International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Wireless Communication Between Multi-FPGAs

Suhas D B[1], Sumanth Heblikar[2], Savan S. Shivanagutti[3], Pooja A P[4], Vasundhara Patel[5]

[1] *Electronics and Communication & BMS College Of Engineering*
[2] *Electronics and Communication & BMS College Of Engineering*
[3] *Electronics and Communication & BMS College Of Engineering*
[4] *Electronics and Communication & BMS College Of Engineering*
[5] *Electronics and Communication & BMS College Of Engineering*

---***---

Abstract – PHY Layer implementation of the IEEE 802.11 a/g/n Wi-Fi standard using Software-Defined Radio (SDR) architecture. It addresses the challenge of implementing Wi-Fi on SDR platforms, which has been difficult due to the strict timing requirements of Wi-Fi protocols such as the Short Inter-Frame Space (SIFS) and the need for low-latency communication between the radio front-end and the processing unit. This architecture enables the implementation of the PHY (Physical layer) and low-level MAC (Media Access Control) functionalities in the FPGA, ensuring low latency and precise timing control.

***Key Words*:** PLCP, OFDM, SDR, BPSK, QPSK, QAM, PSDU, NCS

1.INTRODUCTION

The Wireless communication has become an integral part of our daily lives, with Wi-Fi being the backbone of our connected world. As the demand for faster data rates, improved coverage, and reliable connectivity continues to rise, there is a need for innovative approaches to enhance Wi-Fi systems. Software Defined Radio (SDR) has emerged as a transformative technology that enables the dynamic reconfiguration and adaptation of wireless communication systems.SDR allows for the separation of the physical hardware components from the software-defined functionalities, providing greater flexibility and programmability. This enables the implementation of various wireless protocols, including Wi-Fi, on a common hardware platform. By leveraging SDR, Wi-Fi systems can benefit from improved performance, increased spectrum efficiency, and easier deployment of new features and standards.

2. LITERATURE SURVEY

The Distributed or network control systems (NCSs) have become popular research topic. Along with sensors, actuators, and controllers, the communication infrastructure of NCSs plays a crucial role in their performance. One of the main challenges of NCSs is the time constraint for transmitting sensor data or state variables between different parts of the system. Failure to meet this constraint can lead to degraded performance or even system instability. Another challenge is data loss or dropout in the network [1].

The new era of computing is not CPU-centric but enriched with all the heterogeneous computing resources including the reconfigurable fabric. In multi-FPGA architecture, either deployed within a data center or as a standalone model, inter-FPGA communication is crucial. Network-on-chip exhibits a promising performance for the integration of one FPGA [2].

[4] This paper introduces a new method for wireless communication between an FPGA device and a microcontroller using inexpensive Bluetooth modules. The FPGA's parallel processing ability and precise timing features are utilized to offload computationally intensive tasks from the microcontroller. Bluetooth operating at 2.4 GHz is used for communication, and the UART protocol facilitates data exchange between the microcontroller, FPGA, and Bluetooth module.

[5] This paper introduces a novel approach for wireless communication between a microcontroller and FPGA using the MQTT protocol over Wi-Fi. By leveraging Wi-Fi, the system benefits from a larger network coverage and faster transmission speeds compared to traditional RF or Bluetooth modules. The implementation utilizes the MSP432 microcontroller and BASYS3 FPGA board, with the FPGA acting as the central processing unit to optimize system performance and handle computationally intensive tasks.

3. PHY LAYER DESIGN ARCHITECTURE

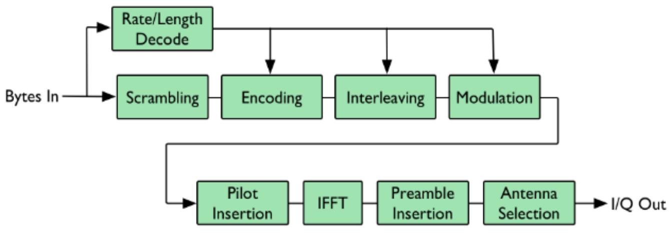


Fig 1, PHY_TX [6]

The PHY_TX of OFDM is a fundamental component in wireless communication system. OFDM is a modulation technique widely used in various wireless standards due to its ability to combat frequency-selective fading and achieve high data rates.

