Technical Report

How Does EDA Licensing Structure Affect Carbon Emissions? A Cost, Time, and Accounting-Scope Analysis

Aishwarya Sivakumar
Andreas Gerstlauer

UT-CERC-26-01

August 7, 2026

Computer Engineering Research Center
Chandra Family Department of Electrical and Computer Engineering
The University of Texas at Austin

The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
Cockrell School of Engineering

How Does EDA Licensing Structure Affect Carbon Emissions? A Cost, Time, and Accounting-Scope Analysis

Aishwarya Sivakumar, Andreas Gerstlauer

Electrical and Computer Engineering
The University of Texas at Austin

Abstract

This report examines how EDA licensing structure affects hardware refresh decisions and the resulting cost, time, and carbon outcomes. It compares per-core-year licensing with a performance-agnostic work-based alternative, using measured performance and power trends for Intel and AMD processors and evaluating fixed-budget and fixed-deadline scenarios. The analysis finds that per-core licensing creates a refresh incentive, while work-based licensing can reduce hardware replacement under several customer constraints. However, under the measured processor trends used in this study, continuously refreshing to newer, more efficient cores can result in lower cumulative carbon emissions than retaining older hardware. The report therefore identifies the conditions under which changing the licensing structure would also improve carbon outcomes, including the relative rates of per-core performance and power improvement and the granularity at which workloads use processor capacity.

Table of Contents

Chapter 1: Introduction

The growth of the semiconductor industry has resulted in tremendous improvements in power, performance, and area of CPUs. At the same time, both the usage and manufacturing of such CPUs requires carbon-fueled energy sources. At the current rates of usage, the carbon emissions released by the Information Communication Technology sector ranges from 1.8-2.8% as of 2021, and are expected to rise yearly [1, 2]. This is comparable to the share attributed to the aviation industry [3]. Given the current demand for GPUs and CPUs used for mass computing, such as within data centers and for AI, this emission rate is expected to increase to almost 8% over the next decade [3].

Carbon emissions from computing are comprised of two categories: embodied emissions from manufacturing, and operational emissions from the energy consumed while running the CPU. The operational emissions depend on the carbon intensity of the energy source used [3]. Embodied emissions are meanwhile determined by the manufacturing process for the node type and the rate of CPU manufacturing, and therefore the rate of CPU replacement. CPU lifetimes are often 5-10 years, though companies frequently replace them en masse as more performant CPUs are released [3]. This paper investigates one particular driver of that replacement behavior: EDA tool licensing.

EDA (Electronic Design Automation) vendors such as Cadence and Synopsys typically license their silicon design and verification tools on a per core-year basis. Customers use these tools to run millions of validation jobs before fabrication.

The per-core licensing scheme tends to favor more performant CPUs since the per-core-year licensing charges the same no matter the CPU model used. A faster core completes more work within the same licensed year, making constant hardware refresh economically attractive, regardless of whether the old hardware still works.

Against this default model, we construct and analyze a performance-agnostic alternative that charges by unit of work completed rather than by core-year. Similar models have slowly started to be offered by EDA companies. Both Synopsys and Cadence have started offering token-based licensing [4]. Under this model, customers pre-purchase a pool of tokens rather than a fixed number of per-core licenses, and each tool usage draws down tokens at a rate that varies by tool and feature. AI-driven design tools such as Synopsys DSO.ai and Cadence Cerebrus, which run many automated iterations per project, consume tokens substantially faster than a single conventional simulation run [4]. This creates an incentive for companies to use older hardware, if completing one simulation run uses the same amount of tokens as it would on the newer hardware. Given that token-based licensing is not fully detailed, this paper constructs an idealized performance-agnostic model to serve as an alternate scheme. Both the per-core licensing and the per-work licensing serve valid needs for different financial budgets, such as for large en masse data centers vs. smaller semiconductor start ups.

This report explores how the choice of licensing structure interacts with hardware refresh economics. It also investigates what that interaction implies for embodied and operational carbon emissions. We find that the two schemes result in varying emission outcomes, impacted by the relative growth of per-core power and performance across hardware generations, and by the granularity at which a workload actually uses a socket. Specifically, we investigate:

- Whether, and under what customer time and budget constraints, per-core-year EDA licensing incentivizes the replacement of functional CPUs with newer, performant ones.
- How that replacement incentive translates into embodied and operational carbon emissions, and how sensitive that translation is to the rate at which per-core power and performance change generation to generation.

- Whether a performance-agnostic (work-based) licensing model changes this picture for the EDA vendor, the customer, and cumulative carbon.
- When the work-based licensing does not uniformly reduce emissions, and what conditions would need to hold for it to in the future.

Overall, for the per-core licensing model, we find that *current* industry power and performance trends use less carbon by annually refreshing cores, rather than the work-based alternative. Both Intel and AMD presently sit below the crossover point (Section 4.5.1) at which work-based licensing would reduce operational carbon. Per-core efficiency is improving fast enough, for both vendors, that replacing cores with newer generations currently produces less net carbon than holding onto older, less efficient ones. The report's contribution is therefore twofold: (1) licensing model simulations with several parameter sweeps, showing that per-core-year licensing creates a refresh incentive that work-based pricing removes by construction; and (2) a quantitative characterization of exactly when removing that incentive would also help carbon outcomes, via finding the crossover points when one licensing model is more carbon efficient than the other.

Chapter 2 first collects the empirical inputs: license and core costs, measured power/performance trends across five hardware generations for both vendors, and the baseline carbon parameters. Chapter 3 then builds the complete mathematical model symbolically: cost and time bounds general to both schemes, equations specific to the core-based and work-based schemes, and finally the carbon model. Chapter 4 applies that model to three customer scenarios and reports the resulting cost, time, and carbon outcomes, followed by Chapter 5 with this report's modeling assumptions and their consequences, and Chapter 6.

Chapter 2: Empirical Background

We first collect the starting parameters needed to build each licensing model: license costs, core costs, measured power/performance trends, and baseline carbon intensities.

2.1 License and Core Costs

EDA vendors price licenses on an individualized, non-public basis per company. We used industry-forum estimates and conversations with leadership in companies that use such tools to estimate that per-core licenses cost 10-100× the cost of a single core [5]. Based on the same conversations / forums, we take $10,000/seat as a representative per-core license price per year. In terms of work-based license pricing, the sliding scale of cost per work is discussed below in Section 3.4. This is the pricing scale assumed for the framework of this report.

Customers running EDA tools typically seek processor sockets with high frequency, throughput and core counts for maximum job parallelism [5]. We selected AMD EPYC Zen-series and Intel Xeon Platinum/Gold parts. AMD EPYC cores are priced $65-$85 / core [6] while Intel Xeon Platinum/Gold cores $65-$190 / core [7]. We take $65 - $200 as the overall core-cost range and $100 as a representative value. This gives a representative license-to-core (LC) cost ratio of 100:1.

2.2 Performance Trends

To determine typical per-generation performance improvements, we collected SPECint_rate2017 results for multiple generations of Intel Xeon Gold, Intel Xeon Platinum, and AMD EPYC. We filtered the results to dual-socket configurations (`EnabledChips = 2`), taking the top Base score per generation (Figure 2.1, Table 2.1).

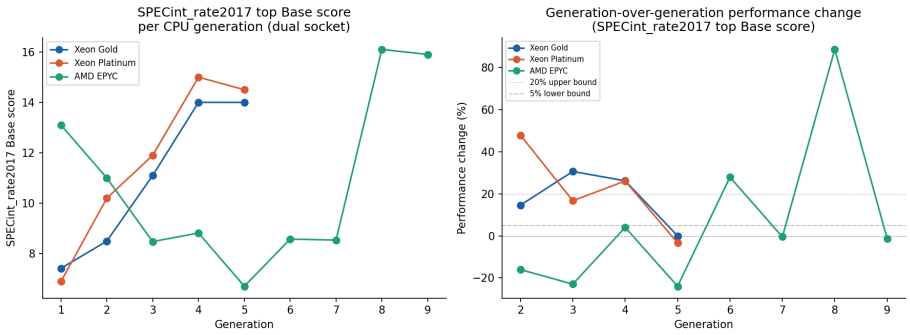

Figure 2.1: Left: SPECint_rate2017 top Base score per generation for Intel Xeon Gold, Intel Xeon Platinum, and AMD EPYC (dual-socket configurations). Right: per-generation performance delta (%).

CPU	**CAGR**	**Mean YoY**	**Range**
Xeon Gold	+17.2%	+17.8%	0% to +31%
Xeon Platinum	+20.4%	+21.8%	−3% to +48%
AMD EPYC	+2.5%	+7.0%	−24% to +89%

Table 2.1: SPECint_rate2017 socket-level throughput trends per CPU family. CAGR = compound annual growth rate across generations. YoY = year-over-year change

These are *socket-level* throughput figures. To determine the per-core throughput instead, SPECrate2017_int_base is divided by the core count. The report proceeds with the per-core figure to stay consistent with the per-core licensing that is currently offered, for consistent modeling. Per-core throughput rose approximately 26% for Intel (from 6.98 to 8.83, 2018–2023) [8]. For AMD, it rose approximately +15.4%/gen. As endpoint CAGRs, these give per-core performance-growth factors $f_{\text{Intel}} = 1.082$ and $f_{\text{AMD}} = 1.154$ per *generation*.

Performance growth f (and power growth i below) here are measured per hardware generation, but the model in Section 3 indexes growth by year t and applies it as f^t (and i^t). This report therefore models one generation as one year. In reality, new processor generations are released at a rate of 1.5-2 years. For the sake of easier modeling, we assume new generations are released every year. Section 5 records this as an assumption.

As a note, AMD shows an irregularity in the trend of SPEC CPU2017 scores, vs. the constantly increasing Intel scores. One possible reason for this is that the scores are self-submitted by individual OEMs over months or years after a chip launches. Therefore, the top score for any generation reflects how many, and which, configurations have been submitted rather than the chip's true ceiling. A documented, closely analogous case is AMD Milan's early public benchmark appearance, which scored below the outgoing Rome generation in a similar dual-socket configuration, before higher-scoring Milan submissions were published in subsequent quarters [9].

A second reason could be that AMD has not always scaled memory bandwidth and cache proportionally with core count. The Naples-to-Rome transition doubled core count while memory channels stayed fixed, roughly halving memory bandwidth per core [10]. More recently, AMD's highest-density Turin SKUs explicitly trade per-core L3 cache for core count, and thus production deployments report measurably worse cache contention under load [11]. SPECint_rate2017 runs many simultaneous copies of each workload, which is the access pattern most exposed to shared cache and memory-bandwidth pressure. Therefore, a generation that increases core count faster than it scales memory bandwidth or cache can result in reduced per-copy throughput.

Regardless of the reason for this irregularity, this report attempts to address this via the endpoint-CAGR vs. generational-volatility robustness check in Section 4.5.1.

2.3 Power Trends

Table 2.2 and Table 2.3 contain maximum socket Thermal Design Power (TDP) per generation from vendor datasheets and hardware reviews, for the same five generations used above.

Total socket TDP rises across generations since core counts grow faster than total power. This is indicated as Intel experienced +71% growth in TDP over five generations, while AMD experienced +178% over the five generations. *Per-core TDP*

Generation (model)	TDP	Cores	TDP/core
Skylake-SP (8180)	205W	28	7.3W
Cascade Lake (8280)	205W	28	7.3W
Ice Lake-SP (8380)	270W	40	6.8W
Sapphire Rap. (8490H)	350W	60	5.8W
Emerald Rap. (8592+)	350W	64	5.5W

Table 2.2: Intel Xeon Scalable max socket TDP per generation [12–16].

Generation (model)	TDP	Cores	TDP/core
Naples, EPYC 7601	180W	32	5.6W
Rome, EPYC 7H12	280W	64	4.4W
Milan, EPYC 7763	280W	64	4.4W
Genoa, EPYC 9654	360W	96	3.8W
Turin, EPYC 9965	500W	192	2.6W

Table 2.3: AMD EPYC max socket TDP per generation [17–21].

(TDP ÷ cores, the last column) instead *falls* across generations. For Intel, the fall is from 7.3W $\rightarrow$ 5.5W, indicating a -6.83%/gen CAGR. For AMD, the fall was from 5.6W $\rightarrow$ 2.6W, indicating a -17.45%/gen. Per-core TDP is once again used to stay consistent with the per-core units for the models, including the carbon model. These results overall provide per-generation power-growth factors of $i_{\text{Intel}} = 0.932$ and $i_{\text{AMD}} = 0.826$.

