# PX-CGRA: Polymorphic Approximate Coarse-Grained Reconfigurable Architecture

Omid Akbari[1,3], Mehdi Kamal[1], Ali Afzali-Kusha[1], Massoud Pedram[2], Muhammad Shafique[3]
[1]School of Electrical and Computer Engineering, University of Tehran, Iran
[2]Department of Electrical Engineering, University of Southern California, USA
[3]Institute of Computer Engineering, Vienna University of Technology (TU Wien), Austria
{akbari.o, mehdikamal, afzali}@ut.ac.ir, pedram@usc.edu, muhammad.shafique@tuwien.ac.at

***Abstract*—Coarse-Grained Reconfigurable Architectures (CGRAs) provide tradeoff between the energy-efficiency of Application Specific Integrated Circuits (ASICs) and the flexibility of General Purpose Processors (GPPs). State-of-the-art CGRAs only support exact architectures and precise application executions. However, a majority of the streaming applications such as multimedia and digital signal processing, which are amenable to CGRAs, are inherently error resilient. Therefore, these applications can greatly benefit from the emerging trend of Approximate Computing that leverages this error-resiliency to provide higher energy efficiency proportional to the tolerable accuracy loss (can even be constrained).**

**This paper, for the first time, introduces the novel concept of Polymorphic Approximate CGRA (PX-CGRA) that employs heterogeneous tiles of Polymorphic-Approximated ALU Clusters (PACs) connected in a 2-D mesh style connection. These PACs can implement different approximate modes as well as accurate modes depending upon their selected configuration as per the run-time requirements of executing applications. For designing an efficient PX-CGRA, we propose a bottom-up design flow. In addition, the flow of application mapping on PX-CGRA is discussed including accuracy-level mapping, scheduling, and binding steps. To comprehensively evaluate the efficacy of the proposed CGRA, the complete PX-CGRA architecture in different sizes as well as with different PACs configurations are synthesized using a 15-nm FinFET technology. Our results show up to 15%-45% energy efficiency improvement for 5%-35% output quality degradation, respectively, when compared to the state-of-the-art exact-mode CGRA. Our proposed architecture and design methodology enable a new era of accuracy-configurable CGRAs to provide significant energy gains.**

***Keywords*— *Coarse-Grained Reconfigurable Architecture, Approximate Computing, Heterogeneous, Energy-Efficiency, Dark Silicon, Adder, Multiplier, Quality, Design.***

## I. Introduction

Coarse-Grained Reconfigurable Architectures (CGRAs) with word-level operations provide higher energy-efficiency in comparison to FPGAs. Moreover, dynamic reconfigurability makes CGRAs a more promising solution over ASICs for accelerating applications from a given domain [1]. The architectural view and the internal connections of a typical CGRA in a reconfigurable system are shown in Fig. 1. It is composed of an array of processing elements (PEs) connected in a 2-D mesh style interconnect, a main processor, a context memory, and a data memory [2]. The main processor sets connection and functionality of the PEs, by loading the context words from the context memory to context registers. The mapping process of applications into this type of CGRAs is a 1-to-1 mapping (e.g., see [1][2]). That is, each node in the application is mapped to a single PE in the reconfigurable array [3] (Fig. 2.a). This mapping method may limit the performance of applications due to extra needed clock cycles for performing an expression which is composed of several operations, e.g., expressions A and B in Fig. 2.b [4]. A few prior works tried to overcome this limitation by proposing reconfigurable interconnect [5], expression-grained reconfigurable arrays [3][4], mixed-grained fabrics [6], and deploying more than one processing elements in each cell [7]. The expression-grained CGRAs (Fig. 2.b) consist of an array of ALU sets, which provides the ability of mapping an entire expression into one cell, resulting in higher performance [3][4].

The current generation of CGRAs only supports the exact execution of applications. However, most of the stream-based applications, such as multimedia processing and signal processing, tolerate some imprecise results [8]-[10]. Approximate computing leverages this intrinsic resiliency to offer higher performance and lower power/energy consumption at the cost of accuracy loss [8]-[10]. Most of the prior works on hardware level approximate computing have focused on improving the performance of specific applications by employing approximate arithmetic units, such as adders and multipliers. An attractive research venue is to employ approximate arithmetic units in coarse-/expression-grained reconfigurable architectures (Fig. 2.c) to reach higher speed and lower energy consumption without violating the expected output quality constraint for a set of application domains. Hence, designing an efficient approximate CGRA bears a huge potential for an emerging research direction towards high performance and energy-efficient reconfigurable computing.

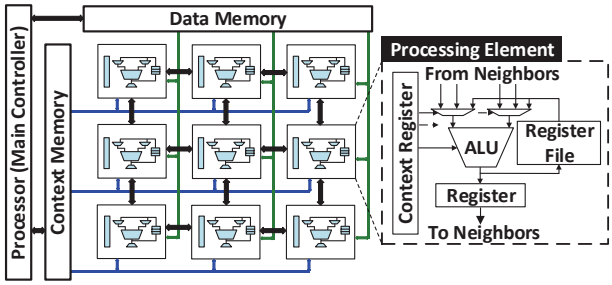


Fig. 1. Architectural Model of a typical 3x3 CGRA.

