

RAPID: An Efficient and Accurate Ratio-Based Analytical Proxy for Interconnect Delay

Haichuan Liu¹, Zizheng Guo¹², Zuodong Zhang², Yuanqing Cheng³,
Guannan Guo^{1*}, Runsheng Wang¹²⁴, and Yibo Lin^{124*}

¹ School of Integrated Circuits, Peking University, Beijing, China ² Institute of EDA, Peking University, Wuxi, China

³ Integrated Circuit Science and Engineering, Beihang University, Beijing, China

⁴ Beijing Advanced Innovation Center for Integrated Circuits, Beijing, China

haichuanliu@stu.pku.edu.cn, {gzz, guannan.guo, r.wang, yibolin}@pku.edu.cn,
zhangzd@pkueda.org.cn, yuanqing@buaa.edu.cn

Abstract

Iterative physical-design optimization requires accurate timing feedback at proxy-timer speed. Conventional interconnect models expose an accuracy–efficiency tradeoff: Elmore delay is fast but loses waveform-shape information, whereas higher-order methods improve fidelity at the cost of more expensive distribution or waveform evaluation. We introduce **RAPID**, a ratio-based analytical proxy that moves Elmore delay toward higher-order timing fidelity through a positive, slew-aware delay-ratio correction. RAPID derives the correction from Wilson–Hilferty asymptotics of a moment-matched Gamma response and extends it to finite input transitions through normalized-slew dilution. The resulting ratio is computed by an Elmore-isomorphic linear-time traversal and used to calibrate interconnect delay and slew within a nonlinear delay model (NLDM)-compatible effective-load flow. On TAU benchmarks, RAPID achieves average $R^2 = 0.9905$ and end-to-end report_timing speedups of 9.84× over PrimeTime and 32.70× over OpenSTA. Without retuning, it reaches average $R^2 = 0.9949$ and mean absolute error (MAE) = 0.526 ps on unseen industrial ARM designs at Elmore-level runtime. In the evaluated repair flow, RAPID improves sign-off worst negative slack (WNS) and total negative slack (TNS) by up to 22.28% and 15.44%, respectively.

Keywords

static timing analysis, interconnect delay, circuit moments, GPU acceleration

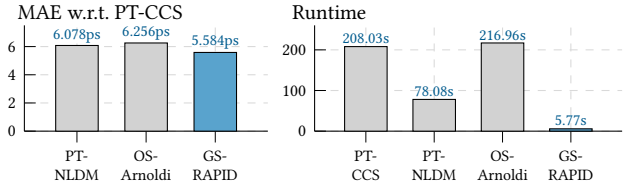
ACM Reference Format:

Haichuan Liu, Zizheng Guo, Zuodong Zhang, Yuanqing Cheng, Guannan Guo, Runsheng Wang, and Yibo Lin. 2027. RAPID: An Efficient and Accurate Ratio-Based Analytical Proxy for Interconnect Delay. In *32nd Asia and South Pacific Design Automation Conference (ASPDAC '27)*, January 25–28, 2027, Tokyo, Japan. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3840404.3848149>

*Corresponding authors.



This work is licensed under a Creative Commons Attribution 4.0 International License. ASPDAC '27, Tokyo, Japan
© 2027 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2923-2/2027/01
<https://doi.org/10.1145/3840404.3848149>



(a) Accuracy.

(b) Runtime.

PT: PrimeTime [40]; OS: OpenSTA [33]; GS: graphics processing unit (GPU) STA engine; suffix: delay model.

MAE is relative to PT's composite current source (CCS) configuration.

Figure 1: Accuracy and end-to-end runtime on leon2_iccad [17]. PT-CCS is the sign-off reference; the other configurations expose RAPID's target fidelity–runtime tradeoff.

1 Introduction

Iterative physical-design optimization depends on fast, accurate static timing analysis (STA) feedback [8, 17]. Sign-off models preserve waveform fidelity but incur high inner-loop cost (Fig. 1) [40, 42]. Proxy timers instead use the nonlinear delay model (NLDM) and fast net-delay approximations to trade some fidelity for throughput [21, 22, 25]. Within buffering and gate-sizing loops, proxy timing feedback guides the choice of optimization moves, while sign-off timing evaluates the resulting design.

Interconnect delay remains a major proxy-timing error. Elmore retains only the first response moment [10, 38], while Gaussian slew assumes symmetry [10, 19]. Shielding, long tails, and finite input slew create asymmetric responses and shifted crossings beyond a first-moment model (Fig. 2). Full transient analysis captures these effects but is too costly for inner-loop evaluation [11, 26, 31].

Between these extremes, moment-based model-order reduction (MOR), including asymptotic waveform evaluation (AWE) and Arnoldi methods, uses circuit moments to construct compact pole-residue models of interconnect responses [13, 32, 34, 35, 39]. Distribution fitting obtains delay from a moment-matched cumulative distribution function (CDF) median through a look-up table (LUT), special function, or numerical inversion [6, 20, 27, 28, 44]. The effective capacitance metric (ECM) and delay metric based on two moments (D2M) avoid reconstruction but use specific waveform assumptions or empirical ratios [4, 5, 18]. Learned predictors reduce inference cost but require training data and deployment integration [12, 16, 30, 41].

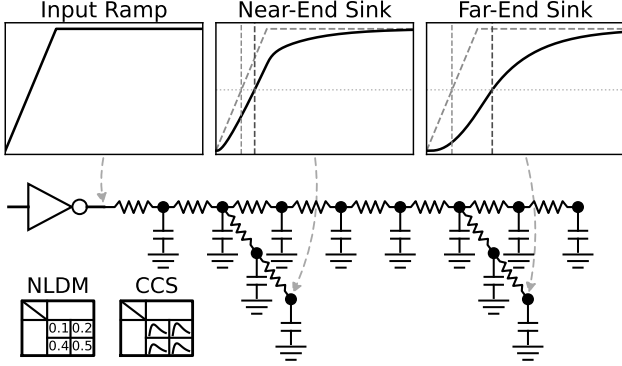


Figure 2: Finite-slew propagation through a resistor-capacitor (RC) tree: shielding and distributed loading produce asymmetric, long-tailed sink responses with shifted 50% crossings.

Together, these tradeoffs motivate **RAPID**, a ratio-based analytical proxy that moves Elmore delay toward higher-order timing fidelity through a direct, slew-aware delay-ratio correction without waveform reconstruction. The correction is first derived from Wilson–Hilferty asymptotics [43] of a moment-matched Gamma response and then extended to finite input transitions by normalized-slew dilution. Through an Elmore-isomorphic linear-time traversal, the resulting ratio is computed and used to correct interconnect delay and slew within an NLDM-compatible effective-load flow. Our contributions are:

- (1) An analytically derived delay-ratio correction that combines Wilson–Hilferty asymptotics of a moment-matched Gamma response with a positive exponential reparameterization, moving Elmore delay toward higher-order timing fidelity without inverse-CDF or reduced-waveform evaluation.
- (2) A physically constrained correction combining normalized-slew dilution, canonical RC-ladder anchoring, and ratio-modulated coupling to slew propagation.
- (3) An Elmore-isomorphic $O(N)$ RC-tree kernel—the same post-order and pre-order accumulation pattern as Elmore—that evaluates the delay-ratio correction with $O(1)$ work per node without matrix inversion or waveform solves.

RAPID reaches average $R^2 = 0.9905$ on TAU [17] and 0.9949 on unseen ARM designs [7] at Elmore-level runtime. In the evaluated repair flow, PrimeTime reports WNS gains or a match and TNS reductions on all four designs.