2.4 Carbon Parameters

Embodied carbon per die is estimated as $C_{\text{die}} = CPA \times A$, where $CPA = 3.0$ kg CO_2e/cm^2 is the Xeon-specific manufacturing intensity [22], and $A = 7$cm^2 is an assumption for a high-core-count server die, based on numbers cited by consumers who unofficially analyzed the silicon die on sources such as Chips and Cheese, Reddit, etc. The real die area is seldom revealed since this figure remains under high contention for competing semiconductor companies. This gives $C_{\text{die}} \approx 21{,}000$g CO_2 per die. Since

this report's licensing unit is the core, we convert to a per-core figure by dividing by a baseline core count of Intel's earliest generation, 28 cores (Table 2.2). This results in $C_{\text{emb},0}^{\text{Intel}} = 21{,}000/28 = 750\text{g}$. For AMD's own baseline (32 cores, Table 2.3), $C_{\text{emb},0}^{\text{AMD}} = 21{,}000/32 \approx 656\text{g}$. AMD chips are fabricated by TSMC on a substantially more carbon-intensive grid than Intel's own fabs (roughly 6% renewable vs. 95%) [23, 24], so both figures likely understate true embodied carbon and should be read as lower bounds.

Operational carbon uses a baseline rate of $\text{CI}_0 = 3.154\text{g CO}_2\text{/core-hr}$ [25]. This rate is sourced externally as a "per-CPU-hour" figure without specifying whether "CPU" means core or socket. Unlike $C_{\text{emb},0}$, which we could re-derive from a stated die-area formula, we have no formula to decompose CI_0 similarly. We state this plainly as an inherited assumption, not an independently verified one: CI_0 is treated as per-core throughout this report.

The embodied-carbon shrink factor s captures the fact that per-core embodied carbon falls as more cores are placed on a die of fixed area: $s_{\text{Intel}} = (28/64)^{1/4} = 0.813$ (-18.7%/gen) and $s_{\text{AMD}} = (32/192)^{1/4} = 0.639$ (-36.1%/gen). Both i values established in Section 2.3 are likewise *under* 1, meaning per-core power intensity *falls* generation over generation. This is the opposite direction from what an aggregate socket-level reading would suggest. Section 4.5.1 additionally uses an abstract sweep of $i \in [0.8, 1.2]$ for analyzing the relationship, separate from the real-vendor result.

Chapter 3: Model

This section builds the cost, time, and carbon model for both licensing schemes. These model equations are then used with a few realistic examples to compare the licensing schemes in Section 4. Table 3.1 collects every symbol used below and marks each as a constant or a variable of the simulation.

3.1 Cost and Time Bounds

For the remainder of this report, we keep the amount of work done in a year as a constant, W. We have no intuition towards how W changes per company, so keeping this constant allows us to sweep other customer constraints, such as fleet size P, budget B, and deadline H. We specifically consider two distinct scenarios: a customer with a **fixed yearly deadline**, and a customer with a **fixed yearly budget**. These form two bounds for a customer choosing P. Total annual cost must stay within budget B. This is made up of two components: new hardware purchased that year (N), and license costs for running P cores that year. This gives:

$$P \cdot c_\ell + N \cdot c_h \leq B \tag{3.1}$$

Since you must purchase a core before running work on it, we will set $P = N$ for now, and will consider hardware reuse cases later (where P and N may differ). The P processors, each running at rate y (work units/hr), must complete total work W before deadline H:

$$\frac{W}{y \cdot P} \leq H \tag{3.2}$$

With an unlimited budget, one license per job (maximum parallelism) minimizes completion time. The cost bound is created since this is not affordable in practice.

Solving both budget and time bounds together results in a range of P:

$$\frac{W}{H \cdot y} \leq P \leq \frac{B}{c_\ell + c_h} \tag{3.3}$$

Symbol	Type	Description
Base parameters (shared across all cases)		
c_ℓ	const	Cost per license, per core, per year
c_h	const	Cost per core (hardware)
W	const	Total work per year (work units)
H	const	Job completion deadline (hours) per year
B	const	Customer's annual budget
L	const	Core lifetime in years; equivalently, the number of cohorts in a work-based fleet
Y	const	Simulation duration (years)
y_0	const	Baseline per-core performance (work units/hr)
y	derived	Per-core performance in a given year, $y = y_0 f^{\,t}$
f	const	Per-core performance growth factor, per generation
i	const	Per-core power growth factor, per generation
t	var	Year index, $t = 0, \ldots, Y-1$
d	var	Work-based price per unit of work
d^*	derived	Work-based breakeven price per unit of work
$N(t)$	var	Number of cores purchased in year t
Core-based, fixed-budget case		
$P_{\text{lic}}^{\text{budget}}$	derived	Fleet size (constant per t)
$N_{\text{lic}}^{\text{budget}}$	derived	Annual purchase, equals $P_{\text{lic}}^{\text{budget}}$ since the fleet is refreshed annually
$M_{\text{lic}}^{\text{budget}}$	derived	Annual monetary cost (constant per t)
$T_{\text{lic}}^{\text{budget}}(t)$	derived	Completion time in year t (falls as t rises)
Core-based, fixed-deadline case		
$P_{\text{lic}}^{\text{deadline}}(t)$	derived	Fleet size in year t (falls as t rises)
$N_{\text{lic}}^{\text{deadline}}(t)$	derived	Purchase in year t, equals $P_{\text{lic}}^{\text{deadline}}(t)$ since the fleet is refreshed annually
$M_{\text{lic}}^{\text{deadline}}(t)$	derived	Monetary cost in year t (falls as t rises)
$T_{\text{lic}}^{\text{deadline}}$	derived	Completion time (constant, $= H$ by construction)
Work-based, fixed-budget case		
$P_{\text{work}}^{\text{budget}}$	derived	Fleet size, $= L \cdot N_{\text{work}}^{\text{budget}}$
$N_{\text{work}}^{\text{budget}}$	var	Annual purchase equals $N_{\text{lic}}^{\text{budget}}$ when budgets and vendor revenue match
$M_{\text{work}}^{\text{budget}}$	derived	Annual monetary cost (constant per t)
$T_{\text{work}}^{\text{budget}}(t)$	derived	Completion time in year t (falls as t rises)
Work-based, fixed-deadline case		
$P_{\text{work}}^{\text{deadline}}(t)$	derived	Fleet size in year t, = sum of the last L purchases
$N_{\text{work}}^{\text{deadline}}(t)$	derived	Work-based cores purchased in year t under a fixed deadline; solves eq. (3.20)
$M_{\text{work}}^{\text{deadline}}(t)$	derived	Monetary cost in year t
$T_{\text{work}}^{\text{deadline}}$	derived	Completion time (constant, $= H$ by construction)
Cumulative quantities over Y years		
$\mathcal{M}(Y)$	derived	Cumulative monetary cost
$\mathcal{T}(Y)$	derived	Cumulative completion time
$\mathcal{C}_{\text{lic}}(Y)$	derived	Cumulative carbon, core-based model
$\mathcal{C}_{\text{work}}(Y)$	derived	Cumulative carbon, work-based model
Carbon model		
$C_{\text{emb},0}$	const	Baseline embodied carbon per core
CI_0	const	Baseline operational carbon per core-hr
s	const	Embodied-carbon shrink factor per generation
$C_{\text{emb}}(t)$	derived	Embodied carbon per core in year t
$\text{CI}(t)$	derived	Operational intensity in year t
$\Phi(i,t)$	derived	Cohort-generation intensity factor for the work-based fleet as a function of t
$\Delta C(Y)$	derived	Percentage carbon savings, work-based rel. to core-based

Table 3.1: Notation grouped by which of the four cases (core-based/work-based $\times$ fixed-budget/fixed-deadline) each quantity belongs to. "const" denotes a fixed input, "var" a swept parameter, and "derived" a quantity computed from the two.

the minimum number of cores needed to finish within deadline H on the left, and the maximum cores within budget B on the right. For a practical solution, the left side cores should not exceed the right.

3.2 Core-Based Licensing

Under per-core-year licensing, a customer annually purchases N_{lic} cores and one license per core. This is because we assume the cores are annually refreshed ($P = N$) - a further proof showing this is the most cost effective refresh strategy is provided under Section 4.1.

We now apply both scenarios from Section 3.1 to the core-based model in turn.

Core-based under the fixed-budget scenario: Applying Eq. (3.1), with B held constant, gives a fleet size that is constant in t:

$$P_{\text{lic}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}} = \left\lfloor \frac{B}{c_h + c_\ell} \right\rfloor, \tag{3.4}$$

$$M_{\text{lic}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}} c_h + P_{\text{lic}}^{\text{budget}} c_\ell = N_{\text{lic}}^{\text{budget}} \cdot (c_h + c_\ell). \tag{3.5}$$

If B is picked as tightly as possible, eq. (3.4) coincides with eq. (3.8) evaluated at $t = 0$, up to the difference between the floor and ceiling operators. The two scenarios agree at year 0 and diverge afterward, since fixed-budget holds cost flat while fixed-deadline lets it shrink.

Annual spend is constant in every year since $N_{\text{lic}}^{\text{budget}}$, c_h, and c_ℓ are all constant, so cumulative cost is $\mathcal{M}_{\text{lic}}^{\text{budget}}(Y) = M_{\text{lic}}^{\text{budget}} \cdot Y$. Completion time, by contrast, shrinks every year as the fixed-size fleet is replaced with faster cores:

$$T_{\text{lic}}^{\text{budget}}(t) = \frac{W}{P_{\text{lic}}^{\text{budget}} \cdot y_0 \cdot f^{\,t}}, \tag{3.6}$$

$$\mathcal{T}_{\text{lic}}^{\text{budget}}(Y) = \sum_{t=0}^{Y-1} T_{\text{lic}}^{\text{budget}}(t). \tag{3.7}$$

Core-based under the fixed-deadline scenario: Applying Eq. (3.2) with H held constant gives the number of cores needed in year t, $N_{\text{lic}}^{\text{deadline}}(t)$, and the cost in year t, $M_{\text{lic}}^{\text{deadline}}(t)$:

$$N_{\text{lic}}^{\text{deadline}}(t) = \left\lceil \frac{W}{H \cdot y_0 \cdot f^{\,t}} \right\rceil, \tag{3.8}$$

$$M_{\text{lic}}^{\text{deadline}}(t) = N_{\text{lic}}^{\text{deadline}}(t) \cdot (c_h + c_\ell). \tag{3.9}$$

t is years since baseline year 0. At each year's improved performance $y_0 \cdot f^{\,t}$, a new fleet size is calculated. This fleet size and the yearly cost shrink over time while keeping the constant deadline, since f is assumed consistently above 1. This customer always finishes exactly at H, spending less every year as performance improves. Cumulative cost over Y years is

$$\mathcal{M}_{\text{lic}}^{\text{deadline}}(Y) = \sum_{t=0}^{Y-1} M_{\text{lic}}^{\text{deadline}}(t). \tag{3.10}$$

3.3 Work-Based Licensing

Under work-based pricing, the customer pays a flat rate d per unit of work, so license cost is fixed at dW and there is no cost-based incentive to minimize active cores. Deploying an additional core costs only its hardware price, so the customer's lever becomes deploying *more* cores rather than faster ones. Unlike the core-based case, the fleet is no longer one year's purchase. Instead, cores are retained for their full lifetime L. Writing $N_{\text{work}}(t)$ for the number of cores purchased in year t, the fleet in year t is the sum of the last L purchases:

$$P_{\text{work}}(t) = \sum_{a=0}^{L-1} N_{\text{work}}(t-a). \tag{3.11}$$

This summation represents the total set of cores collected within only the last L years, since any cores purchased before that would have already reached lifetime and been disposed of. Everything below follows from $N_{\text{work}}(t)$. How it is set depends on which bound the customer is under.

As a note, within cohort sums we write $N_{\text{work}}(t-a)$ for the purchase made in year $t-a$, with the case (fixed budget/fixed deadline) understood from context.