### A. Motivating Example

To show the effect of the number of approximate operations on the output quality and the energy consumption, we explored a number of approximate operations under different expected quality levels for a 32-tap finite impulse response (FIR) filter (composed of 32 multiplications and 31 additions) and a 32nd order polynomial evaluation (PoE) [7] (composed of 32 multiplications and 32 additions). The experimental setup will be discussed in Section I.C. The results of this study are shown in Fig. 3, which illustrates that the energy consumption as well as the number of approximate operations (secondary axis) depend on the expected output quality.

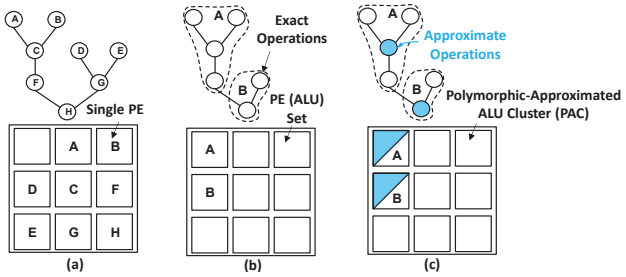


Fig. 2. (a) Typical CGRA with one ALU in each cell; (b) Expression-Grained CGRAs with a set of ALUs in each cell; (c) PX-CGRA with an array of PACs.

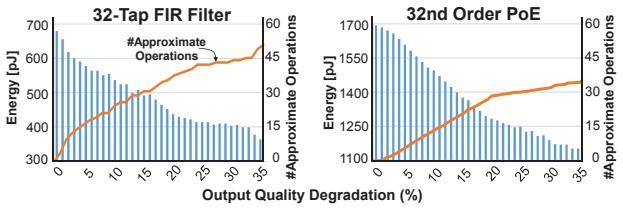


Fig. 3. Energy and number of approximate nodes of "32-Tap FIR filter" and "32nd order PoE" applications under different output qulaity constarints.

As the results show, the studied 32-Tap FIR filter may achieve up to 46% energy consumption reduction at the cost of 35% accuracy loss when 84% of the operations are imprecise. Also, in the case of the 32nd order PoE, by accepting 35% quality degradation, 32% energy consumption reduction has been achieved by performing 63% of the operations imprecisely. The selected approximate arithmetic units and the approximation method of this exploration are discussed in Sections III and IV, respectively.

**In summary,** our energy-quality analysis shows that different applications require different number of approximate operators depending on the output quality constraints to lower the energy consumption. Therefore, to design an approximate CGRA with the support for various quality levels of different applications, it may be beneficial to have a set of approximation-customized CGRA tiles that support distinct quality levels as well as different performance and power/energy consumption tradeoffs. These tiles can be dynamically allocated to an application to match the performance, power/energy, or the quality demands. Thus, we define the proposed approximation-customized CGRA as Polymorphic Approximate CGRA (PX-CGRA). However, provisioning tiles with various approximation levels may lead to significant area overhead. This can be used as an advantage in the dark silicon era, in which due to aggressive technology scaling only a fraction of the chip can be powered-on under a given Thermal Design Power (TDP) constraint [11][12].

**Dark Silicon Era:** Recently, many works leveraged dark silicon to improve performance/power/reliability of manycore systems and heterogeneous architectures , e.g., TDP budgeting for improving the performance/reliability [12][13], and using the architectural heterogeneity for power and performance [14][15]. The challenges of leveraging dark silicon for an adaptive approximate CGRA can be divided into two main categories:

**1) Design-Time Challenges:** To efficiently utilize the dark silicon area to design a PX-CGRA with support for different quality levels, it is important to have an architectural customization towards integrating many approximate CGRA tiles with different approximation features. The customization needs to account for all design steps, from selecting the proper approximate arithmetic units for constructing the overall architecture to proposing the corresponding mapping method including, accuracy level mapping, scheduling and binding steps.

**2) Run-Time Challenges:** Given such an approximation-customized architecture, it is necessary to efficiently allocate proper tiles to an application under the output quality constraint, while offering the least energy consumption.

### *B. Novel Contributions of This work*

In this paper, we show how the emerging paradigm of approximate computing can be leveraged to design energy-efficient CGRAs in the dark silicon era. Our novel contributions in a nutshell are:

1) Proposing a quality-configurable coarse-grained reconfigurable architecture composed of PX-CGRA tiles, and discussing the corresponding application mapping method including accuracy level determining, scheduling, and binding steps.
2) A bottom-up design methodology for customizing the Polymorphic-Approximated ALU Clusters (PACs) to build the PX-CGRA tiles that provide lowest energy-delay-area product per quality loss (i.e., lowest cost function).
3) Proposing a flow to efficiently allocate the PX-CGRA tiles for a given application to reach lower energy consumption without violating the expected quality.
4) Exploring state-of-the-art approximate arithmetic units, different configurations of PACs, and various PX-CGRA tiles using 15nm Fin-FET technology.

To provide a better understanding the reported results in the rest of the paper, we discuss the evaluation tool flow in next sub-section.

