SAADI: A Scalable Accuracy Approximate Divider for Dynamic Energy-Quality Scaling

Setareh Behroozi, Jingjie Li,
Jackson Melchert, Younghyun Kim

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

24th Asia and South Pacific Design Automation Conference (ASP-DAC)
Tokyo, Japan, January 22–24, 2019

Human Brain vs. Machine Brain

10.012 apples per student
9.9822 students per class
How many apples per class?

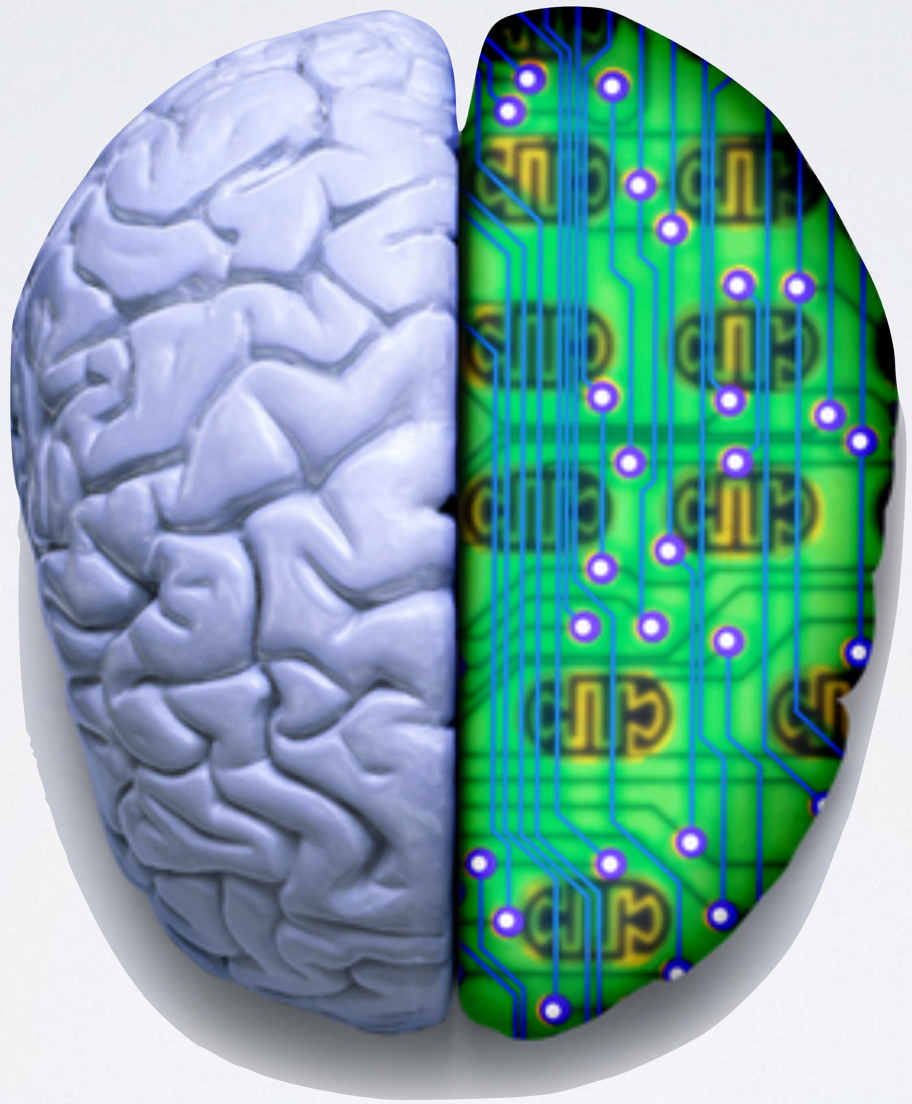
10.012×9.9822=?

What?

10.012×9.9822
***roughly*?**

100

Slow, but always efficient



10.012×9.9822=?

99.9417864

10.012×9.9822
***roughly*?**

99.9417864

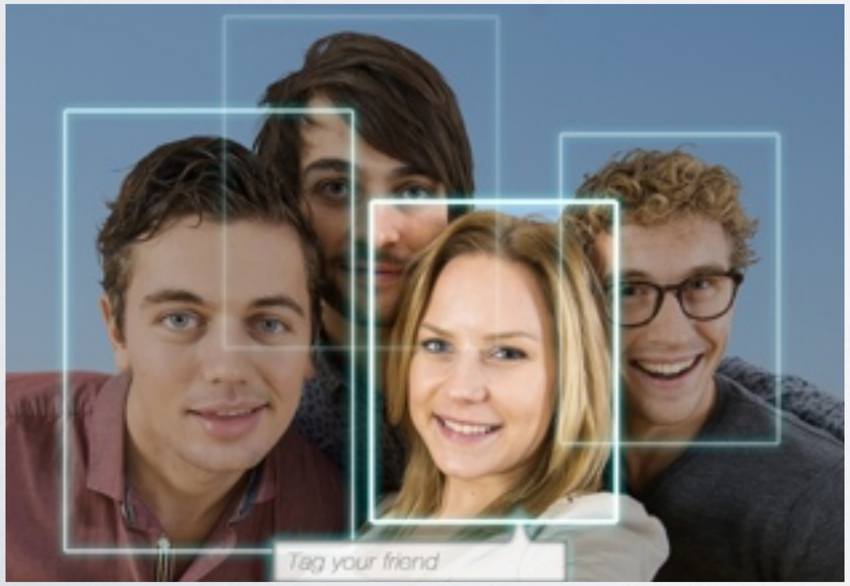
What's "*roughly*"?

