

Pre-print version.

The final publication is available at Springer via

https://doi.org/10.1007/978-3-319-93082-4_8

Palumbo F., Sau C., Fanni T., Raffo L. (2019)

Challenging CPS Trade-off Adaptivity with Coarse-Grained Reconfiguration.

In: De Gloria A. (eds) Applications in Electronics Pervading Industry, Environment and Society. ApplePies 2017. Lecture Notes in Electrical Engineering, vol 512. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-93082-4_8

Challenging CPS Trade-Off Adaptivity with Coarse-Grained Reconfiguration

Francesca Palumbo¹, Carlo Sau², Tiziana Fanni², Luigi Raffo²

¹University of Sassari (Italy), fpalumbo@uniss.it

²University of Cagliari (Italy), (name.surname)@diee.unica.it

Abstract

Cyber Physical Systems are highly adaptive systems, prone to change behaviour due to external/internal conditions. From the computation point of view, reconfigurable systems may address adaptation. In this paper, by a set of examples we show how coarse-grained reconfiguration may successfully allow achieving dynamic trade-off management, while considering different technology targets and different design flows.

1 Introduction

In the era of Cyber-Physical Systems (CPS) designers need to cope with complex devices composed of different interacting and deeply intertwined components, with multiple and distinct behavioural modalities variable over time. CPS are characterized by three dominant layers: namely, the functional, the physical and the communication ones. These layers can be specified through different levels of abstraction and require deep inter/intra-communication. Computing devices, human users and physical environment are tightly bound, making CPS prone to changes. They no longer offer hard-wired performance with identical and predictable behaviour or execution profiles over time: CPS are highly evolvable systems where functional (F) and non-functional (NF) requirements are variable, and autonomous adaptation to environmental changes or unpredictable human requests should be supported. Varying workloads and performance objectives have to be guaranteed, while continuously optimizing performance goals, i.e. minimizing energy consumption, meeting the available power budget, and so on.

This paper demonstrates that, from the computing perspective, adopting a Coarse-Grained Reconfigurable (CGR) approach can suite the scope of providing flexibility and adaptation to changeable F/NF requirements. As an example you can consider an embedded video decoding system that lowers the quality of its output to augment battery life [1]. Embedded systems for acquisition, encoding and transmission of environmental images may benefit from this variability support. By continuously monitoring F (decoding quality) and NF (remnant battery) requirements, it would be possible to drive the CGR support to serve user request or to suggest him/her a better device configuration.

This paper is organized as follow. Sect. 2 presents foundations and methodology to enable dynamic trade-off management. Sect. 3 provides, through different use-cases, a proof of concept of the feasibility of adopting a CGR approach to serve dynamic and variable environments. Sect. 4 concludes with final remarks.

2 Hardware Adaptivity Support in CPS Designs

CPS adaptivity implies system level flexibility. This latter typically collides with performance. Indeed, general purpose entities (i.e. CPUs, GPUs and DSPs) implement potentially any application described in the supported programming language, but their performance are quite limited due to their poor specialization. On the contrary, ASICs platforms boost performance,

but execute only the application they have been designed for. In between, an appealing solution is provided by the adoption of reconfigurable platforms. Reconfigurable architectures are basically meshes of processing elements (PEs) whose functionality and connections can be configured at runtime. Depending on the granularity of the PEs, it is possible to have fine-grained or coarse-grained reconfigurability. The former, typical of FPGAs, involves bit-level PEs. It presents high flexibility though bit-level programmability, but implies some configuration time overhead. CGR systems deal with word-level PEs, guaranteeing less flexible, but faster, reconfiguration. Combinatorial switching logic allows single-cycle reconfiguration and, on top of that, a CGR approach can make ASIC designs flexible, allowing them to switch among a finite set of input functionalities. In the following we explain the contribution of this paper and the adopted methodology.

2.1 Contribution

CGR platforms already demonstrated to be effective in flexible, but constrained, scenarios [4], and, in our opinion, they can tackle the adaptation needs of CPS designs. In this paper, we intend to show how CGR systems are naturally capable of guaranteeing dynamic trade-off management among relevant system metrics.

Mapping CGR platforms is not so straightforward, requiring a deep knowledge on the functionalities/kernels to be implemented. To mitigate this issue automated or semi-automated design environments have been proposed in literature [5, 6]. In this paper, we intend to prove that automated flows does not affect trade-off guarantees. On the contrary, in some cases, the features of those environments could possibly improve the trade-off itself.

