Statistical Analysis of Random Telegraph Noise in Digital Circuits

Xiaoming Chen[1], Yu Wang[1], Yu Cao[2], Huazhong Yang[1]

[1]EE, Tsinghua University, Beijing, China

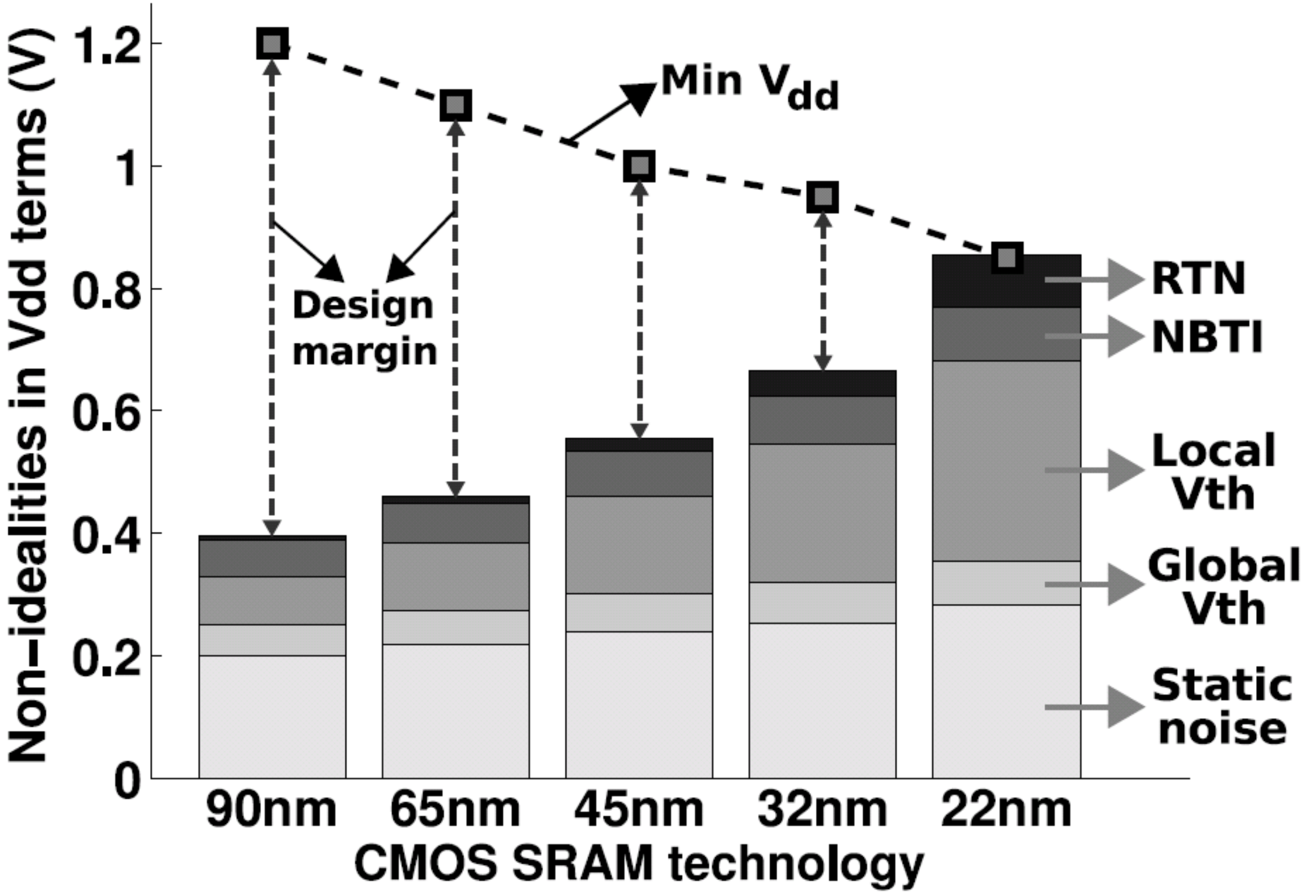
[2]ECEE, Arizona State University, Tempe, AZ, USA

Email: chenxm05@mails.tsinghua.edu.cn

Outline

- ➢ Introduction
- ➢ RTN Modeling
- ➢ Our Method
- ➢ Simulation Results
- ➢ Conclusions

Introduction: reliability problems

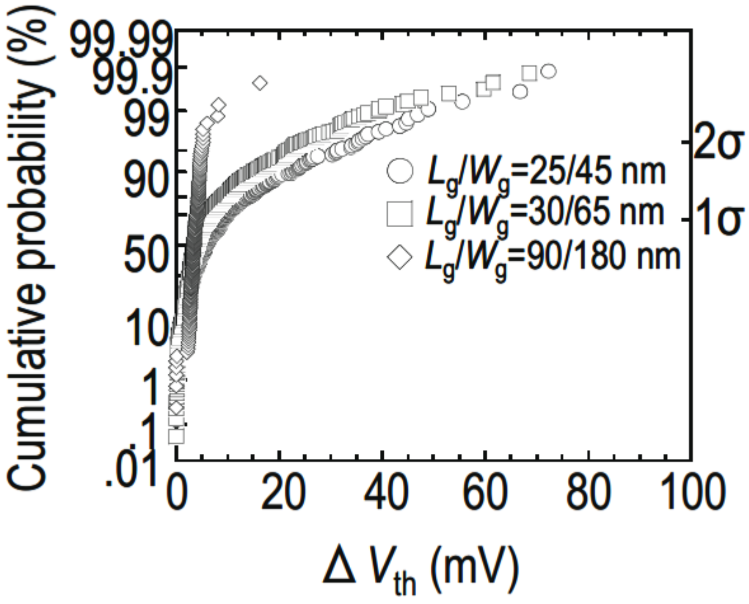


[Aadithya, DAC2011 & Yasumasa Tsukamoto, Renesas Electronics Corporation]

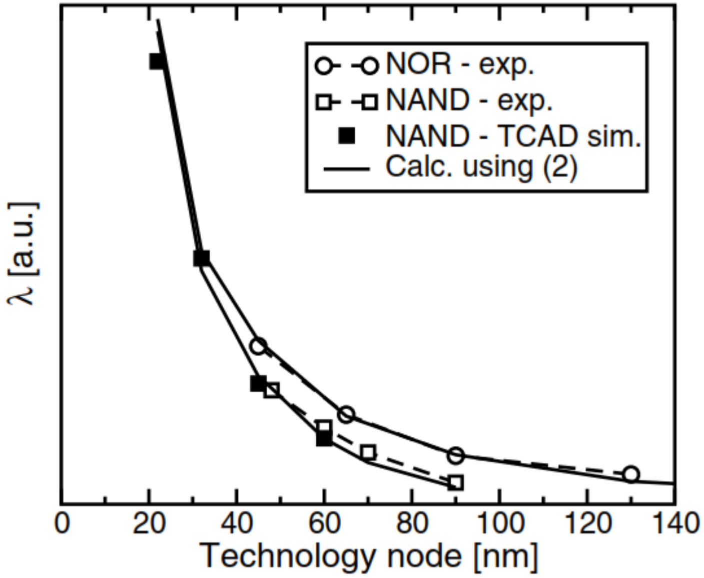
RTN: random telegraph noise, also known as random telegraph signal
NBTI: negative bias temperature instability

Introduction: random telegraph noise

- RTN: a serious reliability mechanism that can increase Vth and Id variations
  - Vth variation can be ~70mV @25nm [Tega, VLSIT'09]
  - RTN increases superlinearly with the scaling down of the technology node [Ghetti, IEDM'08]



