SAADI: A SCALABLE ACCURACY APPROXIMATE DIVIDER FOR DYNAMIC ENERGY-QUALITY SCALING

Setareh Behroozi, Jingjie Li, Jackson Melchert, Younghyun Kim

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

10.012 × 9.9822 = ?

What?

What’s “roughly”?

10.012 × 9.9822

roughly?

100

10.012 × 9.9822

roughly?

99.9417864

What?

99.9417864

10.012 × 9.9822

What’s “roughly”? 

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

Slow, but always efficient

Fast, but sometimes inefficient
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

<table>
<thead>
<tr>
<th>DIV</th>
<th>VS</th>
<th>MUL</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDIV</td>
<td>9-25 cycles (32 bit)</td>
<td></td>
</tr>
<tr>
<td>IMUL</td>
<td>3 cycles (32 bit)</td>
<td></td>
</tr>
</tbody>
</table>

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
Application accuracy requirement varies over time

Dynamic quality configuration

Previous approx. dividers


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 Scalable Accuracy Approximate Divider 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

\[ A = 2^{e_a} \times a \quad B = 2^{e_b} \times b \]

**Divider**

\[ Q = \frac{A}{B} \]

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

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

**Multiplier**

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

**Divide**

**Multiplier**

\[ R(b) = \frac{1}{b} \]

**Approximate Reciprocal**

\[ \tilde{R}(b) \]

\[ A = 2^{e_a} \times a \quad B = 2^{e_b} \times b \]
**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 $1 \leq t \leq n-1$**

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

Runtime accuracy control for dynamic quality configuration
\[ \tilde{R}_t(b) = 1 + |x| + |x|^2 + |x|^3 + |x|^4 + \ldots + |x|^t \]

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

**Design time parameter:** Multiplier width: \( n \)

**Run time parameter:** Number of cycles: \( t \)
<table>
<thead>
<tr>
<th>Cycle</th>
<th>Norm</th>
<th>2’s comp</th>
<th>Multiplier</th>
<th>Accumulator</th>
<th>Adder</th>
<th>Shifter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Norm B</td>
<td>$b \rightarrow</td>
<td>x</td>
<td>$</td>
<td>$</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>Norm A</td>
<td>$</td>
<td>x</td>
<td>^3$</td>
<td>$</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$t-1$</td>
<td></td>
<td></td>
<td>$</td>
<td>x</td>
<td>^t$</td>
<td>$</td>
</tr>
<tr>
<td>$t$</td>
<td></td>
<td></td>
<td>$a \times \tilde{R}_t(b)$</td>
<td></td>
<td>$e_a - e_b$</td>
<td>Shift</td>
</tr>
</tbody>
</table>
Sources of Error

\[ \epsilon_1 \]: Inputs \( A \) and \( B \) normalized to \( n \) bits

\[ \epsilon_2 \]: \( \tilde{R}_t(b) \) is the sum of limited number of \( |x|^t \) terms

\[ \epsilon_3 \]: Each \( |x|^t \) computed by an approximate multiplier

\[ \epsilon_4 \]: \( \tilde{R}_t(b) \) truncated from \( n+2 \) bits to \( n \) bits
SOURCE OF ERROR

![Diagram showing the source of error in a digital signal processing system.]

- $\epsilon_1$: Norm
- $\epsilon_2$: $2'$s comp $|x|$
- $\epsilon_3$: Multiplier
- $\epsilon_4$: $\tilde{R}(b)

Block diagram includes:
- Norm
- Mux
- Adder
- Shifter

Input:
- $B$ to $\epsilon_1$
- $A$ to $\epsilon_1$

Output:
- $\tilde{Q}$

Error contributions:
- $e_b$
- $e_a$

Operations:
- $b$ multiplied by $|x|$
- $+1$ added to $|x|^i$
- $\cdot \tilde{R}(b)$
SAADI Example

\[ B = 11 \quad e_b = -4 \]
\[ b = 0.68750 \]
\[ A = 190 \quad e_a = 0 \]
\[ a = 0.74219 \]

\[ |x| = 0.31250 \quad \tilde{R}_1(b) = 1.31250 \quad \tilde{Q}_1 = 15.5000 \text{ (error: -10.26\%)} \]

