EURASIP Journal on
Embedded Systems

RESEARCH **Open Access**

CrossMark

ARM-FPGA-based platform for reconfigurable wireless communication systems using partial reconfiguration

Mohamad-Al-Fadl Rihani[1,2]* , Mohamad Mroue[2], Jean-Christophe Prévotet[1], Fabienne Nouvel[1] and Yasser Mohanna[3]

Abstract

Today, wireless devices generally feature multiple radio access technologies (LTE, WIFI, WIMAX, ...) to handle a rich variety of standards or technologies. These devices should be intelligent and autonomous enough in order to either reach a given level of performance or automatically select the best available wireless technology according to standards availability. On the hardware side, system on chip (SoC) devices integrate processors and field-programmable gate array (FPGA) logic fabrics on the same chip with fast inter-connection. This allows designing software/hardware systems and implementing new techniques and methodologies that greatly improve the performance of communication systems. In these devices, Dynamic partial reconfiguration (DPR) constitutes a well-known technique for reconfiguring only a specific area within the FPGA while other parts continue to operate independently. To evaluate when it is advantageous to perform DPR, adaptive techniques have been proposed. They consist in reconfiguring parts of the system automatically according to specific parameters. In this paper, an intelligent wireless communication system aiming at implementing an adaptive OFDM-based transmitter and performing a vertical handover in heterogeneous networks is presented. An unified physical layer for WIFI-WIMAX networks is also proposed. The system was implemented and tested on a ZedBoard which features a Xilinx Zynq-7000-SoC. The performance of the system is described, and simulation results are presented in order to validate the proposed architecture.

Keywords: FPGA, SoC, Partial reconfiguration, Adaptive technique, Vertical handover algorithm

1 Introduction

In the last decade, wireless communication systems have greatly evolved in terms of mobility, covered range, and throughput. A lot of standards with different specifications, both in the PHY (bandwidth, throughput, and modulation encoding techniques) and MAC layers, have been provided to meet different technical requirements and give access to new services. In this context, it has become very interesting to consider a multi-standards device, capable of implementing several MAC/PHY layers within the same chip. In general, this is performed by implementing several radio access technologies, in parallel, within

the same chip. This solution is obviously very costly and not satisfactory in terms of hardware resources usage. Moreover, only few protocols can be handled in this case.

Today, users are facing a huge diversity of communication networks and protocols that are generally very complex and heterogeneous. This is especially true for mobile devices that have to deal with multiple radio access technologies (Multi-RAT), during their mobility.

Adaptive techniques enable applying modifications in real-time to both PHY or MAC layers of a wireless system. Using these techniques, modifications are performed according to specific conditions and to the state of the communication channel within the overall network. This allows achieving the best performance by adapting and reconfiguring the communication system in real-time without interruption. The operation of switching

*Correspondence: mrihani@insa-rennes.fr
[1]IETR-INSA, 20 Avenue des Buttes de Coesmes, 35708 Rennes, France
[2]Faculty of Engineering III, Lebanese University, Hadath Campus, Beirut, Lebanon
Full list of author information is available at the end of the article

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 2 of 17

between radio access technologies is denoted as vertical handover.

Hardware reconfiguration of wireless communication systems is the most interesting solution for implementing adaptive techniques and vertical handover algorithm (VHA). In this case, some system functions may be modified during runtime, while other functions continue to run without any interruption. Field-programmable gate arrays (FPGAs) constitute the ideal circuits to implement such reconfigurable systems. Such devices are now sufficiently mature to implement very complex systems with a high level of performance. Partial reconfiguration (PR) is one of the interesting features that has been added by FPGAs' vendors to ensure even more flexibility. Furthermore, partial reconfiguration decreases the time needed to reconfigure parts of the circuit and reduces its power consumption by considering only partial bitstreams instead of complete ones.

Another interesting feature in recent FPGA devices is the presence of embedded hard processor cores that are implemented with the hardware FPGA fabric, in the same chip. For example, the ZYNQ SoC devices from Xilinx feature a dual-core ARM Cortex-A9 based Processing System (PS) as well as a programmable logic (PL) within a single device. Compared to their soft-core processor counterparts, these processors offer much more computing power and speed. SoC devices are appropriate for designing joint software and hardware (SW/HW) systems. The existence of such systems in the end devices allows designing high-performance smart communication objects.

In this work, we present an ARM-FPGA-based platform for reconfigurable wireless communication systems that benefits from the partial reconfiguration technique. The baseband functions of the wireless system are implemented within the FPGA fabric, whereas the reconfiguration of the different functional blocks is managed by processes running on the ARM processor. The role of these processes is to take decisions and manage the reconfiguration process of the concerned modules. A custom micro-kernel that is dedicated to partial reconfiguration [1] is used to schedule and share resources among processes.

The main contributions of the paper are as follows:

1. Designing an original FPGA reconfiguration manager based on a custom micro-kernel that concurrently executes several independent tasks.
2. Proposing a vertical handover algorithm based on a scoring system that makes it possible to switch between two standards according to predefined metrics.
3. Quantitative evaluation of the system performance based on a case study.

This paper is organized as follows: Section 2 deals with the related works on reconfigurable radio systems using the partial reconfiguration technique, vertical handover algorithms for heterogeneous (Multi-RAT). Section 3 describes the general system model, i.e., the proposed SW/HW platform for reconfigurable wireless systems. In Section 4, an implementation study of an adaptive reconfigurable OFDM transmitter is provided. In Section 5, VHA in WIFI-WIMAX networks is applied to the proposed system. Section 6 presents the experimental results and demonstrates the feasibility of the proposed approach. Finally, we conclude the paper in Section 7.

2 Background and related works

This section discusses background concepts and related works in the context of reconfigurable radios and adaptive techniques. The use of PR in reconfigurable radios, as well as VHAs for heterogeneous networks, is also described in this section.

2.1 Adaptive techniques

Increasing the performance of wireless communication systems has always been one of the first objectives of designers. Due to the characteristics of wireless channels (path loss, fading, shadowing ...), the multiplicity of standards, frequency allocations, and mobility features provided by wireless devices, the operating environment has become more and more complex to comprehend.

In this context, researchers have proposed adaptive mechanisms to allow wireless systems to adapt waveforms according to the channel properties. Such systems may sense, learn, and decide to reconfigure themselves dynamically. Various low-level parameters such as the received signal strength indicator (RSSI), signal-to-noise ratio (SNR), and bit error rate (BER) have been used to guarantee adaptability. Other high-level parameters, such as resource allocation, quality of service (QoS), and power consumption, have also been investigated to decide and appropriately adapt the system to the environment in real-time. The main benefit of the adaptive techniques consists in maximizing the channel capacity while minimizing the power consumption.

Adaptive modulation is one of the techniques that has been proposed to set up the best constellation scheme according to the channel conditions [2–4]. Adaptive channel encoding and code rate is another adaptive technique that has been proposed to select the best encoder or the code rate according to the channel status, as presented in [5–7].

Other algorithms and methodologies have also been proposed to deal with bandwidth adaptation. In this case, the system is able to adapt the bandwidth according to

Rihani *et al. EURASIP Journal on Embedded Systems*   (2017) 2017:35

Page 3 of 17

the spectrum status. For example, in OFDM systems, fast Fourier transforms (FFT) and sub-carrier mapper modules can be adapted according to parameters provided by the device and network, as presented in [8] and [9]. In [10] and [11], the authors show that it is crucial to adapt the FFT size with respect to power consumption. These two papers describe how power consumption increases as the FFT size increases. These different studies show that many parameters may be taken into account in an adaptive system. In our work, we consider multiple adaptive algorithms that are operating in parallel in order to reconfigure several modules in a wireless system. The SW/HW platform proposed in this paper aims at managing and scheduling multiple processes based on adaptive algorithms to decide when partial reconfiguration should be applied to hardware.

2.2 Reconfigurable radios using PR

Numerous studies that were proposed for reconfigurable wireless communication systems and cognitive radio (CR) have already taken the advantage of the partial reconfiguration technique in FPGAs. In [12], a review of these techniques in wireless communication systems is presented. The authors suggest benefiting from partial reconfiguration to design and implement a Multi Carrier Code Division Multiple Access (MC-CDMA) system which combines both OFDM and CDMA without presenting the technical details of the proposed system.

In [13] and [14], the authors propose M-Phase Shift Modulation (M-PSK) and M-Quadrature Amplitude Modulation (M-QAM) modems implemented in FPGA. These modems are implemented with reconfigurable modulation and demodulation modules based on the partial reconfiguration technique for software-defined radio (SDR) and CR applications. Also, in [15], the author continued his work by proposing a partially reconfigurable OFDM transmitter. The DPR technique is applied by reconfiguring the modulation and encoding modules. In this work, a microblaze soft-core processor implemented in FPGA has been used to manage the PR process. In [16], the authors propose a full reconfigurable OFDM transmitter, in which most of the modules (modulator, encoder, and FTT) of the OFDM are partially reconfigured. However, they do not describe how partial reconfiguration is managed and controlled in real-time. Also, they do not explain under which conditions the system decides to apply the PR process.

In [17], a similar work based on the DPR concept is discussed. It consists in applying DPR on the modulation and encoder modules of the IEEE 802.11g physical layer. Nevertheless, the authors do not describe how they manage the PR process and do not mention the required real-time control.

In [18], the authors describe an adaptive reconfigurable transmitter for OFDM-based cognitive radio. In this transmitter, the modulation module is adapted according to the SNR value. The authors implement a simple configuration controller but do not describe how the bitstream is transferred to the FPGA. Moreover, it is not clear whether the system can run multiple adaptive processes in parallel or not.

The authors in [19] and [20] benefit from the PR technique to propose reconfigurable radio systems. PR management is controlled by a microblaze soft processor implemented in FPGA with an Internal Configuration Access Port (ICAP) controller. In this case, additional resources in the FPGA are used for both the microblaze processor and for the ICAP IP. In our proposed system, we are using the ARM embedded core and the Processor Configuration Access Port (PCAP) interface is used instead of ICAP. In this case, no additional resources in the FPGA are required.