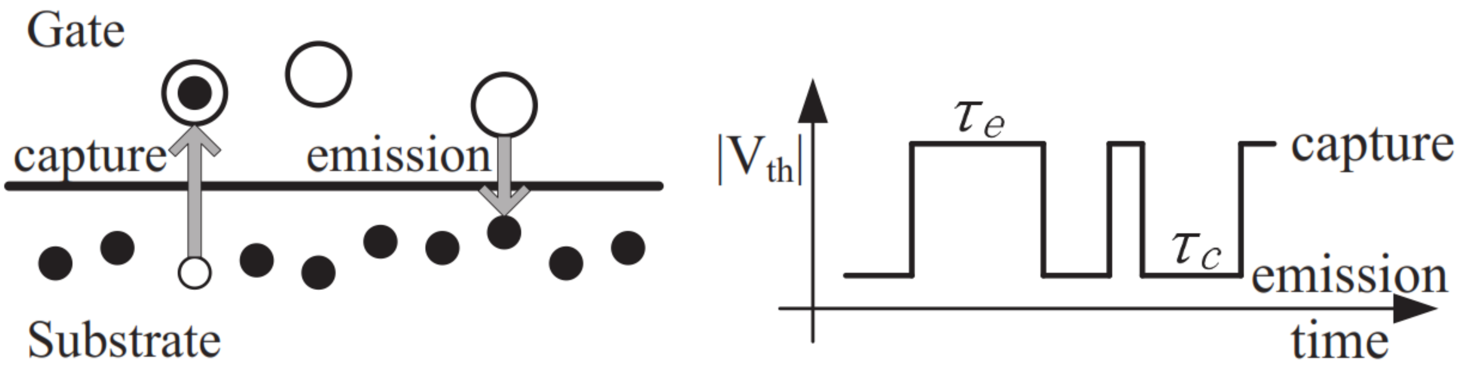
[Tega, VLSIT'09]



[Ghetti, IEDM'08]

Introduction: physics of RTN

- Charge trapping/de-trapping [Campbell, IRW'08]
  - Traps (defects) exist in the gate dielectric
  - A trap can occasionally capture a charge carrier
    - ✓ Vth increases
  - The captured carrier can be emitted back to the channel
    - ✓ Vth decreases
  - The capture/emission process causes random Vth fluctuation



(a) Capture/emission of traps.

(b) V_{th} fluctuation.

Introduction: related work & Motivation

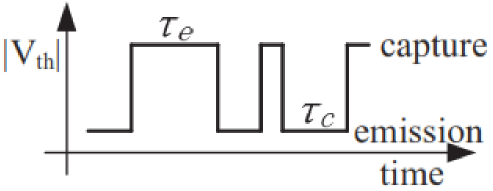
- Physics of RTN
  - [Campbell, IRW'08] [Grasser, IEDM'09]
- RTN evaluation/simulation
  - [Tega, IRPS, 08] [Aadithya, DAC'11 & DATE'11] [Ito, ISQED'11] [Joe, TED'11] [Toh, IEDM'09] …
- Weakness of existing studies on RTN simulation
  - SPICE-based simulation, high time complexity
  - Single-trap RTN
    - ✓ But there are 2-3 detectable traps in each device [Nagumo, IEDM'09]
- Motivation
  - Propose a fast algorithm to evaluate multi-trap RTN

Introduction: our contributions

- We investigated the impact of multiple traps-induced RTN on digital circuits.
- We pointed out that the multi-trap problem can be solved by a statistical static timing analysis (SSTA) method, so our method integrated RTN analysis into timing analysis, to obtain the temporal distribution of circuit delay
- A statistical algorithm was proposed, which is on average 41X faster than Monte-Carlo.

RTN Modeling

- Capture/emission time constants
  - τ_e : emission time, time to be emitted back
  - τ_c : capture time, time to be captured
  - $\tau_c = 0.001s \sim 0.01s$, $\tau_e = 0.1s \sim 10s$ [Campbell, IRW'08]
  - Bias condition dependent time constants [van der Wel, TED'03]



$$\tau_c^{(\mathrm{off})} = \tau_c^{(\mathrm{on})} \times m_c \qquad \tau_e^{(\mathrm{off})} = \frac{\tau_e^{(\mathrm{on})}}{m_e}$$

  - Average time constants. *SP* is the duty cycle

$$\tau_c = SP \times \tau_c^{(\mathrm{on})} + (1 - SP) \times \tau_c^{(\mathrm{off})}$$
$$\tau_e = SP \times \tau_e^{(\mathrm{on})} + (1 - SP) \times \tau_e^{(\mathrm{off})}$$

RTN Modeling

➢ RTN-induced Vth shift

- Single-trap induced Vth shift [Tega, IRPS'08]

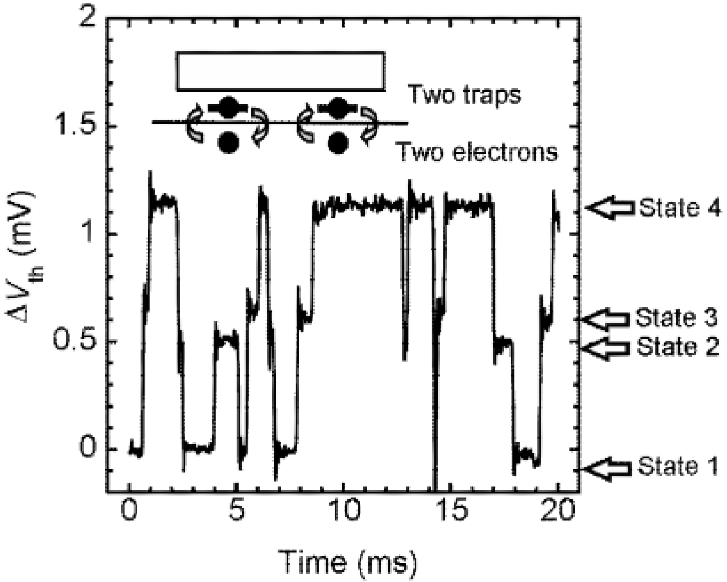
$$\Delta V_{th}^{(\mathrm{S})} = \frac{q}{C_{ox}WL}$$

- Number of (detectable) traps [Nagumo, IEDM'09]

$$N_{trap} \sim Poisson(\lambda)$$

- Multi-trap induced Vth shift, caused by N_{fill} filled traps

$$\Delta V_{th}^{(\mathrm{M})} \approx \Delta V_{th}^{(\mathrm{S})} \times N_{fill}$$



[Tega, IRPS'08]

RTN Modeling

- RTN-induced gate delay shift
  - Each trap can be described by a two-value discrete random variable X: 0<-->empty and 1<-->filled

$$P(X=0)=\frac{\tau_c}{\tau_e+\tau_c}, P(X=1)=\frac{\tau_e}{\tau_e+\tau_c}$$

  - Multi-trap induced gate delay shift

$$\Delta D=\sum_{i=1}^{N_{fill}}\Delta D_S,$$

$$N_{fill} \sim Poisson(r\lambda), r=P(X=1)$$

ΔD_S is the delay shift caused by a single filled trap, λ is the average number of traps

