Applying Partial Reconfiguration Technique on ARM-FPGA Systems in Context of Vertical Handover in Wireless Heterogeneous Networks

Mohamad-Al-Fadl Rihani\textsuperscript{1,2}, Mohamad Mroue\textsuperscript{2}, Jean-Christophe Prévotet\textsuperscript{1}, Fabienne Nouvel\textsuperscript{1}, and Yasser Mohanna\textsuperscript{3}

\textsuperscript{1} IETR-INSA, Rennes France  
\textsuperscript{2} Lebanese University, Faculty of Engineering III, Beirut, Lebanon  
\textsuperscript{3} Lebanese University, Faculty of Science I, Beirut, Lebanon

Abstract—Nowadays wireless communication networks consist of multiple radio access technologies (LTE, WIFI, WIMAX ... ). Thus, end-nodes may use different radio technologies according to their availability and location. End-nodes should be intelligent and autonomous enough in order to select automatically the best available wireless technology and switch to it seamlessly. The System on Chip (SoC) technique integrates processors and FPGA logic fabrics on the same chip with fast interconnection between them. This feature provides the ability to design Software/Hardware systems with optimized performance. The partial reconfiguration (PR) feature of FPGAs has added more flexibility to these high performance devices. In this work, we propose an ARM-FPGA platform to perform vertical handover between multiple wireless systems using PR technique. The proposed system consists of software and hardware parts. The software part runs on an ARM processor. It has the task to read specific parameters, decide, and control the partial reconfiguration process that applies to the reconfigurable blocks on the FPGA. Also, the software part has the ability to update reconfigurable modules by transferring partial bit-streams on the network. Regarding the hardware part, two reconfigurable architectures for a unified chain are proposed and compared. The evaluation results for both reconfigurable architectures have been collected from a unified WIFI-WIMAX chain. The system is implemented on a ZedBoard featuring a Zynq SoC device.

Index Terms—ARM-FPGA systems, Partial Reconfiguration, Reconfigurable Architectures, Vertical Handover, Heterogeneous Networks, SoC.

I. INTRODUCTION

Today, wireless communication networks deal with multiple wireless standards (LTE- WLAN- WIMAX ...) and it can be stated that these various standards will continue evolving in the near future. Another key point is that the number of these standards will also completely explode, due to the IoT advances that tend to provide new communication schemes among very heterogeneous, mobile and fixed, objects and networks. In these networks, during the end-nodes mobility, multiple radio access technologies (Multi-RAT) will then be available. To achieve an efficient communication with best quality and performance, end-nodes devices should be able to select the best available wireless standard. The operation of switching between two standards is called vertical handover.

From a technological point of view, it can be noticed that wireless communications systems are more and more powerful. Most of them are implemented in System on Chip (SoC) devices. Today, these devices integrate Field Programmable Gate Array (FPGA) and processors on the same chip, with shared memory and fast interconnection. In our study, we have adopted the ZYNQ-7000 from Xilinx, which consists of a dual–core ARM Cortex A9 processor and an Artix-7 or Kintex-7 FPGA. A complete Software/Hardware (SW/HW) application can be implemented on such chips. In this case, the system consists of a software part running on the processor connected to hardware accelerators running on the FPGA. FPGAs are considered as high performance reconfigurable parallel hardware architectures. More reconfiguration flexibility has been added to these devices by implementing the partial reconfiguration (PR) technique. This interesting feature allows to reconfigure a part of the system implemented on the FPGA while the remaining parts are still running in real-time. The PR technique allows designers to reduce the resources used on the FPGA. It also diminishes the time and power consumption needed to reconfigure parts of the system. ZYNQ devices, from Xilinx, provide the Port configuration Access Port (PCAP) to reconfigure the FPGA from the Processor. On the boot stage, partial bit-streams are loaded to the memory (DRAM). Then, during run-time, partial bit-streams can be transferred to the FPGA on demand. This provides the ability to update the partial bit-streams accordingly. The mechanism for updating the partial bit-streams is discussed in the next sections.