Fast, but sometimes inefficient

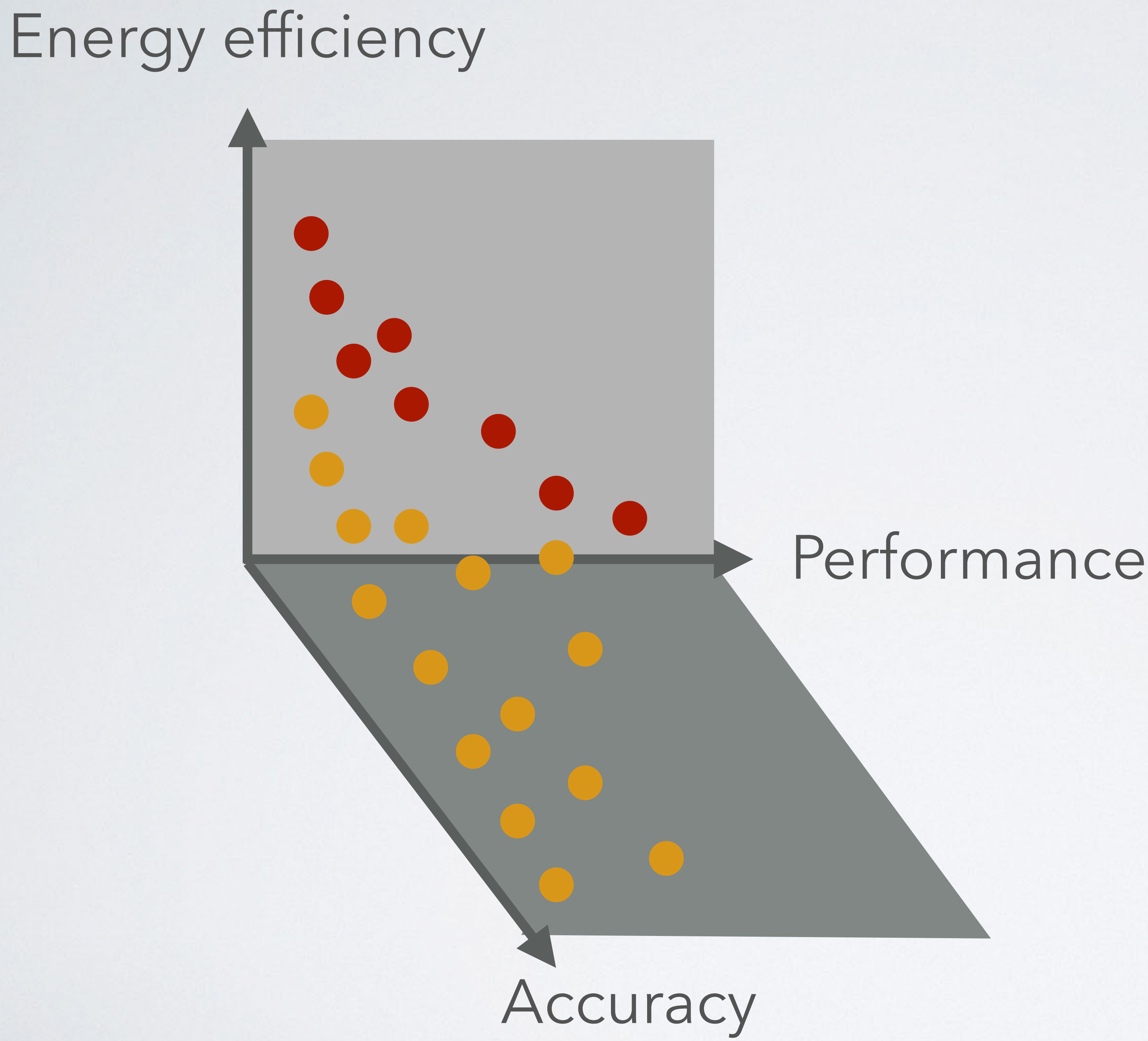
If *approximate* results are good enough, can we do it efficiently?

APPROXIMATE COMPUTING

- Happy with **good enough** solution
- Maximize **quality-per-effort**, not quality
- Many applications are **resilient to errors** in underlying computing
  - Audio/video signal processing, machine learning, search and data mining




APPROXIMATE COMPUTING



- **Simpler, faster, more efficient** hardware and software
- More opportunities to **improve energy efficiency and performance**
- Improved **application-level quality**

Division Operation

Capsule neural network (CapsNet)

Color quantization

Image division (difference detection)

DIVISION IS EXPENSIVE

DIV ÷ vs × MUL

IDIV	9-25 cycles (32 bit)
IMUL	3 cycles (32 bit)

