CR-GRT: A Novel SDR Platform Optimized for Real-time Cognitive Radio Applications

Jun Liu
Peking University
Beijing, China
juneliu@pku.edu.cn

Haoyang Wu
Peking University
Beijing, China
wuhaoyang@pku.edu.cn

Zhiwei Li
Peking University
Beijing, China
zhiwei.li@pku.edu.cn

Boyan Ding
Peking University
Beijing, China
dboyan@pku.edu.cn

Tao Wang
Peking University
Beijing, China
wangtao@pku.edu.cn

ABSTRACT

Cognitive radio (CR) technology aims to provide real-time sensing and efficient dynamic spectrum access to improve the efficiency of spectrum resource usage. However, none of the existing SDR platforms is capable of supporting CR applications while maintaining high performance and programmability. In this paper, we propose CR-GRT, an SDR platform designed for cognitive radio applications. CR-GRT supports real-time sensing, analysis, decision-making and dynamic adjustment. It also provides interfaces for extensibility. Based on CR-GRT, we implement a comprehensive sensing strategy using both PHY and MAC information. The evaluation result shows that CR-GRT has advantages in high performance and programmability.

CCS CONCEPTS

• **Networks → Network architectures**; **Cognitive radios**;

KEYWORDS

software-defined radio, cognitive radio, FPGA

ACM Reference Format:

Jun Liu, Haoyang Wu, Zhiwei Li, Boyan Ding, and Tao Wang. 2018. CR-GRT: A Novel SDR Platform Optimized for Real-time Cognitive Radio Applications. In *21st ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM '18), October 28-November 2, 2018, Montreal, QC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3242102.3242114

1 INTRODUCTION

The rapid development of wireless communication has made the limited spectrum resource increasingly scarce. Non-authorized frequency bands are becoming increasingly crowded. However, research [14] discovered that about 70% of the spectrum resources in non-authorized bands were not used effectively. To improve the spectrum utilization, the concept of cognitive radio emerged [15, 18].

The general approach of CR can be divided into three steps: sensing, decision-making and rapid control [7, 16, 17]. CR systems can change communication parameters (such as transmission power, bandwidth and modulation schemes) in real-time through the learning of the current environment. Thus, besides ensuring the performance of real-time wireless communication, CR system should also incorporate algorithms for sensing, decision making, and to change wireless parameters rapidly [19].

Software defined radio (SDR) platforms, because of their versatility, are ideal for coginitive radio research. Unfortunately, none of state-of-art SDR systems have been optimized for cognitive radio.

Considering the requirements for cognitive radio, we propose a new SDR platform for real-time CR application with high performance and programmability. The main contributions of this paper can be summarized as:

- We design a novel SDR platform for CR applications, achieving real-time sensing, decision-making and rapid control.
- We implement CR-GRT based on FPGA to meet the high performance and reconfigurable requirements, enabling users to rapidly develop and validate related CR theories and algorithms.
- We propose a comprehensive sensing strategy that includes instantaneous sensing and statistical sensing, and make different cognitive decisions based on different scenarios and the information obtained.
- We provide sufficient and simple user interfaces on the host side, low MAC and PHY layers for sensing information and control information.

The remainder of this paper is organized as follows. Section 2 introduces related works of CR applications and presents several state-of-art SDR platforms. Section 3 analyzes the overall design of CR-GRT and describes each component in detail. Section 4 describes the implementation of CR-GRT platform. Section 5 evaluate the functionality and performance of CR-GRT. Section 6 concludes this paper and discusses our future work.

MSWIM '18, October 28-November 2, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5960-3/18/10…$15.00
https://doi.org/10.1145/3242102.3242114

2 RELATED WORK

Recent years have witnessed plenty of researches in cognitive radio due to its potential in more effective spectrum utilization, in which a number of SDR platforms are used.

SpecInsight [21] is a set of real-time broadband spectrum sensing system. However, limited by the performance of USRP [5], it cannot carry out analysis or decision-making in real time.

For dynamic configuration of channel frequency and bandwidth, a spectrum adaptive system called SEER [27] is proposed. Similar to the case of SpecInsight, the performance limitation of USRP also cripples SEER's ability to perceive the ambient environment in real time.

As shown above, the majority of the relevant researches are not verified on a complete real-time platform. The effectiveness of the algorithm is usually determined by simulation. Universal software-based SDR platforms such as USRP [5], GNU Radio [4], Sora [10, 24], Iris [22] and OSSIE [23] are highly programmable, but impossible to realize real-time sensing and decision making.

BEE [25] and WARP [13] are typical hardware-based SDR platforms. BEE uses a PowerPC core in FPGA for minimized latency and maximized data throughput, but it does not provide structural design in PHY and MAC layer, making it difficult to program. WARP ensures high performance. However, it does not provide sufficient support for modular design. In addition, WARP doesn't provide a friendly interface with host, making it difficult to connect to the upper layer for further decision making. So, hardware-based SDRs are difficult to implement cognitive radio application.

We previously proposed an SDR platform called GRT [26, 29], which can achieve theoretical 54Mbps data rate with standard 802.11a/g. GRT achieves real-time communication while ensuring high programmability through software and hardware co-design. Therefore, compared with other SDR platforms, the GRT platform has greater advantages for CR applications. What we need to do in this paper is to optimize GRT for the cognitive radio requirements while maintaining performance and programmability.

3 DESIGN OF CR-GRT

3.1 Design Goals

Our goal is to realize a high-performance, programmable SDR platform for CR application. High-performance implies real-time sensing, low-latency information transmission and fast decision making. While programmability includes the supports of API for PHY, MAC and upper network layers, and facilitates the implementation of sensing and decision making algorithms.

3.2 Design Challenges

The following three challenges arise when considering the design of CR-GRT.

