Area Bloating and the Future of Specialization

Qixuan Yu
Electrical and Computer Engineering
Princeton University
Princeton, NJ
qixy@princeton.edu

David Wentzlaff
Electrical and Computer Engineering
Princeton University
Princeton, NJ
wentzlaf@princeton.edu

***Abstract*—Throughout history, technology trends have greatly influenced the development of computer architecture. For the past decade, continuing transistor density scaling combined with stalling voltage scaling created great opportunities for specialization. As a step forward from multi-core processors, hardware accelerators make use of vastly available transistors by dedicating them to specific applications, in exchange for performance and energy efficiency. Unfortunately, the scaling of transistor size is now also slowing down. The complete failure of Moore's Law will raise challenges and demand new approaches with specialization.**

In this work, we model technology scaling, accelerator scaling, and chip scaling to determine future demands of hardware acceleration in a collection of benchmarks for the next eight technology nodes. We show that if current approaches continue, area is becoming a major limiting factor of achievable performance. In a phenomenon we call area bloating, the area and hence manufacturing cost of chips can increase by at least $78.4\times$ and $117.4\times$ respectively from 3nm to A1.8 to match performance gain expectations like those in the past. 3D stacking can deliver more area under the same footprint, at the cost of multiplied power density. But for a limited number of layers, different from the dark silicon prediction, power density plateaus and silicon can be kept bright under liquid cooling.

Architectural innovation is required to combat area bloating. We present several directions for future studies. Reduced levels of integration and reduced levels of specialization can help future accelerators adapt to these new trends. By adding limited generality to accelerators, reconfigurable specialization can combat area bloating in platform-specific SoCs, saving significant amounts of area. Future architectures must also consider longevity as chip lifetime increases as a result of slow adoption.

***Index Terms*—Technology Scaling, Hardware Specialization.**

I. Introduction

The past decade of computer architecture was a mixture of opportunities and uncertainties. Specialized hardware accelerators gained popularity, creating unprecedented demands for architectural customization. As general-purpose multi-cores faced diminishing returns, accelerators delivered orders of magnitude performance and efficiency improvements. Advanced applications like machine learning and autonomous driving would not be feasible without specialization. The advent of chiplet technology and 3D integration blended the boundary between chips, packages, and systems. Many more transistors can be tightly integrated, breaking performance bounds constrained by the single reticle limit.

On the flip side, the remaining potential of technology scaling started to become questionable. The life or death of Moore's Law [39] is under heated debate. But Dennard scaling [10] has failed. The dark silicon outlook [11] set a looming bound on chip power. As they approach physical limits, new technology nodes are now improving more slowly in terms of density and performance [24]. Alternative forms of scaling transistors and computing methods have yet to become promising. Halting technology scaling may soon limit computer performance and efficiency on short notice.

In the future, which side is going to prevail? Is technology scaling reaching its eventual limits? What does it take to keep performance doubling every two years, and are those demands achievable? In what ways will we change computer architectures in response to new scaling trends?

In this study, we answer the above questions by projecting into the future, assuming that 1) current trends of specialization continue; 2) current trends of technology scaling continue; and 3) current trends of computing demands continue. Prior studies have speculated architectural and technological trends in their respective times for different architectures [11], [15], [30], [37], [42], [50], [61]. Now deep into the hardware specialization era and approaching the end of Moore's Law, this study reexamines prior conclusions and sets expectations for the next decade.

Importantly, instead of passively deducing an absolute performance limit based on pre-determined area and power budgets, we project future demands on area, power, and cost based on performance scaling goals. We then examine the challenges of meeting such demands and proactively explore future architectural solutions. While we believe that specialization will continue to be effective, this study identifies under-explored design directions and strategies that will become important as technology scaling continues to slow down.

We combine the following three models to project future demands of specialized chips:

- Technology model: density, power, and latency achievable by the next eight technology nodes from 2nm to A1.8, based on IRDS projections [22]–[24]; cost per transistor projections based on recent estimations [25].
- Accelerator model: accelerator design space of 25 benchmarks from SHOC [9], MachSuite [47], and select functions in SPEC2017 [6], [57], explored using the Aladdin simulator [53].
- Chip model: different performance demands and optimization targets for today's different types of chips utilizing accelerators, like server chips and mobile SoCs.

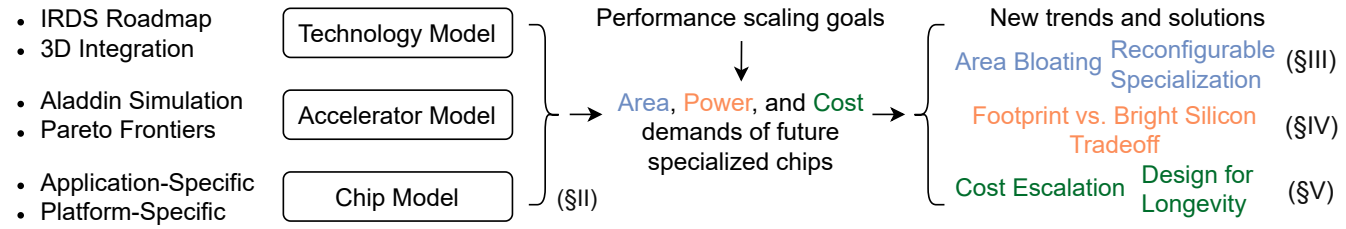


Fig. 1: An overview of the projection method and main conclusions. The color cues will be used throughout the paper.

Our models are likely highly optimistic. Technological gains may be short of IRDS projections. Additional system and memory overheads may greatly limit achievable accelerator performance. Actual future area, power, and cost demands are likely higher than what we present. Nonetheless, the results from this study confirm and quantify the following new trends of demands to keep performance improving at rates comparable with those in the Moore's Law era:

- **Area bloating** (§III) Transistor density no longer scales. To match Moore's Law era throughput improvement rates, the total area of chip(s) may bloat by at least $78.4\times$ in the A1.8 node vs. 2022's 3nm node. Area becomes an important constraint.
- **Footprint vs. Bright Silicon Tradeoff** (§IV) Power density plateaus for a single layer of transistors. Future chips can choose bright silicon, or reduced footprint, by tuning the number of 3D integrated layers.
- **Cost escalation** (§V) Wafer cost increases. Over 80% of the performance gain can be attributed to cost factors rather than technology scaling by the A1.8 node.

We identify the following directions that deserve more investigation in response to the new demands:

- **Reconfigurable specialization** (§III) As area becomes an important constraint, chips with heterogeneous accelerators in constrained devices can enjoy area reduction using a specialized form of reconfigurability.
- **Design for longevity** (§V) Moving to new nodes will not greatly reduce cost. Chips will need to last longer, both architecturally and physically, when new node adoption may take multiple years after initial production.

II. Modeling Method

Figure 1 shows the workflow of this study and a summary of its main conclusions. Three models are proposed and combined in deriving the demands of future specialized chips:

The technology model provides transistor density, power, latency, and cost scaling factors from 3nm to A1.8 nodes. It is based on data in the International Roadmap for Devices and Systems (IRDS) 2020 [22], 2022 [23], and 2024 [24] editions.[1]. We project costs per transistor from 3nm to A1.8 nodes based on an industry analysis [25].

[1]We use data primarily from the IRDS More Moore report including its tables file. At the time of writing, the IRDS 2024 Edition is the latest edition.

The accelerator model estimates area-performance and power-performance characteristics of accelerators specialized to each benchmark. It uses the Aladdin pre-RTL accelerator simulator [53] to sweep the design space of hardware accelerators. For each program, the model fits curves to Pareto points to derive the minimum area and power to achieve different levels of performance targets.

The chip model bridges the gap between individual accelerators and chips utilizing accelerators. We identify two distinct types of chips that heavily rely on specialization: An "application-specific chip" dedicates the entire chip to a single application, often residing in data centers and targeting throughput. A "platform-specific chip" needs to support all applications running on a specific platform, from mobile phones, personal computers, to cars and robots. It needs heterogeneous accelerators and sometimes has latency requirements. The chip model maps chip performance goals to the targets of composing accelerators.

A. Technology Model

Our technology model projects characteristics of future process nodes based on data in recent IRDS reports [22]–[24] and other sources [14], [15], [25]. They set future technology scaling expectations towards A1.8 reached by the year 2039.

1) Area vs. Footprint: In modeling technology scaling, it is important to distinguish between area and footprint. Stacking PCBs, packages, dies, or transistor layers on top of each other has always been an effective method to reduce the footprint of silicon. However, they are usually not considered as able to improve the "area density" of transistor technology, for the following reasons:

- Applying 3D integration does not change the dimensions of transistors, whereas area serves as a traditional proxy for transistor size normalized to a single 2D layer;
- 3D integration as a system design choice is applicable to many technology nodes, whereas area density as a traditional metric measures the limit of a particular node.