### *C. Tool Flow and Simulation Setup*

We developed a comprehensive experimental setup to provide an appropriate design space exploration for elementary approximate arithmetic blocks, different configurations of PACs, and various PX-CGRA tiles (see Fig. 4). First, Verilog HDL code for different configurations of PACs (PX-CGRA) was verified using gate level simulations with Modelsim HDL simulator. Afterward, the verified designs were synthesized using Synopsys Design Compiler with a 15-nm Fin-FET NanGate technology [16] at the 0.8V operating voltage level and under 2 GHz clock frequency constraint.

Next, the Value Change Dump (VCD) of the synthesized design was obtained using Modelsim HDL simulator, which corresponds to activity of internal nodes. Afterward, the obtained VCD file was translated to Switching Activity Interchange Format (SAIF) file that was used by Synopsys Prime Time to extract the accurate power consumption of the synthesized designs. Additionally, a MATLAB based quality analysis flow was developed to calculate the output quality of the studied applications. Finally, the extracted design parameters and output qualities were used to evaluate the efficiency of the design. Note that, the same flow was employed to extract the design parameters and accuracy analysis of approximate adders and multipliers presented in Section III.

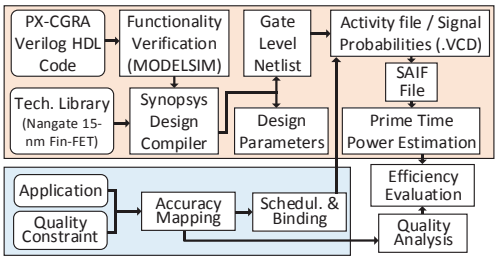


Fig. 4. Experimental flow and simulation setup.

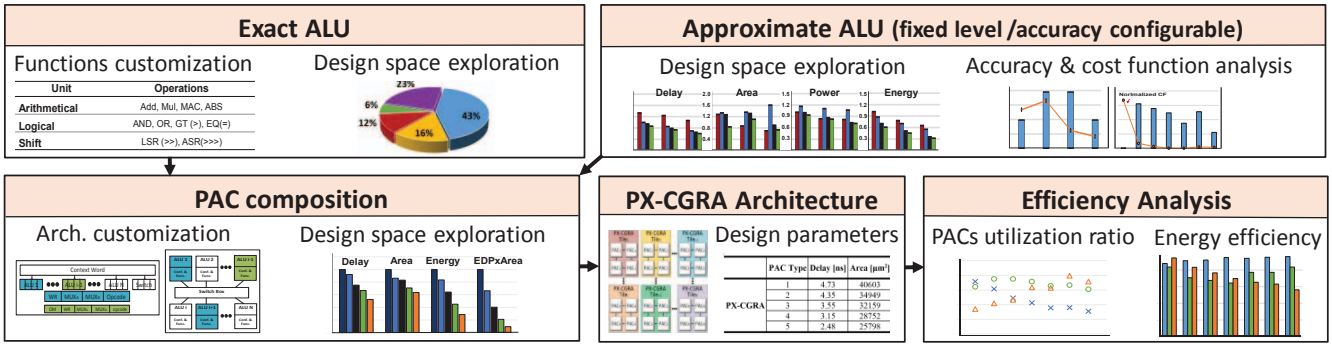


Fig. 5. PX-CGRA design steps along with evaluation method.

## II. Related Works

Recently, much attention has been devoted to approximate computing at different system layers. Exhaustive surveys on approximate computing have been conducted in [8],[17] and [18]. At the hardware layer, most prior works in approximate computing have focused on proposing approximate arithmetic units, including approximate adders [9],[19]-[22] and multipliers [10],[23]-[24]. Utilized methods in these approximate units were mainly based on logic manipulation and circuit simplification.

**Approximate Adders:** The approximate adders of [19], reduced the energy consumption by removing some transistors in the circuit of a mirror adder. However, using these approximate full adders in MSB bits of an approximate adder may lead to a considerable error [17]. An accuracy gracefully-degrading adder (GDA) was proposed in [20], which divided an adder into several sub-adders. Input carries of sub-adders could be accurate or inaccurate, which is specified by multiplexers. A low latency accuracy configurable adder was proposed in [21] that provides various latencies as well as different qualities. A reconfigurable approximate carry look ahead adder was proposed in [9], that supports switching between exact and approximate operating modes on the fly. A segmented ripple carry adder was proposed in [22] which used a carry prediction scheme for the input carry of each segment.

**Approximate Multipliers:** An approximate 2×2 multiplier was proposed in [23] which is mainly based on the simplification of the Karnaugh map of a 2×2 multiplier. This block can be used for constructing larger multipliers. Two approximate 4:2 compressors were proposed in [24] that were utilized in the reduction stage of an approximate Dadda multiplier. Four different accuracy configurable 4:2 compressors with the ability of switching between exact and approximate modes have been proposed in [10], which were assessed by utilizing in the structure of approximate Dadda multipliers.