RTN Modeling

- RTN-induced gate delay shift
  - Each trap can be described by a two-value discrete random variable X: 0<-->empty and 1<-->filled

$$P(X=0)=\frac{\tau_c}{\tau_e+\tau_c},\; P(X=1)=\frac{\tau_e}{\tau_e+\tau_c}$$

  - Multi-trap induced gate delay shift

$$\Delta D=\sum_{i=1}^{N_{fill}}\Delta D_S,$$

Compound Poisson distribution

$$N_{fill}\sim Poisson(r\lambda), r=P(X=1)$$

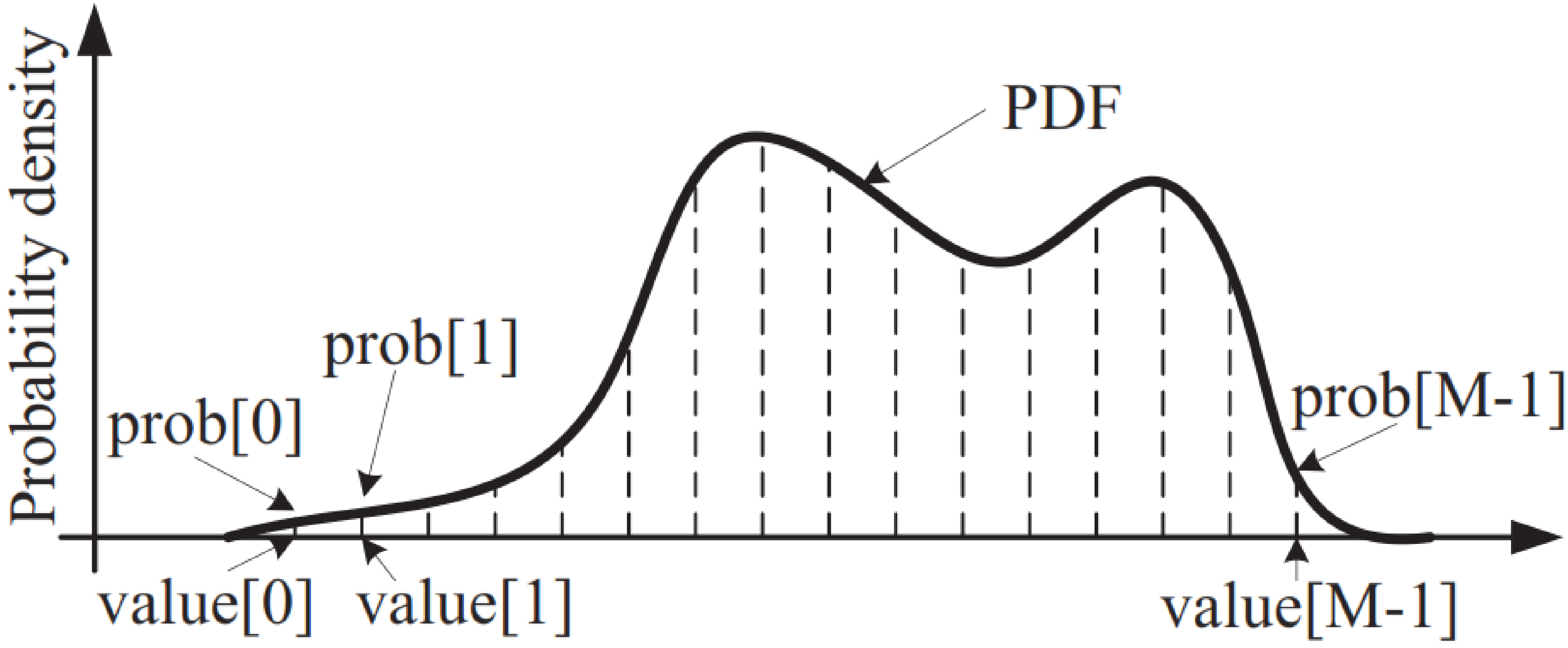
ΔD_S is the delay shift caused by a single filled trap, λ is the average number of traps

Our Method

- General ideas
  - SSTA may be used, because delay of each gate is a distribution
    - ✓ ADD: calculate the delay of a path
    - ✓ MAX: calculate the maximum arrival time of all fan-ins of one gate
  - SSTA faces some difficulties in RTN simulation
    - ✓ In SSTA, a canonical form is usually used to express the delay of all gates
    - ✓ However, MAX of two compound Poisson distributions is not a compound Poisson distribution. More specifically, it has no analytical form

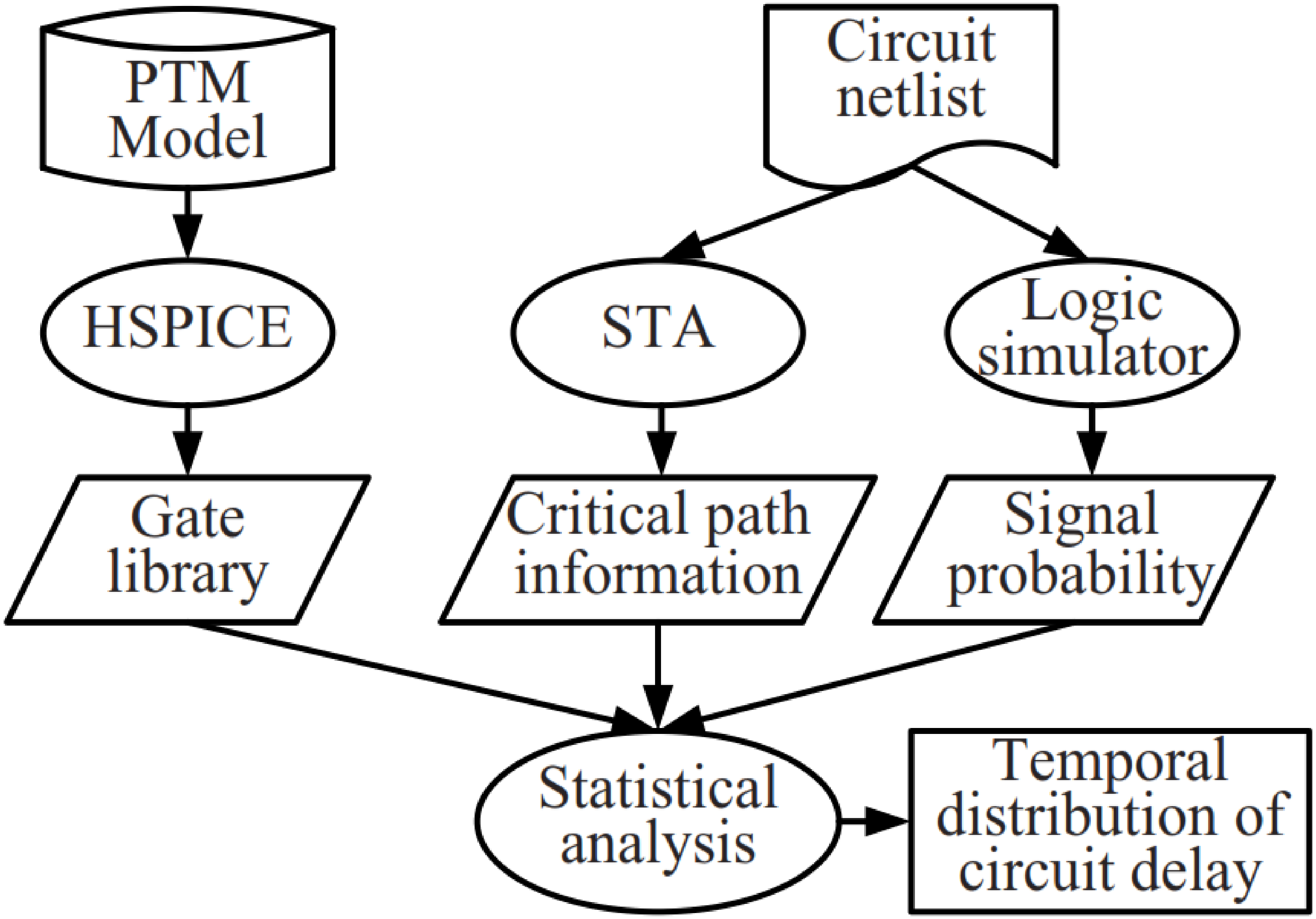
Our Method

- General ideas
  - Represent an arbitrary distribution by sampling the probability density function (PDF), using M nodes



