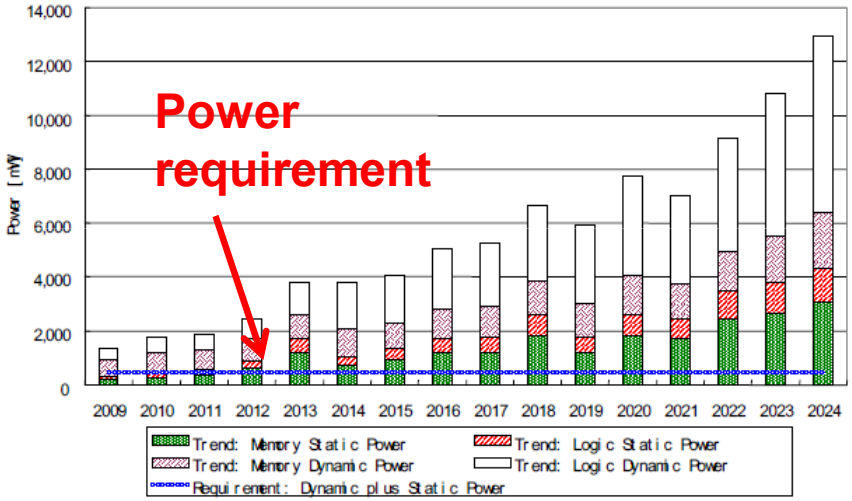
Charles L. Brown Department of
Electrical & Computer Engineering
University of Virginia School of Engineering and Applied Science

Exploring the Fidelity-Efficiency Design Space using Imprecise Arithmetic

Jiawei Huang
John Lach
University of Virginia
01/28/2011

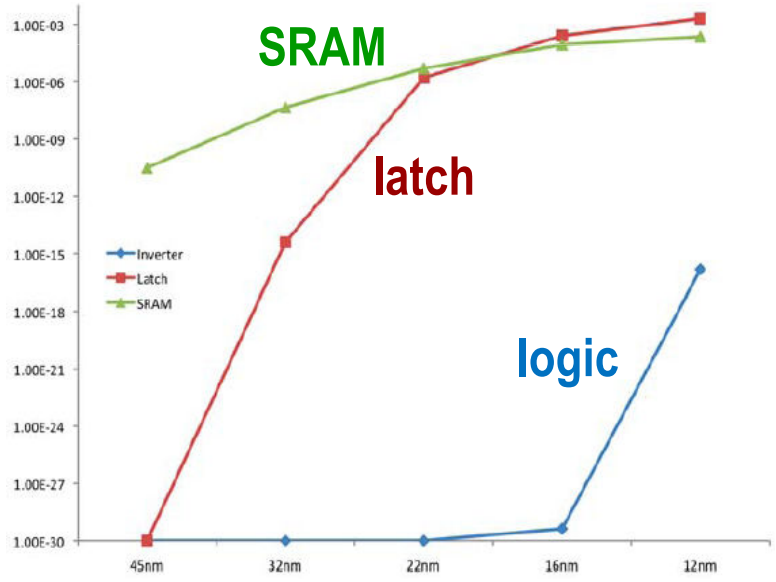
Challenges in Nanometer CMOS Era

- Power consumption



Portable electronics

- Reliability
  - PVT variation
  - Soft errors
  - Aging effects



failure rate

Technology

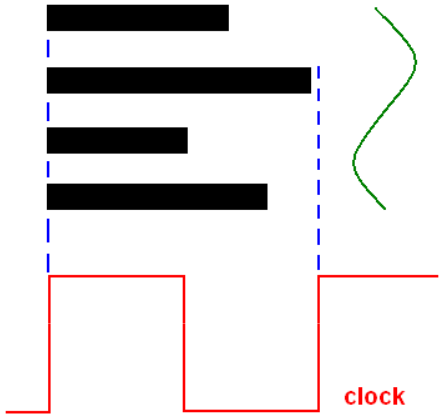
State-of-the-art Solutions

- Power reduction:
  - Power/clock gating
  - DVS
  - Sub-Vt
  - Multi-Vth
- Design for manufacturability:
  - Design for the worst-case
  - Adaptive circuits with sensors
  - Locally asynchronous design
- Fault-tolerant design:
  - Error-correcting code (ECC)
  - Redundancy

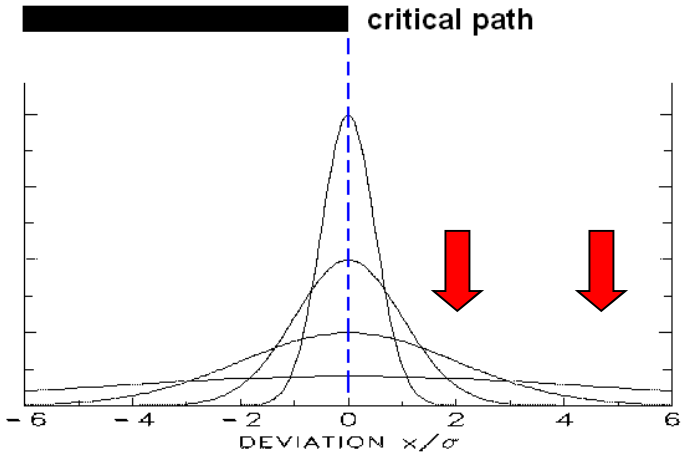
Overhead !

Limitation of Current Solutions

- Max performance or min power set by critical path.

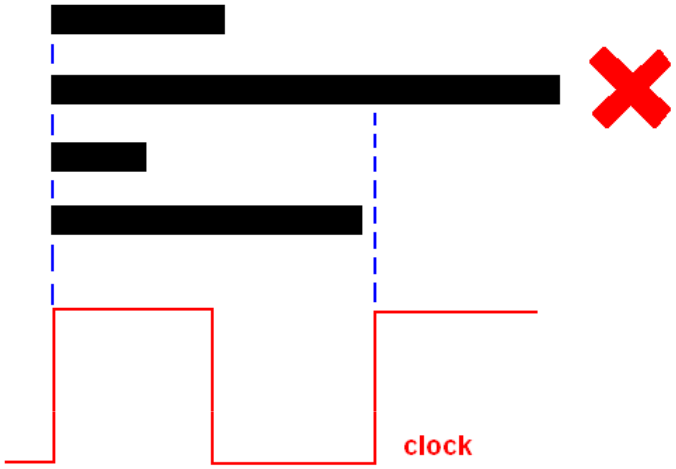


- Technology scaling exacerbates critical path variation.