**CGRAs:** Comprehensive surveys on reconfigurable architectures, in terms of the reconfiguration method, granularity (fine/coarse), and application domains can be found in [25][26]. Recently, some works have focused on improving the performance and energy efficiency of CGRAs by utilizing traditional methods such as hardware/software co-design and dynamic voltage and frequency scaling (DVFS) [2][27]. Among these methods, the dual supply voltage ($V_{DD}$) approach is one of the most common techniques (*e.g.*, see [28][29]). In [2], high $V_{DD}$ ($V_{DDL}$) was assigned to PEs that perform long delay operations such as multiplication, and low $V_{DD}$ ($V_{DDH}$) was assigned to PEs that perform short delay operations such as addition. An autonomous parallelism voltage and frequency selection (APVFS) method to enhance the energy efficiency of CGRAs was proposed in [27].

**To summarize,** state-of-the-art has not yet explored the potential of using approximate computing to design customizable expression-grained approximate CGRAs in the dark silicon era. PX-CGRA is the first architecture that uses this potential by systematically considering trade-offs between the energy efficiency improvement and quality loss.

## III. Proposed PX-CGRA

In this section, we discuss the bottom-up design flow of the PX-CGRA which starts from designing the ALU. We perform a quality analysis of approximate arithmetic units to build a proper approximate ALU considering different design parameters (delay, area, power/energy) and accuracy metrics. Fig. 5 shows the design steps and the evaluation method.

### A. ALU Design

**Exact ALU:** ALUs as the main components of CGRAs are configurable to perform various functions. The functions which are supported by the ALUs of PX-CGRA (which are most common in the streaming applications [7]) are listed in TABLE I. Obviously, these functions could be customized for the given target application. In addition, the maximum number of supported functions by the ALUs depends on design parameters (e.g., area) and allocated bit-width to the opcode field in the ALU context register (see Section III.B).

TABLE I. Supported Functions by ALUs of PX-CGRA

| Unit | Operations |
|---|---|
| **Arithmetical** | Add, Mul, Sub, MAC, ABS |
| **Logical** | AND, OR, XOR, NOT, GT (>), LT(<), EQ(=) |
| **Shift** | LSR (>>), LSL (<<), ASR(>>>) |

**Exploration of Approximate Adders and Multipliers for Approximate ALU Composition:**

Fig. 6 shows the energy distribution of the components of a typical exact PE (see Fig. 1). As shown in this figure, energy consumption of arithmetic units (adder/subtractor and multiplier) is about 59% of the total energy consumption. Therefore, approximation of these units may lead to a significant energy reduction of PX-CGRA.

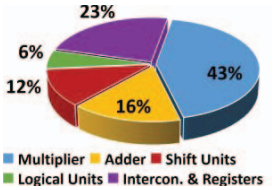


Fig. 6. Energy consumption breakdown (%) of an exact PE composed of the considered exact ALU under a random input set.

For designing the approximate ALU, we explore the state-of-the-art approximate adders and multipliers studied in Section II, in terms of their design parameters, including delay, area, power, and energy. The results of this exploration are shown in Fig. 7 and Fig. 8.

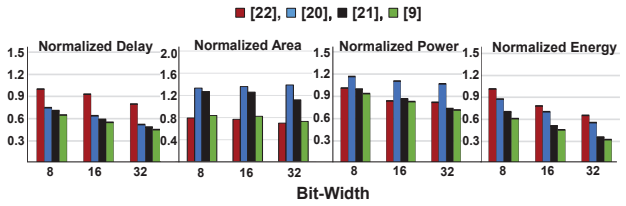


Fig. 7. Normalized design parameters of the studied approximate adders (when their approximate carry chain/sub-adder size is 4) for different bit widths compared to the those of an exact carry look ahead adder.

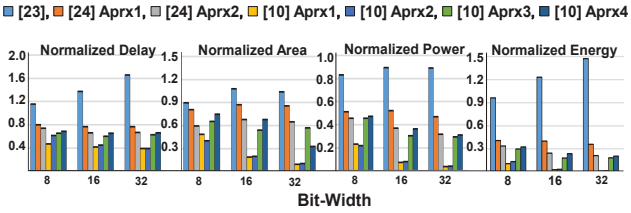


Fig. 8. Normalized design parameters of the studied approximate multipliers under different widths compared to the those of an exact Dadda multiplier.

However, to select the proper approximate arithmetic units for utilizing in the approximate ALUs, their accuracy is another important parameter that should be explored. Note that the approximate ALU supports all the functions reported in TABLE I.

**Accuracy Metrics of Approximate Arithmetic Units:** There are different metrics for investigating the accuracy of approximate arithmetic units [9]. Error Rate (ER) is the fraction of the number of inexact outputs to the total number of outputs. Error Distance (ED) is the difference between the exact and approximate outputs. Mean Error Distance (MED) is the mean of EDs. Normalized Error Distance (NED) is the normalized MED compared to the maximum error value of an approximate arithmetic unit. We selected the NED accuracy metric, because it is almost independent of the width of arithmetic units [10].

To select the approximate arithmetic units that provide a proper trade-off between design parameter improvement and quality loss ($Q_{loss}$), we used the cost function (CF) defined by [9]

$$CF = \frac{Energy \times Delay \times Area}{1 - Q_{loss}} \qquad (1)$$

where $Q_{loss}$ may be considered as NED for the approximate arithmetic units. The approximate arithmetic units that offer the lowest CF should be selected to compose approximate ALUs. Fig. 9 shows the NED of the studied approximate arithmetic units along with their normalized CF compared to the exact arithmetic units. Based on the results, we select approximate adder of [9] and 4th approximate multiplier of [10] which provide lowest CF.

