Multi-Purpose Systems: A Novel Dataflow-Based Generation and Mapping Strategy

Nicolas Siret, Matthieu Wipliez, Jean François Nezan, Francesca Palumbo, Luigi Raffo

Multi-Purpose Systems: A Novel Dataflow-Based Generation and Mapping Strategy

Nicolas Siret, Matthieu Wipliez and Jean-François Nezan
European university of Brittany, France
INSA, IETR, UMR 6164, F-35708 RENNES
Email: name.surname@insa-rennes.fr

Francesca Palumbo and Luigi Raffo
DIEE - University of Cagliari, Italy
Piazza D'Armi 09123 Cagliari
Email: <francesca.palumbo; luigi>@diee.unica.it

***Abstract*— The Dataflow Process Networks (DPN) Model of Computation (MoC) has been used in different ways to improve time-to-market for complex multi-purpose systems. The development of such systems presents mainly two problems: (1) the manual creation of the multi-purpose specialized hardware infrastructures is quite error-prone and may take a lot of time for debugging; (2) the more hardware are the details to be handled the greater the effort required to define an optimized components library. This paper tackles both problems, leveraging on the combination of the DPN MoC with a coarse-grained reconfigurable approach to hardware design and on the exploitation of the DPN MoC for the synthesis of target-independent hardware codes. Combining two state of the art tools, namely the Multi-Dataflow Composer tool and the Open RVC-CAL Compiler, we propose a novel dataflow-based design flow that provide a considerable on-chip area saving targeting both FPGAs and ASICs.**

I. Introduction.

The embedded systems world in the last few years has changed: new systems are required to be more flexible with respect to their predecessors. Standards in video codecs and telecommunication infrastructures are constantly evolving and applications are getting more complex every day. These requirements led to the development of complex, heterogeneous, multi-purpose systems. An optimal solution to target them is offered by the reconfigurable hardware design paradigm, able to guarantee: (1) flexibility (2) high performance (through native specialization capabilities) and (3) possible long-term adaptability.

One of the most critical aspect with such highly specialized platforms turned out to be the compilation toolchain: dealing with multi-purpose systems, the definition of the proper bit-streams set for platform configuration is not trivial [1], [2]. Moreover, the more is the specialization the more can be the time required to develop and debug the complex and heterogeneous Funtional Units (FUs) that may compose the overall platform.

In this paper we aim at addressing the above-mentioned issues to decrease the time-to-market of modern multi-purpose computational intensive systems, leveraging mainly on the Dataflow Process Networks (DPN) Model of Computation (MoC). The main novelty of this research resides in proposing a novel design flow for multi-purpose reconfigurable systems, exploiting two state of the art tools: the Multi-Dataflow Composer, MDC tool, and the Open RVC-CAL Compiler, Orcc. Coarse-grained runtime reconfiguration (CGRR) and the DPN MoC have been combined to address the tricky problem of assembling, in a reasonable time and without any user intervention, specialized and flexible CGRR platforms automatically mapping on them complex software applications.

The rest of this paper is organized as follow: Sect. II provides an overview of the MDC tool and Orcc, together with the description of the proposed dataflow-based design paradigm, Sect. III discusses the achieved performance and Sect. IV concludes with some final remarks.

