Dynamic Reconfiguration Technologies Based on FPGA in Software Defined Radio System

Ke He · Louise Crockett · Robert Stewart

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 of the device continues to operate unaffected. Using this strategy, 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 PR processes 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 also the clock frequency. The architecture is demonstrated to reduce hardware utilization significantly compared with standard, static FPGA design.

Keywords Field programmable gate array · Software defined radio · Partial reconfiguration · Dynamic reconfigurable port · 3GPP LTE · IEEE 802.16 · IEEE 802.11 · 3GPP WCDMA

K. He (✉) · L. Crockett · R. Stewart
EEE Department, University of Strathclyde,
204 George Street,
Glasgow G1 1XW Scotland, UK
e-mail: ke.he@eee.strath.ac.uk

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 [2–4, 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]. More recently new wireless opportunities such as TV White Spaces (the secondary use of TV spectrum for applications such as rural broadband and machine to machine communication), opens a new interesting area where SDR can be applied. The secondary use of TV spectrum is unlikely to be driven by only one standard, and therefore there exists the opportunity to support multi-standards at (low power) community basestations and use one FPGA hardware platform to potentially support multiple standards from the myriad of new and emerging candidate TV White Space

standards (IEEE 802.11af, IEEE 802.22, 3GPP LTE-TDD) [22].

Xilinx provides a dynamic reconfiguration technology, referred to as *partial reconfiguration* (PR), which allows one or more parts of the FPGA to be reconfigured on the fly while the rest continue to operate normally [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 language based on the C language to represent hardware, to build a PR architecture.

However, in supporting multiple standards, e.g. 3GPP LTE, IEEE 802.16, IEEE 802.11 and 3GPP 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. This requires the consideration of dynamic clock frequency switching for the Digital Front End (DFE) using PR and is a core component of this paper. Normally PR design does not directly dynamically change the clock frequency for a given input oscillator, because the Digital Clock Manager (DCM) used to synthesize 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 assume that the input oscillator has a single, fixed frequency, and employ a second reconfiguration technology, Dynamic Reconfigurable Port (DRP), to reconfigure the DCM output frequency while the radio is operating. The technique of DRP could be combined with PR to address the difficulties 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 four standards and a subset of their modes (3GPP LTE, IEEE 802.16e, IEEE 802.11n and 3GPP 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 reusability 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.

The rest of this paper is organized 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 (3GPP LTE, IEEE 802.16e, IEEE 802.11n and 3GPP WCDMA). Section 4 introduces the hierarchical design methodology involving both PR and DRP, and provides an overview of the system architecture. Sections 5 and 6 present results, analysis and conclusions.

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, synthesize 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 Eq. 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.

$$f_{clk\_output} = f_{clk\_input} \times \frac{\text{Multiplier} + 1}{\text{Divisor} + 1} \qquad (1)$$

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 Fig. 1.

The essence of the DRP architecture is to integrate a state machine alongside an advanced DCM primitive (labeled as "DCM_ADV" in Fig. 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 performs a read operation from the hex address *50 h* if either the multiplier or divisor value is reconfigured.

Figure 1 Architecture of DRP.
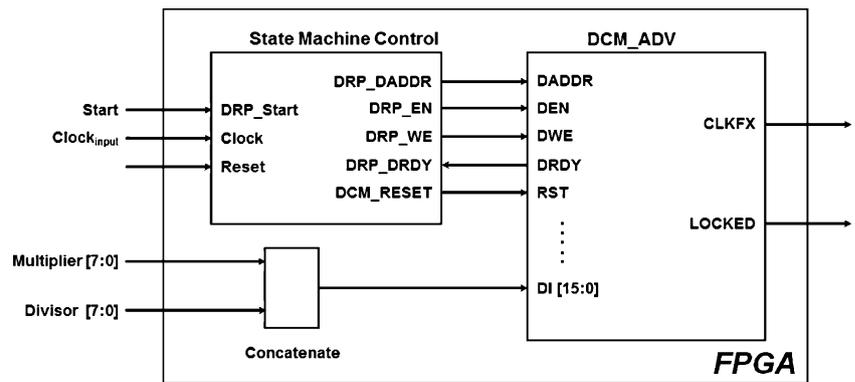
Third, it sends the command by writing to the same hex address to instruct the DCM primitive to receive new values. Finally, the DCM reset is released and the output frequency is changed.

