Design Space Exploration in an FPGA-Based Software Defined Radio
Matthieu Gautier, Ganda Stephane Ouedraogo, Olivier Sentieys

To cite this version:
Matthieu Gautier, Ganda Stephane Ouedraogo, Olivier Sentieys. Design Space Exploration in an FPGA-Based Software Defined Radio. Euromicro Conference on Digital System Design, Aug 2014, Verona, Italy. 10.1109/DSD.2014.44 . hal-01084781

HAL Id: hal-01084781
https://inria.hal.science/hal-01084781
Submitted on 20 Nov 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright
Design Space Exploration in an FPGA-Based Software Defined Radio

Matthieu Gautier, Ganda Stephane Ouedraogo and Olivier Sentieys
INRIA, University of Rennes, IRISA, France
6 rue Kerampont, 22300 Lannion, France
matthieu.gautier@inria.fr, ganda-stephane.ouedraogo@inria.fr, olivier.sentieys@inria.fr

Abstract—The FPGA (Field Programmable Gate Array) technology is expected to play a key role in the development of Software Defined Radio (SDR) platforms. To this aim, leveraging the nascent High-Level Synthesis (HLS) tools, a design flow from high-level specifications to Register-Transfer Level (RTL) description can be thought. Based on such a flow, this paper describes the Design Space Exploration (DSE) that can be achieved using loop optimizations. The main objective is to demonstrate the compile-time flexibility of an architecture when associated with a reconfigurable platform. Throughout both IEEE 802.15.4 and IEEE 802.11g waveform examples, we show how the FPGA resources can be tuned according to a targeted throughput.

Keywords—Software Defined Radio (SDR), Design Space Exploration, Field Programmable Gate Array (FPGA), High-Level Synthesis (HLS).

I. INTRODUCTION

Software Defined Radio (SDR) is an emergent technology that aims at being the hardware support of future smart radios that have the capability to change their configuration according to the environmental conditions [1]. The main feature of an SDR is its flexibility and fast prototyping capabilities from an high-level description. Thus, one of the mainstream approaches to specify an SDR platform consists in implementing the processing on Digital Signal Processors (DSP) [2]. However, DSP also suffers from important power consumption and limited performance as compared to specialized hardware fabrics. Thus, Field Programmable Gate Array (FPGA) turned out to be an interesting alternative to DSP. FPGA-based SDR is an old paradigm [3] offering a good tradeoff between reconfiguration capability and power consumption.

Our study relies on a novel design flow for the rapid prototyping of FPGA-based SDR applications [4]. This flow leverages High-Level Synthesis (HLS) principles and tools to generate Register-Transfer Level (RTL) description from high-level specifications. Its entry point is a Domain-Specific Language (DSL) [5] which partly handles the complexity of programming an FPGA. It also considers heterogeneous description of signal processing IPs (Intellectual Properties) and integrates SDR features.

Based on this flow, this paper addresses the rapid prototyping of various designs of a given waveform, an architecture exploration is performed showing the flexibility feature of such a design flow. Indeed, HLS tools and their associated compilers give a special emphasis to the timing, area or throughput constraints and make it easy to explore a set of solutions via Design Space Exploration (DSE) [6][7] considering a given architecture.

The main contributions of this article are:
- To insert compile-time flexibility feature in the design flow.
- To analyze throughput/area tradeoff using HLS tools.
- To explore different designs for two useful standards: IEEE 802.15.4 and IEEE 802.11g.

The paper is organized as follows. A discussion over related works is given in Section II. The FPGA-based SDR design flow is introduced in Section III. Section IV details the DSE based on the throughput/area tradeoff and the implementation of the flexibility capabilities in the design flow. A DSE validation is given in Section V for architectures based on IEEE 802.15.4 and IEEE 802.11g standards. Finally, conclusions and perspectives are drawn in Section VI.

II. RELATED WORKS

Several proposals attempted to meet the flexibility requirements of an SDR by using software-based approaches. Indeed, software gives an abstraction level that enables more control over the hardware design flow. Two complementary approaches have been proposed namely, the SDR-specific languages to design the waveform [8][9][10] and the SDR middleware to provide the building environment [11][12]. They both take advantage of the abstraction level given by the software.

Our proposal aims at keeping a higher level of specification while addressing FPGA-based SDR. To this end, HLS turns out to be a good candidate to achieve such a high level of abstraction and a specific language has been proposed to describe the waveform. Just like in these related works, the compile-time flexibility is achieved by the fast prototyping capability which is enabled in our proposal by the HLS.

III. INTRODUCTION TO FPGA-SDR DESIGN FLOW

The design flow used in this work is shown in Fig. 1. The flow relies on three main steps: a Domain Specific Language (DSL) for the high-level waveform description, the Waveform Compiler that converts the high-level description to an RTL description and finally the Platform Integration step to program the design on a dedicated hardware platform. A DSL is a language that, through its syntax, remains specific.
Fig. 1. FPGA-SDR design flow.

Fig. 2. Simplified design constraint management scheme based on throughput/area tradeoff.

Fig. 2. Simplified design constraint management scheme based on throughput/area tradeoff.