AMD 12h family

Area	1.35x to 3x
Delay	1.27x

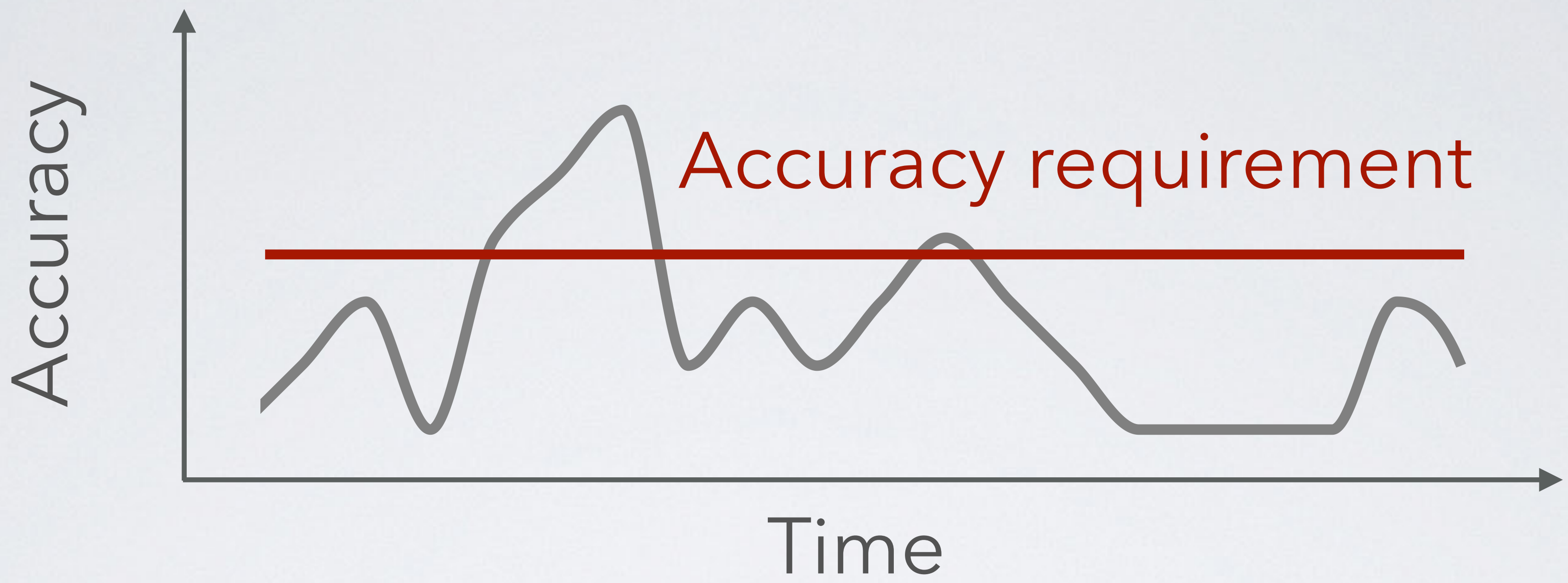
Intel FPGA

Challenge: A hardware divider is a costly module

~~Exact results~~ → Just good enough results

Approximate divider

ACCURACY REQUIREMENT



Application accuracy requirement varies over time

Dynamic quality configuration

Previous approx. dividers		
	SEERAD	R. Zendegani et al., SEERAD: A High Speed Yet Energy-Efficient Rounding-based Approximate Divider. In DATE 2016
	TruncApp	S. Vahdat et al., TruncApp: A Truncation-based Approximate Divider for Energy Efficient DSP Applications. In DATE 2017
	AAXD	H. Jiang et al., Adaptive Approximation in Arithmetic Circuits: A Low-Power Unsigned Divider Design. In DATE 2018

Approximate accuracy is fixed at design time

PROPOSED APPROACH: SAADI

SAADI = A **S**calable **A**ccuracy **A**pproximate **Di**vider for **Dynamic Energy-Quality Scaling**

Key features

Approximate

Multiplicative

Dynamic quality configuration

8-bit SAADI for 32 bit division (NanGate 45nm CMOS)

92.5%-99.0% average accuracy
0.66–4.67 pJ energy consumption

32 bits precise SRT Radix-2 divider: 351 pJ

MULTIPLICATIVE DIVISION

Division

$A = 2^{e_a} \times a \qquad B = 2^{e_b} \times b$

Divider

$$Q = \frac{A}{B}$$

$$= 2^{e_a - e_b} \times \frac{a}{b}$$

$$= 2^{e_a - e_b} \times a \times R(b)$$

Multiplicative division

$A = 2^{e_a} \times a \qquad B = 2^{e_b} \times b$

Multiplier

$$Q = 2^{e_a - e_b} \times a \times R(b)$$

~~**Divider**~~

~~$R(b) = \frac{1}{b}$~~

Approximate Reciprocal

$$\tilde{R}(b)$$

APPROXIMATE RECIPROCAL $\tilde{R}(b)$

Tyler series

$$x = b - 1$$

$$R(b) = \frac{1}{b} = \frac{1}{1+x} = \sum_{i=0}^{\infty} |x| = 1 + |x| + |x|^2 + |x|^3 + |x|^4 + \cdots$$

Stop earlier

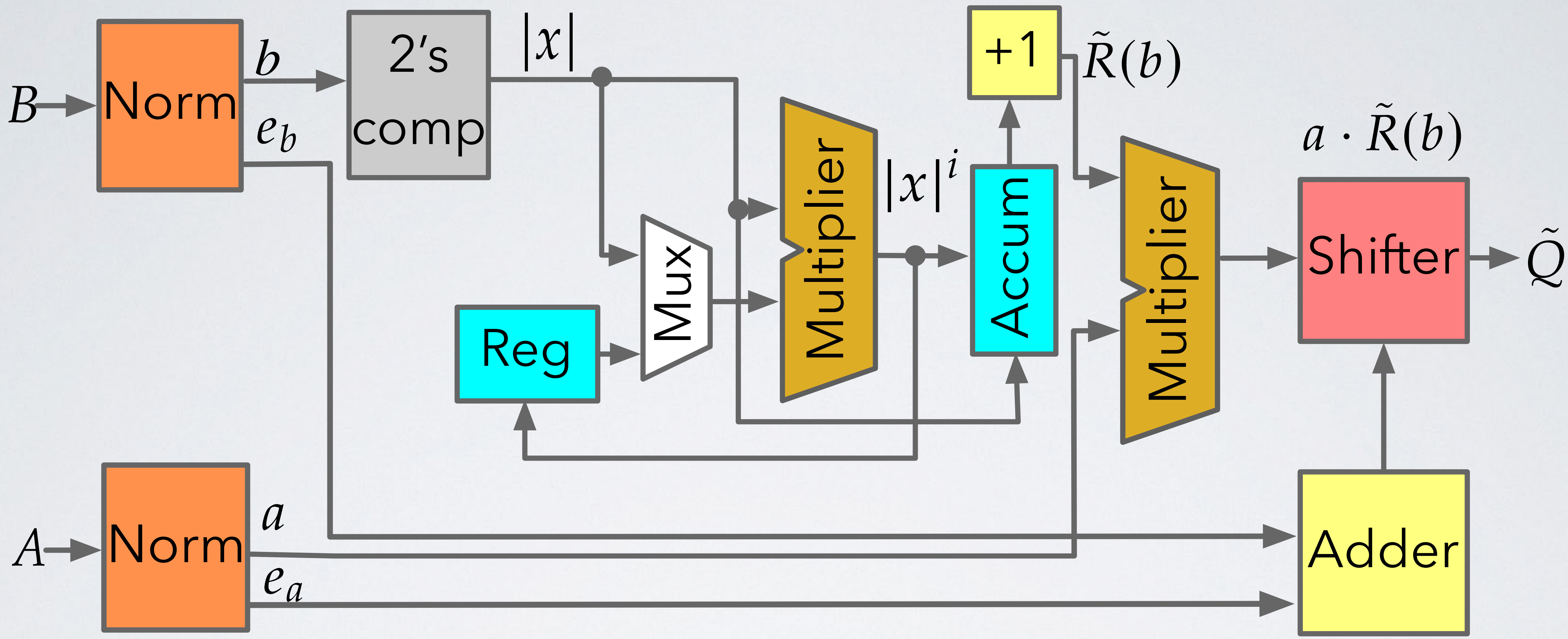
$$\tilde{R}_t(b) = \sum_{i=0}^{t} |x|^i = 1 + |x| + |x|^2 + |x|^3 + |x|^4 + \cdots + |x|^t$$