The "DRDY" port of 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 view of the DRP behavior 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. More recently, with the release of their ISE 12.x and 13.x design suite [12], Xilinx have introduced a new reconfiguration flow based on hierarchical design, which offers improvements in timing results and design reusability 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 regions of 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. retained. This enhances reusability, and reduces the time for the FPGA implementation process, thus shortening the design cycle [13].

3 Multiple Standards Analysis

The SDR system aims to build 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 four different standards (3GPP LTE, IEEE 802.16e, 3GPP WCDMA and IEEE 802.11n), and consider specifically the modulation and DFE components.

3.1 Modulation Analysis

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

In the case of the 3GPP 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 CP length of the 10 MHz bandwidth variant may be 80 or 72 samples, while the 5 MHz variant employs 40 and 36

Table 1 Modulation parameters.

Standards	Mapper	OFDM		OVSF
		FFT size	CP Length	Spreading Factor
3GPP LTE 5/10 MHz	QPSK 16QAM 64QAM	512 1024	80,72 40, 36 samples	None
IEEE 802.16e 3.5/5/7/10 MHz	QPSK 16QAM 64QAM	512 1024	1/4, 1/8, 1/16, 1/32 of frame duration	None
IEEE 802.11n 20 MHz	QPSK 16QAM 64 QAM	64	1/4, 1/8 of frame duration	None
3GPP WCDMA	QPSK	None	None	4,8,16,32,64,128,256,512

samples CP length. Similarly, IEEE 802.16e 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]. With regard to the IEEE 802.11n standard, the FFT size is fixed to 64 and CP length can be 1/4 and 1/8 of the frame duration [19].

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 (SF) ranges from 4 up to 512 [20].

The parameters of the four considered standards are summarised in Table 1. Note that as a subset of all possible 3GPP LTE, IEEE 802.16e and IEEE 802.11n modes are analysed, only three of the FFT sizes mentioned above require to be supported.

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, as the Digital Down Converter (DDC). In this paper, the example is based on the DUC. The purpose of the DUC are to perform channelization filtering, and to increase the sampling rate to the IF sampling rate, at which modulation of the IF carrier is undertaken [21]. The architecture of a DUC is described by Fig. 2.

The channel filter is employed for pulse shaping so that out of band emissions can be reduced to achieve the requirements 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 synthesizes sine and cosine IF carrier frequencies 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 FPGA implementation cost to be achieved, most significantly in terms of multipliers which are often implemented using dedicated resources.

The DUC design parameters of the four standards are shown in Table 2. The sample rates are defined by the corresponding standard or mode drafts, but the IEEE 802.11n is an exception. In this paper, 30 Msps was selected for IEEE 802.11n 20 MHz bandwidth. 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 3GPP LTE and WCDMA standards, while IEEE 802.16e 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. In addition, IEEE 802.11n requires a 240 MHz clock. Therefore, in total four different clock frequencies are needed to ensure the correct DUC implementation for the standards and modes considered.

Figure 2 Architecture of DUC.
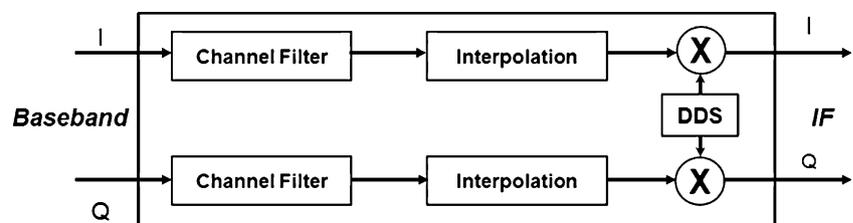
Table 2 DUC design parameters.

Design	Input sample rate (Msps)	IF (MHz)	System clock (MHz)
3GPP LTE 5 MHz	7.68	61.44	245.76
3GPP LTE 10 MHz	15.36	61.44	245.76
3GPP WCDMA	3.84	61.44	245.76
IEEE 802.16e 3.5 MHz	4	64	256
IEEE 802.16e 5 MHz	5.6	44.8	179.2
IEEE 802.16e 7 MHz	8	64	256
IEEE 802.16e 10 MHz	11.2	44.8	179.2
IEEE 802.11n 20 MHz	30	60	240

4 Hierarchical Design Methodology and System Architecture

4.1 Design Methodology

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 Fig. 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.