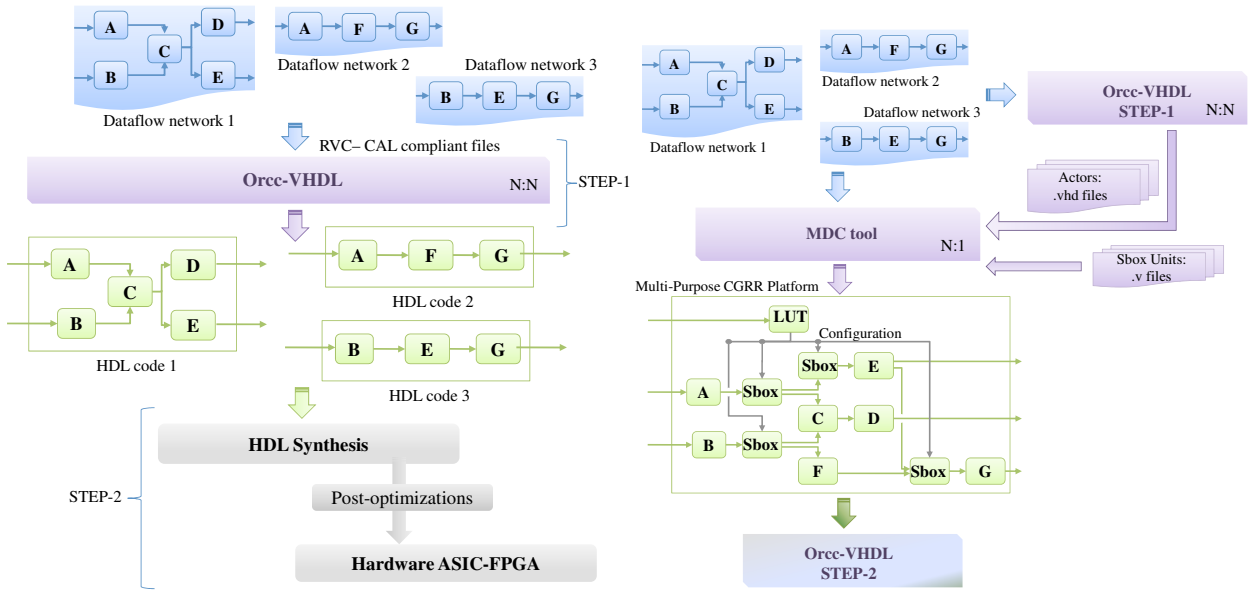
II. Dataflow-Based Design Flow

The main goal of the proposed novel design flow is the reduction of the time-to-market for the design of complex multi-purpose reconfigurable systems, combining together two approaches based on the DPN MoC: the MDC and Orcc.

A. The Multi-Dataflow Composer tool

The Multi-Dataflow Composer tool exploits the composability property of the DPN models to enable high level dataflow descriptions to be mapped directly on hardware. Originally conceived to be compliant to MPEG Reconfigurable Video Coding (RVC) [3] applications, it was then extended to any RVC compliant application [4], [5].

The MDC analyses input dataflows, recognizes similarities among them and assembles a unique HDL datapath where the common FUs are shared, through the intervention of dedicated data switching units (the Sbox units in Fig.1b). The output of the MDC is the multi-purpose CGRR platform implementing all the given input networks. The Sbox units allow on-the-fly switching among the implemented dataflows without any low-level reconfiguration. This reconfiguration mechanism, provides an efficient and quick substrate reconfiguration and allows the multi-purpose system to switch in a single clock cycle among the dataflows, without any shut-down overhead. Moreover, the produced CGRR platform is completely target-independent.



(a) Orcc-VHDL Synthesis Flow. (b) Dataflow-Based Novel Design Flow.

Fig. 1. Summary of Design Flows.

B. The Open RVC-CAL Compiler

Orcc was initially conceived to produce software code from dataflow programs in the RVC domain [6], [7]. It has been recently extended, to implement an Electronics System Level Design (ESLD) framework, by adding the High Level Synthesis (HLS).

The ESLD is a recent methodology which aims at improving the productivity of system architects by offering an efficient support of hardware/software co-designing, and providing HLS [8]. In one word, HLS tools compile high-level of abstraction programs into synthesizable hardware codes. Thereby, Orcc (namely Orcc-VHDL for the hardware code generation) generates a target-independent hardware code [9] which can be simulated and synthesized on any kind of hardware platforms.

In our approach of the HLS, a dataflow program is compiled in hardware code in two-steps, depicted in Fig 1a. First the VHDL code is generated using Orcc-VHDL, then it is synthesized for a specific target using any state of the art HDL-to-gates synthesizers. The main advantages of such two-steps approach are: (1) the generated VHDL is understandable, updatable and portable, and, (2) the design flow is closer to the one designers are accustomed to.

C. Novel Design Flow

Fig.1b presents the novel dataflow-based design toolchain proposed in this paper. By coupling the MDC tool and Orcc-VHDL to process a common set of RVC-CAL compliant networks, it is possible (without users intervention) to: (1) automatically derive the HDL CGRR platform of the multi-purpose system (MDC), (2) to generate the hardware library of FUs (Orcc-VHDL) and (3) to handle the on-the-fly runtime programmability (MDC).

