Sustainable Hardware Specialization

*Pranav Dangi[†], *Thilini Kaushalya Bandara[†], Saeideh Sheikhpour[‡], Tulika Mitra[†] and Lieven Eeckhout[‡]

[†]National University of Singapore and [‡]Ghent University, Belgium

ABSTRACT

Hardware specialization is commonly viewed as a way to scale performance in the dark silicon era with modern-day SoCs featuring multiple tens of dedicated accelerators. By only powering on hardware circuitry when needed, accelerators fundamentally trade off chip area for power efficiency. Dark silicon however comes with a severe downside, namely its environmental footprint. While hardware specialization typically reduces the operational footprint through high energy efficiency, the embodied footprint incurred by integrating additional accelerators on chip leads to a net overall increase in environmental footprint, which has led prior work to conclude that dark silicon is not a sustainable design paradigm.

We explore sustainable hardware specialization through reconfigurable logic that has the potential to drastically reduce the environmental footprint compared to a sea of accelerators by amortizing its embodied footprint across multiple applications. We present an abstract analytical model that evaluates the sustainability implications of replacing dedicated accelerators with a reconfigurable accelerator. We derive hardware synthesis results on ASIC and CGRA (a representative reconfigurable fabric) for chip area and energy numbers for a wide variety of kernels. We input these results to the analytical model and conclude that reconfigurable fabric is more sustainable. We find that as few as a handful to a dozen accelerators can be replaced by a CGRA. Moreover, replacing a sea of accelerators with a CGRA leads to a drastically reduced environmental footprint (by a factor of 2.5× to 7.6×).

KEYWORDS

Coarse Grained Reconfigurable Arrays, Sustainable Computing

1 INTRODUCTION

The end of Dennard scaling [14] has dramatically changed how we design processors. Increased power density as we transition to new chip technology nodes leads to dark silicon [17, 46], which means that we cannot power on the entire chip while keeping thermals within a safe operating range. Hardware specialization enables continuous performance scaling despite dark silicon by executing specific kernels on dedicated hardware accelerators. Powering on an accelerator provides high performance when needed; when not in use, an accelerator is powered off to save power. The advent of dark silicon has created a flurry of work in hardware specialization, sometimes referred to as the 'golden age for computer architecture' [26], with accelerators for machine learning, video coding and decoding, image signal processing, security encryption/decryption, etc. In fact, a modern-day computer is a system-on-chip (SoC) in which general-purpose CPU and GPU cores are complemented with a sea of (multiple tens of) domain-specific accelerators (DSAs) [27]. This is the case across the computing spectrum from mobile application processors (e.g., Qualcomm Snapdragon [41]) to laptop and desktop processors (e.g., Apple M2 [3], Intel Sapphire Rapids [30]) and server processors (e.g., AMD EPYC [2], IBM Telum [29]). The number of DSAs integrated on chip has steadily increased over time: Shao et al. [44] report that for Apple SoCs, the number of DSAs has increased from less than 10 in the A4 (2010) to more than 40 in the A12 (2018).

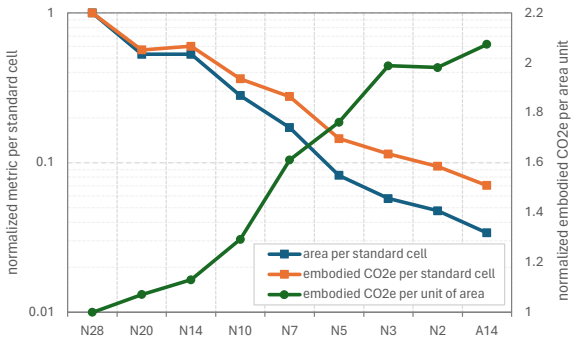


Figure 1: Chip area and embodied footprint per standard cell (left axis) and embodied footprint per unit of chip area (right axis) for various chip technology nodes normalized to 28 nm [21, 35]. *Keeping chip area constant to accommodate dark silicon comes at the cost of an increased embodied footprint.*

Fundamentally, dark silicon trades off chip area for power efficiency, i.e., hardware specialization boosts performance within the available power budget by powering on hardware resources only when needed. Dark silicon thus comes at the cost of additional transistors to implement the various DSAs on chip. Fortunately, transistors have become exponentially cheaper over time thanks to Moore's Law [9], which makes dark silicon economically viable (even today, despite Moore's Law slowing down). In other words, hardware specialization offers continuous performance scaling without increasing the area (and cost) per chip.

There is a severe downside to dark silicon though, which has been largely ignored, namely its environmental footprint. With information and communication technology (ICT) being responsible for 2.1% to 3.9% of the global greenhouse gas (GHG) emissions world-wide [19] — currently on par with the aviation industry, and it is projected to continue to grow [15]. Figure 1 (vertical axis on the left) reports chip area and the embodied carbon footprint (i.e.,

*Equal Contribution.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICCAD '24, October 27–31, 2024, New York, NY, USA

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1077-3/24/10.
https://doi.org/10.1145/3676536.3676777

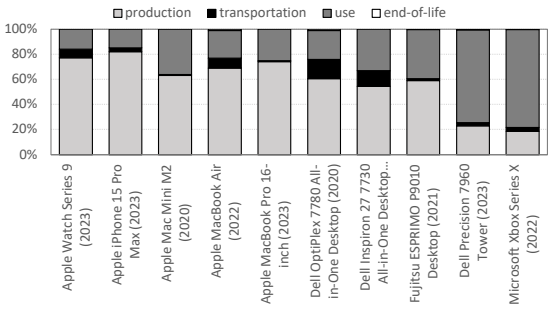


Figure 2: Breakdown of the environmental footprint in production, transportation, use and end-of-life processing. *The embodied footprint dominates for most computing devices.*

environmental footprint due to manufacturing, measured in CO2e equivalent) per standard cell for nine available and projected technology nodes as provided by imec [21, 35], normalized to 28 nm. Continuous advancements in chip technology have dramatically reduced chip area as well as the embodied footprint per standard cell. However, the embodied footprint does not reduce at a similar pace as chip area due to increased complexity in manufacturing, i.e., more processing steps leading to increased energy consumption and GHG emissions. This implies that for a constant unit of chip area, the embodied footprint for logic has substantially increased, see Figure 1 (vertical axis on the right). In other words, dark silicon comes at the cost of a substantially increased embodied footprint.

The question now is whether the increase in embodied footprint is offset by the decrease in operational footprint due to device use during its entire lifetime. While hardware specialization typically reduces energy consumption (due to higher performance and/or lower power) when in use, thereby reducing the operational footprint, the embodied footprint for integrating the DSA on chip has to be incurred regardless. This suggests that dark silicon only leads to a net reduction in environmental footprint if the DSAs are frequently used, which seems to contradict the notion of dark silicon. This has led prior work to suggest that dark silicon is not sustainable from an environmental perspective [10, 16]. In this work, we explore a sustainable alternative to dark silicon, namely hardware specialization through reconfigurable logic. The intuition that underpins this work is that a fabric that can be dynamically reconfigured or reprogrammed across applications can possibly achieve the best of both worlds, i.e., yield high performance at high power efficiency without incurring the vast embodied footprint of dark silicon. Replacing the sea of DSAs by a reconfigurable fabric with overall smaller area has the potential to drastically reduce the embodied footprint. On the flip side, a reconfigurable fabric is less efficient and leads to increased energy consumption and thus a higher operational footprint. The fundamental question hence is whether the decrease in embodied footprint offsets the increase in operational footprint — if so, reconfigurable logic is a more sustainable design paradigm than dark silicon.