2.3 VHAs for heterogeneous networks

In the context of vertical handover in heterogeneous networks, several works can be mentioned. In [21], the authors proposed an algorithm composed of two steps. In the first step, the handover is triggered according to the data rate requested by the user. Before starting the handover, the system evaluates the speed of the node. If the speed is higher than a certain threshold that does not allow benefiting efficiently from switching to another system, the handover is canceled and considered as unnecessary. Otherwise, if the speed is appropriate, the available networks are compared according to five parameters: the bandwidth, jitter, delay, cost, and bit error rate. Then, the best standard is selected and the vertical handover is performed if necessary.

The authors in [22] propose an adaptive network scanning algorithm in the context of heterogeneous wireless networks. In this paper, the main idea is to prevent unnecessary scanning processes. The power consumption is then reduced. The device speed is the main parameter that the algorithm considers to adapt the scanning period. A speed limit is defined for every wireless standard.

In [23], the authors propose a vertical handover algorithm between LTE and WLAN consisting of three steps. The first step aims to initiate the handover based on RSS (received signal strength). In the second step, a fuzzy logic system compares four parameters provided by the two networks (bandwidth, SNR, battery life, and network load) to decide which network provides the best connectivity. In the third step, a seamless vertical handover is executed.

Based on the signal to interference plus noise ratio (SINR), the authors in [24] propose a vertical handover between WIFI and WIMAX. According to a custom mathematical function, the equivalent SINR value for each

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 4 of 17

network is computed and compared. The handover decision is based on the best equivalent SINR value.

In the context of creating a unified architecture for two standards, interesting studies for WIMAX and WIFI systems are proposed in [25–27]. The similarities and differences between WIMAX and WIFI physical layers are shown in these articles. The studies describe how WIMAX and WIFI physical layers are quite similar in their global architecture with only few differences in specific modules. Note that it can be appropriate to apply the PR technique to switch between these two standards by considering only these specific modules.

In our work, we aim to provide a complete study for using the PR technique on ARM-FPGA systems in the context of reconfigurable radios. This article briefly explains the tasks and modules that are executed on the processor, and how partial reconfiguration is managed. All the proposed tasks run on top of a light and custom micro-kernel dedicated to partial reconfiguration. We will also discuss more about the PR design for reconfigurable architectures and focus on the switching between multiple wireless standards. In addition to evaluating the needed reconfiguration time and reserved resources of the proposed reconfigurable architecture, we will also estimate the additional power consumption due to the reconfiguration process.

3 General system model

The proposed system can be divided into two parts: the first part is running on the ARM processor, whereas the second part is executed in the FPGA logic. The general system design is presented in Fig. 1.

3.1 Overview for ARM-FPGA platform

The ARM-FPGA platform is based on the Zynq-7000 device. This SoC integrates both software (PS) and hardware (PL) parts within the same chip. The PL part, which is equivalent to a classic FPGA fabric, is ideal for implementing high-speed logic arithmetic and data flow subsystems. On the other hand, the PS part supports software algorithms and operating systems. A high-speed connection between both parts is achieved using the Advanced Extensible Interface (AXI) interconnection [28].

Partial reconfiguration can be applied to the FPGA through two interfaces. ICAP is the first interface, in which the controller is an intellectual property (IP) entity implemented in the FPGA. The second one is the PCAP interface, which enables the ARM processor to partially reconfigure the FPGA in real-time. In this case, the PS uses the Device Configuration Interface (DevCfg) integrated on the ZYNQ device. The PCAP interface is driven by a 100-MHz clock and deals with a word width of 32 bits. The theoretical throughput is 3.2 Gbit/s, but due to the speed limit of transferring bit-streams from RAM, the interface speed is limited to 1.2 Gbit/s. In our case, we use the PCAP interface for its better compatibility with software applications and because it does not require additional logic resources in the FPGA.

In addition to providing more flexibility, one of the benefits of partial reconfiguration is the cost reduction. This is achieved by implementing multiple functions within the same region of the FPGA and modifying their functionality on demand. It then allows reducing the amount
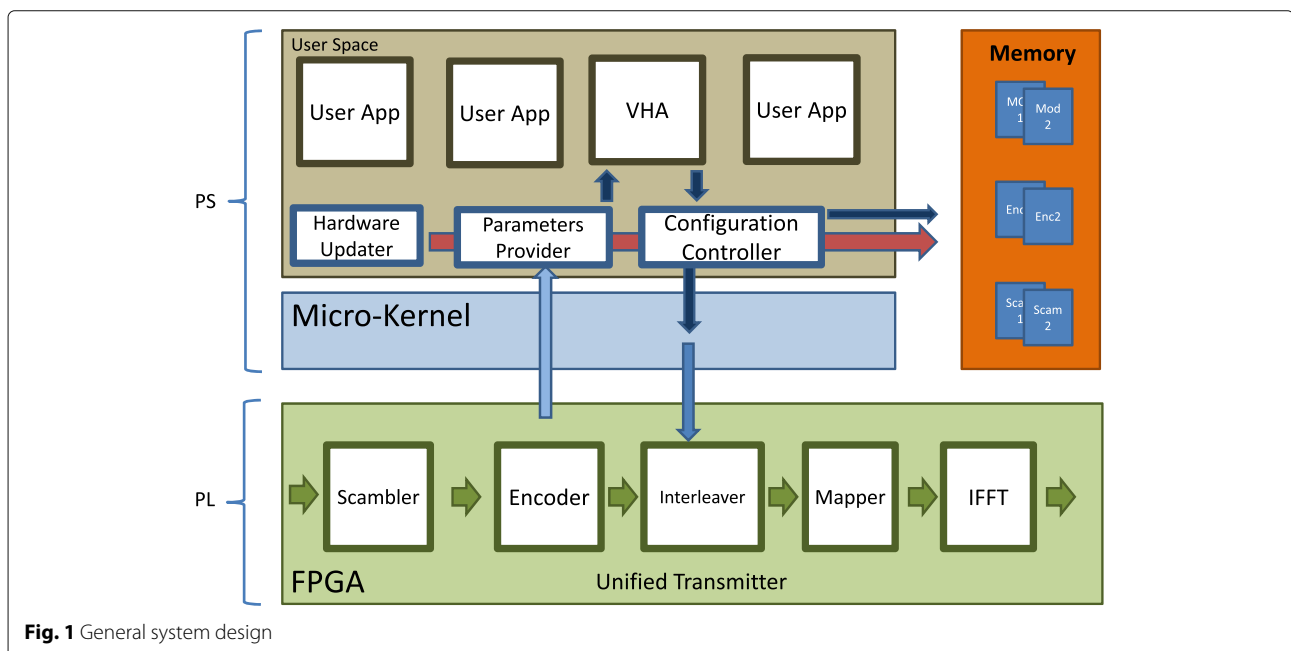

Fig. 1 General system design

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 5 of 17

of required resources when modules are implemented concurrently.

3.2 Processing system modules

The proposed modules running on the ARM are the configuration controller, parameter provider, and hardware updater. All these modules run as tasks in the user-space of a custom micro-kernel.

3.2.1 Micro-kernel

Ker-ONE is a lightweight micro-kernel that provides para-virtualization on ARM embedded systems. In this work, it is seen as a simple kernel that may implement several isolated tasks at user level. Since Ker-ONE is very simple, it only provides fundamental functions such as round-robin scheduling, inter-process communication (IPC), and memory management. It ends up with a small trust computing base (TCB) as described in [1]. In the proposed system, multiple adaptive processes run concurrently on the top of the kernel. Each task may access the DevCfg in order to reconfigure specific hardware blocks in the PL region of the FPGA. Note that all the mechanisms that deal with reconfiguration management are implemented in different isolated tasks: the configuration controller, parameters provider, and hardware updater.

3.2.2 Configuration controller

The configuration controller has two main objectives. The first objective consists in transferring partial bit-streams to the FPGA. For additional security reasons, the partial bit-streams memory locations are only accessible from the configuration controller. This controller eliminates any conflict that may occur when several user processes decide to access the DevCfg at the same time. In this case, it transfers the bit-streams consecutively. The adaptive processes do not manage the bit transfer operations but only send requests and provide the memory addresses to the configuration controller that performs the operation.

The configuration controller module can also configure the concerned hardware modules by updating their input parameters in real-time. This operation is called parametric reconfiguration. In this case, the system reconfigures the hardware either by updating the input parameters of the concerned module or by sending a new version (bit-stream) of the reconfigurable module. As an example, we can consider the IFFT module which can be reconfigured in real-time either by modifying the IFFT size parameter or by applying PR. This enables to combine both methods of reconfiguration (partial and parametric reconfiguration) to create a reconfigurable system. Choosing the configuration method depends on the architecture of the hardware module. Another important task of the configuration controller is to control the data flow during partial reconfiguration. When a system is divided into a set of configurable modules, it is mandatory to make sure that the reconfiguration does not cause any loss of data in the reconfigurable modules during the PR process. An example of the control management is described in Fig. 2 where the second process (*Process3*) is requesting to modify the third module (*Block3*). In this case, the configuration controller suspends all the precedent modules (*Block1* and *Block2*) that produce inputs to the reconfigured module (*Block3*). Note that the reconfiguration time of the concerned module is provided to the controller. This helps the controller to determine the time during which the modules should be suspended. The hardware
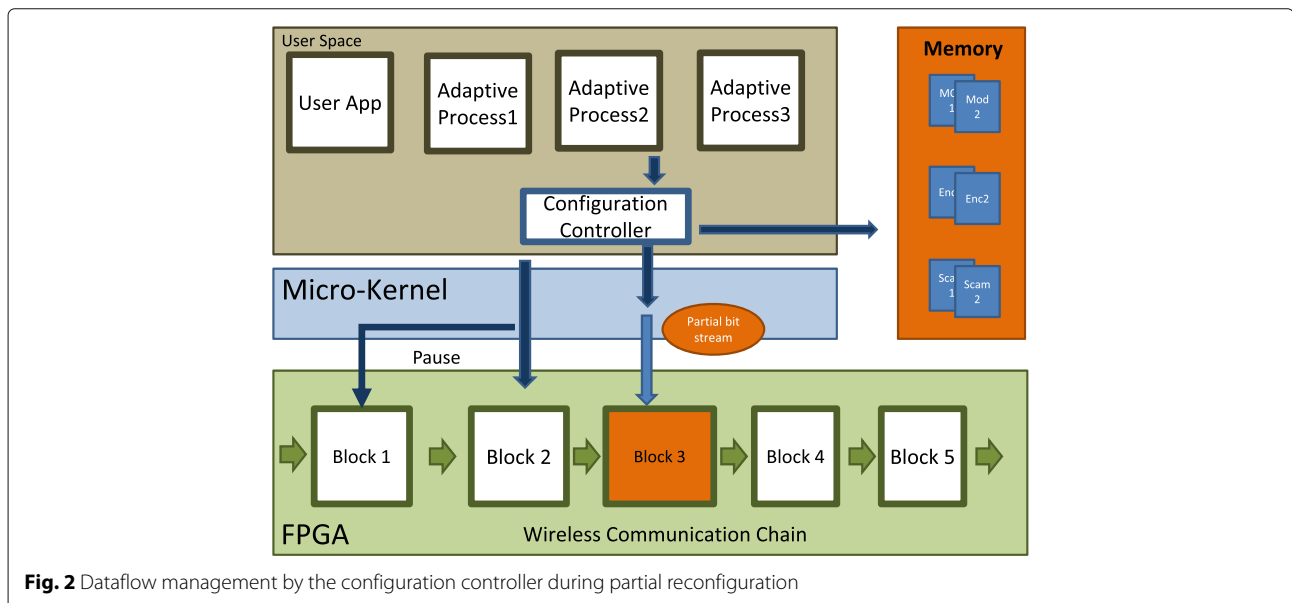