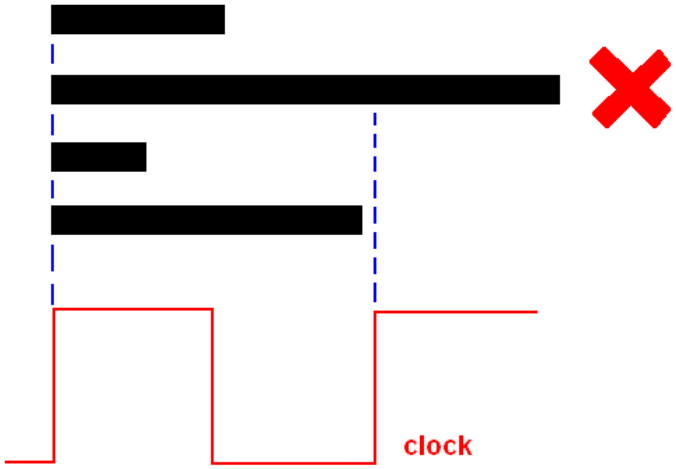
What if We Sacrifice Correctness?

- Timing errors occur.



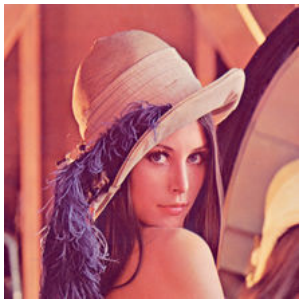
What if We Sacrifice Correctness?

- Timing errors occur.



- But applications may be able to tolerate errors.
  - Digital signal processing
  - Communication
  - Recognition, Mining and Synthesis
  - Numerical methods

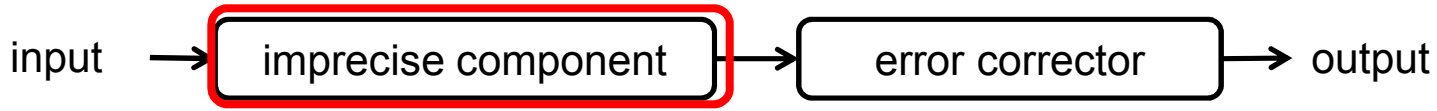
CDMA



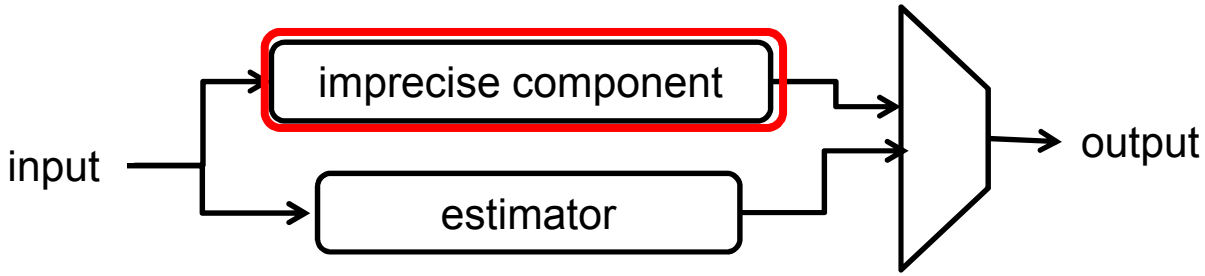
Wi Fi™

Circuit Design that Allows Errors

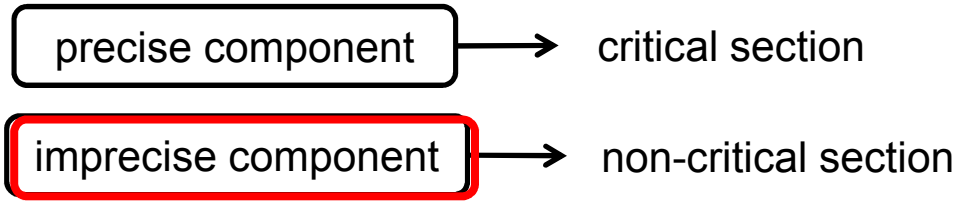
- BTWC (T. Austin et al., Michigan)



- ANT (B. Shim et al., UIUC)



- ERSA (S. Mitra et al., Stanford)

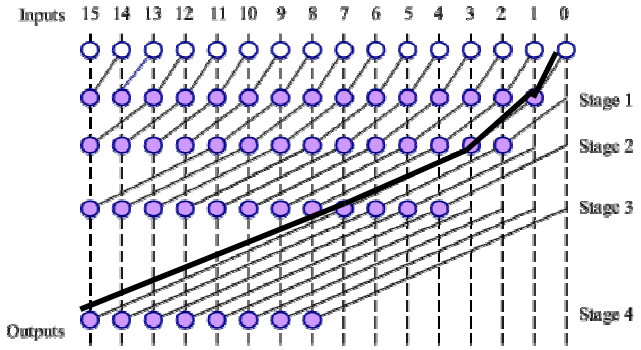


- Stochastic computation (N.R. Shanbhag et al., UIUC)

Overview of This Work

- Motivation
  - Lack of comparison of different imprecise circuits
  - Lack of design methodology of imprecise circuits
- Scope
  - Arithmetic circuits: adders (ACA, ETAIIM, VOS and reduced precision)
  - Errors from aggressive design (shorten critical path)
- Contributions
  - Imprecise circuits characterization
  - Comparison framework and design methodology
  - Imprecise design rules of thumb

Imprecise Adder (ACA)

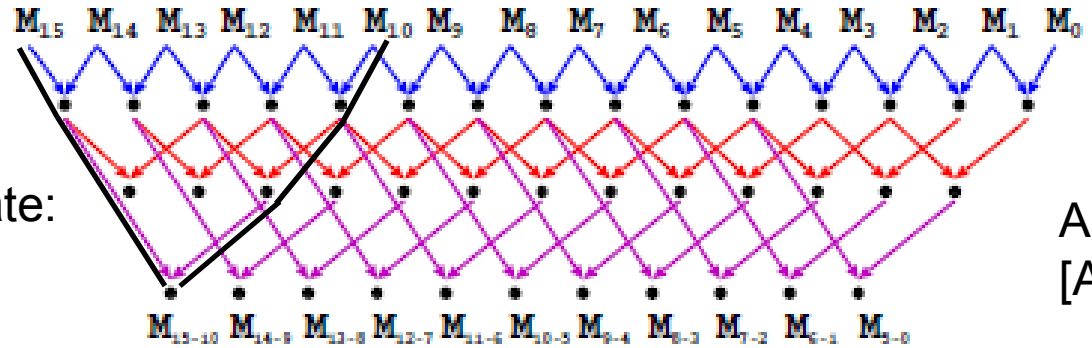


100% ⟵ **KSA** `log(n)`

delay

Reliability (1-error rate)

K



For fixed error rate:
K → log(n)

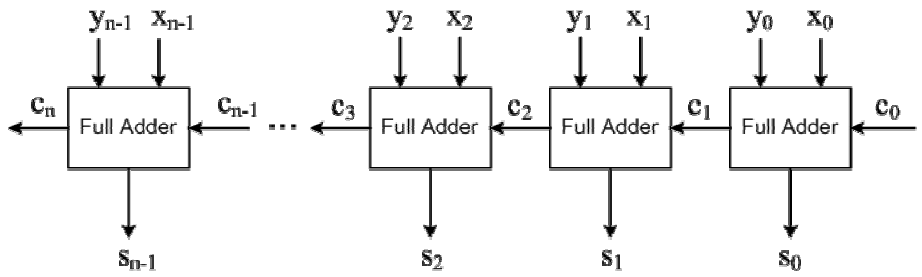
Almost-correct adder
[A. Verma et al.]