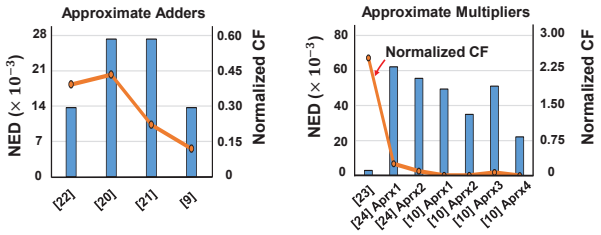


Fig. 9. NED of the studied approximate arithmetic units along with their normalized CF compared to the those of exact arithmetic units.

**Accuracy Configurable ALUs:** Some state-of-the-art approximate adders and multipliers provide run-time accuracy reconfigurability feature (e.g. see [9], [10], [20], and [23]), i.e., their operating mode (exact or approximate) is selectable by an accuracy operating mode (OM) signal. Therefore, they may be employed for constructing accuracy configurable ALUs. Adders and multipliers employed in these ALUs can have separate OM signals, which provide different accuracy levels. However, based on the reported results in [9], [10], [20], and [23], these components lead to higher design parameters in their exact operating mode in comparison with typical adders and multipliers. For example, by using the adder and multiplier proposed in [9] and [10], respectively, in their accuracy configurable mode, the accuracy configurable ALU has about 5% higher energy consumption in comparison with an exact ALU.

Fig. 10 shows the normalized values of energy consumption for the accuracy configurable ALU compared to the exact ALU. For the four accuracy levels, this ALU leads to, on average, 26% lower energy consumption in comparison with the exact ALU. TABLE II shows the design parameters of the exact, approximate and accuracy configurable (in four levels) ALUs.

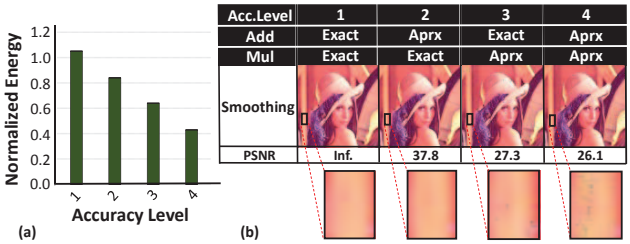


Fig. 10. (a) Normalized energy consumption of an accuracy configurable ALU with four different accuracy levels compared to the exact ALU; (b) quality impact of the accuracy levels for image smoothing filter [10].

TABLE II. DESIGN PARAMETERS OF EXACT, APPROXIMATE, AND ACCURACY CONFIGURABLE ALUS.

| ALU Type | Accuracy Level | Delay [ps] | Area [μm²] | Energy [fJ] | EDP×Area [ps×μm²× fJ ×10⁷] |
|---|---|---|---|---|---|
| Exact | - | 278 | 2319 | 182 | 11.7 |
| Approximate (fixed level) | - | 150 | 1483 | 70 | 1.6 |
| Accuracy Configurable | 1 | 286 | 2458 | 191 | 13.4 |
| | 2 | 251 | | 153 | 9.4 |
| | 3 | 183 | | 116 | 5.2 |
| | 4 | 161 | | 78 | 3.1 |

## B. PAC Composition

Fig. 11 shows the architecture of a PAC composed of a set of exact, approximate, and accuracy configurable ALUs connected with a switch box. The number of ALUs in the PACs and their accuracy type (exact/approximate/configurable) is determined at design time. As shown in Fig. 11, the functionality and the connections of each ALU is configured with 14 bits of the context word. The 5-bit opcode determines the function of the ALU (see TABLE I). The 3-bit $MUX_A$ and $MUX_B$ are used to specify the inputs of the ALU. The 2-bit WR determines the destination of the ALU output. To support OM signals by accuracy configurable ALUs, a m-bit OM field (corresponding to m separate OM signals) should be considered in the context words (see Fig. 11). The overall size of the context register can be customized in design time depending on the number and the type of ALUs in each PAC.

As an example, Fig. 12 shows the design parameters of five different configurations of a PAC, varying the number of exact and approximate ALUs. Results are normalized to the those of a PAC composed of exact ALUs. As shown in this figure, with an increasing number of approximate ALUs in PACs, the overall design parameters are decreased significantly. Those configurations of the PAC which include approximate ALUs (PAC2 to PAC5) have from 34% up to 90% smaller EDP×Area parameter compared to the PAC composed of only exact ALUs (PAC1).

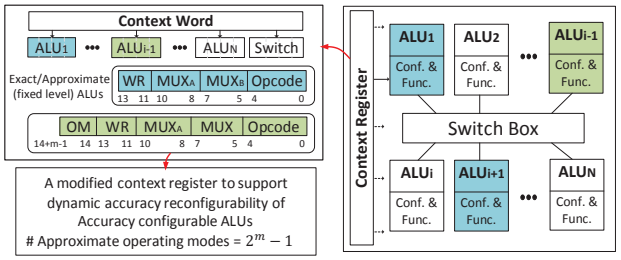


Fig. 11. The architecture of a PAC, which is composed of a set of approximate (blue colored), accuracy configurable (green colored), and exact (white colored) ALUs connected to a switch box.