Fig. 2 Dataflow management by the configuration controller during partial reconfiguration

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 6 of 17

blocks data flow is controlled through the general purpose input output (GPIO) ports connecting the PS and PL.

3.2.3 Parameter provider

The parameter provider is another important task running on the micro-kernel. It is considered as an interface for all the parameters of the system. This task provides all the required parameters to the user processes in order to take a decision regarding the configuration of a specific module. The parameters are collected from different layers in the system.

3.2.4 Hardware updater

The third module running on the processor is the hardware updater. In this module, we benefit from one of the advantages of the partial reconfiguration that makes it possible to implement new versions of modules or even update other available modules during runtime. In a partial reconfiguration system design, reconfigurable modules are defined as black boxes with predetermined input and output ports. This means that all the versions of the reconfigurable module must have the same input and output ports.

Many partial bit-streams can be generated but only one is transferred to the FPGA at a time. The other partial bit-streams are transferred to the reconfigurable module on demand. In such designs, we have the possibility to reconfigure any reconfigurable module without reconfiguring all the hardware chain.

The task of the hardware updater is to check for any new update or new version of a reconfigurable modules. If any update is available, a module retrieves the new partial bit-stream in a secure way. The idea is similar to updating or sending patches to a software process, but applied to hardware.

3.3 Reconfigurable chain on programmable logic

On the FPGA, let us assume having a reconfigurable chain for processing wireless standards. The reconfigurable chain may be reconfigured during runtime either by transferring partial bit-streams or by parametric reconfiguration. According to the decision made by the adaptive processes, one or multiple modules are reconfigured to adapt the waveform to fit the new system status. In [29], we have proposed and compared several reconfigurable architectures to switch between multiple wireless standards. In a first approach, we have considered the full chain as an unique reconfigurable block. Switching between standards consists in transferring a single large partial bit-stream. This architecture is called one reconfigurable block architecture (ORBA). The second approach is the multiple reconfigurable blocks architecture (MRBA). In this architecture, multiple blocks of the chain are reconfigured to switch from a standard to another. As

shown in [29], MRBA is better in case of switching between standards with similarities in their physical layer. On the other hand, ORBA is easier and more suitable when wireless standards have many differences in their physical layer.

3.4 Data flow control

Partial reconfiguration process needs time to be achieved which leads to an interruption in the hardware chain. Compared to the parallel implementation design, PR design decreases the overall hardware resources and power consumption. On the other hand, the reconfiguration time must be considered in PR design compared to parallel design one. The additional reconfiguration time is related to the size of the reconfigurable module. The configuration controller, proposed in our design, handles the data flow during the partial reconfiguration process as explained above. FIFOs implemented between the reconfigurable modules buffer the data flow during the PR process without causing an interruption at the upper layers. In case of switching between multiple wireless standards, there are two possible designs as explained before. In MRBA, the system is formed by multiple small reconfigurable blocks. However, in ORBA, the system includes only one large reconfigurable block. In both designs, the reconfiguration time is acceptable since the handover between multiple wireless standards depends also on signaling from the upper layers. To avoid losing any data, a FIFO buffers data from the upper layer until the PR process is achieved.

4 Auto adaptive OFDM transmitter

To study the impact of applying partial reconfiguration in the context of reconfigurable radios, we have implemented an auto-adaptive OFDM transmitter in the proposed system. The OFDM is an advanced multi-carrier modulation technique used in the new generations of fixed and mobile wireless communication systems (ex: Wifi, LTE, ...). This technique divides high-rate data streams into multiple low-rate data streams. The information is then modulated and transmitted over multiple sub-carriers.

4.1 Reconfigurable hardware

The OFDM transmitter is composed of an encoder, a modulator, a mapper, and an IFFT (see Fig. 3). Based on the Xilinx Vivado tool, multiple partial bit-streams have been generated for these modules. The different versions of the reconfigurable modules and the parameters used in the transmitter are given in Table 1. The OFDM transmitter has been tested and implemented on the ZedBoard. The data path operates on the FPGA as follows: A random bit source generator provides random data to the encoder. Then, data are encoded and

Fig. 3 OFDM transmitter base modules
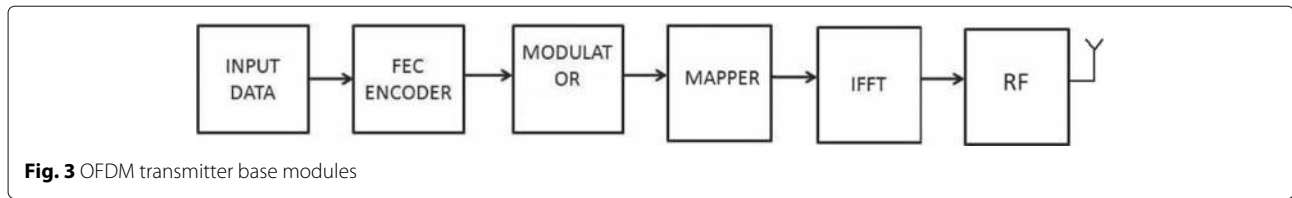
provided to the modulator in which frequency domain symbols are generated. The symbols are mapped into sub-carriers in the IFFT module through the mapper. Finally, ten OFDM symbols with the corresponding cyclic prefix are generated by the IFFT module and concatenated with pilot bits to build a frame. The frame is then ready to be transferred to the Radio Frequency (RF) module.

The reconfigurable regions on the FPGA (Pblocks) are selected using the floor-planning tool. Selecting a Pblock with an amount of resources larger than necessary leads to larger partial bit-streams. Therefore, the selected Pblocks are tuned to the maximum resource utilization that is required by the different versions of the reconfigurable modules. Figure 4 shows the floorplan design of the reconfigurable and static modules of the OFDM transmitter. As required in a PR design, the different versions of the reconfigurable modules should have the same interface (input and output ports).

For example, if a 4-QAM modulator has a 2-bit input whereas a 256-QAM modulator has an 8-bit input, then the reconfigurable modulator will match the largest module with 8-bit input. Therefore, 6 bits will be unused in the data input port of the 4-QAM. A similar concept is applied to all the reconfigurable modules in the chain. Note that these additional bits have no impact since they are likely to disappear in the optimization process of the design flow.

Figure 5 presents the adaptive OFDM transmitter implemented in the proposed platform. The modulator, encoder, and IFFT blocks are reconfigurable modules whereas the other blocks remain static. The partial

bit-streams corresponding to multiple versions of each reconfigurable module are located in memory, and transferred to FPGA on demand by the configuration controller. Note that in some cases, such as for the (I)FFT architectures, the concerned module can be reconfigured by parametric reconfiguration without initiating the DPR process and transferring a bit-stream from memory. This can be applied by simply modifying the input size parameter of the module. This situation is also taken into account in the configuration controller.

4.2 Adaptive processes

The Xilinx SDK tool has been used to implement three adaptive processes running on the ARM processor. The processes are implemented in C code and run in parallel on top of the micro-kernel.

The first (*adaptive_mod*) process is based on the adaptive modulation technique and aims to monitor the SNR value to select the best modulation scheme. The role if this process is to reconfigure the modulation block during run-time. The second process, *adaptive_encoder*, is based on the adaptive channel coding technique. This process reads the BER and SNR values to obtain the wireless channel status in order to select the best coding rate or the efficient encoder type. This second process reconfigures the encoder module of the OFDM chain. The purpose of the third process, *adaptive_IFFT*, is to modify the IFFT size according to the power consumption and energy left in the battery of the system. It is then given the permission to access this metric from a sensor.

Based on offline experiments, we found out that this is the most power consuming module in the hardware design. This process monitors the power consumption in order to select the efficient IFFT size. The three adaptive processes are executed as user processes. In this work, we have considered a realistic scenario in which all the three processes are based on theoretical adaptive techniques and reconfigurable wireless system methodologies as proposed in [2, 7, 10].

