serial byte* (Figure 8) to transform the data packet back to serial bit.

From Figure 8, we observed the *count temp 2* (Figure 8 Green Box) is constructed using the normal counter algorithm. Thus, when the counter resets its count again, the data kept inside the *serial byte* (Figure 8 Red Box) is transmitted out to the *final bits*. The clock time to transmit the data out is 82.03125 us. The sampling time is faster because we want to ensure that the data is transmitted out in time before the *serial byte* receive another IEEE 802.3 frame from the output. The data from *finals bits* is transmitted into

IEEE 802.11 b/g Access Point so that it will be sent wirelessly on the internet. Figure 9 demonstrates the complete simulation of how the data packets are converted into serial bits.

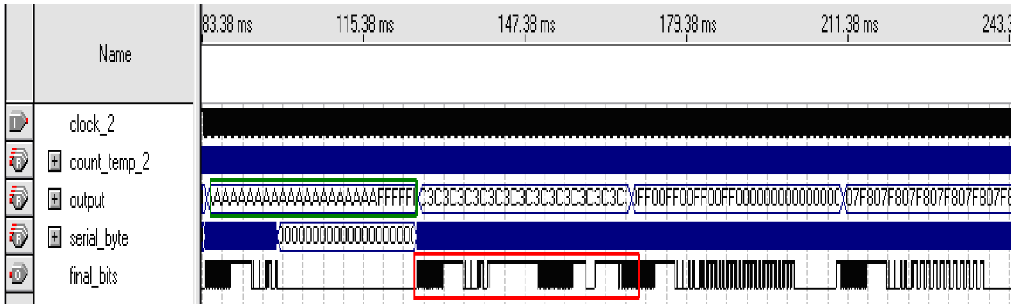


Figure 9: Complete IEEE 802.3 frames been converted into serial bits

7. IEEE 802.3 Maximum Data Rate

It is important to know the data rate of the IEEE 802.3 frame format to ensure that maximum data is transmitted out. In order to perform the calculation, we need to know the sampling time and the data size that is transmitted. The sampling time for 1 bit from the MC145484 chip is 125 us. Thus, the FPGA clock needs to be set to 125 us to sample the bit from the IC. The rest of the parameters that are used to calculate the data rate can be found in the calculations below.

Time to sample 1 bit = 125 us
Number of bits need to be sampled = 336 bits
Time to complete 1 data sampling = 125 us X 336 bits
= 0.042 bps
Number of bits for 1 complete IEEE 802.3 frame format
= 512 bits
Time to sample 1 bit for IEEE 802.3 frame format
= 0.042 bps / 512 bits = 82.03125 us.
Data rate for 1 s = (1s / 0.042 bps) X 512 bits
= 23.8 X 512 bits
= 12190 bps
= 12.19 kbps

The most important parameter that determines the data rate transmitted is the number of bytes that is sampled. The data rate that is transmitted is inversely proportional to the number of bytes that is sampled. This statement can be observed in Figure 10. The formula to calculate the data rate is given as:

Formula to calculate data rate

= ((n X 8) + 176) / (125 us X n X 8)
Number of bytes that is sampled = n
IEEE 802.3 header = 176 bits

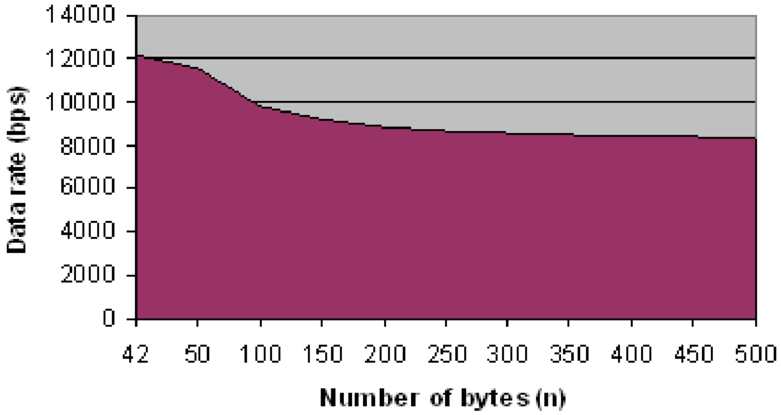


Figure 10: Data rate vs. number of bytes sample

8. Conclusions:

VHDL modelling of the wireless audio transceiver that occupied on top of the IEEE802.3 and IEEE802.11 protocols has been done. The wireless audio transmitter segmented the audio signal to UDP packets format and compatible with the IEEE802.3 protocol. The audio packets are broadcasted out through IEEE802.11 wireless AP. All the IEEE802.11 wireless adapter will received the audio packets. The wireless header of the audio packets is removed and the audio packets remained in IEEE802.3 frame format. The wireless audio receiver removed the IEEE802.3 header and converted the packets back to the audio signal. The wireless audio transceiver modelled has 12kbps data rate and the data rate is sufficient for audio streaming through IEEE802.3 and IEEE802.11 network protocols.

References:

[1] IEEE Standard, "IEEE Standard VHDL Language Reference Manual", IEEE Standard 1076-1987.

[2] Du, D.H.C.; Chang, S.P.; Subbaro, G.; "Multiple packet multiple channel CSMA/CD protocols for local area networks", INFOCOM '89. Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 1, April 1989 Page(s):163 - 172

[3] Wu, J.-L.C.; Wu, J.; Lee, T.-C.; "Performance analysis of interconnected CSMA/CD networks with finite population", Seventh Annual Joint Conference of the IEEE Computer and Communcations Societies, IEEE, March 1988 Page(s):1005 - 1011

[4] Whetten, B.; Steinberg, S.; Ferrari, D.; "The packet starvation effect in CSMA/CD LANs and a solution", Proceedings of Local Computer Networks, Oct. 1994 Page(s):206 – 217

[5] Beuerman, S.L.; Coyle, E.J.; "The delay characteristics of CSMA/CD networks", IEEE Transactions on Communications, Volume 36, Issue 5, May 1988 Page(s):553 - 563

[6] IEEE Standard, "IEEE Standard VHDL Language Reference Manual", IEEE Standard 1076 - 1987