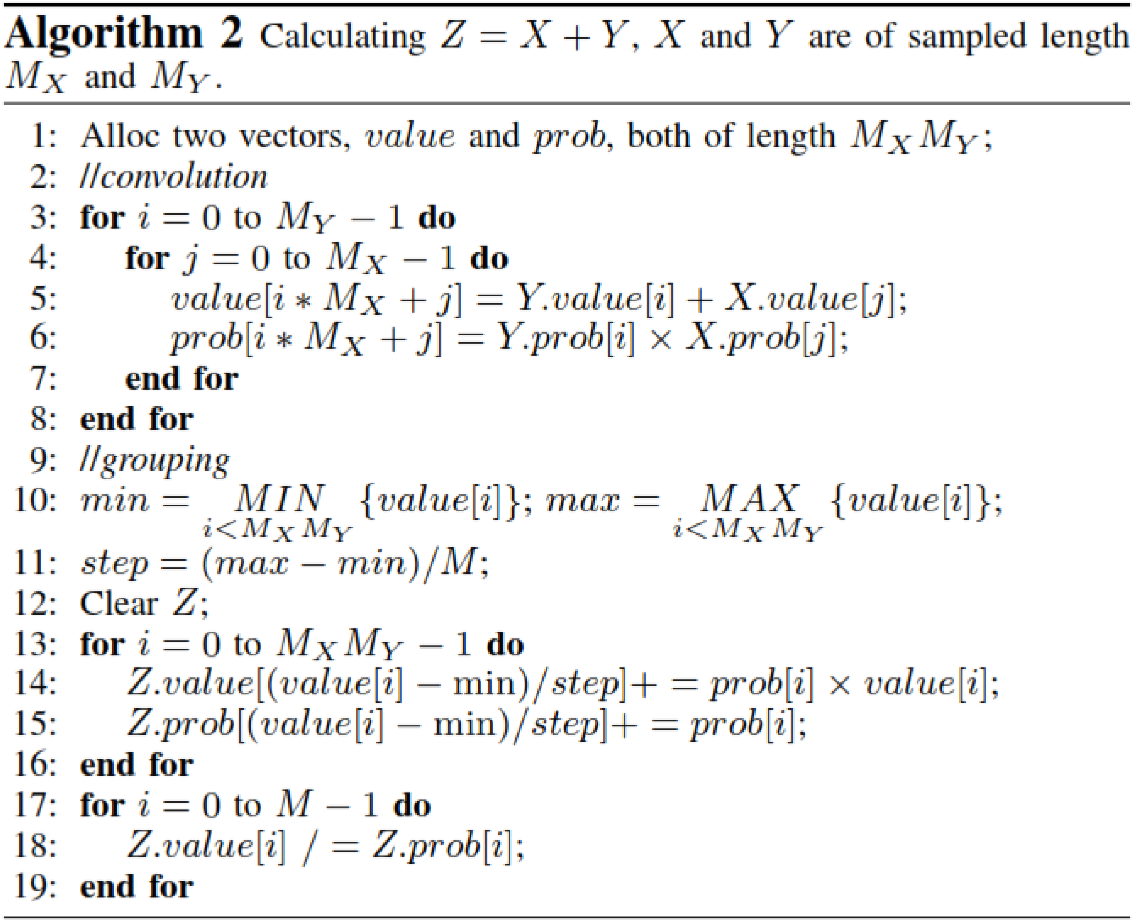
Our Method

- RTN simulation framework



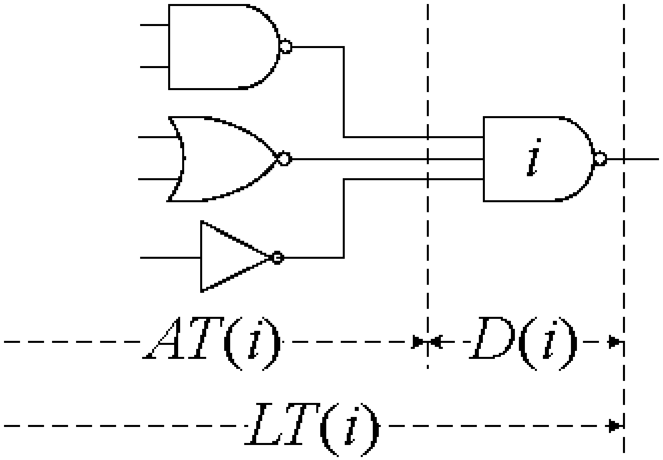
Our Method

- ADD operation: $LT(i) = AT(i) + D(i)$



Algorithm 2 Calculating $Z = X + Y$, X and Y are of sampled length M_X and M_Y.

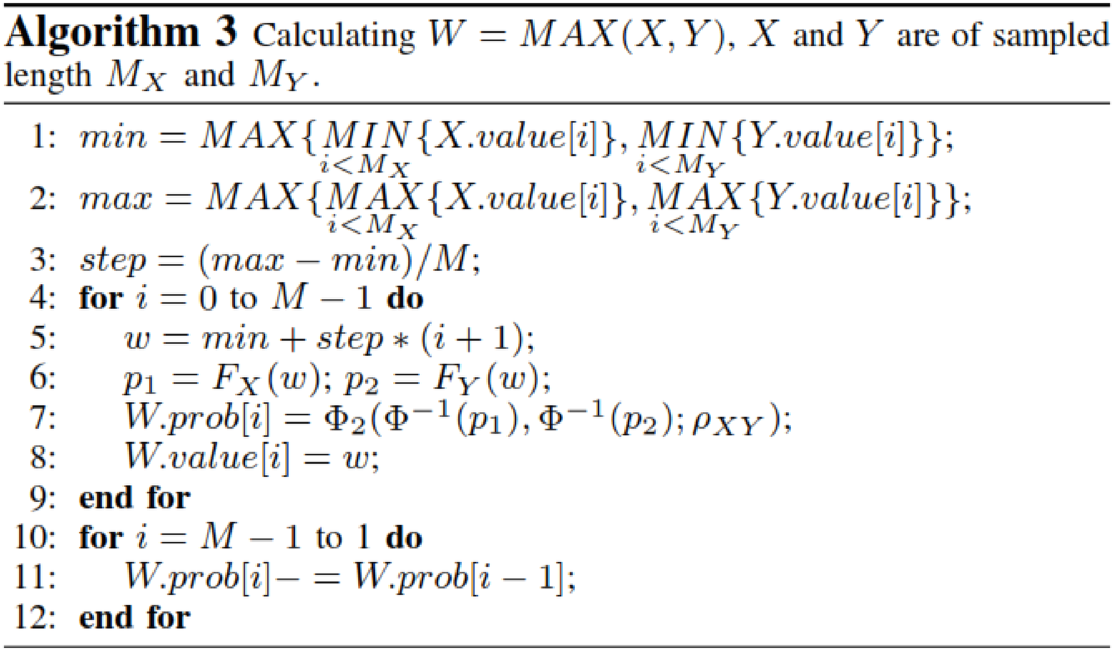
```
Alloc two vectors, $value$ and $prob$, both of length $M_X M_Y$;
//convolution
for $i = 0$ to $M_Y - 1$ do
    for $j = 0$ to $M_X - 1$ do
        $value[i * M_X + j] = Y.value[i] + X.value[j]$;
        $prob[i * M_X + j] = Y.prob[i] \times X.prob[j]$;
    end for
end for
//grouping
$min = \underset{i<M_X M_Y}{MIN}\{value[i]\}$; $max = \underset{i<M_X M_Y}{MAX}\{value[i]\}$;
$step = (max - min)/M$;
Clear $Z$;
for $i = 0$ to $M_X M_Y - 1$ do
    $Z.value[(value[i] - \min)/step] += prob[i] \times value[i]$;
    $Z.prob[(value[i] - \min)/step] += prob[i]$;
end for
for $i = 0$ to $M - 1$ do
    $Z.value[i] \; / = Z.prob[i]$;
end for
```