The implemented *adaptive_mod* process can be explained as follows. The process is first initialized (for example with a value corresponding to a 4-QAM modulation) then it waits until another SNR value is received. Once it is received, it will be analyzed to check if it belongs to the current range. If the new value is within

Table 1 OFDM transmitter parameters

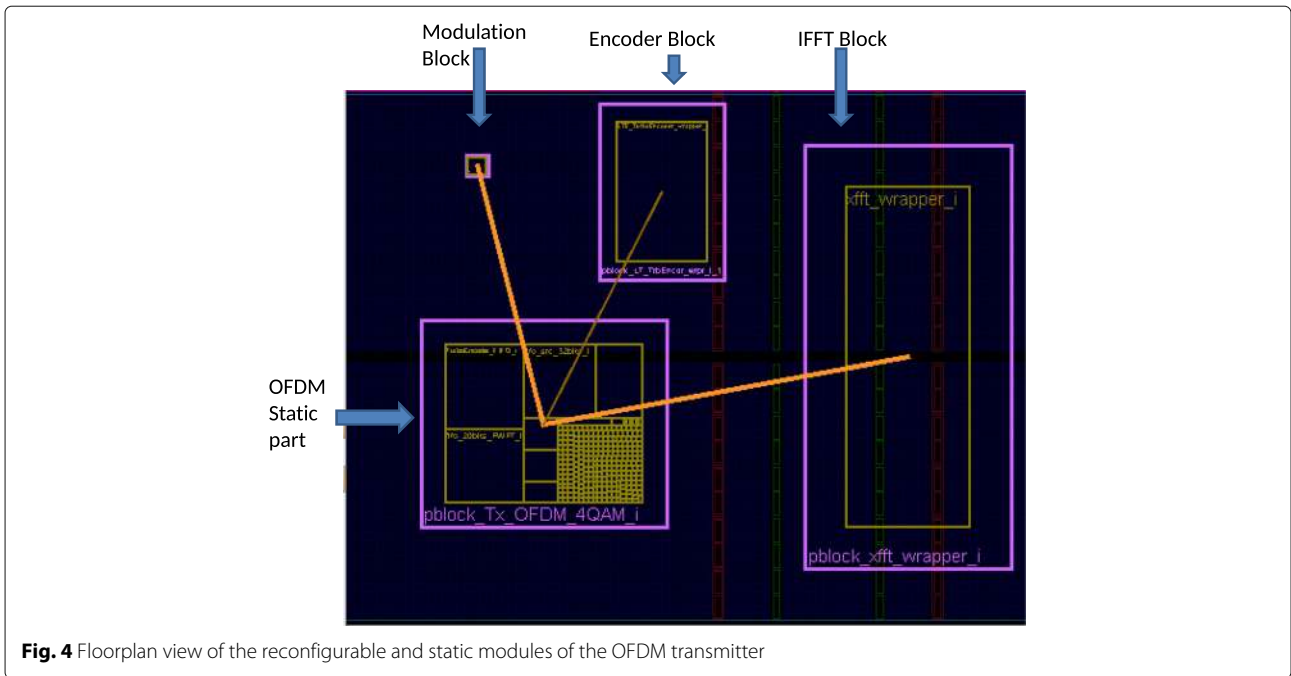| Parameters	Scenario
Modulation	4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM
(I)FFT size	16, 32, 64, 128, 256, 512, 1024
Cyclic prefix length	2, 4, 8, 16, 32, 64, 128
Encoder	Convolutional 1/2, 1/3, 1/4, 1/5, 3GPP LTE Turbo Encoder 1/3
Frame	1 OFDM pilot + 10 OFDM data symbols
FPGA	Zynq-7000 xc7z020
Clock	200 MHZ

Rihani *et al. EURASIP Journal on Embedded Systems*   (2017) 2017:35

Page 8 of 17

Fig. 4 Floorplan view of the reconfigurable and static modules of the OFDM transmitter

this range, the process returns to a waiting state. Otherwise, a new version of the modulation block is selected according to the new range. The partial bit-stream of the selected version is then transferred to the modulator. Finally, the SNR range is updated and the process waits for another SNR value.

The adaptive coding process, *adaptive_encoder*, is similar to the *adaptive_mod* process. It uses the BER parameter to select the best encoder. Similarly, the *adaptive_IFFT* process monitors the power consumption parameter of

the system to select the appropriate IFFT size. This process can monitor a power regulator to read power values from the device through the parameters provider. Note that the *adaptive_mod* process and the *adaptive_encoder* process can read the latest SNR and BER values from the parameters provider at the same time.

5   Switching between WIFI and WiMax

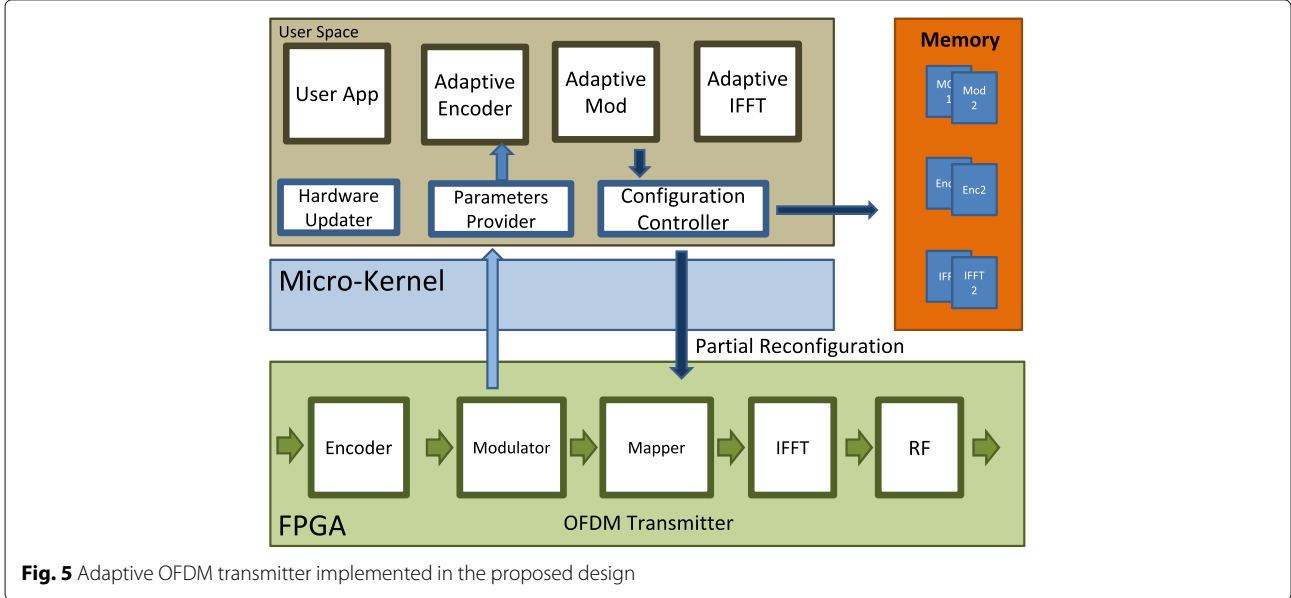In heterogeneous networks, during their mobility, end-node devices may detect multiple wireless standards such

Fig. 5 Adaptive OFDM transmitter implemented in the proposed design

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 9 of 17

as mobile broadband wireless access (3G, 3.5G, and 4G) and Wireless Local Area Network (WLAN). Each radio access technology features different specifications in terms of supported Uplink and DownLink data rate and coverage distance. For example, WIFI 802.11a supports shared data rate up to 54 Mbit/s in 100 m coverage distance. On the other hand, WIMAX supports shared data rate up to 70 Mbit/s and ranges up to 10 km. Both WIFI and WIMAX standards support the Internet Protocol (IP) that is widely used nowadays for voice, data, and video streaming. More details on both standards are given in [30].

In this section, PR was used to switch between multiple wireless standards. In the proposed scenario, an heterogeneous network composed of both WIMAX and WIFI networks is considered. An ARM-FPGA end-node features a custom VHA running on the PS part and an unified partially reconfigurable WIMAX-WIFI physical layer chain implemented in the FPGA.

5.1 Vertical handover algorithm

Vertical handover algorithms read different parameters to learn, study, and then decide whether to perform a vertical handover or keep communicating through the current standard. The key of taking a good decision is driven by the collected and studied parameters from the environment. The known parameters can be divided into four categories:

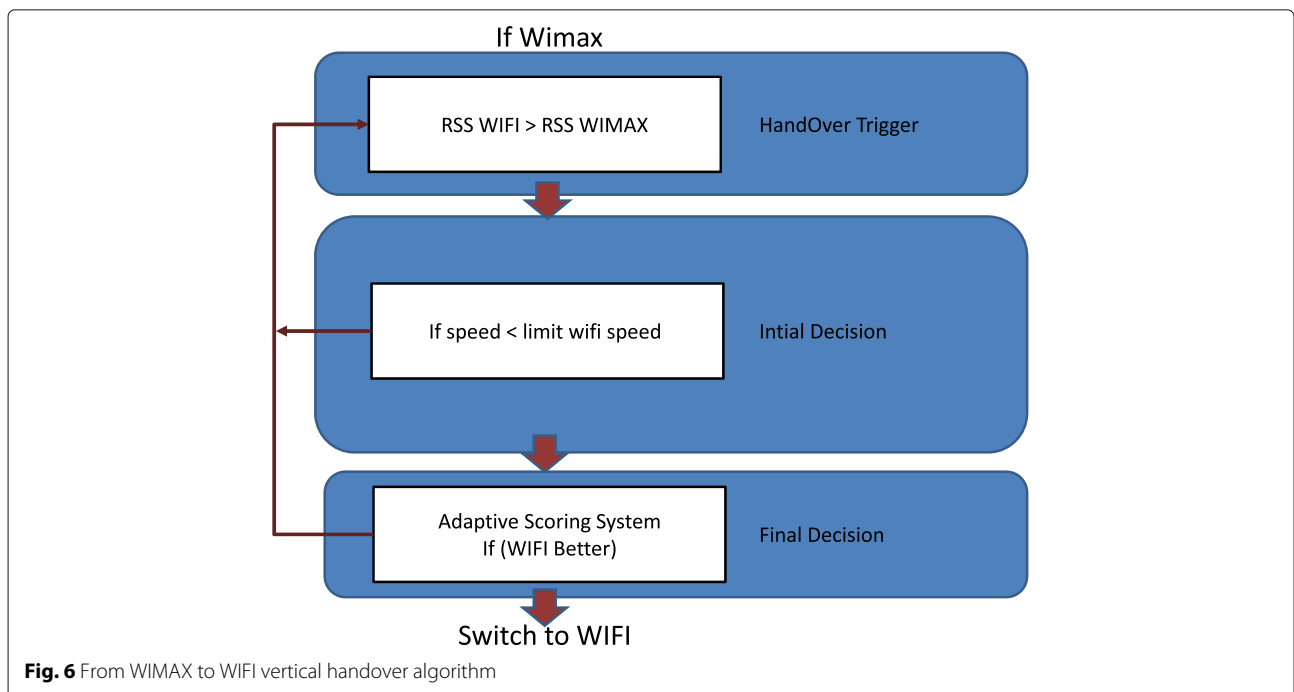1. **Wireless channel parameters:** Reading parameters from the wireless channel to inform the VHA about

the channel status. RSS, SNR, and BER are examples of such parameters.

