i-DPs CGRA: An Interleaved-Datapaths Reconfigurable Accelerator for Embedded Bio-signal Processing

Loris Duch, *Graduate Student Member, IEEE,* Soumya Basu, *Graduate Student Member, IEEE,*
Miguel Peón-Quirós, *Member, IEEE,* Giovanni Ansaloni, *Member, IEEE,* Laura Pozzi, *Member, IEEE,*
David Atienza, *Fellow, IEEE*

***Abstract*—Smart edge sensors for bio-signal monitoring must support complex signal processing routines within an extremely small energy envelope. Coarse-Grained Reconfigurable Arrays (CGRAs) are good candidates for tackling these conflicting objectives because, thanks to their flexibility and high computational density, they can efficiently support the computational hot-spots characterizing bio-DSP applications. The *Interleaved-Datapaths* (i-DPs) CGRA presented in this paper further leverages the benefits of this architectural paradigm, focusing on ultra-low energy operation. Its defining feature is the complex design of its computing cells, which, by embedding multiple i-DPs, allow a high ratio between computing and control logic, effectively speeding up computations, and resulting in a marginal impact on the required IC area. Interleaved datapaths increase the energy efficiency of up to 33 %, with respect to a single-DP alternative, when executing common kernels in the multi-lead ECG signal processing field.**

***Index Terms*—Ultra-low Power, Bio-signal Processing, Coarse-Grained Reconfigurable Arrays, SIMD.**

I. Introduction

COARSE-Grained Reconfigurable Arrays (CGRAs) are flexible architectures that efficiently execute the intensive loops (i.e., computational kernels) that characterize applications in the embedded systems domain [7]. They are structured as 2-dimensional meshes of Reconfigurable Cells (RCs), with each cell embedding a set of Configuration Registers (CRs) and a Datapath (DP). The datapath usually comprises an Arithmetic Logic Unit (ALU) and a local Register File (RF).

The authors of [3] highlighted that important efficiency gains can be obtained by employing CGRAs in Digital Signal Processing (DSP) architectures. A similar conclusion is drawn in [4] and [5], which propose an embedded platform for bio-signal processing in personal health monitors, an increasingly relevant domain with ultra-low power constraints [8]. Similarly to us, [5] describes a CGRA with multi-DP RCs operating in SIMD (single instruction-multiple data) mode. Their proposed strategy, however, is only beneficial when the same acceleration is requested by different processors, themselves executing in SIMD mode. SIMD CGRAs speed up the execution of kernels, while also reducing the ratio between control and processing logic, both of which contribute to increase computational efficiency. Nonetheless, they also require a high memory bandwidth toward the data memory.

Against this backdrop, we illustrate an optimized CGRA mesh that, by employing cells with *Interleaved-Datapaths* (i-DPs), parallelizes the execution of kernels without impacting the width of the CGRA-memory link. As opposed to [5], our approach results in high efficiencies regardless of the structure and operating states of other system components (e.g., processors) at run-time. Our contribution is two-fold:

- We describe a novel i-DPs CGRA architecture that, by featuring interleaved DPs governed by the same control logic, is able to concurrently minimize the energy envelope of the mesh and increase its performance.
- We perform a systematic investigation of the benefits of our architectural choices, considering various kernels with different characteristics belonging to two real-world bio-signal analysis applications.

This work proceeds as follows: Section II provides details on the mesh structure and its interface with a domain-specific multi-processor system [2]. Section III assesses the

