Power-Awarness in Coarse-Grained Reconfigurable Designs: a Dataflow Based Strategy

Francesca Palumbo
University of Sassari
PolComIng Dept. - Information Engineering Unit
Email: fpalumbo@uniss.it

Carlo Sau and Luigi Raffo
University of Cagliari
DIEE Dept. Microelectronics and Bioengineering Lab
Email: carlo.sau@diee.unica.it and luigi@diee.unica.it

Abstract—**Applications and hardware complexity management in modern systems tend to collide with efficient resource and power balance. Therefore, dedicated and power-aware design frameworks are necessary to implement efficient multi-functional runtime reconfigurable signal processing platforms. In this work, we adopt dataflow specifications as a starting point to challenge power minimization.**

I. Introduction

The power reduction issue in modern embedded systems is challenging. It collides with the trend of having more and more complex heterogeneous systems, not favoring cutting down designer effort. Reconfigurable Video Coding (RVC) adopts dataflow-based techniques to challenge those issues. It exploits modularity to provide dynamic and incremental configuration and reconfiguration.

The methodology we have developed to cope with power management is related to the Multi-Dataflow Composer (MDC) tool [1], born within MPEG-RVC studies, but already adopted in different contexts of application [2]. MDC addresses automatic design of runtime reconfigurable coarse-grained heterogeneous platforms. It combines the dataflow high-level formalism and the coarse-grained reconfigurable approach to deploy multi-functional systems, featuring flexibility and area minimization. The definition of automated methodologies for power management is of primary importance to reduce debugging and deployment costs. Within MPEG-RVC, this is still an open issue. In this context, we present the combination of structural and dynamic strategies to master power reduction in coarse-grained reconfigurable architectures. At the structural level, the optimal system specification(s) (capable of maximizing performance while minimizing the implementation costs) are selected. Whereas, at the dynamic one, disjointed logic regions functionally homogeneous are identified. These latter can be turned on and off simultaneously by means of clock/power gating techniques. Preliminary synthesis results have been assessed on an image zoom application targeting an ASIC 90 nm technology.

The rest of this paper is organized as follows. Section II defines the scientific context of this work. Section III describes the proposed power management approach. Section IV discusses the achieved results, prior to conclude in Section V.

II. Background

In this section we are going to present the context of application we are referring to (Sec.II-A), the basic instrument (Sec.II-B) we have extended to provide power management and the foundations of the proposed techniques (Sect. II-C).

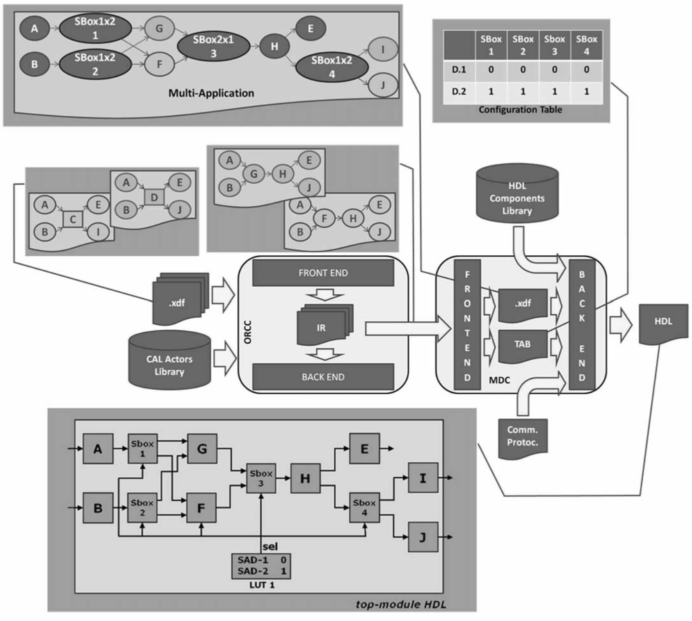
A. Dataflow and Reconfigurable Video Coding

A *dataflow program* is a directed graph where nodes represent computational units (*actors*), while edges represent loss-less, order-preserving point-to-point *connections* between actors, used to communicate sequences of data packets (*tokens*). In terms of notation, let's define $DFG\langle V,E\rangle$ as a directed graph, where V is the set of vertices of the graph (the actors) and E is the set of edges (the connections). Actors asynchronously concur to the computation and can be transformed either into software agents or into physical Functional Units (FUs). Communication among the actors is token mediated. Several variations of this kind of dataflow model have been introduced in literature ([3], [4]). In the MPEG-RVC context *Dataflow Process Networks* (DPNs) [4] with firing rules are used. DPN is extensively used due to its expressiveness and for the existence of a formal programming language, called CAL actor language, supporting it [5]. To support MPEG-RVC, since 2010 when it was standardized, several tools have been conceived, examples are Orcc, Xronos, Turnus etc.