99.99% ⟵ **ACA** `log(log(n))`

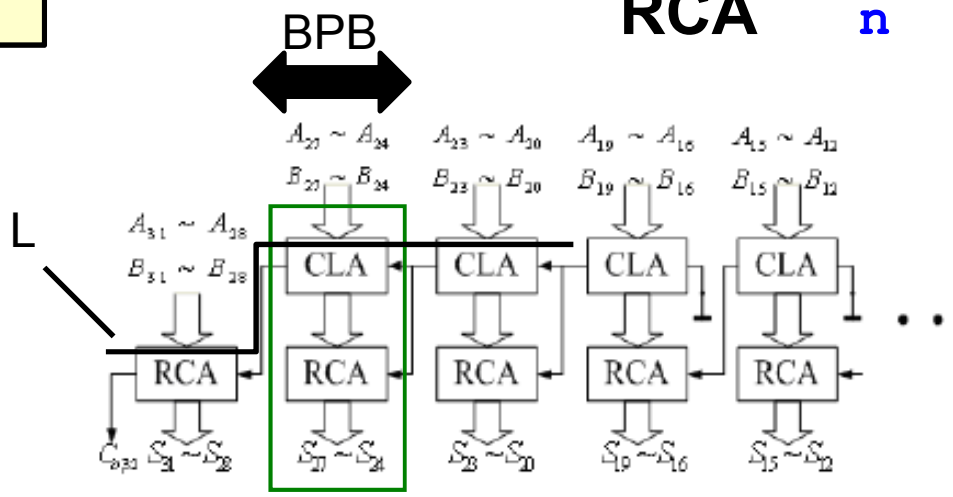
Imprecise Adder (ETAIIM)

delay

WC precision



RCA `n` ➡ `n`



ETAIIM `BPB×L < n` ➡ **(L-1)BPB**

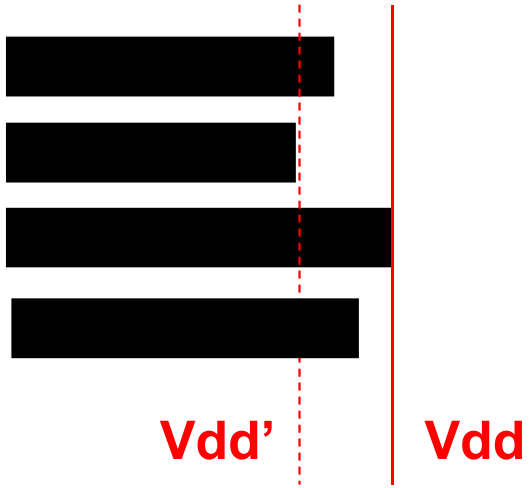
Error-tolerant Adder Type II M
[N. Zhu et al.]

Imprecise Adder (VOS)

- Voltage-overscaling [J. Sartori et al.]

power

Error rate



- Reduced precision:

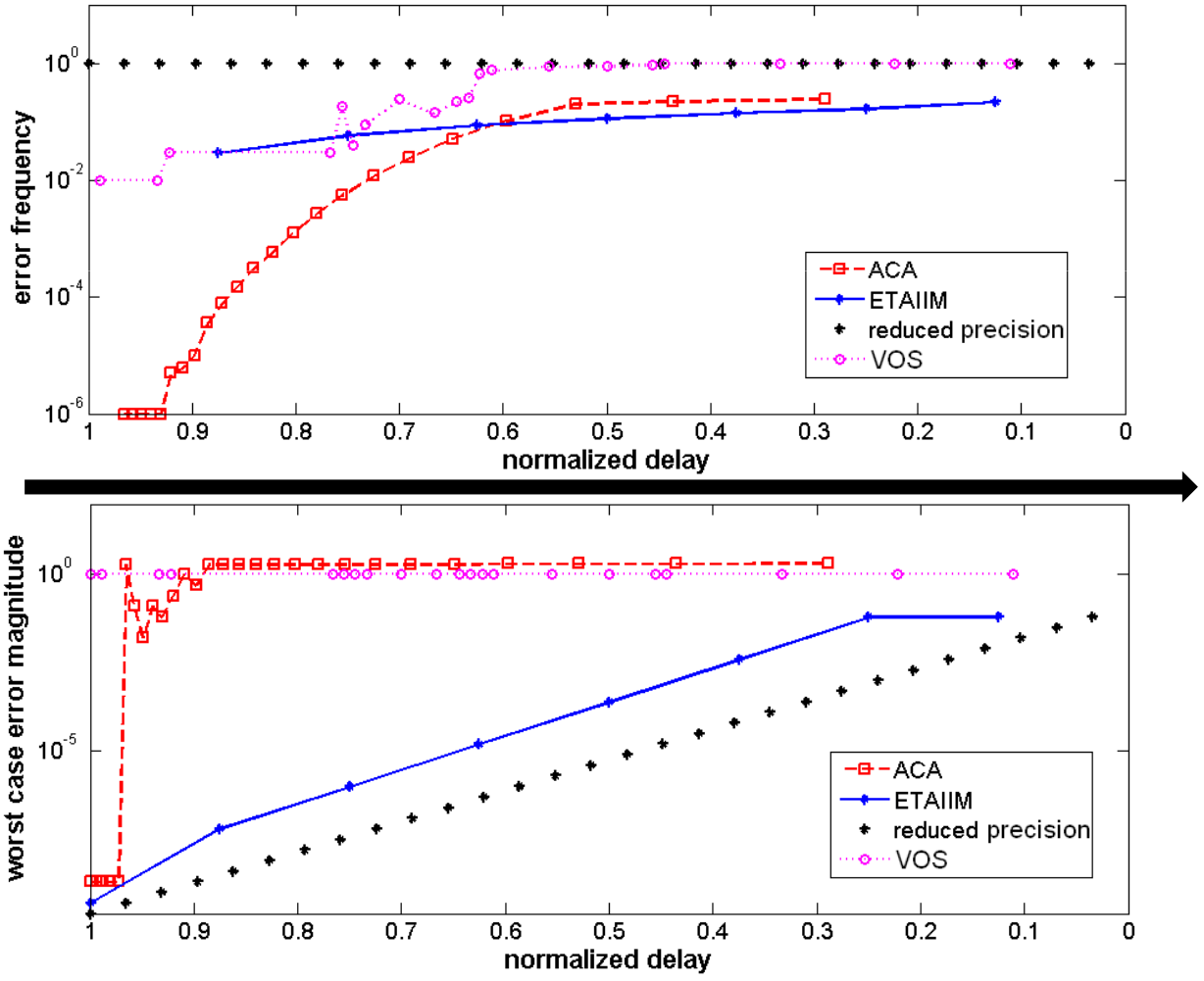
32-bit adder → 24-bit adder + Zero padding