2. **Network information parameters:** These parameters inform the VHA about the network status. This category includes the number of connected nodes to the access point, data rate, delay, jitter, and cost.

3. **User parameters:** These parameters reflect user's data requirements at the application layer such as required data rate and high priority data.

4. **System parameters:** Such type of parameters provides the algorithm with the device status. These parameters may deal with power consumption, battery level, and device speed.

The proposed VHA for an WIMAX–WIFI network is composed of three stages: an Handover Trigger, an Initial Decision and a Final Decision. The algorithm identifies two execution contexts. In the first context, the system communicates through the WIMAX standard, whereas in the second case, the system is initially operating with WIFI.

The handover algorithm that switches from WIMAX to WIFI is illustrated in Fig. 6. The trigger is initiated as soon as the handover process detects a RSS_WIFI value greater than RSS_WIMAX. The Initial decision state is used to make fast decisions and detect fake handover. Fake handover is detected if the speed of the device is greater than WIFI_max_speed_limit.

If the initial decision state is successfully passed, the system goes into the final decision stage. In this final

Fig. 6 From WIMAX to WIFI vertical handover algorithm

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 10 of 17

stage, an adaptive scoring system reads multiple parameters to decide which network is the best according to the user preferences. If the scoring system decides that WIFI is better than WIMAX, then vertical handover occurs.

The handover algorithm from WIFI to WIMAX is illustrated in Fig. 7. Since the initial state is WIFI, the trigger is initiated when the handover process detects that RSS_WIMAX is greater than RSS_WIFI. The Initial Decision state explores two conditions. The first condition consists in checking if the battery level is less than the battery_low_level_limit. The second condition makes sure that RSS_WIFI is less than RSS_WIFI_limit. These two conditions should be verified; otherwise, the handover is canceled.

If the initial decision state is successfully passed, the system reaches the final decision stage. At this stage, an adaptive scoring system reads multiple parameters to decide which network is suitable according to the user application processes preferences. If the scoring system decides that WIMAX is better than WIFI, then vertical handover occurs.

An example dealing with the operation of the proposed scoring system is shown in Table 2, where each parameter is assigned different points. The scoring system operates as follows: If a standard provides a better behavior regarding a given parameter (ex. SNR, BER), then points related to this concerned parameter are added to the score of the corresponding network. For example, if the data rate parameter is assigned 3 points, and WIFI provides better throughput than WIMAX, then 3 points are added to the

WIFI score. On the other hand, if the delay in WIMAX is less than that in WIFI, 4 points are added to WIMAX. Finally, the network with the highest score is selected.

Adding adaptivity to the scoring system improves the efficiency of the system. In this case, it will be possible to adapt the weight of each parameter in the scoring system according to the user preferences. As shown in Table 3, the points attributed to a given parameter may change according to the type of the running application. For example, when running a voice over IP (VoIP) application, the required data rate, delay, and jitter parameters have more weights than in case of running a browsing application.

The same concept can be applied to the power consumption parameter. In this case, the weight of this parameter may increase as the battery level decreases. Therefore, selecting the best network is not only related to the user applications requirements but also to the power status of the system.

5.2 Unified physical layer for WIFI-WIMAX
Using PR technique, we aim to design a unified reconfigurable physical layer for WIFI and WIMAX standards. In this section, we study the similarities and differences in the physical layer of the two standards. A unified reconfigurable architecture based on PR is then proposed and described in details. The corresponding design flow is also presented.

IEEE defines the PHY layer and MAC layer specifications for both WIFI and WIMAX. The physical layer of these two standards is based on the OFDM modulation
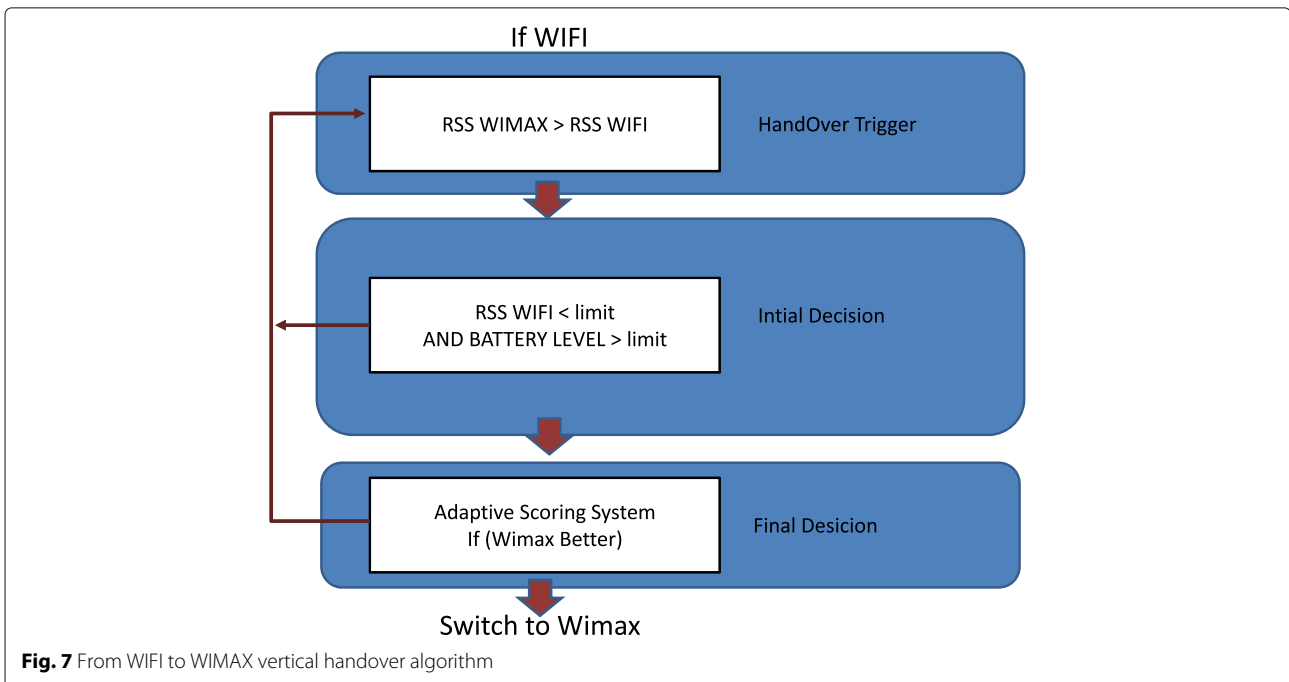

Fig. 7 From WIFI to WIMAX vertical handover algorithm

Rihani *et al. EURASIP Journal on Embedded Systems*   (2017) 2017:35

Page 11 of 17

Table 2 Example of a scoring system

Parameter	WIFI	WIMAX
Data rate (3)	3	–
Delay (4)	–	4
Jitter (3)	–	3
SNR (3)	–	3
BER (3)	–	3
Power consumption (5)	5	–
Total score	8	13

scheme. The main modules are Scrambler/Descrambler, Forward Error Correction (FEC) Encoder/Decoder, Interleaver/De-Interleaver, Mapper/De-Mapper, Pilot Guard Insertion/Removal, IFFT/FFT, and Cyclic Prefix Insertion/Removal. The differences in the transmitter architectures are illustrated in Fig. 8.

As shown in Fig. 8, the physical layer architectures of both standards are quite similar with few differences in the functionality of some modules. For example, WIMAX uses a Reed-Solomon FEC with a convolution encoder whereas WIFI uses the convolution encoder only. The differences in the receiver are similar to those in the transmitter. Based on this, it seems suitable to apply PR with the multiple reconfigurable blocks architecture (or MRBA) method.

Applying PR in the system design using MRBA requires to divide the system into two parts, static and dynamic. The static part consists of shared blocks used in both standards. During the switching process between standards, this part of the system remains unchanged. The dynamic part is composed of modules that can be reconfigured when necessary. Multiple versions of these reconfigurable modules are stored in memory.

To test the system, the choice has been made to reconfigure the Scrambler, Interleaver, FEC, and IFFT blocks in order to perform vertical handover during runtime. The configuration controller reconfigures these modules according to the VHA requests.

5.3   Unified receiver chain with adaptive scanning period

Scanning available wireless networks is important to retrieve the required information and decide which wireless standard to select. To allow implementing a unified receiver chain and scanning multiple standards within a

Table 3 Scoring system example according to different applications

Parameter	VOIP	Video	Browsing	Ideal
Data rate	3	4	2	1
Delay	4	4	2	1
Jitter	4	4	1	1

short delay at the same time, the receiver should support reconfiguring itself rapidly to switch from a standard to another. Similar to the approach proposed in [22], an adaptive and specific scanning period has been provided to sense the available standards. In this case, a unified receiver chain is implemented in the FPGA. At a specific time, the receiver chain is reconfigured to sense the network. The sensing period is given by the adaptive scanning algorithm which directly depends on the speed of the device. After collecting some parameters, the receiver chain is reconfigured back to the initial operating standard. The VHA uses the collected parameters to decide which standard to select.

6   Experimental results

This section is divided into three parts. The first one focuses on the additional power consumption related to PR. The second part presents the results obtained from the adaptive OFDM transmitter scenario whereas the third part presents the results of the WIFI-WiMAX vertical handover scenario. In the last two parts, the results are described in terms of the time needed to reconfigure the dynamic modules, partial bit-streams size, power consumption and reserved resources by each version of the modules. Finally, a Gantt chart is presented to illustrate the behavior of the system according to the input parameters' variations. From the obtained results, we aim to show how much hardware resources are used and evaluate the corresponding power consumption as well as the reconfiguration time.