B. MDC: Multi-Dataflow Network Composition

The Multi-Dataflow Composer (MDC) was conceived for the automatic creation and management of multi-functional systems. It was meant to address the difficulty of mapping different applications onto a coarse-grained reconfigurable architecture ([6], [7]). Its final goal ([1], [8], [9]) is to automate such a mapping process while minimizing hardware resources, with consequent area/energy saving. This issue is known in literature as the datapath merging problem. MDC solves it by exploiting a heuristic algorithm. Depending on the front-end it is connected to, MDC in principle is able to process any type of dataflow model. At the moment, as it can be seen in Fig. 1, MDC is coupled to Orcc [10]. The MDC front-end leverages the flattened Intermediate Representations generated by the Orcc front-end, to assemble a single multi-functional specification, starting from the XDF files of the given DPNs. MDC front-end also keeps trace of the system programmability through the *Configuration Table*. The MDC back-end then creates the respective HDL coarse-grained reconfigurable hardware, mapping each actor on a different FU. FUs are passed as input to the MDC within the HDL components library and can be manually or authomatically created [11]. FU granularity/functionality does not affect the described flow.

Reconfiguration, implemented in a single-cycle, is guaranteed by low overhead switching modules (Sboxes) placed at the crossroads between the different paths of data and driven by dedicated LUTs, which content is defined according to the *Configuration Table*. Sboxes are simply responsible of data routing, without computational overhead. In the current implementation, they are simple combinatorial multiplexers and they don't need FIFO buffers on the incoming communication channels. Thus the well known dataflow problem of the FIFO buffers optimal sizing does not affect the MDC merging process. Input DPNs have only to be properly sized, in terms of FIFO buffers, before the MDC execution.



	SBox 1	SBox 2	Sbox 3	SBox 4
D.1	0	0	0	0
D.2	1	1	1	1

Configuration Table

Fig. 1. MDC: overview and example.

C. Power-Awareness in Reconfigurable Architectures

In the *dark silicon* era [12], when not all the available resources on a die will be usable due to the limited power budget, power management strategies are of paramount importance. Moreover, the more the integration on a single die the more holistic power minimization approaches are required across the whole design stack [13]. The proposed power management strategy combines structural and dynamic aspects. The former are accounted by selecting power-efficient circuits at the topology level, the latter by automatically deriving different homogeneous logic areas. Leveraging on a dataflow-based design approach, the techniques we are presenting act both at a high-level and are automatically applied.

1) Structural Power Managment: Within MPEG-RVC, a first tentative approach to structural power management has been targeted in [14] through the analysis of the causation trace of a DPN by applying then multi-clocking strategies. This technique has never been applied to reconfigurable systems. In [15] an energy estimation methodology, based on Performance Monitor Counters (PMC), has been proposed to estimate the energy consumption of RVC-CAL video codec specifications. This approach addresses software solutions, but PMC-based energy estimation methodologies are extensible to hardware-based RVC tools. Software frameworks and simulators have been proposed at the state of the art to account for structural power-managment and design space exploration (DSE). Within MPEG-RVC, DSE techniques have been used for actors refactoring to improve throughput [16], without targeting coarse-grained systems nor power-minimization.

2) Dynamic Power Managment: Power consumption in digital devices is composed mainly of two different contributions: dynamic and static. The former is due to capacitance charging/discharging when logic transitions occur (i.e. switching activity). The latter is due to leakage currents and it is consumed while no circuit activity is present. Modern designers need to cope with both terms and are required to conceive smart management strategies that, at the circuit level, will be capable of limiting the power consumption, acting either on the clock tree or on the voltage supply. In both cases, the identification of homogeneous logic regions within the system is fundamental, identifying different hardware blocks that can be turned on and off together.

III. Dataflow-Based Power Awareness

In this section we are going to present the early-stage power management strategy we have envisioned, now embedded in MDC. We have been working on two different extensions. The *Topology Definer* (Sect. III-A) acts at the structural topology level by defining the optimal system configuration(s) minimizing the implementation costs. The *Logic Set Definer* (Sect. III-B) acts at the dynamic level, identifying disjointed logic regions to be physically switched off when unused.