Throughout this paper, "area" and "density" refer to their traditional meaning: for a single layer of transistors in a technology node. In the technology model, we base logic and SRAM density scaling factors on "NAND2-eq gate density - HD" and "HD SRAM area density" lines in the IRDS. These lines show density per unit of area instead of per unit of footprint. The IRDS assumes 2, 4, and 6 stacked

tiers of transistors in A3.5, A2.5, and A1.8, respectively. Some other lines report density in the sense of footprint, which approximately equals the density scaling factors in our model times the number of stacked tiers used. In Section IV, we discuss the tradeoff between footprint and power density when applying tiers and 3D integration.

2) Transistor vs. Circuit Level Scaling: For power and performance, we identify two sets of projection methodologies in IRDS: one based on transistor device and cell characteristics. The other incorporates models for wires and real-world logic cores like CPUs and MAC units. In terms of absolute values, transistor-level characteristics are more optimistic than circuit-level ones, as they do not include realistic loads or designs. But for the scaling trends, circuit-level characteristics sometimes improve faster than transistor-level ones. Techniques like backside power delivery and 3D integration help reduce wire length, which reduces power even when transistors themselves do not improve significantly [24]. In the technology model, we use IRDS' circuit-level scaling data for speed and power projections. We base speed on wireloaded frequency scaling, which follows almost identical normalized trends as CPU and MAC frequency scaling in IRDS [24]. We note that speed and power for A3.5 through A1.8 consider wirelength reduction from 3D integrated tiers. Notably, speed keeps decreasing from A3.5 to A1.8, making a layer in A3.5 have more performance per unit area than one in A1.8. The main improvement from A3.5 to A1.8 is in power efficiency, but added layers increase power density. In Section IV we further discuss this tradeoff.

3) Putting It All Together: Table I summarizes scaling factors for 40nm down to A1.8, normalized to the 3nm node. Here, the dynamic power column considers frequency scaling shown in the speed column. As the Aladdin accelerator simulator [53] is native to 40nm, we extend the technology model back to the 40nm node using scaling factors in prior works [14], [15]. We use the IRDS 2020 edition to project from 5nm to 3nm, use the 2022 edition to project from 3nm to 2nm, then use the 2024 edition to project further from 2nm to A1.8.

Figure 2 shows possible future manufacturing cost per transistor from 3nm to A1.8 based on the extrapolation of a recent projection [25]. Unlike historical cost data, this projection is not affected by inflation. We assume that 3D tier stacking does not significantly impact transistor costs. In Section V, we discuss potential cost reductions as technology nodes mature after mass production.

4) Bias Characterization: The IRDS projection is sometimes seen as an optimistic view towards future technology scaling. For certain metrics like wireloaded frequency and quiescent supply current (IDDQ), IRDS does not provide values for both high-density (HD) and high-performance (HP) cells. Our technology model uses a mixture of available HP and HD values in an effort to provide optimistic projections in each metric. Speed is based on HP cells, while density and power are based on HD cells. As a result, performance per unit of area and performance per unit of power estimations are optimistic. For power per unit area: IRDS 2024 shows NAND2-eq gate density - HD scales 43.9% better than NAND2-eq gate density - HP from 2nm to A1.8 [24]. HD dynamic power at 1GHz scales 18% better than HP dynamic power at 1GHz from 2nm to A1.8 [24]. In designs with a mixture of HP and HD cells, power per unit of area is likely overestimated.

TABLE I: Projected characteristics of future technology nodes normalized to 3nm. Data sources: [14], [15], [22]–[24].

Year	Node	Speed	Logic Density	SRAM Density	Dynamic Power	Static Power
2008	40nm	0.195	0.019	0.021	2.107	8.846
2020	5nm	0.958	0.739	0.810	1.046	1.274
2022	3nm	1.000	1.000	1.000	1.000	1.000
2025	2nm	1.031	1.381	1.000	0.832	1.179
2027	1.4nm	1.065	1.739	1.075	0.664	1.107
2029	A10	0.894	2.199	1.125	0.508	0.821
2031	A7†	1.268	2.915	1.925	0.674	1.036
2033	A5	1.195	3.273	1.925	0.591	0.821
2035	A3.5	1.268	3.887(×2)*	2.025(×2)*	0.512	0.714
2037	A2.5	1.212	3.887(×4)*	2.025(×4)*	0.400	0.607
2039	A1.8	1.158	3.887(×6)*	2.025(×6)*	0.361	0.607

* Numbers in the parentheses show the default number of 3D integrated tiers IRDS associates with the technology node. Numbers to the left of the parentheses are for a single layer/tier of transistors.
† The first node with complementary FET (CFET).

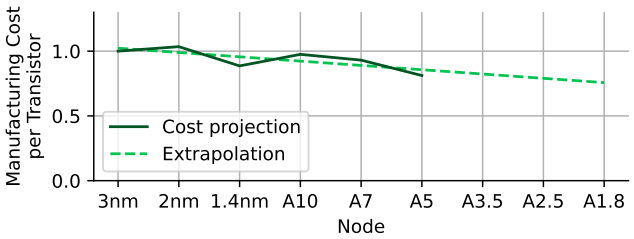


Fig. 2: Manufacturing cost per transistor normalized to 3nm, linearly extrapolated with least squares error. Data source for cost projection: [25].

B. Accelerator Model

Specialized accelerators are known to be difficult to design, test, and model. Unlike general-purpose cores, accelerators for different applications naturally differ in area and power consumption. An accelerator model must explore the design space of each application to determine their respective area-power-performance relationships. However, unlike CAD tools that must provide accurate results for practical implementations, our accelerator model makes generous assumptions and only needs to capture general trends.

1) Design Space Exploration: In this study, we use the Aladdin pre-RTL simulator [53] to explore the design space of accelerated applications. Aladdin takes application programs written in C along with design configurations to estimate the performance, power, and area of specialized accelerators. To provide lower bounds on the hardware demands of future chips, we are generous in assuming the following ideal conditions when configuring Aladdin:

- Precise clock gating of all idle components.
- Private scratchpad memory with no bank conflicts. A true LRU scheme is used to manage scratchpad content.

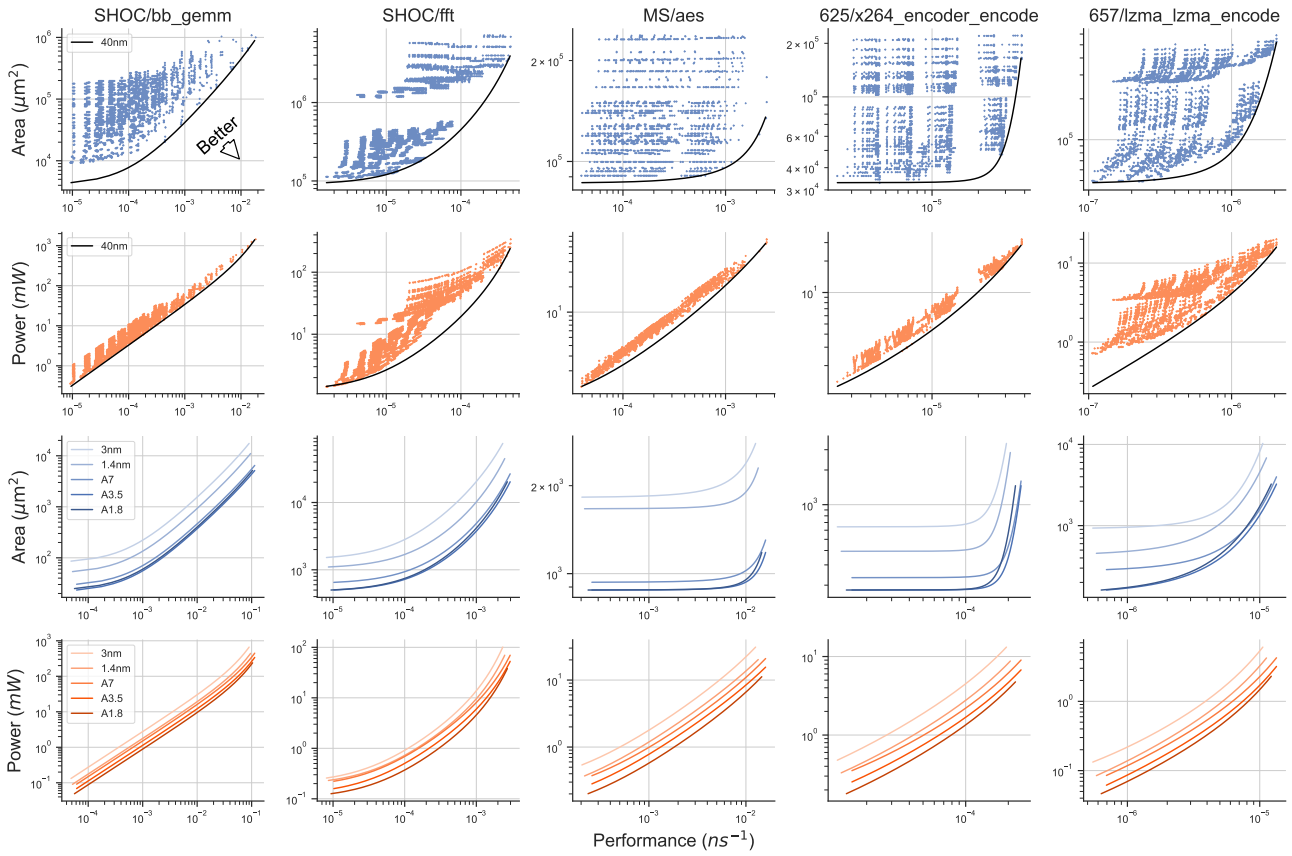


Fig. 3: A demonstration of the accelerator model and the technology model applied to 5 representative applications from SHOC, MachSuite, and select functions in SPEC2017 benchmarks. Horizontally: the five applications. Top row: area-performance design space and fitted Pareto frontier at 40nm. Second row: power-performance design space and frontiers at 40nm. Third row: area-performance frontiers at future nodes. Bottom row: power-performance frontiers at future nodes.