2.2 Dynamic Trade-off Management

PEs used in CGR systems, can be homogeneous (identical computing blocks) or heterogeneous (not identical elements) and the computing fabric may not necessarily be composed of a regular, fully connected, infrastructure. PEs are normally not constrained in granularity, ranging from ALUs to a discrete cosine transform. The more a CGR system is customized to fit application needs, with application specific PEs and avoiding any redundancy or extra connection, the more its efficiency and performance are maximized. On the other hand, heterogeneous and irregular platforms are those that suffer the most mapping issues.

We have addressed the problems of dimensioning the hardware substrate and of mapping several kernels over it by combining dataflow models to the

CGR approach. This combination allowed us facilitating and speeding-up system deployment, while being able to offer adaptivity support.

Design Time Support: Datapath merging techniques are used to minimize the number of PEs and communication links integrated into a CGR datapath. Our dataflow to hardware software infrastructure, named Multi-Dataflow Composer (MDC) tool, starting from different input graphs, combines them into a unique specification, which is then synthesized in hardware according to a one-to-one mapping strategy between graph nodes and PEs [6]. Different input graphs share common PEs accessing them by means of configurable switching elements, responsible of forking/joining the execution flow where needed. Some examples of the outcome of this process are shown in Fig. 1

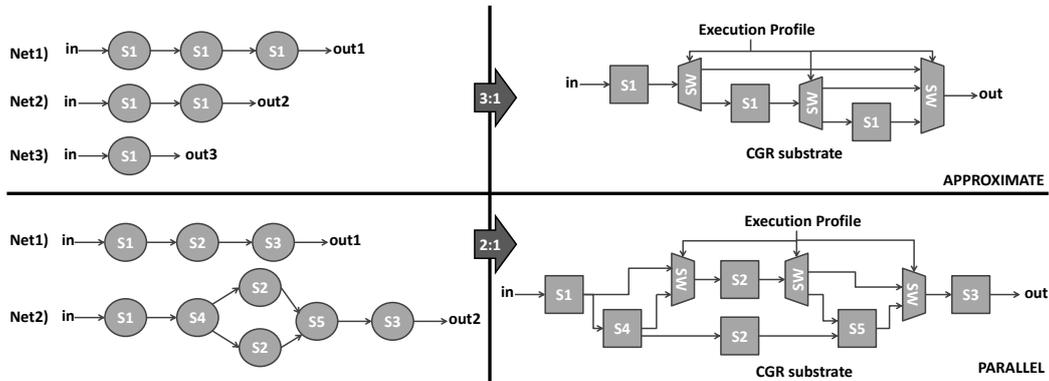


Figure 1: Dataflow to hardware: Trade-off friendly CGR platforms.

Run-Time Management: The CGR substrate executes all the input specifications one at a time. By switching from one configuration to another, varying the *execution profile*, you may change the system behaviour and performance. In CPS designs, the execution profile may vary due to user requests or internal system conditions. On the top of Fig. 1, an example of functional approximate computing is provided: the depth of the computation can be sized to serve different energy versus quality profiles. The more stage you use the more precise is your computation, but it is more energy hungry. At the bottom of Fig. 1, doubling the actor S2 you are able to execute faster, consuming more power. In both cases, different trade-offs between F/NF requirements are implemented over the same substrate and can be tuned according to the CPS needs at run-time.

3 Experimental Results

The effectiveness of the CGR approach in dynamic trade-offs management is demonstrated on three scenarios. The first one is an AES blockcipher manually implemented on a Xilinx Artix-7 FPGA, while the others demonstrate that automatic deployment does not affect dynamic trade-off tuning neither adopting an FPGA technology (HEVC motion compensation interpolator) nor adopting an ASIC 90 nm CMOS one (FFT accelerator). Design automation follows the approach described in Sect. 2.2. All the reported power/energy numbers consider post-synthesis switching activity.

3.1 Manually Implemented CGR AES on FPGA

The 128 bits Advanced Encryption Standard (AES-128) blockcipher is the de-facto encryption standard worldwide [7]. The cipher encrypts 128 bits of data in one pass of the algorithm and uses a secret key of length 128 bits. Mathematically, the ciphertext is achieved in 10 subsequent executions of a *Round Function* over the input plaintext. Lightweight AES designs adopt serialization, reducing the hardware footprint by reusing components over multiple clock cycles: latency increase allows energy saving [8]. In this scenario, Banik et al. demonstrated that energy consumption strictly increases with the number of *rolling rounds* (r) [9]. Therefore, a CGR platform combining designs with different number of r would offer different execution profiles.