Since both the MDC tool and Orcc-VHDL generate target-independent hardware code, the provided RTL description is compliant with any (recent) synthesizing tools. Therefore, runtime reconfiguration on the given coarse-grained substrate can take place both on ASICs and on FPGAs with the following benefits:

- ASIC CGRR provides high performance and gains in flexibility.
- FPGA CGRR allows resource sharing, so that a smaller FPGA may be used, while reducing power consumption and avoiding the overhead of a partial/complete context switch.

The availability of a complete composition and generation framework for multi-purpose heterogeneous systems based on a common formalism, constituted by the DPN MoC and the RVC-CAL standard, enables to move some steps towards the possibility of closing the gap between hardware and software, natively addressing the tricky problem of automatically mapping software on highly specialized heterogeneous hardware. Moreover, this approach has to be considered orthogonal to different reference domains. In fact, by simply providing the applications according to the reference formalism, any multi-flow system can be synthesized.

D. Comparison with Related Works

Other automated tools have been developed to automatically map applications on a coarse-grained reconfigurable system. Nevertheless, as far as we know none of them is able to automatically derive the library of FUs as long as

the CGRR platform from high-level programs descriptions, as the proposed approach does.

Kim et al. in [10] succeed in executing different applications on a fixed coarse-grained tile-based architecture composed of generic Processing Elements, rather the proposed framework allows the synthesis of a library of actor-specific optimized FUs as long as the definition of a customized application-specific dataflow-driven platform. Ogrenci et al. in [11] present a target-dependent approach, where no automatic derivation of the overall system is provided. Angermeier et al. in [12] propose a dataflow-oriented approach to reconfigurable systems design specifically addressing Partial Reconfiguration on FPGAs. This approach is target specific and FUs are not automatically derived.

III. Performance Assessment

This section presents the results obtained adopting the proposed dataflow-based design flow on two different multi-standard platforms: the first one running a very simple 1-D Inverse Discrete Cosine Transform (*IDCT1D*), while the second a 2-D Inverse Discrete Cosine Transform (*IDCT2D*). The *IDCT1D* is a static dataflow network and a reference application in signal and image processing. The *IDCT2D* isa hierarchical dynamic dataflow network that executes two IDCT1D and transposes the signal between the two executions.

A. Performance on the IDCT1D

On the top of Fig. 2 two different *IDCT1D* dataflow models are presented, whereas on the bottom their combination. The original *IDCT1D* dataflow networks share just one single actor; therefore, the benefits of assembling a CGRR platform are not straightforward. However, this use case is sufficient to show the validity the approach with respect to the one cycle runtime reconfiguration. Moreover, despite the number of inserted Sbox units (7) is bigger than the number of FUs (1) placed in common, both on a Xilinx Virtex6 and an Altera Spartan3 FPGAs it was possible to save around the 5% of area.

B. Performance on the IDCT2D

To implement a multi-standard version of the *IDCT2D* we have decided to assemble a CGRR architecture (CGRRA) able use alternatively both the *IDCT1D* models presented on the top of Fig. 2. This platform, for exploration purposes, has been compared to a static dual mode architecture (SDMA). Table I summarizes the overall number of involved FUs, as long as the overheads required for the SDMA and for the CGRRA.

Table II summarizes the performance obtained using the Altera Quartus II and the Xilinx ISE synthesisers for both the CGRRA and the SDMA. The increased complexity due to Sbox units insertion, when FUs are shared, leads to a very small penalty in terms of operating frequency, while area benefits are very promising: nearly 31% fewer slices and, respectively, 28% (Quartus) and 23% (ISE) fewer LUTs.

Model	FUs	Sboxes	LUTs
IDCT 2D model 1	11	0	0
IDCT 2D model 2	12	0	0
SMDA	23	0	0
CGRA	15	7	1

TABLE I
IDCT2D Use-Case: Required Hardware Components.