Fig. 2. Simplified design constraint management scheme based on throughput/area tradeoff.

Fig. 2. Simplified design constraint management scheme based on throughput/area tradeoff.

Fig. 2. Simplified design constraint management scheme based on throughput/area tradeoff.

Fig. 2. Simplified design constraint management scheme based on throughput/area tradeoff.

Fig. 2. Simplified design constraint management scheme based on throughput/area tradeoff.
The Waveform Compiler of the DSL generates .tcl scripts to control the CatapultC software. Using .tcl will help for rapid prototyping with the automation of the transformation.

B. Processing Block Generation

The objective is to study the effect of the optimization tools provided by CatapultC. To illustrate and test the RTL generation of a processing block, we use the correlation bench function (named CorrBench) of an IEEE 802.15.4 Zigbee receiver. The full receiver is detailed in Fig. 7. This CorrBench function is a set of 16 FIR (Finite Impulse Response) filters with 16 taps, each filter performing the correlation between the received data and a code. A filter consists of 2 loops:

- **Shift** that shifts the data of a FIFO register and buffers the new input data,
- **MAC** that performs a multiplication and accumulation to compute the output of the filter.

This function is optimized using the CatapultC tools and the resource estimation of the resulting RTL designs are estimated using both CatapultC area estimation and ISE complexity estimation.

Design Exploration using HLS Tool

Several optimizations are made available on a typical HLS tool [13]. These tools make it possible to optimize the design at the area or throughput level. Two of the design optimizations from CatapultC that we use are loop pipelining and loop unrolling. **Loop pipelining** provides a way to increase the throughput of a loop (or decreasing its overall latency) by initiating the (i+1)-th iteration of the loop before the i-th iteration has completed. The number of cycles between iterations of the loop is called the initiation interval (II). The less the II is the more the pipeline is. **Loop unrolling** reduces the total loop iterations by duplicating (with a factor U) the loop bodies. The number of loop iterations is then reduced but care must be taken to the data dependencies when using this technique. Loop unrolling impacts the design latency and consequently the throughput.

**Area Estimation using HLS Tool**

The objective of the area estimation is to check the influence of loop optimization parameters. This will help the allocation algorithm to set these parameters according to the design constraints. This function is optimized using the CatapultC tools and the resource estimation of the resulting RTL designs are estimated using both CatapultC area estimation and ISE complexity estimation.

**Resource Estimation using Synthesis Tool**

The fine estimation is computed using ISE synthesis tool targeting a Virtex 6 FPGA. Synthesis results are detailed in Fig. 4 giving the estimation of the Look-Up Table (LUT), Flip-Flop (FF) registers and slices utilization. Results show that the number of LUT increases with the pipeline and the unrolling factor (U). Pipelining the design increases the number of FF registers while the loop unrolling has no influence on it. The link between the loop optimizations and the number of slices is more complex to draw. Indeed, the Virtex 6 slices are composed of 4 LUT and 8 FF registers and the ISE synthesis optimizes the slices interconnections and their occupations. It is possible to note that the LUT estimation follows the CatapultC area estimation as a reference.

C. Implementation in the FPGA-SDR design flow

The proposed design exploration is integrated in the proposed design flow in order to achieve a compile-time reconfiguration of the waveforms. To this end, willing to include flexibility requirements, a flexible block is declared with the key word **adaptive** in the DSL source code. This declaration highlights in the description the flexible nature of the block and enables compile-time flexibility through the DSE feature that can be afforded through the HLS tools as previously shown. DSE allows to optimize away a given architecture depending on some throughput, area or energy constraints. The key word **constraint** is used to set the constraints.

As an illustration, Fig. 5 introduces the DSL specification of the CorrBench block, it is declared **adaptive** with a throughput constraint. Thus, different solutions of the block could be quickly explored by tuning the constraints.
The waveform compiler has been defined to parse the DSL-based specification and generate all the required artifacts and source code. For example, in the description of CorrBench block, the key word processing is used to specify the field of the IEEE 802.15.4 frame that the block is intended to process. This information is parsed to generate the control unit.

The proposed methodology interacts with several synthesis tools thereby, it leverages the .tcl scripting language as an entry point toward these tools. As all operations are commands, it allows automation and flexibility in the use of applications. In the case of HLS software, it helps for rapid prototyping. Thus, the compiler generates .tcl scripts to interact essentially with the HLS tools and the Xilinx Synthesis tools. An example of such a .tcl file is detailed in Fig. 6. It controls the CatapultC software and performs the generation of a VHDL file of the desired CorrBench function.

To deal with the Design Constraint, the compiler has included some commands into the .tcl files in order set the desired loop optimizations. In Fig. 6, the MAC loop of the CorrBench function is unrolled by 5 and the 2 loops (Shift and MAC) are pipelined with II=1.

V. DESIGN SPACE EXPLORATION OF RADIO TRANSCEIVER ARCHITECTURES

FPGA-SDR being the overall objective of the study, two waveforms are available in the flow, namely the IEEE 802.15.4 and IEEE 802.11g transceivers. This section details the DSE of both the IEEE 802.15.4 receiver and the IEEE 802.11g transmitter and the experimentation on the Nutaq platform [14] as a validation of the design flow.