Our Method

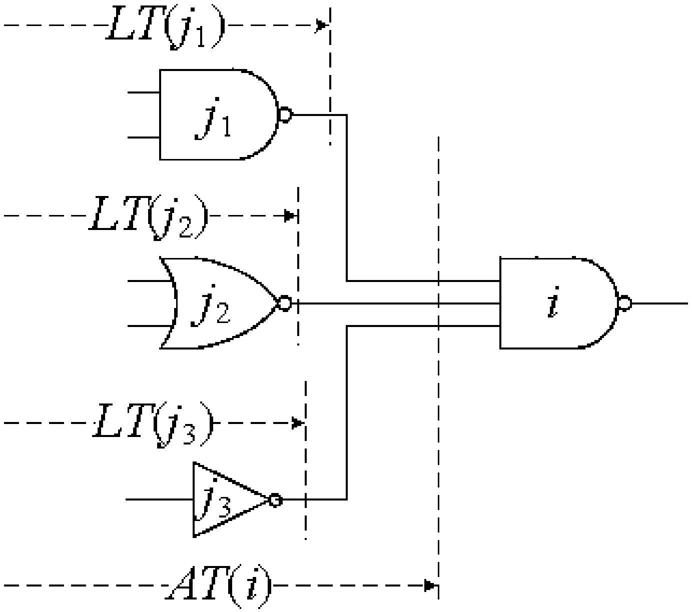
- MAX operation: $AT(i) = \mathrm{MAX}\{LT(j_1), LT(j_2), \cdots\}$

  $j_1, j_2, \cdots$ are the fan-ins of gate i



Algorithm 3 Calculating $W = MAX(X, Y)$, X and Y are of sampled length M_X and M_Y.

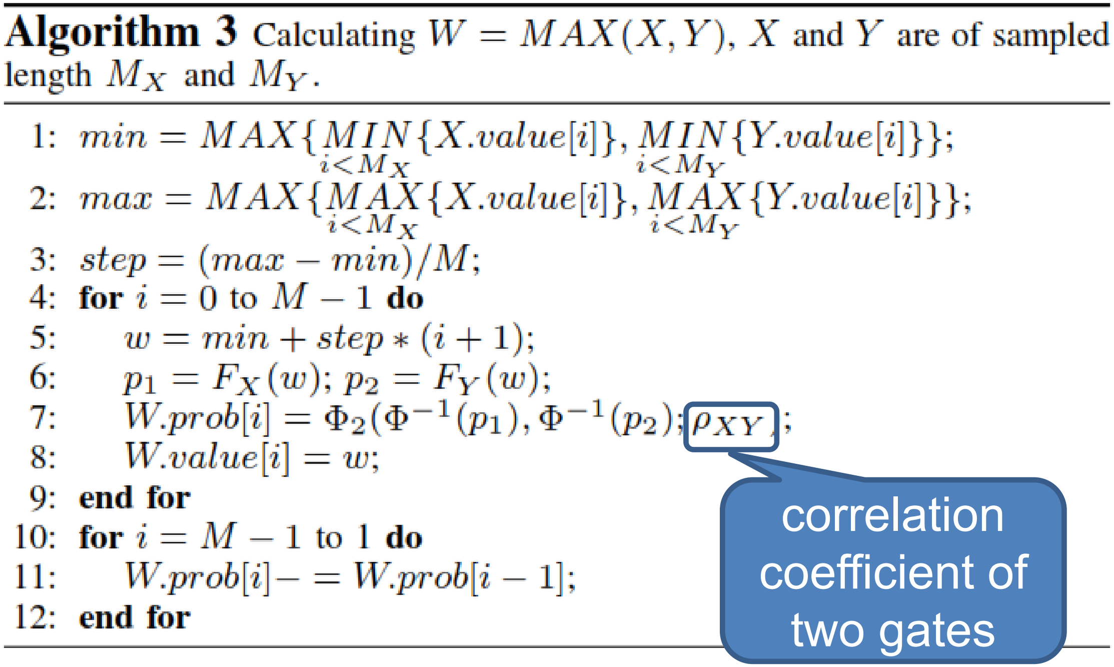
```
$min = MAX\{\underset{i<M_X}{MIN}\{X.value[i]\}, \underset{i<M_Y}{MIN}\{Y.value[i]\}\}$;
$max = MAX\{\underset{i<M_X}{MAX}\{X.value[i]\}, \underset{i<M_Y}{MAX}\{Y.value[i]\}\}$;
$step = (max - min)/M$;
for $i = 0$ to $M - 1$ do
    $w = min + step * (i + 1)$;
    $p_1 = F_X(w)$; $p_2 = F_Y(w)$;
    $W.prob[i] = \Phi_2(\Phi^{-1}(p_1), \Phi^{-1}(p_2); \rho_{XY})$;
    $W.value[i] = w$;
end for
for $i = M - 1$ to $1$ do
    $W.prob[i] -= W.prob[i - 1]$;
end for
```



Our Method

- MAX operation: $AT(i) = \mathrm{MAX}\{LT(j_1), LT(j_2), \cdots\}$

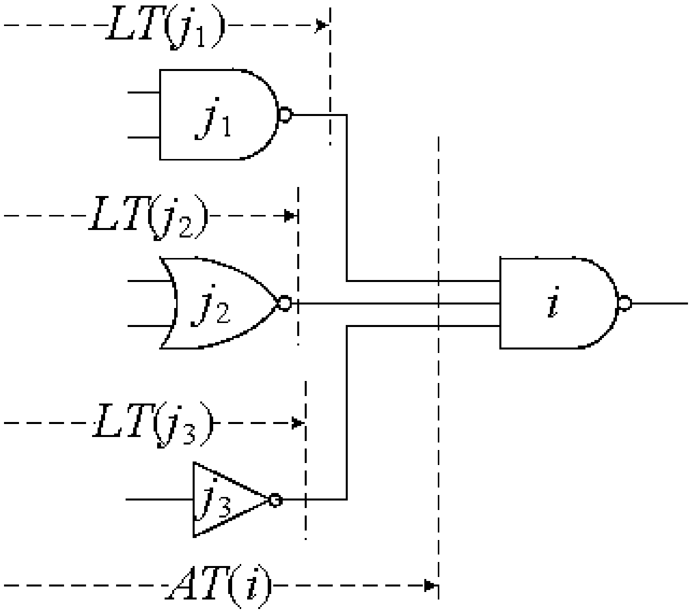
$j_1, j_2, \cdots$ are the fan-ins of gate i



Algorithm 3 Calculating $W = MAX(X, Y)$, X and Y are of sampled length M_X and M_Y.