To investigate the opportunity for sustainable hardware specialization through reconfigurable logic, we formulate an abstract analytical model to compare the environmental footprint of a sea of DSAs versus a reconfigurable fabric. The model determines the *critical DSA count (CDC)* or the number of DSAs the reconfigurable fabric needs to replace for the latter to be more sustainable. Despite the model being parameterized to account for inherent data uncertainty, several interesting insights can be obtained. First, we find that — contrary to common belief — area efficiency is more important than energy efficiency for a DSA to be sustainable. Second, CDC decreases (1) with an increasing contribution of the embodied footprint in the overall environmental footprint of a device, (2) with a decreasing area (and to a lesser extent the energy) efficiency of a DSA relative to the reconfigurable fabric, and (3) with a decreasing degree of DSA concurrency during application execution.

To quantify the area and energy efficiency of a DSA versus a reconfigurable fabric, we map a total of eight widely used application kernels to DSA as well as reconfigurable fabric while assuming iso-performance implementations. We consider standard-cell ASIC implementations for the DSA, and coarse-grain reconfigurable array (CGRA) [22, 31, 39, 40] for the reconfigurable fabric. We report that a DSA incurs on average 0.27× and 0.31× less chip area and energy compared to CGRA, respectively. This suggests that for embodied footprint dominated systems, the carbon footprint of CGRA equals that of 4 to 5 DSAs. In a system comprising a total of 40 DSAs, this reduces the environmental footprint by a factor of 2.5× to 7.6×. The overall conclusion is that CGRA is a sweet spot, paving a way towards sustainable hardware specialization.

2 ANALYTICAL CARBON MODELING

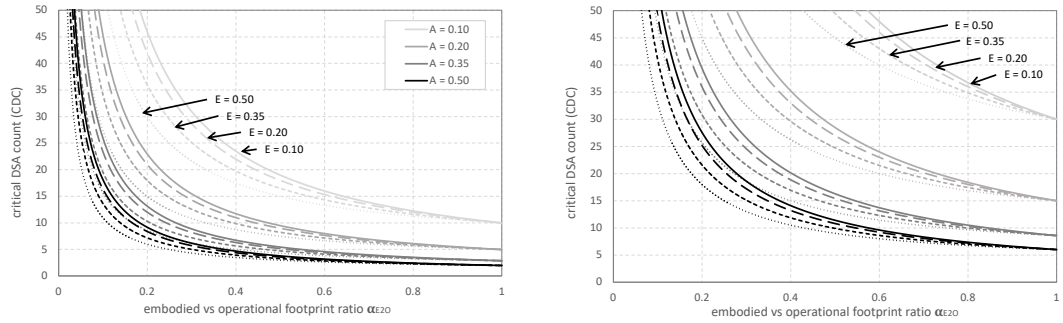
2.1 Background

The environmental footprint for a computing device consists of two major contributors [33]: (1) the *embodied footprint* due to raw material extraction, manufacturing, assembly, transportation, end-of-life processing, and (2) the *operational footprint* due to device use during its entire lifetime. Figure 2 reports the breakdown of the environmental footprint for various computing devices including a smart watch, smartphone, laptop, medium-end and high-end desktop computers as well as a gaming console from different vendors including Apple, Dell, Fujitsu and Microsoft obtained from their corresponding product environmental reports.

The environmental footprint is dominated by the embodied footprint for many computing devices, especially mobile devices such as laptops (embodied footprint accounts for 70–75% of total) and smartphones and smart watches (80–85%). In contrast, high-end desktop computers and gaming consoles are mostly dominated by the operational footprint, i.e., the embodied footprint contributes for 20–25%. Medium-end desktops are somewhere in the middle for which the embodied footprint ranges between 55–60%. This is in line with a prior survey by Gupta et al. [25] which concluded that battery-operated consumer electronics such as smart watches, phones, tablets and laptops are mostly dominated by the embodied footprint, whereas always-connected devices such as desktops and gaming consoles are mostly dominated by the operational footprint.

2.2 Abstract Carbon Model

Analyzing the environmental footprint of a chip is difficult due to inherent data uncertainty [16, 24] resulting from semiconductor manufacturing industry secrecy, unknown or poorly documented material supply chains, hard to anticipate product usage and lifetimes, etc. To make things even more complicated, rebound effects



(a) serial DSA execution ($n = 1$) **(b) concurrent DSA execution ($n = 3$)**

Figure 3: Critical DSA count (CDC) as a function of α_{E2O}, A and E: assuming (a) serial DSA execution ($n = 1$) and (b) concurrent DSA execution ($n = 3$). *CDC decreases with increasing α_{E2O}, A and E, suggesting that a reconfigurable fabric has the potential to be more environmentally friendly than a sea of DSAs if the embodied footprint is substantial and if the area (and energy) efficiency gain of dedicated DSAs is limited relative to the reconfigurable fabric.*

may turn per-device efficiency improvements into increased usage and deployment, ultimately increasing rather than decreasing the overall environmental footprint, referred to as Jevons' paradox [1].

The degree of uncertainty calls for an abstract parameterized analytical model based on first-order principles. We rely on the same proxies used in the FOCAL model [16] to quantify the embodied and operational footprint, and a parameter α_{E2O} ($0 \leq \alpha_{E2O} \leq 1$) to weigh the relative importance of the embodied and operational footprint. The proxy for the embodied footprint is chip area, while the proxy for the operational footprint is energy consumption.[1] As reported previously, the embodied-to-operational ratio α_{E2O} varies across computing devices: 0.7–0.85 for battery-operated devices (watches, smartphones, laptops) versus 0.2–0.6 for always-on devices (desktops, gaming consoles).

We now construct an abstract analytical model to compare the environmental footprint of a sea of DSAs versus the reconfigurable fabric. We initially assume that only a single DSA is used at any given time; we relax this assumption later.

Serial DSA usage. We first assume that only one DSA is active at any time, i.e., DSAs are used serially. The reconfigurable fabric incurs a smaller environmental footprint than a sea of DSAs if

$$\alpha_{E2O} \cdot N \cdot A + (1 - \alpha_{E2O}) \cdot E > 1, \tag{1}$$

with N the number of DSAs, and A and E the (average) chip area and energy consumption per DSA relative to the reconfigurable fabric, respectively. In other words, if the weighted embodied footprint of integrating N DSAs (first term) plus the weighted operational footprint of using a DSA (second term) is larger than one, this implies that the sea of DSAs incurs a larger environmental footprint than a reconfigurable fabric with a normalized environmental footprint equal to one. The reconfigurable fabric is assumed to be large enough to accommodate the largest kernel, i.e., it has sufficient resources available to execute any single kernel in its entirety.