power/delay

WC precision

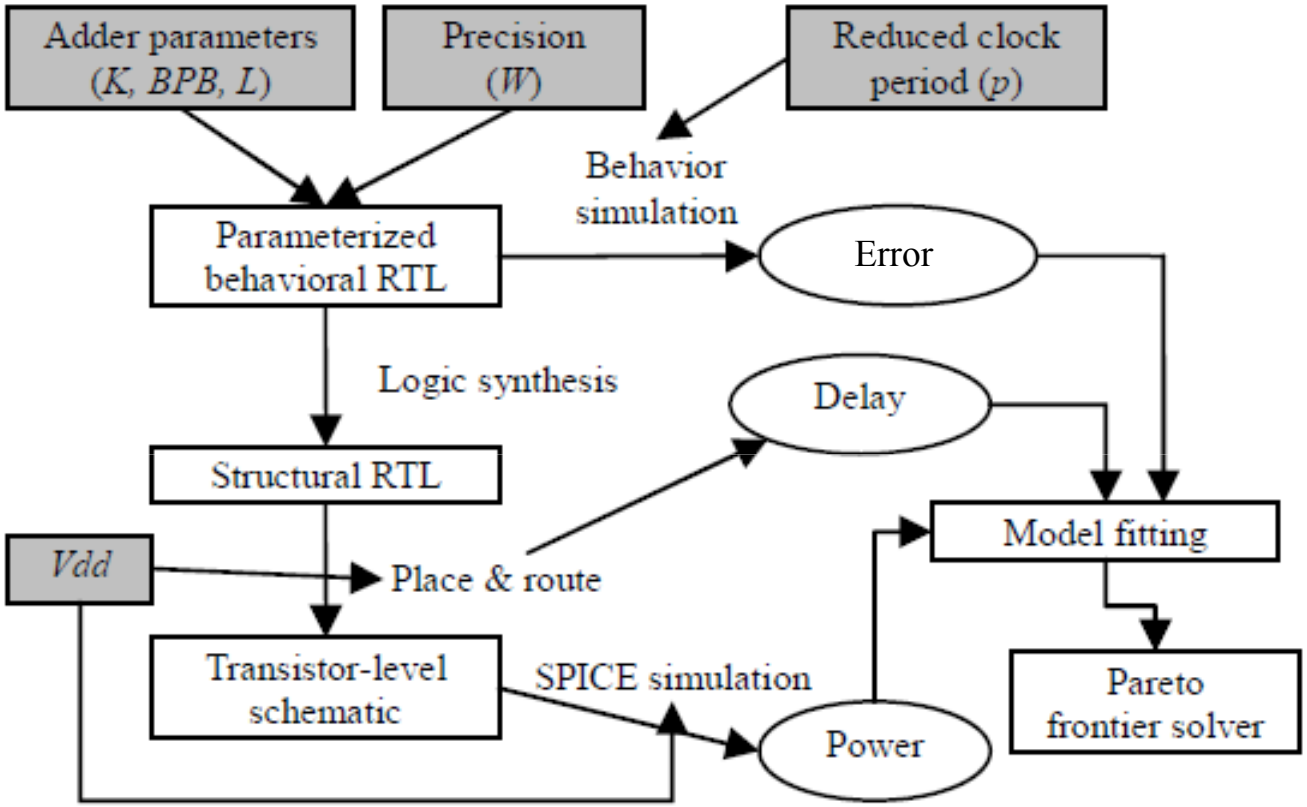
Error Characteristics of Imprecise Adder

- ILM: Infrequent Large-magnitude error — ACA, VOS
- FSM: Frequent Small-magnitude error — ETAIIM, reduced precision