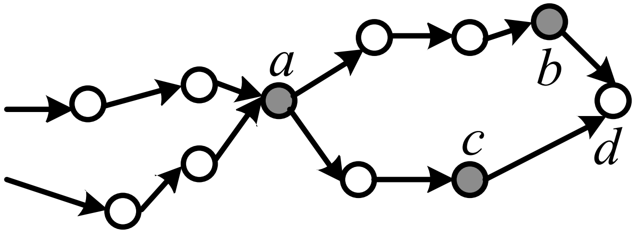
```
min = $MAX\{\underset{i<M_X}{MIN}\{X.value[i]\}, \underset{i<M_Y}{MIN}\{Y.value[i]\}\}$;
max = $MAX\{\underset{i<M_X}{MAX}\{X.value[i]\}, \underset{i<M_Y}{MAX}\{Y.value[i]\}\}$;
step = $(max - min)/M$;
for $i = 0$ to $M - 1$ do
    $w = min + step * (i + 1)$;
    $p_1 = F_X(w)$; $p_2 = F_Y(w)$;
    $W.prob[i] = \Phi_2(\Phi^{-1}(p_1), \Phi^{-1}(p_2); \boxed{\rho_{XY}})$;
    $W.value[i] = w$;
end for
for $i = M - 1$ to 1 do
    $W.prob[i] -= W.prob[i - 1]$;
end for
```

correlation coefficient of two gates



Our Method

- Correlation coefficient of two gates



$$LT(b) = LT(a) + PD(a,b), LT(c) = LT(a) + PD(a,c)$$

$$\rho_{LT(b),LT(c)} = \frac{E(LT(b)LT(c)) - \mu_{LT(b)}\mu_{LT(c)}}{\sigma_{LT(b)}\sigma_{LT(c)}}$$

$$= \ldots\ldots$$

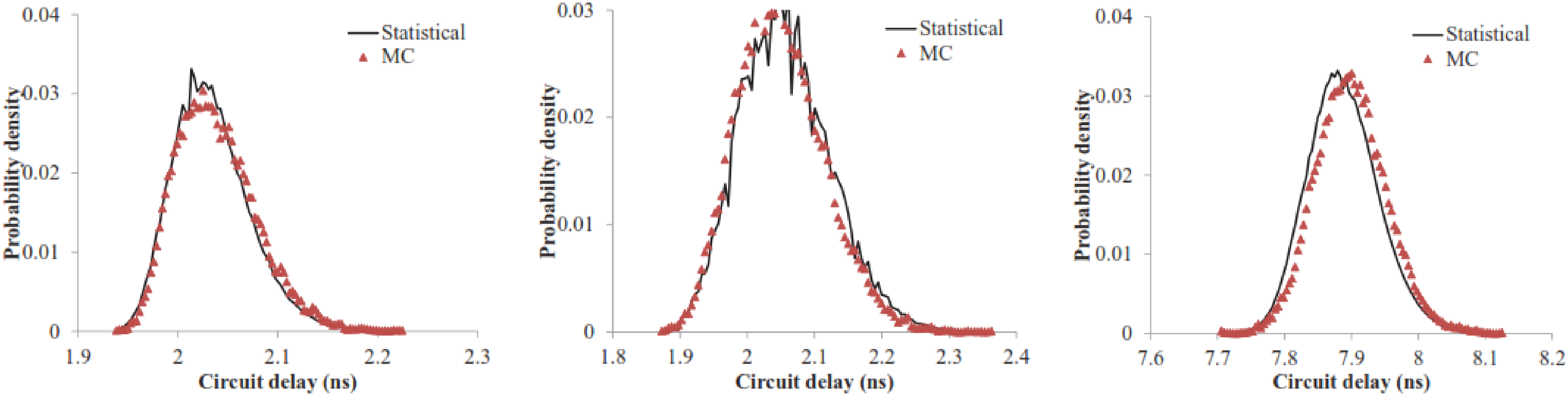
$$= \frac{(\sigma_{LT(a)})^2}{\sigma_{LT(b)}\sigma_{LT(c)}}$$

Simulation Results

- Experimental environment
  - Benchmark: ISCAS85 and some ALU circuits
  - 16nm high-performance PTM model (http://ptm.asu.edu/)
    - ✓ Vdd=0.9V, |Vth|=0.4V
  - Single-trap induced $\Delta V_{th} = 30\,\mathrm{mV}$
  - Algorithms are written in C++