- No contention in higher levels of the memory hierarchy.

We sweep combinations of all major parameters in Aladdin's design configuration to generate enough coverage of accelerator design spaces:

- Clock period: 1ns, 3ns, 6ns, 10ns in the context of Aladdin's native 40nm technology node.
- Loop unrolling factor: 1, 2, 4, or fully unroll, for all loops in the application program. The top 2 most frequent loops sweep the factors independently.
- Loop pipelining: enable or disable for all loops in the application program. The top 4 most frequent loops sweep the pipelining settings independently.
- Scratchpad size: from 64B to 16KB at 2x steps.
- Scratchpad bandwidth: from 8B to 512B per cycle at 2x steps, when commensurate with scratchpad sizes.

Our modeling is highly optimistic. We do not integrate Aladdin with CPU simulators or full memory hierarchy simulators. CPU-accelerator interactions, memory bandwidth, cache management, and many other factors greatly impact accelerator performance. They may become significant bottlenecks in future nodes, or alleviate if memory subsystem improves and programming models evolve. Either way, our method still provides a lower bound on accelerator demands. Our exploration is not meant to guide the design of any specific accelerator under any given technology node. Our technology model can be applied to dedicated system simulators for tighter bounds on specific accelerator designs.

2) Pareto Frontier Fitting: The swept design space forms the basis of the accelerator model. For each benchmark, we fit area vs. performance and power vs. performance frontier curves. These curves represent the trends for optimal design points from area-efficiency and power-efficiency perspectives. Comparing the curves for different benchmarks shows varying levels of resource (area/power) demands under different performance targets. Note that many applications suitable for acceleration are highly parallel, which means throughput, the amount of resources, and the input size of the program can have highly linear correlations. By increasing the problem input size, an accelerator can be more area/power efficient than the curves in the accelerator model, which are derived for programs with a fixed input size. The chip model reconciles this important gap by translating the demands of throughput-oriented chips to individual accelerators optimized under this accelerator model.

TABLE II: Accelerated functions selected from the SPEC2017 benchmark suite and their running time percentages.

Benchmark	Function	Time (%)
600.perlbench	S_regmatch	18.0
602.gcc	cgraph_analyze_function	35.4
605.mcf5	primal_bea_mpp_run	54.7
625.x264	x264_encoder_encode	100
657.xz	lzma_lzma_encode	96.8
619.lbm	LBM_performStreamCollideTRT*	99.6
644.nab	mme34	99.9

* Selected the body of the function's main loop.

For each benchmark, we fit exponential area vs. performance and power vs. performance curves to Pareto-optimal design points using least squares error. The following function is used as the exponential model:

$$\text{Resource} = A(e^{B \times \text{Performance}} - 1) + C$$

where A, B, and C are non-negative parameters. This relationship assumes a diminishing return with more resources specialized to a program with a fixed input size. It is a linear transformation of the exponential function. We further apply the constraint that the fitted curve is no less optimal than any explored design, so the curve maintains Pareto-optimality. We normalize resource and performance for numerical stability, and use a trust region optimization algorithm with multi-start.

3) Applications: We explore the design spaces for 25 benchmarks from SHOC [9], MachSuite [47], and select functions in SPEC2017 [6], [57]. SHOC and MachSuite contain kernels suitable for hardware acceleration. We use programs in the two benchmarks as-is, omitting their overlapped kernels. For SPEC, we use gprof to identify functions that account for large percentages of running time on a typical x86 machine. We only use the selected functions and not the whole benchmarks. We run the functions through Aladdin for the presented results. We choose benchmarks written in C, which is supported by Aladdin. Table II details the selected functions from the SPEC benchmarks. The running time percentages reflect the importance of the selected functions in each benchmark on our profiling machine. It considers the cumulative time of all executions of the function of interest, including the time of any sub-function it calls. The accelerators we model for the function of interest include all sub-functions called. The results in this research paper are not compliant SPEC results. In particular, we only study functions within SPEC benchmark applications and we do not use the SPEC run scripts or methodology.

4) Integration with the Technology Model: As technology scales differently for logic and memory cells and for dynamic and static power, instead of projecting the derived curves, we project all designs to future technology nodes and use the same derivation method to re-fit Pareto points at each node.

Figure 3 demonstrates the accelerator modeling process: design space exploration, Pareto frontier fitting, and projection using the technology model. For readability, we only show the fitted Pareto frontiers at every other future technology node.

TABLE III: Comparison between two different use cases of accelerators.

Chip Type	Application-Specific	Platform-Specific
Deployment	Data centers	Constrained platforms
System	Multi-chip	SoC
Metric	Throughput	Latency / Throughput
Applications	Single	Many
Area budget	High	Lower
Power budget	High	Lower

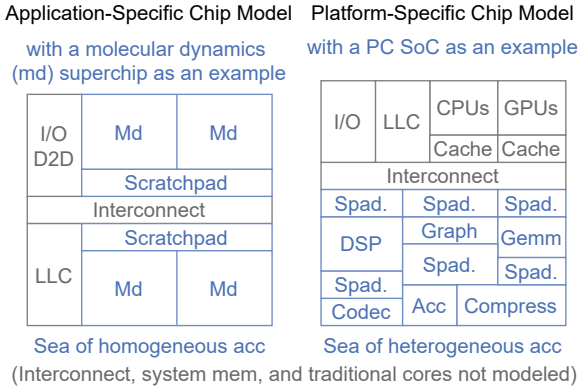


Fig. 4: Two types of chips using accelerators in the chip model, presented with examples.

C. Chip Model

As shown in Figure 4, we clarify two main types of chips nowadays that utilize accelerators heavily yet differently:

1) Application-Specific Chip: Shown in Figure 4 (left), it duplicates homogeneous accelerators to massive scales, sometimes beyond chip(let)s to boards, racks, and data centers. In some cases, the design of interconnects can be more important than the design of the building block accelerators themselves. This often arises when a large number of chips work together on a very large problem. Challenges remain to match the communication topology of the application and to deliver enough off-chip bandwidth. In other cases, a single chip is partitioned in space or time to serve multiple users, with each accelerator working on a small program with different inputs. Either way, the minimum demand from the accelerators themselves provides lower bounds on the area and power needed. From a systems engineer's perspective, this work concerns "compute bounds" for application-specific chips.

Delivering high throughput for a single application is the goal of an application-specific chip. Often deployed in data centers, it has high area and power budgets and enjoys economies of scale. In deriving the demands of an application-specific chip, we do not enforce a fixed input size on the application. We first use the accelerator model to derive the optimum area/performance and power/performance for an accelerator. We then multiply them by the number of homogeneous accelerators needed to reach the throughput target of the entire chip. This model provides an optimistic estimation of the minimum amount of area and power needed for an application-specific chip.

2) Platform-Specific Chip: A platform-specific chip is designed for specific platforms like mobile phones, personal computers, or robots. It must handle all possible applications available on the target platform and equip necessary heterogeneous accelerators for the purpose. On many platforms, latency is a target metric, which can be harder to meet than throughput. To derive the demands for a platform-specific chip targeting latency, we use the area-performance and power-performance frontiers derived by the accelerator model with its fixed input sizes. The area and power budgets are often lower than those for application-specific chips. A typical embedded SoC fits in a single chip. We examine the challenges posed by the stricter area, power, and cost constraints in our discussions. Table III summarizes the differences between the two chip types.

3) Unmodeled Components: We do not model the memory system, the interconnects, or traditional cores like CPUs and GPGPUs. Prior studies have explored the limitations of general-purpose cores [11], [21], [61]. Modeling memory can create tighter bounds on resource demands, but the lower bounds we derive in this study still hold. The opportunities and solutions we propose are also applicable when the actual bound can be higher.