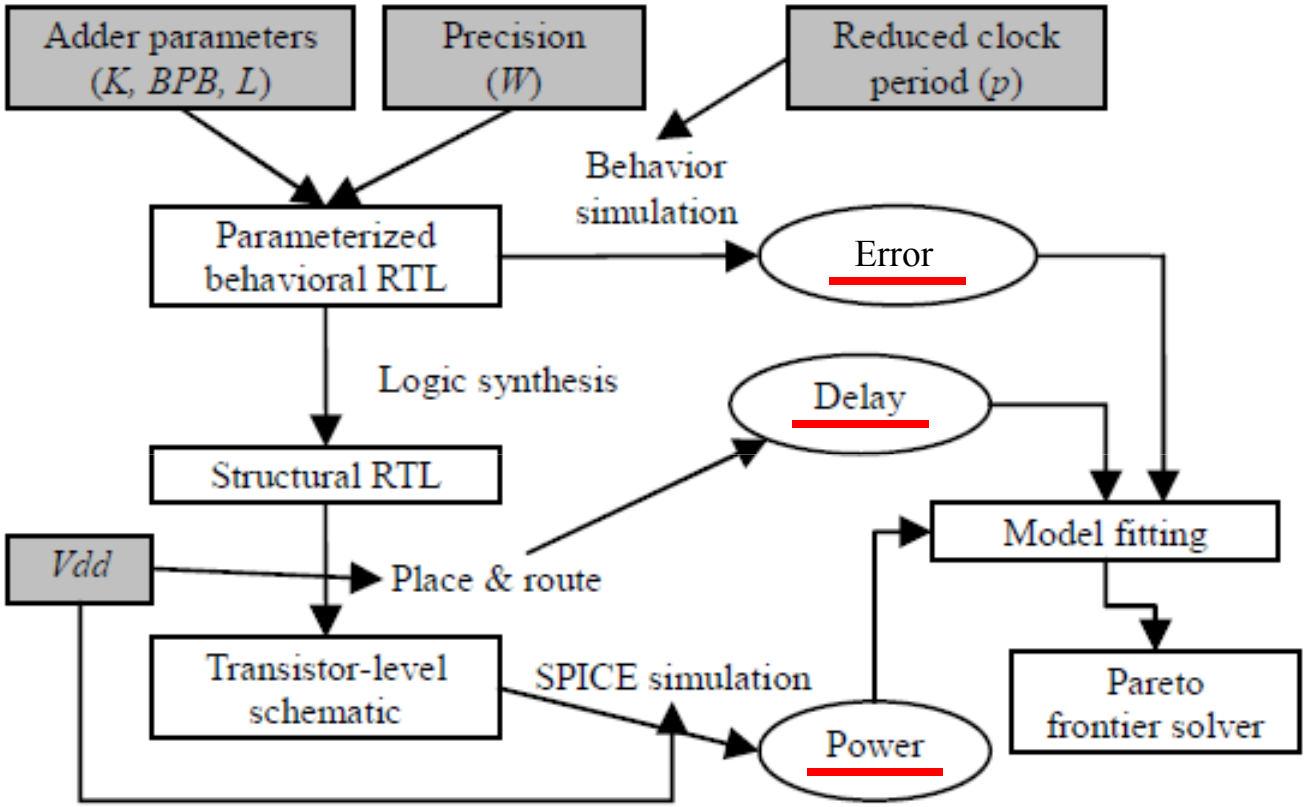


Increasingly imprecise

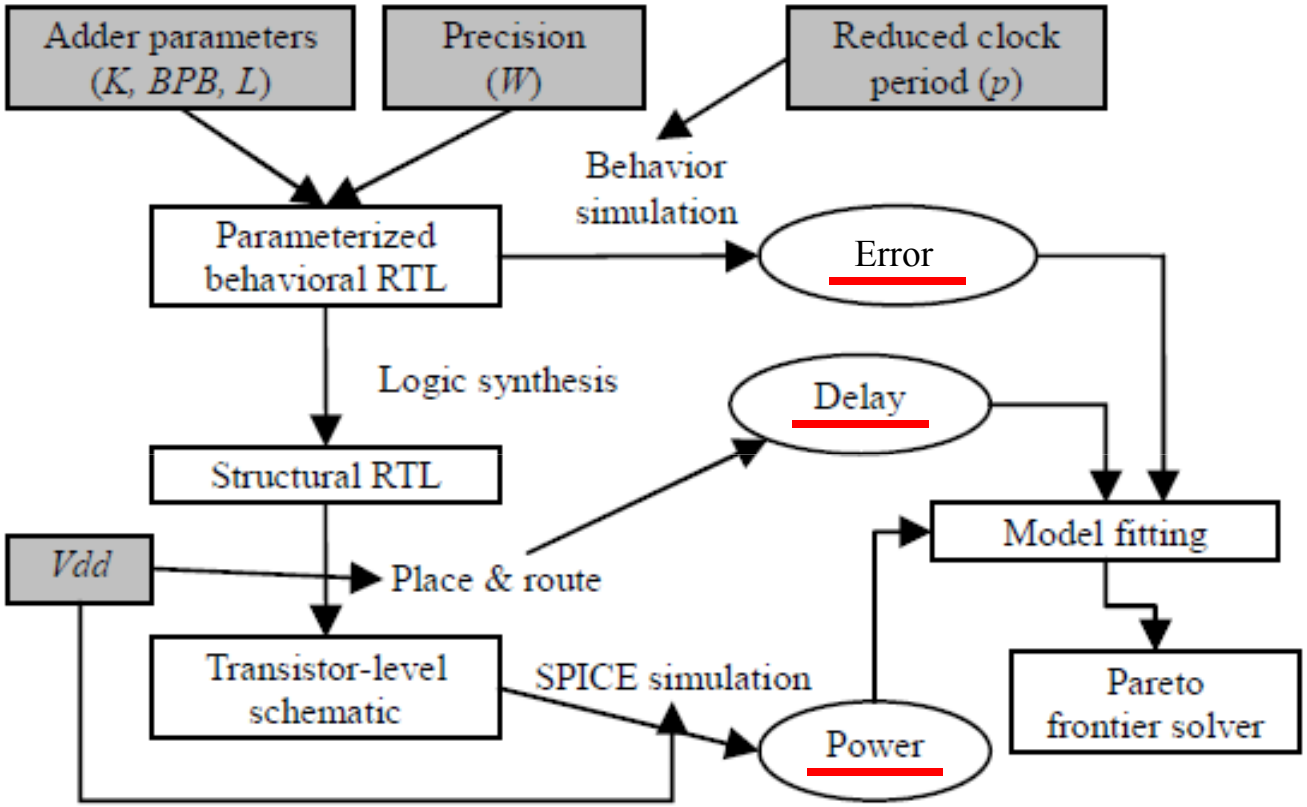
Comparison Framework



Comparison Framework



Comparison Framework



$\text{Error} = E(K, BPB, L, W, V_{dd}, p)$

$\text{Delay} = D(K, BPB, L, W, V_{dd}, p)$

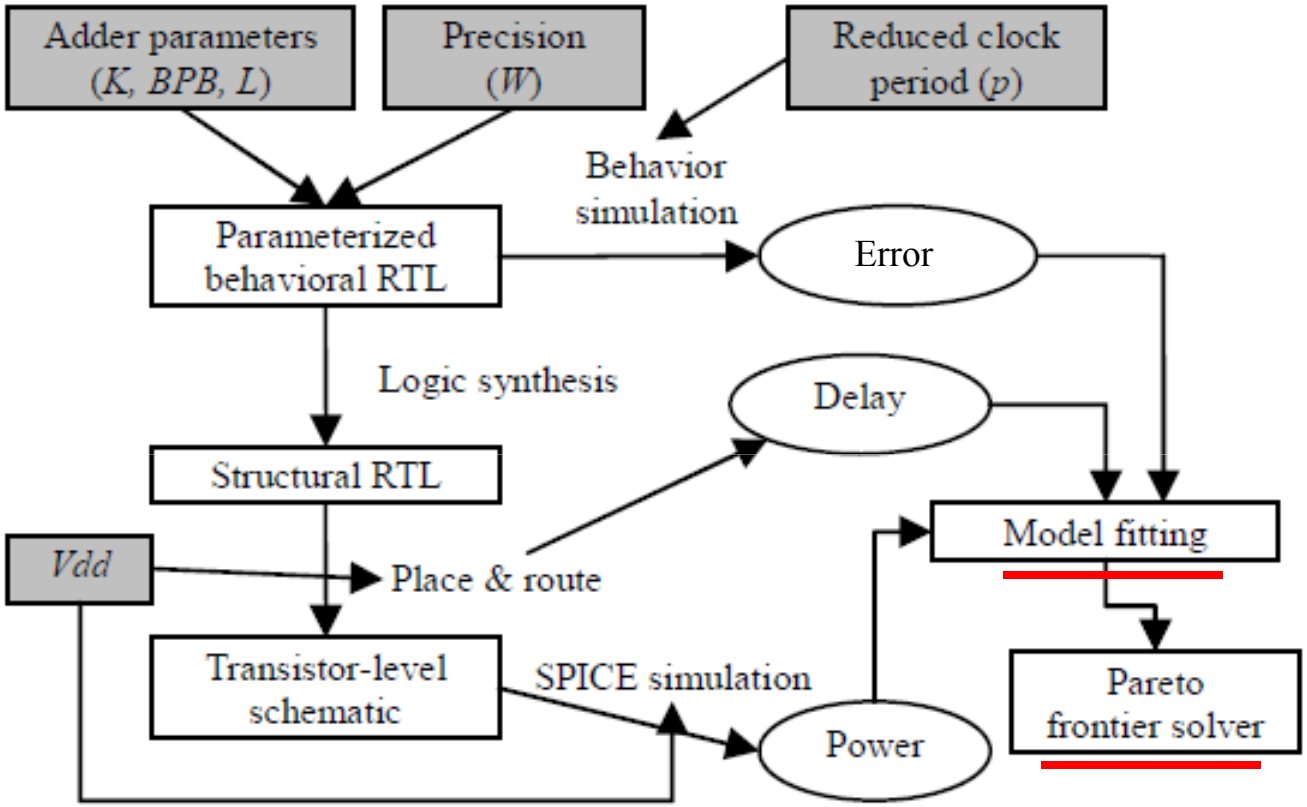
$\text{Power} = P(K, BPB, L, W, V_{dd}, p)$