Multiple concurrent DSAs. While some applications exercise a single DSA at a time, others exercise multiple concurrently, for example, in a streaming fashion whereby one DSA produces input for the next DSA to process. Many modern-day applications exercise several different DSAs concurrently [27]. The number of concurrently exercised DSAs is typically small though, at most a handful, see also Table 1 as reported by several prior works [7, 27], which — fortunately — aligns well with the restrictions imposed by dark silicon. We extend the serial DSA usage model to concurrent execution as follows. The reconfigurable fabric is more sustainable than a sea of DSAs if

$$\alpha_{E2O} \cdot N \cdot A + (1 - \alpha_{E2O}) \cdot n \cdot E > n, \tag{2}$$

with n the number of DSAs exercised concurrently during runtime. This model considers that the operational footprint of the DSAs equals $n \cdot E$. The model further conservatively assumes that the footprint of the reconfigurable fabric is n times larger if it needs to execute n kernels concurrently. This is a conservative estimate because a reconfigurable fabric can inherently reuse hardware resources across kernels, such as sharing memory. Moreover, a reconfigurable fabric designed to accommodate the largest kernel is typically able to spatially accommodate multiple smaller kernels simultaneously, thereby reducing both the area overhead and the energy overhead compared to the linear scaling assumption. This inherent resource-sharing and spatial mapping flexibility of the reconfigurable fabric allows for more efficient execution of multiple kernels, leading to a lower overall area/energy overhead than the conservative estimate used in the model.

Table 1: Number of (concurrent) kernels per application.

Reference	No. kernels	Application domain
Hill and Reddi [27]	2 – 3	camera applications
Bleier et al. [7]	1 – 3	wearable applications
Bleier et al [8]	1 – 3	olfactory computing
Karageorgos et al. [28]	1 – 3	brain-computer interface

Critical DSA count. We now rework the above inequality to:

$$N > n \cdot \left(\frac{E}{A} + \frac{1 - E}{\alpha_{E2O} \cdot A} \right) = CDC, \tag{3}$$

while defining the right-hand side of the inequality as the *critical DSA count (CDC)*. This leads to the overall conclusion that if the number of DSAs N is larger than CDC, the reconfigurable

[1] This assumes a fixed-work scenario. The FOCAL model also includes a fixed-time scenario for which the proxy for the operational footprint is power consumption. This is not considered here as we assume iso-performance design alternatives in this work.

fabric incurs a smaller environmental footprint than the sea of DSAs. Because of the conservative assumptions made in terms of DSA/kernel concurrency, we believe that CDC computed using the above formula is an overestimation.

2.3 Analysis and Discussion

Despite the simplicity of the model, several interesting conclusions can be obtained by analyzing sensitivity of *CDC* with respect to A, E and α_{E2O}, see Figure 3 assuming (a) serial DSA execution and (b) concurrent DSA execution. A and E are in the range of empirically observed values, see also Section 4, with $A = E = 0.35$ close to the average. $n = 3$ is typical for DSA concurrency, see Table 1.

First, CDC is a monotonically decreasing curve as a function of α_{E2O}, and converges to n/A as the embodied footprint dominates, i.e., $\alpha_{E2O} \rightarrow 1$. This implies that *a reconfigurable fabric is especially environmentally friendly for computing devices that are dominated by the embodied footprint, such as battery-operated devices (watches, smartphones, laptops).* Inversely, a sea of DSAs is more environmentally friendly for devices that are dominated by their operational footprint. Which specific design option (a sea of DSAs versus a reconfigurable fabric) is more environmentally friendly depends on the actual difference in area and energy efficiency and the ratio of the embodied versus operational footprint.

Second, area efficiency A has a more significant impact on CDC than energy efficiency E. Indeed, the curves are primarily grouped by different values for A and not E. This implies — contrary to common belief perhaps — that *optimizing the area efficiency of a DSA is more important than optimizing its energy efficiency to reduce its overall environmental footprint.*

Third, the larger the relative area A (and energy consumption E) of the DSA compared to the reconfigurable fabric, the smaller the CDC. In other words, *if the area and energy reduction of the DSA is relatively small compared to the reconfigurable fabric, the latter leads to a smaller environmental footprint.* This suggests that a reconfigurable fabric is particularly beneficial for sustainability if the area gains through a dedicated accelerator are limited. In contrast, if the DSA is substantially more area-efficient, it is potentially more environmentally friendly than a reconfigurable fabric.

Fourth, *a reconfigurable fabric tends to be increasingly more environmentally friendly compared to a sea of a DSAs with limited DSA concurrency.* More concretely, for an embodied-dominated device ($\alpha_{E2O} = 0.8$ and $A = E = 0.35$), a reconfigurable fabric incurs a smaller environmental footprint than at most a handful DSAs (serial execution) and a dozen DSAs (concurrent execution). For an operational-dominated device ($\alpha_{E2O} = 0.25$ and $A = E = 0.35$), a reconfigurable fabric needs to replace 8 DSAs (serial execution) to 25 DSAs (concurrent execution) for it to be more sustainable. With current SoCs featuring more than 40 DSAs [27], there appears to be compelling opportunity to improve processor sustainability for replacing the sea of DSAs with a reconfigurable fabric.

3 METHODOLOGY

We now quantify the typical area and energy efficiency of a DSA relative to a reconfigurable fabric, i.e., A and E, for a set of widely used kernels. We first compare the relative area and energy efficiency of dedicated ASIC-based DSAs (as typically implemented in modern-day SoCs) versus a reconfigurable fabric for realizing accelerators. We choose coarse-grain reconfigurable array (CGRA) as the representative reconfigurable fabric. CGRAs are composed of an array of coarse-grained processing elements (PEs) connected via an on-chip interconnect. Both the PEs and the interconnect can be configured to implement different functions and data paths, enabling the acceleration of diverse workloads on a single hardware fabric through software. Our selection of CGRAs is motivated by the fact that CGRAs occupy the middle ground between the flexibility of FPGA versus the efficiency of ASIC. CGRAs provide superior area and energy-efficiency compared to FPGAs [40] as their basic building blocks for reconfiguration are coarse-grained functional units compared to the individual logic gates of the FPGAs. While not as efficient as the ASICs, CGRAs are flexible to provide excellent acceleration for a wide range of computational kernels.

3.1 Workloads

To analyze the carbon footprint between specialization and generalization, we analyze the DSAs and reconfigurable fabrics using a set of workloads from the MachSuite benchmark suite [42], see Table 2. These workloads encompass a wide range of behaviors of frequently accelerated kernels, including applications from linear algebra in machine learning (*GeMM, KNN, Conv2D*), image processing (*Stencil3D*), speech recognition (*Viterbi*), signal processing (*FFT, FIR*), and security (*AES Encryption*). These tasks exhibit a range of memory access patterns, control logic requirements, and arithmetic intensity. Additionally, these kernels constitute a significant portion of the EarBench [7] benchmark suite, which comprises applications requiring acceleration in hearing aids, earphones, smart glasses, and similar devices that are dominated by embodied footprint. For realistic workload sizes related to these kernels, we use input data dimensions from the Visual Wake Words Challenge [13], which is part of the TinyML Perf benchmark suite [4].