In [1], we previously proposed an ARM-FPGA platform for auto-reconfigurable adaptive wireless systems. Adaptive algorithms processes run on the top of a custom micro-kernel dedicated for the partial reconfiguration management. The system is developed to propose a platform that enables switching between wireless standards, based on vertical handover algorithms. The proposed system has a software part which consists of vertical handover algorithms, a parameter provider, a configuration controller and a reconfigurable modules up-dater. The hardware part consists of a reconfigurable unified
The Society of Digital Information and Wireless Communications, 2018 ISSN: 2225-658X (Online); ISSN 2412-6551 (Print)

The current one. whether to switch to another system or to keep operating with from the available communication systems, learn, and decide

Algorithm (VHA). The role of the VHA is to read parameters management [9]. Three modules form the main architecture a light Micro-Kernel dedicated for the partial reconfiguration

A. Software Part

The software part of the system runs in the user space of a light Micro-Kernel dedicated for the partial reconfiguration management [9]. Three modules form the main architecture of the software part in addition to the Vertical Handover Algorithm (VHA). The role of the VHA is to read parameters from the available communication systems, learn, and decide whether to switch to another system or to keep operating with the current one.

This paper is organized as follows: Section II deals with the related works for unified chain designs and reconfigurable architectures using partial reconfiguration. Section III describes the general system model. Section IV explains the difference between the two proposed reconfigurable architectures. Section V presents the application of the two reconfigurable architectures on WIFI-WiMax reconfigurable unified chain. Finally, we conclude in section VI.

II. RELATED WORK

The partial reconfiguration technique has been adopted in many researches and works to propose new systems in the domain of reconfigurable and cognitive radios. In [2], the authors propose a full reconfigurable OFDM transmitter, in which most of the blocks (modulator, encoder, and FFT) of the OFDM are partially reconfigured using the PR technique. In [3], a similar work based on the DPR concept is discussed. It consists in applying DPR on the modulation and encoder blocks of the IEEE 802.11g physical layer. In [4], the authors describe an adaptive reconfigurable transmitter for OFDM-based cognitive radio. The authors in [5] benefit from the PR technique to propose a reconfigurable radio system. In these works, the authors did not mention the idea of creating a unified architecture for a multi-standard system.

In the context of creating a unified chain for a multi-standard wireless system, interesting studies for WIMAX and WIFI systems are proposed in [6], [7] and [8]. The similarities and differences between WIMAX and WIFI physical layers are shown in these articles, but the authors did not use the partial reconfiguration technique to reconfigure the unified chain.

In our system, we will use the partial reconfiguration technique to propose and compare different reconfigurable and unified architectures.

III. SYSTEM DESIGN

This section describes the proposed system design. The overall system is depicted in Fig.1. The system is divided into two parts: software and hardware.

A. Software Part

The software part of the system runs in the user space of a light Micro-Kernel dedicated for the partial reconfiguration management [9]. Three modules form the main architecture of the software part in addition to the Vertical Handover Algorithm (VHA). The role of the VHA is to read parameters from the available communication systems, learn, and decide whether to switch to another system or to keep operating with the current one.

The Parameter Provider has the task to provide the VHA with all the available parameters that are necessary to make a correct decision. During run-time, the Parameter Provider module collects the values of the parameters from different layers. For example, channel status parameters like the Signal to Noise Ratio (SNR), Received Signal Strength (RSS), and Bit Error Rate (BER) are collected from the hardware part on the FPGA. User preference parameters (throughput, real time services, low cost communication, low power communication, high security connection, ...) come from processes running in the user space. Also, the Parameter Provider reads parameters from the hardware and software parts such as the power consumption, battery level, speed of the system as well as the location information.

The VHA process manages the requests to switch from a standard into another. The configuration controller manages and controls the partial reconfiguration of the blocks in the FPGA. In [1], it is explained how the configuration controller works in details. The processor uses the Device Configuration Interface (DevCfg) driver to reconfigure the FPGA through the PCAP interface. The PCAP 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 bottleneck of transferring bit-streams from RAM, the interface speed is limited to 1.2 Gbit/s. The Configuration Controller manages the data flow during the reconfiguration process.

The third module belonging to the software part is the Hardware Updater. In this module, we benefit from one of the advantages of the PR technique. With PR, it is possible to implement new versions of reconfigurable modules or even update other available modules at runtime. In a partial reconfiguration system design, reconfigurable modules are defined as black boxes with fixed input and output ports. This means that all the versions of a 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 design, we have the possibility to update any partial bit-stream without reconfiguring all the hardware chain. The task of the hardware updater is to check for any new update or new version for the reconfigurable modules as shown in Fig.2. If any update is available, the module retrieves the new partial-bitstream in a secure way. The idea is similar to updating or sending patches to a software process, but now it is being applied on the hardware modules.