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 3GPP LTE 10 MHz bandwidth to IEEE 802.16e 10 MHz bandwidth, the clock frequency has to be reconfigured from 245.76 MHz to 179.2 MHz. Similarly, the clock frequency has to be changed from 245.76 MHz to 240 MHz when operation switches from 3GPP LTE 5 MHz to IEEE 802.11n.

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 four 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
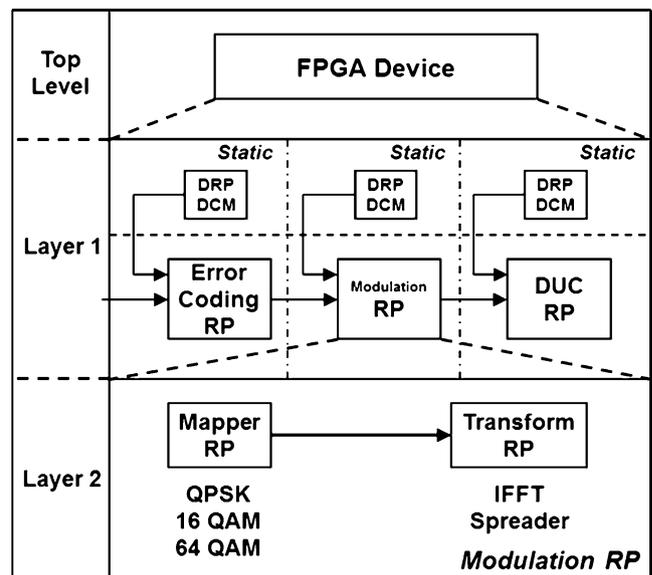
Figure 3 Hierarchical design methodology for SDR transmitter architecture.

Figure 4 An implementation of SDR architecture to support three standards on FPGA.
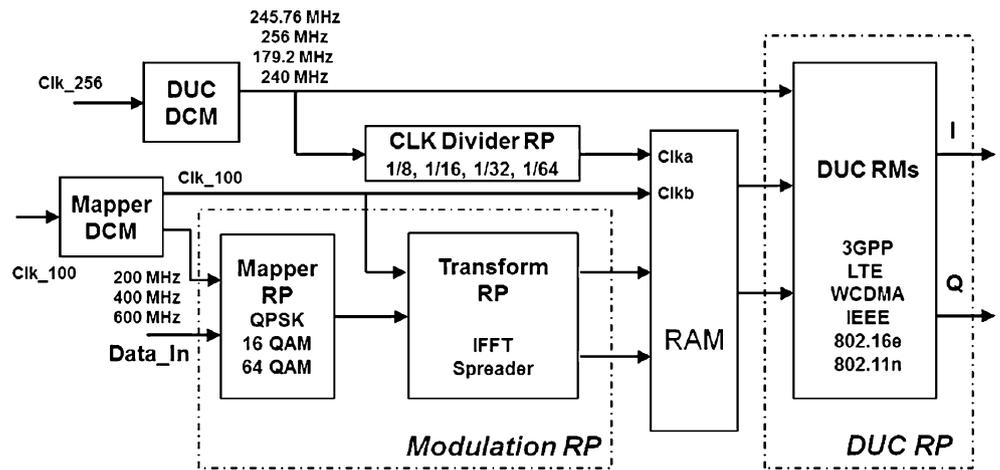
addition, the DRP technology provides the ability to reconfigure clock frequency, which could reduce the number of clock oscillators required. The combination of PR and DRP reconfiguration technologies strengthens the degree of flexibility in the design, and hence is very relevant to the requirements 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 into account, 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 the hierarchical design approach described in Section 4.1, the implementation architecture to support the four standards is illustrated in Fig. 4 (note that the error coding component is excluded from this study). Two DCMs with DRP are employed: one is used 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 Radix-4 burst I/O structure to process the data. This structure uses fewer resources than the pipelined streaming structure. In the

case of the 3GPP LTE standard, the subcarrier spacing (Δf) is 15 kHz, hence the interval between two points is equal to 66.7 μs. In order to accommodate the 3GPP 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 Radix-4 burst structure and supporting FFT sizes up to 1024 in ISE 12.4, the latency is 34.26 μs when the clock frequency is set to 100 MHz. In other words, the core could meet the needs of the 3GPP LTE standard with a 100 MHz clock. Using similar logic, the processing ability of the core could also meet the requirements of the IEEE 802.16e. The latency is 2.38 μs when the FFT size is set to 64 with the same clock, which also meets the requirements of IEEE 802.11n 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 3GPP LTE, IEEE 802.16e and 802.11n standards, and modes thereof.