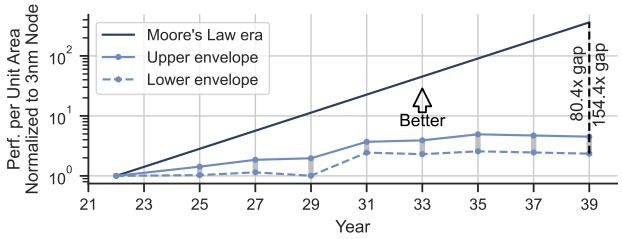
III. Area Bloating

A. The Challenge

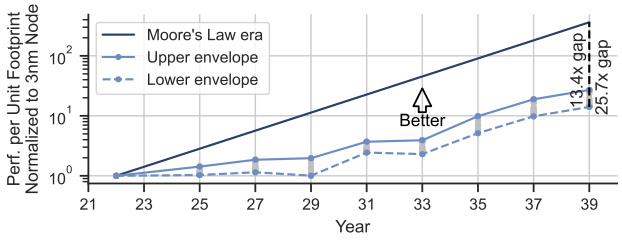
Figure 5 highlights the deficiency of transistor density scaling in future technology nodes. Under the colloquial understanding of Moore's Law, as technology improves, a design can fit into exponentially smaller areas, while keeping the same, if not better, performance. Even without architectural changes, the performance per area of a design can double every two years, as represented by the "Moore's Law era" line. Figure 5 shows actual gains of all designs explored in the next eight technology nodes, enclosed by upper and lower bounds shown as envelopes. Future technology nodes will fall short of delivering such gains and create a $80.4\times$ to $154.4\times$ area gap by the A1.8 node as shown in Figure 5a. The gap is not even closed from a performance per footprint perspective as shown in Figure 5b, when as many as 6 stacked tiers of transistors are applied to A1.8 in 2039:

Area Bloating
❶ As transistor density scaling slows down, more area is required to keep pace with performance needs.

For application-specific chips targeting throughput, accelerators can be replicated to meet performance goals. Further, designs can change to adapt to each technology node's characteristics. We replicate area-optimized accelerator designs from our fitted Pareto frontiers at each node for each benchmark and assess the total amount of area needed. Figure 6a shows the expected severity of area bloating with this approach. In the era when Moore's Law was alive and well, no bloating is expected, hence the solid line at the bottom showing a constant one. The dashed line on the top assumes no technology scaling,

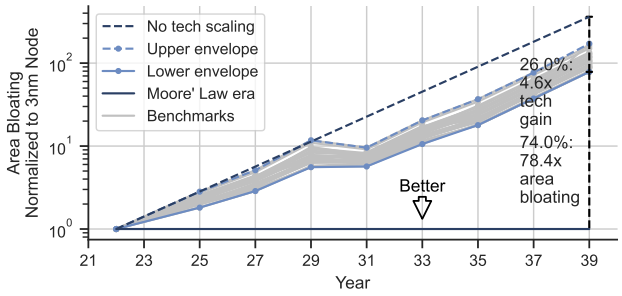


(a) Performance per unit area.

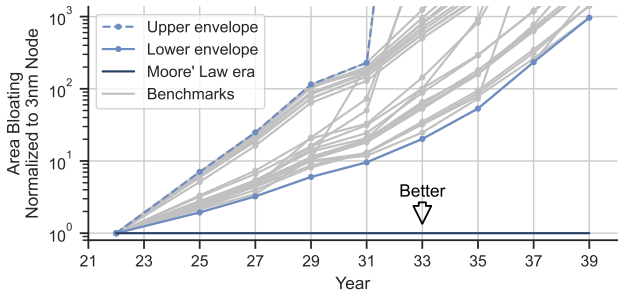


(b) Performance per unit footprint with IRDS anticipated stacked tiers of transistors.

Fig. 5: Performance per unit area/footprint improvements envelope among all accelerator designs for all applications, compared with the empirical trend in the Moore's Law era.



(a) Application-specific chip targeting throughput.



(b) Platform-specific chip targeting latency.

Fig. 6: Projected area bloating for area efficiency optimized designs.

where all performance gains come from area increases. For throughput targeting accelerators:

❷ By the A1.8 node in 2039, area bloating will contribute more to throughput improvements than technology gains, with required area at least 78.4 times that of 3nm application-specific chips.

TABLE IV: Reconfigurable specialization in the spectrum of general to specialized hardware.

	General Purpose	Reconfigurable General-Purpose	Reconfigurable Specialization	Fully Specialized
Examples	CPU, GPGPU	FPGA, CGRA, dataflow	Multi-purpose accelerators	Accelerators
Control	General instructions	General bitstreams, routing	Selection between supported apps	Fixed control
Datapath	General FUs General operand network	General or selected FUs Programmable network	Specialized FUs Fixed network with configurations	Specialized FUs Fixed network
Memory	Cache, general formats	Customizable data formats	Specialized memory and format	Specialized

Figure 6b shows area bloating for platform-specific chips targeting latency. We assume the design at the 3nm baseline node is at the most area-effective point. In the Moore's Law era, transistor speed increase can lead directly to latency improvements, hence the solid no-bloating line at the bottom showing a constant one. When technology improvements are not sufficient, designs must climb up the Pareto frontier for better latency, assuming such designs are feasible. Latency remains hard to improve and varies significantly by application. A platform-specific chip may bloat more severely than application-specific chips if it includes latency targets.

Figures 5 and 6 both show a major step of transistor area density improvement at 2031's A7 node, when complementary FET (CFET) stacks NMOS and PMOS for better density on top of lateral gate-all-around (LGAA) devices [24]. Regular improvements in other nodes do not bring significant area benefits. Figure 6 further shows that without underlying technology improvements, common accelerator architecture designs can not bring significant area benefits alone. Area bloating is likely to continue if there are no major technological shifts.

> ❸ The area bloating challenge demands more area-effective architectures. Area, or the number of transistors, becomes an important constraint.

Under this new landscape, future advanced chips will need to consume more area to keep on increasing performance. Instead of developing methods to utilize the vastly available transistors, architects must now treat them as a scarce resource.

B. Reconfigurable Specialization

We predict that an under-explored architectural direction can alleviate area bloating for platform-specific chips. The observation is that many platforms need to support multiple applications, but likely do not run them concurrently. For example, the video decoder in a mobile phone, as the user is watching a video, may seldom need to run concurrently with the NPU. Power and cooling also prevent stressing multiple accelerators together on many platforms. With under-utilization, the area efficiency of the entire chip is not optimized even if all of the accelerators contained within it are. A platform-specific chip can be more area-efficient if its accelerators can be reconfigured among time-multiplexed applications on the platform. Adding and controlling a limited level of generality can help alleviate area bloating.

We identify this approach as reconfigurable specialization. A fully specialized accelerator only supports a single application. A reconfigurable specialized one supports several applications, but not more. The provided reconfigurability is very low: no exposure to underlying functional units or switches like those in FPGAs or CGRAs, but only a few bits choosing between several applications known at design time. Table IV compares architectures from being more general to more specialized. Reconfigurable specialization introduces limited area overheads in comparison with general reconfiguration, in exchange for area reuse among the expected set of applications on a platform running at different times. Designs must carefully balance this tradeoff to combat area bloating.

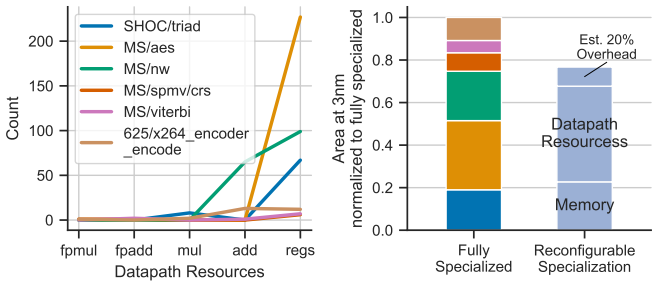


Fig. 7: A motivational study of reconfigurable specialization with a set of applications similar in problem size but differing in resource demands. Legend in the left figure.

Figure 7 provides an idealized case study of reconfigurable specialization applied to 6 diverse applications, assuming that they do not need to run simultaneously. The most area-effective fully-specialized accelerators for the applications have similar areas but differ in the demanded datapath resources. An estimated reconfigurable accelerator provides a superset of memory and datapath resources to match the performance of each individual fully-specialized accelerator. Suppose a 20% area overhead is introduced in the process: one can still achieve a 22.9% area reduction. An increased area may also help reduce the maximum power density if activated transistors can be spread out for each application. A reconfigurable accelerator is less area-effective for each of its supported applications, but becomes more area-effective for the platform-specific chip as a whole. This study is optimistic. We acknowledge challenges in designing reconfigurable specialized accelerators, and this direction deserves more investigation.

Note that reconfigurable specialization cannot help application-specific chips, which constantly run at high utilization. At the scale of data centers, deploying different chips for different applications eliminates any overhead

otherwise introduced by reconfigurability. Under this spirit, we predict that these chips will further divide current applications down to fine-grain tasks to specialize on specific stages or even inputs to gain area effectiveness.

