DYNAMIC RECONFIGURATION TECHNOLOGIES BASED ON FPGA IN SOFTWARE DEFINED RADIO SYSTEM

Ke He (University of Strathclyde, Glasgow, UK; ke.he@eee.strath.ac.uk);
Louise Crockett (University of Strathclyde, Glasgow, UK; louise.crockett@eee.strath.ac.uk);
Robert Stewart (University of Strathclyde, Glasgow, UK; r.stewart@eee.strath.ac.uk)

ABSTRACT

Partial Reconfiguration (PR) is a method for Field Programmable Gate Array (FPGA) designs which allows multiple applications to time-share a portion of an FPGA while the rest continues to operate unaffected. As a result, the physical layer processing architecture in Software Defined Radio (SDR) systems can benefit from reduced complexity and increased design flexibility, as different waveform applications can be grouped into one part of a single FPGA. Waveform switching often means not only changing functionality, but also changing the FPGA clock frequency. However, that is beyond the current functionality of Xilinx-based PR as the clock components such as Digital Clock Managers (DCMs) are excluded from the process of partial reconfiguration. In this paper, we present a novel architecture that combines another reconfigurable technology, Dynamic Reconfigurable Port (DRP), with PR based on a single FPGA in order to dynamically change both functionality and clock frequency with ease. The architecture is demonstrated to reduce hardware utilization significantly compared with standard, static FPGA design. The results presented are based on the Xilinx ISE 12.4 design suite and the Xilinx Virtex-5 LX110T device.

1. INTRODUCTION

Software Defined Radio (SDR) is a technique to support multiple communication standards and services with a single programmable terminal device. Normally the SDR platform employs a set of programmable hardware devices, such as Field Programmable Gate Arrays (FPGAs) and Digital Signal Processors (DSPs) to perform different radio functions to meet the requirements of multiple standards, with these radio functions being controlled or defined by software [1]. Ultimately customers obtain benefits from SDR: they are able to receive waveform expansion or service updating by downloading the relevant software, rather than acquiring new hardware.

Over the past decade, the design of SDR platforms has been widely investigated [2], [3], [4], [7], [8], [14], [15], [20]. In terms of SDR implementation in the physical layer, the Joint Tactical Radio System (JTRS) proposed the SDR architecture for military use, involving a combination of FPGAs, DSPs and General Purpose Processors (GPPs). This architecture is able to switch waveform functionalities and meet the requirements of SDR, but is less well suited to the cost-sensitive consumer market [3], [4].

Xilinx provideds a dynamic reconfiguration technology, referred to as *partial reconfiguration* (PR), which allows one or several portion(s) of the FPGA to be reconfigured on the fly while the rest continue to operate [5]. This enables the end user to dynamically change functionalities by downloading different partial bitstream files, resulting in a higher degree of operational flexibility [6]. Furthermore, PR has commonality with SDR in its core concept: to share the hardware resource and support multiple radio functionalities to the maximum extent. As a result, several studies concerning PR enabled SDR architectures have been published in recent years. Authors in [4] proposed an idea that the SDR architecture may develop a shared resource model with PR based on a single FPGA, replacing the dedicated resource model originally defined by JTRS. In [7], authors proposed an SDR architecture to implement a reconfigurable constellation mapper and FIR filters based on PR. Authors in [8] used Handel-C, which is an efficient language based on the C language to represent hardware, to build a PR architecture.

However, in supporting multiple standards, e.g. LTE, WiMAX and WCDMA or emerging standards, standard switching means that not only the processing logic, but also the clock frequency, has to be reconfigured for some components. To the best of the authors' knowledge, dynamic clock frequency switching for the Digital Front End (DFE) using PR has not been considered before. Normally PR design does not directly dynamically change the clock frequency for a given input oscillator, because the Digital Clock Manager (DCM) used to synthesise the clock is not implemented in reconfigurable logic. Consequently, in isolation PR is insufficient to implement all aspects of switching radio functionalities.

In this paper, we employ another reconfiguration technology, Dynamic Reconfigurable Port (DRP) to

Copyright(c) 2011 The Software Defined Radio Forum Inc. - All Rights Reserved

reconfigure the DCM output frequency while it is operating, and assuming a fixed input oscillator. The technique of DRP could be combined with PR to solve the problems of communication standard or mode switching in terms of clock frequency and dependent functionalities. Furthermore, we propose a hierarchical design methodology based on a single FPGA to support three standards and their different modes (LTE, WiMAX and WCDMA), using PR and DRP technologies. The modulator and DUC components, which operate in the baseband and Intermediate Frequency (IF) processing sections, are further analyzed in detail. By considering both reconfigurable modulation and DUC components, this study constitutes an extension to normal SDR design, which often considers only baseband processing. This architecture increases hardware reuseability significantly and permits standard or mode switching with ease according to the customers' requirements. The implementation results obtained demonstrate that this combined DRP-PR approach offers a great improvement in terms of hardware resource usage, and degree of design flexibility.