2 Preliminaries

2.1 NLDM and Effective Capacitance

Fast proxy STA evaluates cell delay and net delay separately. For each timing query, the net-delay calculator receives an extracted RC tree with fixed topology and parasitics, typically containing tens to hundreds of nodes, and computes the sink delays and slews for a given driver slew. RAPID serves as this net-delay calculator, directly operating on the original RC tree through recursive computation and coupling to cell timing through the NLDM load-and-slew

interface:

$$\text{delay} = \text{LUT}_{\text{delay}}(T_{\text{in}}, C_{\text{load}}), \quad (1)$$

$$T_{\text{out}} = \text{LUT}_{\text{slew}}(T_{\text{in}}, C_{\text{load}}). \quad (2)$$

Resistive shielding makes the effective load smaller than C_{total} [37]. Closed-form C_{eff} approximations avoid iterative charge conservation [9] with comparable NLDM accuracy [29, 36]. This interface supplies load; threshold-crossing delay still needs a response-shape descriptor.

2.2 Circuit Moments and Recursive Computation

Circuit moments provide this descriptor without waveform reconstruction. For a rooted RC tree $\mathcal{T}$ with $N = |\mathcal{T}|$ nodes, node capacitance C_i , parent-edge resistance r_i , and parent p of node i , expand the transfer function around $s = 0$:

$$H_i(s) = \int_{t=0}^{\infty} h_i(t) e^{-st} dt = \sum_{\ell=0}^{\infty} \tilde{m}_\ell(i) s^\ell, \quad (3)$$

$$\tilde{m}_\ell(i) = \frac{(-1)^\ell}{\ell!} \int_{t=0}^{\infty} h_i(t) t^\ell dt, \quad (4)$$

where $\tilde{m}_\ell(i)$ is the ℓ -th circuit moment and $\tilde{m}_0(i) = 1$ for a passive RC tree. Kirchhoff’s current and voltage laws give the linear-time recursion

$$\tilde{m}_\ell(i) = \tilde{m}_\ell(p) - r_i \sum_{j \in \mathcal{T}_i} C_j \tilde{m}_{\ell-1}(j), \quad (5)$$

where $\mathcal{T}_i$ is the subtree rooted at i ; each additional moment order costs $O(N)$. The raw moments determine mean m_1 and central moments μ_2, μ_3 , describing dispersion and asymmetry. The 50% step-response crossing is the CDF median, connecting these descriptors to delay.

2.3 Probabilistic Interpretation of Circuit Response

The normalized RC impulse response satisfies $h(t) \geq 0$ and $\int_0^\infty h(t) dt = 1$, so $h(t)$ is a probability density function (PDF) and the step response $g(t)$ is its cumulative distribution function (CDF) [10, 19]. Its ℓ -th raw moment is

$$m_\ell = \int_0^\infty t^\ell h(t) dt = (-1)^\ell \ell! \tilde{m}_\ell. \quad (6)$$

Here $m_1 = -\tilde{m}_1$ is Elmore delay [10]. The relevant central moments are

$$\mu_2 = m_2 - m_1^2 = 2\tilde{m}_2 - \tilde{m}_1^2, \quad (7)$$

$$\mu_3 = m_3 - 3m_1m_2 + 2m_1^3 = -6\tilde{m}_3 + 6\tilde{m}_1\tilde{m}_2 - 2\tilde{m}_1^3. \quad (8)$$

Variance μ_2 measures dispersion and serves as a proxy for slew [10], while μ_3 captures asymmetry.

Prior work matches a parametric PDF to two or three circuit moments and evaluates delay as its median. A Gaussian is symmetric and extends to $t < 0$; a Gamma PDF captures causal, right-skewed RC responses more faithfully (Fig. 3).

A three-moment shifted Gamma model is

$$h(t) \approx h_\Gamma(t) = \frac{(t - t_0)^{k_3 - 1}}{\theta^{k_3} \Gamma(k_3)} \exp\left(-\frac{t - t_0}{\theta}\right), \quad t \geq t_0. \quad (9)$$

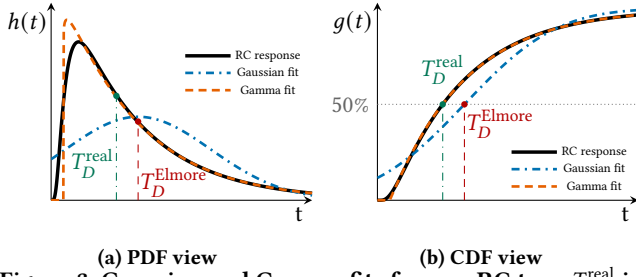


Figure 3: Gaussian and Gamma fits for one RC tree; T_D^{real} is its 50% crossing and $T_D^{\text{Elmore}} = m_1$. Gamma better captures crossing asymmetry.

Moment matching determines (t_0, k_3, θ) :

$$m_1 = t_0 + k_3\theta, \quad \mu_2 = k_3\theta^2, \quad \mu_3 = 2k_3\theta^3, \quad (10)$$

thus $k_3 = 4\mu_2^3/\mu_3^2$ [27]. Its median requires a special function, LUT, or numerical inversion. RAPID instead extracts a closed-form ratio from an unshifted two-moment Gamma bridge.

3 RAPID Methodology

This section derives RAPID's slew-aware delay-ratio correction, anchors its step-response rate on uniform RC ladders, and evaluates it through an Elmore-isomorphic traversal for delay and slew propagation.

3.1 Slew-Aware Delay-Ratio Correction

3.1.1 Response Shape and Normalized Slew. Under the probabilistic RC interpretation [10], we summarize the RC response by

$$m_1 = \int_0^\infty t h(t) dt, \quad \mu_2 = \int_0^\infty (t - m_1)^2 h(t) dt, \quad (11)$$

where m_1 is Elmore delay and μ_2 is temporal dispersion. We separate structure from excitation by

$$\kappa = \frac{\mu_2}{m_1^2}, \quad \eta_\sigma = \frac{T_{in}}{\sqrt{\mu_2}}, \quad (12)$$

where T_{in} is the driver slew at the net input, κ is the response-shape factor, a dimensionless squared coefficient of variation summarizing moment-derived temporal dispersion, and η_σ is normalized input slew. We define the target normalized delay

$$\mathcal{R} = \frac{t_{50}^{\text{ramp}}}{m_1}, \quad (13)$$

where t_{50}^{ramp} is the output 50% crossing time relative to the input 50% crossing. In the step-input limit, normalization by m_1 yields a ratio in $(0, 1]$ under Elmore's upper bound for monotone RC responses [15]. We first derive this step-input ratio and then extend it to finite input slew.

3.1.2 Step-Response Asymptotics via Wilson-Hilferty Transformation. To derive the step-response asymptotic, we approximate the right-skewed response by an unshifted, two-moment $\Gamma(k_2, \theta)$ PDF

$$f_X(x) = \frac{1}{\Gamma(k_2)\theta^{k_2}} x^{k_2-1} e^{-x/\theta}, \quad x > 0, \quad (14)$$

with $m_1 = k_2\theta$ and $\mu_2 = k_2\theta^2$, hence

$$k_2 = \frac{m_1^2}{\mu_2} = \frac{1}{\kappa}, \quad \theta = \frac{\mu_2}{m_1} = m_1\kappa. \quad (15)$$

Applying the Wilson-Hilferty approximation [43] to this moment-matched Gamma response yields the step-response ratio.