Work-based under the fixed-budget scenario: Under a work-based license there is no longer a need to keep $P = N$. To determine how many cores are purchased each year for a given budget we run a similar calculation to core-based licensing. The annual spend is the work-rate charge $d\,W$ plus hardware cost $N_{\text{work}}^{\text{budget}}\,c_h$. For a budget B the purchase rate under a fixed budget is thus:

$$N_{\text{work}}^{\text{budget}} = \frac{B - d\,W}{c_h}. \tag{3.12}$$

We assume the EDA company charges a per-work rate that makes the annual license cost the same to not lose money, where $d^* \cdot W = N_{\text{lic}}^{\text{budget}} \cdot c_\ell$. Assuming a budget B that is equal to the core-based $N_{\text{lic}}^{\text{budget}}(c_h + c_\ell)$, we are then left with a hardware budget of $N_{\text{lic}}^{\text{budget}} \cdot c_h$, giving:

$$N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}. \tag{3.13}$$

The two models therefore purchase the same number of cores per year as a consequence of equal budgets and equal vendor revenue.

With cores retained until end of life, the fleet reaches a steady state of L equally sized cohorts, one of which retires and is replaced each year. Its size is a fixed multiple of the core-based fleet,

$$P_{\text{work}}^{\text{budget}} = L \cdot N_{\text{work}}^{\text{budget}} = L \cdot N_{\text{lic}}^{\text{budget}} \tag{3.14}$$

Section 4.5.2 sweeps $N_{\text{work}}^{\text{budget}}$ independently of L, to determine what occurs if the cores purchased per year are reduced.

In all cases, annual cost is constant,

$$M_{\text{work}}^{\text{budget}} = d\,W + N_{\text{work}}^{\text{budget}}\,c_h, \tag{3.15}$$

while completion time shrinks over time as newer, faster cohorts phase in and raise the fleet's cohort-averaged throughput:

$$\bar{y}(t) = \sum_{a=0}^{L-1} N_{\text{work}}^{\text{budget}} \cdot y_0 \, f^{\,t-a}, \tag{3.16}$$

$$T_{\text{work}}^{\text{budget}}(t) = \frac{W}{\bar{y}(t)}. \tag{3.17}$$

A cohort purchased in year $t - a$ keeps that year's performance for its whole life, so at $t < L$ the sum indexes pre-baseline generations. The fleet is assumed already in rolling steady state at $t = 0$ rather than starting from scratch. Section 4.4.1 notes the consequence for year-0 comparisons.

Work-based under the fixed-deadline scenario: The purchase count $N_{\text{work}}^{\text{deadline}}(t)$ is no longer constant under this scenario. Each year the oldest cohort, purchased in year $t - L$, retires, and the replacement must be sized so that the surviving $L - 1$ cohorts along with the new one meet the deadline. The throughput of the surviving cohorts is:

$$\bar{y}_{\text{surv}}(t) = \sum_{a=1}^{L-1} N_{\text{work}}^{\text{deadline}}(t-a) \, y_0 f^{\,t-a} \tag{3.18}$$

The deadline condition in year t determines the constraint on purchases:

$$W/H = \bar{y}_{\text{surv}}(t) + N_{\text{work}}^{\text{deadline}}(t) \, y_0 f^{\,t}, \tag{3.19}$$

which rearranges to an explicit recursion in the purchase history:

$$N_{\text{work}}^{\text{deadline}}(t) = \frac{1}{y_0 f^{\,t}} \left[\frac{W}{H} - \sum_{a=1}^{L-1} N(t-a) \, y_0 f^{\,t-a} \right]. \tag{3.20}$$

Completion time is kept equivalent to deadline H. Since each $N_{\text{work}}^{\text{deadline}}(t)$ depends on the previous $L - 1$ purchases, we evaluate Eq. (3.20) numerically, year by year, from the steady-state starting fleet. Sections 4.3.2 and 4.3.3 apply it under two different replacement schedules.

3.4 Breakeven and Time Advantage

Eq. (3.13) set d^* by paying the EDA company the same amount under both models in a fixed-budget case:

$$\boxed{d^* = \frac{N_{\text{lic}}^{\text{budget}} \cdot c_\ell}{W}} \tag{3.21}$$

Under a fixed budget this is also the rate at which the customer pays the same, since work-based spends $N_{\text{lic}}^{\text{budget}} c_\ell$ on the work rate and $N_{\text{lic}}^{\text{budget}} c_h$ on hardware, which together are exactly $M_{\text{lic}}^{\text{budget}}$. With the budget fully spent either way, equal vendor revenue and equal customer cost turn out to be the same condition.

Any $d < d^*$ gives the customer lower cost paid to the EDA company, v.s. the core-based, and any $d > d^*$ gives higher cost paid to the EDA company. This breakeven is time-independent if both models are in their fixed-budget case. Neither $M_{\text{lic}}^{\text{budget}}$ nor $M_{\text{work}}^{\text{budget}}$ depends on t, so their equality does not either. Under the fixed-deadline scenario both models' costs shrink with t, so a breakeven there would vary with t.

In steady-state, the work-based fleet holds one cohort from each of the last L years, so the ratio of completion times under the fixed-budget scenario is constant in t:

$$\frac{T_{\text{lic}}^{\text{budget}}(t)}{T_{\text{work}}^{\text{budget}}(t)} = \frac{N_{\text{work}}^{\text{budget}}}{N_{\text{lic}}^{\text{budget}}} \sum_{a=0}^{L-1} f^{-a}. \tag{3.22}$$

The work-based model completes the same workload faster because no per-core license fee restricts how many cores may run simultaneously. In the budget-matched case $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$, the ratio reaches its ceiling of L only when $f = 1$. As performance growth rises, the fleet's older cohorts fall further behind the current generation that core-based always runs, minimizing the advantage. This time advantage holds for all values of d, since d does not appear in either time equation. The EDA vendor retains full control over pricing independent of the customer's time benefit. As a note, Eq. (3.22) is specific to the fixed-budget scenario. Under a fixed deadline both

models finish at exactly H, so no time advantage exists there and the comparison is purely in terms of cost and carbon.

3.5 Carbon Model

The carbon model takes P_{lic} and P_{work} as inputs from whichever case is under analysis. Both baseline parameters from Section 2.4 evolve annually. Embodied carbon per core decreases as die efficiency increases, and operational carbon intensity changes with per-core power draw.

$$C_{\text{emb}}(t+1) = C_{\text{emb}}(t) \cdot s \tag{3.23}$$

$$\text{CI}(t+1) = \text{CI}(t) \cdot i. \tag{3.24}$$

3.5.1 Core-Based Annual Carbon

All P_{lic} cores share the current generation's performance and energy intensity each year and are assumed idle once workload W completes. We treat idle time as *not* accruing additional operational carbon. Spare capacity on owned hardware is assumed to be repurposed for other engineering work rather than left drawing power, so the carbon cost attributable to the licensing choice itself is the active-compute difference between models, not the full clock-time difference. Section 4.5.5 tests this assumption directly and reports how much the result changes if idle capacity is instead treated as wasted. Under the current framing, operational carbon is accounted for only for the time actually spent computing:

$$C_{\text{lic}}^{\text{emb}}(t) = N_{\text{lic}}(t) \cdot C_{\text{emb},0} \cdot s^t \tag{3.25}$$

$$C_{\text{lic}}^{\text{op}}(t) = T_{\text{lic}}(t) \cdot \text{CI}_0 \cdot i^t \cdot P_{\text{lic}}(t). \tag{3.26}$$

3.5.2 Work-Based Annual Carbon

The same idle-time treatment applies symmetrically here. Work-based cores also accrue no additional operational carbon while idle. Each cohort of age a in year

t keeps the performance, intensity, and embodied carbon of its own manufacture year $t-a$ until replaced. The fleet's aggregate throughput, completion time, and annual carbon follow by summing each cohort's contribution. Only the cores purchased that year contribute to the embodied carbon:

$$C^{\text{emb}}_{\text{work}}(t) = N_{\text{work}}(t) \cdot C_{\text{emb},0} \cdot s^t \tag{3.27}$$

$$C^{\text{op}}_{\text{work}}(t) = T_{\text{work}}(t) \cdot \sum_{a=0}^{L-1} N_{\text{work}}(t-a) \cdot \text{CI}_0 \, i^{\,t-a}. \tag{3.28}$$

Equations (3.27)-(3.28) hold for any replacement schedule. The schedule enters only through $N_{\text{work}}(t)$. In the fixed-budget case $N_{\text{work}}(t) = N^{\text{budget}}_{\text{work}}$ for every t, while fixed-deadline scenarios set $N_{\text{work}}(t)$ from their own replacement rules. When $N^{\text{budget}}_{\text{work}} = N^{\text{budget}}_{\text{lic}}$, Eq. (3.25) and Eq. (3.27) are equal. Both models replace the same number of cores per year at the same embodied-carbon cost.

This equality holds independent of the absolute value of $C_{\text{emb},0}$, and is therefore insensitive to embodied-carbon estimation uncertainty (Section 5).

3.5.3 Cumulative Carbon and the Savings Metric

Cumulative carbon for each model over Y years is $\mathcal{C}(Y) = \sum_{t=0}^{Y-1}[C^{\text{emb}}(t) + C^{\text{op}}(t)]$. The percentage carbon savings of the work-based model relative to core-based is:

$$\Delta C(Y) = \frac{\mathcal{C}_{\text{lic}}(Y) - \mathcal{C}_{\text{work}}(Y)}{\mathcal{C}_{\text{lic}}(Y)} \times 100\%, \tag{3.29}$$

This value is positive when work-based produces less cumulative carbon. Because the work-based fleet averages across L cohort generations rather than always running the newest generation, its operational carbon intensity carries a cohort-generation factor $\Phi(i,t)$ relative to core-based:

$$\overline{\text{CI}}_{\text{work}}(t) = \text{CI}_0 \cdot i^{\,t} \cdot \underbrace{\frac{\sum_{a=0}^{L-1} N_{\text{work}}(t-a)\, i^{-a}}{\sum_{a=0}^{L-1} N_{\text{work}}(t-a)}}_{\Phi(i,t)}, \tag{3.30}$$

$\Phi(i,t)$ is the cohort-size-weighted mean of i^{-a} across the fleet's L generations, so operational carbon can be written $C^{\mathrm{op}}_{\mathrm{work}}(t) = T_{\mathrm{work}}(t)\, P_{\mathrm{work}}(t)\, \mathrm{CI}_0 i^t\, \Phi(i,t)$. This is identical to the core-based expression of Eq. (3.26) apart from the factor Φ. The two models therefore differ in operational intensity by exactly Φ.

The direction of Φ is set entirely by whether per-core power is rising or falling, and holds for any cohort sizes since every weight is positive. When $i > 1$, older cohorts are less power-intensive than the current generation, every $i^{-a} < 1$, and $\Phi < 1$: cohort averaging is a genuine discount. When $i < 1$, which is the trend currently demonstrated by both vendors in Section 2.3, every $i^{-a} > 1$ and $\Phi > 1$, so cohort averaging becomes a penalty. When N_{work} is constant the weights are equal and Φ reduces to $\frac{1}{L}\sum_{a=0}^{L-1} i^{-a}$, independent of t. At this report's baseline $L = 7$ this evaluates to $\Phi = 1.25$ for Intel and $\Phi = 1.91$ for AMD for constant N_{work}.

Chapter 4: Results

All results below use the baseline parameter set in Table 4.1 unless stated otherwise. L, c_l and c_h are based on values discussed in Section II. Meanwhile values for B, H, W, and y_0 are examples, so they may be orders of magnitude off in reality. The numbers are meant to be replaced for each individual company to make their own comparisons.

Plugging these values into Eq. (3.4) and Eq. (3.5) gives the core-based fixed-budget fleet,

$$P_{\text{lic}}^{\text{budget}} = \left\lfloor \frac{10^6}{100 + 10{,}000} \right\rfloor = 99 \text{ cores}, \tag{4.1}$$

at an annual cost of $M_{\text{lic}}^{\text{budget}} = 99 \cdot \$10{,}100 = \$999{,}900$.

For the work-based scenario, we assume that $P_{\text{lic}}^{\text{budget}} = N_{\text{work}}^{\text{budget}}$. From this, we can derive $P_{\text{work}}^{\text{budget}} = L \cdot N_{\text{work}}^{\text{budget}} = 693$ as per Eq. (3.14). This gives seven cohorts of 99 cores each.