(1) **PHY & MAC layer sensing.** PHY layer data processing pipeline needs to ensure high performance. At the same time, it needs to obtain various sensing information in the pipeline. How to obtain the spectrum sensing information of PHY layer and MAC Layer in real-time without affecting normal communication is a challenge. We also need to solve the problems that how to analyze spectrum conditions from these information [9].

(2) **Low-latency and high throughput transmission.** The sensing and transmission latency of PHY layer and MAC layer is very important for CR applications [20]. How to achieve low latency and high throughput transmission of sensing information is a challenge. In some cases, there is a large amount of sensing information (such as constellation points) need to be transmitted. Transmitting these information is another challenge.

(3) **Decision-making strategy.** The joint sensing of PHY layer and MAC layer can greatly improve the sensing capability. The challenge is what kind of strategy can be used to cross-layer information for decision-making in different scenarios. How to change the parameters of PHY and RF according to the different analytic result is also a big issue.

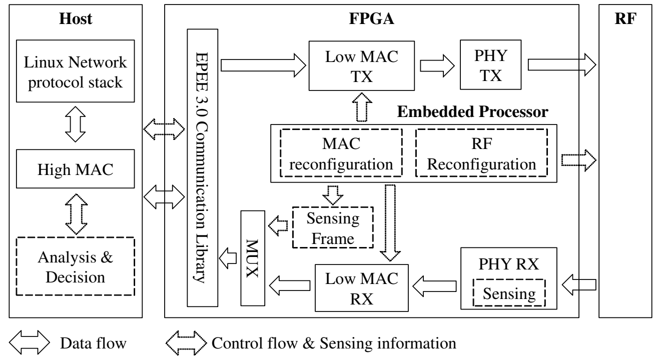

Figure 1: Overall architecture design of CR-GRT

3.3 CR-GRT Architecture Design

To deal with the above-mentioned challenges, we propose the corresponding architectural designs of CR-GRT.

Overview of CR-GRT. A hierarchical design of CR-GRT architecture is shown in Figure 1. CR-GRT consists of three main components, RF front-end, hardware logic in FPGA and software logic in host. Specifically, RF front-end controls the transceiver of wireless signals. Low-MAC logic and PHY processing modules are implemented on FPGA, with RF and USB library drivers attached. We provide flexible interface with host to ensure the customized wireless implementations. Host provides corresponding drivers and the development environment. A user-friendly communication library between host and FPGA enables an efficient software-hardware co-design. Therefore, users can control and change the working mode of the wireless networks through the APIs we provided.

Design of PHY layer. There are many intermediate processing results in receiver of PHY that can indicate the current wireless channel status, such as signal noise ratio (SNR), channel state information (CSI), received signal strength indicator (RSSI), frequency offset, etc. Some of them can be directly obtained from processing modules in PHY, such as frequency offset. Others need further processing after extracting raw data from the pipeline of PHY layer.

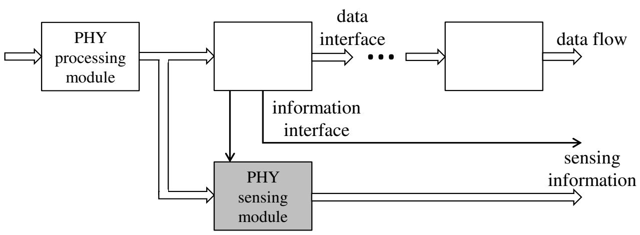

Figure 2: Design of PHY structure

For example, we can get SNR result through statistical analysis of constellation points.

PHY layer design needs to collect and process variety of information from different pipeline stages in PHY layer in real-time, while ensuring the performance of baseband processing. As shown in Figure 2, PHY sensing modules should be added into the baseband data processing pipeline.

There are two ways to acquire sensing information. One is the data flow between PHY processing modules in the pipeline, which requires higher throughput. The other is the intermediate result inside data processing module, which is more sensitive to latency. We design two interfaces, data interface is used to support the data flow, while information interface used to deliver the intermediate result.

Sensing module is mainly used for further statistical analysis of the information from the PHY layer pipeline. The structure of sensing modules is the same as data modules. The input of a sensing module either comes from the data or the information interface. After processing by the sensing module, the result can be output through the information interface. In some specific cases, there are still a large amount of data after processing. We also support data interface to handle such large amount of data.

In order to meet the low latency requirements, both data and information interfaces need to be carefully designed. The widths of data interface can be changed dynamically to ensure a high degree of parallelism. We also adopt simple control and status registers (CSRs) in CR-GRT to ensure low-latency.

We adopt loosely coupled connections between each modules in PHY. With this design, adding a sensing module is simple and will not affect other data processing modules.

Design of MAC layer. MAC layer needs to obtain the sensing information from PHY layer for real-time decision-making and control, e.g., changing parameters of PHY and RF. MAC also provides auxiliary information for better sensing effect. For example, MAC layer can detect hidden nodes through CTS/RTS frames, or estimate the number of nearby wireless devices through the analysis of MAC address, which is hard to achieve, if not impossible, with PHY layer information alone.

Figure 3 illustrates the MAC layer's structural design of CR-GRT. It is an extension to our previous work GRT, which provides a high-performance and extensible MAC framework, including an embedded processor and several IPs, all connected to a virtual bus.

In cognitive radio context, we use the embedded processor to support simple analysis and decision-making algorithms. The dedicated, light-weight processor can also guarantee the real-time state machine scheme of low-MAC. The extensible IPs are used to offload heavy tasks. Thus more sensing IPs can be added to MAC layer.