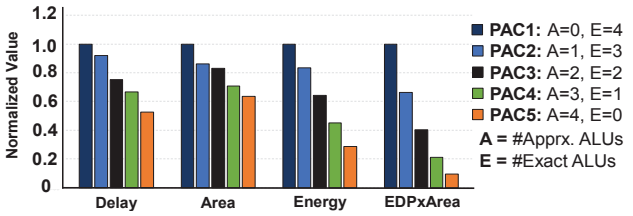


Fig. 12. Normalized design parameters of a PAC with various numbers of exact and approximate ALUs (fixed level) compared to PAC1.

### C. PX-CGRA Architecture

Fig. 13 shows the overall architectural model of PX-CGRA. It includes an array of different PX-CGRA tiles organized in form of heterogeneous approximate CGRA tiles. Each tile is composed of an array of similar PACs connected in a 2-D mesh style to support a specific quality range for a given application. The functionality of PX-CGRA tile selection unit is inspired from [30], which consists of a look up table (LUT) contains entries such as utilization ratio, output quality, and performance/energy constraints. Depending on the output quality constraint (tolerable error) of the application, the proper tile is selected for implementing the application which offers more utilization ratio (energy-saving), while the other unused tiles are power gated.

TABLE III shows the design parameters of the 2×2 PX-CGRA tiles composed of the studied PACs. Since the accuracy configurable ALU imposes some overheads in accuracy level 1 (see TABLE II), we build the PX-CGRA tiles using PACs composed of fixed level approximate ALUs.

TABLE III. DESIGN PARAMETERS OF VARIOUS 2×2 PX-CGRA TILES.

| | PAC Type | Delay [ns] | Area [μm²] | Energy [pJ] | EDP×Area [ns×μm²× pJ ×10¹²] |
|---|---|---|---|---|---|
| PX-CGRA | 1 | 4.73 | 40603 | 3.03 | 581.91 |
| | 2 | 4.35 | 34949 | 2.53 | 385.25 |
| | 3 | 3.55 | 32159 | 1.95 | 222.38 |
| | 4 | 3.15 | 28752 | 1.36 | 123.44 |
| | 5 | 2.48 | 25798 | 0.87 | 55.71 |

## IV. MAPPING PROCESS

Generally, mapping applications onto CGRAs includes scheduling and binding steps [31]. However, for the PX-CGRA, we need an extra step to determine the accuracy level (exact/approximate) of the operations in the given data flow graph (DFG ($V$, $D$) where $V$ and $D$ represent the node set and data dependencies, respectively). This step determines the accuracy operating mode (exact/approximate) of each ADD and MUL nodes of the input DFG with the objective of reducing the energy consumption ($E$) such that the DFG output quality ($Q_O$) meets the predefined output quality constraint ($Q_{Const}$). This optimization problem may be expressed by:

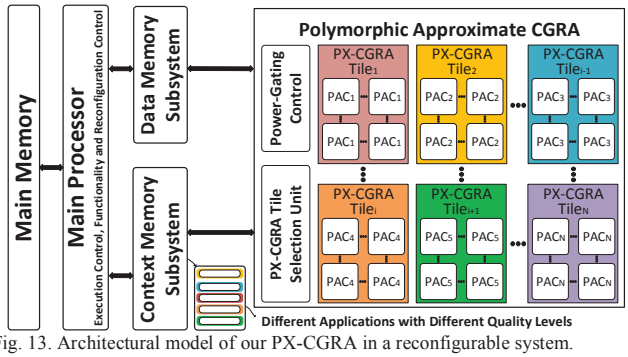


Fig. 13. Architectural model of our PX-CGRA in a reconfigurable system.

Objective:

$$Minimize \sum_{i=1}^{n} E_i \tag{2}$$

Constraint:

$$Q_O \geq Q_{Const}$$

where $n$ is the number of nodes in the given DFG. For the quality evaluation, we use the precision optimization method proposed in [32] which calculates the effect of approximate nodes for a given DFG at the output, based on the Error Sensitivity (ES) of nodes and their error variance ($v$). ES is obtained by [32]

$$ES_i = \frac{\epsilon_{i,o}}{\epsilon_i} \tag{3}$$

where $\epsilon_{i,o}$ and $\epsilon_i$ are error distance at the output of the DFG and the error distance of the i[th] node, respectively. The output error variance ($v_o$) in a given DFG is defined by [32]

$$v_o = \sum_{i=1}^{n} ES_i^2 . v_i \tag{4}$$

In this method, the error variance metric is utilized, since it provides a proper propagation scheme to investigate the effect of the approximate operations (nodes) at the output of a DFG [32]. Similarly, we used the output error variance as the quality metric of our evaluations. In addition, we calculate the energy savings induced by approximation of each node. Afterward, we use an ILP solver to utilize the results of the previous steps to determine the candidate nodes for approximation, which lead to more energy-saving in their approximate operating mode without violating the $Q_{Const}$. For the scheduling step, we suggest to use the conventional list scheduling method [33]. For the binding step, the clique partitioning algorithm (based on the graph coloring algorithm) may be exploited. In this step, similar to [3] the target DFG is divided into sub-graphs that can be bounded to PACs. Note that, the main effort of this step is to improve the utilization ratio (discussed in next section) of PACs [3]. Finally, the context words of each application corresponding the different output quality constraints are stored in the context memory.