This rest of this paper is organised as follows. In Section 2, the DRP and PR technologies are discussed. Section 3 describes the parameters and requirements of the modulation and IF processing blocks for the standards considered in this study (LTE, WiMAX, and WCDMA). Section 4 introduces the hierarchical design methodology involving both PR and DRP, and provides an overview of the system architecture. Section 5 presents results and analysis. Finally, the paper is concluded in Section 6.

2. RECONFIGURABLE TECHNOLOGIES

2.1 Overview of DRP Technology

The DCM component plays an important role in FPGA design: depending on the application, its task is to eliminate clock skew, synthesise a desired clock frequency, or shift phase. Of these tasks, frequency synthesis could be considered one of the DCM's most significant functions [9]. The output frequency of a DCM is controlled via the relationship between the input clock, and the supplied multiplier and divisor parameters, as given in equation (1) [10]. Any integer values within a defined range can be supplied as the multiplier and divisor in order to obtain the desired output frequency. In the case of Virtex-5 DCMs, the multiplier range is from 2 to 33, and the divisor range is from 1 to 32.

$$f_{clk_{output}} = f_{clk_{input}} \times \frac{Multiplier+1}{Divisor+1} \quad (1)$$

In Virtex-5 devices, the DCM primitive includes reconfigurable ports called DRPs, which allow the multiplier and divisor values to be supplied at run-time such that the operating clock frequency may be dynamically changed according to users' requirements, for a single, fixed frequency clock source. The configuration is shown in Figure 1.

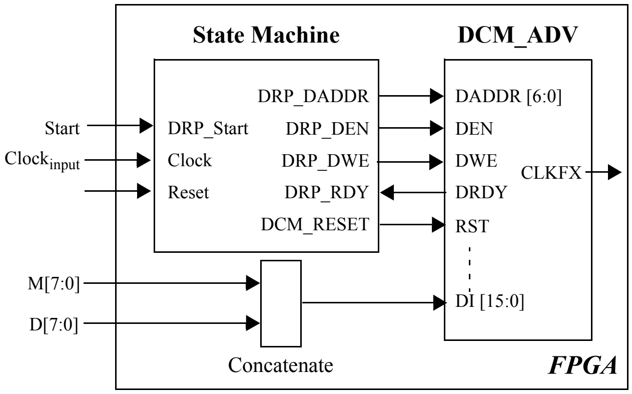

Figure 1 - Architecture of DRP

The essence of the DRP architecture is to integrate a state machine alongside an advanced DCM primitive (labelled as "DCM_ADV" in Figure 1) to make full use of these ports, and to control multiplier and divisor values dynamically. According to [10], several defined steps have to be performed in sequence to achieve reconfiguration. First, the DCM has to go to the reset state. Second, the state machine starts to read from the hex address *50h* if either the multiplier or divisor value is reconfigured. Third, it starts to send the command by writing to the same hex address to instruct the DCM primitive to receive new values. Finally, the DCM reset is released and the output frequency is changed.

The "DRDY" port in the DCM is used to indicate that the read and write cycles are completed, and to instruct the state machine to start the next step. Multiplier and divisor values, both with a wordlength of 8 bits, are concatenated to form a 16-bit control word, where the multiplier value occupies the most significant portion, and the divisor the least significant portion. This combined word is supplied to the "DI" port of the DCM primitive. The "DRP_start" port determines when to start the DRP cycle, and is controlled by external logic.

In the view of the DRP behaviour mentioned above, designers can create logic capable of changing multiplier and divisor values dynamically, in order to obtain different output clock frequencies from a single oscillator source, as required by the user application.

2.2 Overview of PR Technology

PR is a flexible technology which allows functional modification of an FPGA to be achieved by downloading partial reconfiguration bitstream(s) while the device is in operation [6]. The PR design method was initially a difference-based reconfiguration flow which only allowed small changes, e.g. block RAM contents and LUT equations [11]. After that, it developed into a more advanced, "module-based reconfiguration flow" design methodology. This allowed two or more modules which are similar in function to be reconfigured. With the release of their ISE 12.x design suite [12], Xilinx have introduced a new reconfiguration flow based on hierarchical design, which offers improvements in timing results and design reuseability compared to previous tool versions and flows.