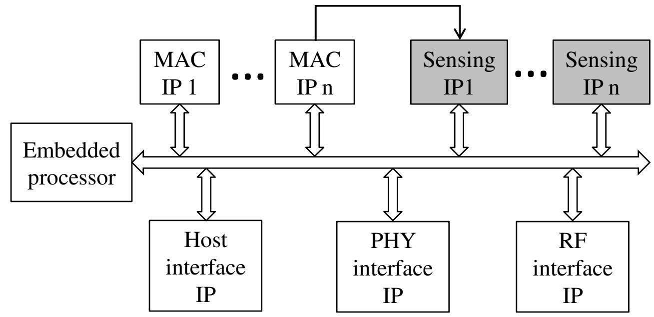

Figure 3: MAC layer structural design

We have defined several types of IPs, as shown in Figure 3. MAC IP is responsible for the basic functions of low-MAC. Sensing IP is used to further process collected MAC information. There are also specified interface IPs to handle the communication between MAC and other parts, including PHY, host and RF. All IPs communicate with embedded processor using bus interface.

Similar to PHY layers, Sensing information in MAC layer comes from two sources: the embedded processor and intermediate result of MAC IPs.

Cognitive radio can be classified into instantaneous decision-making and statistical decision-making in different scenarios.

Instantaneous decision-making. For instantaneous decision-making, cognitive radio platform need to quickly capture the spectrum holes, change the center frequency and bandwidth rapidly. In this strategy, real-time sensing information at the PHY layer is sent to the PHY interface IP. Then, the embedded processor acquires the information through the bus interface, after simply analysis and decision-making algorithms, the embedded processor sends the control information to the PHY interface IP and RF interface IP to change the parameters of PHY layer and RF front-end.

Statistical decision-making. Statistical decision-making requires more statistical information from PHY and MAC layer, so that the upper layer of wireless network can determine the optimal transmission parameters through statistical analysis and comprehensive judgement.

For cognitive radio research, CR-GRT needs to handle low-latency communication between MAC and PHY. The communication includes downlink control and uplink sensing information. We use a simple unidirectional interface to handle low-latency communication between MAC and PHY. The unidirectional interface is separate from standard interface. With this dedicated communication interface, MAC layer can obtain the latest sensing information from PHY.

Software design in host. The host and FPGA are connected with the EPEE communication library [11, 12], which handles the data transmission. Host receives sensing information from PHY and MAC to make decisions, then control information is sent back to MAC, further help to change the parameters in PHY and RF. Host also provide user interface with APIs. We can implement custom analysis and decision algorithms on the host.

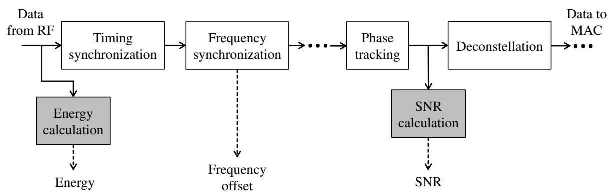

Figure 4: CR-oriented PHY layer implementation

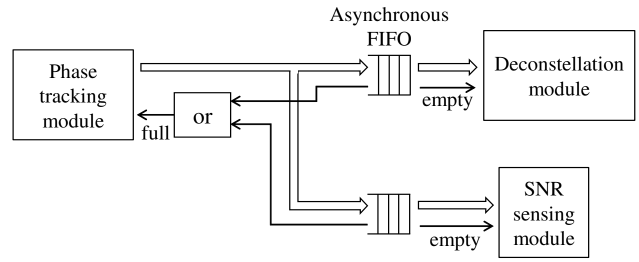

Figure 5: Implementation of SNR sensing module

EPEE supports either PCIe or USB for communication between host and FPGA with the same interface to the user. For lower latency application, we choose the PCIe communication library. USB communication library is used in portable application. EPEE provides three interfaces: bulk data interface, register interface and interrupt interface, to facilitate efficient communication between host and FPGA. The bulk data interface is designed for high-performance frame data transmission. Register interface delivers control and status information between host and FPGA. Interrupt interface informs host some events happened in FPGA. In CR-GRT, we use bulk data interface to transmit data and uplink sensing information. Register interface transmits downlink control parameters. Interrupt interface to use the above two interfaces.

4 IMPLEMENTATION

In this section, we will mainly describe the implementation of new designs in CR-GRT, including the implementation of PHY, MAC and host control. We will provide some specific cases for each component. Then, we will clarify the cognitive radio process using CR-GRT platform.

4.1 Implementation of CR-GRT Platform

Implementation of PHY. As we discussed, PHY layer has a simple pipeline consisting of baseband processing modules and sensing modules. The module connection at PHY receiver is shown in Figure 4. Sensing information with frequency offset, energy value and SNR has been illustrated.

There are three key points in PHY implementation: (1) achieving low coupling module connections; (2) separating data from the original pipeline into the sensing module; and (3) implementing the sensing algorithms in PHY.

In order to achieve low coupling PHY connections, we adopt asynchronous FIFO interface between each module. As a buffer zone, the asynchronous FIFO matches rates of different modules. Thus the sensing module can work in a different clock domain than the data processing modules. Since the clock domain of sensing module is not limited by the global clock, we can increase the clock frequency of the sensing module as much as possible, which has positive effect on improving the performance of sensing modules. In addition, when a sensing module is added to cognitive radio system, it will never affect the normal processing of the exist baseband processing pipeline. To prevent overflow and underflow when using FIFO, each FIFO provides an empty and a full signal.

To separate data flow from PHY processing pipeline, we implement two-way ports based on standard asynchronous FIFO as shown in Figure 5.

Taking SNR calculation as an example, after the phase tracking module, the constellation point has been generated in the output port. The same constellation data is written to both FIFOs at the same time. The next processing module and the sensing module can get data from the two FIFOs respectively. It should be noticed that both FIFOs have *full* signal. The two *full* signals connect to the former processing module through an “or” logic gate. The former module blocks when either FIFO is *full*. Addition of an SNR sensing module does not affect the overall structure of the original PHY pipeline.