Theorem 1 (Wilson-Hilferty First-Order Median Approximation). *Let the RC impulse response be approximated by a moment-matched Gamma distribution $X \sim \Gamma(k_2, \theta)$. As $k_2 \rightarrow \infty$ (equivalently, $\kappa = \mu_2/m_1^2 \rightarrow 0$), the Wilson-Hilferty approximation gives the first-order step-response delay ratio*

$$\mathcal{R}_{\text{step}} = \frac{t_{50}^{\text{step}}}{m_1} = 1 - \frac{1}{3}\kappa + O(\kappa^2). \quad (16)$$

PROOF. Let $Y = 2X/\theta \sim \chi_\nu^2$ with $\nu = 2k_2$. Wilson-Hilferty gives

$$\left(\frac{Y}{\nu}\right)^{1/3} \approx \mathcal{N}\left(1 - \frac{2}{9\nu}, \frac{2}{9\nu}\right). \quad (17)$$

The normal distribution is symmetric, so its median equals its mean. Since the cube-root transformation is increasing, setting $\nu = 2k_2$ gives

$$\text{median}(Y) \approx 2k_2 \left(1 - \frac{1}{9k_2}\right)^3. \quad (18)$$

Rescaling by $X = \theta Y/2$ and $m_1 = k_2\theta$ gives

$$\begin{aligned} \frac{t_{50}^{\text{step}}}{m_1} &\approx \frac{(\theta/2) \text{median}(Y)}{k_2\theta} = \left(1 - \frac{1}{9k_2}\right)^3 \\ &= 1 - \frac{1}{3k_2} + \frac{1}{27k_2^2} - \frac{1}{729k_2^3}. \end{aligned} \quad (19)$$

Substituting $\kappa = 1/k_2$ completes the proof. $\square$

Because $1 - \kappa/3$ becomes negative when $\kappa > 3$, we introduce the strictly positive surrogate

$$\widehat{\mathcal{R}}_{\text{step}} = \exp(-\lambda\kappa) = 1 - \lambda\kappa + O(\kappa^2), \quad (20)$$

whose Wilson-Hilferty first-order match gives $\lambda_{\text{theory}} = 1/3$, asymptotic as $\kappa \rightarrow 0$. For any positive λ , the exponential satisfies $\widehat{\mathcal{R}}_{\text{step}} \in (0, 1]$, consistent with Elmore's bound for monotone RC step responses [15]. But this $\kappa \rightarrow 0$ asymptote does not hold for realistic RC trees with finite κ ; we instead anchor λ to canonical structures (Section 3.2).

3.1.3 Normalized-Slew Dilution. To extend the step correction to finite input transitions, let $x(t)$ be a unit-amplitude input ramp, whose normalized derivative integrates to one, and let $y(t)$ be the output; convolution gives

$$y'(t) = x'(t) * h(t), \quad (21)$$

Theorem 2 (Variance Additivity Under Convolution). *Let σ_{in}^2 denote the variance of the normalized input derivative $x'(t)$. The variance of the output derivative satisfies*

$$\mu_2^{(\text{out})} = \mu_2 + \sigma_{in}^2. \quad (22)$$

Writing $\sigma_{in}^2 = a_s T_{in}^2$ for a waveform-dependent constant a_s yields

$$\mu_2^{(\text{out})} = \mu_2 (1 + a_s \eta_\sigma^2). \quad (23)$$

PROOF. The normalized kernels $x'(t)$ and $h(t)$ represent independent variables X_{in} and X_{rc} . Their convolution is the density of $X_{in} + X_{rc}$, whose variance is

$$\begin{aligned}\text{Var}(X_{in} + X_{rc}) &= \text{Var}(X_{in}) + \text{Var}(X_{rc}) \\ &\quad + 2 \text{Cov}(X_{in}, X_{rc}) \\ &= \sigma_{in}^2 + \mu_2,\end{aligned}\quad (24)$$

where independence makes the covariance term zero. $\square$

As $\eta_\sigma \rightarrow \infty$, input broadening dominates and κ should have less influence. We require

$$\Psi(0) = 1, \quad \lim_{\eta_\sigma \rightarrow \infty} \Psi(\eta_\sigma) = 0, \quad (25)$$

and choose a first-order rational model satisfying these limits,

$$\Psi(\eta_\sigma) = \frac{1}{1 + \beta \eta_\sigma}, \quad (26)$$

which yields

$$\widehat{\mathcal{R}} = \exp(-\lambda \kappa \Psi(\eta_\sigma)) = \exp\left(-\frac{c_1 \kappa}{\eta_\sigma + c_2}\right), \quad (27)$$

where $c_2 = 1/\beta$ and $c_1 = \lambda/\beta$. This positive, slew-aware correction preserves the step-input limit and suppresses structural sensitivity for slow inputs.

3.2 Canonical RC-Ladder Anchoring

We anchor the correction's remaining step-response coefficient c_1/c_2 to a uniform M -section RC ladder between lumped and distributed extremes.

Theorem 3 (Moments of the Uniform M -Section RC Ladder). *Consider $M \geq 1$ identical sections with total resistance R and capacitance C , section values $r = R/M$, $c = C/M$, and far-end node M . Then*

$$m_1(M) = \frac{RC}{2} \frac{M+1}{M}, \quad \mu_2(M) = \frac{(RC)^2}{6} \frac{(M+1)(M^2 + M + 1)}{M^3}, \quad (28)$$

and therefore

$$\kappa(M) = \frac{2}{3} \left(1 + \frac{1}{M(M+1)}\right). \quad (29)$$

PROOF. With shared resistance $R_{i,j} = \min(i, j)r$, the first raw moment at node j is

$$\begin{aligned}m_1(j) &= rc \sum_{i=1}^M \min(i, j) \\ &= rc \left[\frac{j(j+1)}{2} + j(M-j) \right] \\ &= \frac{rc}{2} j(2M - j + 1).\end{aligned}\quad (30)$$

At the far end, $j = M$ gives

$$m_1(M) = \frac{rc}{2} M(M+1) = \frac{RC}{2} \frac{M+1}{M}. \quad (31)$$

For the second raw moment, substituting $R_{M,j} = jr$ gives

$$\begin{aligned}m_2(M) &= 2rc \sum_{j=1}^M j m_1(j) \\ &= (rc)^2 \sum_{j=1}^M [(2M+1)j^2 - j^3].\end{aligned}\quad (32)$$

Using the power sums

$$\sum_{j=1}^M j^2 = \frac{M(M+1)(2M+1)}{6}, \quad \sum_{j=1}^M j^3 = \frac{M^2(M+1)^2}{4}, \quad (33)$$

we obtain

$$m_2(M) = (rc)^2 \frac{M(M+1)(5M^2 + 5M + 2)}{12}. \quad (34)$$

The second central moment is therefore

$$\begin{aligned}\mu_2(M) &= m_2(M) - m_1(M)^2 \\ &= (rc)^2 \frac{M(M+1)(M^2 + M + 1)}{6}.\end{aligned}\quad (35)$$

Substituting $r = R/M$, $c = C/M$ yields the stated $\mu_2(M)$. Finally,

$$\kappa(M) = \frac{\mu_2(M)}{m_1(M)^2} = \frac{2(M^2 + M + 1)}{3M(M+1)}, \quad (36)$$

which gives the stated shape factor. $\square$

Theorem 4 (Monotonicity of Dominant-Mode Ratio and Effective Decay Rate). *For the uniform M -section RC ladder, let t_{50}^{dom} be the 50% crossing time of its dominant-mode step-response reference and define*

$$\mathcal{R}_{\text{dom}}(M) = \frac{t_{50}^{\text{dom}}(M)}{m_1(M)}, \quad \Lambda(M) = -\frac{\ln \mathcal{R}_{\text{dom}}(M)}{\kappa(M)}. \quad (37)$$

Then $\mathcal{R}_{\text{dom}}(M)$ is strictly increasing under continuous extension to real $M \geq 1$, and $\Lambda(M)$ is strictly increasing for integers $M \geq 1$.