Stop at cycle *t*–1 and 1≤ *t* ≤ *n*–1

$$Q = 2^{e_a - e_b} \times a \times \tilde{R}_t(b)$$

Runtime accuracy control for dynamic quality configuration

HARDWARE ARCHITECTURE



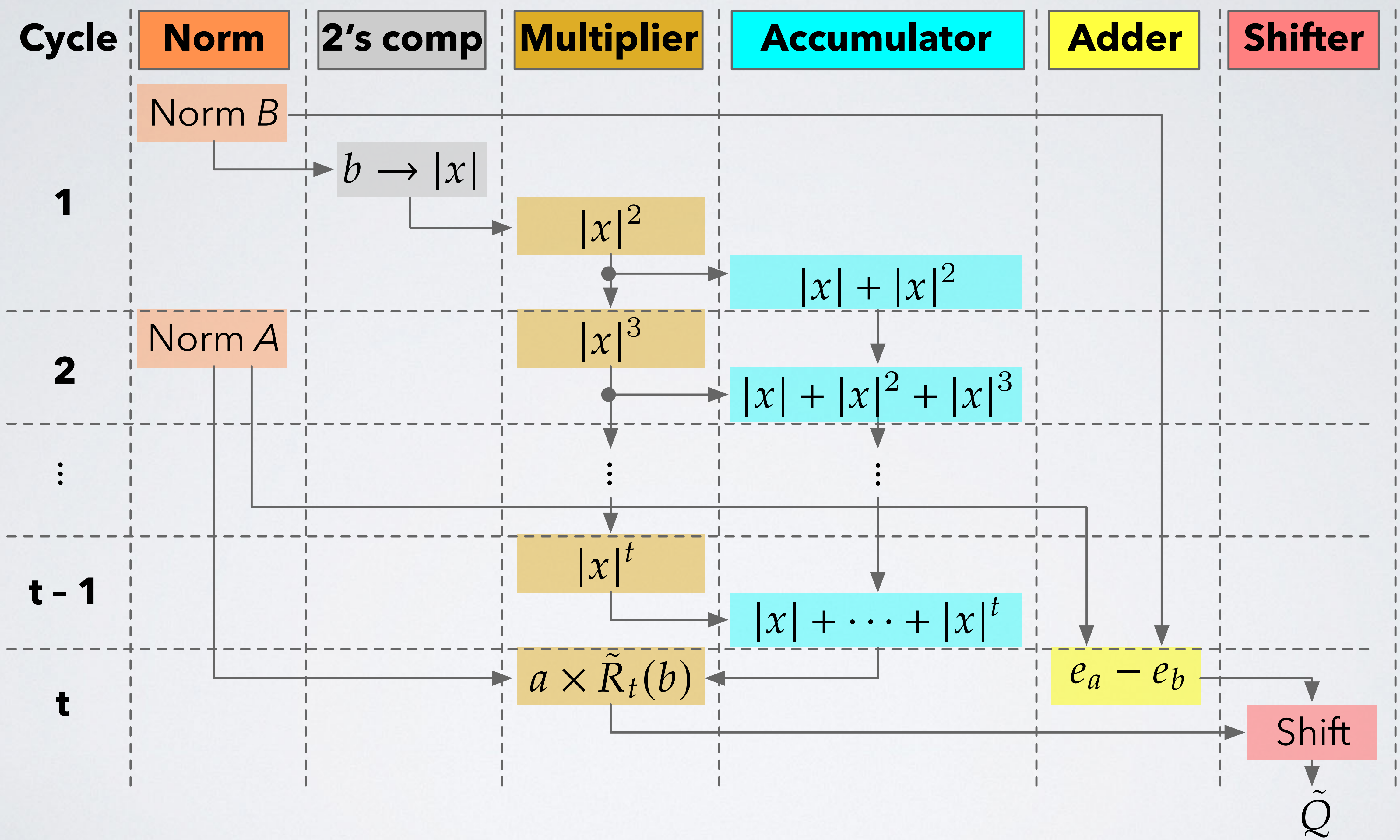
$$\tilde{R}_t(b) = 1 + |x| + |x|^2 + |x|^3 + |x|^4 + \cdots + |x|^t$$