Then, we take energy value and SNR calculated by constellation point as two examples to show the implementation of PHY sensing modules. For other sensing information like frequency offset and CSI, we can directly obtain them from frequency synchronization module and channel estimation module.

Energy value uses a relatively simple signal detection method that does not require the prior information of the primary user signal. The energy value is obtained by the calculation of the baseband sampling waveform. Compared with the threshold, we can indicate that the current channel status is either *busy* or *idle*. If the energy value is greater than the threshold, we know that there is a primary user in the frequency band and we should avoid accessing this channel for now, otherwise, we can use it immediately. The energy value is denoted by E, shown in Equation 1.

$$E = 10\,\log\left(\frac{1}{n}\sum_{n=0}^{N-1} Y(n)^2\right) \tag{1}$$

In this equation, $Y(n)$ denotes baseband sampling values, we square each point and accumulate consecutive N values together.

However, the energy value E is inaccurate due to automatic gain control (AGC) in RF front-end. Therefore, it is necessary to compensate energy value with the AGC gain. If the AGC gain is denoted by G, the calibrated energy value E' is shown in Equation 2:

$$E' = E - G \tag{2}$$

SNR is another common sensing information in cognitive radio systems. It reflects the quality of the current channel. Higher SNR indicates better channel, on the other hand, lower SNR suggests larger noise. The constellation points in receiver can intuitively

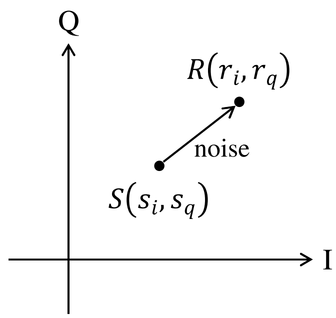

Figure 6: Constellation point in Tx and Rx

represent SNR, thus providing a convenient way to study the performance distribution both in time-domain and frequency domain. Higher density of constellation points indicates higher SNR.

Figure 6 depicts the calculation of SNR. In transmitter, each constellation data should be mapped to a fixed location. However, affected by noise, the constellation point in receiver will deviate from its original fixed location. In Figure 6, R denotes a constellation point in receiver. S denotes the original signal location in transmitter. Their distance reflects the channel noise. According to the statistical calculation with N successive constellation points, SNR can be calculated as:

$$SNR = 10\ \log\left(\frac{\sum_{n=0}^{N-1}\left(s_i(n)^2 + s_q(n)^2\right)}{\sum_{n=0}^{N-1}\left((r_i(n) - s_i(n))^2 + (r_q(n) - s_q(n))^2\right)}\right) \quad (3)$$

$s_i(n)$ and $s_q(n)$ denote the real and imaginary parts of S respectively. While $r_i(n)$ and $r_q(n)$ denotes signal R.

Since the average energy of transmitter normalized to 1 in 802.11 protocol, the calculation of SNR can be simplified to Equation 4

$$SNR = 10\ \log\left(\frac{N}{\sum_{n=0}^{N-1}\left((r_i(n) - s_i(n))^2 + (r_q(n) - s_q(n))^2\right)}\right) \quad (4)$$

In our implementation, we not only calculate the overall SNR, but also calculate SNR distribution over time domain and the frequency domain. If the SNR changes drastically over time domain, it possibly means the channel status changes quickly. We choose shorter frames in this circumstance. Low SNR value in some specific frequencies means large noise in these subcarriers. So cognitive radio need to change frequency band to avoid accessing these subcarriers.

Implementation of Low MAC. Figure 7 shows the implementation of MAC. The MAC layer consists of an embedded processor and multiple IPs. Embedded processor is implemented with MicroBlaze. IPs are directly implemented in hardware logic, which are connected to the embedded processor using AXI bus.

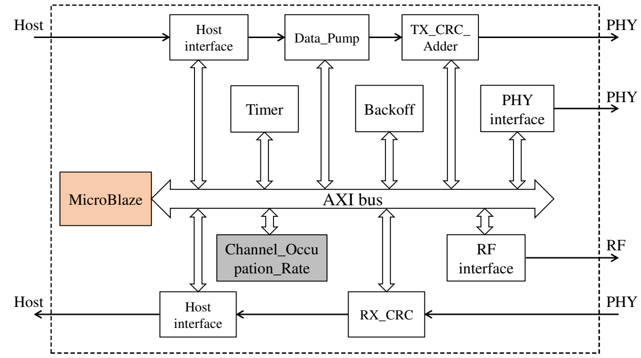

Figure 7: Microblaze and IPs in low MAC

According to Figure 7, MAC IPs include Timer, Backoff, etc. These IPs support the basic functions of MAC. Sensing IPs are also connected to the AXI bus. For example, The Channel_Occupation_Rate IP downloads the timestamps and lengths of all frames from MicroBlaze, analyzes the current channel utilization for a period of time, and delivers this information to host interface IP. The interface IPs (host interface, PHY interface, RF interface) provide communication interfaces between MAC and other layers. Standard AXI bus is used for communication between the MAC and all IPs.

Next, we use the calculation of average backoff time and channel occupation rate as two examples to illustrate how to carry out the MAC implementation.

The average backoff time indicates the mean waiting time for a frame to start transmitting in the air. If the average backoff time is short, the channel is considered to be idle. If the average backoff time is above a certain threshold, MAC of CR-GRT will consider changing the frequency bands. The calculation of average backoff time is relatively simple and does not require extra sensing IP. When MAC gets a frame, MicroBlaze acquire a start time from Timer IP. After Backoff IP informs the frame to begin transmit, MicroBlaze reads an end time from the same Timer IP. The average backoff time is obtained with a subtraction of the two time values. Microblaze tells the host the average backoff time through host interface IP.