A. Structural Level: Profiling-Aware Topology Definition

To understand the need of structural power management for coarse-grained reconfigurable architectures you should consider that:

- in a multi-functional environment not always the solution merging all the DPNs together is power-optimal. The overhead of the switching elements may overcome the benefits of actors minimization.
- in the adopted iterative algorithm, processing two DPNs at a time, the merging order may impact on the operating frequency. You can end up with longer critical paths.

Therefore, it is fundamental to determine the (sub-)optimal design specification(s). The topology definer coupled to MDC, as demonstrated in [17], is capable of determining the design costs of different implementations, before prototyping, back annotating low level information on the DFGs. Figure 2 depicts the implemented profiling-aware topology strategy. Here follows the applied steps.

Ordering and Sequences Extraction: The *Sequences Generator* defines all the D possible DPNs sequences MDC will be fed

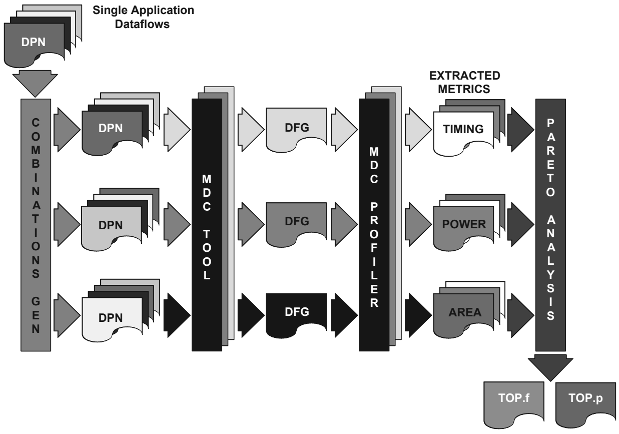


Fig. 2. Topology Definer Flow.

with according to the following equation:

$$\begin{aligned}
&D = D\_notMer + D\_Mer + D\_partMer = \\
&D = 1 + N! + \sum_{k=1}^{N-2} C_{N,k} * (N-k)! \\
&D = 1 + N! + \sum_{k=1}^{N-2} \frac{N!}{(N-k)! * k!} * (N-k)! \\
&D = 1 + N! + \sum_{k=1}^{N-2} \frac{N!}{k!}
\end{aligned} \tag{1}$$

where *D_notMer* is the *static*, not merged, composition of the *N* DPNs in parallel; *D_Mer* is the *all merged* term that, maximizing resource sharing, is given by all the possible permutations of the DPNs; *D_partMer* is the *partially-merged* term that, not following the resource maximization principle, provides all the sequences composed of the combinations[1] that can be extracted from a subset of *k* input DPNs placed in parallel with all the merged permutations of the other *N-k* networks.

Multi-Dataflow Specification Definition: For each sequence the MDC tool extracts the DFG (Sect. II-B).

Profiling: For each DFG, implementation costs are computed. Having already back-annotated the HDL components library with one value of estimated area and power consumption[2] per FU, the *MDC Profiler* retrieves $\forall v_i \in V$ the correspondent back-annotated values, a_i and p_i, and sum them up to estimate respectively area and power consumption of each DFG.

Operating frequency estimation is less straightforward. As already said, different feeding orders may result in different cascades of Sboxes that may negatively impact on the critical path (CP). The *MDC Profiler* $\forall n_i \in InN$ (being InN the set of input DPNs) retrieves the correspondent back-annotated CP[3], CP_i, and defines $CPstatic = max(CP_i)$ as the CP of the static system configuration. Then it estimates the longest cascade of Sboxes ($seqSB$) within the considered DFG. Given N_S as the number of Sboxes composing $seqSB$ and cpS_i as the back-annotated CP associated to the $i-th$ Sbox within the HDL components library, the possible CP due to the cascade of Sboxes is given by:

$$CP\_seqSB = \sum_{i=1}^{N_S} cpS_i \tag{2}$$

The *MDC Profiler* finally compares $CPstatic$ and CP_seqSB: the maximum of these two values determines the maximum operating frequency of the given design point.