The mapper DCM uses a 100 MHz crystal oscillator as the input clock to generate three clock frequencies: 200, 400 and 600 MHz to serve the QPSK, 16-QAM and 64-QAM modules respectively. The data output by the mapper

Table 3 Hardware resource utilization without PR.

		LUTs	FFs	Slices	DSP48Es	BlockRAMs	f_{max} (MHz)
DUC	3GPP LTE 5	1504	1438	799	11	9	424.1
	3GPP LTE 10	1438	2004	872	14	9	355.6
	3GPP WCDMA	1493	2020	722	8	9	419.1
	IEEE 802.16e 3.5	1629	2223	869	14	9	386.0
	IEEE 802.16e 5	1474	2083	817	14	9	390.9
	IEEE 802.16e 7	1540	2194	878	17	9	312.0
	IEEE 802.16e 10	1449	2130	842	20	9	295.3
	IEEE 802.11n 20	1253	1653	555	15	9	274.4
Modulation	OFDM	1714	2183	797	9	10	297.3
	Spreader	34	9	13	0	0	375.7

Table 4 Hardware resource utilization with PR and DRP.

		LUTs	FFs	Slices	DSP48Es	BlockRAMs	f_{max} (MHz)
DUC	3GPP LTE 5	1532	2005	637	11	9	368.9
	3GPP LTE 10	1455	1873	611	14	9	391.1
	3GPP WCDMA	1531	1899	566	8	9	344.8
	IEEE 802.16e 3.5	1662	2079	648	14	9	341.6
	IEEE 802.16e 5	1497	1964	634	14	9	335.2
	IEEE 802.16e 7	1564	1984	658	17	9	380.2
	IEEE 802.16e 10	1465	2014	603	20	9	376.9
	IEEE 802.11n 20	1263	1653	490	15	9	358.2
Modulation	OFDM	1725	2170	748	9	10	315.7
	Spreader	45	9	15	0	0	204.3

RP can then be fed to the transform RP operating at a clock frequency of 100 MHz. For the WCDMA implementation, the spreader modules contain the OVSF code generator with SFs from 4 to 512.

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 (245.76 MHz, 179.2 MHz and 240 MHz) may be derived from this frequency by setting different integer multiplier and divisor values. The relations are as follows:

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

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

$$240\text{MHz} = 256\text{MHz} \times \frac{15}{16} \qquad (4)$$

The clock divider RP involves four RMs (which divide by factors of 8, 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 3GPP LTE with 10 MHz bandwidth is 15.36 Msps. Therefore, the factor 16 clock divider is applied to ensure that the data from the Block RAM is fed into the DUC at the corresponding rate.

The RMs of 3GPP LTE with 5 and 10 MHz, 3GPP WCDMA, IEEE 802.16e with 3.5, 5, 7 and 10 MHz and IEEE 802.11n 20 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 the desired complex sinusoid so that the data frequency can be modulated from baseband to f_{IF}.

5 Results and Analysis

In this section, the implementation results of normal FPGA configuration design and proposed PR/DRP design are first listed and compared. Then the architectures of fixed function FPGA and proposed PR/DRP to support multi-standards and modes are described in detail and evaluated. Finally, the bitstream size of each RMs in the proposed PR/DRP architecture is presented.

5.1 Implementation Results

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 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 is
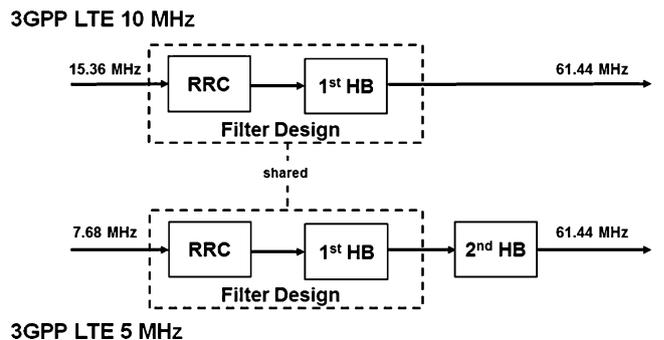
Figure 5 Filter designs of 3GPP LTE 10 and 5 MHz.

insignificant compared to the DUC and transform RPs, and can be omitted from further analysis. As a result, Table 3 only covers the hardware utilization of the DUC and transform RPs.