	SDMA	CGRA	Benefit	Synth.
Max. Frequency [MHz]	119	124	+4,2	Quartus II
Slices [#]	2242	1537	-31,5	
LUTs [#]	3178	2278	-28,3	
Power Dissip. [mW]	420	408	-2,7	
Max. Frequency [MHz]	116	114	+1,7	Xilinx ISE
Slices [#]	2462	1666	-32,3	
LUTs [#]	2465	1894	-12,2	
Power Dissip. [mW]	409	401	-2	

TABLE II
IDCT2D Use-Case: Synthesis on FPGAs.

For ASIC synthesis we have adopted a 90nm CMOS technology. In Fig.3 the variation of the area of the CGRRA and the SDMA as the operating frequency is changed is shown. The longer critical paths, as in the FPGA exploration, led to a small frequency penalty (3% less in the CGRRA case). On the contrary, the benefit in terms of area is very promising (nearly the 40%).

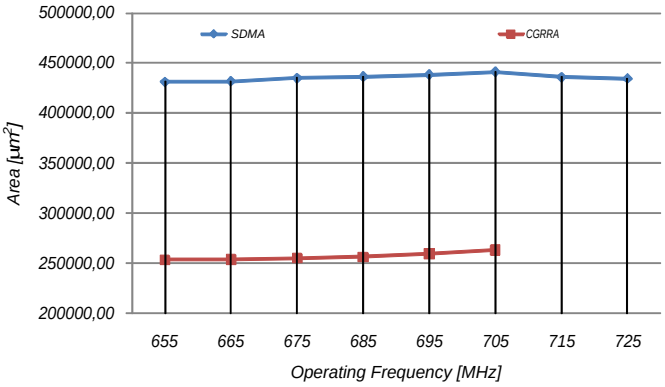


Fig. 3. Implementation results on a 90nm technology.

IV. Conclusion and perspectives

In this paper we have proposed a novel design flow whose aim is two-fold. First, it contributes to close the gap between hardware specification and software development by automatically mapping complex software application on a given hardware platform. The proposed flow also

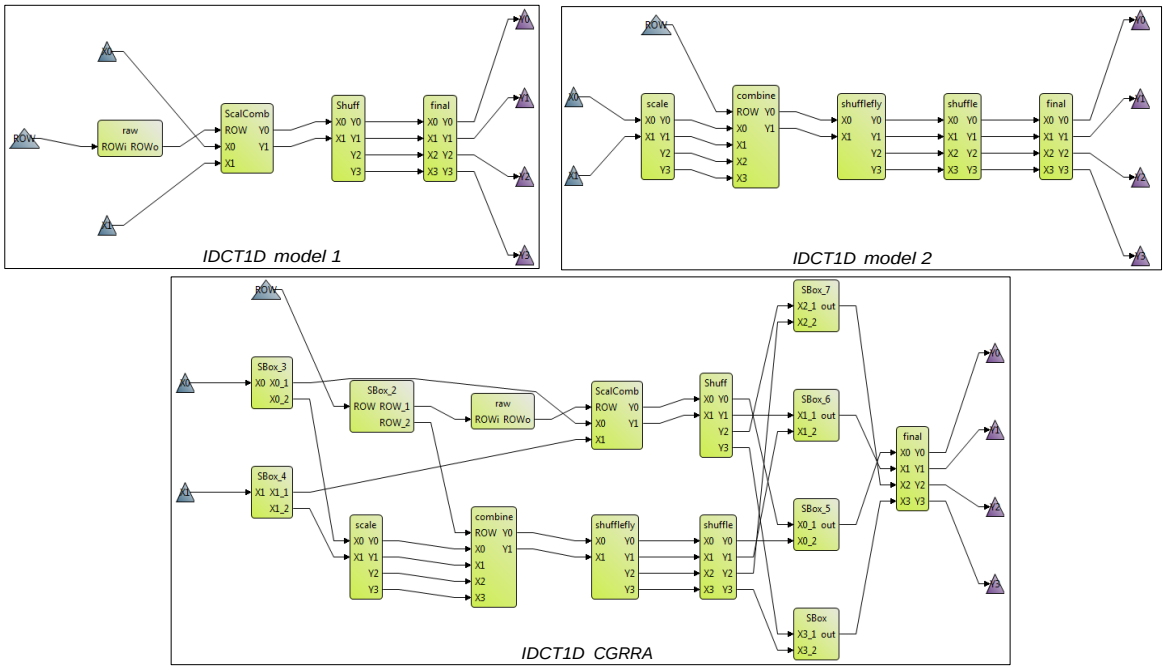


Fig. 2. The *IDCT1D* Use-Case.

fulfils integration and specialization needs required by modern embedded systems design by leveraging on a reconfigurable hardware substrate.