Pareto Analysis: The Pareto-based analysis is carried out exhaustively on the entire design space to determine the optimal system configuration(s) according to the selected design effort. For power management purposes it will be extremely important to determine the least consuming configuration, but minimizing power consumption not necessarily implies having also the best operating frequency. Therefore, two sub-optimal $DFGs$ are provided as output: the area/power (*TOP.p*) and the frequency (*TOP.f*) one.

B. Dynamic Level: Early-Stage Homogeneous Logic Regions Definition

To understand the need of dynamic power management for coarse-grained reconfigurable architectures you should consider that:

- within a multi-functional architecture, when you execute an application, the rest of the design is idleing;
- the part of unused resources is variable, it does not depend on the number of input specifications, but it is fixed as soon as the architecture is deployed.

According to these considerations, it was possible to conceive an algorithm that, working at the specification level, it is capable of automatically identifying disjointed logic regions. Actors active/inactive together are grouped within homogeneous logic sets. At the moment, as a proof of concept for the algorithm, despite more advanced power management strategies can be applied to the different logic regions, a simple coarse-grained clock gating procedure has been adopted. Therefore, the different identified regions take the name of Clock-Gated Sets (CGSs). Clock gating acts at the clock net level (responsible for more than the 50% of the dissipation [18]). On the MDC GUI, users can choose enabling or not powering down policies and, in the former case, they are required to specify the final target device. Different targets require different physical solutions. At the moment, both AND gates cells (applied directly on the clock to disable it) for ASIC designs and BUFGCE cells (clock network cannot be modified by the insertion of any custom logic) for FPGA implementations on Xilinx boards are featured by MDC. Here follows the algorithm steps.

Identification: At first, the minimum ideal number of CGSs is identified. This first optimization phase is performed regardless the chosen target, to minimize on-chip redundancy. For any input DPN, its set of actors is compared with the already identified CGSs and three different situations may occur. The current set is disjointed with respect to the previous CGSs: a

[1] A combination is a selection of all or part of a set of objects, regardless to the order in which they are selected. Given A, B and C, the complete list of possible selections would be: AB, AC, and BC.

[2] Synthesis trials with the RTL Compiler of Cadence SoC Encounter. 90 nm CMOS technology.

[3] SoC Encounter has been used to extract the CP associated to the *N* different input specifications synthesized stand-alone. 90 nm CMOS technology.

TABLE I. COMPUTATIONAL KERNELS OF THE ZOOM APPLICATION.

kernel	# actors	# occ	data size	functionality
abs	1	3150	1	absolute value calculation
min_max	1	1050	2	maximum/minimum finding
chgb	7	3072	16x16	bilevel/grayscale block checking
median	9	1069	4	median calculation
cubic_conv	6	408	16x16	cubic filter convolution
cubic	10	1070	4	linear combination calculation
sbwlabel	17	2722	16x16	edge block checking

new CGS is issued along with the correspondent entry within the map of DPN pointers to the CGSs. The current set is already contained within one of the identified CGSs: just the entry on the map of DPN pointers to the pre-determined CGS is added. The current set intersects one of the identified CGSs: a new CGS, containing the intersected instances, is issued and the the pre-determined CGS is modified by removing the intersection.

Fusion: Dealing with FPGA technologies, it can happen that the amount of CGSs exceeds the number of allowed logic regions (limited by the amount of BUFGCE on the chosen board/device). Within this phase sub-optimal CGSs are determined: switching activity in unused FUs will be present. To determine the CGSs candidate for merging users can select two different cost functions: resource minimal (minimizing the number of FUs per CGSs) and power minimal (minimizing the estimated power budget per CGSs).

C. Step-by-step example

To clarify the different logic phases of the proposed methodology Fig. 3 depicts a step-by-step flow. It highlights the starting point of our dataflow-based power management strategy, 3 different DPN specifications (*Net1*, *Net2* and *Net3*), and the final output, a coarse-grained reconfigurable platform where different logic regions are provided. By applying Eq. 1 13 points constitute the design space as possible system specifications. Among these, the *Topology Definer* is capable of identifying the power-optimal and the frequency-optimal ones. The former, *TOP.p*, is an *all-merged* solution (the optimal feeding order is *Net1*, *Net3* and *Net2*), while the latter, *TOP.f*, is a *partially-merged* one (*Net1* is in parallel to the merged *Net2* and *Net3*). On the bottom of Fig. 3 the *Logic Set Definer* processes *TOP.p*. Given the intersections among the actor sets of the different input specifications, 5 different CGSs are identified so that 5 clock-gating cells are instantiated: *Net1* activates CGS1, CGS2 and CGS3, *Net2* activates CGS1 and CGS4 and, finally, *Net3* activates CGS3 and CGS5.