The breakeven rate of Eq. (3.21) evaluates to

$$d^* = \frac{99 \cdot \$10{,}000}{10^{12}} \approx 9.90 \times 10^{-7} \text{ \$/work unit}, \tag{4.2}$$

and the time ratio of Eq. (3.22) is 5.59× for Intel.

4.1 Per-Core Licensing's Refresh Incentive

Under per-core licensing it is cheaper to keep buying newer and faster cores yearly than to keep older, slower ones.

We prove this by comparing the relative cost of keeping the cores already owned against replacing them with the current generation, over a single year. The license-to-core ratio is LC = c_ℓ/c_h. Since a license costs the same regardless of core

Symbol	Quantity	Value
B	Annual budget	\$1,000,000
c_ℓ	Lic cost per core-year	\$10,000
c_h	Hardware cost per core	\$100
W	Annual work	10^{12} work units
H	Annual deadline	1,500 hr
y_0	Baseline per-core perf	6.734×10^6 work units/hr
L	Core lifetime	7 yr
Y	Horizon	50 yr

Table 4.1: Baseline parameter set. y_0 is fixed by the requirement that the year-0 deadline and budget bounds of Eq. (3.3) coincide, i.e. $y_0 = W/(H \cdot P_{\text{lic}}^{\text{budget}})$. Cores are treated as continuous rather than integer in all sweeps.

speed, the refreshing customer needs fewer cores to do the same work, and therefore buys fewer licenses.

For a logical walkthrough, we can consider a customer deciding, at the start of a year, whether to keep cores of performance y or replace them with the current generation at fy. Completing the same work needs N old cores or N/f new ones. The old cores are already owned, so they only account for license cost for the year. The new cores account for the cost of purchasing both a core and a license. Replacing is cheaper when

$$\frac{N}{f}\,(c_\ell + c_h) < N\,c_\ell, \tag{4.3}$$

which reduces to $c_h < c_\ell(f-1)$, or

$$\boxed{\mathrm{LC}^* = \frac{1}{f-1}} \tag{4.4}$$

Refresh is cheaper whenever $\mathrm{LC} > \mathrm{LC}^*$. The fleet size N cancels, so the threshold depends on performance growth alone. As $f \to 1$, it diverges. With no performance growth, no ratio justifies discarding working hardware.

The same threshold governs the time comparison. Under a fixed deadline the two strategies both finish at H and differ only in spend. Under a fixed budget they spend equally and differ only in completion time. Either way the quantity being

traded is the factor f against the added hardware cost, so above LC^* annual refresh is simultaneously cheaper and faster.

Equation (4.4) is the conservative form of this test, since it compares against hardware only one generation old. A customer weighing refresh against holding cores for their full lifetime L faces a larger performance gap and therefore a lower threshold.

4.2 Refresh Strategy Under Work-Based Pricing

Work-based pricing removes the license term from the replacement decision, so a customer's refresh strategy is driven only by hardware cost and by how efficiently a fleet of mixed generations meets deadline. Annual full refresh is ruled out: replacing every core each year buys L times as much hardware as replacing once per lifetime, for a fleet that is no larger, so it is a $7\times$ hardware penalty at $L = 7$ with no compensating benefit. Instead, the form of partial refresh depends on which bound the customer is under.

Under a **fixed budget** the fleet size is fixed by spend rather than by work, so the natural policy is to have L equally sized cohorts and replace one each year. The fleet is constant in size since every year the budget is used to replace $1/L$ of the total fleet.

Under a **fixed deadline** the required fleet shrinks every year as cores get faster, and two policies are explored. *Lifetime replacement* buys a complete fleet that meets the deadline, keeps it for L years, then replaces it in full. The fleet consists of a single generation at any moment. *Rolling replacement* instead divides the fleet into L slots each of one generation of cores, retires one slot per year, and only buys as many cores as are needed for to meet the deadline.

The two fixed-deadline policies do not cost the same. The rolling replacement has a mix of generations and therefore needs more cores in total to deliver the same throughput. Rolling buys fewer cores per purchase but carries a larger and older fleet

between purchases, which trades hardware spend against carbon in a way Section 4.4 quantifies.

4.3 Case Comparisons

The three cases below compare the two licensing models on *cost and time only*. Each case fixes one of the two bounds of Eq. (3.3) and lets the other float, so exactly one of the two quantities is free to differ between models. Under a fixed budget the models differ in completion time, and under a fixed deadline they differ in spend. Carbon is deferred entirely to Section 4.4, which applies the same three cases to the emissions model.

4.3.1 Fixed Budget

Under a fixed annual budget, a customer spends as much as the budget allows on cores, for the sake of reducing completion time as much as possible. Core-based therefore has $P_{\text{lic}}^{\text{budget}} = 99$ and refreshes annually. Work-based holds $P_{\text{work}}^{\text{budget}} = 693$ in seven cohorts and uses rolling refresh, replacing 99 cores per year. Because spend is fixed by construction, **completion time is the quantity that differs**, and cost enters only through the vendor's choice of d. The license cost is fixed as per Table 4.1.

Table 4.2 shows why the two columns diverge. Core-based cannot grow its fleet without buying licenses, so its only lever is core speed and its completion time falls by exactly f each year. Work-based can deploy seven times the cores for the same cost, and its time falls at the cohort-averaged rate instead. Both fall in step, so the ratio between them is constant at 5.6× for Intel, which is the value Eq. (3.22) gives at $f = 1.082$.

That ratio is bounded above by the fleet-size multiplier and eroded by performance growth. Figure 4.1 plots it across the plausible range of f. It reaches its ceiling of $L = 7$ only where cores stop improving. With every generation equally fast,

	Core-based		Work-based	
t	**Cores**	T **(hr)**	**Cores**	T **(hr)**
0	99	1,500	693	268
1	99	1,386	693	248
7	99	864	693	154
14	99	498	693	89
25	99	209	693	37
49	99	32	693	6

Table 4.2: Case 1 walkthrough, Intel parameters, at sampled years. Both fleets are constant in size because the budget bound fixes them, and both replace 99 cores per year. Annual spend is \$999,900 for core-based and $d \cdot W +$ \$9,900 for work-based in every year.

f **(perf. improvement/gen)**	**Time ratio**
0%	7.00×
5%	6.08×
10%	5.36×
15%	4.78×
20%	4.33×

Table 4.3: Case 1 - Core-based/work-based completion-time ratio by performance-improvement rate, from Eq. (3.22) at $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$. The ratio is constant in t when both models start at steady state.

the ratio collapses to the fleet-size ratio, and the work-based fleet is L cohorts deep against core-based's one. It declines from there because core-based captures each new generation immediately while the rolling fleet averages across L vintages, and that gap widens as f rises. Both measured vendors sit well inside the range, at 5.59× (Intel) and 4.74× (AMD), and even a hypothetical 20%/generation improvement leaves work-based finishing more than four times sooner (Table 4.3). The time advantage is therefore structural rather than marginal, given it survives across the whole span of performance growth considered.

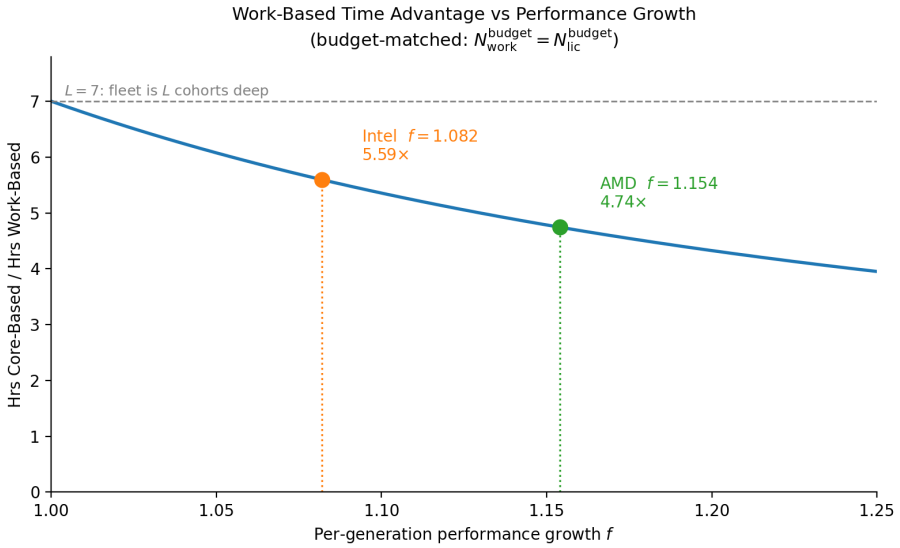

Figure 4.1: Case 1 work-based time advantage as a function of per-generation performance growth f (Eq. (3.22), $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$). The work-based fleet is L cohorts deep against core-based's one, giving a ceiling of $L = 7$. The advantage is capped when cores stop improving and decays as f rises. Both vendors remain above 4.7×.

4.3.2 Fixed Deadline, Work-Based Lifetime Replacement

In fixed-deadline cases, cost is the quantity that differs. For our cost analysis, we hold the vendor's revenue equal between models. This is the same condition that defines d^* in Eq. (3.21). There, $d^*W = P_{\text{lic}}c_\ell$ is exactly core-based's annual license revenue. The condition is unchanged here, but the core-based fleet now shrinks with t, so the rate that satisfies it does too:

$$d^*(t) \cdot W = P_{\text{lic}}^{\text{deadline}}(t) \cdot c_\ell. \tag{4.5}$$

The licensing term is therefore identical on both sides in every year and cancels, leaving hardware as the only quantity that differs. We hold revenue equal because it removes the vendor's pricing from the comparison: whatever the customer gains under Eq. (4.5) does not come out of the vendor's revenue, so it must come from the licensing structure itself.

	Core-based		Work-based		
t	**Cores**	**Cum. \$**	**Cores**	**Bought**	**Cum. \$**
0	99	9,900	99	99	9,900
1	92	19,100	99	0	9,900
7	58	61,500	58	58	15,700
14	33	91,300	33	33	19,000
25	14	115,000	19	0	20,900
49	3	130,400	3	3	23,400

Table 4.4: Case 2 walkthrough, Intel parameters, at sampled years. Both models complete at exactly $H = 1{,}500$ hr and pay the vendor the same amount (Eq. (4.5)), so the cumulative columns are hardware spend alone and carry the entire cost comparison.

Under a fixed deadline, completion time is fixed at H for both models. With vendor revenue held equal by Eq. (4.5), **hardware is the only quantity that differs**. Core-based resizes its fleet annually via Eq. (3.8), shrinking as cores get faster, and buys that fleet outright every year. Work-based instead replaces its entire fleet once per lifetime, sizing each purchase to just clear the deadline at that year's performance and holding it fixed until the cores are fully replaced at lifetime. Its fleet therefore consists of a single generation at any moment. This is the lifetime-replacement strategy of Section 4.2, and it is the natural choice for a customer whose licensing does not penalize retaining hardware.

Table 4.4 makes the asymmetry concrete. The two models are identical in year 0 and diverge immediately afterward. Core-based purchases its entire fleet again in year 1, while work-based purchases nothing until year 7. The consequence is visible in the *Cores* columns at $t = 25$, where work-based still runs the fleet it bought at $t = 21$ against core-based's current-generation one. Work-based is therefore oversized relative to core-based for six years out of seven, while manufacturing cores at roughly one-seventh the rate. There are only 234 cores over the horizon against core-based's 1,304, for a 5.6× difference in both hardware spend and manufacturing volume.

Because the fleet's age cycles over L years, the reported figures depend on

	Core-based		Work-based		
t	**Cores**	**Cum. $**	**Cores**	**Bought**	**Cum. $**
0	99	9,900	123.6	14.1	1,414
1	92	19,100	114.7	12.1	2,622
7	58	61,500	71.2	8.1	8,534
14	33	91,300	41.0	4.7	12,635
25	14	115,000	18.2	1.4	15,737
49	3	130,400	2.6	0.3	17,852

Table 4.5: Case 3 walkthrough, Intel parameters, at sampled years. Both models complete at exactly $H = 1{,}500$ hr and pay the vendor the same amount. Core-based is identical to Case 2, since its policy does not depend on what work-based does. Cores are treated as continuous (Table 4.1).

where that cycle sits at $t = 0$. We align it so that year 0 is a replacement year, which is the most favorable phase for work-based.