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 primarily 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 compared to the corresponding modules as given in Table 3. This is mainly due to the implementation compression, as the RMs have to be placed and routed within the RPs regions, rather than the entire device. However, 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.

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.

5.2 Fixed Function and PR/DRP Designs

In this study, the Root Raised Cosine (RRC) filter interpolated by 2 is employed in the channel filter section, and the Half Band (HB) filter is employed in the interpolation filter section when up-sampling by 2. Therefore, as an example, the 3GPP LTE 10 MHz and 5 MHz use one and two HB filters respectively to achieve sample rate conversion from their input sample rates to the selected IF. The filter designs of LTE 10 MHz can be shared by the LTE 5 MHz as

Table 5 Hardware usage for the three HB filters.

| Design	Taps	Slices	DSP48Es	RAMs
2nd HB 3GPP LTE 5	7	104	1	0
3rd HB IEEE 802.16e 3.5	7	108	1	0
2nd HB IEEE 802.16e 5	7	102	1	0
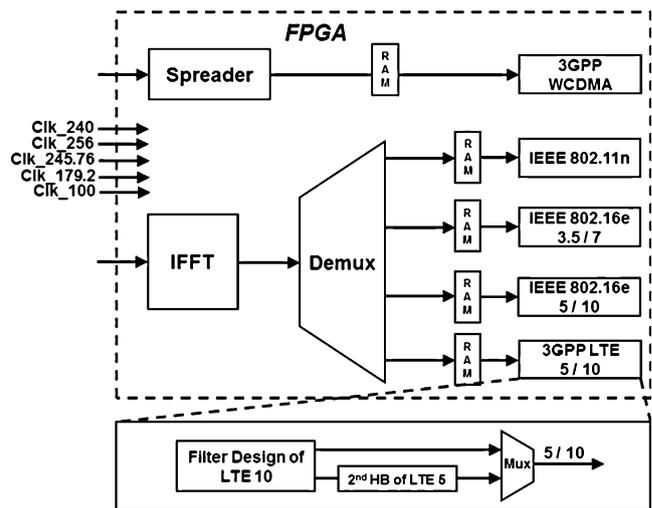
Figure 6 Fixed function FPGA design to support multi-standards.

their normalized passband and stopband requirements are identical, as shown in Fig. 5. The DDS component could also be shared, because both the 5 MHz and 10 MHz variants use the same IF carrier frequency.

As a result, the hardware usage of 3GPP LTE 10 MHz bandwidth DUC, plus the second HB filter in 3GPP LTE 5 MHz DUC could represent that of both the 5 MHz and 10 MHz bandwidth modes combined. Similarly, the hardware utilization of the IEEE 802.16e 7 MHz DUC, plus the third HB filter required for the 3.5 MHz mode, and the IEEE 802.16e 10 MHz DUC plus the second HB filter in the 5 MHz DUC could represent that of all the IEEE 802.16e modes combined.

The hardware usage of the HB filters is illustrated in Table 5. Each HB filter occupies 1 DSP48E and approximately 100 slices. Taking into account the identified commonality between modes of the same standard, as discussed above, the DUC would require
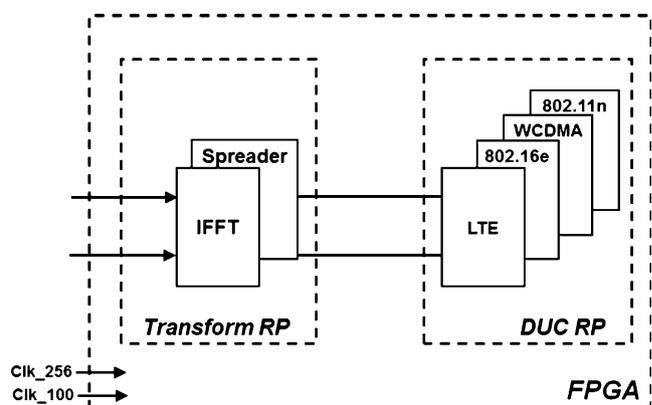
Figure 7 PR and DRP design architecture.