IV. EXPERIMENTAL PROOF OF CONCEPTS

To validate the proposed power-aware design flow we have adopted MDC to assemble the computing core of an accelerator for an image zoom application. The application has been profiled to identify the most computationally intensive code segments (*computational kernels*). The seven identified kernels have been modeled as DPNs in terms of RVC-CAL dataflow specifications. Table I depicts an overview of the kernels composition, functions and occurrences within the zoom application. These seven input DPNs required to the *Topology Definer* the evaluation of a design space of 13693 points, which Pareto graphs are depicted in Fig. 4 and Fig. 5.

Power-optimal solution: It is possible to identify 3 different clusters in the graph: *z1* involving points with 0 to 4 merged specifications (MSs), *z2* with 2 to 6 MSs and *z3* with 5 to 7 MSs. According with the area/power graph, the optimal design points are *z3p1* and *z3p2*, where all the 7 input DPNs are merged. Looking at the frequency domain (Fig. 5) the optimal area/power points loose the 35% (*z3p1*) and the $27,8\%$ (*z3p2*) with respect to the maximum achievable frequency. The *Topology Definer* selects *z3p2* as *TOP.p*, presenting the best area/power value at the higher frequency.

Frequency-optimal solution: In the frequency domain five clusters are visible, not correspondent to the three area/power ones. The optimal configuration in this domain is *z2p1* where 5 over 7 input DPNs are merged. The kernels kept in parallel, *chgb* and *cubic_conv*, involve shareable actors with low area and power values. It turned out that, if they are not merged, the design frequency remains the maximum achievable one without big overheads of area and power. With respect to *TOP.p*, *z2p1* presents an overhead of 9.6% in area and 15.3% in power. The *Topology Definer* selects *z2p1* as *TOP.f*: the configuration with the highest frequency at the smallest power/area penalty.

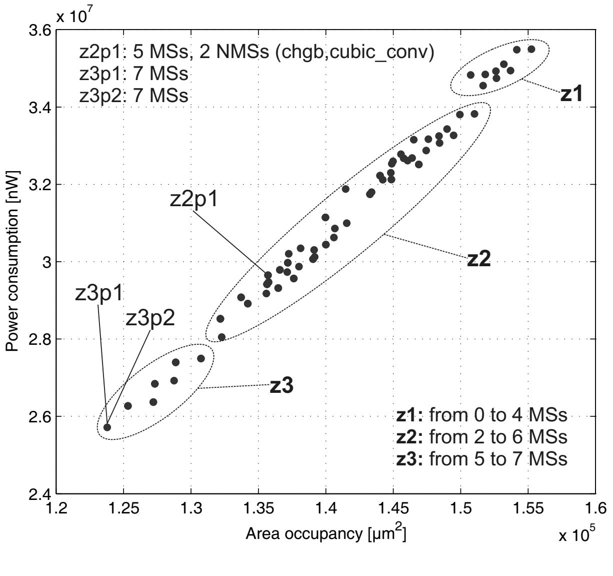


Fig. 4. Area vs Power. [MSs = Merged Specifications; NMSs = Not MS]

The effectiveness of the proposed dynamic power management strategy has been evaluated on the high-level multi-functional *TOP.f* and *TOP.p* specifications, considering 4 different platforms:

- *freq_nocg*: *TOP.f* without CGSs identification;
- *freq_cg*: *TOP.f* with CGSs identification;
- *pwr_nocg*: *TOP.p* without CGSs identification;
- *pwr_cg*: *TOP.p* with CGSs identification.

These designs have been synthesized using Cadence SoC Encounter on the same technology and at the same frequency, 250 MHz, used to carry out the structural power management process. The synthesis frequency is compliant with the