4.3.3 Fixed Deadline, Rolling Work-Based Replacement

Case 2 replaces the entire fleet at once in a work-based model, which means the fleet is at its most oversized just before each replacement. The alternative, Case 3, retires one cohort slot per year and sizes the replacement to just clear the deadline given the surviving $L - 1$ cohorts. This is calculated by solving Eq. (3.20) at each t, and buys a variable number of cores each year rather than a fixed batch every seven years. Vendor revenue is held equal as in Case 2, so hardware again determines the comparison.

Table 4.5 shows the two properties that distinguish this policy from Case 2. Purchases fall steadily rather than arriving in seven-year batches, which makes cumulative hardware spend the lower of the two ($17,852 against $23,400, or 178 cores manufactured against 234). However, the work-based fleet sits roughly 25% above core-based at every sampled year, since a mix of L generations needs more cores than a single current-generation fleet to deliver the same throughput. Rolling replacement therefore *buys* less and *keeps* more. This distinction is significant in

Case	Vendor	Core-based	Work-based	Ratio
Fixed budget: total spend equal; time differs				
1	Intel	19,408 hr	3,469 hr	5.59×
1	AMD	11,232 hr	2,367 hr	4.74×
Fixed deadline: time and vendor revenue equal; hardware differs				
2	Intel	$130,400	$23,400	5.57×
2	AMD	$76,900	$16,000	4.81×
3	Intel	$130,400	$17,852	7.30×
3	AMD	$76,900	$9,887	7.78×

Table 4.6: Fifty-year totals by case. Case 1 reports cumulative completion time, since spend is fixed at $999,900/yr for both models. Cases 2 and 3 report cumulative hardware spend, since time is fixed at H and vendor revenue is equalized by Eq. (4.5). The hardware ratio equals the ratio of cores manufactured, at $100 per core.

Section 4.4 once emissions are calculated.

The policy meets the deadline exactly in every year, though it does not settle into an evenly shrinking sequence of purchases. It reaches a limit cycle in which one cohort is roughly twice the size of the others and rotates with period L: a large purchase in a year when the surviving cohorts are weak allows the next $L-1$ purchases to be small, and when that large cohort retires another large purchase is required.

4.3.4 Cost Across the Three Cases

Table 4.6 collects the 50-year totals. The two bounds produce qualitatively different comparisons, because each fixes a different quantity. Under a fixed budget both models spend the same by construction, so the difference appears as completion time, and work-based finishes 5.6× sooner for Intel. Under a fixed deadline both models finish at the same moment and pay the vendor the same amount, so the difference appears as hardware, and work-based manufactures five to eight times fewer cores.

First, the advantage is of the same order in every case, between roughly 4.7× and 7.8×, even though the quantity being measured changes between them. This

Case	Intel	AMD
1 (fixed budget)	−50.2%	−164.0%
2 (deadline, lifetime)	−39.3%	−105.2%
3 (deadline, rolling)	−48.6%	−187.7%

Table 4.7: Carbon savings $\Delta C(50)$ of work-based relative to core-based (Eq. (3.29)), all three cases, real vendor parameters. Negative values indicate work-based produces *more* cumulative carbon.

occurs since there is no longer a cost constraint on what is paid to the EDA vendor based on how many cores are run. Second, among the two fixed-deadline policies, rolling replacement is the more hardware-efficient, since it buys only what each year's deadline requires rather than a full fleet every seventh year.

4.4 Carbon Results for the Cases

Sections 4.1 through 4.3.4 establish that work-based licensing removes the refresh incentive without sacrificing the cost/time benchmark. This section explores whether removing that incentive also reduces carbon output. Rather than having a straightforward answer, many parameters and sensitivities influence which licensing model outputs less carbon under different conditions. As a note, the sensitivity analyses in this section are computed on Case 1, which isolates the licensing mechanism from the refresh-schedule differences that distinguish Cases 2 and 3.

4.4.1 Cumulative Carbon Emissions

Using each vendor's own measured per-core values (Sections 2.2, 2.3, and 2.4: $i_{\text{Intel}} = 0.932$, $f_{\text{Intel}} = 1.082$; $i_{\text{AMD}} = 0.826$, $f_{\text{AMD}} = 1.154$) in the model of Section 3, Table 4.7 reports $\Delta C(50)$ for all three cases.

As the primary result on real-core data, **core-based licensing is favored on carbon in every case and for both vendors** (Figure 4.2). This is the reports's central empirical finding, previewed in Section 1. This occurs since real per-core

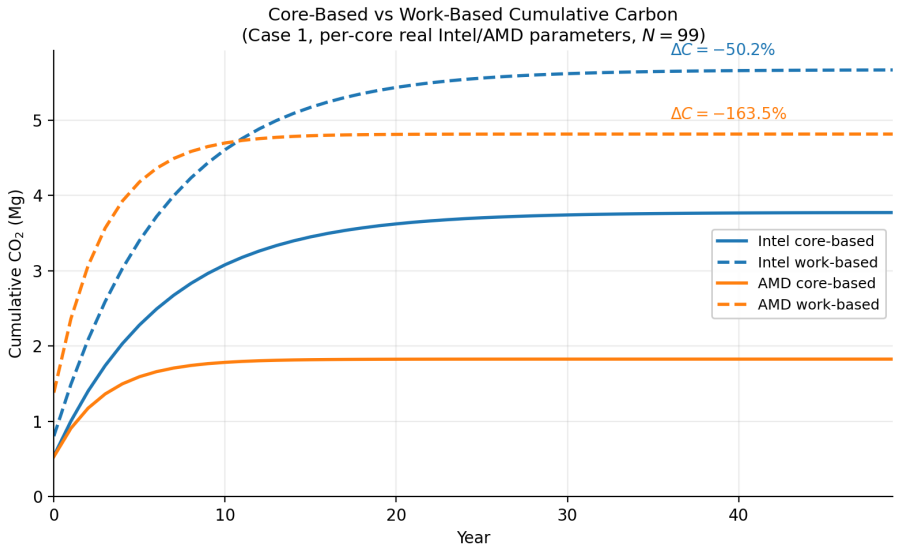

Figure 4.2: Core-based vs. work-based cumulative carbon, real per-core Intel and AMD parameters, Case 1, $N = 99$, 50 years.

efficiency gains are large enough compared to the embodied costs that constantly refreshing to the newest generation captures more efficiency in contrast to a work-based fleet's rolling average of older cohorts. In Cases 1 and 3 the work-based fleet holds L cohort generations at once, so the penalty is the $\Phi(i,t) > 1$ cohort-averaging factor of Eq. (3.30). Case 2's fleet is always a single generation, so no cohort averaging occurs. Its penalty is instead that the fleet is frozen at one generation for L years, falling progressively further behind the current generation before recovering in a single step. The carbon direction is the same in both.

As seen above, the result is robust to scenario. Core-based is favored in every case and for both vendors, so it does not depend on which customer profile or refresh policy a reader finds most realistic. This finding may change depending on which parameters a customer plugs in now, or in the future. Second, and more useful to a customer, cost and carbon disagree about *which* work-based policy to adopt. Section 4.3.4 found rolling replacement the more hardware-efficient of the two fixed-deadline policies (7.30× against 5.57× for Intel). On carbon the ordering

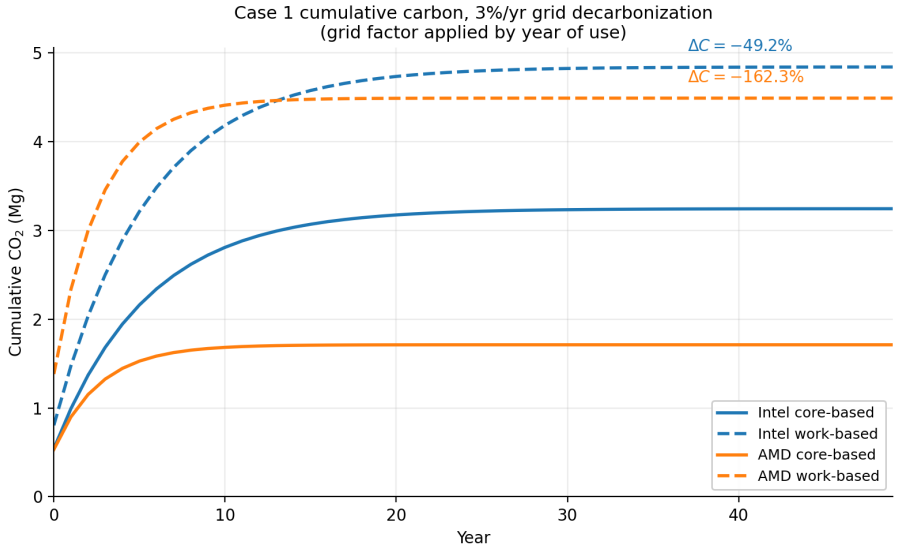

Figure 4.3: Case 1 cumulative carbon with 3%/yr grid decarbonization, real per-core Intel and AMD parameters, $N = 99$. Compared to Fig. 4.2, the curves fall but their ratio barely moves.

reverses: lifetime replacement is the better of the two (-39.3% against -48.6% for Intel, -105.2% against -187.7% for AMD). Rolling replacement *buys* less but *keeps* more, and it is what a fleet holds, not what it buys, that draws power each year. A customer minimizing hardware spend and a customer minimizing emissions would therefore choose different replacement schedules under the same licensing model.

Note that because the work-based fleet is modeled as already being in rolling steady state at $t = 0$ (Section 3.3), its cohorts carry pre-baseline generations in the first L years and the two models do *not* start from equal carbon: $\Delta C(0) = -48.4\%$ for Intel and -159.3% for AMD.

Figure 4.3 repeats the Case 1 comparison with a 3%/yr grid decarbonization rate. Grid carbon intensity is a property of the electricity supply in a given year rather than of the hardware drawing it, so unlike the per-core power factor i it is applied by year of use rather than year of manufacture. The effect on the comparison is therefore small, moving $\Delta C(50)$ from -50.2% to -49.2% for Intel and from

Decarb. rate	Grid at yr 50	$\Delta C(50)$	Embodied share (core-based)
0%/yr	100%	−50.2%	10.5%
3%/yr	21.8%	−49.2%	12.2%
10%/yr	0.5%	−47.1%	16.0%
20%/yr	0.0%	−44.4%	20.9%
50%/yr	0.0%	−37.8%	32.6%

Table 4.8: Effect of uniform grid decarbonization on Case 1 carbon savings, Intel parameters. "Grid at yr 50" is the remaining carbon intensity relative to year 0. As the grid cleans, embodied carbon becomes a larger share of the total and ΔC drifts toward the embodied-only floor, where the two models are equal i.e. $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$

−164.0% to −162.3% for AMD. A cleaner grid scales both models' operational carbon by the same factor each year, shrinking the term they are competing over without changing the i-versus-f ratio that decides the winner, so ΔC drifts slightly toward the embodied-only floor at which the two models are equal. This is the same convergence Section 4.5.1 reports in the limit $\text{CI}_0 \to 0$.

Table 4.8 shows this drift is monotone and bounded. Even a grid decarbonizing at 50%/yr, effectively reaching zero intensity within a decade, moves Intel only from −50.2% to −37.8%. The cumulative total is dominated by early years, when the grid was still dirty. No decarbonization rate reverses the sign. This could change in future investigations, however, once the year-0 grid intensity is already low.

Embodied carbon accounts for roughly 10% of cumulative emissions for both vendors at this operating point (9.8% AMD, 10.5% Intel; Figure 4.6). Though significant, embodied carbon does not drive the sign of the result. Operational carbon dominates, consistent with Eq. (3.25)/Eq. (3.27)'s equality when $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$ (Section 3.5.2).

4.4.2 Embodied vs. Operational Carbon

The headline figures of Table 4.7 are net totals, and they conceal that the two components move in opposite directions. Table 4.9 separates them with the following key observations.