Two specific changes have been introduced which lead to the above stated improvements. Firstly, the *bus macro* hardware components, which were used in previous PR design flows to enable communication between static and reconfigurable logic, have been removed. This has the effect that signal delay may be reduced, and hence timing results improved. Secondly, the implementation results of a reconfigurable module can be preserved and imported to another PR project, with timing results etc. preserved. This enhances reuseability, and reduces the time for the FPGA implementation process, thus shortening the design cycle [13].

3. MULTIPLE STANDARDS ANALYSIS

The SDR system aims at building a single architecture to support multiple standards or modes, rather than designing different architectures to cater for each. Consequently, it is necessary to identify the commonalities between these standards in order to develop an efficient PR-based design [14]. With respect to the transmitter chain of wireless protocols, there are three major components: coding, modulation and DFE [15]. In this paper, we analyse the example of the downlink transmitter chain for three different standards (LTE, WiMAX and WCDMA), and consider specifically the modulation and DFE components.

3.1 Modulation Analysis

The function of modulator is to map the bits from the data source into symbols for transmission. The modulation architectures and parameters are hugely different for the communication standards considered here.

In the case of the LTE downlink physical layer, QPSK, 16-QAM and 64-QAM modulation schemes are all employed, and the baseband signal has an adaptive OFDM structure, supporting FFT sizes from 128 up to 2048. The Cyclic Prefix (CP) is necessarily added after the IFFT component to counter inter symbol inference [16]. The normal CP length varies according to the channel bandwidth defined in the specification [17]; for example, the 10MHz bandwidth variant may be 80 or 72 samples, while the 5MHz variant employs 40 and 36 samples CP length. Similarly, WiMAX has a scalable OFDM physical layer to support FFT sizes from 128 to 2048. In this case, the CP length can be 1/4, 1/8, 1/6 or 1/32 of the frame duration [18].

However, the WCDMA standard is somewhat different, in the sense that the baseband signals are not based on an OFDM structure, and the physical layer of the WCDMA standard features a spreader block instead of an IFFT. QPSK and orthogonal variable spreading factor (OVSF) codes are selected to perform modulation. The spreading factor ranges from 4 up to 512 [19].

The parameters of the three considered standards are summarised in Table 1. Note that as a subset of all possible LTE and WiMAX modes are analysed, only two of the FFT sizes mentioned above require to be supported.

Table 1: Modulation Parameters

Standards	Mapper	OFDM		OVSF
		FFT Size	CP Length	Spreading Factor
LTE 5 / 10 MHz	QPSK 16QAM 64QAM	512 , 1024	80, 72, 40, 36 samples	None
WiMAX 3.5 / 5 / 7 / 10 MHz	QPSK 16QAM 64QAM	512 ,1024	1/4, 1/8, 1/16, 1/32 of frame duration	None
WCDMA	QPSK	None	None	4, 8, 16, 32, 64, 128, 256, 512

3.2 Digital Front End Analysis

The DFE section may be considered as a bridge between baseband processing components and the ADC/DAC. In the transmitter, the DFE is referred to as the Digital Up Converter (DUC), and in the receiver, the Digital Down Converter (DDC). In this paper, the example is based on the DUC. The purposes of the DUC are to perform channelisation to remove the out-of-band power and adjacent channel interference to meet the requirements of the standard definition, and to increase the sampling rate to the IF sampling rate, where modulation onto the IF carrier is undertaken [20]. The architecture of a DUC is described by Figure 2.

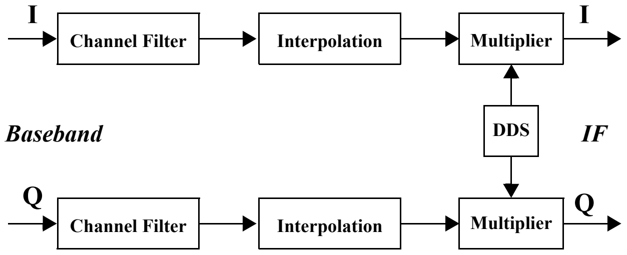

Figure 2 - Architecture of DUC

The channel filter is employed for pulse shaping so that the out of band emissions can be reduced to achieve the requirement of the spectral emission mask. In the interpolation section, a set of FIR filters are used to remove the spectrum image effect produced when the sample rate is raised. The Direct Digital Synthesizer (DDS) component synthesises sine and cosine waves which modulate the interpolated I and Q data from baseband to IF.

The selection of a reasonable IF sample rate and system clock plays an important role in the process of DUC design. From the perspective of efficient implementation, a system clock frequency of at least double the IF sampling rate should be chosen to allow the filters to be time division multiplexed. Sharing hardware permits a reduction in hardware cost to be achieved, most significantly in terms of the number of multipliers which are often implemented using dedicated resources.