PROOF. For a ladder mode u_j with pole s , Kirchhoff's laws give

$$\begin{aligned}-u_{j-1} + 2u_j - u_{j+1} &= -src u_j, \\ u_0 &= 0, \quad u_{M+1} = u_M.\end{aligned}\quad (38)$$

Setting $u_j = \sin(j\vartheta)$ gives $-src = 2 - 2\cos\vartheta$. The far-end boundary requires $\sin((M+1)\vartheta) = \sin(M\vartheta)$, hence the nonzero modes have

$$\vartheta_\ell = \frac{(2\ell-1)\pi}{2M+1}, \quad \ell = 1, \dots, M. \quad (39)$$

The slowest mode has $\vartheta_1 = 2\alpha_M$, where $\alpha_M = \pi/(4M+2)$.

The normalized ladder conductance matrix is symmetric, so its modes are orthogonal. Under a unit step, the initial voltage error relative to unit steady state is 1. Projecting it onto $u_j = \sin(j\vartheta_1)$ gives the far-end residue

$$A_1 = \frac{u_M \sum_{j=1}^M u_j}{\sum_{j=1}^M u_j^2}. \quad (40)$$

The sine-sum identities and $(2M+1)\alpha_M = \pi/2$ give $u_M = \cos\alpha_M$ and

$$\sum_{j=1}^M u_j = \frac{\cos\alpha_M}{2\sin\alpha_M}, \quad \sum_{j=1}^M u_j^2 = \frac{2M+1}{4}. \quad (41)$$

Thus the dominant pole and residue are

$$s_1 = -\frac{4M^2 \sin^2 \alpha_M}{RC}, \quad A_1(M) = \frac{4}{\pi} \frac{\alpha_M \cos^2 \alpha_M}{\sin \alpha_M}. \quad (42)$$

For $M = 1$ this is the exact single-pole response; for $M > 1$ it defines the dominant-mode reference used for coefficient anchoring:

$$g_{\text{dom}}(t) = 1 - A_1 e^{s_1 t}. \quad (43)$$

Solving $g_{\text{dom}}(t_{50}^{\text{dom}}) = 1/2$ with $\tau_{\text{dom}} = -1/s_1$ gives $t_{50}^{\text{dom}} = \tau_{\text{dom}} \ln(2A_1)$, so

$$\mathcal{R}_{\text{dom}}(M) = \frac{\ln \left[\frac{8}{\pi} \left(\frac{\alpha_M \cos^2 \alpha_M}{\sin \alpha_M} \right) \right]}{2M(M+1) \sin^2 \alpha_M}. \quad (44)$$

Extend M through $\alpha = \pi/(4M+2)$, where $d\alpha/dM < 0$ and $\alpha \in (0, \pi/6]$, and write $\mathcal{R}_{\text{dom}} = U/V$ with

$$U(\alpha) = \ln \left[\frac{8}{\pi} \left(\frac{\alpha \cos^2 \alpha}{\sin \alpha} \right) \right], \quad V(\alpha) = \left(\frac{\pi^2}{8\alpha^2} - \frac{1}{2} \right) \sin^2 \alpha, \quad (45)$$

On this interval, $\alpha/\sin \alpha > 1$ and $\cos^2 \alpha \geq 3/4$ imply $U > \ln(6/\pi) > 0$, while $V > 0$. Primes denote $d/d\alpha$. Define $P = -V'/V$ and $Q = -U'/U$; then

$$\begin{aligned} \frac{\mathcal{R}'_{\text{dom}}}{\mathcal{R}_{\text{dom}}} &= P - Q, \\ P &= 2 \left(\frac{1}{\alpha} - \cot \alpha + \frac{4\alpha}{\pi^2 - 4\alpha^2} \right), \\ Q &= \frac{2 \tan \alpha + \cot \alpha - 1/\alpha}{U}. \end{aligned} \quad (46)$$

Let $\phi_P = P/\alpha$ and $\phi_Q = Q/\alpha$. The Mittag-Leffler expansion of $\cot \alpha$ gives

$$\phi_P = 4 \sum_{\ell=1}^{\infty} \frac{1}{\ell^2 \pi^2 - \alpha^2} + \frac{8}{\pi^2 - 4\alpha^2}, \quad (47)$$

so ϕ_P increases. The Bernoulli expansions of $\tan \alpha$ and $\cot \alpha$ give

$$\phi_Q = \frac{1}{U} \sum_{\ell=1}^{\infty} \frac{|B_{2\ell}| 2^{2\ell} (2^{2\ell+1} - 3)}{(2\ell)!} \alpha^{2\ell-2}, \quad (48)$$

where $B_{2\ell}$ are Bernoulli numbers. Its numerator increases while U decreases and remains positive; hence ϕ_Q increases. Moreover,

$$\lim_{\alpha \rightarrow 0^+} \phi_Q = \frac{5}{3 \ln(8/\pi)} > \frac{81}{\pi^2} - \frac{12\sqrt{3}}{\pi} = \phi_P(\pi/6). \quad (49)$$

Thus $Q > P$, so $\mathcal{R}'_{\text{dom}}(\alpha) < 0$ and $d\mathcal{R}_{\text{dom}}/dM > 0$.

For Λ , set $S = -\ln \mathcal{R}_{\text{dom}}$ and use

$$\kappa(\alpha) = \frac{2}{3} \frac{\pi^2 + 12\alpha^2}{\pi^2 - 4\alpha^2}, \quad \frac{\Lambda'}{\Lambda} = \frac{S'}{S} - \frac{\kappa'}{\kappa}. \quad (50)$$

Differentiating the shape factor gives

$$\frac{\kappa'}{\kappa} = \frac{32\pi^2 \alpha}{(\pi^2 - 4\alpha^2)(\pi^2 + 12\alpha^2)}. \quad (51)$$

On $\alpha \in (0, \pi/10]$ (i.e., $M \geq 2$), the denominator increases with α^2 and is at most $\pi^4(1 - 4/100)(1 + 12/100) = 1.0752\pi^4$. Together with

monotonicity and endpoint evaluation, this gives

$$\begin{aligned} \frac{Q}{\alpha} &\leq \phi_Q(\pi/10) < 2.081, & \frac{P}{\alpha} &> \lim_{\alpha \rightarrow 0^+} \phi_P = \frac{2}{3} + \frac{8}{\pi^2} > 1.477, \\ S &> -\ln \left[\frac{8}{\pi^2} \ln \left(\frac{8}{\pi} \right) \right] > 0.277, \\ \frac{\kappa'}{\kappa} &\geq \frac{32\alpha}{1.0752\pi^2} > 2.894\alpha. \end{aligned} \quad (52)$$

Since $S' = Q - P$, these bounds imply $S'/S < 0.604\alpha/0.277 < 2.181\alpha < \kappa'/\kappa$. Therefore $\Lambda'(\alpha) < 0$ and $\Lambda(M)$ increases for integers $M \geq 2$. Finally, $\Lambda(1) \in (0.36651, 0.36652)$ and $\Lambda(2) \in (0.38276, 0.38277)$ establish $\Lambda(2) > \Lambda(1)$, completing the proof. $\square$

Theorems 3–4 give a monotone interpolation from $\kappa = 1$ ($M = 1$) to $2/3$ ($M \rightarrow \infty$). RAPID uses the resulting lumped and distributed endpoint rates as physics-guided coefficient anchors.