Case	Vendor	Embodied	Operational
1 (fixed budget)	Intel	0.0%	−56.1%
	AMD	0.0%	−181.4%
2 (fixed deadline)	Intel	+71.4%	−49.1%
	AMD	+54.8%	−119.3%
3 (fixed deadline)	Intel	+86.6%	−60.6%
	AMD	+87.2%	−211.9%

Table 4.9: $\Delta C(50)$ decomposed by component. Positive values favor work-based. Operational carbon accounts for 89-92% of the core-based total. The embodied column is also the result under a fully decarbonized grid, since operational carbon vanishes in that limit.

Embodied carbon favors work-based whenever the deadline binds: Under a fixed budget the two models manufacture the same 99 cores per year by construction, since $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$, so their embodied totals are identical and $\Delta C_{\text{emb}} = 0$. Under a fixed deadline that symmetry breaks. Core-based buys a complete fleet every year while work-based buys one every seventh year (Case 2) or a shrinking slice each year (Case 3). Over the horizon core-based manufactures 0.301 Mg CO_2 of embodied carbon against work-based's 0.086 Mg in Case 2 and 0.040 Mg in Case 3, a 71% and 87% reduction in embodied carbon from the core-based model.

Operational carbon favors core-based in every case: By 49% to 212%, for the reasons of Section 4.4.1: a fleet of older cohorts draws more power per unit of work than one that is always current.

Operational dominates, so it decides: It is 89–92% of the core-based total in every case. Though the embodied advantage is large in percentage terms, it accounts for a much smaller share of the total carbon emissions.

This decomposition also sharpens the limit examined in Section 4.4.1. As the grid decarbonizes, operational carbon shrinks toward zero and the net result converges on the embodied column. In the fixed-budget case that floor is zero, so the two models converge to parity. In the fixed-deadline cases the floor is *not* zero: a fully decarbonized grid would leave work-based ahead by 71% (Case 2) and 87% (Case

3). Grid decarbonization can therefore reverse this report's finding, but only for a deadline-bound customer, and only in a limit no current grid approaches. Table 4.8 shows that even 50%/yr decarbonization leaves the sign unchanged over a 50-year horizon. It is a different mechanism from the architectural crossover of Section 4.5.1, which turns on i relative to f rather than on the carbon intensity of electricity.

4.5 Sensitivity Analyses

We perform further experiments and sensitivity analyses for carbon emission calculations.

4.5.1 The Crossover, When Would Work-Based Help

The sign of ΔC is determined by where i sits relative to f (Eq. (3.30)): below the crossover $i = f$, core-based wins since its advantage from always running the newest generation outweighs work-based's parallelism, and the cohort-generation factor $\Phi(i)$ works against work-based rather than for it. Above the crossover, that balance flips and the work-based approach wins instead, independent of the embodied-carbon baseline.

Figure 4.4 makes the vendor result legible against the full range: both Intel's $i = 0.93$ and AMD's $i = 0.83$ sit well to the left of the $i = f = 1.125$ crossover, in the region where core-based wins decisively. For work-based licensing to produce less carbon under current performance-growth rates, per-core power growth would need to *rise* toward or past $i = 1.125$. In other words, per-core efficiency gains would need to slow substantially from their current pace. This condition is not currently foreseen, though is noted in case it becomes a reality.

Another phenomenon is responsible for furthering this crossover point: grid decarbonization. We tested this directly by driving CI_0 toward zero in the model of Section 3.5. The total carbon for both licensing models converges toward the embodied-carbon-only floor, which is *equal* between them in the fixed-budget case at

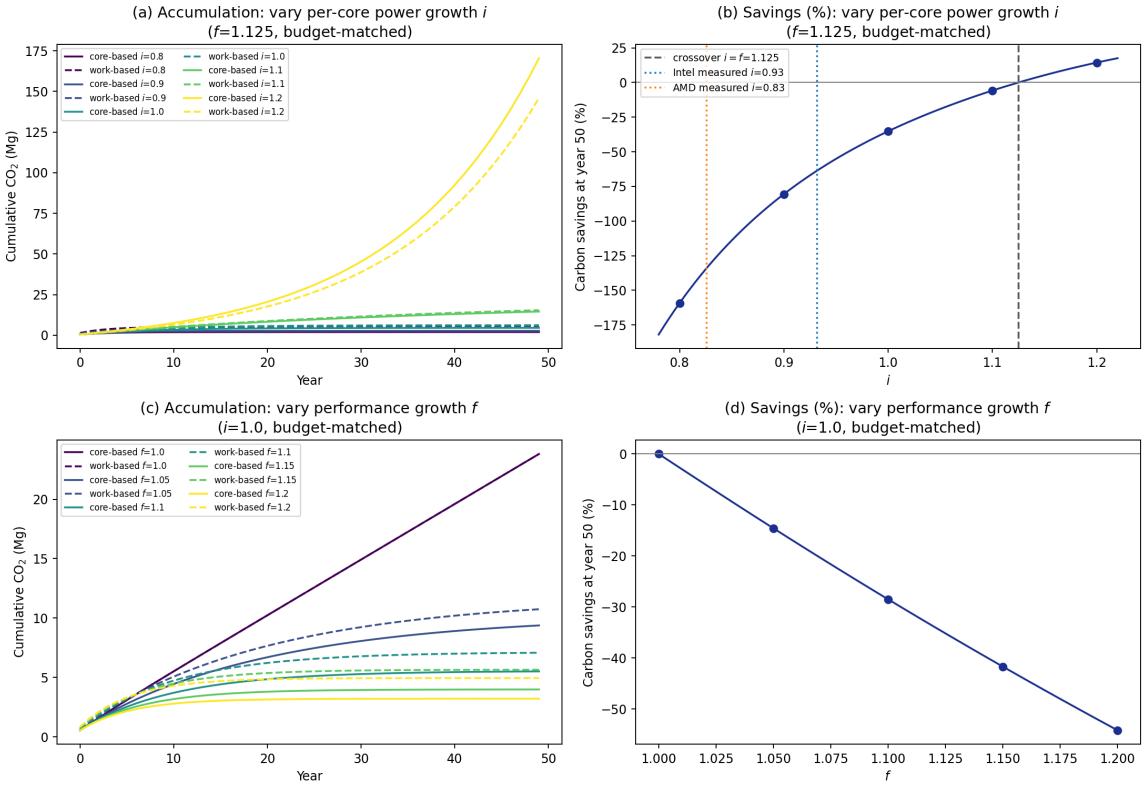

Figure 4.4: Carbon savings $\Delta C(50)$ across an abstract per-core power-growth sweep $i \in [0.8, 1.2]$ (chosen to bracket both vendors' real measured i, marked as dotted lines), $f = 1.125$ fixed, $N_{\text{work}} = N_{\text{lic}}$: (a) Cumulative accumulation; (b) Percentage savings, crossover at $i = f = 1.125$ exactly, savings range from -159.3% ($i = 0.8$) to $+14.4\%$ ($i = 1.2$), a span of 174 points; (c,d) Same exploration varying f at $i = 1.0$ fixed, savings fall monotonically as f rises, since faster core-based cores finish work faster every year.

$N_{\text{lic}}^{\text{budget}} = N_{\text{work}}^{\text{budget}}$ (Eq. (3.25) equals Eq. (3.27)), rather than creating a work-based advantage. This matches Figure 4.3: renewable adoption shrinks the operational-carbon term both models are fighting over, without changing the i-vs-f ratio that decides who wins it. Intel reports approximately 95% global renewable electricity as of end of 2025 [26], while TSMC, AMD's chip manufacturer, targets only 60% renewable by 2030 [27]. Neither figure moves either vendor across the crossover, because the crossover is an architectural question (per-core power vs. performance growth) rather than an energy-sourcing one. The condition that would move i upward is architectural. The "power wall" following the end of Dennard scaling around 2005 is precisely why the industry shifted to multi-core designs with falling per-core power in the first place (the trend this report's TDP data reflects), and continued horizontal core-count scaling faces its own thermal-density limits ("dark silicon"). Whether further scaling eventually requires rising per-core power to sustain performance growth, or whether efficiency gains continue at their current pace, is an open architectural question this report does not resolve.

Two additional checks were used against this methodology. First, Intel's f was measured over a 3-generation-transition span while i used 4 (Section 2.2). Recomputing the crossover under a matched 4-transition cadence shifts it from $i = f = 1.08$ to $i = f = 1.06$. Intel's measured $i = 0.93$ stays below either version, so the qualitative result is unaffected. Second, the i values above use 2-point endpoint CAGRs. By fitting a log-linear trend through all 5 generations in Tables 2.2/2.3, we instead get $i_{\text{Intel}} = 0.92$, $i_{\text{AMD}} = 0.85$. Both estimates move slightly but neither crosses its respective crossover. A third check, converting f and i from per-generation to per-calendar-year rates (Section 2.2), gives $f_{\text{Intel}} = 1.048$, $i_{\text{Intel}} = 0.954$, $f_{\text{AMD}} = 1.100$, $i_{\text{AMD}} = 0.880$. Both vendors remain below their crossover under this convention as well.

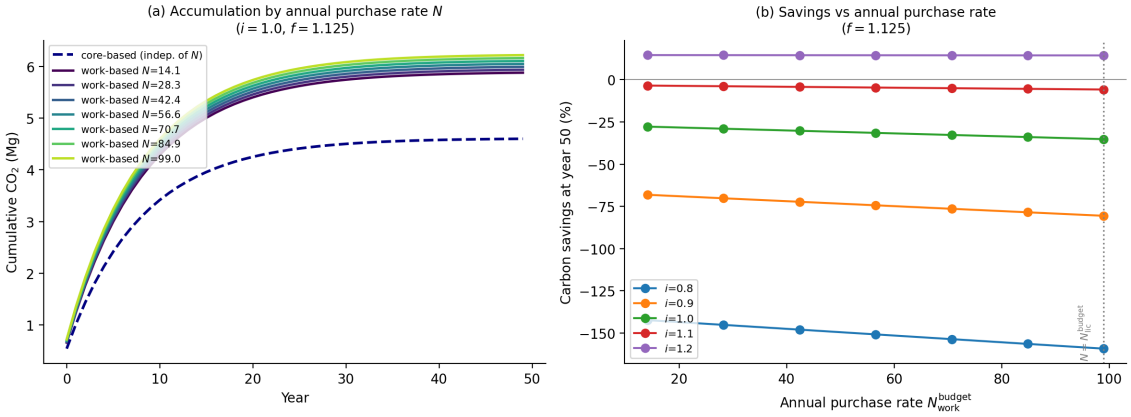

Figure 4.5: Effect of annual core purchase rate $N_{\text{work}}^{\text{budget}}$ on carbon savings across the same abstract i sweep ($f = 1.125$, year 50): (a) Core-based (dashed) is independent of N by construction; (b) The lines are near-flat: purchase rate moves ΔC by only 7-8 points across the full sweep, while i moves it by more than 174; Only $i = 1.2$, above the crossover, is positive at any purchase rate.

4.5.2 Fleet Size and Sensitivity

Figure 4.5 shows $N_{\text{work}}^{\text{budget}}$, the number of cores bought per year under the work-based scheme, is a real but secondary lever compared to i itself: at $i = 1.0$, moving from $N_{\text{work}}^{\text{budget}} = 14.1$ to 99 swings savings from -27.8% to -35.2%, but the sign of the *overall* result is set by where i sits relative to f, not by fleet-size choice.

The near-flatness is not a coincidence. Because the annual work W is fixed, halving the fleet doubles the time needed to complete it, so total core-hours are unchanged and operational carbon does not depend on how many cores the customer buys. At $i = 1$ the operational gap between the two models is therefore identical at every purchase rate, and all dependence on $N_{\text{work}}^{\text{budget}}$ falls on embodied carbon alone. Since embodied carbon is only about 9% of the total, the spread across the whole sweep is roughly 8 percentage points, against more than 170 for i.