$$Q = a \times \tilde{R}_t(b) \times 2^{e_a - e_b}$$

Design time parameter: Multiplier width: *n*

Run time parameter: Number of cycles: *t*

HARDWARE UTILIZATION



Sources of Error

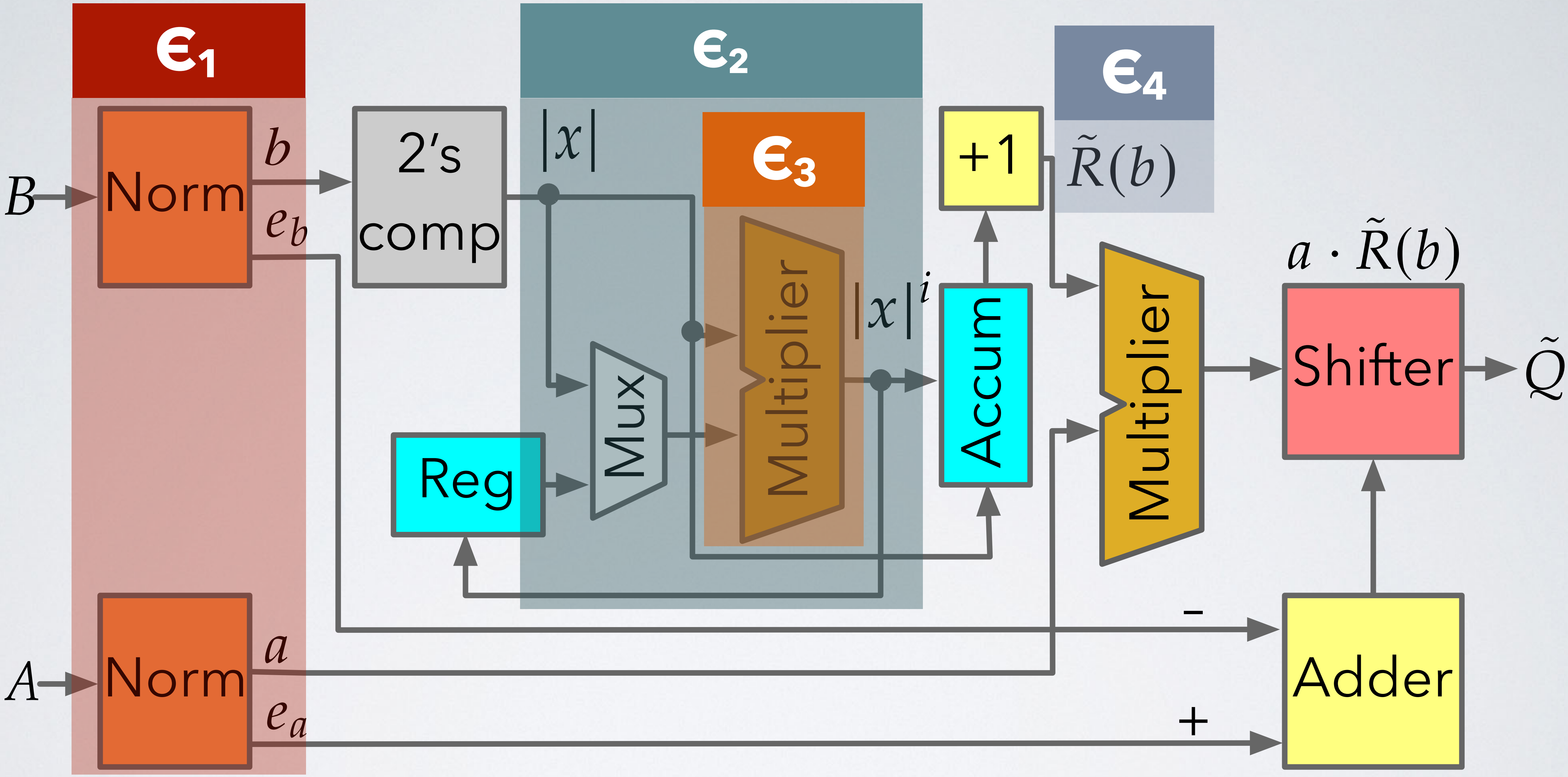
ϵ_1 Inputs *A* and *B* normalized to *n* bits

ϵ_2 $\tilde{R}_t(b)$ is the sum of limited number of $|x|^t$ terms

ϵ_3 Each $|x|^t$ computed by an approximate multiplier

ϵ_4 $\tilde{R}_t(b)$ truncated from *n*+2 bits to *n* bits

Source of Error



SAADI Example

$B = 11$ → $b = 0.68750$, $e_b = -4$

$A = 190$ → $a = 0.74219$, $e_a = 0$

$|x| = 0.31250$ → $\tilde{R}_1(b) = 1.31250$ → $\tilde{Q}_1 = 15.5000$ (error: -10.26%)

$|x|^2 = 0.09766$ → $\tilde{R}_2(b) = 1.40625$ → $\tilde{Q}_2 = 16.6250$ (error: -3.75%)

$|x|^3 = 0.02930$ → $\tilde{R}_3(b) = 1.43750$ → $\tilde{Q}_3 = 17.0000$ (error: -1.58%)

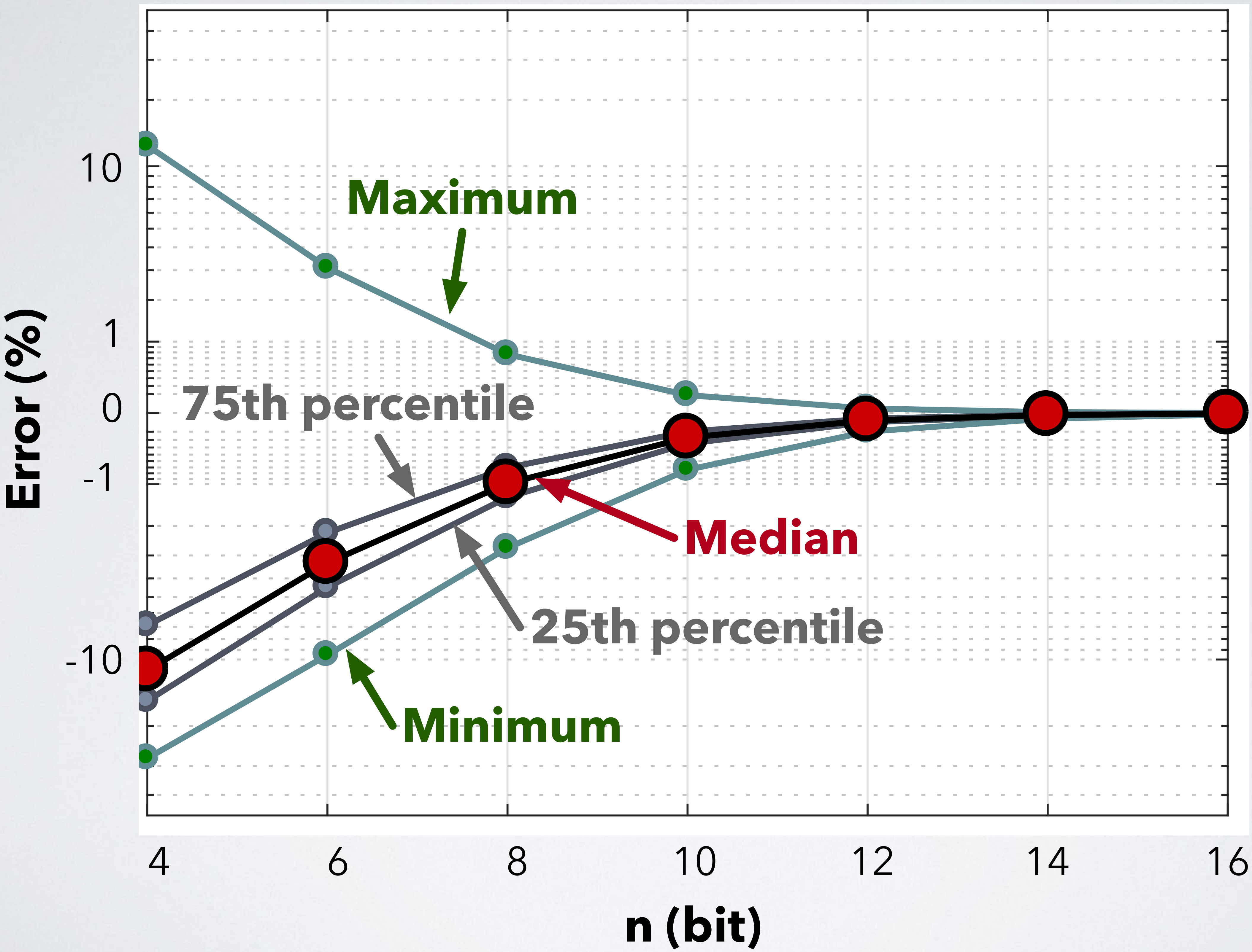
$|x|^4 = 0.00781$ → $\tilde{R}_4(b) = 1.44531$ → $\tilde{Q}_4 = 17.1250$ (error: -0.86%)