The PHY_TX of OFDM involves several key operations. Firstly, the digital data to be transmitted is encoded using error correction codes to enhance the reliability of the transmitted signal. This encoding process adds redundancy to the data, allowing for the detection and correction of errors at the receiver.Next, the encoded data is mapped to a set of subcarriers. These subcarriers are closely spaced and orthogonal to each other, which enables simultaneous transmission of multiple data streams. Each subcarrier carries a portion of the encoded data, thereby allowing for parallel transmission and increased spectral efficiency. After subcarrier mapping, an Inverse Fast Fourier Transform (IFFT) is performed on the subcarrier symbols to convert them from the frequency domain to the time domain. This transformation creates a time-domain signal, which consists of a series of complex-valued samples representing the modulated symbols.

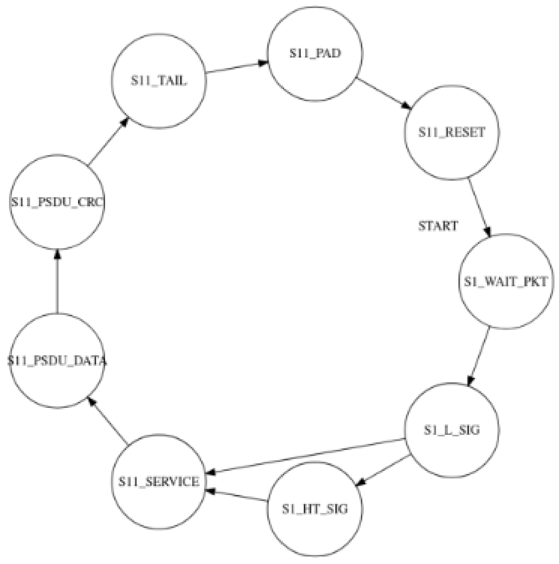


Fig -2: Data Collection State

For Data Collection States (state11):

- S1_WAIT_PKT : This state represents the waiting state where the module is waiting for a new packet to be transmitted.
- S1_L_SIG : This state indicates that the module is processing the legacy signal portion of the packet.
- S1_HT_SIG : In this state, the module is processing the high-throughput (HT) signal portion of the packet.
- S1_DATA : This state represents the processing of the data portion of the packet.

For Sub-States within S1_DATA (state11):

- S11_SERVICE : This sub-state indicates that the module is processing the service field of the data packet.
- S11_PSDU_DATA : In this sub-state, the module is processing the PSDU (Physical Layer Service Data Unit) data of the packet.
- S11_PSDU_CRC : This sub-state represents the processing of the PSDU CRC (Cyclic Redundancy Check) field.
- S11_TAIL : The module is processing the tail bits of the packet in this sub-state.
- S11_PAD : This sub-state indicates the processing of padding bits in the packet.
- S11_RESET : In this sub-state, the module resets its internal state and prepares for the next packet transmission.

These states and sub-states are used to control the flow of data collection within the OpenWiFi_tx module. The module transitions between these states based on the completion of specific tasks and signals in the packet processing pipeline.