From received frames at the MAC layer, we can also estimate the current channel occupation rate. First, for each received frame, we can calculate the frame durations in the air with parameters in MAC layer, including frame length, data rate and modulation schemes. After that, we add up durations of all frames together over a period of time. The current channel utilization rate can be calculated compare to the total interval time.

Moreover, the MAC layer can also collect the NAV information in CTS/RTS, which indicates the occupied time of hidden nodes. With these information, MAC layer can provide more accurate channel occupation rate. The statistics calculation is implemented on Channel_Occupation_Rate IP. This sensing IP obtains information from other MAC IPs, with corresponding processing algorithms, and then sends the information to the host interface IP.

There is more MAC information that can also be used to assist in the decision-making, such as the number of surrounding devices with the count of received MAC address and the number of retransmissions, etc.. They can provide auxiliary information for channel analysis in host.

Implementation of host. In CR-GRT, the host can obtain a lot of sensing information, and can also actively control the behavior of the low MAC and PHY layers in the FPGA, such as parameter modification, spectrum sensing, and so on.

Although the latency of host-based analysis and decision-making is larger than MAC, it is possible to support more complex and detailed analysis and spectrum control strategies.

The host can collect the sensing information both from PHY and MAC layer. At the PHY layer, information such as CSI, energy value, SNR and frequency offset can be obtained. The MAC layer can get information such as average backoff time, channel occupation rate, number of surrounding devices and the number of retransmissions. The following codes shows the structures of the sensing information interface:

```
struct cr_phy_info{
    u8 csi_variance; // CSI fluctuations
    u8 energy_val; // Mean energy over time
    u8 snr;
    u8 frequency_offset;
    ...
};

struct cr_mac_info{
    u8 ave_backoff_time;
    u8 channel_occupation_rate;
    u8 dev_num;   //number of devices
    u8 retry_num; //number of retransmissions
    ...
};
```

The user can evaluate and analyze the current channel status according to the strategies mentioned in next subsection, finding spectrum holes, complet the decision-making process, and change the PHY and RF parameters. The following code structure shows how to configure these parameters:

```
struct cr_config {
        u32 retry_limit;
        u32 sifs_time;
        u32 slot_time;
        u32 ack_timeout;
        u32 tx_freq;
        u32 rx_freq;
        u32 sample_rate;
        u32 bandwidth;
        u32 tx_atten;   // TX attenuation
        ...
};
```

4.2 Comprehensive Sensing Strategy

The comprehensive sensing strategy includes instantaneous and statistical sensing. For instantaneous sensing, CR-GRT needs to quickly capture the spectrum holes, changing the center frequency and bandwidth rapidly to use the current spectrum resources more efficiently. Statistical sensing requires more statistical information

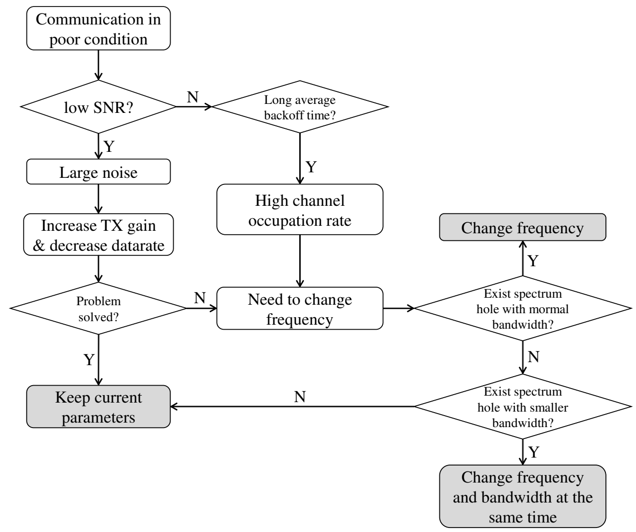

Figure 8: Cognitive process of CR-GRT

from PHY and MAC layer. So the host can adapt the transmission parameters through statistical analysis and comprehensive judgement.

With PHY and MAC sensing information, we can further calculate the current channel status including the channel noise level, channel occupation rate and channel stability. If there is large noise in the surrounding environment, low MAC should quickly modify the RF parameters (such as modulation scheme, transmission gain, etc.). Using this way, CR-GRT achieves real-time decision-making process. If the communication status still doesn't reach the desired level, it's better to upload sensing information to the host. Then, the host carries out a detailed analysis of various information and adopt the best parameter reconfiguration scheme.

Figure 8 illustrates the implementation of the strategy. In this implementation, we use channel noise level and channel occupation rate as the basic information. The SNR and the average number of retransmissions determine whether the noise is large. If the noise is large, low MAC will make an instantaneous decision and use a more robust modulation scheme to increase the transmission gain. If the noise is small, and the communication status is still bad, we continue to send the average backoff time of the MAC layer to host. We carry out the analysis in host to determine whether the bad communication status is due to the large number of devices in the band or the high channel occupation rate. If the channel occupation rate is high, it is necessary to change the center frequency and bandwidth to optimize the communication status.

5 EVALUATION

This section uses a series of experiments to evaluate the CR-GRT platform. We first demonstrate the functionality of CR-GRT. Next, we assess the overall performance of throughput, resource consumption and latency of CR-GRT. In this section, we mainly use 802.11a/g [8] for validation.

Figure 9: Evaluation setup of CR-GRT

5.1 Experiment setup

As shown in figure 9, we use AD9361 [2] board with FMC interface as RF front-end device. PHY and low MAC layers are implemented on the Xilinx KC705 [6] evaluation board. Specifically, Microblaze provided by Xilinx is adopted to realize the low MAC logic, AXI bus is used to connect the hardware IPs. We use Cypress CYUSB3KIT-003 [3] explorer kit and an FMC interconnect board [1] to support the implementation of USB 3.0 communication library. Ubuntu 14.04 operating system is used on host computer. We develop verilog HDL code and C code of low MAC in VIVADO 2015.2 and SDK respectively.