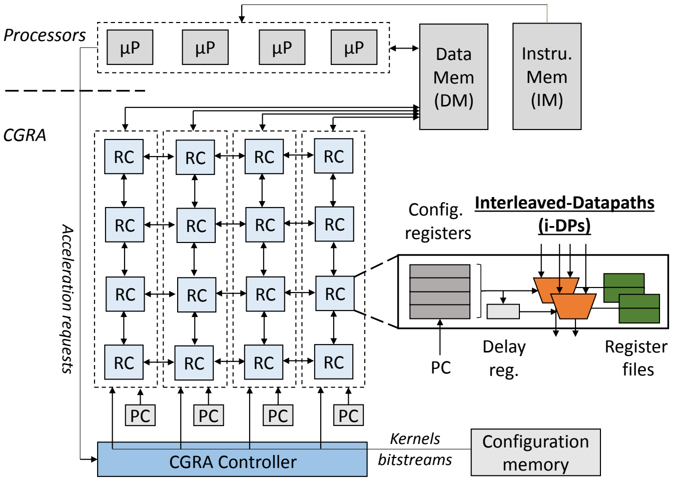

Fig. 1: i-DPs CGRA block scheme, interfaced with a multi-processor system.

L. Duch, S. Basu, M. Peón-Quirós and D. Atienza are with the Embedded Systems Laboratory (ESL), Swiss Federal Institute of Technology (EPFL), Switzerland, e-mail: {loris.duch, soumya.basu, miguel.peon, david.atienza}@epfl.ch.

G. Ansaloni and L. Pozzi are with the Università della Svizzera Italiana (USI), Lugano, Switzerland, e-mail: {giovanni.ansaloni, laura.pozzi}@usi.ch.

This work has been partially supported by the E4Bio RTD project (no. 200021 159853) evaluated by the Swiss NSF, by the Award Grant (no. N62909-17-1-2006) by ONR-G, and by the MyPreHealth project (no. 16073) founded by the Hasler Stiftung.

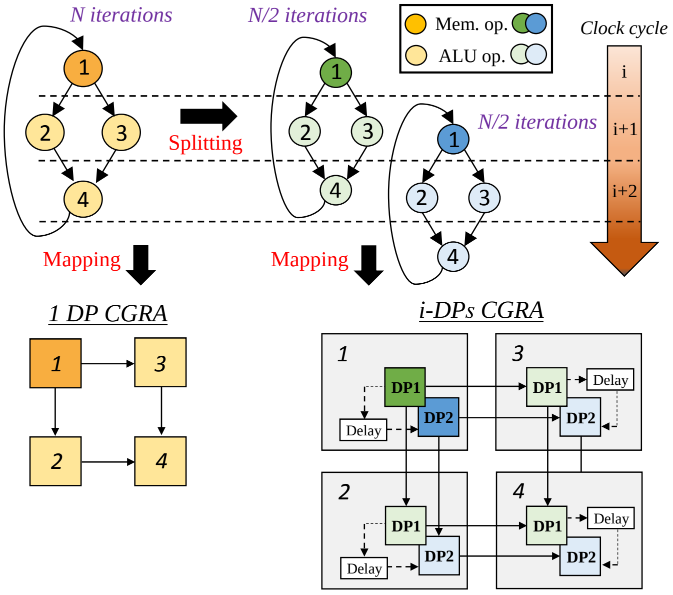

Fig. 2: Example of a simple kernel mapped in a single-DP and an i-DPs CGRA. In the latter case, the kernel is split into two slices and mapped on the interleaved DPs of the RCs.

i-DPs CGRA performance from a run-time, energy and area perspective. Section IV concludes the paper.

II. Accelerator Architecture

The main structure of the programmable accelerator is a homogeneous mesh of Reconfigurable Cells (RCs), with nearest-neighbor connections. At its periphery, a CGRA controller is in charge of responding to acceleration requests from the processors, which are issued by the software application using dedicated instructions. When an acceleration is required, the controller checks if enough resources (RC columns) are available in the mesh, and, in this case, programs them by transferring the appropriate bit stream to the RCs configuration registers. This mechanism allows the accelerator to serve requests with various numbers of CGRA columns, and to concurrently execute multiple requests that originate from different processors.

Then, during kernel execution, the active configuration word is determined by column-wise program counters, enabling a spatio-temporal mapping of operations on the mesh [1]. The active configuration word dictates which arithmetic or logic operation is performed by the cell, the source of the operands (either the neighboring cells or the local RF) and the output destination. Moreover, only few bits are necessary to encode this information, enabling the support for multiple configuration words per RC.