$\text{Energy} = D \cdot P$

$\text{Current} = I_{nom}(K, BPB, L, W, V_{dd}, p)$ at Vdd_{nom} and p_{nom}

$$P = I_{nom} \cdot \frac{V_{dd}}{Vdd_{nom}} \cdot \frac{p_{nom}}{p} \cdot V_{dd}$$

Comparison Framework



$\text{Error} = E(K, BPB, L, W, V_{dd}, p)$

$\text{Delay} = D(K, BPB, L, W, V_{dd}, p)$

$\text{Power} = P(K, BPB, L, W, V_{dd}, p)$

$\text{Energy} = D \cdot P$

$\text{Current} = I_{nom}(K, BPB, L, W, V_{dd}, p)$ at Vdd_{nom} and p_{nom}

$$P = I_{nom} \cdot \frac{V_{dd}}{Vdd_{nom}} \cdot \frac{p_{nom}}{p} \cdot V_{dd}$$

Model Fitting

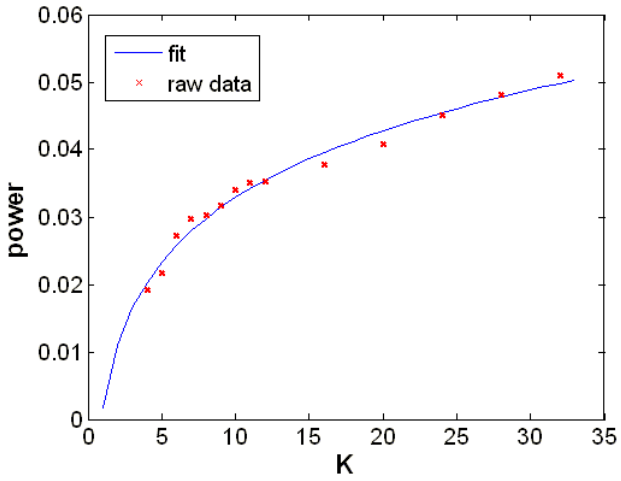
- Remove variables that have no effect.

$$I_{ACA}(K, B, L, W, V_{dd}, p)$$

- Determine function types based on implementation knowledge.

$$I_{ACA}(K, V_{dd}, p) = (A \log_2 K + B \cdot K + C) \cdot \frac{V_{dd}^2}{p}$$

- Curve-fitting



$$I_{ACA}(K, V_{dd}, p) = (9.2 \times 10^{-3} \log_2 K + 6.563 \times 10^{-5} K + 1.684 \times 10^{-3}) \cdot \frac{5 V_{dd}^2}{1.2 p}$$

Pareto Frontier Solver

- Pareto frontier: the collection of best possible (E,D,P) points

  error delay power

- Small-sized problems: exhaustive search
  - Sweep the ranges of all variables, find the points not dominated by others

- Sensitivity balancing [D. Markovic et al., Z. Qi et al.]:

$$\begin{vmatrix} \frac{\partial E_{ETAIIM}(BPB,L,V_{dd})}{\partial BPB} & \frac{\partial D_{ETAIIM}(BPB,L,V_{dd})}{\partial BPB} & \frac{\partial P_{ETAIIM}(BPB,L,V_{dd})}{\partial BPB} \\ \frac{\partial E_{ETAIIM}(BPB,L,V_{dd})}{\partial L} & \frac{\partial D_{ETAIIM}(BPB,L,V_{dd})}{\partial L} & \frac{\partial P_{ETAIIM}(BPB,L,V_{dd})}{\partial L} \\ \frac{\partial E_{ETAIIM}(BPB,L,V_{dd})}{\partial V_{dd}} & \frac{\partial D_{ETAIIM}(BPB,L,V_{dd})}{\partial V_{dd}} & \frac{\partial P_{ETAIIM}(BPB,L,V_{dd})}{\partial V_{dd}} \end{vmatrix} = 0$$

Case Study: CORDIC

- Goal
  - ❑ to test the effectiveness of various imprecise design techniques for implementing a real application
  - ❑ under the same accuracy constraint { Power-delay Pareto frontier; Energy-delay product }
  - ❑ to discover rules of thumb for choosing imprecise designs

CORDIC

- Compute any elementary function using *shift* and *add*

$$+ \; - \; \times \; / \; \sin \; \cos \; \arctan \; \sinh \; \ln \; \exp \; \sqrt{\;} \; \cdots$$

- Iterative algorithm

$$\begin{aligned} X_{i+1} &= X_i - Y_i \cdot d_i \cdot 2^{-i} \\ Y_{i+1} &= Y_i + X_i \cdot d_i \cdot 2^{-i} \\ Z_{i+1} &= Z_i - d_i \cdot \tan^{-1}(2^{-i}) \\ \text{where } d_i &= -1 \text{ if } Z_i < 0, \; +1 \text{ otherwise} \end{aligned}$$

precomputed

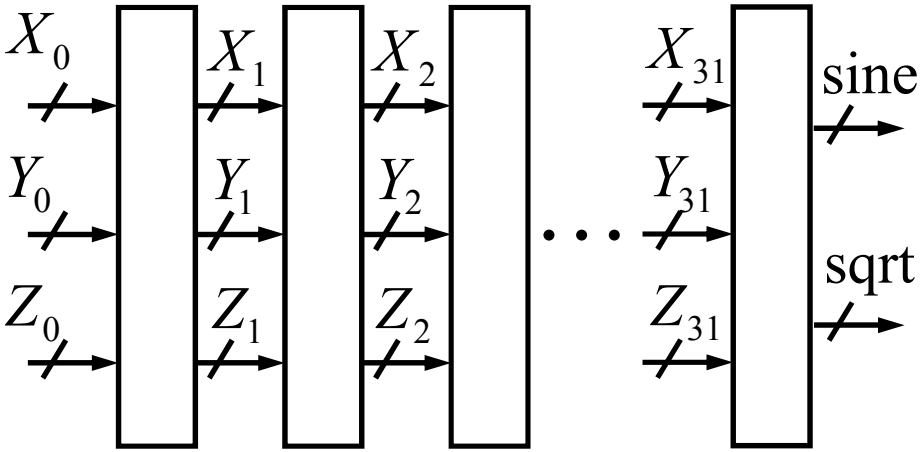
- To compute $\sin\theta$

set $\begin{cases} X_0 = x_0 \\ Y_0 = 0 \\ Z_0 = \theta \end{cases}$ **get** $\begin{cases} X_n = K_1 x_0 \cos\theta \\ Y_n = K_1 x_0 \sin\theta \end{cases}$

- To compute $\sqrt{\omega}$

set $\begin{cases} X_0 = \omega + 1/4 \\ Y_0 = \omega - 1/4 \\ Z_0 = \text{don't care} \end{cases}$ **get** $\begin{cases} X_n = K_{-1}\sqrt{\omega} \\ Y_n = 0 \end{cases}$

Our CORDIC Architecture



Data width	CORDIC function	Candidate adders	Fidelity metric	Technology
32 bit	sine sqrt	Precise: RCA-32, KSA-32 Imprecise: ETAIIM-32, ACA-32 Reduced precision: RCA-24, KSA-24 VOS: VOS-RCA-32, VOS-KSA-32	Mean absolute error	130nm