Table 2: DUCs Design Parameters

Design Bandwidth	Input Sample Rate (Msps)	IF (MHz)	System Clock (MHz)
LTE 5 MHz	7.68	61.44	245.76
LTE 10 MHz	15.36	61.44	245.76
WCDMA	3.84	61.44	245.76
WiMAX 3.5 MHz	4	64	256
WiMAX 5 MHz	5.6	44.8	179.2
WiMAX 7 MHz	8	64	256
WiMAX 10 MHz	11.2	44.8	179.2

The DUC design parameters of the three standards are shown in Table 2. In this paper, a system clock frequency of 4 times the IF sampling rate is chosen for reasons of implementation efficiency. It is important to note that the required clock frequencies differ according to the standards and modes: specifically, a 245.76 MHz clock is needed for the LTE and WCDMA standards, while WiMAX requires either a 256 MHz clock (for 3.5 MHz and 7 MHz bandwidths), or a 179.2 MHz clock for 5 MHz and 10 MHz bandwidths. Therefore, in total three different clock frequencies are needed to ensure the correct DUC implementation for the standards and modes considered.

4. HIERARCHICAL DESIGN METHODOLOGY AND SYSTEM ARCHITECTURE

4.1 Design Methodology Diagram

Based on the analysis of the DRP and PR reconfiguration technologies, and the studied standards and modes, we propose a design method for an SDR transmitter architecture based on a single FPGA device, controlled by a GPP. The architecture is illustrated in Figure 3.

The first layer is divided into three Reconfigurable Partitions (RPs): error coding, modulation and DUC, in accordance with the functions in the transmitter chain. An RP is defined as an area of the FPGA device to which PR is applied; it has the ability to dynamically change function. Each RP is mutually independent of the others in physical implementation. In other words, the logic and functionality of an RP may be swapped using the technique of partial reconfiguration, while the rest of the FPGA device (that is, the other RPs and static logic) can continue their operation unaffected. A Reconfigurable Module (RM) is defined as the swappable functionality within the RP. One RP may have multiple associated RMs, only one of which occupies the RP at any given time, i.e. they share the allocated hardware resources with time multiplexing.

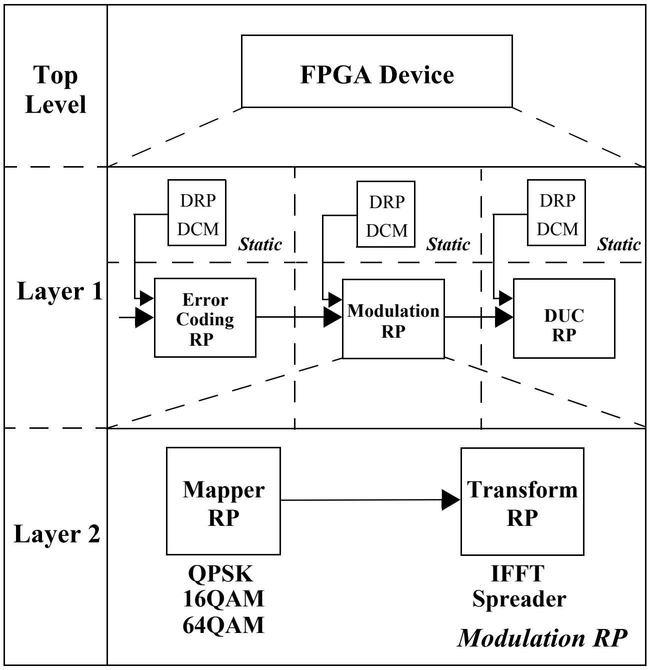

Figure 3 - Hierarchical Design methodology Diagram for SDR transmitter architecture