For a step input, the endpoints are:

- **Lumped RC:** $\kappa = 1$ and median $RC \ln 2$ give

$$\mathcal{R}_L = \ln 2, \quad \lambda_L = -\ln(\ln 2) \approx 0.3665. \quad (53)$$

- **Distributed limit:** As $M \rightarrow \infty$, $\alpha_M \rightarrow 0$ gives $A_1 \rightarrow 4/\pi$ and $\tau_{\text{dom}} \rightarrow 4RC/\pi^2$. With $m_1 \rightarrow RC/2$ and $\kappa \rightarrow 2/3$, this yields

$$\mathcal{R}_D = \frac{8}{\pi^2} \ln \left(\frac{8}{\pi} \right), \quad \lambda_D = -\frac{3}{2} \ln \left(\frac{8}{\pi^2} \ln \left(\frac{8}{\pi} \right) \right) \approx 0.4163. \quad (54)$$

An equal-weight log-domain least-squares fit over these anchors minimizes

$$J(\lambda) = \sum_{i \in \{L,D\}} (\ln \mathcal{R}_i + \lambda \kappa_i)^2, \quad (55)$$

Setting its derivative to zero gives

$$J'(\lambda) = 2 \sum_{i \in \{L,D\}} \kappa_i (\ln \mathcal{R}_i + \lambda \kappa_i) = 0. \quad (56)$$

Since $J''(\lambda) = 2 \sum_i \kappa_i^2 > 0$, the unique minimizer is

$$\lambda^* = -\frac{\sum_{i \in \{L,D\}} \kappa_i \ln \mathcal{R}_i}{\sum_{i \in \{L,D\}} \kappa_i^2} \approx 0.3818. \quad (57)$$

The derived correction uses the analytically anchored rate $\lambda = \lambda^* = c_1/c_2$, superseding the local first-order value λ_{theory} . We use the fixed empirical slew scale $c_2 = 0.7515$; the numerator coefficient c_1 then follows from $c_1 \simeq \lambda^* c_2$:

$$c_2 = 0.7515, \quad c_1 = 0.2870, \quad \frac{c_1}{c_2} \simeq \lambda^* \simeq 0.3818. \quad (58)$$

3.3 Elmore-Isomorphic Traversal

With c_1, c_2 fixed, Algorithm 1 evaluates $\widehat{\mathcal{R}}$ using Elmore's post-order and pre-order dataflow. Here L_0 is subtree capacitance, L_1 its first-moment aggregate, and net delay is $m_1 \widehat{\mathcal{R}}$; $O(1)$ work per node gives $O(N)$ complexity.

3.4 Consistent Slew Propagation

NLDM effective-capacitance evaluation depends on propagated slew, commonly modeled as [29]

$$T_{\text{out}} = \sqrt{T_{\text{in}}^2 + k_s^2 \cdot \mu_2}, \quad (59)$$

Algorithm 1 RAPID Delay-Ratio Computation for an RC Tree**Require:** Rooted tree $\mathcal{T}$, C_v , non-root edge r_v , T_{in} , and fixed c_1, c_2

```

1: Initialize  $m_1, \tilde{m}_2, \mu_2, \kappa, \eta_\sigma, \hat{\mathcal{R}}$  to zero;  $L_0[v] \leftarrow C_v$  for node  $v$ 
2: for each non-root node  $v$  in post-order do
3:    $p \leftarrow$  parent of  $v$ ,  $L_0[p] \leftarrow L_0[p] + L_0[v]$ 
4: for each non-root node  $v$  in pre-order do
5:    $p \leftarrow$  parent of  $v$ ,  $m_1[v] \leftarrow m_1[p] + r_v L_0[v]$ 
6:  $L_1[v] \leftarrow C_v m_1[v]$  for every node  $v$ 
7: for each non-root node  $v$  in post-order do
8:    $p \leftarrow$  parent of  $v$ ,  $L_1[p] \leftarrow L_1[p] + L_1[v]$ 
9: for each non-root node  $v$  in pre-order do
10:   $p \leftarrow$  parent of  $v$ ,  $\tilde{m}_2[v] \leftarrow \tilde{m}_2[p] + r_v L_1[v]$ 
11:   $\mu_2[v] \leftarrow 2\tilde{m}_2[v] - m_1[v]^2$ 
12:   $\kappa[v] \leftarrow \mu_2[v]/m_1[v]^2$ ,  $\eta_\sigma[v] \leftarrow T_{in}/\sqrt{\mu_2[v]}$ 
13:   $\hat{\mathcal{R}}[v] \leftarrow \exp\left(-\frac{c_1 \kappa[v]}{\eta_\sigma[v] + c_2}\right)$ 
return  $\hat{\mathcal{R}}$  for all non-root nodes

```

Table 1: Statistics of TAU Benchmark Circuits.

Benchmark	Tech.	#Gates	#Nets	#Pins
aes_core	14 nm	22,938	23,199	66,221
vga_lcd_iccad	14 nm	259,067	259,152	662,179
b19_iccad	14 nm	255,278	255,300	776,320
cordic_ispd	14 nm	45,359	45,393	127,993
des_perf_ispd	14 nm	138,878	139,112	371,587
edit_dist_ispd	14 nm	147,650	150,212	416,609
fft_ispd	14 nm	38,158	39,184	116,139
leon2_iccad	14 nm	1,616,369	1,616,984	4,178,874
leon3mp_iccad	14 nm	1,247,725	1,247,979	3,267,993
netcard_iccad	14 nm	1,496,719	1,498,555	3,901,343
matrix_mult_ispd	14 nm	164,040	167,242	475,186
mge_edit_dist_iccad	14 nm	161,692	164,254	444,693
mge_matrix_mult_iccad	14 nm	171,282	174,484	489,670
pai_bridge32_ispd	14 nm	40,790	40,950	108,172
tau2015_cordic_core	14 nm	3,585	3,665	9,670
tau2015_crc32d16N	14 nm	478	495	1,251
tau2015_softusb_navre	14 nm	6,943	6,978	19,546
tau2015_tip_master	14 nm	37,715	38,493	95,524
usb_phy_ispd	14 nm	923	938	2,447

where k_s is the slew coefficient. RAPID replaces D2M’s empirical delay-ratio scaling [1] with its derived correction; applying the ratio as a temporal scaling multiplies the variance term by $\hat{\mathcal{R}}^2$:

$$T_{out} = \sqrt{T_{in}^2 + 2\pi\hat{\mathcal{R}}^2 \cdot \mu_2}. \quad (60)$$

This places delay and slew on a consistent temporal scale.

4 Experimental Results

We implement RAPID in HeteroSTA [14] and evaluate it on a server with two Intel Xeon 8358 CPUs, one NVIDIA A800 GPU, and 1007 GiB of memory. The evaluation uses 19 TAU benchmarks in 14 nm [17] and four ARM intellectual property (IP) designs in 14 nm and 7 nm [7] before and after design-rule violation (DRV) repair (Tables 1–2).

Table 2: Statistics of ARM IP Circuits.

Design	Stage	Tech.	#Gates	#Nets	#Pins
ARM_IP1	pre-fixDRV	14 nm	443,163	469,895	1,675,088
	post-fixDRV	14 nm	447,997	475,399	1,684,756
ARM_IP2	pre-fixDRV	7 nm	380,973	409,862	1,538,039
	post-fixDRV	7 nm	384,115	413,495	1,544,323
ARM_IP3	pre-fixDRV	7 nm	499,676	588,794	2,063,587
	post-fixDRV	7 nm	502,403	591,665	2,069,041
ARM_IP4	pre-fixDRV	7 nm	378,569	396,522	1,505,790
	post-fixDRV	7 nm	380,099	398,084	1,508,850

4.1 Standalone Timer Performance

PT-NLDM is the accuracy reference; PT, OS, and GS denote PrimeTime [40], OpenSTA [33], and GPU-enabled HeteroSTA. GS variants share one engine, STA pipeline, and hardware; suffixes identify interconnect calculators. GS-Arnoldi implements coordinate-transformed Arnoldi MOR with effective-capacitance delay and slew evaluation [13, 39]; it and OS-Arnoldi are timer-integrated higher-order baselines.