IV. Footprint vs. Bright Silicon Tradeoff

A. The Opportunity

The deficiency of transistor density scaling is creating an unexpected opportunity. In the past, the continuation of density scaling and the stagnation of power scaling have created the dark silicon phenomenon [11], [56], [60]. Now, as both scaling trends slow down, silicon can become bright for a single layer of transistors[2]. As shown in Figure 8, power per area, or power density, does not observe a clear trend of increase, getting significantly better than the dark silicon outlook:

Bright Silicon

❶ Power density plateaus. Power delivery and cooling do not get significantly harder, allowing more transistors to be turned on.

❷ By the A1.8 technology node in 2039, the worst power density demanded for a single layer is capped at $3\times$ of 3nm's level. Liquid cooling can be sufficient.

Transistor latency, density, and power efficiency are all approaching physical limits. Future power density is unlikely to increase significantly. To assess worst-case cooling needs, we consider a potential large datacenter GPU at a TDP up to 500W with a reticle sized die. Its max power density can be up to $0.625W/mm^2$. Applying a $3\times$ multiplier from Figure 8 gives $1.875W/mm^2$. It is above the heat flux limit of air-cooled systems, but within a $3W/mm^2$ capability demonstrated by liquid cooling [1], [4], [26]. More expensive immersion cooling technology may not be necessary yet from a chip power density perspective. Power density is no longer an imminent hard bound on chip performance for a traditional single layer of transistors.

The total amount of demanded power, however, continues to increase, limiting constrained platforms. Figure 9a shows the demanded total power for application-specific chips targeting throughput. Dennard scaling has long been broken. To keep performance doubling every two years, at least 79.6% of the performance gain comes from the increase in total power. Figure 9b assumes the design at the 3nm baseline node is at the most power-effective point. It confirms that latency is even harder to improve from power demand pressure than area bloating pressure.

B. The Delivery of Area and Power

Area bloating and increased total power demand do *not* mean that future performance gains are infeasible. In addition

[2]There are two sources of dark silicon: the lack of parallelism in programs and the power limitation in technology scaling. Specialization helps uncover more parallelism, but limits still exist. Bright silicon suggests that the power limit will not be as severe as predicted.

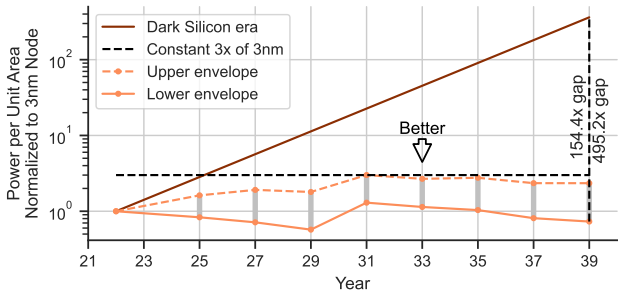
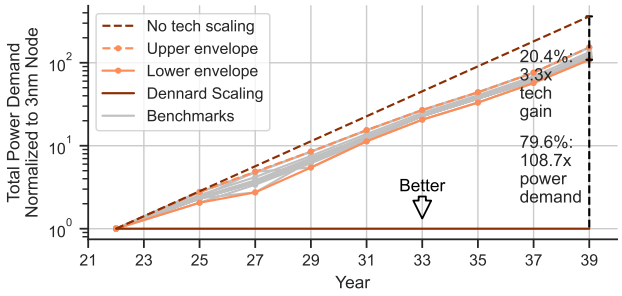
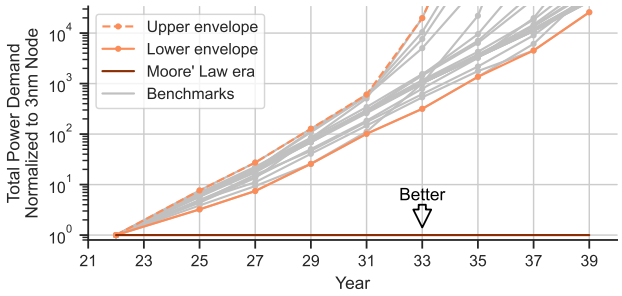


Fig. 8: Power per unit area envelope among all accelerator designs for all applications, compared with the dark silicon outlook. Power density may not exceed 3 times of 3nm's level.



(a) Application-specific chip targeting throughput.



(b) Platform-specific chip targeting latency.

Fig. 9: Projected total power demand for power efficiency optimized designs.

to miniaturization, the trend of integration has also reached a turning point. With area bloating, integrating all transistors needed onto a single monolithic die is largely infeasible. The level of integration can decrease in future nodes, with multiple chiplets, packages, and boards delivering the amount of total area demanded. This reduced level of integration makes it easier to deliver sufficient power and cooling to each individual package. Figure 10 shows the main methods to deliver the bloated area, where the reticle limit is no longer a bound on the total area. Here, each parallelogram represents a die. The total demanded area can be delivered as a monolithic die (top right), as 2D and 3D integrated layers (bottom left), or even on different boards in network-connected server racks (bottom right). Meeting the power requirements of wafer-scale integration is feasible with today's technology [32]. Server racks and data centers will need to increase their power limits and thermal capacity to fit chips of larger total area and power.

For application-specific chips, the pressure on physical dimensions is low. In very large-scale problems with man-

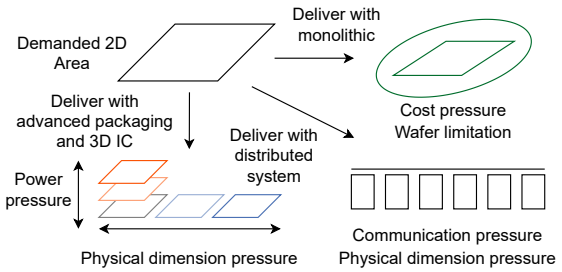


Fig. 10: Diverse methods exist to deliver area and power demands, creating different pressures in the process.

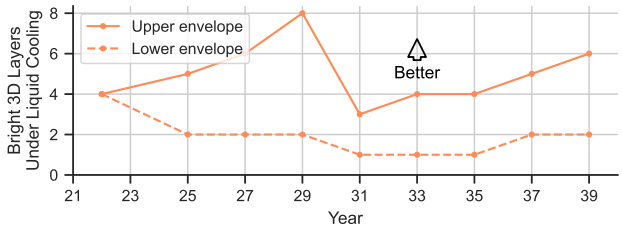


Fig. 11: Projected number of bright 3D-integrated layers allowed by liquid cooling in future technology nodes.

ageable communication pressure, the bloated area does not need to stay on the same chip or in the same package. Rather, it can span entire data centers, as we observe in today's machine learning model training clusters. For platform-specific chips, however, many constrained platforms will *not* tolerate significantly larger areas. For a 10 to 30 times larger area, 3D integration leads to high power density, and 2D integration will be challenging to fit in the physical dimensions of today's mobile phones, tablets, and personal computers.

C. The 3D Integration Tradeoff

Bright silicon does not allow an unlimited number of 3D integrated layers. Chip footprint reduction leads to increased power density. Figure 11 shows the maximum amount of 3D-stacked layers allowed by liquid cooling for each technology node, assuming a starting point at the power density of the hypothetical GPU in Section IV-A and then following the trends in Figure 8. Power efficiency improvements in A3.5 through A1.8 allow more room for 3D integration, at the cost of slightly reduced performance. We believe that application-specific chips will likely refrain from more 3D integration, as dark silicon does not contribute to chip utility. Platform-specific chips in dimensionally constrained platforms may exceed cooling limits as it does not expect high utilization.

❸ Footprint reduction cannot be enjoyed with bright silicon at the same time. Under liquid cooling, 3D integration will be limited to 2 to 6 active layers by A1.8 in 2039.

Table V summarizes the new scaling relationship in the bright silicon era, where technology scaling no longer forces

TABLE V: Projected new scaling trends in the bright silicon era, inspired by and builds upon a similar table in Conservation Cores [60].

Param.	Description	Dennard	Dark Silicon	Bright Silicon
D	device per sys.	S^2	S^2	S^2
P	total power	1	1	$\approx S^2$
A	total area	1	1	$\approx S^2$
F	total footprint	1	1	design choice
\$	cost	1	1	$\approx S^2$
p	device power	$1/S^2$	1	≈ 1
W, L	transistor dimension	1/S	1/S	≈ 1
f	frequency	S	S	≈ 1
P_d	power density	1	S^2	$\approx A/F$
U	utilization	1	$1/S^2$	$\approx F/A$

dark silicon. The target D, integrating more transistors per system stays the same. In the era of Dennard Scaling, the number of integrated components was increased under constant power, area, and cost. Power scaling led to frequency scaling, contributing to most of the performance gains. The main limitation in the golden age of transistor scaling was that the number of integrated transistors had not yet become sufficient. In the dark silicon era, power efficiency no longer scales. Architectures needed to make use of more transistors to gain performance. Both multi-core systems and specialized accelerators are in response to this new scaling trend. When transistor density continued to scale, increasing power density led to utilization limits, causing dark silicon. Finally, as we show, area density improvements are slowing down. The increase in power density returns to a low level. More transistors can be turned on. Footprint is largely a design choice balancing cooling and dimensional constraints, leading to tunable power density and utilization. This tradeoff is largely limited if dark silicon forces low utilization even for a single layer of transistors. Performance and the number of integrated transistors become more directly correlated with the total amount of area and power that can be dedicated to an application. In the subsequent section, we will discuss cost as another resource that will also be correlated.