Kernels must access Data Memory (DM) to process inputs and store outputs. In our implementation, transfers between the CGRA and DM are supported by multiplexing the port of the processor that issued the acceleration, hence allowing the transfer, at each clock cycle, of one word of data per active kernel. To avoid access conflicts, the i-DP CGRA skews (using a delay register) by one cycle the active configuration word between DPs (see Fig. 1).

Fig. 2 shows an example of a simple kernel mapped on a single-DP and on an i-DPs CGRA with two datapaths. The two architectures have the same configuration words, but in the i-DPs case the same set of configuration registers govern two different DPs. Separate kernel slices can then be executed concurrently on the CGRA (skewed by one clock cycle), with the ensuing gain in parallelism being only limited by the amount of load/store operations, and the available bandwidth between the data memory and the CGRA.

Little hardware and run-time overhead is required to support interleaved datapaths. On the hardware side, delay registers in each RC store one configuration word per DP (32 bits in our implementation). A multiplexer is also required to select, at each clock cycle, which DP can access the data memory. During run-time, scalar constants have to be written for each slice in the local RF of the DPs, and scalar results transferred back to data memory. We show in Section III-B that these overheads are dwarfed by the gains attained in execution time.

The adopted strategy can effectively map two common kernel structures. First, kernels that do not present loop carries can perform a slice of all iterations in each DP, without further modifications. Second, reduction kernels, which compute one (or few) scalar values from an input array, can be divided into multiple parts, but require a wrap-up phase in software to aggregate the obtained results. Even in this last case, notable speed-ups can be obtained with i-DPs with respect to a single-DP arrangement, when the input set is sufficiently large.

III. Experimental evaluation

A. Experimental Setup and Bio-signal Processing Benchmarks

Similarly to [5], we characterized the system components (including processors, memories and the i-DP CGRA) at the post-synthesis level, targeting a 65 nm UMC technology. The obtained energy parameters were then employed to annotate a cycle-accurate virtual platform (specified in SystemC), allowing fast whole-system simulations of entire applications.

To evaluate the performance of the i-DPs CGRA, we consider two applications that process electrocardiogram (ECG) samples. First, 8-lead Compressed Sensing (8L-CS) [5] derives a number of random features as linear combinations of input samples [6]. In the targeted implementation, we adopted signal windows of 1024 samples, and a compression ratio of 50 %. The CS kernel computes the random indexes using a linear feedback shift register. It does not present loop-carried dependencies, and can therefore be straightforwardly mapped on our accelerator by distributing its iterations equally among the available DPs. Secondly, 6-lead Morphological Filtering (6L-MF) cancels the baseline wandering of an ECG record [9]. Its kernels compute the first and second maximum and minimum along sample windows. They can therefore be divided in slices, each returning the two higher/lower values in a sub-window, with a (short) software wrap-up routine that determines the two final outputs among the values computed by the CGRA.

B. Performance Analysis

While kernels can be conceivably divided in a high number of slices, the attainable gains may offer diminishing returns

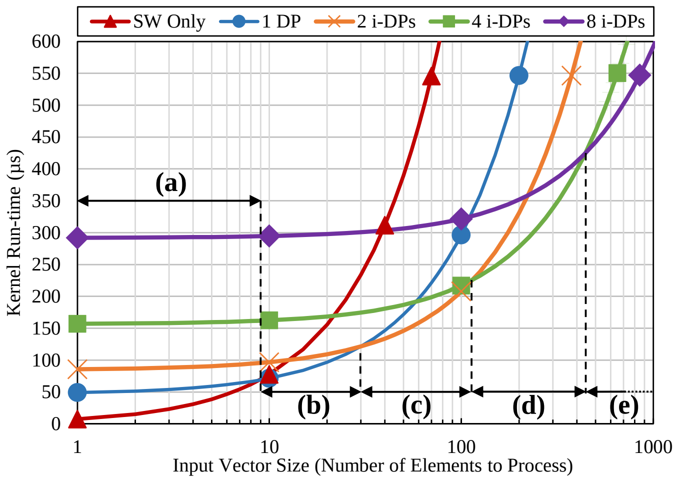

Fig. 3: Runtime of *Dbl Min Srch* kernel (first and second minimum element of an array), on a single-DP CGRA and on i-DPs with different widths, varying the input data size.