3.2 CGRA Modeling

We design a clustered spatio-temporal CGRA architecture in RTL with an 8x8 processing element (PE) grid, as illustrated in Figure 4 which is modeled after generic contemporary CGRAs [11, 31, 39]. Each PE includes a dedicated 32-bit ALU, router/crossbar, and configuration memory. We derive the area and power consumption of the CGRA architecture by synthesizing it on a UMC 40 nm technology node at 100 MHz frequency. We map the benchmark suite's kernels onto the CGRA using Morpher [48], an open-source specialized compiler for CGRAs, to obtain the performance estimates. To minimize dynamic power consumption, we use clock gating to disable unused PEs based on the mapping configurations of each kernel. The PEs are connected to a 32-bank memory with enough capacity (256 KB) to store input and intermediate data for the kernel with the highest data memory requirements in the suite, namely *Stencil3D*, see also Table 1.

3.3 DSA Modeling

To ensure a fair comparison, we explore iso-performance design points for the ASIC-based DSAs for each kernel against the CGRA. It is computationally expensive to identity the iso-performance ASIC-based design point at RTL level. We use Aladdin [43], a modeling

Table 2: Application kernels used in the evaluation.

App. Kernel	Domain	Description	Memory (in KB)
GeMM	Machine Learning	General Matrix Multiplication 32x32 tile size, 96x96 input size	108
KNN		K-nearest neighbour 16 maximum neighbours	22
Conv2D		2D convolution Filter size of 3x3, input size 96x96	72
Stencil3D	Image Processing	3D stencil calculation with data size 16x32x32	256
Viterbi	Speech Recognition	Viterbi algorithm 64 hidden states 32 observations	52
FFT	Signal Processing	128-pt fast Fourier transform	1.5
FIR		32-tap FIR filter	108
AES Encryption	Security	Rijndael ciphers with 16B block size	0.5

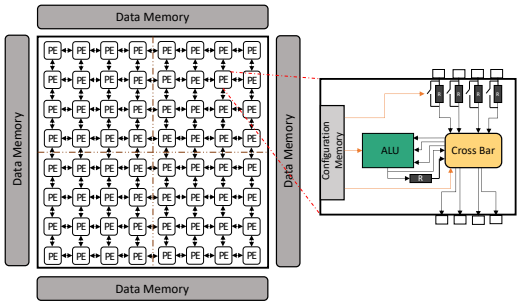


Figure 4: The modeled CGRA architecture.

tool that allows us to quickly explore a variety of ASIC design points for each kernel, showcasing a range of power, performance and area characteristics without the need for extensive RTL re-writing.

Aladdin estimates the area and power of the accelerators according to its calibrated data at a 40 nm technology node, aligning with the CGRA modeling. For consistency, we model the DSAs at 100 MHz and limit each DSA to 32 SRAM banks, matching the CGRA's maximum on-chip traffic capacity. The on-chip memory requirements of each DSA vary according to the kernel's input and intermediate data storage needs. The DSA for the *Stencil3D* kernel requires the largest data memory, equivalent to that of the CGRA.

We leverage Aladdin to pinpoint design points that closely mirror CGRA performance while prioritizing area and power optimization. This results in DSAs that deviate from their corresponding CGRA kernel implementations by an average of 10% in performance.

The selection of our design points is premised on the CGRA (and consequently, the DSAs) adequately meeting the application's performance requirements before optimizing area and power characteristics. Increasing performance further will require higher hardware resources (area, power) in the accelerator architectures. However, it is crucial to assess whether the increase in resource usage translates to a uniform performance gain for both the DSAs and the CGRA.

To evaluate the performance scaling of the DSAs and CGRA with additional area and power, we systematically model the architectures with doubled hardware resources. First, we double the available on-chip memory bandwidth allocated to both the architectures. Then, in the case of CGRA, we double the number of PEs. The CGRA compiler maps to each PE cluster individually, resulting in a 2× performance improvement as the number of clusters doubles. This proportional scaling is shown in Figure 5.

For each kernel, similar to the baseline 1× DSA performance point (described previously and used throughout this text), we now

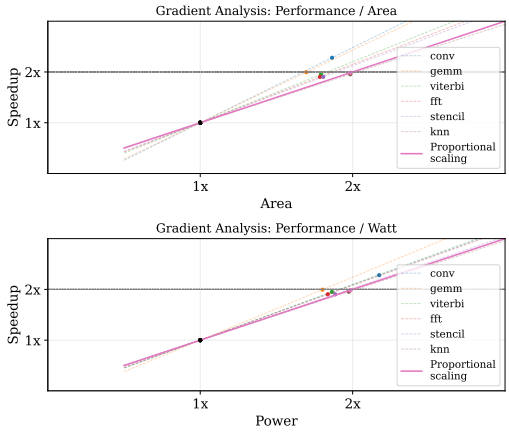


Figure 5: Relationship between performance and resource scaling. *The proportional scaling slope indicates performance versus resources for CGRA. The rest of the lines represent performance versus resource scaling for DSAs.*

find a new, roughly 2× DSA performance point while optimizing area and power using Aladdin. Subsequently, we plot the new point with the corresponding resource utilizations. We could not find DSA design points for *AES* and *FIR* that are close to 2× performance.

The robustness of our area and power comparison between DSAs and CGRA relies on a linear relationship between performance and resource scaling. In reality, we observe a steeper increase in performance versus resources for the DSA kernels (slope > 1). Specifically, the median performance versus area slope stands at 1.2, and the median performance versus power slope is 1.09. Aladdin's estimation methodology incorporates optimizations tailored to the specific input data and kernel characteristics, which may contribute to the more favorable performance scaling observed for the DSAs. This contrasts with earlier studies on simulators and real-world accelerators [12, 37], which show that performance gains do not always match up with the increase in on-chip resources linearly. Nonetheless, the observed deviations from the proportional scaling line are within a reasonable range.

Overall, performance scales almost proportionally with hardware resources (i.e., chip area and power) for both DSA and CGRAs. The near-proportional scaling observed across various performance levels suggests that our analysis and conclusions are robust and applicable across a range of performance targets, rather than being limited to a specific operating point. Consequently, the specific iso-performance data point considered in this work does not impact the generality of our findings. We could have chosen a different iso-performance data point for our modeling, and the conclusions would remain largely similar.

4 EVALUATION

The abstract analytical model to compare the environmental footprint of a sea of DSAs versus the CGRA depends on the area A and energy efficiency E of the DSAs relative to the CGRA. As area and energy here serve as proxies for the embodied and operational footprint, it is important to study their impact at an individual level before considering their impact on sustainability.