V. Cost Escalation

A. The Challenge

Cost scaling was an important observation leading to the introduction of Moore's Law. At its time, the cost per integrated component decreased exponentially as technology evolved [39]. Cost governs the hidden condition in the most important colloquial implication of Moore's Law: a computer (of the same price) will have double the performance every two years. Unfortunately, our projection shows that cost scaling is no longer happening. Based on cost scaling projections from our technology model, Figure 12 shows very limited performance per unit cost improvement for the same design scaled to future nodes:

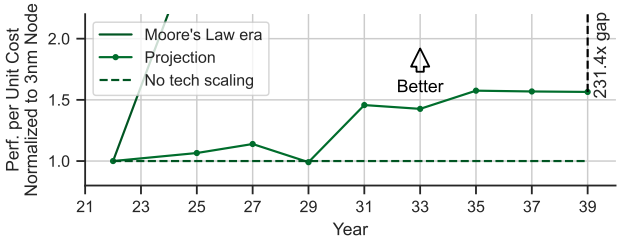


Fig. 12: Projected performance per unit cost of the same design based on speed and transistor cost scaling factors.

Cost Escalation
❶ With stalling performance per unit cost, more performance requires more money.

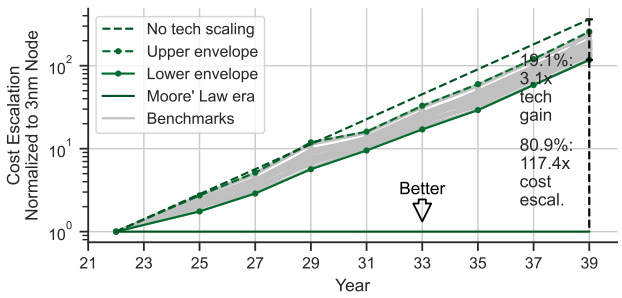
To meet Moore's Law era exponential performance targets, accelerators use more area and transistors as we show in the area bloating phenomenon. We estimate a lower bound on cost per area increase based on transistor cost and minimum density scaling factors, then apply the cost increase to area bloating. We assume that 3D tier stacking does not significantly impact transistor costs. Figure 13 shows cost escalation for throughput and latency targets:

Cost Escalation
❷ By A1.8 in 2039, in order to match Moore's Law era performance gain expectations, costs can hike to over 117.4× of 3nm's, contributing to over 80% of performance gains.

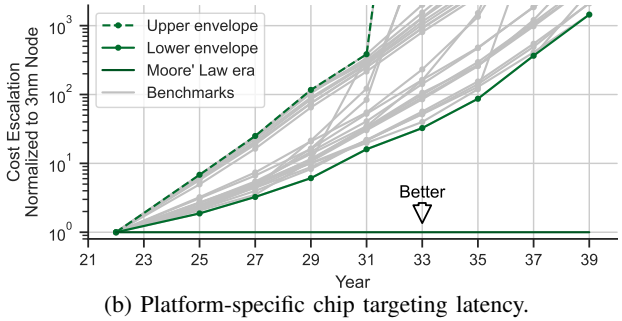
The cost curves in Figure 13a are dangerously close to, and sometimes exceeding, the no-technology-scaling line. In that scenario, there would be minimal motivation to move to new technology nodes from a cost-effectiveness perspective. Mass production still has the potential to drive down the costs of new nodes in the long run, but initial adoption of new technology will be limited. The dynamism of the semiconductor industry will be greatly harmed if cost-effectiveness cannot be improved. Even with the generous assumption that optimum design partition and chiplet integration can cut the manufacturing cost in half by A1.8 [18], [36], it still hardly alleviates the escalation. Aggressive area-saving designs and alternative technologies are to be expected.

B. The Adoption of New Nodes

As shown in Figure 12, new technology nodes do not have significantly better cost effectiveness. Manufacturers may choose not to adopt new nodes right away but instead wait until they become mature. As technology matures, cost is expected to go below what is predicted by our technology model, which reflects costs at initial mass production. We use an idealized cost and demand relationship to quantify the adoption of advanced technology nodes after they enter mass production. We predict that future studies will also include



(a) Application-specific chip targeting throughput.



(b) Platform-specific chip targeting latency.

Fig. 13: Projected cost escalation incurred by area bloating for area efficiency optimized designs.

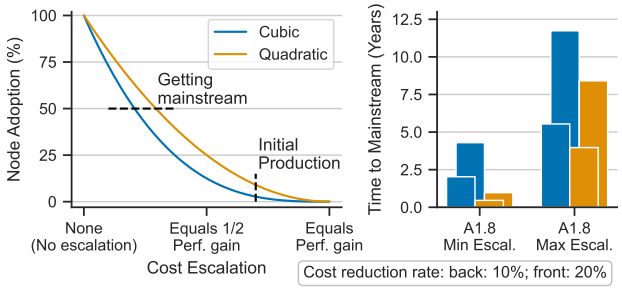


Fig. 14: Estimated adoption of future technology nodes. Legend in the left figure.

more analysis on the maturing of technology nodes. As shown in Figure 14 (left), we assume two models of adoption vs. cost for a technology node: quadratic and cubic. The two models show possible scenarios where the adoption is expected to be more rapid when the cost is lower. With no cost escalation, we assume a 100% adoption. We assume no adoption for cost escalation that equals the performance gain from the node. We evaluate the time it takes for a technology node to reach 50% adoption from its initial production. We assume the cost escalation at initial production to be those for area efficiency optimized application-specific chips in Figure 13a.

To estimate the pace of cost reductions of technology nodes, we follow Wright's Law [64]. In the Moore's Law era, transistor costs achieved 20% to 30% reductions per year [13] with a combination of node advancement and maturation. Future cost reductions can be more difficult. We show estimations assuming an optimistic 20% per year and a more realistic 10% reduction for future technology nodes. Applying Wright's

Law to the node adoption curves, we calculate the number of years it takes for a node to reach mainstream after its initial production. Figure 14 (right) shows that the adoption of the A1.8 technology node can be very slow. Under the realistic cost reduction scheme:

> ❸ High initial costs of future technology nodes slow down their adoption. Mainstream adoption of A1.8 could happen 4.3 to 11.7 years after initial production.

This estimation may still be optimistic, as the delay in adoption will slow down the cumulative production of the most advanced nodes, further reducing their cost reduction rates. We believe that cost escalation is a much more severe problem than stalling density or power scaling. New technological solutions await to improve future transistors, but the industry may not recover from a vicious cycle of escalating costs and declining adoption.

C. Design for Longevity

From an architectural perspective, in the next decade, designs must be prepared to last longer. With the adoption of new technology nodes slowed down, platforms that used to update rapidly, like mobile phones and personal computers, will likely see increased lifespans. Application-specific chips in data centers will also be updated less frequently if power efficiency plateaus. Chips may also need to function reliably for up to 14 years, as shown in Figure 14.

To reduce silicon upgrade costs, we predict that specialized accelerators will anticipate future applications. Fully specialized accelerators can achieve better performance per cost at one time, but frequent replacements drive up the cost in the long run. Reconfigurable specialization may achieve better performance per cost by supporting more applications. In response to rapidly changing and emerging programs, the inclusion of general-purpose reconfigurable architectures may also bring benefits to performance by extending the specialization envelope. We leave a quantitative analysis of hardware design for longevity to future work.

From a software perspective, programs may expect to run on the same hardware for longer periods of time instead of constantly switching to new chips and architectures. Algorithm, program, and compiler optimizations tailored for specific hardware may become more predominant. Program analysis can also become more important in guiding specialized hardware to anticipate and adapt to future applications.