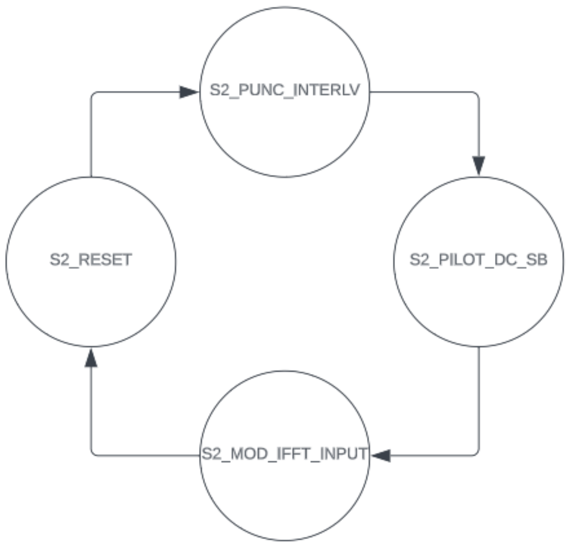


Fig -3: IQ Sample Generation State

For IQ Sample Generation States (state2):

- S2_PUNC_INTERLV : This state represents the puncturing and interleaving stage in the IQ sample generation process. Puncturing refers to the removal of certain bits from the data stream, and interleaving refers to the reordering of the bits to introduce resilience against burst errors.
- S2_PILOT_DC_SB : In this state, the module is generating the pilot, DC (Direct Current), and sideband symbols. Pilot symbols are known reference symbols used for channel estimation, DC symbols are used for DC offset compensation, and sideband symbols carry additional information.
- S2_MOD_IFFT_INPUT : This state represents the modulation and preparation of the input for the Inverse Fast Fourier Transform (IFFT). The data is modulated using a specific modulation scheme, and the resulting symbols are prepared for the subsequent IFFT operation.
- S2_RESET : This state is a reset state where the module resets its internal state and prepares for the next round of IQ sample generation.

These states are used to control the flow of IQ sample generation within the OpenWiFi_tx module. The module transitions between these states based on the completion of specific tasks and signals in the IQ sample generation pipeline. Each state represents a specific stage in the process of generating IQ samples for transmission.

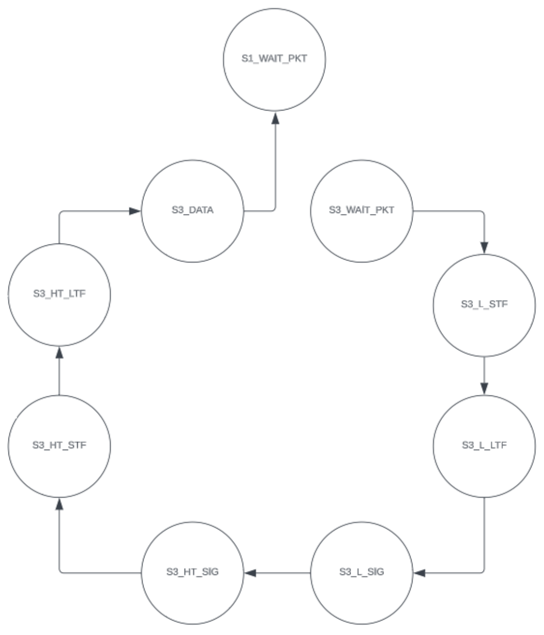


Fig -4: IQ Forwarding State

For IQ Sample Forwarding States (state3):

- S3_WAIT_PKT: This is the initial state where the module waits for a packet to be available for processing.
- S3_L_STF: In this state, the module is forwarding the Legacy Short Training Field (L-STF) IQ samples. The L-STF is a known reference signal used for channel estimation and synchronization in legacy Wi-Fi systems.
- S3_L_LTF: This state represents the forwarding of the Legacy Long Training Field (L-LTF) IQ samples. The L-LTF is another known reference signal used for channel estimation and synchronization in legacy Wi-Fi systems.
- S3_L_SIG: This state corresponds to the forwarding of the Legacy Signal (L-SIG) IQ samples. The L-SIG contains important information such as the modulation scheme, coding rate, and length of the transmitted packet.
- S3_HT_SIG: In this state, the module is forwarding the High-Throughput Signal (HT-SIG) IQ samples. The HT-SIG carries similar information as the L-SIG but is used for high-throughput transmissions in 802.11n and later Wi-Fi standards.
- S3_HT_STF: This state represents the forwarding of the High-Throughput Short Training Field (HT-STF) IQ samples. The HT-STF is a known reference signal used for channel estimation and synchronization in high-throughput Wi-Fi systems.
- S3_HT_LTF: This state corresponds to the forwarding of the High-Throughput Long Training Field (HT-LTF) IQ samples. The HT-LTF is another known reference signal used for channel estimation and synchronization in high-throughput Wi-Fi systems.
- S3_DATA: This state represents the forwarding of the actual data IQ samples..