Figure 8 Hardware utilization comparison between fixed multi-standards and PR/DRP designs.
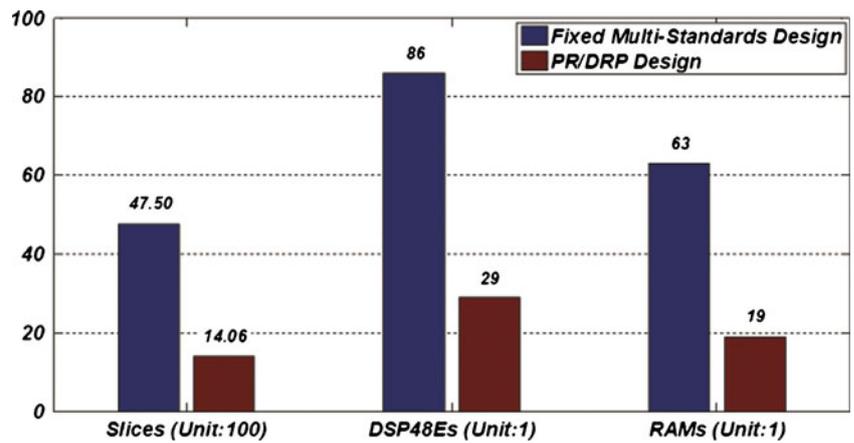
to implement only a subset of modes, with HB filters added to achieve the others. That is to say, if the DUC comprises implementations of the DUCs for 3GPP LTE (10 MHz), 3GPP WCDMA, IEEE 802.16e (7 MHz), IEEE 802.16e (10 MHz), IEEE 802.11n, and three additional HB filters, then the full set of considered modes can be supported. Therefore the architecture is as shown in Fig. 6.

The dual port Block RAMs also have to be considered to bridge the clock domain boundary between the IFFT component and the various DUC components. The implementation results of 3GPP LTE 10 MHz, 3GPP WCDMA, IEEE 802.16e 7 MHz and IEEE 802.16e 10 MHz show that 4,436 slices, 83 DSP48Es and 63 Block RAMs are used. As a result,

the hardware usage of the fixed function multi-standards design is the sum of the implementation results above and three HB filters, i.e. 4,750 slices, 86 DSP48Es and 63 Block RAMs from the hardware utilization perspective. 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 Fig. 7. The transform RP contains the IFFT and spreader RMs, and the DUC RP involves the RMs of four standards and eight modes. As a result, the hardware usage is the sum of the largest

Table 6 Partial Bitstream size onVirtex-5 lx 110 t device.

RP	RMs	Number of frames	Bitstream size (KB)
DUC	3GPP LTE 5 3GPP LTE 10	42	282
	3GPP WCDMA		
	IEEE 802.16e 3.5		
	IEEE 802.16e 5		
	IEEE 802.16e 7		
	IEEE 802.16e 10		
	IEEE 802.11n		
Transform	IFFT Spreader	46	292
Mapper	QPSK 16 QAM	4	24
	64 QAM		
CLK Divider	1/8 1/16	3	19
	1/32		
	1/64		
Configuration File		Whole FPGA	3799

DUC and transform RPs, equivalent to 1,406 slices, 29 DSP48Es and 19 Block RAMs, as shown in Eq. 5.

$$\text{Resource} = \max(\text{Transform}(RP)) + \max(\text{DUC}(RP)) \quad (5)$$

As shown in Fig. 8, the proposed architecture could achieve reductions of 70.4%, 66.3% and 69.8% in terms of the number of slices, DSP48Es and RAMs required, respectively. Therefore, the PR/DRP design method is seen to reduce FPGA resource utilization significantly, compared to a fixed-function approach. Moreover, only two clock oscillators are employed (100 MHz and 256 MHz), while clock oscillators for 245.76 MHz, 179.2 MHz and 240 MHz are not needed, resulting in further reduction in device cost, and a simpler architecture.

5.3 Partial Bitstream

Table 6 shows the bitstream size of the various RPs considered in this study. The size of RP area is based on the *frame* in the floorplanning process. The configuration memory is grouped by columns and the column can be further divided into sub-columns called frames. The frame is the smallest unit of configuration memory and hence all operations must be based on whole frames. In terms of the Virtex-5 device, the frame is with 20 CLB heights and 1 CLB width [6, 10]. Since each RM within one RP shares the same hardware resource, they have the same bitstream size. The size of partial bitstream is crucial to the reconfiguration overhead assuming that the download speed is fixed. It may therefore be seen from Table 6 that switching functions can be achieved in significantly less time using PR, as compared to full-FPGA reconfiguration.