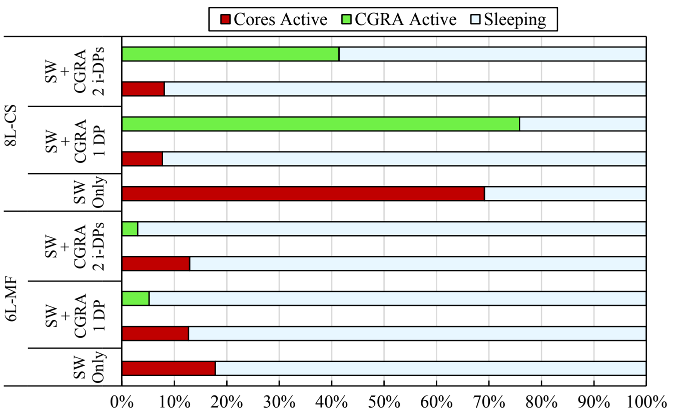

Fig. 5: Multi-core and CGRA utilization time in the considered platforms (% of the total run-time at a 2MHz clock frequency).

in terms of execution time, due to the bandwidth bottleneck between the data memory and the reconfigurable fabric. Moreover, a large number of DPs may incur a large timing overhead for the transfer of initialization values and scalar outputs to/from the CGRA for each slice. In addition, for reduction kernels, the time required for the wrap-up phase (cf. Section II) increases proportionally with the number of slices. Therefore, the selection of a proper DP width for a kernel depends on the amount of its memory accesses and on its number of iterations.

We investigate this last aspect in Fig. 3, which showcases the trade-offs between the number of elements to process and the number of interleaved DPs, depicting the global execution time of the *Dbl Min Srch* kernel, a hotspot of the 6L-MF benchmark. For very small datasets (region (a)), it is not worthwhile to configure and invoke the CGRA, as the entailed overhead is larger than the benefit of hardware acceleration. As the data size increases, also does the complexity of the best performing cells. Indeed, 8 i-DP cells present the lowest run-time if the input size exceeds 446 elements (region (e)).

Since, in the considered benchmarks, the kernel input data vectors have an average size of 100 elements (which is a typical scenario for bio-signal analysis applications), in the rest of this paper we consider only a 2 i-DPs CGRA configuration. Similar results were obtained for the other investigated kernels. Fig. 4 illustrates their execution time on a single-DP and on an i-DPs CGRA with 2 DPs, without considering the software overhead required by the processors to configure, launch and recover the results from an acceleration. The graph shows that, by adopting i-DPs, a large reduction is obtained in the time required for computing the kernel outputs (*Execution & Data Transfer* phase), which is almost halved. We also measured an increase in the *Fetch & Decode* phase, due to the extra register settings required for the initialization of multiple kernel slices, but its impact is negligible (less than 1.6 % in all cases).

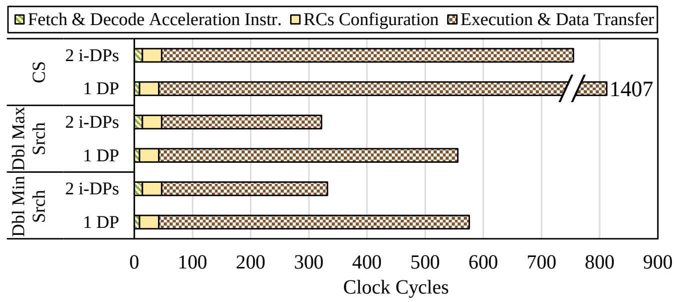

Fig. 4: Average kernels run-time (in clock cycles) executing on CGRAs with 1 DP and 2 i-DPs.