PrimeTime¹ and OpenSTA² each use 16 CPU threads; GS uses 16 CPU threads and one GPU. GS-ML reimplements PathLearn’s Transformer predictor with the FE2 feature set (Elmore, D2M, ECM, and modified D2M (MD2M) descriptors) [30]. Because PathLearn predicts only net delay, GS-ML uses iSTA’s implementation in iEDA to compute cell delay and slew [23, 24]. GS-ML uses two models trained separately on TAU and ARM post-fixDRV circuits. In contrast, we select RAPID’s slew-scale parameter c_2 on TAU and use the same coefficient pair for all ARM designs. Runtime averages three end-to-end `report_timing` runs; R^2 and MAE use paired delays for all reported timing arcs against PT-NLDM. Cross-tool speedups compare complete timer implementations.

TAU Benchmarks. Table 3 shows implementation-level speedups of 9.84 $\times$ over PT-NLDM and 32.70 $\times$ over OS-Arnoldi, while GS-RAPID remains within 6.3% of GS-Elmore runtime. Relative to GS-Elmore, average R^2 rises from 0.8873 to 0.9905 and MAE falls from 10.278 to 2.079 ps; GS-RAPID also achieves higher average R^2 and lower average MAE than GS-Arnoldi and GS-ML. The correction therefore improves agreement with PT-NLDM in the GS-Elmore runtime regime.

ARM IPs. On the ARM IPs (Table 4), GS-RAPID achieves the best average R^2 (0.9949) and MAE (0.526 ps) among approximate timers at GS-Elmore-level runtime. GS-Arnoldi and OS-Arnoldi require 1.50 $\times$ and 11.65 $\times$ its runtime, respectively. These results demonstrate the superior fidelity–runtime tradeoff on the 14 nm and 7 nm ARM designs.

4.2 Robustness Study on Synthetic Net Families

We generate seven 256-net stress levels of increasing net complexity through wider RC time-constant spread, higher root fanout, and more concentrated far-end loading. GS-RAPID maintains near-unit R^2 and, at the highest stress, retains higher R^2 and lower MAE than

¹For PrimeTime, the commands `set rc_driver_model_mode basic` and `set rc_receiver_model_mode basic` were used.

²For OpenSTA, `set_delay_calculator arnoldi` was used.

Table 3: End-to-end report_timing accuracy and runtime on TAU; PT-NLDM is the reference.

Benchmark	PT-NLDM RT	OS-Arnoldi			GS-Arnoldi			GS-Elmore			GS-ML			GS-RAPID		
		R^2	MAE	RT	R^2	MAE	RT	R^2	MAE	RT	R^2	MAE	RT	R^2	MAE	RT
aes_core	828	0.9955	0.178	1283	0.9902	0.233	432	0.9915	0.187	201	0.9893	0.198	555	0.9922	0.178	201
vga_lcd_iccad	6665	0.9883	6.404	18222	0.9875	6.545	2792	0.8271	22.133	867	0.9860	9.007	4264	0.9850	4.620	907
b19_iccad	8261	0.9777	9.863	16271	0.9781	9.582	3553	0.8311	28.272	844	0.9791	12.600	4903	0.9867	5.773	1034
cordic_ispd	1766	0.9833	0.719	3082	0.9827	0.810	2669	0.9269	1.480	326	0.9622	1.228	1037	0.9922	0.565	314
des_perf_ispd	2709	0.9802	0.673	14408	0.9787	0.775	1148	0.9110	1.334	431	0.9635	1.050	2475	0.9949	0.428	485
edit_dist_ispd	4894	0.9788	1.048	12438	0.9772	1.176	2030	0.8671	3.274	602	0.9159	2.827	2744	0.9897	0.892	717
fft_ispd	1134	0.9798	0.685	3682	0.9790	0.700	1012	0.9312	1.359	246	0.9606	1.118	928	0.9958	0.472	264
leon2_iccad	78080	0.9639	5.998	216964	0.9642	5.792	8152	0.8365	15.187	5592	0.9752	7.390	26628	0.9923	2.763	5771
leon3mp_iccad	50421	0.9766	9.682	81029	0.9741	10.028	7377	0.8495	33.003	4087	0.9810	12.361	20791	0.9862	5.753	4506
netcard_iccad	42870	0.9846	8.919	281866	0.9792	9.628	7030	0.8190	33.289	4736	0.9840	12.605	24810	0.9853	6.380	4924
matrix_mult_ispd	3631	0.9833	0.906	15772	0.9772	1.004	2290	0.8642	2.501	609	0.9524	1.883	3171	0.9951	0.656	670
mgc_edit_dist_iccad	6356	0.9245	3.543	15638	0.9236	3.774	3109	0.8861	5.970	564	0.9237	4.920	2961	0.9849	2.130	568
mgc_matrix_mult_iccad	5795	0.9336	2.695	34487	0.9284	3.004	3545	0.8671	5.046	632	0.8959	4.472	3220	0.9801	1.840	616
pci_bridge32_ispd	1200	0.9924	1.364	2456	0.9928	1.391	635	0.8999	4.498	218	0.9690	2.780	829	0.9948	0.918	229
tau2015_cordic_core	369	0.9676	1.950	419	0.9684	1.940	647	0.9199	3.653	148	0.9717	2.374	220	0.9931	1.181	142
tau2015_crc32d16N	183	0.9822	0.680	124	0.9760	1.012	175	0.9531	1.544	94	0.9613	1.384	115	0.9912	0.627	95
tau2015_softusb_navre	387	0.9605	1.333	640	0.9590	1.479	900	0.9178	2.238	147	0.9407	1.888	286	0.9874	0.865	154
tau2015_tip_master	1103	0.9817	7.356	1787	0.9565	11.028	4730	0.8248	29.019	279	0.9847	9.274	820	0.9985	3.050	342
usb_phy_ispd	222	0.9857	0.614	105	0.9824	0.730	215	0.9358	1.300	118	0.9577	1.073	116	0.9946	0.416	101
Average	11414	0.9747	3.401	37930	0.9713	3.717	2760	0.8873	10.278	1092	0.9607	4.760	5309	0.9905	2.079	1160
Avg. Ratio	9.84	0.9841	1.635	32.70	0.9806	1.788	2.38	0.8958	4.943	0.94	0.9699	2.289	4.58	1.0000	1.000	1.00

PT: PrimeTime; OS: OpenSTA; GS: GPU STA; suffix: calculator. MAE: ps; RT: end-to-end runtime (ms); Ratio: column average divided by GS-RAPID average; bold: best.

Table 4: End-to-end report_timing accuracy and runtime on unseen ARM IPs; PT-NLDM is the reference.