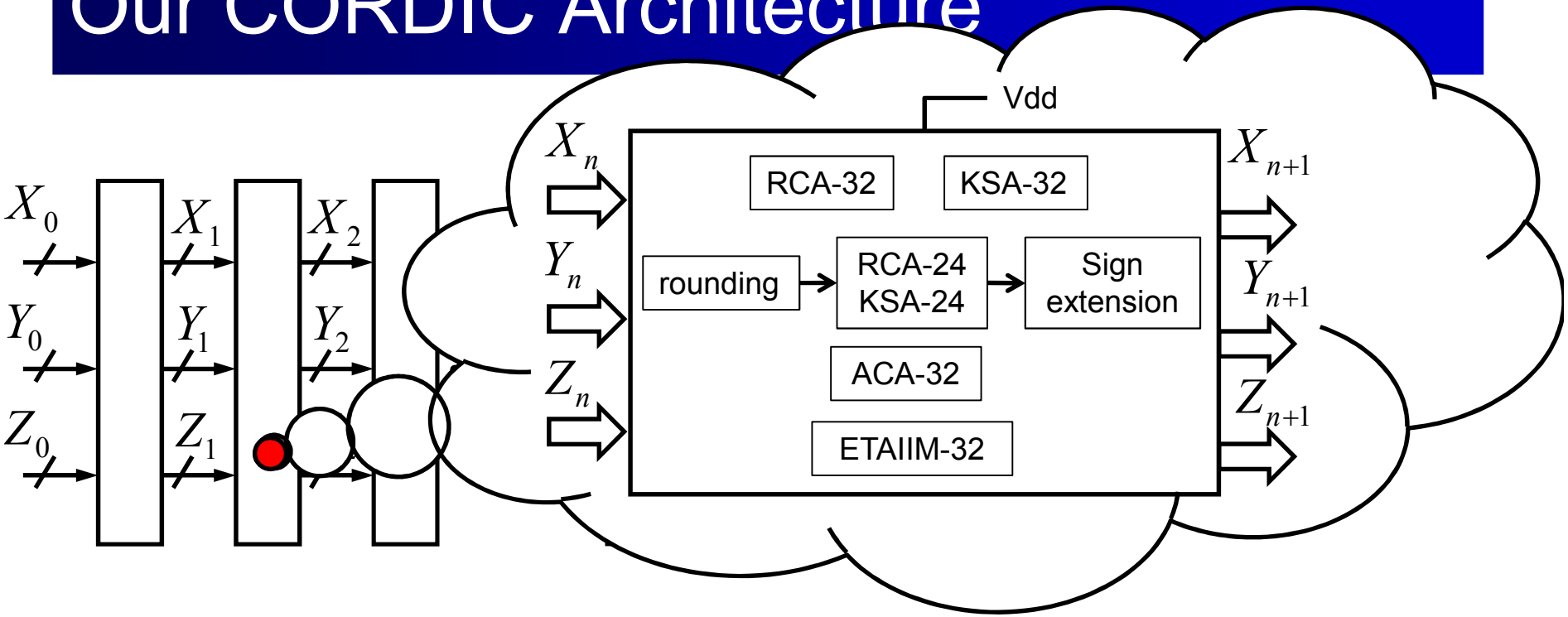
$\sin 0^\circ \sim \sin 90^\circ$

$\sqrt{0} \sim \sqrt{0.75}$

$V_{dd} = 0.5V \sim 1.2V$

$$\frac{\sum |O' - O|}{n}$$

Our CORDIC Architecture



Data width	CORDIC function	Candidate adders	Fidelity metric	Technology
32 bit	sine sqrt	Precise: RCA-32, KSA-32 Imprecise: ETAIIM-32, ACA-32 Reduced precision: RCA-24, KSA-24 VOS: VOS-RCA-32, VOS-KSA-32	Mean absolute error	130nm