These states are used to control the flow of IQ sample forwarding within the OpenWiFi_tx module. The module transitions between these states based on the completion of forwarding specific types of IQ samples and the availability of the next set of IQ samples for forwarding. Each state corresponds to a specific stage in the process of forwarding IQ samples for transmission.

4. PHY TX TESTBENCH

The PHY TX testbench was implemented after complete understanding of OFDM. The testbench is responsible for simulating the behavior of the module and capturing the output results. The module is designed to handle the physical transmission in a wireless communication system. It performs operations such as scrambling, encoding, modulation, and packet generation. To verify the correctness of the module, the testbench initializes signals, reads input data from a memory array, controls the start of the transmission, captures the IQ results, and monitors other relevant output signals.

The algorithmic representation of the code is given below to understand its functionality in detail.

Step-1: Set up the simulation environment and initialize variables.
Step-2: Toggle the clock signal at a frequency of 200MHz.
Step-3: On the positive edge of the clock:
  - If the system is in the reset state:
    - Reset the relevant data and count values.
  - Otherwise:
    - Retrieve data from memory based on the specified address.
    - Increment the count value.
    - If the IQ result is valid:
      - Write the IQ result, input data, PHY transmission status, and count to a file.
Step-4. At the start of the simulation, load the data from a specified memory file.
Step-5. Set the clock and reset signals and initiate the PHY transmission.
Step-6. Initialize the pilot and data scrambling states.
Step-7. Run the dot11_tx module with the provided inputs and capture the outputs.
Step-8. After the simulation completes, finish writing to the file.
Step-9. Exit the simulation.

The following output signals were obtained after simulating the PHY TX with the testbench.

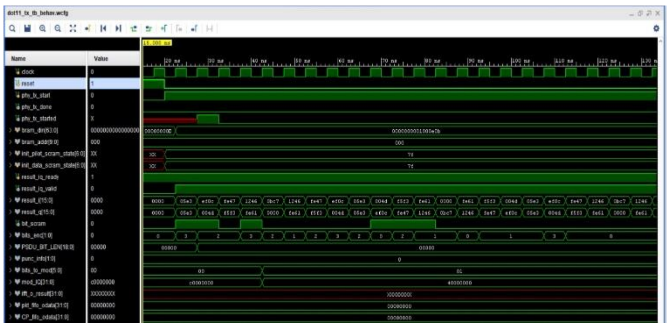


Fig -5: Simulation result

5. FUTURE IMPLIMENTATION

Software-Defined Radio (SDR) technology has revolutionized the implementation of Wi-Fi Direct, enabling direct peer-to-peer communication between devices. With SDR, the hardware components traditionally used in wireless communication, such as modulators and demodulators, are replaced by software running on a programmable platform. SDR-based Wi-Fi Direct implementations offer unparalleled flexibility and adaptability. The wireless transceiver on an SDR platform can be programmed to support various frequency bands and modulation schemes used in Wi-Fi Direct. This means that devices can seamlessly operate on different channels and adapt to changing channel conditions. One of the key advantages of SDR-based Wi-Fi Direct is the ability to implement the Wi-Fi Direct protocol stack in software. This allows for customization and modification of the protocol stack to meet specific requirements or incorporate additional features. The software-based approach also simplifies testing and prototyping, as modifications can be made without changing the underlying hardware.