Design	Stage	PT-NLDM RT	OS-Arnoldi			GS-Arnoldi			GS-Elmore			GS-ML			GS-RAPID		
			R^2	MAE	RT	R^2	MAE	RT	R^2	MAE	RT	R^2	MAE	RT	R^2	MAE	RT
ARM_IP1	pre-fixDRV	11782	0.9878	0.658	45905	0.9935	0.693	5595	0.9580	0.967	3741	0.9632	0.892	12158	0.9928	0.545	3842
	post-fixDRV	11635	0.9940	0.632	43367	0.9939	0.696	5145	0.9894	0.831	3472	0.9893	0.906	11829	0.9977	0.548	3224
ARM_IP2	pre-fixDRV	10554	0.9820	0.697	35476	0.9910	0.735	6148	0.9595	0.890	3622	0.9641	0.862	11730	0.9935	0.552	3607
	post-fixDRV	10622	0.9890	0.718	36069	0.9896	0.712	5347	0.9820	0.893	3382	0.9825	0.997	11079	0.9964	0.580	3198
ARM_IP3	pre-fixDRV	14926	0.9662	0.754	59526	0.9886	0.733	6286	0.9599	0.736	4588	0.9559	0.765	15105	0.9884	0.572	5063
	post-fixDRV	14025	0.9843	0.709	53065	0.9895	0.688	5768	0.9886	0.696	3952	0.9874	0.828	14683	0.9970	0.537	3892
ARM_IP4	pre-fixDRV	10589	0.9733	0.522	34863	0.9934	0.578	4803	0.9607	0.637	3790	0.9728	0.606	11487	0.9955	0.421	3502
	post-fixDRV	18746	0.9935	0.523	40255	0.9942	0.531	5745	0.9893	0.652	3649	0.9891	0.763	11683	0.9979	0.457	3591
Average	-	12860	0.9838	0.652	43566	0.9917	0.671	5605	0.9734	0.788	3775	0.9755	0.827	12469	0.9949	0.526	3740
Avg. Ratio	-	3.44	0.9888	1.238	11.65	0.9968	1.274	1.50	0.9784	1.496	1.01	0.9805	1.571	3.33	1.0000	1.000	1.00

PT: PrimeTime; OS: OpenSTA; GS: GPU STA; suffix: calculator. MAE: ps; RT: end-to-end runtime (ms); Ratio: column average divided by GS-RAPID average; bold: best.

both Arnoldi baselines while its MAE grows far less than GS-Elmore (Fig. 4). These trends show that the anchored correction remains accurate over this stress range.

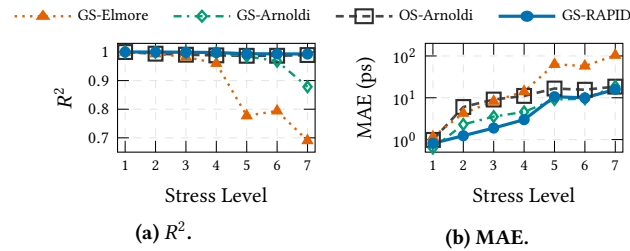


Figure 4: Accuracy over seven 256-net RC-topology stress levels. OS: OpenSTA; GS: GPU STA engine.

4.3 Improvement on Timing Optimization

We run two deterministic repair experiments in an in-house physical-design tool following OpenROAD’s repair_timing methodology [2, 3]. Both use the same initial placement, candidate generation, settings, termination criteria, and cleanup stages and execute DRV repair, TNS-driven buffering, and gate sizing; only the

Table 5: Post-route PrimeTime quality of results (QoR) after GS-Elmore- and GS-RAPID-guided repair (bold: best).

Design	WNS (ns)			TNS (ns)		
	GS-Elmore	GS-RAPID	Improv.	GS-Elmore	GS-RAPID	Improv.
ARM_IP1	-0.3091	-0.2946	4.69%	-2422.5573	-2048.4690	15.44%
ARM_IP2	-0.3384	-0.2963	12.44%	-1112.6407	-996.5298	10.44%
ARM_IP3	-0.1535	-0.1193	22.28%	-19.8934	-18.1515	8.76%
ARM_IP4	-0.1371	-0.1371	0.00%	-217.1207	-201.8370	7.04%

GS-Elmore or GS-RAPID calculator changes. PrimeTime evaluates final WNS and TNS on post-route Standard Parasitic Exchange Format (SPEF) data.

Across four ARM IPs, PrimeTime reports up to 22.28% WNS gains and 7.04–15.44% reductions in TNS magnitude, demonstrating the value of more accurate proxy timing for optimization decisions in this repair flow (Table 5).

5 Conclusion

RAPID moves Elmore delay toward higher-order timing fidelity with an analytically derived, slew-aware delay-ratio correction and an Elmore-isomorphic traversal. Across TAU and unseen ARM designs, it matches GS-Elmore runtime while achieving higher average R^2 and lower average MAE than both Arnoldi baselines.

In the evaluated repair flow, RAPID-guided repair improves or matches PrimeTime WNS and reduces TNS magnitude on all four ARM designs.

The current implementation couples RAPID to NLDL cell timing on RC trees without crosstalk modeling. Future work will investigate extending RAPID to composite current source (CCS) timing and multi-corner, multi-mode (MCMM) analysis.

Acknowledgments

This work is supported in part by Jiangsu National Science and Technology Major Project (BG2025014) and Beijing High Innovation Plan (202604841450).