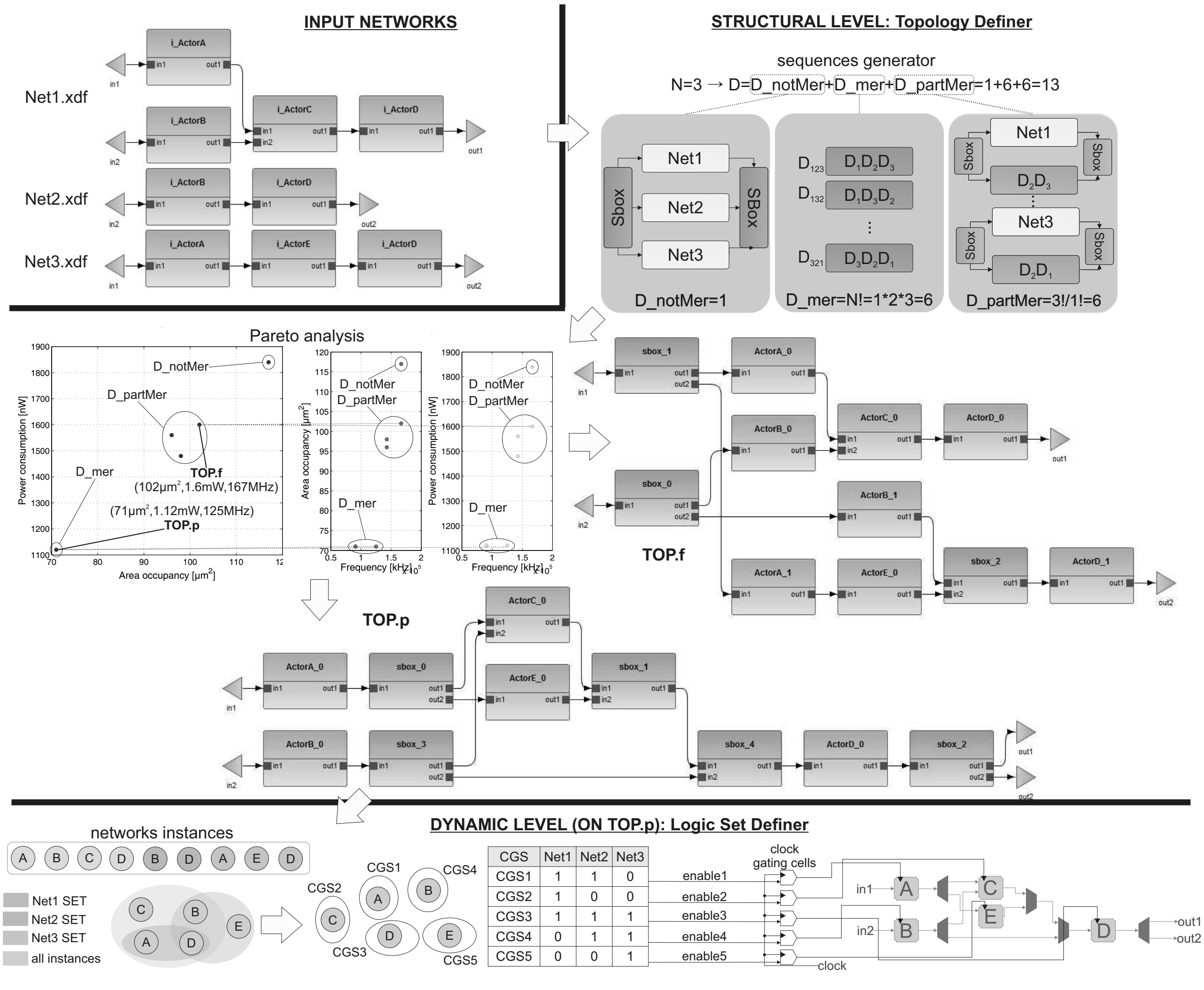


Fig. 3. Dataflow based power management strategy: example.

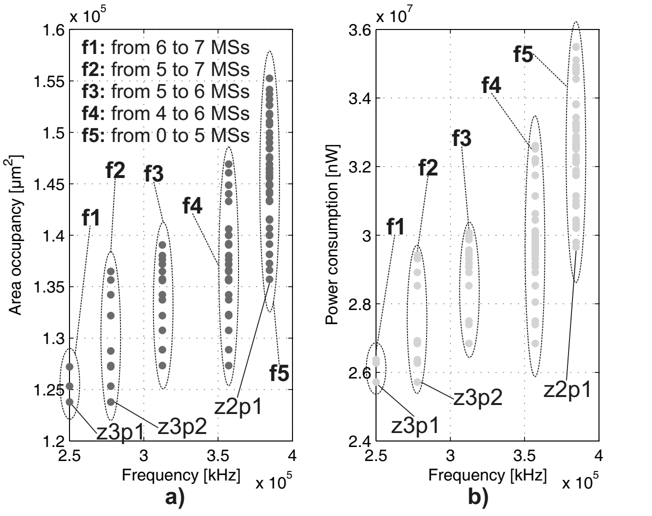


Fig. 5. Area vs. Frequency (a). Power vs. Frequency (b). [MSs = Merged Specifications.]

maximum frequency allowed for TOP.p, that is 277.78 MHz. The *Logic Set Definer* provided different results for *TOP.f* and *TOP.p*, due to their different compositions (respectively 5 and 7 merged networks). In particular, 9 CGSs are identified for *TOP.f* and 13 for *TOP.p*. On ASIC the CGSs merging process it is not necessary.