5.2 Functionality test

According to the PHY layer sensing information, we can perform detailed analysis, such as spectrum occupation rate with fixed frequency bands over time and different frequency bands. In order to verify the above statement, we conducted two experiments with CR-GRT.

We first test the spectrum occupation rate from 2.402 GHz to 2.502 GHz. The test is conducted in our lab with normal WiFi traffic going on around, except that in the 2.492GHz channel, we used other GRT systems to simulate a congested channel. As shown in Figure 10, the horizontal axis is the center frequency, and the vertical axis is the spectrum occupation rate. We can see that the occupation rate of the 2.452 GHz band is less than 3% and can be regard as a spectrum hole over frequency. The occupation rate of the 2.492 GHz band reaches 72.29%, it's not a good choice to send wireless frames in this channel.

Then, we monitor the energy value over a period of time in 2.492 GHz band. As shown in Figure 11, the energy value changes quickly over the time. The intervals between two energy bars represent the spectrum hole over time.

From the above two experiments, CR-GRT can sense the spectrum holes among wide frequency range. It can also provide energy value in microsecond level. So CR-GRT can meet the requirements of high performance cognitive radio applications.

5.3 Performance Evaluation

Next, we test the performance of the CR-GRT platform, including resource utilization, throughput and latency.

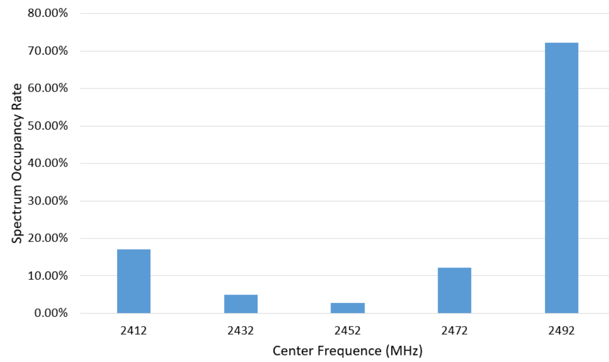

Figure 10: Spectrum occupation rate with different frequency

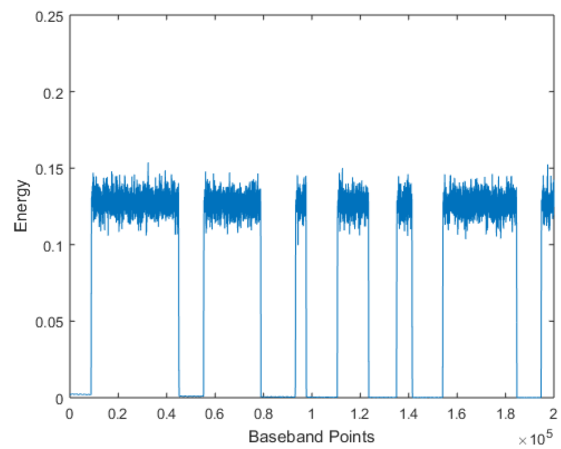

Figure 11: Spectrum occupation rate over time

Table 1: Resource utilization

Resource	Utilization	Available	Usage
Slice LUTs	98359	203800	48.26
Slice Registers	77035	407600	18.90
Memory	251	445	56.40
DSP	321	840	38.21

Resource utilization. Table 1 lists the resource utilization of CR-GRT. LUT (look-up table) and registers are main resources on FPGA used for logic and storage respectively. We can see that 48.26% of LUTs and 18.9% of the register resources have been used in KC705 FPGA evaluation board. Therefore, sufficient resources have been remained for sensing algorithms.

Throughput. We set up four experimental variables, which include modulation types, frame length, center frequency and bandwidth. All of these parameters can be dynamically modified for better use of spectrum holes to improve communication performance and reliability. During the experiment, the distance between

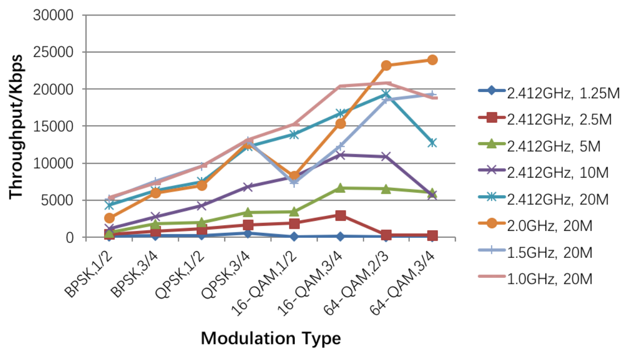

Figure 12: Throughput with different center frequency and bandwidth of customized Wi-Fi

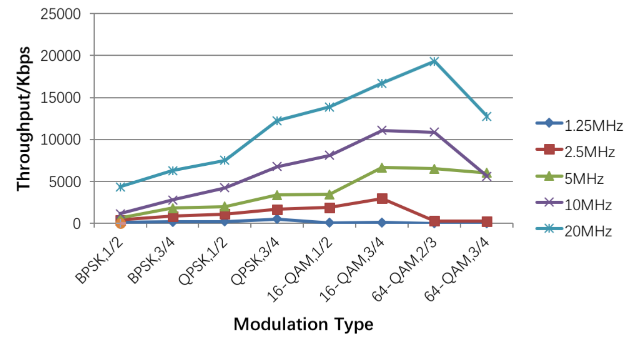

Figure 13: Throughput with different bandwidth of customized Wi-Fi

the two devices and the TX gain are fixed. All the following experiments are tested in a real indoor environment.