$|x|^5 = 0.00195$ → $\tilde{R}_5(b) = 1.44531$ → $\tilde{Q}_5 = 17.1250$ (error: -0.86%)

$|x|^6 = 0.00000$ → $\tilde{R}_6(b) = 1.44531$ → $\tilde{Q}_6 = 17.1250$ (error: -0.86%)

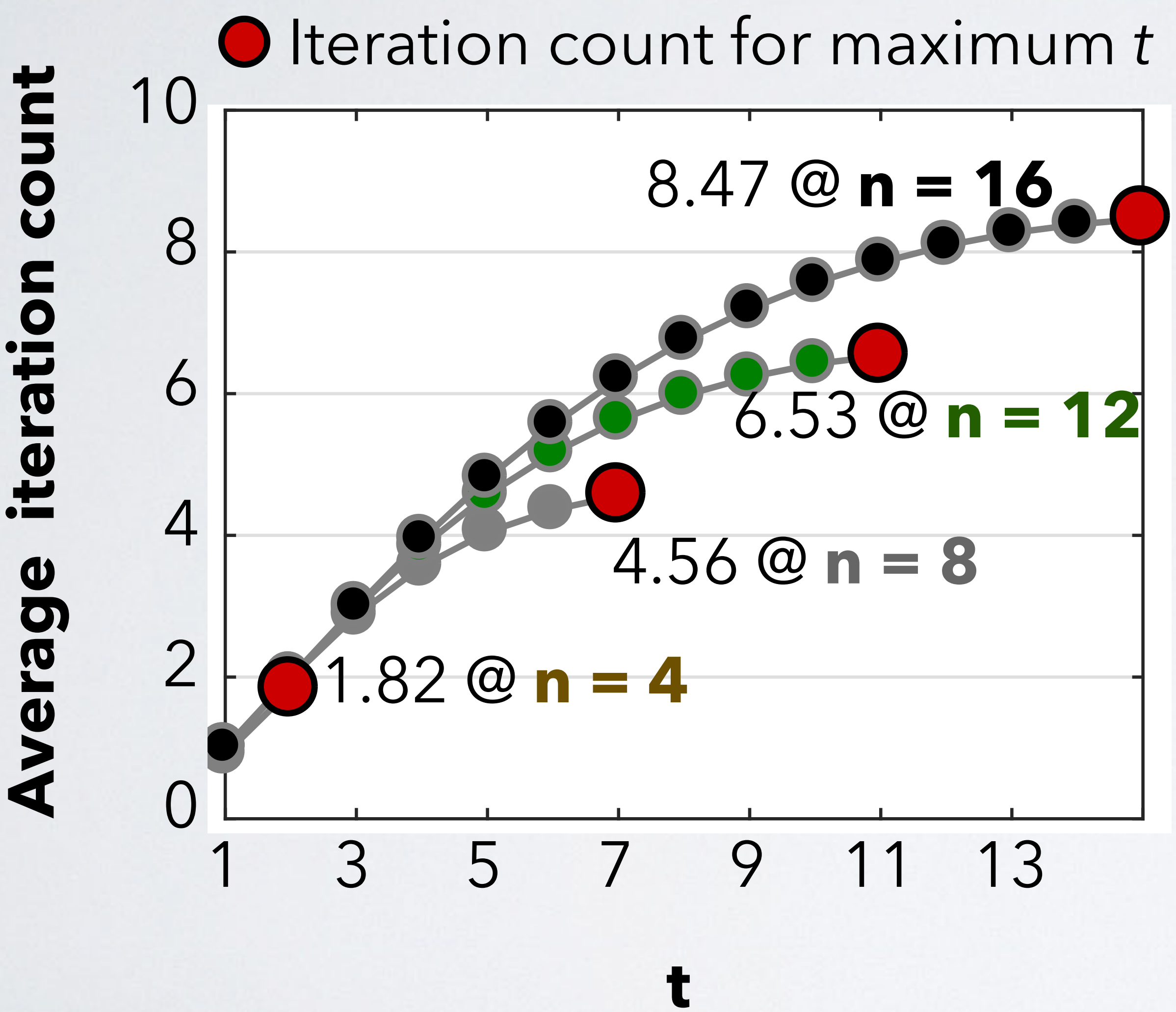
Exact $R(b) = 1.45455$ Exact $Q = 17.2727$