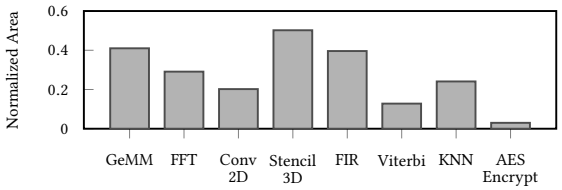


Figure 6: Chip area of dedicated DSAs normalized to 8x8 CGRA. *A DSA incurs on average* 0.27× *less area compared to a CGRA with fixed area (not all kernels utilize full CGRA resources.)*

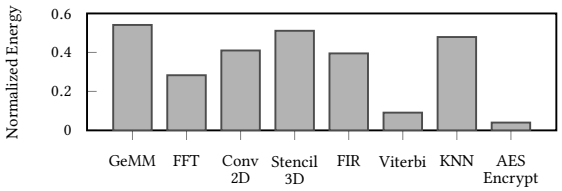


Figure 7: Energy consumption of dedicated per-kernel DSAs normalized to the CGRA. *A DSA consumes on average* 0.31× *less energy compared to a CGRA.*

4.1 Chip Area

There has been a lot of effort in the hardware design community on improving device energy efficiencies. As aforementioned, chip area, serving as a proxy for the embodied footprint, turns out to be the more significant metric affecting carbon emissions for computing devices dominated by their embodied footprint (i.e., $\alpha_{E2O} \geq 0.5$). Figure 6 reports chip area for the various DSAs relative to the CGRA. The DSAs consume between 0.03× and 0.55× compared to the fixed 8x8 CGRA area, with an average of 0.27× across all kernels.

CGRAs feature a fixed number of PEs and, consequently, a fixed number of Arithmetic Logic Units (ALUs) for computations. ALUs are versatile and capable of executing various operations such as multiplication, accumulation, shift, and bitwise operations. It is important to note that the area required for a multiplication operation significantly exceeds that of a bitwise operation. Further, CGRAs have complex, resource-expensive crossbars and routers to handle various kinds of dataflows. This observation aligns with the findings in Figure 6 and Figure 7, where DSAs involving a lot of multiply-accumulate operations, like *GeMM, Stencil3D* and *FIR*, exhibit significantly higher area and energy utilization compared to others. In turn, this also results in noticeably lower resource consumption for DSAs like *AESEncrypt* and *Viterbi*, which primarily rely on bitwise operations, lookups, or iterative updates.

4.2 Energy Consumption

The total energy consumption of the architecture for the duration of its operation serves as a proxy for the operational footprint in the model. Our analysis in Figure 7 indicates that the DSAs consume between 0.03× to 0.49× less energy compared to the CGRA, with an average of 0.31× across all kernels.

For compute-intensive kernels with a high density of multiply-accumulate operations, such as *GeMM*, *Stencil3D* and *FIR*, DSAs consume slightly more than half the energy compared to the corresponding CGRA implementation of those kernels. Conversely, for tasks with irregular loops like *Viterbi* and complex routing like *FFT*,

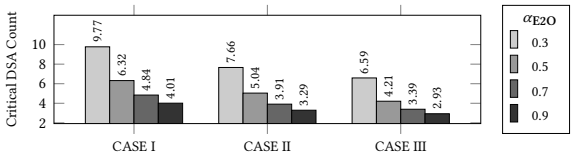


Figure 8: CDC for scenarios I, II and II for different values of α_{E2O}; no concurrency ($n = 1$). *CDC decreases with increasing α_{E2O} (higher contribution of embodied footprint) and decreasing area efficiency of the DSAs relative to the CGRA (increasing A corresponding to scenarios I through III).*

DSAs demonstrate notably lower energy consumption. This advantage stems from the challenges CGRAs face in efficiently routing and mapping these types of applications onto their architecture. Furthermore, the CGRA's inherent flexibility becomes overkill for such tasks with moderate computational demands. The reconfigurability and large ALUs of the CGRA result in significantly higher energy usage compared to DSAs specifically designed for these applications without the extra features.

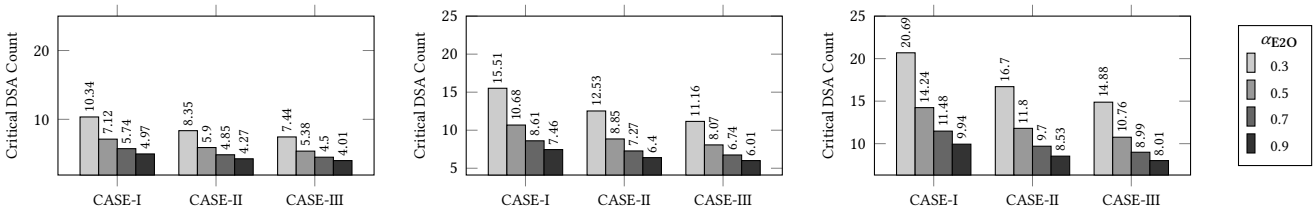
4.3 Critical DSA Count

We now explore the potential for replacing DSAs with a CGRA from a sustainability perspective. If the number of DSAs (i.e., N) surpasses the Critical DSA Count (*CDC*), then the CGRA becomes the preferable and more sustainable option.

The CDC demonstrates considerable variation influenced by the area, performance, and power characteristics of the DSAs anticipated to be substituted by a CGRA. Considering this, we undertake a comprehensive analysis of various scenarios as case studies, aimed at evaluating the relative impact of DSA types and their characteristics. Through these case studies, we seek to provide insight into the decision-making process and feasibility considerations associated with transitioning from a sea of accelerators to a CGRA. We consider the following scenarios:

- **CASE-I: All DSAs.** In this scenario, we examine the CDC when all DSAs are potentially up for replacement by a CGRA. This scenario works without detailed insights into the specific characteristics of individual DSAs.
- **CASE-II: All DSAs minus AES.** Here we analyze the CDC in a scenario where we exclude only the smallest outlier kernel (*AESEncrypt*) from consideration for replacement. This approach proposes a case for leaving the smallest outlier kernel as an individual DSA on the chip (because it consumes miniscule area) while replacing all other kernels.
- **CASE-III: All DSAs minus AES/Viterbi.** Finally, we extend CASE-II to exclude the two small outlier kernels (*AESEncrypt* and *Viterbi*) from consideration for replacement. This approach prioritizes the replacement of larger kernels with a CGRA while retaining all smaller kernels on the chip as individual DSA accelerators with minimal impact on area.

Figure 8 reports CDC for the three cases while considering different values of α_{E2O}. Our analysis focuses mainly on $\alpha_{E2O} \geq 0.5$ which is the region where embodied carbon starts dominating over operational. Battery-operated devices typically fall in the region of $0.7 \leq \alpha_{E2O} \leq 0.9$. We assume serial DSA execution for now; we consider DSA concurrency in the next subsection.



(a) $n = 2$ (b) $n = 3$ (c) $n = 4$

Figure 9: Sensitivity of CDC with respect of DSA characteristics (CASE I, II, III), α_{E2O} (legend) and kernel concurrency (subfigures (a), (b) and (c)). *CDC decreases with (1) reduced area and energy efficiency of the DSA (from CASE-I to CASE-III), (2) increased weight of the embodied footprint (α_{E2O}) in the total footprint, and (3) decreasing kernel concurrency (n).*

CASE-I: Figure 8 demonstrates that even under the most conservative conditions for battery-operated devices ($\alpha_{E2O} = 0.7$), merely 4–5 accelerators already account for a carbon footprint equivalent to that of a CGRA. As this case overlooks the specifics of individual DSA characteristics, it serves as a valuable approach for a blanket replacement of all DSAs with a CGRA on a chip. Furthermore, its applicability extends across a broader spectrum of scenarios for sustainability-focused architectural decisions.