## V. RESULTS AND DISCUSSION

Delay, area, and power/energy of state-of-the-art approximate arithmetic units, different configurations of PACs that employ different ALU types, and PX-CGRA tiles have already been presented in previous sections. In this section, we investigate the efficacy of implementing different application benchmarks on the proposed PX-CGRA. To evaluate the PX-CGRA tiles composed of different configurations of the PACs in terms of their resource utilization under different applications, we defined utilization ratio as the average ratio of the number of operating ALUs to the total number of ALUs in PACs. Additionally, our evaluation results include energy consumption and quality degradation. From the five studied configurations of PACs in Section III, PAC2, PAC3, and PAC4 were utilized for implementing the studied application benchmarks.

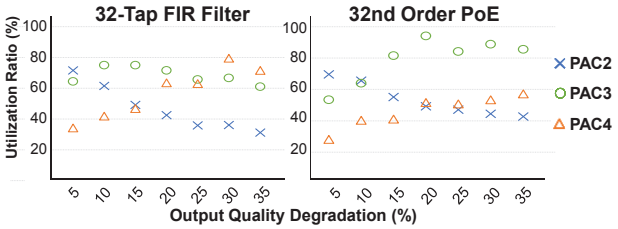


Fig. 14. Utilization ratio of the different configurations of PACs utilized in PX-CGRA under various output qualities.

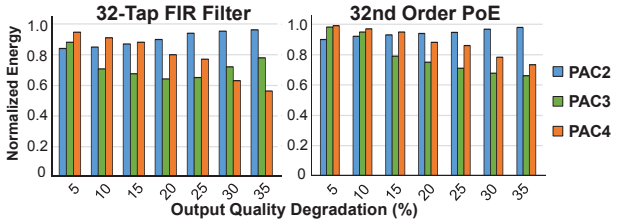


Fig. 15. Normalized energy consumption of PX-CGRA composed of the different configurations of PACs compared to the exact CGRA for different qualities.

In the studies of this section, we ignored PAC1 and PAC5, because they are totally exact and approximate, respectively. Hence, they are not suitable for implementing an application under different qualities. Fig. 14 and Fig. 15 show the utilization ratio and normalized energy consumption of the PX-CGRA, respectively, composed of different configurations of PACs under various output qualities.

As shown in Fig. 14, for both of the studied applications, with decreasing output quality, PAC2 led to the lowest utilization ratio. The reason is that the number of approximate operations is increased at the lower output qualities (see Fig. 3). For different studied output qualities, using PAC2 resulted in, on average, 10% and 6%, energy consumption reductions compared to exact CGRA for the 32-Tap FIR filter and 32$^{nd}$ order PoE, respectively. Also, using PAC3 (PAC4) for the 32-Tap FIR filter and 32$^{nd}$ order PoE at the different output qualities, led to, on average, 28% (21%) and 21% (12%) energy consumption reductions compared to exact CGRA, respectively.

Additionally, results show that the PAC with the highest utilization ratio has the highest energy efficiency compared to the other PACs. The reason is that depending on the resource utilization ratio, the cycles required for performing the application are different, e.g., for lower utilization ratio, more clock cycles are needed to process an application (temporal extension), which lead to lower energy-savings. Therefore, it may be beneficial to use the utilization ratio of tiles for each application at various output quality ranges, and allocate the tile with the most utilization ratio (energy-saving) to the application at run-time. Finally, depending on the tolerable error of the applications, various PX-CGRA tiles composed of different configurations of PACs may be utilized in the overall architecture to provide higher performance and lower energy consumption compared to typical exact-mode CGRAs.

## VI. Conclusion

In this paper, we presented a polymorphic approximate coarse-grained reconfigurable architecture (PX-CGRA). It leveraged approximate arithmetic units to build polymorphic-approximated ALU clusters (PACs), resulting in significant design parameters (delay, area, power/energy) improvement at the cost of constrained accuracy loss. Therefore, for designing efficient PX-CGRA, we have proposed a bottom-up design flow. In addition, the potential of using quality configurable arithmetic units in the architecture of PX-CGRA was studied. We synthesized different configurations of PACs and PX-CGRAs using a 15-nm FinFET technology. Results showed that the PX-CGRA can be employed as an energy-efficient accelerator for error tolerant applications, while retaining the capability of exact computing for error-sensitive/critical applications.

## Acknowledgment

The contributions of Omid Akbari and Muhammad Shafique were partially supported by the German Research Foundation (DFG) as part of the GetSURE project in the scope of SPP-1500 priority program "Dependable Embedded Systems".