EXPERIMENTAL RESULTS: ACCURACY

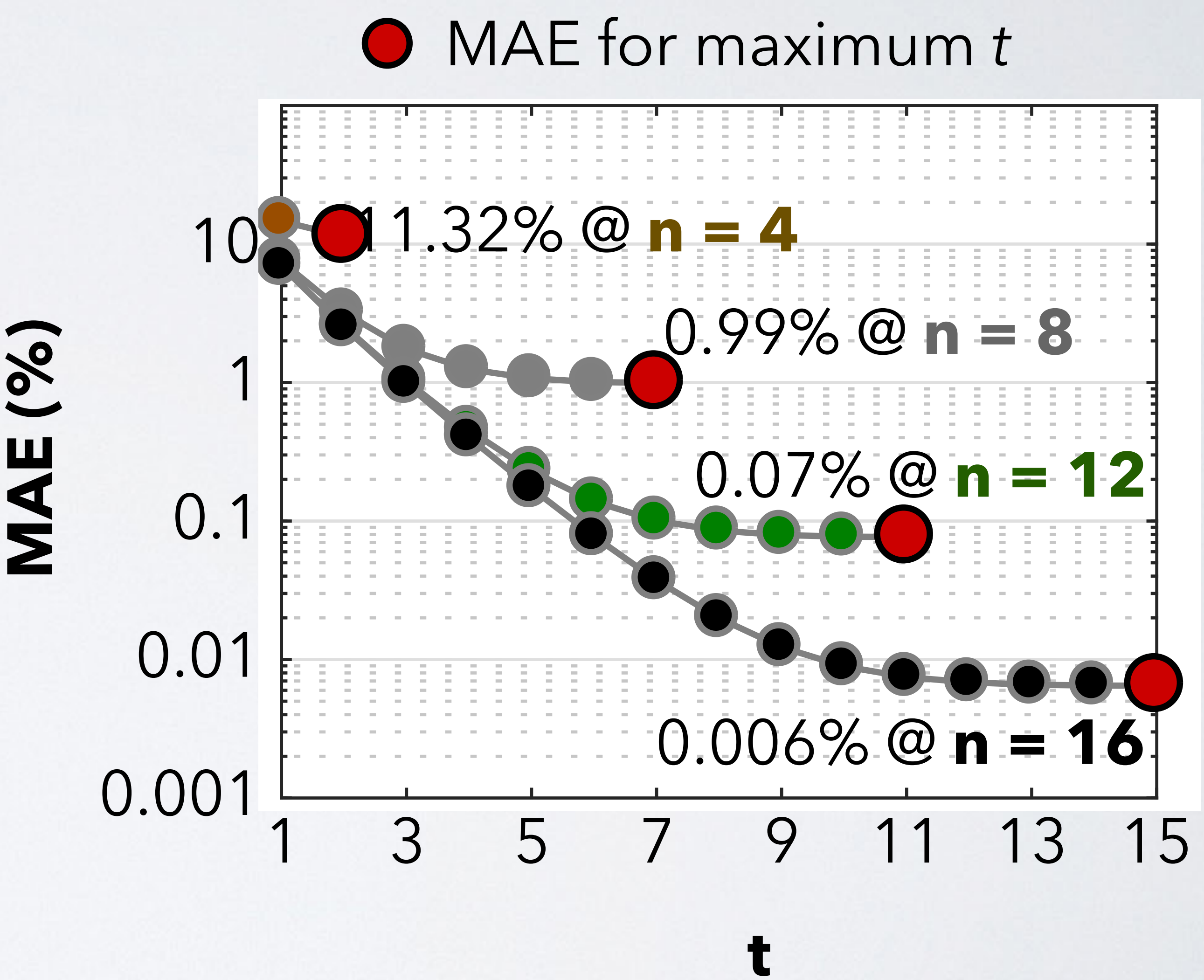


Accuracy

Average number of iterations for varying *n* and *t*



MAE for varying *n* and *t*



AREA, POWER, AND DELAY

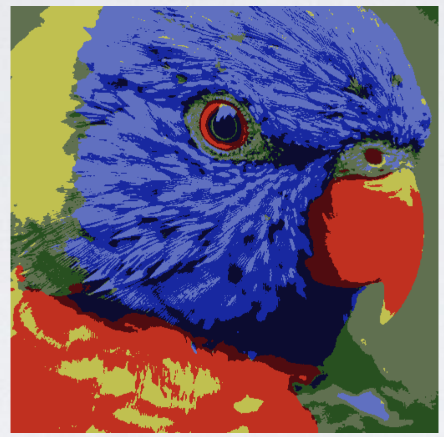
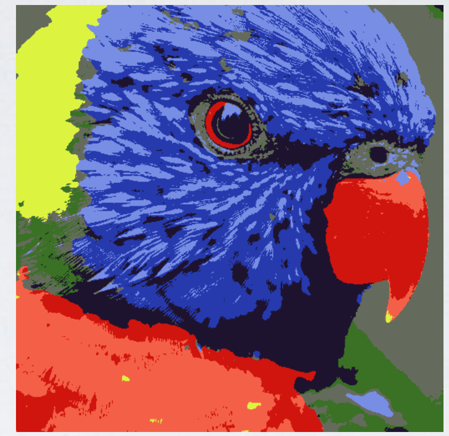
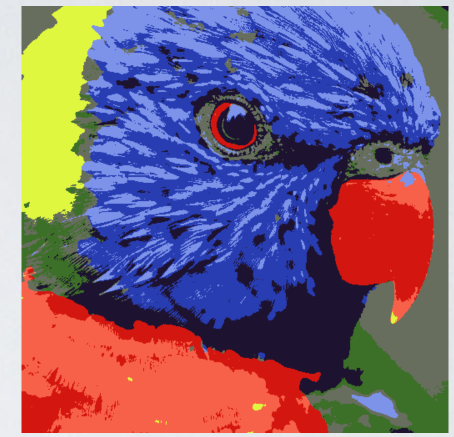
Bit width n(bit)		4	8	12	16
Area (μm²)		1,199	1,963	3,068	4,872
Delay (ns)		1.07	1.13	1.43	1.60
Power (mW)		0.31	0.59	1.09	1.94
Energy per cycle (pJ)		0.33	0.66	1.56	3.11
Target accuracy: 88%	*t*	2	1	1	1
	Energy (pJ)	0.66	0.66	1.56	3.11
Target accuracy: 99%	*t*	×	6	4	3
	Energy (pJ)	×	4.01	6.26	9.35
Target accuracy: 99.9%	*t*	×	×	7	6
	Energy (pJ)	×	×	10.96	18.73

Color Quantization using *K*-means Clustering

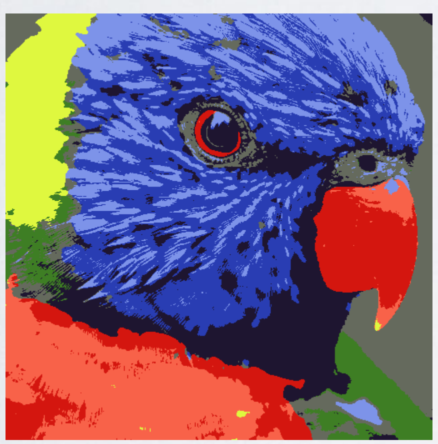
Original image

n↓ ⟵⟶ *n*↑

SAADI ($t = n - 1$)