\[ |x|^2 = 0.09766 \quad \tilde{R}_2(b) = 1.40625 \quad \tilde{Q}_2 = 16.6250 \text{ (error: -3.75\%)} \]

\[ |x|^3 = 0.02930 \quad \tilde{R}_3(b) = 1.43750 \quad \tilde{Q}_3 = 17.0000 \text{ (error: -1.58\%)} \]

\[ |x|^4 = 0.00781 \quad \tilde{R}_4(b) = 1.44531 \quad \tilde{Q}_4 = 17.1250 \text{ (error: -0.86\%)} \]

\[ |x|^5 = 0.00195 \quad \tilde{R}_5(b) = 1.44531 \quad \tilde{Q}_5 = 17.1250 \text{ (error: -0.86\%)} \]

\[ |x|^6 = 0.00000 \quad \tilde{R}_6(b) = 1.44531 \quad \tilde{Q}_6 = 17.1250 \text{ (error: -0.86\%)} \]

**Exact** \[ R(b) = 1.45455 \]
**Exact** \[ Q = 17.2727 \]
**Experimental Results: Accuracy**

![Graph showing error (%) vs n (bit) with indications of maximum, 75th percentile, median, 25th percentile, and minimum values.](image)
Accuracy

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

MAE for varying $n$ and $t$
### Area, Power, and Delay

<table>
<thead>
<tr>
<th>Bit width n(bit)</th>
<th>Area (μm²)</th>
<th>Delay (ns)</th>
<th>Power (mW)</th>
<th>Energy per cycle (pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1,199</td>
<td>1.07</td>
<td>0.31</td>
<td>0.33</td>
</tr>
<tr>
<td>8</td>
<td>1,963</td>
<td>1.13</td>
<td>0.59</td>
<td>0.66</td>
</tr>
<tr>
<td>12</td>
<td>3,068</td>
<td>1.43</td>
<td>1.09</td>
<td>1.56</td>
</tr>
<tr>
<td>16</td>
<td>4,872</td>
<td>1.60</td>
<td>1.94</td>
<td>3.11</td>
</tr>
</tbody>
</table>

#### Target accuracy:
- **88%**
  - Energy (pJ): 0.66
  - Delay: 1
  - Power: 0.31

- **99%**
  - Energy (pJ): 4.01
  - Delay: 6
  - Power: 1.09

- **99.9%**
  - Energy (pJ): 10.96
  - Delay: 7
  - Power: 1.56
COLOR QUANTIZATION USING K-MEANS CLUSTERING

Original image

Exact 32-bit div. (reference)

SAADI(t = n - 1)

\( n \downarrow \)

\( n = 4 \)

PSNR: 17.7dB
MSE: 1115
SSIM: 79.8%

\( n = 8 \)

PSNR: 25.0dB
MSE: 224
SSIM: 94.6%

\( n = 12 \)

PSNR: 35.6dB
MSE: 21
SSIM: 99.5%

SAADI(n = 8)

\( t \downarrow \)

\( t = 2 \)

PSNR: 22.3dB
MSE: 397
SSIM: 92.7%

\( t = 4 \)

PSNR: 24.4dB
MSE: 260
SSIM: 94.2%

\( t = 6 \)

PSNR: 25.0dB
MSE: 224
SSIM: 94.6%
COLOR QUANTIZATION USING K-MEANS CLUSTERING

Original image

Exact 32-bit div. (reference)

SAADI (n = 8, t = 7)

<table>
<thead>
<tr>
<th></th>
<th>PSNR</th>
<th>MSE</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image</td>
<td>24.2dB</td>
<td>248</td>
<td>79.7%</td>
</tr>
<tr>
<td>Exact 32-bit div. (reference)</td>
<td>27.1dB</td>
<td>126</td>
<td>84.9%</td>
</tr>
<tr>
<td>SAADI (n = 8, t = 7)</td>
<td>25.7dB</td>
<td>179</td>
<td>88.9%</td>
</tr>
<tr>
<td></td>
<td>27.7dB</td>
<td>115</td>
<td>96.7%</td>
</tr>
</tbody>
</table>
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