Simulation Results

- ➢ Comparison between our method and Monte-Carlo

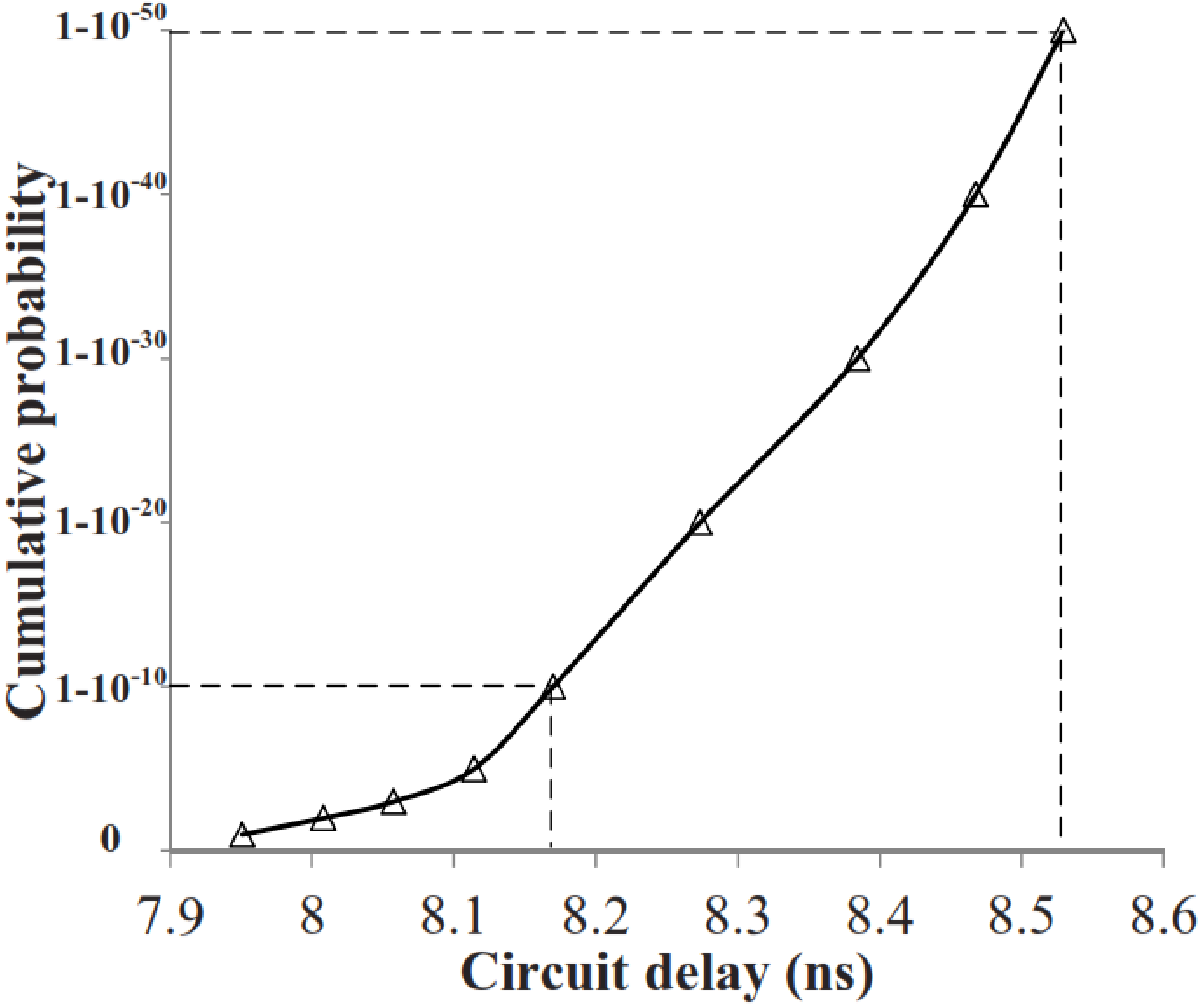


(a) booth9x9 (small size) (b) c7552 (medium size) (c) pmult32x32 (big size)

Fig. 7: Comparison with MC on circuit delay distribution.

Simulation Results

- Long tail of RTN-induced circuit delay distribution

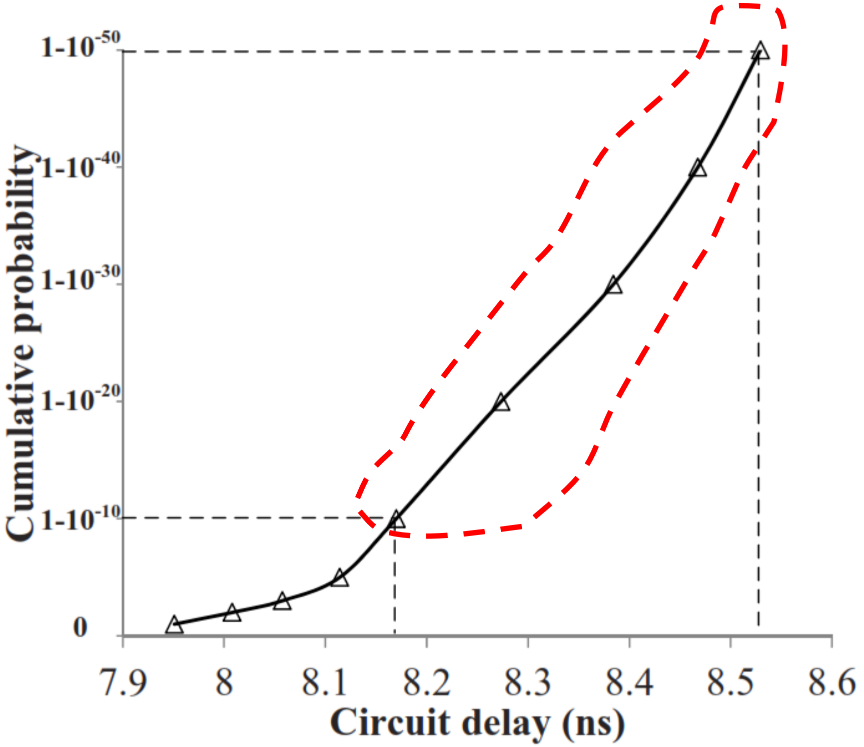


Simulation Results

- RTN-induced circuit delay varies in time randomly
  - If chip operates long enough, circuit delay can always get its worst value
  - For a reliable design, we may ensure that the maximum possible delay does not violates the design specification
    - ✓ large design redundancy and large design overheads

Simulation Results

- Large delays can hardly appear during circuit lifetime
- Ensuring that circuit functions correctly with $1-10^{-10}$ probability or $1-10^{-50}$ probability has no essential difference
  - However, the worst-case delay of the two conditions is quite different

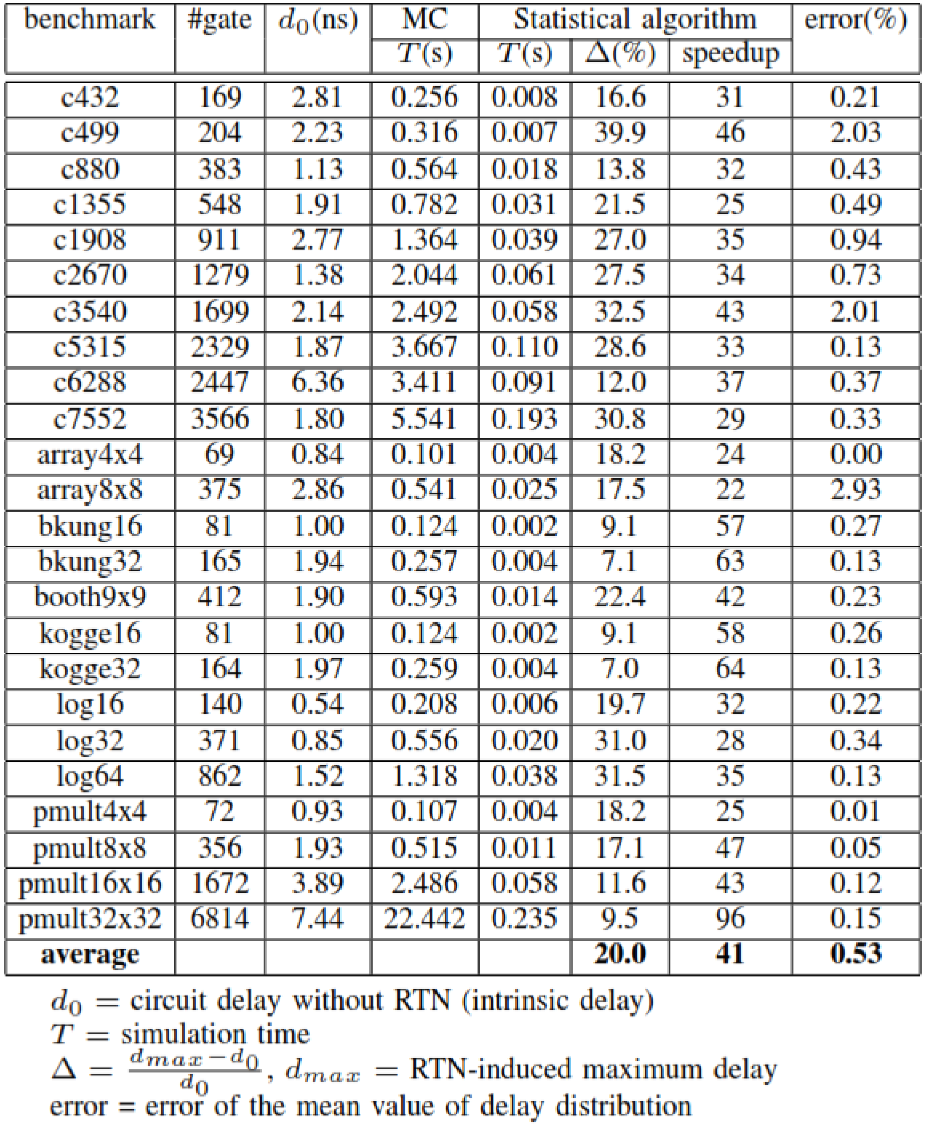


Simulation Results

- It is NOT practical to protect the circuit under the maximum possible delay due to large design redundancy and large design overheads
- We can ensure that circuit functions correctly with a certain probability P≈1 during the whole lifetime, such that timing violations can hardly happen
  - We use $P = 1 - 10^{-9}$, according to circuit lifetime and τ_e, τ_c
  - The corresponding circuit delay is called “RTN-induced maximum delay” ($d_{\max}$)