Figure 12 illustrates the throughput in different configurations, plotted by modulation types on the horizontal axis and throughput on the vertical axis. From Figure 12, we achieve maximum throughput of over 23Mbps. In this test, we found that two CR-GRT devices can communicate with each other using very low bandwidth.

Further more, we test throughput in different bandwidths with fixed center frequency of 2.412GHz (standard Wi-Fi channel 1). The experimental result is shown in Figure 13. Compared to the 23Mbps in 20MHz bandwidth, we can get the maximum throughput of 6Mbps in 5MHz bandwidth. So narrowing bandwidth does not affect the communication efficiency.

With fixed bandwidth of 20MHz, we get the throughput of different center frequencies, shown in Figure 14. On different frequencies, we get different throughput. This experiment shows that CR-GRT has the ability to communicate in non-standard frequency bands to avoid collisions. The RF front-end we selected supports the frequency range from 70 MHz to 6 GHz, so users can have more choices of center frequency.

Finally, we test the throughput with different frame length. In our common sense, higher order of modulation or higher frame length can help achieving higher transmission throughput. In real wireless environment, however, throughput is influenced by factors such as noise and multi-path fading. As shown in Figure 15, in the 64-QAM modulation mode, a 2048-Bytes-length frame achieves a higher throughput than a 4091-Bytes-long frame. Therefore, in the scenario of dynamic spectrum access, when the signal environment is poor, we can reduce the frame length to decrease the error rate and improve the transmission stability.

Throughput. Table 2 to Table 4 show the various aspects of CR-GRT latency we tested. The latency information is divided into three categories:

(1) Table 2 shows the delay of sensing information transmission. It is mainly caused by the EPEE communication library. In the case of the PCIe connection, the latency can be reduced to 15.6us/15.9us.
(2) Table 3 shows the delay of sensing information processing inside the FPGA. Since this part of the function is implemented by the FPGA hardware logic, it can also be completed in a very short time.
(3) Table 4 shows the delay of parameter reconfiguration. The parameter reconfiguration delay of MAC layer and PHY layer is almost negligible. After finishing the transmission of the current frame, these parameters can be reconfigured. The RF delay is limited by the RF board. Low latency of parameter reconfiguration can be guaranteed in CR-GRT.

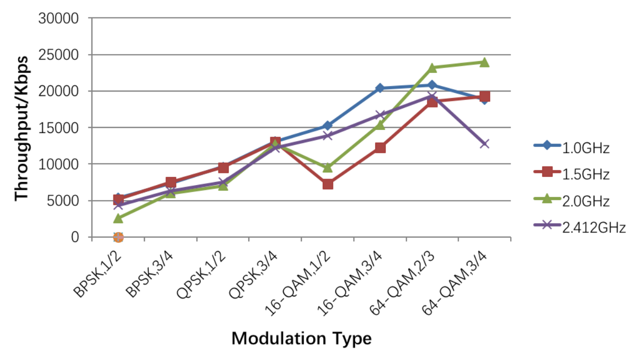

Figure 14: Throughput with different center frequency of customized Wi-Fi

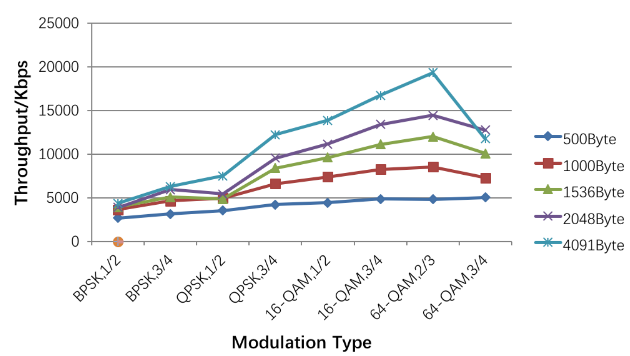

Figure 15: Throughput with different length of standard Wi-Fi configuration

As we can see from the three tables, the interaction between host and FPGA, and the interaction between PHY and MAC layer all achieve the low-latency target.

Table 2: Sensing information transmission delay

Item	Latency
host to FPGA	52.6us (USB), 15.9us (PCIe)
FPGA to host	52.6us (USB), 15.6us (PCIe)
PHY to MAC	Real time

Table 3: Sensing information processing delay

Item	Latency
MAC layer	<3us
PHY layer	<0.3us

Table 4: Parameter reconfiguration delay

Item	Latency
MAC layer parameters	Real time
PHY layer parameters	Real time
RF center frequency	1467us
RF bandwidth	2244us
RF transmit gain	3785us

6 CONCLUSION

In this paper, we present CR-GRT, a novel high-performance and programmable SDR platform for cognitive radio. CR-GRT mainly solves the problem of real-time spectrum sensing for PHY and MAC layers. It can obtain the more accurate and detailed spectrum conditions in order to give the best strategies. The evaluation result shows that, by carefully designing the architecture of PHY and MAC, CR-GRT is suitable for CR application for high performance and programmability. Our future work is to support higher bandwidth and throughput on Tick [28] platform, and verify more sensing algorithms and models for cognitive radios.

ACKNOWLEDGMENTS

The work is funded by National Key Research and Development Plan of China (2017YFB0801702) and key research project of National Natural Science Foundation (No. 61531004).

REFERENCES

[1] 2014. CYUSB3ACC-005 FMC Interconnect Board. http://www.cypress.com/documentation/development-kitsboards/cyusb3acc-005-fmc-interconnect-board-ez-usb-fx3-superspeed.

[2] 2018. AD9361. http://www.analog.com/en/products/rf-microwave/integrated-transceivers-transmitters-receivers/wideband-transceivers-ic/ad9361.html.

[3] 2018. CYUSB3KIT-003 SuperSpeed Explorer Kit. http://www.cypress.com/documentation/development-kitsboards/cyusb3kit-003-ez-usb-fx3-superspeed-explorer-kit.