Table II summarizes the synthesis results in terms of area occupancy and power consumption. The *freq_nocg* design, as expected, occupies more area than *pwr_nocg*. The percentage of saving adopting the latter is 8.68%. Table II shows also clock-gating overhead: $0,19\%$ (*freq_nocg* vs. *freq_cg*) and $0,45\%$ (*pwr_nocg* vs. *pwr_cg*). The estimation error (*vs. est* rows in Tab. II) of the area occupancy is below 1% for both *TOP.f* and *TOP.p*. It may be caused by the overhead of the wires among the actors and the presence of the LUTs (controlling the Sboxes) not considered within the area/power models. Power results are evaluated in two ways. *Stat. Power* does not consider the switching activity due to the application execution. It aims at evaluating the profiling estimation accuracy. In this case the estimation error for *TOP.f* and *TOP.p* is respectively 1.77% and 0.45%. *Dyn. Power* is intended to evaluate the clock gating effectiveness. Estimations take into account the actual switching activity executing each kernel and are computed as the sum of the single kernel dissipation values

TABLE II. ASIC SYNTHESIS RESULTS (*est* = ESTIMATION GIVEN BY THE TOPOLOGY DEFINER; *auto* = SYNTHESIZER AUTOMATIC REGISTER LEVEL CLOCK GATED DESIGN).

	Area [%]	Stat. Power [%]	Dyn. Power [%]
freq nocg vs. est	+0.08	+1.77	-
freq cg vs. nocg	+0.19	-95.54	-74.86
freq cg vs. auto	+12.33	-95.15	-69.06
pwr nocg vs. est	+0.19	+0.45	-
pwr cg vs. nocg	+0.45	-92.57	-71.30
pwr cg vs. auto	+12.16	-89.18	-63.75
nocg pwr vs. freq	-8.68	-14.39	-12.75

weighted for their occurrences within the zoom execution (in Tab. I). Dynamic power management led to save respectively the $74,86\%$ and the $71,3\%$ of power in the *TOP.f* and the *TOP.p* designs. This impressive result is determined by the fact that clock gating techniques have been conceived with the specific intent of limiting power dissipation, reaching up to the 50% in literature. In multi-functional scenarios, where several FUs may be unused while executing a specific computation, the saving amount can be considerably higher.

Table II presents also the area and power estimations of the design obtained by *TOP.f* and *TOP.p* adopting the automatic register level clock gating feature provided by the synthesizer (*freq_auto* and *pwr_auto* respectively). The results show that the implemented strategy can achieve better savings than the synthesizer automatic one, in terms of power consumption. However the *freq_auto* and *pwr_auto* designs have an area occupancy that is more than 10% smaller than the area of the *cg* ones.

V. CONCLUSIONS

In this paper we addressed the problem of proving high-level power management while implementing coarse-grained platforms running different applications on shared resources. Heterogeneous runtime reconfigurable systems are deployed. Two are the basic assumptions of this work: *1)* power reduction is of paramount importance in modern battery dependent designs; *2)* systems heterogeneity and complexity is so challenging that manual strategies requires too much effort to be considered still viable.

The proposed two-step approach extends a dataflow-based design framework for coarse-grained reconfigurable designs, the Multi-Dataflow Composer tool, conceived within MPEG-RVC research studies. The importance of the proposed power-aware extension of the MDC tool is mainly related to the fact that, within the MPEG-RVC framework, power consumption is still an open issue and, as far as we know, early-stage power estimation/management strategies are currently under investigation. Our solution combines structural and dynamic strategies applied automatically at the modelling level. On the ASIC implementation of the computing core of an accelerator for an image zoom application, we have demonstrated how topology may affect power in coarse-grained architectures. In the considered scenario it is possible to adopt a graph-based strategy to lower also dynamic power consumption, up to the 70%. These benefits are appreciable also with respect to the synthesizer automatic clock gating feature.