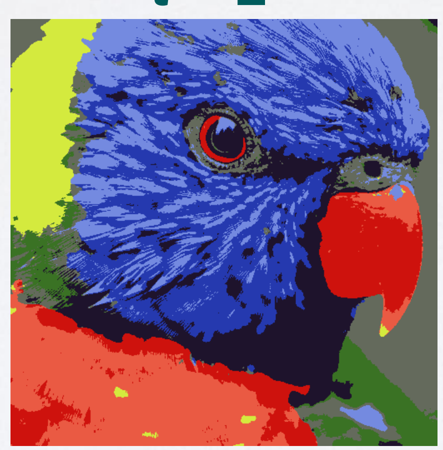
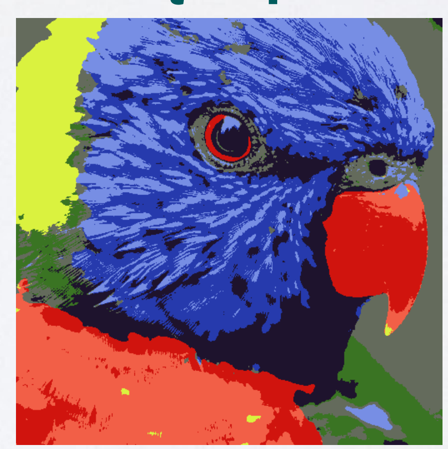
	$n = 4$	$n = 8$	$n = 12$
			
PSNR:	17.7dB	25.0dB	35.6dB
MSE:	1115	224	21
SSIM:	79.8%	94.6%	99.5%

Exact 32-bit div. (reference)



t↓ ⟵⟶ *t*↑

SAADI ($n = 8$)

	$t = 2$	$t = 4$	$t = 6$
			
PSNR:	22.3dB	24.4dB	25.0dB
MSE:	397	260	224
SSIM:	92.7%	94.2%	94.6%

COLOR QUANTIZATION USING *K*-MEANS CLUSTERING

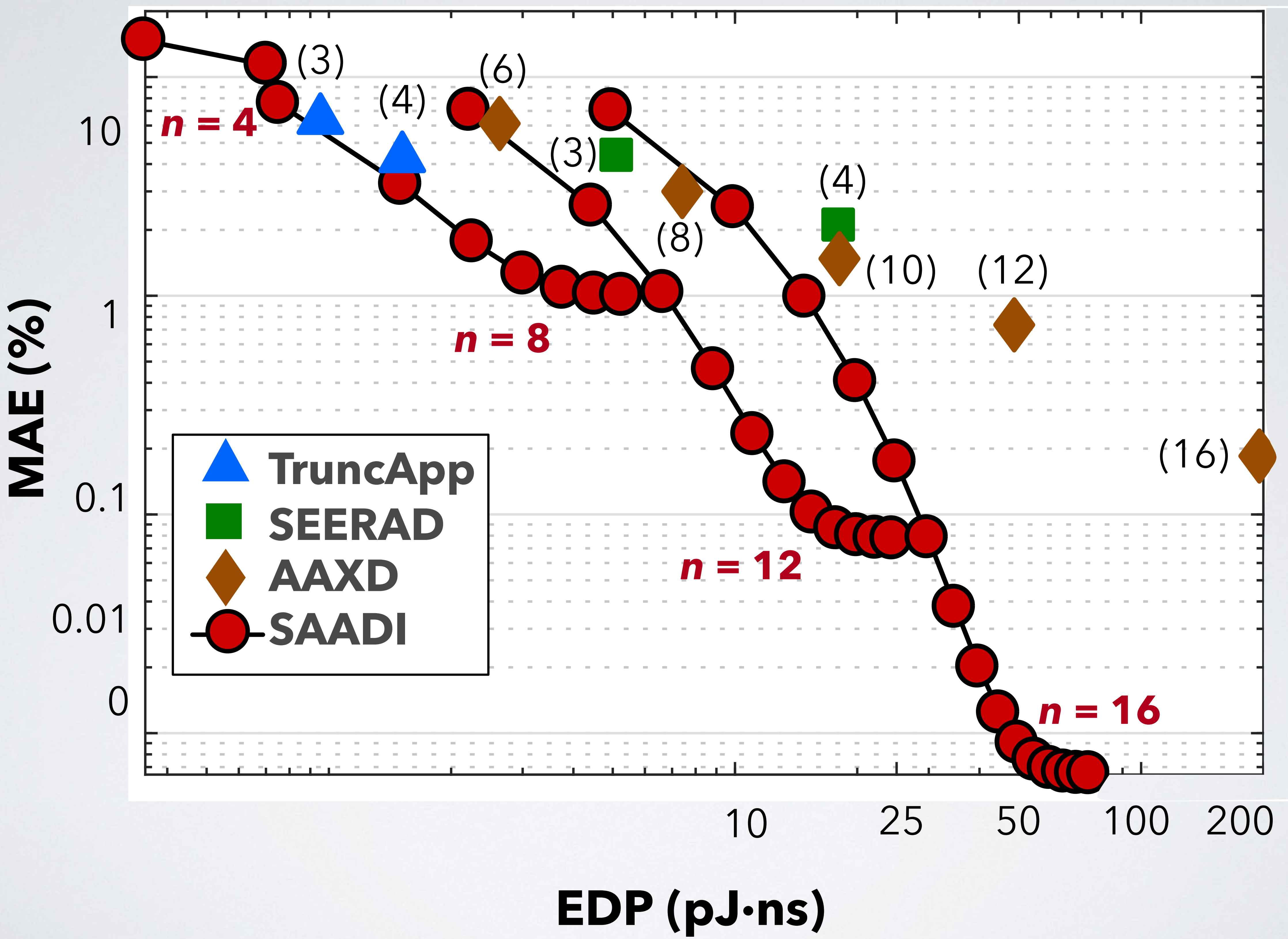
Original image

Exact 32-bit div.
(reference)

SAADI
(n = 8, t = 7)

PSNR:	24.2dB	27.1dB	25.7dB	27.7dB
MSE:	248	126	179	115
SSIM:	79.7%	84.9%	88.9%	96.7%

ENERGY-ACCURACY TRADE-OFF COMPARISON



Conclusions: SAADI

- **"Approximate":** Exploits error resiliency of applications - neural networks, signal processing
- **"Dynamic quality configurability":** First accuracy-scalable divider
- Significant energy saving with minimum accuracy degradation
- 8-bit SAADI achieves average accuracy between 92.5% to 99.0% compared to 32-bit precise divider
- Application demonstrated for image processing