The reconfigurable architecture is divided into two parts: software and hardware. The software part can be summarized as follows: The parameter provider provides the VHA with the needed parameters from different layers in the system. The VHA analyzes the values of the parameters and makes a decision regarding a possible and useful handover. Accordingly, a request to switch to another system is sent from the VHA to the configuration controller. Then, the Configuration Controller manages the transfer of the partial bit-streams to the reconfigurable modules on the FPGA through the PCAP interface. Finally, the partial bit-streams are updated from the network through the Hardware Updater.

B. Hardware Part

The hardware part of the system consists of physical layers of communication systems implemented on the FPGA. Using partial reconfiguration, we aim to implement a single unified reconfigurable chain for different wireless standards. As an example, Fig.3 depicts the switching between two wireless standards on the FPGA. In state1 the WiMax blocks are running while the partial bit-streams of the WIFI blocks are stored in memory. When VHA decides to switch to WIFI (state2), partial bit-streams of WIFI are transferred to replace the WiMax blocks. The communication system is supposed to run in parallel with other hardware accelerators on the FPGA. When partial reconfiguration is applied, the other accelerators continue running.

The unified chain for multiple wireless standards can be implemented in two possible reconfigurable architectures. A comparison between these two architectures is discussed in the next section.

IV. PROPOSED RECONFIGURABLE ARCHITECTURES

In this section two reconfigurable architectures using PR are proposed and compared in terms of reconfiguration time, design complexity, and functionality. In the first architecture, the entire communication chain is considered as one reconfigurable module. In this case, each wireless standard owns only one partial-bitstream. Fig.4 provides an example related to the switching process. For n systems, n partial bitstreams are stored in memory. At runtime, only one partial-bitstream is transferred to the reconfigurable block. When the system decides to switch to another configuration, the desired partial bit-stream is transferred to replace the existing one. This reconfigurable architecture is denoted as One Reconfigurable Block Architecture (ORBA).
Both reconfigurable schemes are compared to define the characteristics of each architecture. The advantages and disadvantages of each architecture can determine which architecture is better to use in a given application.

In terms of design complexity, ORBA is considered as easier to design than MRBA. In MRBA, studying the similarities and differences of the blocks of \( n \) wireless systems is needed. This study is useful to determine which ones are reconfigurable and which ones are static. The static blocks are considered as common and thus can be used in all the systems without the need of reconfiguration. It is not the case in ORBA for which all the blocks of a wireless system are grouped to form one reconfigurable block with an interface to the upper layer. In this case, it is easier to implement the system using partial reconfiguration. In ORBA, the possibility to have static modules in the chain (if it exists) is discarded.

Both architectures have similar resources utilization in the FPGA. In MRBA, the reconfiguration time consists of the sum of the reconfiguration times of the concerned reconfigurable blocks. The time needed to transfer a partial bit-stream to the FPGA is directly related to its size. The size of the reconfigurable blocks varies according to their complexity. In ORBA, only one partial bit-stream is transferred in order to switch between systems. Since the unified chain contains all the blocks, the size of the partial bit-stream is relatively large compared to the partial bit-streams in the multiple blocks.

In most cases, there are many shared blocks between wireless standards, and some blocks can be configured by only changing input parameter values (ex. size of IFFT). This will reduce the number of reconfigurable blocks. In this case, the time of reconfiguration in ORBA is greater than that in MRBA. To update one reconfigurable block in the chain when using ORBA, all the wireless chain must be updated. On the other hand, it is possible to update only one reconfigurable block when using MRBA. Thus, the updated PR bit-streams are relatively large when dealing with ORBA.

During runtime, switching from one system to another is simple when using ORBA. The data flow from the upper layer is buffered until the partial reconfiguration ends. On the other hand, switching in MRBA is more complex since all the reconfigurable blocks are partially and sequentially reconfigured. In addition to buffering the incoming data from the upper layer, and during the reconfiguration of each block, all previous blocks in the chain are paused until all the PR blocks are successfully reconfigured. Table I summarizes the comparison between both schemes.

### V. IMPLEMENTATION AND EVALUATION RESULTS

In this section, both reconfigurable architectures are used to implement a unified reconfigurable chain for WIFI-WiMax standards. The Orthogonal Frequency Division Multiplexing (OFDM) is the core of the physical layer of many recent wireless communication systems, such as WIFI and WiMax. To evaluate and test both schemes, partial reconfiguration is applied to an OFDM chain implemented on the ZedBoard. Some blocks are added to the OFDM chain and others are modified in order to make it similar to the physical layer chain of the studied wireless standards. As mentioned earlier, to use MRBA, the blocks of the different chains are compared to determine the similarities and differences. Similar blocks with the same interface and functionality are defined as static parts in the design. Blocks with same interface but different functionality are defined as reconfigurable parts in the design. Fig.5 shows that the architecture of the physical layer of WIFI and WiMax are globally similar with a slight difference in the functionality of some blocks. Using MRBA, the blocks defined as reconfigurable blocks are: TX Controller, Interleaver, Encoder, Scrambler, Modulator, and IFFT. The static part is formed by the remaining blocks that are used for both standards without reconfiguration. The partial bit-streams for the different versions of the reconfigurable blocks have also been generated. When using ORBA for switching between WIFI and WiMax, all the blocks in Fig.5 are grouped to form one reconfigurable block. In this case, only two partial bit-streams for the two systems are created.