Our CGR infrastructure offers different working points: smaller r values give slower energy efficient ciphers, larger r quicker less energy efficient ciphers. Fig. 2(a) depicts the trade-off curve achievable by a CGR AES-128 design, where different profiles with 2, 3 and 4 r are implemented. Energy consumption is proportional to r : to reach 25% additional throughput, going from $2r$ to $4r$, you need to pay an extra 6% of energy consumption. Smaller r profiles suite less computational demanding tasks, e.g. communicating with a RFID. Larger r profiles, enabling the maximum throughput, can be used to serve high performance applications, e.g. video stream encryption. In terms of resources, a CGR system typically implies an overhead. Here we have no FF penalty, but +16% LUTs are required wrt a $4r$ AES stand-alone design (on the XC7A35t1CPG236 FPGA).

3.2 Automatically Derived HEVC Interpolator on FPGA

The High Efficiency Video Coding (HEVC) standard provides 50% more subjective video quality with no penalty on the bit rate, at the price of an

increased complexity and system consumption [10]. Within HEVC decoding, motion compensation is one of the most computationally intensive portions of the algorithm. When fractional pixel motions have to be compensated, the block prediction is performed through an interpolation (two cascaded N taps FIR filters, one for horizontal and one for vertical motion) of its reference block. To reduce and dynamically tune the energy consumption of the interpolator, Nogues et al., [2] exploited functional approximate computing. They demonstrated in software that it is possible to waive some image quality (# of taps reduction) to save energy. In hardware, an analogue behaviour can be obtained leveraging on CGR architectures, as presented in [3]: starting from a legacy implementation of the interpolation filters, run-time adaptive solutions can be derived by dynamically excluding some taps from the computation (as presented on top of Fig. 1).

In this paper, we present the results achieved by feeding different data-flows, representing variable interpolation filter sizes, to the MDC tool to automatically produce a CGR luma interpolator (*reconf_luma*), performing 8, 7, 5 and 3 taps filtering, and a CGR chroma one (*reconf_chroma*), performing 4, 3 and 2 taps filtering. Reconfiguration has a negligible impact on FFs (legacy luma +1.6%, legacy chroma +5%), but largely impact on LUTs (legacy luma +186%, legacy chroma +63%) on the considered FPGA target (XC7A100TCSG324). The trade-off curves for both the colour space components are shown in Fig. 2(b). The 3-taps CGR luma profile saves 15% of energy per block wrt the 8-tap legacy configuration, while CGR chroma saves up to 5%. This variability can be exploited on a smart device equipped with a proximity sensor: when the user is close to the device, quality should be high, but when he/she is far, and cannot distinguish details, the quality can be lowered to save energy. This demonstration testifies that, even adopting a completely automated flow (which was not the case in [3]), dynamic trade-off management can be still offered.

3.3 Automatically Derived and Optimized FFT on ASIC

Fast Fourier Transform (FFT) is an optimised algorithm for the Discrete Fourier Transform calculation, widely adopted in several applications (from differential equations to digital signal processing). This use case involves a radix-2 FFT of size 8, obtained by means of three pipelined stages of four butterflies each, meaning 12 butterflies overall. From the baseline 12 butterflies design, variants have been derived decreasing the butterflies number: resources are multiplexed in time and reused, latency increases and throughput becomes lower. The CGR infrastructure, automatically derived with the MDC tool, includes the following profiles: (1) *12b* is the baseline 12 butter-