Figure 4.6 shows this split directly at the budget-matched point. Panel (a) stacks the two components for Intel. The pale bands are embodied carbon, nearly equal between the models because both manufacture 99 cores per year, while the

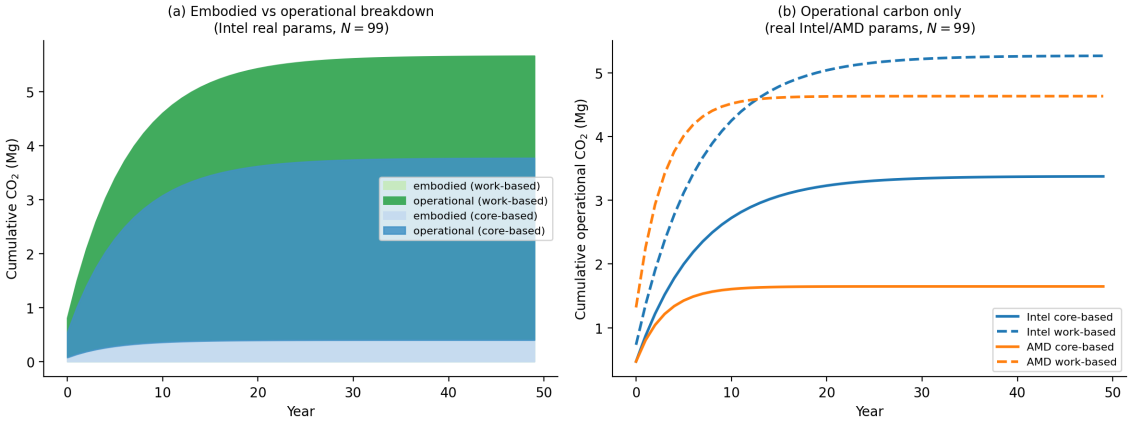

Figure 4.6: Carbon breakdown by component, real Intel and AMD parameters, Case 1, $N = 99$: (a) Stacked embodied/operational carbon, Intel; (b) The gap between core-based and work-based is almost fully attributed to operational carbon, while embodied carbon (Eq. (3.25)=Eq. (3.27) when $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$) contributes equally to both.

gap between the two totals is almost entirely the darker operational bands. Panel (b) isolates operational carbon for both vendors. Taken alone, it gives a gap slightly wider than the net ΔC of Table 4.7 (-56.1% against -50.2% for Intel), since work-based's marginally lower embodied carbon narrows the total. Embodied carbon thus works weakly in work-based's favor. The two curves also flatten by roughly year 25, since both models' annual emissions decay as s^t and i^t. The implications of this decay are further discussed in Section 5.

For a follow up, we are curious if a smaller work-based fleet can flip the sign using less embodied carbon, especially for the case where the customer is buying $N_{\text{lic}}^{\text{budget}}/L$ (Section 3.5.2). Table 4.10 answers this directly, sweeping $N_{\text{work}}^{\text{budget}}$ with each vendor's real measured $i, f, C_{\text{emb},0}, s$ rather than the abstract range.

We have not yet found such a crossover. Even at $N_{\text{work}}^{\text{budget}} = 14.1$, with a genuine $7\times$ lower manufacturing rate than core-based's 99 cores/yr, work-based still produces 41.3% more carbon for Intel and 155.5% more for AMD. The gap only widens from there as the purchase rate rises, so no fleet size in the sweep comes closer

$N_{\mathbf{work}}^{\mathbf{budget}}$	**Intel**	**AMD**
14.1	−41.3%	−155.5%
28.3	−42.8%	−156.9%
42.4	−44.3%	−158.3%
56.6	−45.8%	−159.7%
70.7	−47.3%	−161.2%
84.9	−48.8%	−162.6%
99.0	−50.2%	−164.0%

Table 4.10: Fleet-size sweep in Case 1 using real vendor parameters (not the abstract i sweep of Figure 4.5), year 50. $N_{\text{work}}^{\text{budget}}$ is simultaneously the work-based fleet's annual manufacturing rate and, via Eq. (3.14), one Lth of its fleet size. It is non-integer below the budget-matched point because cores are treated as continuous (Table 4.1).

to parity than the smallest. Operational carbon, not embodied carbon, sets the sign at every fleet size tested, even though embodied carbon produces the variation across the sweep. The real per-core efficiency gap is too large for a smaller, slower-replacing fleet to offset. This also means the smallest fleet tested is not a viable compromise for a customer. Buying $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}/L$ cores per year yields a fleet of $L \cdot N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$ cores, the same size as core-based, so it offers no parallelism advantage (time ratio 1×, Eq. (3.22)). A customer would give up the entire time benefit from Section 3.4 without recovering a carbon advantage in exchange.

4.5.3 Core-Limited Workloads

Some EDA tools, like Synopsys PrimeTime, are memory-bound signoff tools. They cannot use every core on a socket regardless of how many exist. Synopsys's own documentation states traditional STA workloads have used 8 or 16 CPU cores [28], versus 96+ cores on modern server hardware, and names 16-core scalability as a specific PrimeTime feature [29]. For this specific workload class, the socket (not the core) is the physical unit that determines embodied carbon since the whole die is manufactured regardless of active core count. Operational power follows an idle+active decomposition rather than scaling linearly with core count. Using a 20%

	Full-utilization (primary)	Core-limited (16c)
Intel	−50.2%	−13.0%
AMD	−164.0%	−13.0%

Table 4.11: Carbon savings: full-utilization per-core primary result vs. core-limited (16-core) sensitivity check, Case 1, at the budget-matched baseline $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$, year 50.

idle-power fraction and the same TDP data as Section 2.3, per-job power growth is a modest +0.97%/gen (Intel) and +5.41%/gen (AMD). This near-cancellation is not a coincidence: total socket TDP is rising across generations (pushing the job's power up), while the fixed 16-core job claims a shrinking fraction of an ever-growing socket (28→64 cores for Intel, 32→192 for AMD, Tables 2.2–2.3). This pushes the job's power back down by nearly the same amount. The two effects largely offset, leaving the power attributable to a fixed-size job nearly unchanged generation to generation, although socket-level and full-utilization per-core accounting result in vastly different conclusions.

Table 4.11 and Figure 4.7 show the core-limited case reaches the same conclusion as the primary full-utilization result at the budget-matched fleet (core-based favored), but by a much smaller margin, and only there. Below roughly $N_{\text{work}}^{\text{budget}} = 60$–70 the sign reverses and work-based produces less carbon. The two vendors converge to within a percentage point of each other for memory-bound, core-limited workloads specifically. Vendor choice matters less than fleet-size and idle-power effects. This is a different, narrower workload class from this report's primary full-utilization assumption, rather than a competing accounting scope for the same workload. We report it as a secondary result for readers whose EDA tools fit this profile.

4.5.4 Socket-Level Accounting

Since total socket TDP *rises* across generations (Section 2.3) while per-core TDP *falls*, aggregating at the socket level instead of the core level flips the sign of this result. Using socket-level growth rates ($i_{\text{Intel}} = 1.15$, $i_{\text{AMD}} = 1.31$) gives work-

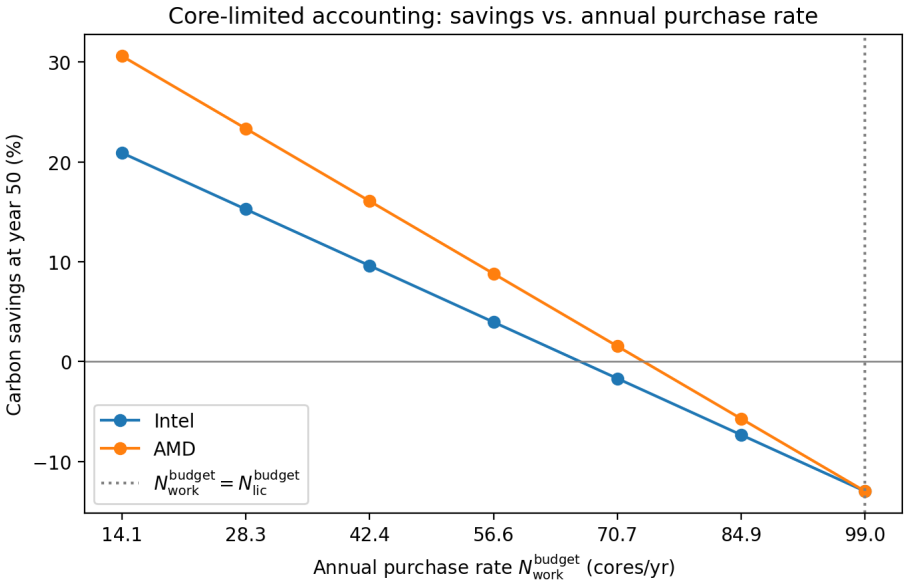

Figure 4.7: Carbon savings vs. annual purchase rate $N_{\text{work}}^{\text{budget}}$ under core-limited (16-core) accounting, swept over the same seven points as Table 4.10. Both vendors show savings at the smallest fleet (+20.9% Intel, +30.6% AMD at $N_{\text{work}}^{\text{budget}} = 14.1$), crossing into net additional carbon above roughly $N_{\text{work}}^{\text{budget}} = 60$–$70$, and reaching −13.0% at the budget-matched $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}} = 99$.

based *savings* of +5.6% (Intel) and +28.8% (AMD), the opposite conclusion from Section 4.4.1. We do not treat this as a competing primary result: licenses are sold per core, and this report's entire cost and time model (Section 3) is denominated per core. Socket-level TDP growth conflates real per-core efficiency gains with the unrelated fact that more cores now fit on a similarly-sized die. We report the socket-level number here, once, for completeness, rather than developing it as a parallel analysis.

4.5.5 Idle Capacity Repurposed vs. Wasted

Sections 3.5.1 and 3.5.2 both assume idle cores accrue no additional operational carbon, applied identically to both models. We test that assumption two ways. First, symmetrically: if idle time instead draws power at 20% of active intensity for both

models (the same idle-power fraction already used in Section 4.5.3), the primary result strengthens by roughly an order of magnitude (Intel −568%, AMD −890%, vs. −50%/−164% in the primary result). Active compute time is a small fraction of the year (roughly 17% at baseline, $H/8760$), so the much larger work-based fleet accumulates proportionally more idle-state carbon under this framing.

Instead, one could argue core-based's small, purpose-built fleet is less likely to be used as shared general-purpose infrastructure than work-based's larger, low-marginal-cost fleet. In such a case, idle time should be charged for core-based but not work-based. Under that framing, the result reverses for Intel (+47.9%, work-based favored) and nearly vanishes for AMD (−4.1%). Unlike the other parameters, this is an unproven assumption concerning differential customer infrastructure-sharing behavior. Section 4.4.1 uses the symmetric, no-idle-charge assumption throughout.

Chapter 5: Limitations

In the following, we discuss several limitations of this study as pointers for future work:

1. **Cores idle after job completion.** Operational carbon accumulates only for $T_{\text{lic}}(t)$ or $T_{\text{work}}(t)$ hours per year, not the full 8,760. This assumes that idle capacity is repurposed rather than wasted for both models (Section 3.5.1). Section 4.5.5 quantifies both alternatives: one where idle time is counted for both models, strengthening the primary result by roughly an order of magnitude, and one where idle power is counted only for core-based's smaller dedicated fleet, which reverses the sign for Intel and nearly zeroes it for AMD. The carbon-favorable result is highly sensitive to how idle power is counted, more so than to any other parameter tested in this report.

2. **One generation is modeled as one year.** Both f and i are measured per hardware generation but applied per simulation year (Section 2.2). Real cadence is closer to 1.5–2 years, so a 50-year horizon should be read as 50 generations. Converting both factors to per-calendar-year rates preserves $i < f$ for both vendors and therefore the sign of every result reported here, but compresses the magnitudes.

3. **Work-based fleet startup condition.** The work-based fleet is modeled as already in rolling steady state at $t = 0$, so its cohorts carry pre-baseline generations during the first L years and the two models do not start from parity (Section 4.4.1). The alternative build-new-at-t=0 convention starts at parity but understates the steady-state gap. In Case 1 it gives -33.6% and -64.7% at year 50, roughly 17 and 99 percentage points short of the steady-state figures.

4. **Refresh and lifetime assumptions.** Core-based always refreshes all P_{lic} cores annually; Sections 4.1–4.2 establish why this is the economically rational choice. Work-based cohorts are replaced at $L = 7$ years, with $N_{\text{work}} = N_{\text{lic}}$ representing maximum parallelism and smaller N_{work} giving proportionally fewer annual replacements.

5. **The grid is static.** CI_0 is held at its year-0 value for the full horizon, and applied identically to both vendors, since operational carbon depends on where the customer's datacenter draws power rather than on where the silicon was fabricated. Table 4.8 quantifies the first assumption: decarbonization at any rate shifts ΔC by at most a few points toward the embodied-only floor and never changes its sign. A customer on a substantially cleaner or dirtier grid than assumed would see the same sign with a smaller or larger operational gap.