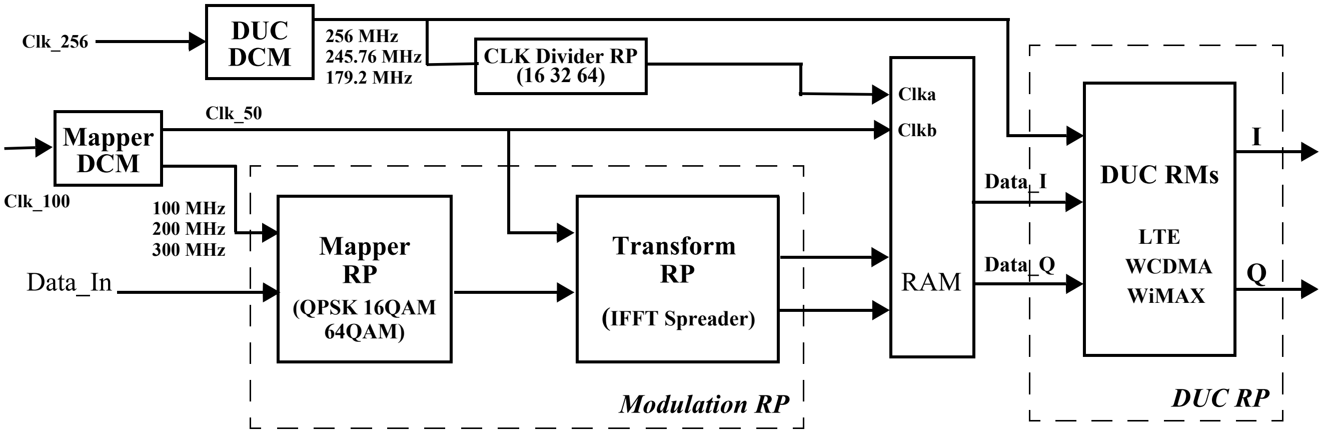

Figure 4 - An Implementation of SDR Architecture to support three standards on FPGA

As is evident from the requirements in Table 2, in order to support standard or mode switching, not only the RMs, but also the clock frequencies require to be reconfigured. Consequently, DCMs with the DRP architecture are created to generate the various clock frequencies required to serve each RP, e.g. in order to change standards from LTE 10 MHz bandwidth to WiMAX 10 MHz bandwidth, the clock frequency has to be reconfigured from 245.76 MHz to 179.2 MHz.

The second layer derives from the further division of the first layer. For example, the modulation RP can be split into a mapper RP and a transform RP. Based on the analysis of the modulation parameters of three standards, the mapper RP has three associated RMs: QPSK, 16-QAM and 64-QAM constellation modules. The transform RP provides two RMs to implement the IFFT and spreader functions respectively.

The benefits of this architecture are primarily that it increases the FPGA device utilization efficiency dramatically, and reduces the number of hardware devices in the physical layer compared with the SDR architecture described in Section 1.

Judiciously choosing and floorplanning reasonable RPs is an important factor in making efficient use of FPGA hardware resources, and allows all of the radio functions for the transmitter to be integrated on a single FPGA device. As a result, this novel architecture is composed of one FPGA and one GPP, resulting in lower cost and power consumption compared with the dedicated resource model architectures, which comprise several discrete hardware components. The GPP is employed to control when and which partitions of the FPGA are reconfigured according to users' requirements.

This proposed architecture has several advantages. It permits enhanced device reuse, because via PR, various RMs can time-share the hardware resource in one RP. In addition, the DRP technology can provide the reconfigurable ability in clock frequency, which could reduced the number of clock oscillators required. The combination of PR and DRP reconfiguration technologies strengthens the degree of flexibility in the design, and hence is very relevant to the requirements of SDR. The increasing sophistication of PR is fundamental to SDR applications, as advances in the technology allow RMs to be switched with shorter reconfiguration time. Taking all of these factors together, the combination of PR and DRP leads to a less complex and lower cost SDR architecture.

4.2 An Overview of the System Architecture

Following a hierarchical design approach, the implementation architecture to support the three standards is illustrated in Figure 4 (note that the error coding component is excluded from this study). Two DCMs with DRP are employed: one is to control the mapper RP and the other is for the DUC RP.

The IFFT module in the transform RP is implemented using a Xilinx FFT core, with the pipelined streaming I/O structure to process the data continuously. In the case of the LTE standard, the subcarrier spacing (Δf) is 15 kHz, hence the interval between two points is equal to 66.7μ s. In order to accommodate the LTE standard, the processing time required by the FFT core must be shorter than the interval so that the core can process the data correctly and continuously. For the FFT core with the pipelined streaming structure and supporting FFT sizes up to 1024 in ISE 12.4, the latency is 63.78 μs when the clock frequency is set to 50 MHz. In other words, the core could meet the needs of the LTE standard with a 50 MHz clock. Using similar logic, the processing ability of the core could also meet the requirements of the WiMAX standard. It is also notable that the FFT core is capable of changing the FFT size and CP length during operation, thus enabling various types of OFDM symbols to be created to comply with the requirements of mode switching between the LTE and WiMAX standards, and modes therein.

The mapper DCM uses a 100 MHz crystal oscillator as the input clock to generate three clock frequencies: 100, 200 and 300 MHz to serve the QPSK, 16-QAM and 64-QAM modules respectively. The data output by the mapper RP can then be fed to the transform RP operating at a clock frequency of 50 MHz. For the WCDMA implementation, the spreader modules contain the OVSF code generator with Spreading Factors (SFs) from 4 to 512.