CASE-II: As depicted in Figure 6, the dedicated DSA for *AESEncrypt* kernel occupies an extremely small footprint compared to the other DSAs. Simply excluding this kernel from consideration for replacement leads to a substantial reduction in CDC. Figure 8 reports that the CDC for CASE-II in the region dominated by embodied emissions falls below 4. Just the exclusion of one small kernel makes a noticeable difference in the replaceability of DSAs with a reconfigurable fabric, emphasizing the need for insight into individual DSA characteristics before swapping them out for a CGRA.

CASE-III: The analysis in Case II underscores the importance of further studying the impact of excluding smaller DSAs from the replacement pool. DSAs corresponding to the *Viterbi* and *AES Encrypt* kernels are the small outlier kernels within the benchmark suite, as shown by the earlier energy and area comparisons. By targeting only the larger-footprint kernels for replacement, we observe a further reduction in CDC. In fact, for $\alpha_{E2O} = 0.9$, the CDC dips below 3. This suggests that sustainability benefits could be realized by replacing as few as three DSAs with a CGRA architecture. The analysis emphasizes the importance of selecting the appropriate set of dedicated DSAs for replacement with a reconfigurable fabric to minimize the carbon footprint.

4.4 CDC for Concurrent Execution

For the above analysis, we focused on applications with only one active DSA ($n = 1$), which suits applications requiring sequential execution of various kernels. However, when $n > 1$, i.e., applications where multiple DSAs are required to be active simultaneously, the area or number of CGRAs must increase to accommodate multiple kernels while maintaining consistent performance. The worst-case scenario occurs when each kernel utilizes 100% of the CGRA resources, necessitating a proportional area increase by a factor n. However, most kernels do not fully utilize the computational resources of the CGRA. Out of the kernels considered in this study, *GeMM* and *FIR* utilize 100% of the compute resources, while resource utilization is less than 50% for *Conv2D*, *Stencil3D*, *Viterbi*, and *AES Encryption*, resulting in an average utilization of 64%.

This allows multiple kernels to utilize the CGRA fabric simultaneously. To analyze the sustainability advantages for concurrent kernels, we scale up the CGRA footprint according to this average utilization to accommodate n kernels in parallel. This scaling up enables low resource usage kernels to compensate for higher resource kernels when running in parallel on CGRA. For Case I and $n = \{2, 3, 4\}$, the CGRA requires a scaling up factor of $n' = \{1.3\times, 1.9\times, 2.6\times\}$ for the CGRA and DSAs to provide identical performance. n' values are slightly higher for Case II and Case III.

Figure 9 demonstrates CDC across various levels of concurrency for the three scenarios outlined in Section 4.3. Each graph compares CDC for four different values of α_{E2O}, with $\alpha_{E2O} = 0.3$ representing operational-footprint dominance and $\alpha_{E2O} = 0.9$ representing embodied-footprint dominance. The number of active DSAs increases in correlation to concurrency, increasing CDC for all α_{E2O} values. In the case of $\alpha_{E2O} = 0.7$, where embodied carbon emissions start to dominate, the reconfigurable fabric would be equivalent to 9 to 12 DSAs in carbon emissions for these cases.

Furthermore, the reconfigurable fabric enhances data locality by mapping the data-dependent accelerators closer and sharing memory storage while DSAs necessitate frequent data transfers between accelerators. This can lead to a further increase in the energy consumption of the DSAs executing concurrent kernels, which is not quantified in this result.

4.5 Environmental Footprint Savings

While the Critical DSA Count provides valuable insights into the decision-making process for replacing a sea of DSAs with a CGRA, we aim to quantify the carbon footprint reduction associated with replacing a sea of (say tens of) DSAs on a chip with multiple CGRAs, while considering concurrency and performance requirements.

Modern SoCs have upwards of 40 DSAs on a single chip [44], however, strict thermal constraints often prevent the simultaneous use of all these accelerators on-chip, resulting in dark silicon [46]. By combining this understanding with knowledge about the anticipated concurrency levels of kernels in applications, we can strategically replace the array of tens of accelerators with a smaller set of CGRAs. This approach not only optimizes resource utilization but also enhances sustainability by mitigating dark silicon and reducing

the carbon footprint. Our analysis assumes a representative chip with 40 DSAs following prior work and concurrency of up to 5.

Table 3: Carbon footprint savings by replacing DSAs with CGRA at different concurrency levels.

Concurrency (n)	Carbon Footprint Improvement	
	Avg Util ($n' < n$)	100% Util ($n' = n$)
1	-	7.60×
2	6.10×	3.84×
3	4.12×	2.59×
4	3.12×	1.97×
5	2.53×	1.59×

Table 3 quantifies the improvement in carbon footprint achieved by using CGRAs instead of a sea of DSAs. CGRAs achieve between 2.53 × and 7.6 × less carbon footprint when considering the average utilization of the hardware resources. Here, n denotes the expected concurrency level in the application domain for the chip, while n' represents the scaling-up factor for a CGRA to accommodate such concurrency in applications with average resource utilization (see Section 4.4). For a chip where only sequential execution is anticipated in applications, replacing all DSAs with CGRAs leads to a substantial 7.6× improvement in carbon footprint. This saving over a sea of accelerators decreases as we start working with applications requiring higher levels of concurrency. As previously discussed, many applications exhibit concurrency levels ranging from 1 to 3, or even up to 4. Even under such expected levels of concurrency, the choice remains clear: Even in the worst-case scenario, utilizing CGRAs results in close to a 2× reduction in carbon footprint, turning out to be significantly better from the point of view of sustainability. The benefit is slightly lower when every kernel demonstrates 100% utilization of the CGRA, an unlikely scenario.

Building on the CDC replacement discussion, frequently used small DSAs (e.g., *AESEncrypt*) present a unique opportunity. Given its small footprint and prevalence across applications, maintaining *AESEncrypt* as a dedicated DSA while substituting other DSAs with CGRAs could further reduce the carbon footprint, especially at higher concurrency, such as $n = 4$. At this concurrency level, keeping *AESEncrpyt* as a separate DSA alongside CGRAs, compared to a chip with 40 DSAs, results in a 4.05× better carbon footprint in contrast to the 3.12× improvement for $n = 4$ observed in Table 3 previously. This is because this approach allows for a smaller CGRA for the remaining 3 concurrent kernels, alongside the compact DSA, leading to a more efficient overall design (Case II in Section 4.3).

4.6 Second-Order Benefits

While we have extensively discussed the first-order sustainability benefits of using reconfigurable fabrics for hardware specialization, there exist some second-order benefits that are difficult to quantify. When multiple DSAs execute within an application, significant data movement occurs across DSAs and their memory hierarchies. CGRAs can multiplex kernels in space and time, reducing such data movement. Moreover, reconfigurable architectures can be manufactured on a larger scale, agnostic of the application/device/company it will serve, streamlining manufacturing by producing larger quantities of identical chips instead of smaller quantities of different ones. Additionally, designing highly specific accelerators for an application severely limits the chip's adaptability to changes. However, with reconfigurable fabrics, new algorithms, and kernels can be seamlessly integrated, optimizations can be made to existing kernels, and entirely new applications can be supported through simple 'software updates'. This flexibility prolongs the device's lifespan and usability, thereby enabling sustainable hardware specialization.