6.1   Additional power consumption related to PR

To measure the additional power consumption when applying PR on the FPGA, real-time power measurements on the board had been performed during partial bit-stream transfer (see [31]). The power measurements were performed on the PL and PS parts during the PR operation. An increase of 0.125 mW/ms in the average power consumption had been detected on the PS auxiliary circuits when initiating a DPR process. However, no additional power had been noticed on the PL part of the FPGA. These results were obtained by reconfiguring the FPGA through the PCAP interface, from the processor, and not using ICAP that requires internal mechanisms in the PL part.

Using ICAP with MicroBlaze processor IPs increases both the used resources and the power consumption on the FPGA. While using ARM processor and PCAP increases the overall power consumption of the system (SoC) but with no additional resources and power consumption on the FPGA. In [32], the authors measured the additional power consumption when applying partial reconfiguration on Virtex 5 FPGAs. MicroBlaze and ICAP were used to apply partial reconfiguration and results
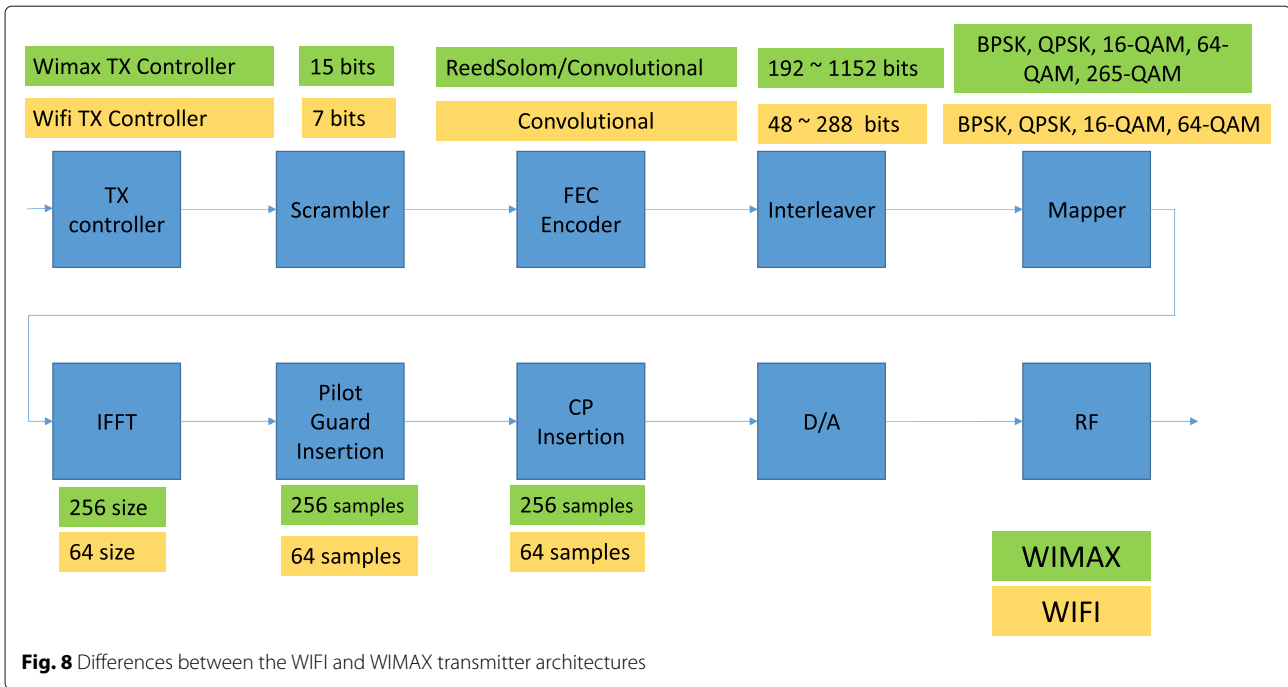
Rihani *et al. EURASIP Journal on Embedded Systems*   (2017) 2017:35

Page 12 of 17

Fig. 8 Differences between the WIFI and WIMAX transmitter architectures

showed that the additional power consumption does not exceed 160 mw, while it was shown that using PCAP the additional power consumption does not exceed 25 mw.

6.2 Adaptive OFDM transmitter

In the adaptive OFDM scenario, we consider three reconfigurable modules: the modulator, encoder and IFFT. The results presented in Table 4 include the partial bit-stream size and the time required to reconfigure a module. As mentioned earlier, the size of the partial bit-stream is related to the size of the reconfigurable module implemented in the FPGA. This size is selected to fit the largest version of the concerned reconfigurable module among all the possible ones.

As noticed in Table 4, the time needed to transfer a bit-stream to hardware using the PCAP interface depends on the size of the partial bit-stream file. In our work, the reconfiguration time is computed by counting all the clock cycles needed to complete the partial bit-stream transfer. Then, the number of clock cycles is multiplied by the clock period (1/Frequency of processor). The partial bit-stream size is related to the floorplan design, as shown in Fig. 4, where the IFFT module occupies more resources

on the FPGA compared to the modulation module. Vivado power and utilization estimation tools have been used to estimate the power consumption and the resources used by the different versions of the reconfigurable modules. Vivado power tools are only used to estimate static power for different versions of each module. These results are then used to compare the static power of the PR-based design to that of the parallel-based one. For both cases, the dynamic power has not been taken into account since it is closely related to the input stimuli provided during timing simulation. These stimuli are similar for both the reconfigurable and parallel architectures. The architecture of the IFFT used in this scenario is the pipeline streaming IFFT (from the Xilinx Intellectual Property (IP) library). This IFFT module allows performing a continuous computation.

In order to analyze the power consumption in the FPGA, Table 5 shows the differences between the IFFT, encoder, and modulation blocks in terms of utilized resources and power consumption. These resources and the on-chip power consumption are almost identical for the different modulation versions. This observation is also true for the convolution encoders. The Turbo encoder clearly consumes much power and occupies more resources on the FPGA than the convolution encoders do. As for the different versions of the IFFT blocks, it is obvious that the power consumption and the utilized resources increase as the IFFT size increases.

Also, it is important to analyze the running process through a Gantt chart as depicted in Fig. 9. As shown in

Table 4 Partial bit-streams size and reconfiguration time

Module	Partial bit stream size (KB)	Reconfiguration time (ms)
IFFT	188	1.48
Encoder	106	0.834
Modulation	30	0.236

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 13 of 17

Table 5 Size and power consumption of the considered blocks

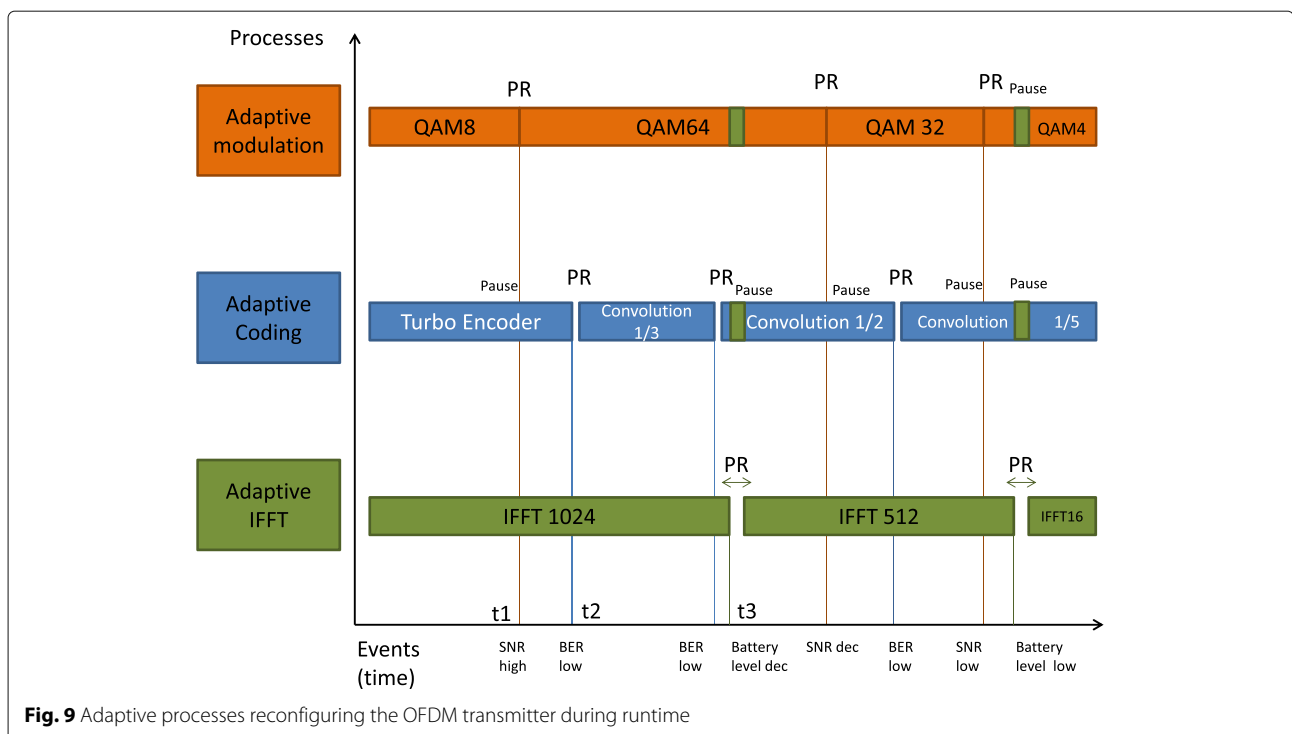| Module	Power (mW)	FPGA resources (LUT-REG-RAM-DSP)
4-QAM	Less than 0.1	15 - 8 - 0 - 0
8-QAM	Less than 0.1	16 - 10 - 0 - 0
16-QAM	Less than 0.1	18 - 15 - 0 - 0
32-QAM	Less than 0.1	20 - 19 - 0 - 0
64-QAM	Less than 0.1	22 - 20 - 0 - 0
Convolution 1/2	Less than 0.1	24 - 29 - 0 - 0
Convolution 1/3	Less than 0.1	25 - 31 - 0 - 0
Convolution 1/4	Less than 0.1	26 - 32 - 0 - 0
Convolution 1/5	Less than 0.1	28 - 35 - 0 - 0
Turbo-Encoder 1/3	2	311 - 555 - 0 - 0
16-point IFFT	9	191 - 1299 - 0.5 - 3
1024-point IFFT	29	2245 - 3554 - 3 - 12

this figure, three adaptive processes run in parallel on the PS part, sense their own parameters and finally detect any change. If necessary, the partial reconfiguration is applied on the appropriate block in order to adapt the system to the new environment. As an example, we consider the scenario illustrated in Fig. 9. During initialization, a default configuration is implemented in the FPGA: an 8-QAM modulator, a turbo-encoder and a 1024-point IFFT. At time $t = t_1$, the *adaptive_modulation* process detects a high SNR. According to the algorithm described for the

adaptive_modulation process, a partial reconfiguration is launched by the configuration controller that implements a 64-QAM in order to increase the throughput. A partial reconfiguration takes place and a new configuration is effective at time $t = t_1 + 0.236$ ms. Meanwhile the encoder block pauses for 0.236 ms.