6. **Absolute carbon values carry source uncertainty [25].** However, since both models use identical base values, ΔC (eq. (3.29)) is unaffected by uniform scaling of $C_{\text{emb},0}$ and CI_0 together. Scaling either alone changes the balance between embodied and operational carbon and therefore does move ΔC, as Table 4.8 shows.

7. **Embodied-carbon model simplifications.** Total die area, and therefore total per-die embodied carbon, is held constant across all five generations while core count grows substantially (28 to 64 Intel, 32 to 192 AMD). Real die area generally grows with core count, so this likely understates absolute embodied carbon in later generations. However, since this is applied identically to both licensing models it does not bias the core-vs-work comparison. The model also assumes a single manufactured die per socket. AMD's later generations (Genoa, Turin) are multi-chiplet packages with additional packaging and interconnect embodied carbon this model does not capture. This is likely the largest single source of embodied-carbon underestimation in this report, and a natural target for an ACT-style [3] methodology in future work.

8. **Estimation precision.** Per-core i and f are computed from first/last-generation endpoints in most of this report. Table 2.1 indicates volatility this smooths over (AMD's generation-over-generation range spans -24% to $+89\%$). Section 4.5.1 reports a 5-point log-linear fit as a robustness check for i. No equivalent multi-point data existed to refit f. Separately, several inputs (die area, manufacturing carbon intensity, core-limited idle-power fraction) are order-of-magnitude estimates. We did not run a sensitivity or Monte Carlo analysis across them, and reported percentages should be read as indicative of magnitude and sign, not precise forecasts.

9. **Socket-level accounting reverses the sign of the primary result** (Section 4.5.4). We use per-core throughout because it matches this report's licensing unit, but a reader who believes socket-level aggregate footprint is the more relevant metric for their use case (e.g., datacenter operator emissions reporting) should use that number instead of this report's per-core finding.

10. **Architectural carbon modeling.** Tools such as ACT [3] already provide more detailed embodied/operational accounting (yield, packaging, memory) than this report attempts. Our contribution is narrower, characterizing how a specific licensing mechanism interacts with refresh incentives, rather than a new general-purpose carbon estimation methodology.

11. **Hardware price is held constant.** c_h stays at \$100 per core in nominal terms for the full 50-year horizon, with no discounting and no salvage value on retired hardware. Cores get faster but never cheaper. This is what makes work-based hardware spend small in Table 4.6, and a declining c_h would narrow the gap in dollar terms while leaving the manufacturing-volume ratio, and therefore the carbon result, unchanged.

12. **The revenue-neutral rate is not a realizable tariff.** Equation (4.5) implies a $d(t)$ that falls as the core-based customer's fleet shrinks. A vendor cannot

observe whether a given customer is budget-bound or deadline-bound, so no single published rate achieves revenue neutrality across both types. A rate calibrated on one would systematically misprice the other. We use Eq. (4.5) as a normalization that isolates the licensing mechanism from the vendor's pricing choice, not as a scheme we propose.

Chapter 6: Summary and Conclusions

In this report, we investigated the impact of EDA licensing on cost, time and carbon emissions. There are four main findings. First, the mechanism effect is unconditional: per-core-year licensing ties completion time to license count in a way that economically favors annual refresh (Section 4.1), and work-based pricing removes that incentive by construction while matching or exceeding the cost/time benchmark it sets (Sections 4.3.1–4.3.4). Under a fixed deadline, with the vendor paid the same under both models by Eq. (4.5), work-based manufactures 4.7x to 7.8x times fewer cores than core-based.

Second, the operational carbon effect is conditional: its sign depends on where per-core power growth i sits relative to per-core performance growth f, and current industry data places both Intel and AMD on the side that favors core-based licensing, not work-based (Section 4.4.1).

Third, embodied carbon is a real but secondary contributor. It accounts for roughly 10% of cumulative emissions for both vendors and does not drive the sign of the primary result, since under a fixed budget at $N_{\text{work}}^{\text{budget}} = N_{\text{lic}}^{\text{budget}}$ core-based and work-based replace an identical number of cores annually.

Fourth, refresh strategy, fleet-size choice N_{work}, and workload core-utilization pattern (Section 4.5.3) change the *magnitude* of the carbon effect without changing its *sign* at this report's baseline $N_{\text{work}} = N_{\text{lic}}$. Across all three cases work-based trails core-based by 39% to 50% for Intel and 105% to 188% for AMD. Cost and carbon do not agree on which work-based replacement schedule is preferable, so a customer cannot optimize both with one policy as far as our current experiments. Core-based remains favored across most accounting granularities this report tests, with three exceptions: socket-level aggregation (Section 4.5.4), the asymmetric idle-power framing of Section 4.5.5, and core-limited accounting at fleets well below the

budget-matched point (Figure 4.7).

This report does not recommend a licensing model, and does not claim work-based pricing reduces carbon under current industry conditions. Instead it identifies the conditions, an i-vs-f crossover specifically, under which it would, and reports that the industry does not currently meet them.

References

[1] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ict: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, p. 100340, 2021.

[2] T. O'Neill, "Cutting the carbon footprint of future computer chips," The Salata Institute at Harvard University, Mar. 2024, accessed: 2026-08-14. [Online]. Available: https://salatainstitute.harvard.edu/cutting-the-carbon-footprint-of-future-computer-chips/

[3] U. Gupta, M. Elgamal, G. Hills, G.-Y. Wei, H.-H. S. Lee, D. Brooks, and C.-J. Wu, "Act: Designing sustainable computer systems with an architectural carbon modeling tool," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 784–799.

[4] SemiAnalysis, "Eda market primer," SemiAnalysis, 2026. [Online]. Available: https://newsletter.semianalysis.com/p/eda-market-primer

[5] S. Krishnapura, M. Ayyalasomayajula, S. K. Achuthan, V. Lal, A. Somasekhara, and T. Tang, "It@Intel: Increasing eda performance and throughput with 4th and 5th generation intel® xeon® scalable processors," Intel Corporation, White Paper, Aug. 2024, accessed: 2026-08-14. [Online]. Available: https://www.intel.com/content/dam/www/central-libraries/us/en/documents/intel-it-increasing-eda-with-xeon-paper.pdf

[6] AMD, "Amd epyc," 2026, accessed: 2026-08-14. [Online]. Available: https://en.wikichip.org/wiki/amd/epyc

[7] Intel, "Intel xeon platinum," 2026, accessed: 2026-08-14. [Online]. Available: https://en.wikichip.org/wiki/intel/xeon_platinum

[8] Standard Performance Evaluation Corporation, "SPEC CPU2017 result search," 2017, accessed: 2026-07-17. [Online]. Available: https://www.spec.org/cpu2017/results/

[9] Notebookcheck, "64-core amd epyc 7763 milan server cpu makes cautious first appearance on geekbench 5 in 128-core dual-processor configuration," Notebookcheck news article, 2021. [Online]. Available: https://www.notebookcheck.net/64-core-AMD-EPYC-7763-Milan-server-CPU-makes-cautious-first-appearance-on-Geekbench-5-in-128-core-dual-processor-configuration.526412.0.html

[10] ServeTheHome, "Memory bandwidth per core and per socket for intel xeon and amd epyc," ServeTheHome article, 2023. [Online]. Available: https://www.servethehome.com/memory-bandwidth-per-core-and-per-socket-for-intel-xeon-and-amd-epyc/

[11] Cloudflare, "Launching cloudflare's gen 13 servers – trading cache for cores for 2x edge compute performance," Cloudflare engineering blog, 2026. [Online]. Available: https://blog.cloudflare.com/gen13-launch/

[12] Intel Corporation, "Intel® Xeon® Platinum 8180 Processor Specifications," 2017. [Online]. Available: https://www.intel.com/content/www/us/en/products/sku/120496/intel-xeon-platinum-8180-processor-38-5m-cache-2-50-ghz/specifications.html

[13] ——, "Intel® Xeon® Platinum 8280 Processor Specifications," 2019. [Online]. Available: https://www.intel.com/content/www/us/en/products/sku/192478/intel-xeon-platinum-8280-processor-38-5m-cache-2-70-ghz/specifications.html

[14] I. Cutress, "Intel Ice Lake Xeon Platinum 8380 Review: 10nm Debuts for the Data Center," 2021. [Online]. Available: https://www.tomshardware.com/news/intel-ice-lake-xeon-platinum-8380-review-10nm-debuts-for-the-data-center

[15] M. Larabel, "4th Gen Intel Xeon Scalable Sapphire Rapids Leaps Forward," 2023. [Online]. Available: https://www.servethehome.com/4th-gen-intel-xeon-scalable-sapphire-rapids-leaps-forward/2/

[16] R. Shrout, "Intel 5th Gen Xeon Emerald Rapids Pushes up to 64 Cores," 2023. [Online]. Available: https://www.tomshardware.com/pc-components/cpus/intel-5th-gen-xeon-emerald-rapids-pushes-up-to-64-cores-320mb-l3-cache

[17] Advanced Micro Devices, Inc., "AMD EPYC™ 7601 Processor Specifications," 2017. [Online]. Available: https://www.amd.com/en/support/downloads/drivers.html/processors/epyc/epyc-7001-series/amd-epyc-7601.html

[18] Microway, Inc., "Detailed Specifications of the AMD EPYC "Rome" CPUs," 2019. [Online]. Available: https://www.microway.com/knowledge-center-articles/detailed-specifications-of-the-amd-epyc-rome-cpus/

[19] R. Shrout, "From Rome to Milan: AMD's Zen 3 EPYC 7003 CPUs Performance Tested," 2021. [Online]. Available: https://techgage.com/article/amd-epyc-milan-launch-performance/

[20] T. P. Morgan, "Why AMD "Genoa" EPYC Server CPUs Take the Heavyweight Title," 2022. [Online]. Available: https://www.nextplatform.com/2022/11/10/amd-genoa-epyc-server-cpus-take-the-heavyweight-title/

[21] P. Alcorn, "AMD EPYC Turin 9005 Series — We Benchmark 192-core Zen 5 Chip with 500W TDP," 2024. [Online]. Available: https://www.tomshardware.com/pc-components/cpus/amd-launches-epyc-turin-9005-series

[22] C. Clemm, L. Stobbe, K. Wimalawarne, and J. Druschke, "Towards green ai: Current status and future research," 2024. [Online]. Available: https://arxiv.org/abs/2407.10237

[23] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing carbon: The elusive environmental footprint of computing," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2021, pp. 854–867.

[24] B. C. Lee, D. Brooks, A. van Benthem, U. Gupta, G. Hills, V. Liu, B. Pierce, C. Stewart, E. Strubell, G.-Y. Wei, A. Wierman, Y. Yao, and M. Yu, "Carbon connect: An ecosystem for sustainable computing," 2024. [Online]. Available: https://arxiv.org/abs/2405.13858

[25] M. Elgamal, D. Carmean, E. Ansari, O. Zed, R. Peri, S. Manne, U. Gupta, G.-Y. Wei, D. Brooks, G. Hills, and C.-J. Wu, "Carbon-efficient design optimization for computing systems," in *Proceedings of the 2nd Workshop on Sustainable Computer Systems (HotCarbon '23)*, 2023, pp. 1–7.

[26] Intel Corporation, "2025 annual report and proxy statement (form ars)," SEC EDGAR filing, 2026. [Online]. Available: https://www.sec.gov/Archives/edgar/data/50863/000005086326000060/intc014890-arsa.pdf

[27] Taiwan Semiconductor Manufacturing Company, "A practitioner of green power," TSMC ESG Report, 2026. [Online]. Available: https://esg.tsmc.com/en-US/file/public/e-APractitionerofGreenPower_1.pdf

[28] Synopsys, Inc., "Sta strategies for fast and efficient signoff in multi-billion instance designs," Synopsys technical article, 2026. [Online]. Available: https://www.synopsys.com/articles/sta-strategies-multi-billion-instance-designs.html

[29] ——, "Synopsys' primetime speeds timing and power closure for complex soc and iot designs," Press release, 2016. [Online]. Available: https://news.synopsys.com/2016-03-23-Synopsys-PrimeTime-Speeds-Timing-and-Power-Closure-for-Complex-SoC-and-IoT-Designs