5 RELATED WORK

Reconfigurable accelerators have emerged as an alternative to reduce the non-recurring engineering costs associated with ASICs. Ours is the first work to motivate reconfigurable logic as a substitute for a sea of DSAs from a sustainability perspective in the dark silicon era. We focus on CGRA instead of FPGA because of the former's superior area and energy efficiency.

While FPGAs are the most widely used reconfigurable accelerators, their efficiency is hindered by the higher power and area requirements resulting from their bit-level reconfigurability. According to Kuon and Rose [34], the area of an FPGA is 40× higher than that of an ASIC in a logic-only FPGA, and it is 21× higher in a modern FPGA with logic, DSPs, and BRAMs. Wong et al. [49] show that an FPGA sees a 17× - 27× area increase compared to a custom CMOS implementation. Kuon and Rose [34] also compared the dynamic power dissipation, which is 9× higher in modern FPGAs. This significantly reduces the viability of FPGAs as a sustainable computing alternative, particularly for battery-operated devices where embodied footprint dominates.

CGRA, with a coarser word-level reconfigurability, offers an efficient alternative to FPGAs, bringing its area and power consumption closer to that of ASICs. Numerous CGRAs are proposed in industry [10, 18, 20, 32] and academia [22, 31, 36, 39]. SoftBrain [39] CGRA shows that its area and energy are within 8× and 2× of the ASIC values, respectively. Performance [38–40, 47] and energy efficiency [5, 6, 22, 23, 45] are primary optimization criteria in modern CGRA design. From a sustainability standpoint, area efficiency of CGRAs is equally crucial, particularly in edge devices where the embodied footprint plays a significant role.

6 CONCLUSION

Dark silicon fundamentally trades off chip area for power efficiency, which is not environmentally sustainable due to its increasing embodied footprint as technology evolves. This work explored a sustainable alternative through reconfigurable logic. Although reconfigurable logic incurs a larger operational footprint, it significantly reduces the embodied footprint by amortizing chip area across many kernels. By combining abstract analytical modeling and hardware synthesis, we find that a representative reconfigurable fabric, namely CGRA, can drastically reduce the environmental footprint of hardware specialization compared to a sea of DSAs.

ACKNOWLEDGMENTS

This research is partially supported by Research Foundation Flanders (FWO) grant no. G018722N and by the National Research Foundation, Singapore, under its Competitive Research Programme Award NRF-CRP23-2019-0003.

REFERENCES

[1] B. Alcott. 2005. Jevons' Paradox. *Ecological Economics* 54, 1 (2005).

[2] AMD. 2023. 4th Gen AMD EPYC Processor Architecture. https://www.amd.com/system/files/documents/4th-gen-epyc-processor-architecture-white-paper.pdf

[3] Apple. 2022. Apple unveils M2, taking the breakthrough performance and capabilities of M1 even further. https://www.apple.com/newsroom/2022/06/apple-unveils-m2-with-breakthrough-performance-and-capabilities/

[4] C. Banbury, V. J. Reddi, P. Torelli, J. Holleman, N. Jeffries, C. Kiraly, P. Montino, D. Kanter, S. Ahmed, D. Pau, U. Thakker, A. Torrini, P. Warden, J. Cordaro, G. Di Guglielmo, J. Duarte, S. Gibellini, V. Parekh, H. Tran, N. Tran, N. Wenxu, and X. Xuesong. 2021. MLPerf Tiny Benchmark. arXiv:2106.07597 [cs.LG]

[5] T. K. Bandara, D. Wijerathne, T. Mitra, and L-S. Peh. 2022. REVAMP: a systematic framework for heterogeneous CGRA realization. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 918–932.

[6] T. K. Bandara, D. Wu, R. Juneja, D. Wijerathne, T. Mitra, and L-S. Peh. 2023. FLEX: Introducing FLEXible Execution on CGRA with Spatio-Temporal Vector Dataflow. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*.

[7] N. Bleier, M. H. Mubarik, S. Chakraborty, S. Kishore, and R. Kumar. 2022. Rethinking Programmable Earable Processors. In *IEEE/ACM International Symposium on Computer Architecture (ISCA)*. 454–467.

[8] N. Bleier, A. Wezelis, L. R. Varshney, and R. Kumar. 2023. Programmable Olfactory Computing. In *IEEE/ACM International Symposium on Computer Architecture (ISCA)*. 1–14.

[9] M T Bohr and I A Young. 2017. CMOS Scaling Trends and Beyond. *IEEE Micro* 37, 6 (2017), 20–29.

[10] E. Brunvand, D. Kline, and A. K. Jones. 2019. Dark Silicon Considered Harmful: A Case for Truly Green Computing. In *Proceedings of the International Green and Sustainable Computing Conference (IGSC)*. 1–8.

[11] A. Carsello, K. Feng, T. Kong, K. Koul, Q. Liu, J. Melchert, G. Nyengele, M. Strange, K. Zhang, A. Nayak, J. Setter, J. Thomas, K. Sreedhar, P-H Chen, N. Bhagdikar, Z. Myers, B. D'Agostino, P. Joshi, S. Richardson, R. Bahr, C. Torng, M. Horowitz, and P. Raina. 2022. Amber: A 367 GOPS, 538 GOPS/W 16nm SoC with a Coarse-Grained Reconfigurable Array for Flexible Acceleration of Dense Linear Algebra. In *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 70–71.

[12] Y-H. Chen, T-J. Yang, J. Emer, and V. Sze. 2019. Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 2 (2019), 292–308.

[13] A. Chowdhery, P. Warden, J. Shlens, A. Howard, and R. Rhodes. 2019. Visual Wake Words Dataset. arXiv:1906.05721 [cs.CV]

[14] R. H. Dennard, F. H. Gaensslen, H-N. Yu, L. Rideout, E. Bassous, and A. R. LeBlanc. 1974. Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions. *IEEE Journal of Solid State Circuits* 9, 5 (1974), 256–268.

[15] L. Eeckhout. 2023. Kaya for Computer Architects: Toward Sustainable Computer Systems. *IEEE Micro* 43, 1 (2023), 9–18.

[16] L. Eeckhout. 2024. FOCAL: A First-Order Model to Assess Processor Sustainability. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Vol. 2. 401–415.

[17] H. Esmaeilzadeh, E. M. Blem, R. S. Amant, K. Sankaralingam, and D. Burger. 2011. Dark Silicon and the End of Multicore Scaling. In *Proceedings of the IEEE/ACM International Symposium on Computer Architecture (ISCA)*. 365–376.

[18] Kermin E Fleming and et al. 2020. Processors, methods, and systems with a configurable spatial accelerator. US Patent 10,558,575.

[19] C. Freitag, M. Berbers-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday. 2021. The Real Climate and Transformative Impact of ICT: A Critique of Estimates, Trends, and Regulations. *Patterns* 2, 9 (2021).

[20] Taro Fujii and et al. [n. d.]. New Generation Dynamically Reconfigurable Processor Technology for Accelerating Embedded AI Applications. In *VLSI'18*. 41–42.