After downloading the initial configuration file, the architecture could switch functions flexibly according to users' requirements. In addition, the extra 40% resource of DUC RP has been considered in order to achieve timing requirements and implement PR designs successfully. The architecture also provides the potential to integrate the more complex DUC designs to extend to more standards and modes in the future. As a result, this PR/DRP architecture could be viewed as providing a high degree of function switching and design flexibility.

6 Conclusion

In this paper, a novel physical layer architecture for an SDR has been proposed, using PR and DRP reconfig-

uration technologies based on a single FPGA device. An example architecture has been developed to support 3GPP LTE, 3GPP WCDMA, IEEE 802.16e and IEEE 802.11n standards and a subset of 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, for the target device considered, the proposed architecture could achieve reductions of 70.4%, 66.3% and 69.8% in respect of slices, DSP48Es and RAMs respectively, while three fewer clock oscillator inputs are required compared to traditional, fixed function FPGA design. Moreover, this architecture provides the potential to integrate new standards or modes into the design with ease. 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.

References

1. Tan, K., Liu, H., Zhang, J., Zhang, Y., Fang, J., & Voelker, G. M. "Sora: High-performance software radio using general-purpose multi-core processors". *Communications of the ACM* vol. 54, Issue 1, January 2011
2. Kountouris, A. A., & Moy, C. "Reconfiguration in software radio system". In *Proceeding of 2nd Karlsruhe Workshop on Software Radio*, Germany March 20–21, 2002
3. Bagherzadeh, N., & Eichenberg, T. "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. Kao, C. "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. Delahaye, J. P., Palicot, J., Moy, C., & Leray, P. "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. Nezami, K. G., Stephens, P. W., & Walker, S. D. "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. Eto, E. *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. Delahaye, J. J., Moy, C., Leray, P., & Palicot, J. "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. Lin, Y., Lee, H., Who, M., Harel, Y., Mahlke, S., Mudge, T., et al. "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. Holma, H., & Toskala, A. (2009). *LTE for UMTS OFDMA and SC-FDMA based radio access*. Chichester, UK.
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. IEEE Standard for Local and metropolitan area networks Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2009.
20. 3GPP TS 25.213, *(Release 7)*, March 2006
21. Tuttlebee, W. (2002). *Software defined radio: Enabling technologies*. Chichester, UK: Wiley.
22. Darbari, F., Brew, M., Weiss, S., & Stewart, R. "Practical aspects of broadband access for rural communities using a cost and power efficient multi-hop/relay network". In *Proceeding of IEEE GLOBECOM.* Miami, USA, December 2010, pp. 731–735.

Louise Crockett received the MEng and PhD degrees from the Department of Electrical and Electronic Engineering, University of Strathclyde, Glasgow, in 2003 and 2008, respectively. She is currently a Research Fellow in the same department. Her research interests are in the area of DSP implementation, in particular for communications architectures based on FPGAs. Louise is also engaged in DSP education, and she is a co-author of the successful DSP Primer course, which is available through the Xilinx University Program.

Ke He received the B.Sc in computer science from the Beijing Jiaotong University, Beijing, China, in 2008. Currently, he is working toward the PhD degree in the electronic & electrical engineering at the University of Strathclyde, Glasgow, Scotland, UK. His research interests include software defined radio, dynamic reconfiguration technologies in FPGA, digital IF processing, PHY implementation for LTE and IEEE 802.16 and TV white space spectrum.

Robert Stewart is currently a Professor in the Department of Electronic and Electrical Engineering, University of Strathclyde, where he runs the DSP Enabled Communications Research Group, focussing on DSP and PHY layer implementations ranging from topics of LTE design, FPGA implementation, MIMO based systems. In 2006, through the support of the Xilinx University Program (XUP) Bob was endowed as the Xilinx Professor of DSP at Strathclyde. Bob is also the Director of the Wireless White Space Centre at Strathclyde, which has a research focus on the white space spectrum and cognitive radio type architectures for communication. Since 1997 Bob has also been a Visiting Professor at UCLA Extension, USA, where he presents an industry short course on DSP and DSP for FPGAs, and since 2004 has been a Director of the technology company, Steepest Ascent Ltd.