Establishing a Wi-Fi Direct protocol stack involves several steps to ensure successful peer-to-peer communication between devices. Here are the key steps involved:

- Device Discovery
- Group Owner (GO) Negotiation
- Security Establishment
- Service Discovery

6. GROUP OWNER ALOGRITHM

In Wi-Fi Direct, the Group Owner (GO) is a crucial component responsible for coordinating communication within a Wi-Fi Direct group. There are different types of GOs based on how they are selected or assigned within the group:

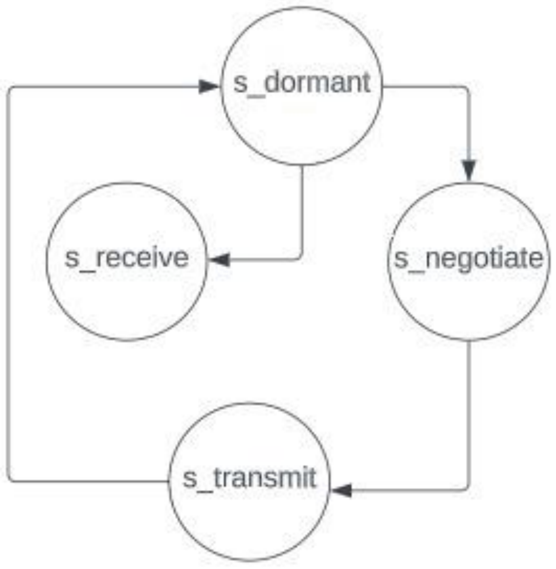
1. Autonomous Group Owner: In this type, a specific device is preconfigured or pre-determined to become the GO without any negotiation or selection process. The device assumes the role of the GO based on predefined criteria like capabilities, priority, or administrative settings. This type is used when a particular device needs to take control as the GO regardless of other devices' capabilities or preferences.

2. Standard Group Owner: In this type, devices in the Wi-Fi Direct group negotiate and select the GO based on specific criteria. The negotiation involves exchanging capabilities and preferences between devices. Factors considered during negotiation can include device capabilities, power-saving features, available resources, or user preferences. The devices collectively determine the most suitable device to become the GO. This type allows for dynamic and optimized GO selection based on the devices' needs and capabilities.

3. Persistent Group Owner: In a Persistent Group, devices have previously established a Wi-Fi Direct connection and saved the group information. When reconnecting, the device that initiated the persistent group becomes the Persistent Group Owner. This ensures consistency and stability in subsequent connections, as the same device retains the GO role. This type is useful when devices frequently need to reconnect to the same Wi-Fi Direct group, maintaining the same GO for ease of communication.

The selection of the Group Owner depends on negotiation processes, predefined criteria, or the presence of a persistent group. Once a device becomes the GO, it assumes additional responsibilities such as managing network parameters, configuring security settings, handling connection requests from other devices, and facilitating data transmission among devices within the Wi-Fi Direct group. By following these steps, a Wi-Fi Direct protocol stack can be established, allowing devices to form direct peer-to-peer connections and facilitate data transfer and communication in a Wi-Fi Direct environment.
The code represents a module called `group owner` in a Wi-Fi Direct network. It is responsible for managing the Group Owner (GO) selection and group formation process. The module handles various scenarios for GO selection based on factors such as device capabilities, power-saving features, resources, and user preferences. By following the defined states and transitions, the module enables devices in a Wi-Fi Direct group to establish and manage direct peer-to-peer connections efficiently.