The implemented strategies are currently under evaluation in a multi-decoder scenario [19] and future developments will regard both structural and dynamic levels. On the one hand, we are working on the definition of accurate analytic models (to include technology-awareness in the early stage model-based system characterization) and on heuristic algorithms to reduce the overall DSE time by computing an approximated Pareto set of configurations. On the other hand, we intend to extend MDC to provide automatic power-gating.

ACKNOWLEDGMENT

Prof. Luigi Raffo and Dr. Carlo Sau are grateful to Sardinia Regional Government for funding the RPCT Project (L.R. 7/2007, CRP-18324) that led to these results. Dr. Sau is also grateful to Sardinia Regional Government for supporting his PhD scholarship (P.O.R. F.S.E., European Social Fund 2007-2013 - Axis IV Human Resources).

REFERENCES

[1] F. Palumbo, N. Carta, D. Pani, P. Meloni and L. Raffo, "The multi-dataflow composer tool: generation of on-the-fly reconfigurable platforms," *J. of Real Time Image Proc.*, v.9, issue 1, pp. 233-249, 2014.

[2] N. Carta, C. Sau, D. Pani, F. Palumbo, and L. Raffo, "A coarse-grained reconfigurable approach for low-power spike sorting architectures," *IEEE/EMBS NER*, 2013.

[3] G. Kahn, "The semantics of a simple language for parallel programming," *Information Proc.*, North Holland (Amsterdam), 1974.

[4] E. Lee and T. Parks, "Dataflow process networks," *Proc. of the IEEE*, vol. 83, no. 5, 1995.

[5] J. Eker and J. W. Janneck, "CAL language report specification of the CAL actor language," *Tech. Rep.*, EECS Department (Univ. of California, Berkeley), 2003.

[6] S. M. Carta, D. Pani, and L. Raffo, "Reconfigurable coprocessor for multimedia application domain," *J. of VLSI Signal Proc. Sys.*, vol. 44, 2006.

[7] V. V. Kumar and J. Lach, "Highly flexible multimode digital signal processing systems using adaptable components and controllers," *EURASIP J. on Applied Signal Proc.*, 2006.

[8] F. Palumbo, D. Pani, E. Manca, L. Raffo, et al., "RVC: A multi-decoder CAL composer tool," *DASIP*, 2010.

[9] F. Palumbo, N. Carta, and L. Raffo, "The multi-dataflow composer tool: A runtime reconfigurable HDL platform composer," *DASIP*, 2011.

[10] ORCC - Open RVC-CAL Compiler - http://orcc.sourceforge.net/

[11] M. Wipliez, N. Siret, N. Carta, F. Palumbo, and L. Raffo, "Design IP faster: Introducing the C high-level language," *IP-SOC*, 2012.

[12] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam and D. Burger, *Dark Silicon and the End of Multicore Scaling*, *ISCA*, 2011.

[13] R. Puri, L. Stok, and S. Bhattacharya, "Keeping hot chips cool," *DAC*, 2005.

[14] S. Casale Brunet, E. Bezati, C. Alberti, et al., "Partitioning and Optimization of high level Stream applications for Multi Clock Domain Architectures," *SiPS*, 2013.

[15] R. Ren, J. Wei, E. J. Martínez, et al., "A pmc-driven methodology for energy estimation in RVC-CAL video codec specifications," *J. Image Communication*, vol. 28, no. 10, 2013.

[16] A. A.-H. A. Rahman, R. Thavot, S. C. Brunet, E. Bezati, and M. Mattavelli, "Design space exploration strategies for FPGA implementation of signal processing systems using CAL dataflow program," *DASIP*, 2012.

[17] F. Palumbo, C. Sau, and L. Raffo, "DSE and profiling of multi-context coarse-grained reconfigurable systems," *ISPA*, 2013.

[18] S. Huda, M. Mallick, and J. H. Anderson, "Clock gating architectures for FPGA power reduction," *FPL*, 2009

[19] C. Sau, L. Raffo, F. Palumbo, et al., "Automated Design Flow for Coarse-Grained Reconfigurable Platforms: an RVC-CAL Multi-Standard Decoder Use-Case," *(to appear) SAMOS*, 2014.

[20] N. Carta, C. Sau, F. Palumbo, D. Pani, and L. Raffo, "A coarse-grained reconfigurable wavelet denoiser exploiting the multi-dataflow composer tool," *DASIP*, 2013.