[21] M Garcia Bardon, P Wuytens, L-A Ragnarsson, G Mirabelli, D Jang, G Willems, A Mallik, A Spessot, J Ryckaert, and B Parvais. 2020. DTCO including sustainability: Power-performance-area-cost-environmental score (PPACE) analysis for logic technologies. In *IEEE International Electron Devices Meeting (IEDM)*. 41.4.1–41.4.4.

[22] G. Gobieski, A. O. Atli, K. Mai, B. Lucia, and N. Beckmann. 2021. Snafu: An Ultra-Low-Power, Energy-Minimal CGRA-Generation Framework and Architecture. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. 1027–1040.

[23] G. Gobieski, S. Ghosh, M. Heule, T. Mowry, T. Nowatzki, N. Beckmann, and B. Lucia. 2022. RipTide: A Programmable, Energy-Minimal Dataflow Compiler and Architecture. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 546–564.

[24] U. Gupta, M. Elgamal, G.-Y. Wei G. Hills, H.-H. S. Lee, D. Brooks, and C.-J. Wu. 2022. ACT: Designing Sustainable Computer Systems With An Architectural Carbon Modeling Tool. In *Proceedings of the ACM/IEEE Inernational Symposium on Computer Architecture (ISCA)*. 784–799.

[25] U Gupta, Y G Kim, S Lee, J Tse, H-H S Lee, G-Y Wei, D Brooks, and C-J Wu. 2021. Chasing Carbon: The Elusive Environmental Footprint of Computing. In *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 854–867.

[26] J. L. Hennessy and D. A. Patterson. 2019. A New Golden Age for Computer Architecture. *Commun. ACM* 62, 2 (2019), 48–60.

[27] M. D. Hill and V. J. Reddi. 2019. Roofline Model for MobilGables: Aes SoCs. In *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 317–330.

[28] Karageorgos I., Sriram K., Vesely J., Wu M., Powel M., Borton D., Manohar R., and Bhattacharjee A. 2020. Hardware-Software Co-Design for Brain-Computer Interfaces. In *IEEE/ACM International Symposium on Computer Architecture (ISCA)*.

[29] IBM. 2021. IBM Telum Processor: The next-gen microprocessor for IBM Z and IBM LinuxONE. https://www.ibm.com/blog/ibm-telum-processor-the-next-gen-microprocessor-for-ibm-z-and-ibm-linuxone/

[30] Intel. 2022. Technical Overview Of The 4th Gen Intel Xeon Scalable processor family. https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html

[31] M. Karunaratne, A. K. Mohite, T. Mitra, and L-S Peh. 2017. HyCUBE: A CGRA with reconfigurable single-cycle multi-hop interconnect. In *DAC'17*. 1–6.

[32] C. Kim and et al. [n. d.]. ULP-SRP: Ultra low power Samsung Reconfigurable Processor for biomedical applications. In *FPT'12*. 329–334. https://doi.org/10.1109/FPT.2012.6412157

[33] D. Kline, N. Parshook, X. Ge, E. Brunvand, R. G. Melhem, P. K. Chrysanthis, and A. K. Jones. 2019. GreenChip: A tool for evaluating holistic sustainability of modern computing systems. *Sustainable Computing: Informatics and Systems* 22 (June 2019), 322–332.

[34] I. Kuon and J. Rose. 2007. Measuring the Gap Between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26, 2 (2007), 203–215.

[35] Boakes L, G. Bardon M, Schellekens V, Rolin C, Liu I-Y Vanhoucke B, Mirabelli G, Sebaai F, V. Winckel L, Gallagher E, and Ragnarsson L-A. 2023. Cradle-to-gate Life Cycle Assessment of CMOS Logic Technologies. In *IEEE International Electron Devices Meeting (IEDM)*. 1–4.

[36] B. Mei, S. Vernalde, D. Verkest, H. Man, and R. Lauwereins. [n. d.]. ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix. In *FPL'03*. https://doi.org/10.1007/978-3-540-45234-8_7

[37] F. Muñoz-Martínez, J. L. Abellán, M. E. Acacio, and T. Krishna. 2021. STONNE: Enabling Cycle-Level Microarchitectural Simulation for DNN Inference Accelerators. In *2021 IEEE International Symposium on Workload Characterization (IISWC)*.

[38] Q. M. Nguyen and D. Sanchez. 2021. Fifer: Practical Acceleration of Irregular Applications on Reconfigurable Architectures. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*. 1064–1077.

[39] T. Nowatzki, V. Gangadhar, N. Ardalani, and K. Sankaralingam. 2017. Stream-dataflow acceleration. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*. 416–429.

[40] R. Prabhakar, Y. Zhang, D. Koeplinger, M. Feldman, T. Zhao, S. Hadjis, A. Pedram, C. Kozyrakis, and K. Olukotun. 2017. Plasticine: A Reconfigurable Architecture For Parallel Paterns. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*. 389–402.

[41] Qualcomm. 2022. Snapdragon 6 Gen 1 Mobile Platform. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/09162022_Prod_Brief_QCOM_SD_6_Gen_1.pdf

[42] B. Reagen, R. Adolf, Y. S. Shao, G-Y Wei, and D. Brooks. 2014. MachSuite: Benchmarks for Accelerator Design and Customized Architectures. In *IISWC'14*.

[43] Shao Y. S., Reagen B., G.-Y. Wei, and D. M. Brooks. 2014. Aladdin: A pre-RTL, power-performance accelerator simulator enabling large design space exploration of customized architecture. In *Proceedings of the IEEE/ACM International Symposium on Computer Architecture (ISCA)*. 97–108.

[44] Y. S. Shao, B. Reagen, G.-Y. Wei, and D. Brooks. 2023. RETROSPECTIVE: Aladdin: a Pre-RTL, Power-Performance Accelerator Simulator Enabling Large Design Space Exploration of Customized Architectures. In *ISCA@50 25-Year Retrospective: 1996-2020*, José F. Martínez and Lizy K. John (Eds.). ACM SIGARCH and IEEE TCCA. https://bit.ly/isca50_retrospective

[45] C. Torng, P. Pan, Y. Ou, C. Tan, and C. Batten. 2021. Ultra-Elastic CGRAs for Irregular Loop Specialization. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 412–425.

[46] G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin, J. Lugo-Martinez, S. Swanson, and M. B. Taylor. 2010. Conservation Cores: Reducing the Energy of Mature Computations. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. 205–218.

[47] D. Wijerathne, Z. Li, M. Karunaratne, A. Pathania, and T. Mitra. 2019. CASCADE: High Throughput Data Streaming via Decoupled Access-Execute CGRA. *ACM Trans. Embed. Comput. Syst.* 18, 5s (2019).

[48] D. Wijerathne, Z. Li, M. Karunaratne, L-S Peh, and T. Mitra. [n. d.]. Morpher: An Open-Source Integrated Compilation and Simulation Framework for CGRA. *WOSET'22* ([n. d.]).

[49] H. Wong, V. Betz, and J. Rose. 2011. Comparing FPGA vs. custom cmos and the impact on processor microarchitecture. In *Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays*. 5–14.