Fig -6: GO state diagram



The following steps represent the algorithm for GO:

Step-1: Initialize the module parameters, including the state values and counters.
Step-2: On the positive edge of the clock:

```
- If the system is in the reset state:
  - Reset the output signals and set the node state to dormant.
- Otherwise, if a group formation is initiated:

  - Set the group formation started flag.
  - Based on the current node state:
    - In the dormant state:
        - Decrement the counter value.
        - If the counter reaches zero:
          - Check if there is a GO selected message.
          - If no GO is selected, transition to the negotiate state.
          - If a GO is selected, send a group join consent to the GO.
      - In the negotiate state:
        - If there is no GO selected message, declare this node as the GO.
        - Transition to the transmit state.
        - If there is a GO selected message, transition back to the dormant state.
      - In the transmit state:
        - Set the group info to the current node ID.
        - Send a group join probe to the GO.
        - If a group join consent is received, transition back to the dormant state.
      - In the receive state:
        - Handle the appropriate actions for receiving group-related signals.
  - If no group formation is initiated, reset the group formation started flag.
```

Step-3. End of algorithm.

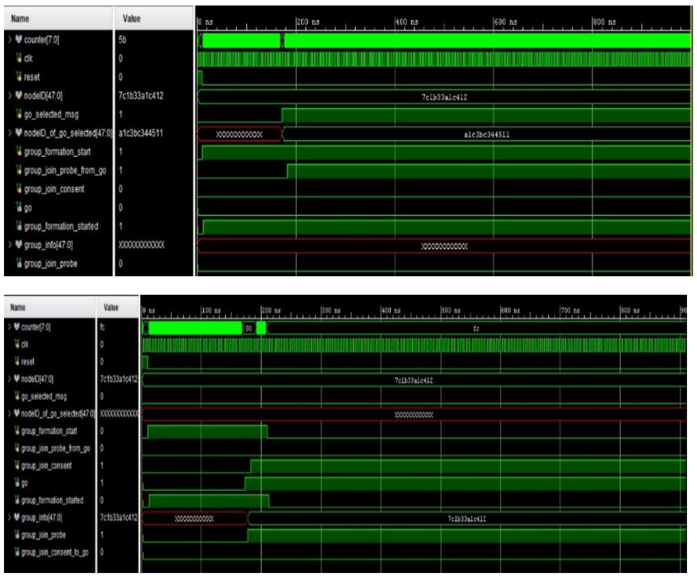


Fig -: GO simulation results

REFERENCES

1. F.A. Samman, F. Philipp and M. Glesner, “Reconfigurable interconnect infrastructure for multi-FPGA-based adaptive multiprocessing systems," 2011 1st International Workshop on Computing in Heterogeneous, Autonomous 'N' Goal-Oriented Environments (CHANGE), Newport Beach, CA, USA, 2011, pp. 1-8, doi: 10.1109/CHANGE.2011.6172451 .
2. Q.Ijaz and E. -B. Bourennane, “Wireless versus Wired Network-on-chip to enable the Multiu-Tenant Multi-Fpga in cloud,” 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 2021, pp. 1-6, doi:10.1109/IEMTRONICS52119.2021.9422654
3. Mango Communications, Inc. 802.11 mac/phy design. [Online].
4. P. Shrestha, B. Subedi, M. Girard, C. Parikh and N. Kandalaft, "Wireless Communication Between FPGA and Microcontroller,". 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0402-0405, doi: 10.1109/CCWC47524.2020.9031196.
5. C. Subedi, B. P. Beauchamp and N. Kandalaft, "MQTT based Wi-Fi communication between Microcontroller and FPGA," 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 2021, pp. 1-4, doi: 10.1109/IPCCC51483.2021.9679442.
6. X. Jiao, W. Liu, M. Mehari, M. Aslam and I. Moerman, "openwifi: a free and open-source IEEE802.11 SDR implementation on SoC," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-2, doi: 10.1109/VTC2020-Spring48590.2020.9128614.
7. C. E. Casetti, C. F. Chiasserini, Y. Duan, P. Giaccone and A. Perez Manriquez, "Data Connectivity and Smart Group Formation in Wi-Fi Direct Multi-Group Networks," in IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 245-259, March 2018, doi: 10.1109/TNSM.2017.2766124.
8. Wi-Fi Alliance , “Wi-Fi Peer-to-Peer Services (P2Ps) Technical Specification”. Version 1.2 ,2014