The two simple dual port Block RAMs are used to bridge the clock domain boundary between the modulation processing components and the DUC RP. The 256 MHz from the clock oscillator is chosen as the input clock for the DUC DCM component as the output frequencies (245.76MHz and 179.2 MHz) may be derived from this frequency by setting different integer multiplier and divisor values. The relations are as follows:

$$245.76\text{MHz} = 256\text{MHz} \times \frac{24}{25} \qquad (2)$$

$$179.2\text{MHz} = 256\text{MHz} \times \frac{7}{10} \qquad (3)$$

The clock divider RP involves three RMs (which divide by factors of 16, 32 and 64) and is added to generate appropriate clock frequencies for reading output data from the Block RAM. For example, the input sample rate of LTE with 10 MHz bandwidth is 15.36 Msps. Therefore, the factor 16 clock divider is used to ensure that the data from the Block RAM is fed into the DUC at the corresponding rate.

The RMs of LTE with 5 and 10 MHz, WCDMA, and WiMAX with 3.5, 5, 7 and 10 MHz all belong to the DUC RP. The channel and interpolation filters are implemented using Xilinx FIR Compiler to support I and Q channels with time division multiplexing. Also, the DDS component is configured using Xilinx DDS Compiler to generate a desired complex sinusoid so that the data frequency can be modulated from baseband to f_{IF}.

5. RESULTS AND ANALYSIS

In this paper, all of the designs are implemented on the Virtex-5 LX110T device using the Xilinx ISE 12.4 software suite. Table 3 gives the hardware resources used by each of the modules without PR in terms of look up tables (LUTs), flip flops (FFs), DSP48Es, Block RAMs and Slices. The f_{max} column shows that the all of the modules can meet the timing requirements according to the demands of the architecture. Since the mapper and clock divider modules occupy few slices without any DSP48Es and RAMs, the number of resources they occupy could be ignored compared with the DUC and transform RPs. As a result, the table only covers the hardware utilization of the DUC and transform RPs.

Table 3: Hardware resource utilization without PR

		LUTs	FFs	Slices	DSP48Es	RAMs	f_{max} (MHz)
DUC	LTE 5	1504	2148	799	11	9	424.1
	LTE 10	1438	2004	872	14	9	355.6
	WCDMA	1493	2020	722	8	9	419.1
	WiMAX 3.5	1629	2223	869	14	9	386.0
	WiMAX 5	1474	2083	817	14	9	390.9
	WiMAX 7	1540	2194	878	17	9	312.0
	WiMAX 10	1449	2130	842	20	9	295.3
Modulation	OFDM	3430	3570	1405	12	5	286.9
	Spreader	34	9	11	0	0	375.7

Table 4 shows the practical hardware utilization statistics of the RMs with the proposed architecture. Compared to Table 3, the number of LUTs increases for the same modules, and this is mainly due to insertion of the *partition pins.* The partition pins enable communication between each reconfigurable module and the surrounding static logic, and their implementation is based on LUTs. In other words, some LUTs must be added to every RM as partition pins in order for successful PR implementation. The numbers of FFs and slices in Table 4 are slightly reduced with the corresponding modules compared with Table 3, but the spreader is an exception. In fact the spreader module has to be artificially augmented with input and output ports, so that the module has the same interface as the OFDM RM (equivalent interfaces are a requirement for RMs associated with a particular RP). In this case, several partition pins are added, hence the number of slices for the spreader increases.

Table 4: Hardware resource utilization with PR and DRP

RP	Standard and Mode RM	LUTs	FFs	Slices	DSP48Es	Block RAMs	f_{max} (MHz)
DUC	LTE 5	1533	2005	543	11	9	336.2
	LTE 10	1455	1873	557	14	9	374.2
	WCDMA	1532	1899	533	8	9	273.8
	WiMAX 3.5	1663	2079	582	14	9	280.0
	WiMAX 5	1497	1964	554	14	9	269.3
	WiMAX 7	1565	2084	612	17	9	354.1
	WiMAX 10	1466	2014	604	20	9	374.0
Modulation	OFDM	3442	3559	1193	12	5	275.6
	Spreader	45	9	12	0	0	188.0

With respect to the PR design rule, the size of the RP must be specified to accommodate the most complicated module. Therefore, the worst case scenarios in terms of slices, DSP48Es and Block RAMs for all RMs associated with a particular RP are selected from Table 4 to determine the required size of that RP.