These goals are ensured by the combination of the Orcc-VHDL and the MDC tool. These tools are respectively able to provide efficient code generation using HLS from RVC-CAL dataflow models to VHDL and to create a CGRR platform, extremely suitable for multi-purpose systems, relieving hardware developers from applications similarities analysis and composition.

The proposed design paradigm is completely target-independent. It allowed to achieve very promising results both on FPGAs and ASICs targets, where respectively the 30% and the 40% of area saving has been reached, at a negligible frequency penalty. More importantly exploiting the DPN MoC and combining the MDC tool with the Orcc-VHDL no users' intervention are required in the design flow.

Research directions in this field will regard mainly two other important hot-topics in embedded systems design: power management (exploiting the Sbox-based reconfiguration mechanism) and design space exploration (allowing an exchange of informations among Orcc-VHDL and the MDC).

References

[1] S. Carta, S. Pani, and L. Raffo, "Reconfigurable Coprocessor for Multimedia Application Domain," *Jorunal of VLSI Signal Processing Systems*, vol. 44, no. 1-2, pp. 135–152, 2006.

[2] V. Kumar and J. Lach, "Highly flexible multimode digital signal processing systems using adaptable components and controllers," *EURASIP J. Appl. Signal Process.*, vol. 2006, january.

[3] M. Mattavelli, I. Amer, and M. Raulet, "The Reconfigurable Video Coding Standard [Standards in a Nutshell]," *Signal Processing Magazine, IEEE*, vol. 27, no. 3, pp. 159 –167, May 2010.

[4] F. Palumbo, D. Pani, E. Manca, L. Raffo, M. Mattavelli, and G. Roquier, "RVC: A multi-decoder CAL Composer tool," in *IEEE international Conference on Design and Architectures for Signal and Image Processing (DASIP)*, 2010, pp. 144–151.

[5] F. Palumbo, N. Carta, and L. Raffo, "The multi-dataflow composer tool: A runtime reconfigurable hdl platform composer," in *to be published in the IEEE international Conference on Design and Architectures for Signal and Image Processing (DASIP)*, 2011.

[6] M. Wipliez, G. Roquier, and J.-F. Nezan, "Software Code Generation for the RVC-CAL Language," *Springer journal of Signal Processing Systems*, 2009.

[7] J. Gorin, M. Wipliez, J. Piat, F. Preteux, and M. Raulet, "An LLVM-based decoder for MPEG Reconfigurable Video Coding," in *IEEE international Workshop on Signal Processing Systems (SIPS)*, 2010.

[8] P. Coussy and A. Morawiec, *High-Level Synthesis: From Algorithm to Digital Circuit*. Springer, 2008.

[9] N. Siret, M. Wipliez, J. Nezan, and A. Rhatay, "Hardware code generation from dataflow programs," in *IEEE international Conference on Design and Architectures for Signal and Image Processing (DASIP)*, 2010, pp. 113 –120.

[10] K. Yoonjin and R. Mahapatra, "Design Space Exploration for Efficient Resource Utilization in Coarse-Grained Reconfigurable Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 10, pp. 1471–1482, oct. 2010.

[11] S. Ogrenci Memik, E. Bozorgzadeh, R. Kastner, and M. Sarrafzadeh, "SPS: A strategically programmable system," in *Proceedings of the Int. Parallel and Distributed Processing Symposium (IPDPS)*, apr 2001.

[12] J. Angermeier, S. Wildermann, E. Sibirko, and J. Teich, "Placing Streaming Applications with Similarities on Dynamically Partially Reconfigurable Architectures," in *Reconfigurable Computing and FPGAs (ReConFig), 2010 International Conference on*, dec. 2010, pp. 91 –96.