Using Vivado tools, partial reconfiguration has been applied to the unified chain to create partial bit-streams. During the floor-planing stage, each reconfigurable block is enclosed in

---

**TABLE I: Comparison between both reconfigurable schemes**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Reconfiguration time</th>
<th>Partial bit-stream</th>
<th>Resources Utilization</th>
<th>Design Complexity</th>
<th>Hardware Update</th>
<th>Runtime Reconfiguration Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One Reconfigurable Unified Chain Architecture (ORBA)</strong></td>
<td>The time needed to transfer the partial bit-stream of the unified chain.</td>
<td>- One partial bit-stream.  - Large bit-stream.</td>
<td>Same resources reserved in two architectures.</td>
<td>- Easier in partial reconfiguration design.  - Lose the ability to have shared blocks as a static part</td>
<td>- All chain is updated when one block is needed to be modified.  - Size of updated bit-stream is Larger.</td>
<td>- Simpler reconfiguration process.  - The data flow from upper layer is buffered until partial reconfiguration ends.</td>
</tr>
<tr>
<td><strong>Multiple Reconfigurable Blocks Architecture (MRBA)</strong></td>
<td>The time needed is the summation of the reconfiguration times of all the reconfigurable blocks.</td>
<td>- ( m ) partial bit-streams.  - Small bit-streams.</td>
<td>Same resources reserved in two architectures.</td>
<td>- Harder Partial Reconfiguration Design.  - Possible to have shared blocks as a static part in the chain</td>
<td>- Possibility to update one reconfigurable block.  - Size of updated bit-stream is smaller.</td>
<td>- More complex reconfiguration process.  - Blocks are reconfigured sequentially and paused until all blocks are reconfigured.</td>
</tr>
</tbody>
</table>
to update a partial bit-stream is greater than that in the case of MRBA. In addition, in ORBA for example, if the system needs to update the Encoder block (40 KB), the partial bit-stream of all the chain (317 KB) is updated. On the other hand, in MRBA, it is possible to update only the Encoder block. Regarding the power consumption during PR, in [10] runtime measurements for power consumed by PS and PL are presented. The results show that an additional power consumption is noticed only in the auxiliary circuits of the PS part.

VI. CONCLUSION

In this paper, the mechanism of applying the partial reconfiguration technique in a reconfigurable system designed for vertical handover between wireless communication systems is discussed. Two reconfigurable architectures (ORBA and MRBA) are proposed and compared. The advantages and disadvantages of these architectures are presented. Selecting the best reconfigurable architecture depends on the desired application. The two reconfigurable architectures were tested for vertical handover in WIFI-WiMax networks. MRBA provides the best reconfiguration time compared to ORBA. However, the ORBA architecture provides a reduced design complexity and simple runtime reconfiguration process.

TABLE II: Partial bit streams size and reconfiguration time

<table>
<thead>
<tr>
<th>Module</th>
<th>Partial Bit-Stream Size</th>
<th>Reconfiguration Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX Controller</td>
<td>30 KB</td>
<td>0.238 ms</td>
</tr>
<tr>
<td>Interleaver</td>
<td>30 KB</td>
<td>0.234 ms</td>
</tr>
<tr>
<td>Encoder</td>
<td>40 KB</td>
<td>0.316 ms</td>
</tr>
<tr>
<td>Scrambler</td>
<td>30 KB</td>
<td>0.231 ms</td>
</tr>
<tr>
<td>Modulator</td>
<td>30 KB</td>
<td>0.241 ms</td>
</tr>
<tr>
<td>IFFT</td>
<td>180 KB</td>
<td>1.41 ms</td>
</tr>
<tr>
<td>One chain</td>
<td>317 KB</td>
<td>4.45 ms</td>
</tr>
<tr>
<td>Full reconfiguration</td>
<td>3951 KB</td>
<td>55.46 ms</td>
</tr>
</tbody>
</table>

REFERENCES