The above-mentioned gains are reflected at the system level. To assess them, we considered a system interfacing a CGRA with the multi-core architecture described in [2]. Fig. 5 reports the active and sleeping time of the multi-core system and the CGRA, as a percentage of the total run-time of the application. A first consideration that can be drawn from this data is that both applications are rather kernel-intensive, with 28 % and 88 % of the active time spent in the kernel functions for 6L-MF and 8L-CS, respectively. Furthermore, the figure shows that the i-DPs CGRA allows a marked decrease of CGRA active time, compared to its single-DP version. In fact, the CGRA activity is decreased by almost half in the case of the 8L-CS, and by one third for the 6L-MF application.

C. Energy Analysis

From an energy viewpoint, the savings obtained by the i-DPs CGRA stem from two sources. First, increased idle

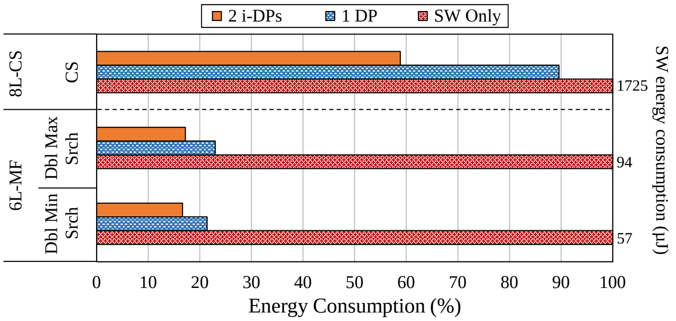

Fig. 6: Energy consumption of the different kernels. For each kernel, the bars are normalized to the SW-only energy.

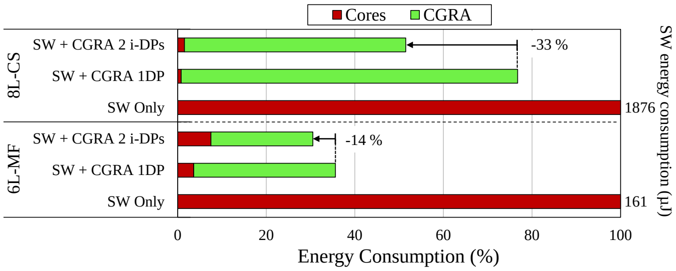

Fig. 7: System energy consumption of the different benchmark applications, normalized to the SW-only consumption for the part of the application transfered to the CGRA.

times are exploited to aggressively clock-gate idle components, resulting in a decrease in dynamic energy. Second, the i-DPs scheme results in a high ratio between the CGRA logic devoted to computing (RCs) and that used to control the execution flow (configuration registers), which results in increased efficiency. Fig. 6 highlights that important savings are attainable for each kernel by employing interleaved datapaths. In the case of the *CS* kernel, the energy budget (with respect to an equivalent 1-DP architecture) is reduced by 34 %. For the two kernels of 6L-MF benchmark (*Dbl Min Srch* and *Dbl Max Srch*), the reductions are 22 % and 25 %, respectively.

Fig. 7 compares the energy consumption of the part of the workload that is accelerated for the two considered benchmarks. Again, three architectural choices are considered: a multi-processor platform [2], that does not embed a reconfigurable mesh; a platform that couples a multi-core platform with a single-DP CGRA [4]; and our proposed system, featuring an i-DPs accelerator. It can be noted that, even in the two latter cases, certain software overhead is required for setting up, launching and retrieving the outputs of an acceleration request. This component of the energy budget of kernels is even more pronounced for the i-DPs case, since, as discussed before, i-DPs requires a more complex initialization phase. The increase is particularly noticeable for the 6L-MF benchmark which, being a reduction algorithm, also requires wrap-up computations, performed in software. The energy efficiency derived from the use of i-DPs is nonetheless substantial: 14 % and 33 % for 6L-MF and 8L-CS, respectively, when compared with a 1-DP CGRA. Furthermore, the resulting energy envelopes are always smaller by a large margin with respect to the processor-based alternative (SW Only in Fig. 7).