At time $t = t_2$, the *adaptive_coding* process detects a decrease in the BER. Therefore, a partial bit-stream of the convolution encoder is transferred to the encoder module. The transfer is done at $t = t_2 + 0.834$ ms. Since the encoder is the first module in the chain, the other modules keep on running during the partial reconfiguration operation.

At $t = t_3$, the *adaptive_FFT* process that senses the power regulators on the device detects a high power consumption with a low battery voltage level. Therefore, this process reconfigures the IFFT module and reduces its size. This leads to reducing the power consumption of the system. The operation takes 1.48 ms and all the precedent modules are suspended during the partial bit-stream transfer operation in order to avoid any data loss.

It may be noticed that the time needed to reconfigure the IFFT module is longer than that of the coding and modulation modules. This is due to the difference in the partial bit-streams size. During partial reconfiguration, the precedent modules in the chain are suspended as clearly shown in Fig. 9. The suspension time depends on the size of the transferred partial bit-streams. As noticed, the suspension overhead only affects the precedent modules of the chain.

Fig. 9 Adaptive processes reconfiguring the OFDM transmitter during runtime

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 14 of 17

As a result of applying partial reconfiguration, the FPGA power consumption and used resources are reduced. As an alternative to this technique, all versions of the different modules may be implemented in the FPGA, with additional multiplexers to select the desired module. This typically requires a lot of resources on the FPGA. As a consequence, the static and dynamic power consumption of the chip would be relatively high compared to PR-based systems. Table 6 shows the amount of resources that is used when applying the PR technique compared to the case where PR is not adopted. It may be seen that there is approximately a factor of 4 between the two cases. Moreover, in the PR case, the resource gain makes it possible to use the remaining parts of the FPGA to implement additional processing units. The percentage of the LUTs, REGs and DSPs used on the SoC is 19.3, 15.9, and 25.9%, respectively when all blocks are implemented in parallel. When using PR technique, the percentage of used resources is 4.8, 3.8, and 5.45%, respectively.

6.3 Handover in WIFI-WIMAX networks

The results of the WIFI-WiMAX vertical handover scenario are presented in this section. The reconfigurable modules in the WiMAX-WIFI scenario are the Scrambler, Interleaver, FEC Encoder, and IFFT. The results presented in Table 7 include the partial bit-stream size, and the time required to reconfigure each module.

In Table 7, the obtained results show that the time needed to transfer a bit-stream to the FPGA using the PCAP interface depends on the size of the partial bit-stream. The time needed to reconfigure the chain from WIFI to WiMAX is equal to the sum of the reconfiguration times needed to reconfigure all the concerned modules. The last row of Table 7 shows the time needed to reconfigure the entire chain. In this case, the entire chain is considered as a single PR module. Reconfiguring only the PR modules is more efficient since some modules are similar in both standards and they do not need to be modified.

As a conclusion, reconfiguring the entire chain can be considered as unnecessary and leads to a waste of time and power. Therefore, it is advantageous to consider reconfiguring the concerned modules separately.

Vivado power and utilization estimation tools had been used to estimate the power consumption and the reserved

Table 6 Comparison of hardware resource usage

Ressources	LUTs	REGs	DSPs
With partial reconfiguration	2578	4129	12
Without partial reconfiguration	10,320	16,938	57
Available on ZYNQ	53,200	106,460	220

Table 7 Partial bit streams size and reconfiguration time

Module	Partial bit stream size (KB)	Reconfiguration time (ms)
Scrambler	30	0.236
Interleaver	30	0.236
IFFT	180	1.41
Encoder	30	0.236
Entire chain	317	4.45

resources of the different versions of the reconfigurable modules.

Table 8 shows the resources utilization and the estimated power consumption of each version of the reconfigurable blocks. As a result of applying the partial reconfiguration, the power consumption and the used resources on FPGA are reduced. As an alternative, both WIFI and WIMAX standards would be implemented in parallel. This typically requires a lot of resources on FPGA. As a consequence, additional static and dynamic power would be consumed in the chip.

Table 9 shows the amount of used resources in two cases: first, when a unified chain is implemented for both standards, and second, when they are implemented in parallel. For the transmitter, the results show that the used resources are reduced approximately by a factor of 1.7, when the partially reconfigured unified chain is used. The percentage of used resources when both standards are implemented is 6.9, 5.6, and 6.81% for LUTS, REGS, and DSPs, respectively; on the other hand, the percentage is 4.18, 3.17, and 4.09% when an unified chain is used for both standards.

Figure 10 illustrates a scenario that shows the state of the reconfigurable modules implemented on FPGA when VHA is applied. The chart is drawn according to the measured reconfiguration times and reflects how the unified chain switches from one standard to another. As shown in the figure, the modules are reconfigured sequentially in an

Table 8 Size and power consumption of considered blocks

Module	Power (mW)	FPGA resources(LUT-REG-RAM-DSP)
Scrambler (WIFI)	Less than 0.1	10 - 19 - 0 - 0
Scrambler (WiMAX)	Less than 0.1	12 - 20 - 0 - 0
Interleaver (WIFI)	Less than 0.1	68 - 95 - 0 - 0
Interleaver (WiMAX)	Less than 0.1	75 - 100 - 0 - 0
FEC Encoder (WIFI)	Less than 0.1	30 - 40 - 0 - 0
FEC Encoder (WiMAX)	0.1	129 - 157 - 1 - 0
64-point IFFT (WIFI)	14	1185 - 1965 - 0.5 - 6
256-point IFFT (WiMAX)	20	1676 - 2690 - 1 - 9

Rihani *et al. EURASIP Journal on Embedded Systems*   (2017) 2017:35

Page 15 of 17

Table 9 Comparison of hardware resources usage

	LUTs	REGs	DSPs
Unified chain for both standards	2229	3376	9
Both standards implemented	3967	6027	15
Available on ZYNQ	53,200	106,460	220

order that is related to their position in the chain. The time needed to achieve switching from one standard to another is the sum of the reconfiguration times required by each module.

As noticed, when a module is reconfigured, the data flow is paused until the subsequent modules are also reconfigured. As mentioned earlier, the time needed to reconfigure a block is related to its size. This is illustrated in Fig. 10 in which the time needed to reconfigure the IFFT block is greater than that required by the other ones. The switching from WIFI to WIMAX is achieved when the last module (for ex. IFFT) is reconfigured successfully. Finally, data are transferred through the WIMAX standard until another VHA occurs.

7 Conclusions

In this paper, an ARM-FPGA-based system was proposed for self-reconfigurable wireless communication systems. The partial reconfiguration technique that is available in the recent FPGA devices has been adopted to reconfigure wireless system modules in real-time. The proposed HW/SW platform is based on a custom micro-kernel that has been developed in our laboratory. Its role consists in managing and controlling the partial reconfiguration process. The proposed architecture enables implementing adaptive wireless systems with custom algorithms to manage and perform switching between multiple wireless standards in heterogeneous networks. Two different cases were considered in this study. First, an adaptive OFDM-based transmitter was implemented and described. Second, an intelligent system aiming at performing a vertical handover in heterogeneous networks was designed. In this paper, we considered two use-cases standards: WIFI and WIMAX. The algorithms implementing the PR process were presented and described for both cases. These algorithms run on an ARM processor and aim at getting knowledge of the environment conditions. This is performed by requesting parameters from the parameters provider service of the micro-kernel. In our work, partial reconfiguration in the FPGA was implemented through the PCAP interface. The sizes of the partial bit-streams were optimized in order to minimize the time overhead caused by PR. Moreover, to avoid data loss when performing a vertical handover or during PR, the data flow was controlled conveniently by the configuration controller. This was made possible by temporarily suspending some blocks of the wireless communication chain. The obtained results show that implementing the PR technique reduces the global on-chip power consumption and saves hardware resources in the FPGA. This study gives an important information about the impact of adopting the PR technique in the context of reconfigurable wireless communication systems.
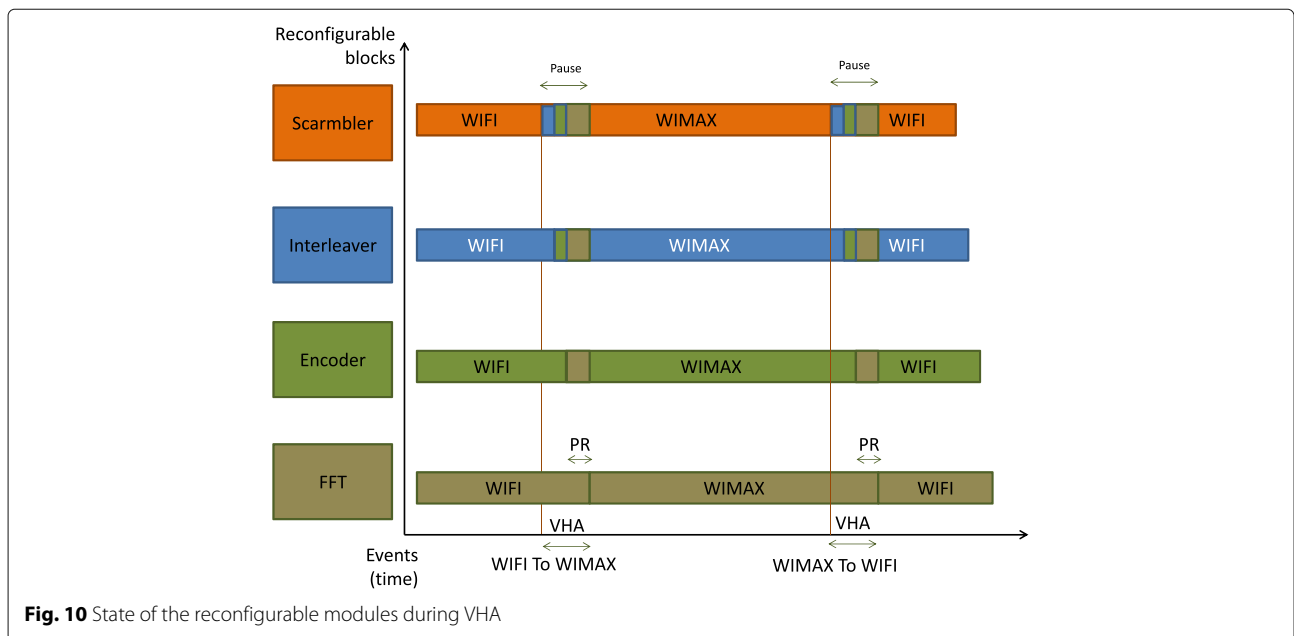