VI. Related Work

Prior studies predicted future trends of computer architecture in their respective times and technology landscapes:

a) 1990s and before: Limits of Single Core Scaling: Several early works in the field studied performance limitations of CPU designs by measuring available instruction-level parallelism (ILP) from benchmark programs. Some studies have focused on the effects of certain constructs in programs like conditional jumps [30], [48]. Some analyze architectural designs to exploit ILP [42], [50], [61]. Many of these studies showed that multiple flows of control are necessary to extract more parallelism and that existing control flow descriptions in programs may hide ILP. These understandings contribute to multi-core processors and parallel programming. As Moore's Law was well alive, studies at the time were optimistic in predicting the continuation of the scaling law [40], [51].

b) 2000s: Multi-Core Scaling to a Billion transistors: As architectures with replicated processing units become popular, studies systematically explore the multi-core design space and characterize parallel applications [2], [37]. Moore provided an optimistic projection on the continuation of his law [41], predicting that the historical rate of improvements can continue "for at least several more generations."

c) 2010s: Limits of Multi-Core Scaling: Studies reveal the limitations of multi-core scaling by extending Amdahl's Law [21] and considering technology scaling trends [7]. Studies predicted the dark silicon phenomenon where a power bound limits chip utilization [11], [56], [60]. Studies focused on CPUs [19], [20] corroborate limits in power-constrained platforms. These studies motivated the development of specialized hardware accelerators. Several studies demonstrated the difficulty of keeping up with Moore's Law and provided alternative directions for investigation [5], [33], [58]. Recent works also studied the cost of silicon. Some studies derive silicon manufacturing cost models of existing technology nodes [27], [28], [43]. Findings show that the latest technology node may not bring the best profitability [43]. We show that cost escalation extends to future nodes and quantify its severity.

Recent works analyzed and modeled specialized accelerators. Aladdin [53]'s modeling of accelerators forms the foundation of the accelerator model in this study. The ISA-independent modeling approach based on LLVM [31] was introduced in a prior work [52]. A subsequent extension to Aladdin [54] and similar works [8], [38], [49] made it possible to simulate accelerators integrated with traditional multi-core systems. They provide more accurate results for practical implementations of accelerator systems, some at cycle-level accuracy. In this study, we use more ideal models to understand fundamental limitations and trends to reveal solutions and directions in the long term. The Accelerator Wall [15] retrospectively separates technology gains and architectural improvements in existing hardware accelerators. As multi-chip systems begin to prevail, single-reticle resource limitations no longer apply. Future accelerator designs may also adapt to changing technology trends. This work quantifies resource demands based on performance targets and provides insight into methods to meet those demands.

Prior works studied reconfiguration. Approaches similar to reconfigurable specialization exist. Behavior-specialized acceleration specializes not to applications but their common behaviors [45], [46]. The LSSD architecture [44] experiments with a more general form of specialization. To achieve better area efficiency, these designs, and CGRAs in general [12], [16], [17], [55], can target narrower ranges of applications

with reduced programmability. Studies explored methods to discover bottleneck regions and common regions for acceleration [3], [29], [59], [62]. They can be used to generate designs for reconfigurable specialization. Studies also explored methods to adapt to future applications for longevity. Conservation cores can patch accelerators to run future versions of the same program [60]. A recent work adapts CGRAs to run non-native operations [63]. Chiplet common interposers [34], [35] allow physical addition of functionality.

VII. Conclusion

Since its advent, computer architecture has bridged the gap between applications and physical technologies. Changes in transistor scaling trends have greatly influenced the entire computing stack. In the age of specialization and approaching the end of Moore's Law, we continue the tradition of projecting limitations, challenges, and possible directions in future architectures, assuming current trends and approaches continue. This time, advanced packaging and high bandwidth inter-chip communication blur the line between chips and systems. But speed, density, and power scaling of transistors are facing limits. We combine a technology model, an accelerator model, and a chip model to understand the demand of future chips to continue a performance scaling trend like that in the Moore's Law era. We find three new trends bringing challenges and opportunities to specialization: area bloating, the footprint vs. bright silicon tradeoff, and cost escalation.

From single core, multi-core, to the specialization age, architects find methods to translate an abundance of transistors to application performance. The looming failure of transistor density scaling and the consequent area bloating phenomenon we observe are bringing a completely different scaling reality, where efforts are likely needed to save transistors and area. Physical dimensions and costs can become bounding factors in the near future. Reconfigurable specialization, an under-explored approach in-between general and specialized architectures, may help respond to the area bloating challenge. Dennard scaling had failed, but the limitation on power is not as severe as dark silicon once predicted. More total power is required, but power density becomes bounded. A few layers of transistors can be bright even without immersion cooling. A more alarming trend comes from the escalation of costs in advanced technology. When price hikes delay the adoption of new nodes, designs may need to consider elongated lifetimes from reliability and functionality perspectives.

Acknowledgments

We would like to thank the entire Princeton Parallel Group for their suggestions on this work. We also thank the anonymous reviewers for their feedback. This research was supported in part by a Princeton SEAS Innovation Award, a Princeton SEAS Seed Grant, and a Princeton Andlinger Center Innovation Award.

References

[1] B. Agostini, M. Fabbri, J. E. Park, L. Wojtan, J. R. Thome, and B. Michel, "State of the art of high heat flux cooling technologies," *Heat transfer engineering*, vol. 28, no. 4, pp. 258–281, 2007.

[2] K. Asanovic, R. Bodik, B. Catanzaro, J. Gebis, P. Husbands, K. Keutzer, D. Patterson, W. Plishker, J. Shalf, and S. W. Williams, "The landscape of parallel computing research: A view from berkeley," 2006.

[3] C. Bai, J. Huang, X. Wei, Y. Ma, S. Li, H. Zheng, B. Yu, and Y. Xie, "Archexplorer: Microarchitecture exploration via bottleneck analysis," in *2023 56th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2023, pp. 268–282.

[4] P. Birbarah, T. Gebrael, T. Foulkes, A. Stillwell, A. Moore, R. Pilawa-Podgurski, and N. Miljkovic, "Water immersion cooling of high power density electronics," *International Journal of Heat and Mass Transfer*, vol. 147, p. 118918, 2020.

[5] S. Borkar and A. A. Chien, "The future of microprocessors," *Commun. ACM*, vol. 54, no. 5, p. 67–77, May 2011.

[6] J. Bucek, K.-D. Lange, and J. v. Kistowski, "SPEC CPU2017: Next-generation compute benchmark," in *Companion of the 2018 ACM/SPEC International Conference on Performance Engineering*, ser. ICPE '18. Association for Computing Machinery, 2018, p. 41–42.

[7] K. Chakraborty, "Over-provisioned multicore systems," Ph.D. dissertation, University of Wisconsin–Madison, 2008.

[8] J. Cong, Z. Fang, M. Gill, and G. Reinman, "PARADE: A cycle-accurate full-system simulation platform for accelerator-rich architectural design and exploration," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '15. IEEE Press, 2015, p. 380–387.

[9] A. Danalis, G. Marin, C. McCurdy, J. S. Meredith, P. C. Roth, K. Spafford, V. Tipparaju, and J. S. Vetter, "The scalable heterogeneous computing (SHOC) benchmark suite," in *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*, ser. GPGPU-3. Association for Computing Machinery, 2010, p. 63–74.

[10] R. Dennard, F. Gaensslen, H.-N. Yu, V. Rideout, E. Bassous, and A. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[11] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *2011 38th Annual International Symposium on Computer Architecture (ISCA)*, 2011, pp. 365–376.

[12] K. Feng, T. Kong, K. Koul, J. Melchert, A. Carsello, Q. Liu, G. Nyengele, M. Strange, K. Zhang, A. Nayak, J. Setter, J. Thomas, K. Sreedhar, P.-H. Chen, N. Bhagdikar, Z. A. Myers, B. D'Agostino, P. Joshi, S. Richardson, C. Torng, M. Horowitz, and P. Raina, "Amber: A 16-nm system-on-chip with a coarse- grained reconfigurable array for flexible acceleration of dense linear algebra," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 3, pp. 947–959, 2024.

[13] K. Flamm, "Measuring Moore's law: evidence from price, cost, and quality indexes," in *Measuring and Accounting for Innovation in the 21st Century*. University of Chicago Press, 2019, ch. 11, pp. 403–470.

[14] A. Fuchs, "Repository for the tools and non-commercial data used for the 'Accelerator wall' paper," 2019. [Online]. Available: https://github.com/PrincetonUniversity/accelerator-wall

[15] A. Fuchs and D. Wentzlaff, "The accelerator wall: Limits of chip specialization," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2019, pp. 1–14.

[16] G. Gobieski, S. Ghosh, M. Heule, T. Mowry, T. Nowatzki, N. Beckmann, and B. Lucia, "RipTide: A programmable, energy-minimal dataflow compiler and architecture," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2022, pp. 546–564.

[17] V. Govindaraju, C.-H. Ho, T. Nowatzki, J. Chhugani, N. Satish, K. Sankaralingam, and C. Kim, "DySER: Unifying functionality and parallelism specialization for energy-efficient computing," *IEEE Micro*, vol. 32, no. 5, pp. 38–51, 2012.

[18] A. Graening, S. Pal, and P. Gupta, "Chiplets: How small is too small?" in *Proceedings of the 60th Annual ACM/IEEE Design Automation Conference*, ser. DAC '23. IEEE Press, 2023, p. 1–6.

[19] M. Halpern, Y. Zhu, and V. J. Reddi, "Mobile CPU's rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016, pp. 64–76.