A. DSE of an IEEE 802.15.4 receiver

The DSE has been validated on a complete IEEE 802.15.4 receiver [15]. Fig. 7 describes the different blocks of the receiver. Each block is associated with an IP file from the library that will be used for the RTL merging step:
- 3 blocks (Delay, Decim, Demod) are designed with a hand-coded RTL IP without DSE (due to their small size),
- 2 blocks (RxFilter, Synchro) are chosen from a set of 3 hand-coded RTL IPs allowing DSE,
- 1 block (CorrBench) is generated using the HLS flow with a .tcl script instructions performing the DSE.

We explored manually some designs that could be achieved by combining the available IPs and the proposed design flow. The exploration is performed manually in the sense that the design constraints are set in the DSL, but each RTL description is generated automatically. Synthesis results are detailed in Fig. 8 giving the estimation of the LUT, FF registers and slices utilization for 18 designs. The CorrBench block is the main knob and 6 designs per CorrBench design have been generated by tuning parameters of the RxFilter and Synchro blocks.

The LUT and FF estimation shows the exploration that can be reached. However, some trade-off consideration has to be done. For example, the throughput of the CorrBench block is declared in Fig. 6 given a delay of 5 cycles.

V. DESIGN SPACE EXPLORATION OF RADIO TRANSCEIVER ARCHITECTURES

FPGA-SDR being the overall objective of the study, two waveforms are available in the flow, namely the IEEE 802.15.4 and IEEE 802.11g transceivers. This section details the DSE of both the IEEE 802.15.4 receiver and the IEEE 802.11g transmitter and the experimentation on the Nutaq platform [14] as a validation of the design flow.

A. DSE of an IEEE 802.15.4 receiver

The DSE has been validated on a complete IEEE 802.15.4 receiver [15]. Fig. 7 describes the different blocks of the receiver. Each block is associated with an IP file from the library that will be used for the RTL merging step:
- 3 blocks (Delay, Decim, Demod) are designed with a hand-coded RTL IP without DSE (due to their small size),
- 2 blocks (RxFilter, Synchro) are chosen from a set of 3 hand-coded RTL IPs allowing DSE,
- 1 block (CorrBench) is generated using the HLS flow with a .tcl script instructions performing the DSE.

We explored manually some designs that could be achieved by combining the available IPs and the proposed design flow. The exploration is performed manually in the sense that the design constraints are set in the DSL, but each RTL description is generated automatically. Synthesis results are detailed in Fig. 8 giving the estimation of the LUT, FF registers and slices utilization for 18 designs. The CorrBench block is the main knob and 6 designs per CorrBench design have been generated by tuning parameters of the RxFilter and Synchro blocks.

The LUT and FF estimation shows the exploration that can be reached. However, some trade-off consideration has to be done. For example, the throughput of the CorrBench block is declared in Fig. 6 given a delay of 5 cycles.
be achieved using the proposed design flow. The scale of the LUT number can vary from 1 to 1.7 and from 1 to 1.9 for the FF number. The link between the loop optimizations and the number of slices is more complex to draws as early explained.

B. DSE of an IEEE 801.11g transmitter

The DSE has been also been validated on a IEEE 802.11.g transmitter [16] based on OFDM modulation. The transmitter includes QAM symbols mapping, IFFT and cyclic prefix insertion. The IFFT is performed on 256 points and uses the radix-4 algorithm. CatapultC resource estimations are shown in Fig. 9 for 7 designs. The IFFT block being the main knob where DSE could be applied, only this block is optimized using HLS tools. Results show the area can be tuned with a factor of 2.7 while the throughput varies with a factor of 4.8.

C. Platform Experimentation

Thus, willing at validating the proposed design flow from high-level specifications to FPGA bitstream generation, we use the Nutaq Perseus 6010 [14]. It is based on an FPGA Virtex-6 device. In addition, a radio front end Radio420x daughter board is connected to the mother board to modulate or demodulate prior or after transmission in the channel. Feedbacks from the development on the Nutaq platform and initial implementation results were published in [17].

An IEEE 802.15.4 transmitter design and one of the explored IEEE 802.11g transmitter designs were prototyped using the FPGA-SDR design flow. Agilent VSA software is used in Fig. 10 to analyse the signal. It shows the received RF spectrum, the baseband waveforms and the demodulated constellation, thus validating the IEEE 802.15.4 waveform.

VI. CONCLUSIONS

This paper presents a Design Space Exploration of two architectures for implementing wireless physical-layer waveforms. The exploration relies upon a design flow that includes an high-level synthesis and its associated design optimizations. Results show the tradeoff that could be achieved between the throughput and the FPGA resource utilization. The design flow is managed by a Domain-Specific Language (DSL) which integrates SDR features and the constraints management. Future work is to include an algorithm in the DSL compiler that will allocate resources to the different blocks according to global specifications.

REFERENCES

Fig. 10. IEEE 802.15.4 received RF signal analysed in Agilent VSA.

Fig. 11. IEEE 802.11g transmitted RF signal analysed in Spectrum analyser.