Fig. 10 State of the reconfigurable modules during VHA

Rihani *et al. EURASIP Journal on Embedded Systems*   (2017) 2017:35

Page 16 of 17

Authors' contributions
MR, MM, JCP, FN, and YM jointly developed the theoretical and practical solutions described in this paper, analyzed the results, and evolved the proposed system model. MR took care of all the implementations and collected all the simulation and measurement results. The five authors actively participated in proposing the techniques presented in Sections 3, 4, and 5. Finally, the five authors contributed in similar amounts to the writing and reviewing of the manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]IETR-INSA, 20 Avenue des Buttes de Coesmes, 35708 Rennes, France. [2]Faculty of Engineering III, Lebanese University, Hadath Campus, Beirut, Lebanon. [3]Faculty of Science I, Lebanese University, Hadath Campus, Beirut, Lebanon.

References

1. T Xia, J-C Prévotet, F Nouvel, Microkernel dedicated for dynamic partial reconfiguration on arm-fpga platform. SIGBED Rev. **11**(4), 31–36 (2015). doi:10.1145/2724942.2724947
2. J Faezah, K Sabira, Adaptive modulation for OFDM systems. Int. J. Commun. Netw. Inf. Secur. (IJCNIS). **1**(2), (2009)
3. T Keller, L Hanzo, Adaptive modulation techniques for duplex OFDM transmission. IEEE Trans. Veh. Technol. **49**(5), 1893–1906 (2000). doi:10.1109/25.892592
4. M Marutham, L Nandita, in *2015 International Conference on Communications and Signal Processing (ICCSP)*. Performance improvement of ofdm-based cognitive radio system, (2015), pp. 1623–1627. doi:10.1109/ICCSP.2015.7322793
5. A Sharif, VM Potdar, RF Ahmad, in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*. Adaptive channel coding and modulation scheme selection for achieving high throughput in wireless networks, (2010), pp. 200–207. doi:10.1109/WAINA.2010.197
6. J Liang, R Tessier, D Goeckel, in *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*. A dynamically-reconfigurable, power-efficient turbo decoder, (2004), pp. 91–100. doi:10.1109/FCCM.2004.3
7. C Desset, A Fort, in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring*. Selection of channel coding for low-power wireless systems, vol. 3, (2003), pp. 1920–19243. doi:10.1109/VETECS.2003.1207158
8. Q Zhang, ABJ Kokkeler, GJM Smit, in *2007 IEEE International Conference on Communications*. An efficient fft for ofdm based cognitive radio on a reconfigurable architecture, (2007), pp. 6522–6526. doi:10.1109/ICC.2007.1079
9. ML Ferreira, A Barahimi, JC Ferreira, *Reconfigurable FPGA-based FFT processor for cognitive radio applications*. (V Bonato, C Bouganis, M Gorgon, eds.) (Springer, Cham, 2016), pp. 223–232
10. Y Zhao, AT Erdogan, T Arslan, in *2005 IEEE International Symposium on Circuits and Systems*. A novel low-power reconfigurable fft processor, (2005), pp. 41–441. doi:10.1109/ISCAS.2005.1464519
11. Q Lu, X Wang, J Niu, in *2009 Asia Pacific Conference on Postgraduate Research in Microelectronics Electronics (PrimeAsia)*. A low-power variable-length FFT processor base on radix-24 algorithm, (2009), pp. 129–132. doi:10.1109/PRIMEASIA.2009.5397429
12. AK Nahar, Y Rahayu, Design and implementation of MC-CDMA wireless communication system using partial reconfiguration in FPGA—a review. Int. J. Advancements Commun. Technol. IJACT. **2**(2015), 42–46 (2015)
13. KAA Kumar, in *2012 Annual IEEE India Conference (INDICON)*. FPGA implementation of PSK modems using partial re-configuration for SDR and CR applications, (2012), pp. 205–209. doi:10.1109/INDCON.2012.6420616
14. KAA Kumar, in *2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy*. FPGA implementation of QAM modems using PR for reconfigurable wireless radios, (2013), pp. 1–6. doi:10.1109/AICERA-ICMiCR.2013.6575999
15. KAA Kumar, in *2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy*. FPGA implementation of QAM modems using pr for reconfigurable wireless radios, (2013), pp. 1–6. doi:10.1109/AICERA-ICMiCR.2013.6575999
16. KAA Kumar, in *2014 3rd International Conference on Eco-friendly Computing and Communication Systems*. An ofdm transmitter implementation using cordic based partially reconfigurable ifft module, (2014), pp. 266–270. doi:10.1109/Eco-friendly.2014.61
17. S Sivanantham, R Adarsh, S Bhargav, KJ Naidu, Partial reconfigurable implementation of IEEE802. 11g OFDM. Indian J. Sci. Technol. **7**(4), 63–70 (2014)
18. C Vennila, K Suresh, R Rather, G Lakshminarayanan, S-B Ko, in *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. Dynamic partial reconfigurable adaptive transceiver for OFDM based cognitive radio, (2013), pp. 1–4. doi:10.1109/CCECE.2013.6567749
19. JP Delahaye, J Palicot, C Moy, P Leray, in *2007 16th IST Mobile and Wireless Communications Summit*. Partial reconfiguration of FPGAS for dynamical reconfiguration of a software radio platform, (2007), pp. 1–5. doi:10.1109/ISTMWC.2007.4299250
20. J Delorme, J Martin, A Nafkha, C Moy, F Clermidy, P Leray, J Palicot, in *2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*. A fpga partial reconfiguration design approach for cognitive radio based on noc architecture, (2008), pp. 355–358. doi:10.1109/NEWCAS.2008.4606394
21. M Khan, C Jung, PC Uzoh, C Zhenbo, J Kim, Y Yoon, A Nadeem, K Han, in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*. Enabling vertical handover management based on decision making in heterogeneous wireless networks, (2015), pp. 952–957. doi:10.1109/IWCMC.2015.7289211
22. I Joe, S Hong, in *Proceedings of the 4th International Conference on Ubiquitous Information Technologies Applications*. An adaptive network scanning algorithm in hybrid wireless networks, (2009), pp. 1–4. doi:10.1109/ICUT.2009.5405738
23. A Aziz, S Rizvi, NM Saad, in *2010 International Conference on Intelligent and Advanced Systems*. Fuzzy logic based vertical handover algorithm between LTE and WLAN, (2010), pp. 1–4. doi:10.1109/ICIAS.2010.5716261
24. A Bathich, M Baba, M Ibrahim, in *2012 IEEE Symposium on Computers Informatics (ISCI)*. IEEE 802.21 based vertical handover in WiFi and WiMAX networks, (2012), pp. 140–144. doi:10.1109/ISCI.2012.6222682
25. MC Ng, M Vijayaraghavan, N Dave, A Arvind, G Raghavan, J Hicks, in *2007 5th IEEE/ACM International Conference on Formal Methods and Models for Codesign (MEMOCODE 2007)*. From wifi to wimax: Techniques for high-level ip reuse across different ofdm protocols, (2007), pp. 71–80. doi:10.1109/MEMCOD.2007.371247
26. B Darji, B Pancholi, Implementation of unified architecture of 802.11 a and 802.16 a phy layers using verilog hdl (r & d). Organization. **2**(4), 573–576 (2012)
27. D Raval, N Ranpura, E Mehul, Z Saiyed, Simulation of unified architecture of IEEE 802.11 a and 802.16 a PHY layers using MATLAB. Int. J. Comput. Sci. Emerg. Tech-nologies (E-ISSN: 2044-6004). **272**, (2011)
28. LH Crockett, RA Elliot, MA Enderwitz, RW Stewart, *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. (Strathclyde Academic Media, 2014)
29. M-A-F Rihani, M Mroue, J-C Prévotet, F Nouvel, Y Mohanna, in *The Third International Conference on Electrical and Electronic Engineering, Telecommunication Engineering and Mechatronics (EEETEM2017)*. Applying partial reconfiguration technique on arm-fpga systems in context of vertical handover in wireless heterogeneous networks, (2017)

Rihani *et al. EURASIP Journal on Embedded Systems* (2017) 2017:35

Page 17 of 17

30. L Korowajczuk, *LTE, WiMAX and WLAN Network Design, Optimization and Performance Analysis*. (Wiley, 2011)
31. MA Rihani, F Nouvel, JC Prévotet, M Mroue, J Lorandel, Y Mohanna, in *2016 International Symposium on Wireless Communication Systems (ISWCS)*. Dynamic and partial reconfiguration power consumption runtime measurements analysis for zynq soc devices, (2016), pp. 592–596. doi:10.1109/ISWCS.2016.7600973
32. A Nafkha, Y Louet, in *2016 International Symposium on Wireless Communication Systems (ISWCS)*. Accurate measurement of power consumption overhead during FPGA dynamic partial reconfiguration, (2016), pp. 586–591. doi:10.1109/ISWCS.2016.7600972