## References

[1] L. Liu, et al., "An Energy-Efficient Coarse-Grained Reconfigurable Processing Unit for Multiple-Standard Video Decoding," IEEE Transactions on Multimedia, 17(10): 1706-1720, 2015.
[2] S. Yin, et al., "Joint Modulo Scheduling and Vdd Assignment for Loop Mapping on Dual-Vdd CGRAs," IEEE **TCAD**, 35(9): 1475-1488, 2016.
[3] G. Ansaloni, et al., "Design and Architectural Exploration of Expression-Grained Reconfigurable Arrays," in Proc. Symposium on Application Specific Processors, pp. 26-33, 2008.
[4] G. Ansaloni, et al., "EGRA: A Coarse Grained Reconfigurable Architectural Template," IEEE **TVLSI**, 19(6): 1062-1074, 2011.
[5] M. Karunaratne, et al., "HyCUBE: A CGRA with Reconfigurable Single-cycle Multi-hop Interconnect," IEEE **DAC,** 2017.
[6] R. Koenig, et. al., "KAHRISMA: A Novel Hypermorphic Reconfigurable-Instruction-Set Multi-grained-Array Architecture," IEEE **DATE**, 2010.
[7] C. Liang, et al., "SmartCell: An energy efficient coarse-grained reconfigurable architecture for stream-based applications," EURASIP J. Embedd. Syst., vol. 2009, pp. 1–15, 2009.
[8] M. Shafique, et al., "Invited: Cross-layer approximate computing: From logic to architectures," IEEE **DAC**, 2016.
[9] O. Akbari, et al., "RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder," IEEE **TCAS II**, doi: 10.1109/TCSII.2016.2633307.
[10] O. Akbari, et al., "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers," IEEE **TVLSI**, 25(4): 1352-1361, 2017.
[11] H. Esmaeilzadeh, et al.,, "Dark silicon and the end of multicore scaling", IEEE **ISCA**, pp. 365 –376, 2011.
[12] F. Kriebel, et al., "ASER: Adaptive soft error resilience for reliability heterogeneous processors in the dark silicon era", IEEE **DAC**, 2014.
[13] S. T. Muthukaruppan, et al., "Hierarchical power management for asymmetric multi-core in dark silicon era", IEEE **DAC**, 2013.
[14] K. Swaminathan, et al., "Steep-slope devices: From dark to dim silicon", IEEE **Micro**, 33(5):50–59, 2013.
[15] G. Venkatesh et al., "Conservation cores: reducing the energy of mature computations", **ASPLOS**, pp. 205–218, 2010.
[16] The Standard Cell Library Optimization Company: http://www.nangate.com/.
[17] Q. Xu, et al., "Approximate Computing: A Survey," in IEEE **Design & Test**, 33(1): 8-22, 2016.
[18] S. Mittal, "A Survey of Techniques for Approximate Computing", ACM Computing Surveys (CSUR), 48 (4): article 62, 2016.
[19] V. Gupta, et al., "Low-Power Digital Signal Processing Using Approximate Adders", IEEE **TCAD**, 32(1): 124-137, 2013.
[20] R. Ye, et al., "On reconfiguration-oriented approximate adder design and its application", IEEE **ICCAD**, pp.48-54, 2013.
[21] M. Shafique, et al., "A Low Latency Generic Accuracy Configurable Adder", IEEE **DAC**, 2015.
[22] J. Hu, et al., "A new approximate adder with low relative error and correct sign calculation," IEEE **DATE**, pp. 1449-1454, 2015.
[23] P. Kulkarni, et al., "Trading Accuracy for Power with an Underdesigned Multiplier Architecture", VLSI Design, pp. 346 – 351, 2011.
[24] A. Momeni, et al., "Design and Analysis of Approximate Compressors for Multiplication," IEEE **TC**, 64(4): 984-994, 2015.
[25] A. Chattopadhyay, "Ingredients of Adaptability: A Survey of Reconfigurable Processors," in VLSI Design, vol. 2013, Jan. 2013.
[26] M. Wijtvliet, L. Waeijen and H. Corporaal, "Coarse grained reconfigurable architectures in the past 25 years: Overview and classification," **SAMOS**, 2016.
[27] S. M. A. H. Jafri, et al., "Energy-aware coarse-grained reconfigurable architectures using dynamically reconfigurable isolation cells," **ISQED**, pp. 104-111, 2013.
[28] J. Zhu, L. Liu, S. Yin and S. Wei, "Low-Power Reconfigurable Processor Utilizing Variable Dual VDD," IEEE **TCAS II**, 60(4): 217-221, 2013.
[29] T. Yamamoto, et. al., "Dynamic vdd switching technique and mapping optimization in dynamically reconfigurable processor for efficient energy reduction," in Proc. 7th symp. Reconf. Computing: Arch., Tools and App. Springer, pp. 230–241, 2011.
[30] W. El-Harouni, et al., "Embracing approximate computing for energy-efficient motion estimation in high efficiency video coding," IEEE **DATE**, 2017.
[31] M. Hamzeh, et al., "Regimap: register-aware application mapping on coarse-grained reconfigurable architectures (cgras)," IEEE **DAC**, 2013.
[32] Chaofan Li, et al., "Joint precision optimization and high level synthesis for approximate computing," IEEE **DAC**, 2015.
[33] G. De Micheli, Synthesis and Optimization of Digital Circuits. New York, NY, USA: McGraw-Hill, 1994.