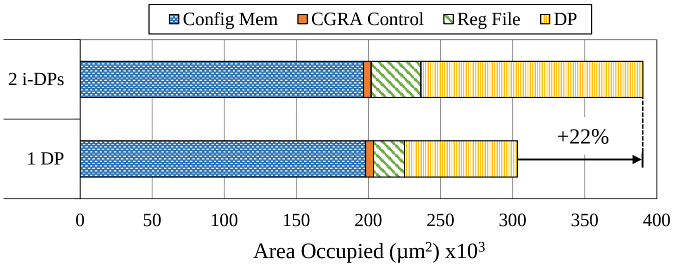

Fig. 8: Area breakdown of the i-DPs CGRA compared to a traditional CGRA.

D. Area Analysis

Fig. 8 presents a breakdown of the silicon real estate required for two CGRAs with a single and two interleaved DPs. In both cases, we considered 4x4 meshes. RCs comprise 32 configuration registers (of 32 bits each), which suffice to map all considered kernels. As in the system in [2], the considered data bitwidth is 16 bits. Register files, local to each DP in the RCs, can store 4 words. Since increasing the i-DPs width only impacts the logic required for the datapaths themselves and the local register files, doubling it from 1 to 2 only entails an area overhead of less than one quarter. In fact, a sizable portion of the CGRA area is employed by the configuration memory, whose area does not depend on the number of datapaths.

IV. Conclusion

The design and use of reconfigurable architectures for wearable bio-medical signal analysis involves a trade-off between the achieved degree of flexibility and the overhead (in terms of control logic and configuration time) required to program a functionality. By being configurable at the operation level, CGRAs strike a good balance among these metrics, leading to a highly efficient support of the computational kernels present in the ultra-low-power DSP field, including the ECG analysis applications considered in this study.

The showcased i-DPs CGRA goes one step further, as it features multiple computing elements governed by the same control logic. Its interleaved run-time scheme maximizes the utilization of resources and of the available bandwidth between the CGRA and the data memory, leading to notable run-time and energy efficiency gains. Our proposed scheme is particularly beneficial for the computation of parallelizable and reduction kernels, common in the bio-DSP domain.

References

[1] M. Bingfeng et al. Exploiting loop-level parallelism on coarse-grained reconfigurable architectures using modulo scheduling. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 296–301, March 2003.

[2] R. Braojos et al. Hardware/software approach for code synchronization in low-power multi-core sensor nodes. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1–6. IEEE, March 2014.

[3] S. Das et al. A 142mops/mw integrated programmable array accelerator for smart visual processing. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, Mar 2017.

[4] L. Duch et al. A multi-core reconfigurable architecture for ultra-low power bio-signal analysis. In *IEEE Biomedical Circuits and Systems (BioCAS)*, pages 1–4, October 2016.

[5] L. Duch et al. Heal-wear: An ultra-low power heterogeneous system for bio-signal analysis. *IEEE TCAS-I*, May 2017.

[6] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes. *IEEE Trans. on Biomedical Engineering*, 58(9):2456–2466, 2011.

[7] L. Ming-hau et al. Design and implementation of the morphosys reconfigurable computing processor. *Journal of Signal Processing Systems*, 24(2-3):147–164, March 2000.

[8] A. Pantelopoulos et al. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Trans. on Systems, Man, and Cybernetics, Part C Applications and Reviews*, 40(1):1–12, October 2010.

[9] Y. Sun et al. ECG signal conditioning by morphological filtering. *Computers in Biology and Medicine (CBM)*, 32(6):465–479, November 2002.