Since the normalized frequency requirements of the filter designs are identical for the LTE 5MHz and 10MHz bandwidths, these parts of them can be shared, and therefore the hardware usage of LTE 10 MHz bandwidth could in fact represent that of both the 5MHz and 10MHz bandwidth modes combined. Similarly, hardware utilization of WiMAX 7 and 10 MHz bandwidths could represent that of all the WiMAX modes combined. To support all standards and modes in a fixed function FPGA design, LTE 10 MHz, WCDMA, WiMAX 7MHz, and WiMAX 10MHz could all be implemented as the DUC components, and the OFDM and spreader modules would be implemented as the transform components. This is shown in Figure 5.

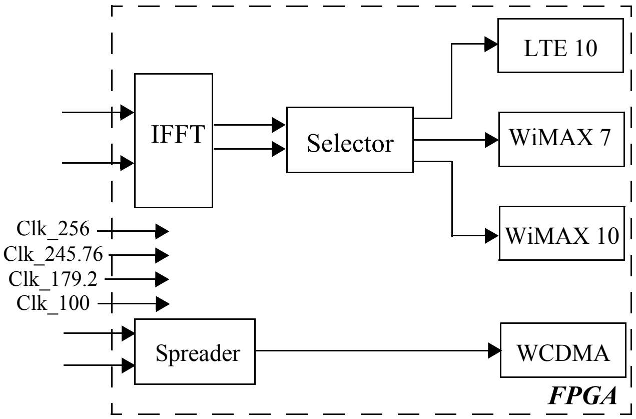

Figure 5 - Fixed function FPGA Design to support multi-standards

The implementation results of the fixed function multi-standards design show that 4657 slices, 71 DSP48Es and 41 Block RAMs are used. Therefore, a larger FPGA device would be needed (there are only 64 DSP48Es in the Virtex-5 LX110T device), resulting in higher terminal size, power consumption and device cost.

By contrast, the new PR/DRP architecture allows multiple RMs to time share the resources, provided that each RP has been adequately defined. This architecture is illustrated in Figure 6. The transform RP contains the IFFT and spreader RMs, and the DUC RP involves the RMs of three standards and seven modes. As a result, the hardware usage is the sum of the largest DUC and transform RPs, equivalent to 1805 slices, 32 DSP48Es and 14 Block RAMs, as shown in Equation 4.

$$Resource = max(transform(RP)) + max(DUC(RP)) \quad (4)$$

As shown in Figure 7, the proposed architecture could achieve reductions of 61.24%, 54.93% and 65.85% in terms

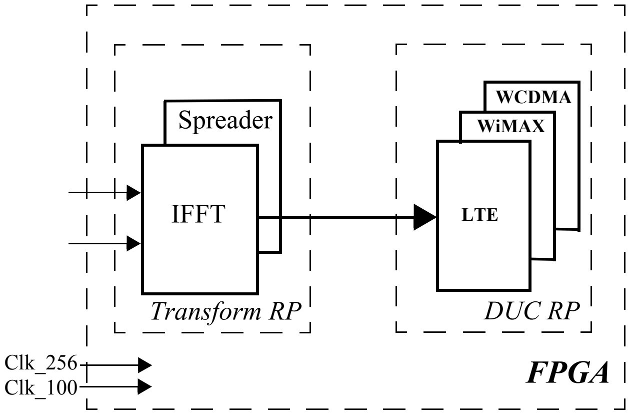

Figure 6 - PR and DRP Design Architecture

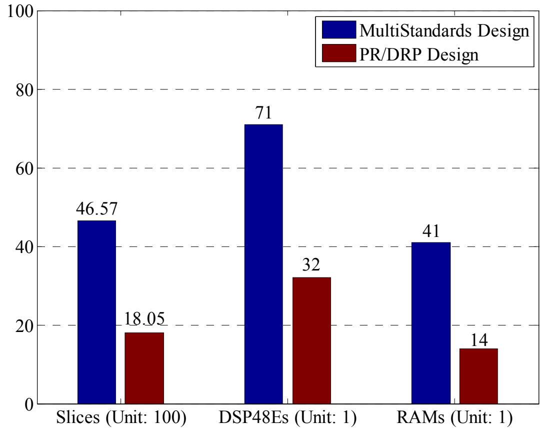

Figure 7 - Hardware utilization comparison between multistandards and PR/DRP designs

of the number of slices, DSP48Es and RAMs required, respectively. Therefore, the PR/DRP design method is seen to reduce FPGA resource utilisation significantly, compared to a fixed-function approach. Moreover, only two clock oscillators are employed (100MHz and 256MHz), while clock oscillators for 245.76 MHz and 179.2 MHz are not needed, resulting in further lower device cost and a simpler architecture.