$\sin 0^\circ \sim \sin 90^\circ$

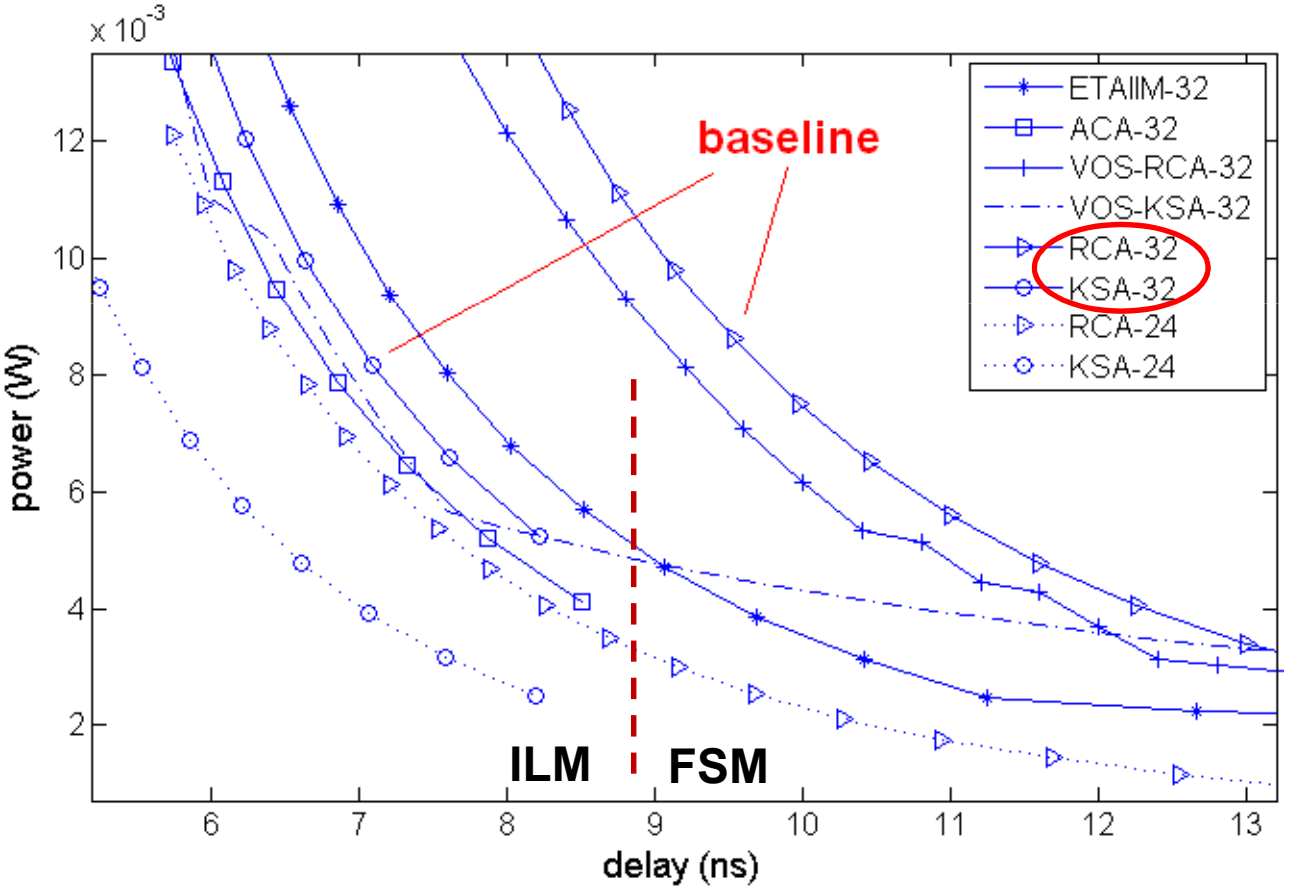
$\sqrt{0} \sim \sqrt{0.75}$

$V_{dd} = 0.5V \sim 1.2V$

$$\frac{\sum |O' - O|}{n}$$

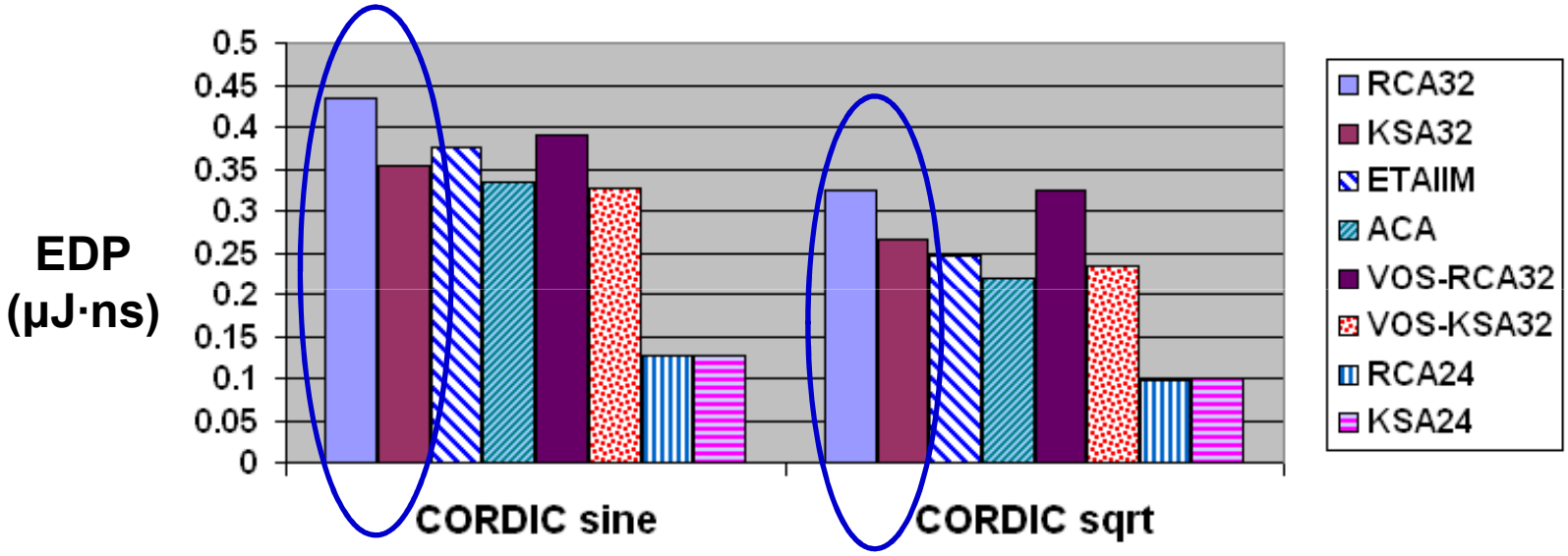
Experimental Results (Pareto frontier)

$$MAE = 2^{-24}$$



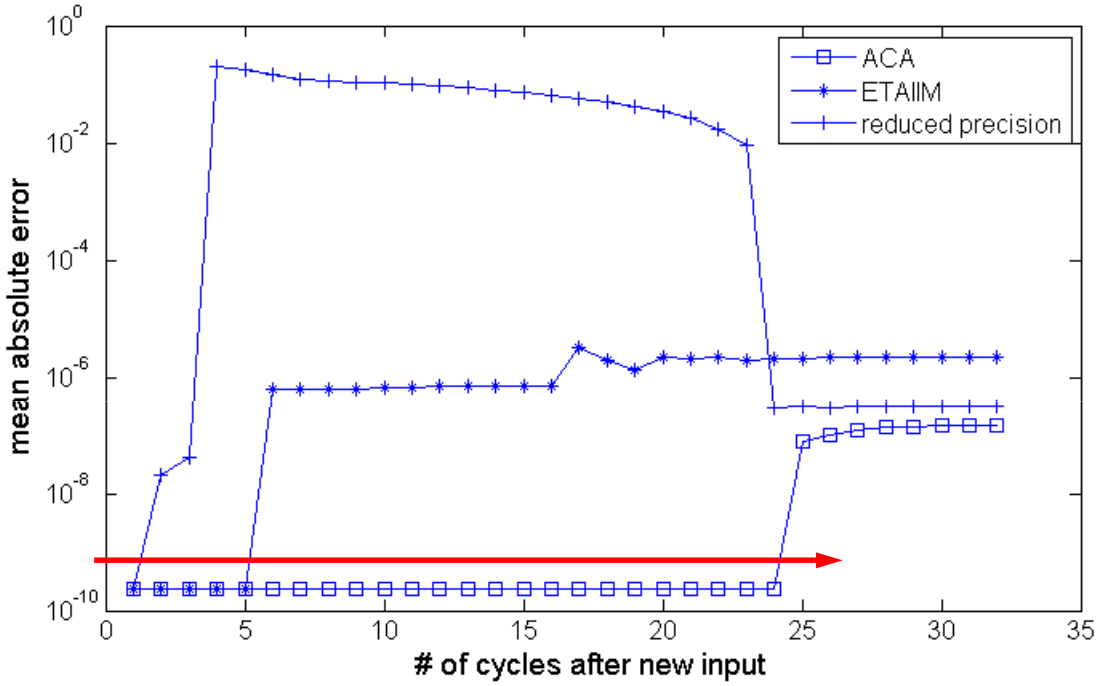
Energy/op-Delay Product Reduction

$$MAE = 2^{-24}$$



ETAIIM	ACA	VOS-RCA	VOS-KSA	RCA reduced precision	KSA reduced precision
19%	11%	5%	10%	**70%**	**63%**

Experimental Results (time plot)



When to use imprecise adders?

Lower accuracy

Error rate

Algorithm

Conclusions

- There exists a fidelity-efficiency continuum for efficiency improvement when fidelity is relaxed.
- For CORDIC, simply lowering the precision gives the best fidelity-efficiency tradeoff than more complex imprecise design techniques.
- Rules of thumb:
  - FSM error → low-power high-latency design
  - ILM error → high-power low-latency design
- Future work:
  - Need protection mechanism (e.g. input pattern detection) for iterative algorithms.
  - Error modeling.
  - Combination of multiple imprecise techniques (e.g. VOS+ACA).

Thank you!