Simulation Results

➢ Results of all benchmarks



benchmark	#gate	d_0(ns)	MC	Statistical algorithm			error(%)
			T(s)	T(s)	Δ(%)	speedup	
c432	169	2.81	0.256	0.008	16.6	31	0.21
c499	204	2.23	0.316	0.007	39.9	46	2.03
c880	383	1.13	0.564	0.018	13.8	32	0.43
c1355	548	1.91	0.782	0.031	21.5	25	0.49
c1908	911	2.77	1.364	0.039	27.0	35	0.94
c2670	1279	1.38	2.044	0.061	27.5	34	0.73
c3540	1699	2.14	2.492	0.058	32.5	43	2.01
c5315	2329	1.87	3.667	0.110	28.6	33	0.13
c6288	2447	6.36	3.411	0.091	12.0	37	0.37
c7552	3566	1.80	5.541	0.193	30.8	29	0.33
array4x4	69	0.84	0.101	0.004	18.2	24	0.00
array8x8	375	2.86	0.541	0.025	17.5	22	2.93
bkung16	81	1.00	0.124	0.002	9.1	57	0.27
bkung32	165	1.94	0.257	0.004	7.1	63	0.13
booth9x9	412	1.90	0.593	0.014	22.4	42	0.23
kogge16	81	1.00	0.124	0.002	9.1	58	0.26
kogge32	164	1.97	0.259	0.004	7.0	64	0.13
log16	140	0.54	0.208	0.006	19.7	32	0.22
log32	371	0.85	0.556	0.020	31.0	28	0.34
log64	862	1.52	1.318	0.038	31.5	35	0.13
pmult4x4	72	0.93	0.107	0.004	18.2	25	0.01
pmult8x8	356	1.93	0.515	0.011	17.1	47	0.05
pmult16x16	1672	3.89	2.486	0.058	11.6	43	0.12
pmult32x32	6814	7.44	22.442	0.235	9.5	96	0.15
average					**20.0**	**41**	**0.53**

d_0 = circuit delay without RTN (intrinsic delay)
T = simulation time
$\Delta = \frac{d_{max} - d_0}{d_0}$, d_{max} = RTN-induced maximum delay
error = error of the mean value of delay distribution

- RTN causes 7%-40% circuit delay degradation, the average degradation rate is 20%
- Our method is on average 41X faster than 10000-times MC, while the average error is 0.53%

Simulation Results

- RTN depends on Vgs [Nagumo, IEDM'10]
  - The time constants have approximate exponential relations with Vgs

$$\tau_c^{(\mathrm{on})} = \gamma_c \exp(-\theta_c \cdot V_{gs})$$

$$\tau_e^{(\mathrm{on})} = \gamma_e \exp(\theta_e \cdot V_{gs})$$

  - The RTN-induced maximum delay $d_{\max}$ keeps decreasing while Vgs increases
    - ✓ A simple guard-banding method can protect circuit from RTN

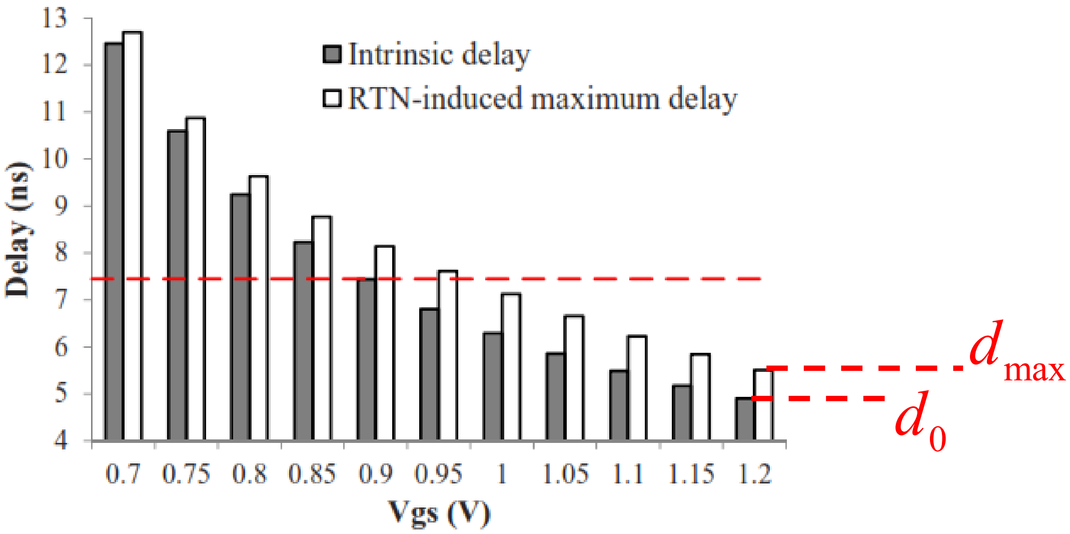


Fig. 9: Intrinsic delay (d_0) and RTN-induced maximum delay (d_{max}), for pmult32x32.

Simulation Results

- RTN depends on Vgs [Nagumo, IEDM'10]

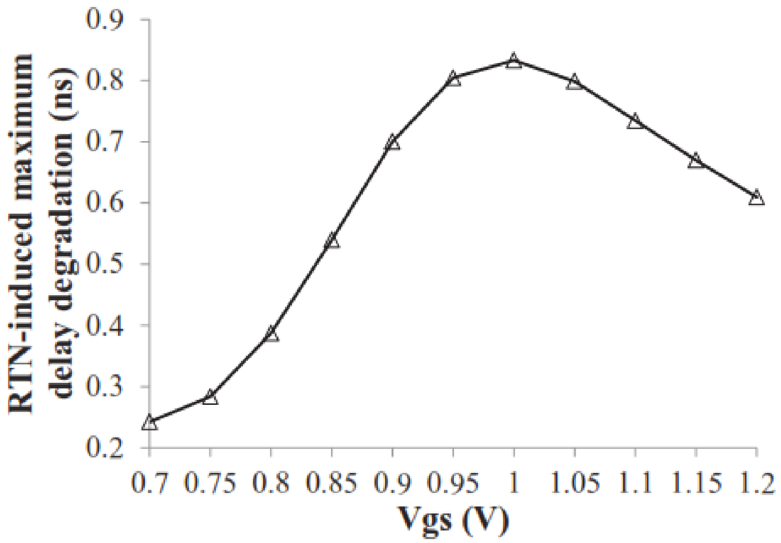


Fig. 10: RTN-induced maximum delay degradation ($d_{max} - d_0$), for pmult32x32.

- Vgs<1V: with Vgs increasing, τ_e increases, τ_c decreases, so the delay degradation increases

- Vgs>1V: $\tau_e >> \tau_c$, all the traps are almost filled (saturated), while the intrinsic delay decreases due to higher Vgs, so the delay degradation decreases

Conclusions

- An SSTA-like algorithm was proposed to study multi-trap RTN. Our method is 41X faster than Monte-Carlo
- Circuit can be protected from a degraded delay which is on average 20% larger than the intrinsic delay to ensure $1-10^{-9}$ correctness during the whole lifetime
- A simple guard-banding approach can effectively protect circuit from RTN

Thanks for your attention!