6. CONCLUSION

In this paper, a novel physical layer architecture for an SDR has been proposed, using PR and DRP reconfiguration technologies based on a single FPGA device. Thus has been developed to support LTE, WCDMA and WiMAX standards and their various modes in the transmitter chain. It is an extension compared to many other SDR architectures as it involves not only baseband but also IF processing. It is demonstrated that the proposed architecture could achieve reductions of 61.24%, 54.93% and 65.85% in respect of slices, DSP48Es and RAMs respectively, while two fewer clock oscillator inputs are required compared to traditional, fixed function FPGA design. Therefore, the proposed method could reduce the SDR device size, power consumption and cost significantly, while maintaining a high degree of design and function switching flexibility. These advantages would enable SDR terminal devices to have smaller sizes and lower prices in the future.

7. REFERENCE

[1] K. Tan, H. Liu, J. Zhang, Y. Zhang, J. Fang and G.M. Voelker. Sora: High-Performance Software Radio Using General-Purpose Multi-Core Processors. *Communications of the ACM* vol.54, Issue 1, January 2011

[2] A.A. Kountouris and C. Moy. "Reconfiguration in Software Radio System". In *Proceeding of 2nd Karlsruhe Workshop on Software Radio*, Germany March 20-21, 2002

[3] N. Bagherzadeh, T. Eichenberg. "Mobile software defined radio solution using high-performance, low-power reconfigurable DSP architecture". In *Proceeding of SDR 2005 Technical Conference and Product Exposition,* Anaheim, CA, USA, November 14-18, 2005

[4] C. Kao "Benefits of Partial Reconfiguration", *Xcell journal*, fourth quarter 2005, pp 65-67

[5] David Dye Partial Reconfiguration of Virtex FPGAs in ISE 12. [online], Xilinx Inc, July 2010. Available at:<www.xilinx.com/support/documentation/white_papers/wp374_partial_reconfig_virtex_fpga.pdf>

[6] Xilinx, Partial Reconfiguration User Guide. [online], Xilinx Inc, October 2010. Available at:<www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/ug702.pdf>.

[7] J.P. Delahaye, J. Palicot, C. Moy and P. Leray, "Partial Reconfiguration of FPGAs for Dynamical Reconfiguration of a Software Radio Platform". In *Proceeding of 16th IST Mobile & Wireless Communications Summit*, Budapest, Hungary 1-5 July 2007.

[8] K.G. Nezami, P.W. Stephens and S.D. Walker, "Handel-C Implementation of Early-Access Partial Reconfiguration for Software Defined Radio". In *Proceedings of IEEE Wireless Communications & Networking Conference (WCNC).* Las Vegas, NV, USA March 31 - April 3, 2008

[9] Xilinx, Virtex-5 FPGA User Guide. [online], Xilinx Inc, May 2010. Available at:<www.xilinx.com/support/documentation/user_guides/ug190.pdf>.

[10] Xilinx, Virtex-5 Configuration User Guide. [online], Xilinx Inc, August 2010. Available at:<www.xilinx.com/support/documentation/user_guides/ug191.pdf>

[11] E. Eto, Difference-Based Partial Reconfiguration. [online], Xilinx Inc, December 2007. Available at:<www.xilinx.com/support/documentation/application_notes/xapp290.pdf>.

[12] Xilinx, PlanAhead User Guide. [online], Xilinx Inc, May 2010. Available at:<www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/planahead_userguide.pdf>.

[13] Xilinx, Hierarchical Design Methodology Guide. [online], Xilinx Inc, September 2010. Available at:<www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/Hierarchical_Design_Methodology_Guide.pdf>.

[14] J.J. Delahaye, C. Moy, P. Leray and J. Palicot, "Managing Dynamic Partial Reconfiguration on Heterogeneous SDR Platforms", In *Proceeding of SDR 2005 Technical Conference and Product Exposition,* Anaheim, CA, USA, November 14-18, 2005

[15] Y.Lin, H.Lee, M.Who, Y.Harel, S.Mahlke, T.Mudge, C.Chakrabarti, and K.Flautner. "SODA: A low-power architecture for software radio". In *Proceeding of 33rd ISCA (International Symposium on Computer Architecture)*. Boston, MA USA, June 17-21, 2006.

[16] H. Holma and A. Toskala *LTE for UMTS OFDMA and SC-FDMA Based Radio Access*. Chichester, UK, 2009

[17] 3GPP TS 36.211, *(Release 9),* January 2010

[18] IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2001.

[19] GPP TS 25.213, *(Release 7)*, March 2006

[20] J W. Tuttlebee, *Software Defined Radio: Enabling Technologies*, Chichester, UK,John Wiley & Sons, 2002.