[4] 2018. GNU Radio. http://www.gnuradio.org/.

[5] 2018. USRP Family of Products. https://www.ettus.com/product.

[6] 2018. Xilinx KC705 evaluation board. http://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html.

[7] Wenchi Cheng, Xi Zhang, and Hailin Zhang. 2017. Pilot-based full-duplex spectrum-sensing and multichannel-MAC over non-time-slotted cognitive radio networks. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE.* IEEE, 1–9.

[8] IEEE Computer Society LAN MAN Standards Committee et al. 1997. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Standard 802.11-1997* (1997).

[9] Claudia Cormio and Kaushik R. Chowdhury. 2009. A survey on MAC protocols for cognitive radio networks. *Ad Hoc Networks* 7, 7 (2009), 1315–1329.

[10] Ji Fang, Zhenhui Tan, and Kun Tan. 2011. Soft MIMO: A software radio implementation of 802.1 In based on Sora platform. (2011).

[11] Jian Gong, Jiahua Chen, Haoyang Wu, Fan Ye, Songwu Lu, Jason Cong, and Tao Wang. 2014. EPEE: an efficient PCIe communication library with easy-host-integration property for FPGA accelerators. In *Proceedings of the 2014 ACM/SIGDA international symposium on Field-programmable gate arrays.* ACM, 255–255.

[12] Jian Gong, Tao Wang, Jiahua Chen, Haoyang Wu, Fan Ye, Songwu Lu, and Jason Cong. 2014. An efficient and flexible host-fpga pcie communication library. In *Field Programmable Logic and Applications (FPL), 2014 24th International Conference on.* IEEE, 1–6.

[13] Ahmed Khattab, Joseph Camp, Chris Hunter, Patrick Murphy, Ashutosh Sabharwal, and Edward W Knightly. 2008. WARP: a flexible platform for clean-slate wireless medium access protocol design. *ACM SIGMOBILE Mobile Computing and Communications Review* 12, 1 (2008), 56–58.

[14] Paul Kolodzy and Interference Avoidance. 2002. Spectrum policy task force. *Federal Commun. Comm., Washington, DC, Rep. ET Docket* 40, 4 (2002), 147–158.

[15] Eva Lagunas and Montse Nájar. 2015. Spectral feature detection with sub-Nyquist sampling for wideband spectrum sensing. *IEEE Transactions on Wireless Communications* 14, 7 (2015), 3978–3990.

[16] Zhi-rui LI, Lu-hua ZHAO, and Wan-li CHENG. 2015. Optimal linear cooperation for spectrum sensing in cognitive radio. *Information Technology* 1 (2015), 047.

[17] Xingya Liu and Jiang Xie. 2017. A 2D heterogeneous rendezvous protocol for multi-wideband cognitive radio networks. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE.* IEEE, 1–9.

[18] Joseph Mitola and Gerald Q Maguire. 1999. Cognitive radio: making software radios more personal. *IEEE personal communications* 6, 4 (1999), 13–18.

[19] Przemyslaw Pawelczak, Keith Nolan, Linda Doyle, Ser Wah Oh, and Danijela Cabric. 2011. Cognitive radio: Ten years of experimentation and development. *IEEE Communications Magazine* 49, 3 (2011).

[20] Thomas Schmid, Oussama Sekkat, and Mani B Srivastava. 2007. An experimental study of network performance impact of increased latency in software defined radios. In *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization.* ACM, 59–66.

[21] Lixin Shi, Paramvir Bahl, and Dina Katabi. 2015. Beyond sensing: multi-GHz realtime spectrum analytics. In *Usenix Conference on Networked Systems Design and Implementation.* 159–172.

[22] Paul D Sutton, Jorg Lotze, Hicham Lahlou, Suhaib A Fahmy, Keith E Nolan, Baris Ozgul, Thomas W Rondeau, Juanjo Noguera, and Linda E Doyle. 2010. Iris: an architecture for cognitive radio networking testbeds. *IEEE communications magazine* 48, 9 (2010).

[23] M Imran Taj, O Hammami, and K Huggins. 2009. Performance evaluation of SDR on embedded platform: The case of OSSIE. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on.* IEEE, 1–5.

[24] Kun Tan, He Liu, Jiansong Zhang, Yongguang Zhang, Ji Fang, and Geoffrey M Voelker. 2011. Sora: high-performance software radio using general-purpose multi-core processors. *Commun. ACM* 54, 1 (2011), 99–107.

[25] Artem Tkachenko, Danijela Cabric, and Robert W Brodersen. 2007. Cyclostationary feature detector experiments using reconfigurable BEE2. In *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on.* IEEE, 216–219.

[26] Tao Wang, Guangyu Sun, Jiahua Chen, Jian Gong, Haoyang Wu, Xiaoguang Li, Songwu Lu, and Jason Cong. 2014. GRT: a reconfigurable SDR platform with high performance and usability. *ACM SIGARCH Computer Architecture News* 42, 4 (2014), 51–56.

[27] Wei Wang, Yingjie Chen, Zeyu Wang, Jin Zhang, Kaishun Wu, and Qian Zhang. 2015. Changing channel without strings: Coordination-free wideband spectrum adaptation. In *Computer Communications.* 460–468.

[28] Haoyang Wu, Jun Liu, Songwu Lu, Tao Wang, Zengwen Yuan, Chunyi Peng, Zhiwei Li, Zhaowei Tan, Boyan Ding, and Xiaoguang Li. 2017. The Tick Programmable Low-Latency SDR System. In *The International Conference.* 101–113.

[29] H Wu, T Wang, J Chen, et al. 2015. GRT: a high-performance customizable HW/SW open platform for underlying wireless networks. *J Univ Electron Sci Technol China* 44, 1 (2015), 123–128.