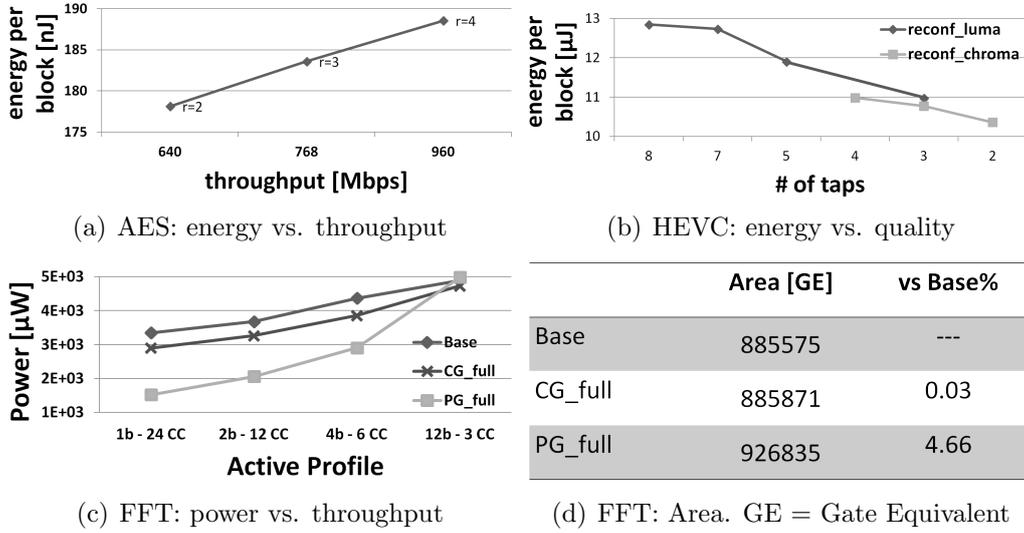


Figure 2: Results Analysis.

flies FFT design, taking 3 clock cycles to execute; (2) $4b$ involves 4 butterflies and computes in 6 cycles; (3) $2b$ 2 butterflies and 12 cycles; (4) $1b$ 1 single butterfly and 24 cycles. The trade-off analysis is presented in Fig. 2(c) as the *Base* curve. This graph presents power versus throughput, and confirms that dynamic trade-off management is achievable, on ASIC too, using an automated design flow. Power instead of energy results are shown: in some cases the rate of producing/consuming energy has to be considered, e.g. to ensure battery would be able to power up the other logic.

On ASIC more efficient power reduction methodologies can be implemented to save the power overhead due to the additional logic requested by CGR systems and due to the fact that these designs may have large portion of the system in idle while one of the profiles (input dataflows) is running. MDC offers the possibility to automatically implement power-gated and clock-gated designs [11]. The results of such implementations are shown in Fig. 2(c) respectively as *PG_full* and *CG_full* curves. The *PG_full* case demonstrates to be capable of achieving the largest power saving going from the $1b$ to the $12b$ FFT implementation. In Fig. 2(d) the area overheads of these more advanced implementations are reported: it is clearly negligible, being always below 4%.

4 Conclusions

In this paper we analysed the possibility of providing adaptivity support to different execution profiles by leveraging on the coarse-grained reconfigurable paradigm. Considering different application scenarios, we proved that a CGR approach can suite the F/NF requirements run-time adjustment with different technologies and when automated design strategies are adopted without any particular need of manual fine tuning. The results of this paper will be used as a starting point for the EU Project CERBERO (<http://www.cerberoh2020.eu/>).

Acknowledgements

This work has received funding from the EU Commission's H2020 Programme under grant agreement No 732105.

References

- [1] Ren R., et al.: Energy-aware decoder management: a case study on RVC-CAL specification based on just-in-time adaptive decoder engine. *IEEE Trans on Consumer Electronics* vol. 60(3) pp. 499-507 (2014).
- [2] Nogues E., et al.: A modified HEVC decoder for low power decoding. *Conf on Computing Frontiers* (2015).
- [3] Palumbo F., et al.: Runtime Energy versus Quality Tuning in Motion Compensation Filters for HEVC. *Programmable Devices and Embedded Systems Conf* (2016).
- [4] Yan M., et al.: ProDFA: Accelerating domain applications with a coarse-grained runtime reconfigurable architecture. *Parallel and Distrib Sys Conf* (2012).
- [5] Ansaloni G., et al.: Integrated Kernel Partitioning and Scheduling for Coarse-Grained Reconfigurable Arrays. *IEEE Trans on CAD of Integrated Circuits and Systems* vol.31(12), pp. 1803–1816 (2012).
- [6] Sau C., et al.: Automated Design Flow for Multi-Functional Dataflow-Based Platforms. *Jrnl of Signal Processing Systems* vol. 85(1), pp. 143-165 (2016).

- [7] Daemen J., Rijmen V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002).
- [8] Moradi A., et al.: Pushing the limits: A very compact and a threshold implementation of AES. Theory and Applic of Cryptographic Tech Conf (2011).
- [9] Banik S., et al.: Exploring Energy Efficiency of Lightweight Block ciphers. Selected Areas in Cryptography (2015).
- [10] Sullivan G.J., et al.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans on Circuits and Systems for Video Tech vol. 22(12), pp. 1649-1668 (2012).
- [11] Palumbo F., et al.: Power-Awareness in Coarse-Grained Reconfigurable Multi-Functional Architectures: a Dataflow Based Strategy. Jnl of Signal Processing Systems vol. 87(1), pp. 81-106 (2017).