References

- [1] K. Agarwal, D. Sylvester, and D. Blaauw. 2004. A Simple Metric for Slew Rate of RC Circuits Based on Two Circuit Moments. *IEEE TCAD* 23, 9 (2004), 1346–1354.
- [2] T. Ajayi, D. Blaauw, T.-B. Chan, C.-K. Cheng, V. A. Chhabria, Kyojin Choo, M. Colletta, S. Dobre, R. Dreslinski, M. Fogaça, et al. 2019. OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain. In *GOMACTech*. 1105–1110.
- [3] Tutu Ajayi, Vidya A. Chhabria, Mateus Fogaça, Soheil Hashemi, Abdelrahman Hosny, Andrew B. Kahng, Minsoo Kim, Jeongsup Lee, Uday Mallappa, Marina Neseem, et al. 2019. Toward an open-source digital flow: First learnings from the OpenROAD project. In *DAC*. 1–4.
- [4] Charles J. Alpert, Anirudh Devgan, and Chandramouli Kashyap. 2000. A two moment RC delay metric for performance optimization. In *ISPD*. 69–74.
- [5] C. J. Alpert, A. Devgan, and C. V. Kashyap. 2001. RC delay metrics for performance optimization. *IEEE TCAD* 20, 5 (2001), 571–582.
- [6] Charles J. Alpert, Frank Liu, Chandramouli Kashyap, and Anirudh Devgan. 2003. Delay and slew metrics using the lognormal distribution. In *DAC*. 382–385.
- [7] Arm Limited. [n. d.]. Arm University Program. <https://www.arm.com>.
- [8] Jayaram Bhasker and Rakesh Chadha. 2009. *Static timing analysis for nanometer designs: A practical approach*. Springer Science & Business Media.
- [9] F. Dartu, N. Menezes, and L. T. Pileggi. 1996. Performance computation for precharacterized CMOS gates with RC loads. *IEEE TCAD* 15, 5 (1996), 544–553.
- [10] W. C. Elmore. 1948. The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers. *Journal of Applied Physics (JAP)* 19, 1 (1948), 55–63.
- [11] Dimitrios Garyfallou, Stavros Simoglou, Nikolaos Sketopoulos, Charalampos Antoniadis, Christos P. Sotiriou, Nestor Evmorfopoulos, and George Stamoulis. 2021. Gate Delay Estimation With Library Compatible Current Source Models and Effective Capacitance. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29, 5 (2021), 962–972.
- [12] Dimitrios Garyfallou, Anastasis Vagenas, Charalampos Antoniadis, Yehia Masoud, and George I. Stamoulis. 2022. Leveraging Machine Learning for Gate-level Timing Estimation Using Current Source Models and Effective Capacitance. In *ACM Great Lakes Symposium on VLSI (GLSVLSI)*. 77–83.
- [13] Zizheng Guo, Tsung-Wei Huang, Zhou Jin, Cheng Zhuo, Yibo Lin, Runsheng Wang, and Ru Huang. 2024. Heterogeneous Static Timing Analysis with Advanced Delay Calculator. In *DATE*. 1–6.
- [14] Zizheng Guo, Haichuan Liu, Xizhe Shi, Shenglu Hua, Zuodong Zhang, Chunyuan Zhao, Runsheng Wang, and Yibo Lin. 2026. HeteroSTA: A CPU-GPU Heterogeneous Static Timing Analysis Engine with Holistic Industrial Design Support. In *ASPDAC*.
- [15] Rohini Gupta, Byron Krauter, Bogdan Tutuianu, John Willis, and Lawrence T. Pileggi. 1995. The Elmore Delay as a Bound for RC Trees with Generalized Input Signals. In *DAC*. ACM/IEEE, 364–369.
- [16] Xu He, Zhiyong Fu, Yao Wang, Chang Liu, and Yang Guo. 2022. Accurate timing prediction at placement stage with look-ahead RC network. In *DAC*. 1213–1218.
- [17] Jin Hu, Greg Schaeffer, and Vibhor Garg. 2015. TAU 2015 contest on incremental timing analysis. In *ICCAD*. IEEE, 882–889.
- [18] C. V. Kashyap, C. J. Alpert, and A. Devgan. 2000. An “effective” capacitance based delay metric for RC interconnect. In *ICCAD*. 229–234.
- [19] Chandramouli V. Kashyap, Charles J. Alpert, Frank Liu, and Anirudh Devgan. 2003. Closed Form Expressions for Extending Step Delay and Slew Metrics to Ramp Inputs. In *ISPD*. 24–31.
- [20] Rony Kay and Lawrence Pileggi. 1998. PRIMO: Probability Interpretation of Moments for Delay Calculation. In *DAC*. 463–468.
- [21] Myung-Chul Kim, Jin Hu, Jiajia Li, and Natarajan Viswanathan. 2015. ICCAD-2015 CAD contest in incremental timing-driven placement and benchmark suite. In *ICCAD*. IEEE, 921–926.
- [22] Shih-Kai Lee, Pei-Yu Lee, and Iris Hui-Ru Jiang. 2025. Multi-Stage CSM Timing Waveform Propagation Accelerated by NLDL Assistance. In *ISPD*. 83–90.
- [23] Xingquan Li, Zengrong Huang, Simin Tao, Zhipeng Huang, Chunan Zhuang, Hao Wang, Yifan Li, Yihang Qiu, Guojie Luo, Huawei Li, Haihua Shen, Mingyu Chen, Dongbo Bu, Wenxing Zhu, Ye Cai, Xiaoming Xiong, Ying Jiang, Yi Heng, Peng Zhang, Bei Yu, Biwei Xie, and Yungang Bao. 2024. iEDA: An Open-source infrastructure of EDA. In *ASPDAC*. IEEE, 77–82.
- [24] Xingquan Li, Simin Tao, Shijian Chen, Zhisheng Zeng, Zhipeng Huang, Hongxi Wu, Weiguo Li, Zengrong Huang, Liwei Ni, Xueyan Zhao, He Liu, Shuaiying Long, Ruizhi Liu, Xiaozhe Lin, Bo Yang, Fuxing Huang, Zonglin Yang, Yihang Qiu, Zheqing Shao, Jikang Liu, Yuyao Liang, Biwei Xie, Yungang Bao, and Bei Yu. 2024. iPD: An Open-source intelligent Physical Design Toolchain. In *ASPDAC*. IEEE, 83–88.
- [25] Rongjian Liang, Anthony Agnesina, Wen-Hao Liu, Matt Liberty, Hsin-Tzu Chang, and Haoxing Ren. 2025. ISPD 2025 performance-driven large scale global routing contest. In *ISPD*. 252–256.
- [26] Shiju Lin, Guannan Guo, Tsung-Wei Huang, Weihua Sheng, Evangeline F. Y. Young, and Martin D. F. Wong. 2024. GCS-Timer: GPU-Accelerated Current Source Model Based Static Timing Analysis. In *DAC*. 71:1–71:6.
- [27] Tao Lin, Emrah Acar, and Lawrence Pileggi. 1998. h-gamma: An RC Delay Metric Based on a Gamma Distribution Approximation of the Homogeneous Response. In *ICCAD*. 19–25.
- [28] F. Liu, C. Kashyap, and C. J. Alpert. 2002. A delay metric for RC circuits based on the Weibull distribution. In *ICCAD*. 620–624.
- [29] Haichuan Liu, Zizheng Guo, Runsheng Wang, and Yibo Lin. 2026. ElmoreCeff: A GPU-Friendly Elmore-Like Delay Calculator with a Closed-Form Effective Capacitance Model. In *International Symposium of Electronics Design Automation (ISED)*. 1–6.
- [30] He Liu, Shengkun Wu, Simin Tao, Biwei Xie, Xingquan Li, and Ge Li. 2023. Accurate Timing Path Delay Learning Using Feature Enhancer with Effective Capacitance. In *International Symposium of Electronics Design Automation (ISED)*. 280–285.
- [31] Zongfeng Ma and Zhong Guan. 2026. Old School Never Die: A Classic Yet Novel Algorithm for Computing RC Current Response in VLSI. In *ASPDAC*.
- [32] A. Odabasioglu, M. Celik, and L. T. Pileggi. 1998. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE TCAD* 17, 8 (1998), 645–654.
- [33] Parallax Software, Inc. [n. d.]. OpenSTA. <https://github.com/The-OpenROAD-Project/OpenSTA>.
- [34] Lawrence T. Pillage, X. Huang, and Ronald A. Rohrer. 1989. AWEsim: Asymptotic Waveform Evaluation for Timing Analysis. In *DAC*. 634–637.
- [35] Lawrence T. Pillage and Ronald A. Rohrer. 1990. Asymptotic waveform evaluation for timing analysis. *IEEE TCAD* 9, 4 (1990), 352–366.
- [36] Ruchir Puri, David S. Kung, and Anthony D. Drumm. 2002. Fast and accurate wire delay estimation for physical synthesis of large ASICs. In *ACM Great Lakes Symposium on VLSI (GLSVLSI)*. 30–36.
- [37] Jessica Qian, Satyamurthy Pullela, and Lawrence Pillage. 1994. Modeling the “Effective Capacitance” for the RC Interconnect of CMOS Gates. *IEEE TCAD* 13, 12 (1994), 1526–1535.
- [38] J. Rubinstein, P. Penfield, and M. A. Horowitz. 1983. Signal Delay in RC Tree Networks. *IEEE TCAD* 2, 3 (1983), 202–211.
- [39] L. Miguel Silveira, M. Kamon, I. Elfadel, and J. White. 1996. A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In *ICCAD*. 288–294. doi:10.1109/ICCAD.1996.569710
- [40] Synopsys. [n. d.]. Synopsys PrimeTime. <http://www.synopsys.com>.
- [41] Anastasis Vagenas, Dimitrios Garyfallou, Nestor Evmorfopoulos, and George Stamoulis. 2024. A learning-based method for performance optimization of timing analysis. In *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. 1–4.
- [42] Xin Wang, Ali Kasnavi, and Harold Levy. 2008. An Efficient Method for Fast Delay and SI Calculation Using Current Source Models. In *IEEE International Symposium on Quality Electronic Design (ISQED)*. 57–61.
- [43] Edwin B. Wilson and Margaret M. Hilferty. 1931. The Distribution of Chi-Square. *Proceedings of the National Academy of Sciences of the United States of America* 17, 12 (1931), 684–688. doi:10.1073/pnas.17.12.684
- [44] Jun-Kuei Zeng and Chung-Ping Chen. 2010. Interconnect delay and slew metrics using the beta distribution. In *DATE*. 1329–1332.