[20] M. Hempstead, G.-Y. Wei, and D. Brooks, "Navigo: An early-stage model to study power-constrained architectures and specialization," in *Workshop on Modeling, Benchmarking, and Simulations*, 2009.

[21] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," *Computer*, vol. 41, no. 7, pp. 33–38, 2008.

[22] IEEE International Roadmap for Devices and Systems, "International Roadmap for Devices and Systems (IRDS™) 2020 Edition," 2020, (accessed July 10, 2025). [Online]. Available: https://irds.ieee.org/editions/2020

[23] ——, "International Roadmap for Devices and Systems (IRDS™) 2022 Edition," 2022, (accessed July 10, 2025). [Online]. Available: https://irds.ieee.org/editions/2022

[24] ——, "International Roadmap for Devices and Systems (IRDS™) 2024 Edition," 2024, (accessed July 10, 2025). [Online]. Available: https://irds.ieee.org/editions/2024

[25] S. Jones, "ISS 2024 – logic 2034 – technology, economics, and sustainability," 2024. [Online]. Available: https://semiwiki.com/semiconductor-manufacturers/342094-iss-2024-logic-2034-technology-economics-and-sustainability/

[26] S. Kandlikar and H. Upadhye, "Extending the heat flux limit with enhanced microchannels in direct single-phase cooling of computer chips," in *Semiconductor Thermal Measurement and Management IEEE Twenty First Annual IEEE Symposium, 2005.*, 2005, pp. 8–15.

[27] S. M. Khan and A. Mann, "AI Chips: What They Are and Why They Matter," 2020, doi: https://doi.org/10.51593/20190014.

[28] M. Khazraee, L. Zhang, L. Vega, and M. B. Taylor, "Moonwalk: NRE optimization in ASIC clouds," in *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '17. Association for Computing Machinery, 2017, p. 511–526.

[29] S. Kumar, N. Sumner, V. Srinivasan, S. Margerm, and A. Shriraman, "Needle: Leveraging program analysis to analyze and extract accelerators from whole programs," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2017, pp. 565–576.

[30] M. S. Lam and R. P. Wilson, "Limits of control flow on parallelism," in *Proceedings of the 19th Annual International Symposium on Computer Architecture*, ser. ISCA '92. Association for Computing Machinery, 1992, p. 46–57.

[31] C. Lattner and V. Adve, "LLVM: a compilation framework for lifelong program analysis transformation," in *International Symposium on Code Generation and Optimization, 2004. CGO 2004.*, 2004, pp. 75–86.

[32] G. Lauterbach, "The path to successful wafer-scale integration: The Cerebras story," *IEEE Micro*, vol. 41, no. 6, pp. 52–57, 2021.

[33] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez, and T. B. Schardl, "There's plenty of room at the top: What will drive computer performance after Moore's law?" *Science*, vol. 368, no. 6495, p. eaam9744, 2020.

[34] F. Li, Y. Wang, Y. Cheng, Y. Wang, Y. Han, H. Li, and X. Li, "GIA: A reusable general interposer architecture for agile chiplet integration," in *2022 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2022, pp. 1–9, ISSN: 1558-2434.

[35] Z. Li and D. Wentzlaff, "LUCIE: A universal chiplet-interposer design framework for plug-and-play integration," in *Proceedings of 57th IEEE/ACM International Symposium on Microarchitecture*, 2024.

[36] X.-W. Lin, V. Moroz, X. Xu, Y. Gao, D. Rennie, P. Asenov, S. Smidstrup, D. Sherlekar, Z. Qin, T. Fang, J. Lee, M. Choi, and S. Jones, "Heterogeneous integration enabled by the state-of-the-art 3DIC and CMOS technologies: Design, cost, and modeling," in *2021 IEEE International Electron Devices Meeting (IEDM)*, 2021, pp. 3.4.1–3.4.4.

[37] J. Mak and A. Mycroft, "Limits of parallelism using dynamic dependency graphs," in *Proceedings of the Seventh International Workshop on Dynamic Analysis*, ser. WODA '09. Association for Computing Machinery, 2009, p. 42–48.

[38] O. Matthews, A. Manocha, D. Giri, M. Orenes-Vera, E. Tureci, T. Sorensen, T. J. Ham, J. L. Aragon, L. P. Carloni, and M. Martonosi, "MosaicSim: A lightweight, modular simulator for heterogeneous systems," in *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2020, pp. 136–148.

[39] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

[40] ——, "Lithography and the future of Moore's law," in *Proc. SPIE*, vol. 2437, 1995, pp. 2 – 17.

[41] ——, "No exponential is forever: but 'forever' can be delayed!" in *2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC.*, 2003, pp. 20–23 vol.1.

[42] A. Nicolau and J. A. Fisher, "Measuring the parallelism available for very long instruction word architectures," *IEEE Transactions on Computers*, vol. C-33, no. 11, pp. 968–976, 1984.

[43] A. Ning, G. Tziantzioulis, and D. Wentzlaff, "Supply chain aware computer architecture," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. Association for Computing Machinery, 2023.

[44] T. Nowatzki, V. Gangadhan, K. Sankaralingam, and G. Wright, "Pushing the limits of accelerator efficiency while retaining programmability," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016, pp. 27–39.

[45] T. Nowatzki, V. Govindaraju, and K. Sankaralingam, "A graph-based program representation for analyzing hardware specialization approaches," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 94–98, 2015.

[46] T. Nowatzki and K. Sankaralingam, "Analyzing behavior specialized acceleration," in *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '16. Association for Computing Machinery, 2016, p. 697–711.

[47] B. Reagen, R. Adolf, Y. S. Shao, G.-Y. Wei, and D. Brooks, "Machsuite: Benchmarks for accelerator design and customized architectures," in *2014 IEEE International Symposium on Workload Characterization (IISWC)*, 2014, pp. 110–119.

[48] E. Riseman and C. Foster, "The inhibition of potential parallelism by conditional jumps," *IEEE Transactions on Computers*, vol. C-21, no. 12, pp. 1405–1411, 1972.

[49] S. Rogers, J. Slycord, M. Baharani, and H. Tabkhi, "gem5-SALAM: A system architecture for LLVM-based accelerator modeling," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020, pp. 471–482.

[50] B. J. Sano and A. M. Despain, "The 16-fold way: a microparallel taxonomy," in *Proceedings of the 26th Annual International Symposium on Microarchitecture*, 1993, pp. 60–69.

[51] R. Schaller, "Moore's law: past, present and future," *IEEE Spectrum*, vol. 34, no. 6, pp. 52–59, 1997.

[52] Y. S. Shao and D. Brooks, "ISA-independent workload characterization and its implications for specialized architectures," in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2013, pp. 245–255.

[53] Y. S. Shao, B. Reagen, G.-Y. Wei, and D. Brooks, "Aladdin: A pre-rtl, power-performance accelerator simulator enabling large design space exploration of customized architectures," in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, 2014, pp. 97–108.

[54] Y. S. Shao, S. L. Xi, V. Srinivasan, G.-Y. Wei, and D. Brooks, "Co-designing accelerators and SoC interfaces using gem5-Aladdin," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016, pp. 1–12.

[55] S. Swanson, K. Michelson, A. Schwerin, and M. Oskin, "WaveScalar," in *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.*, 2003, pp. 291–302.

[56] M. B. Taylor, "Is dark silicon useful? harnessing the four horsemen of the coming dark silicon apocalypse," in *DAC Design Automation Conference 2012*, 2012, pp. 1131–1136.

[57] The Standard Performance Evaluation Corporation (SPEC), "SPEC CPU 2017," 2017. [Online]. Available: https://www.spec.org/cpu2017/

[58] T. N. Theis and H.-S. P. Wong, "The end of Moore's law: A new beginning for information technology," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[59] D. Trilla, J.-D. Wellman, A. Buyuktosunoglu, and P. Bose, "NOVIA: A framework for discovering non-conventional inline accelerators," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '21. Association for Computing Machinery, 2021, p. 507–521.

[60] G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin, J. Lugo-Martinez, S. Swanson, and M. B. Taylor, "Conservation cores: reducing the energy of mature computations," in *Proceedings of the Fifteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS XV. Association for Computing Machinery, 2010, p. 205–218.

[61] D. W. Wall, "Limits of instruction-level parallelism," in *Proceedings of the Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS IV. Association for Computing Machinery, 1991, p. 176–188.

[62] J. Weng, S. Liu, V. Dadu, Z. Wang, P. Shah, and T. Nowatzki, "DSAGEN: Synthesizing programmable spatial accelerators," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, 2020, pp. 268–281.

[63] J. Woodruff, T. Koehler, A. Brauckmann, C. Cummins, S. Ainsworth, and M. F. O'Boyle, "Rewriting history: Repurposing domain-specific CGRAs," *arXiv preprint arXiv:2309.09112*, 2023.

[64] T. P. WRIGHT, "Factors affecting the cost of airplanes," *Journal of the Aeronautical Sciences*, vol. 3, no. 4, pp. 122–128, 1936.