A Comparative Study of Approximate Adders and Multipliers

Honglan Jiang*, Cong Liu*, Naman Maheshwari#, Fabrizio Lombardi§ and Jie Han*

* Department of Electrical and Computer Engineering
  University of Alberta, Edmonton, AB, Canada,
# Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani, Rajasthan, India and
§ Department of Electrical and Computer Engineering, Northeastern University, Boston, USA.
Outline

- Motivation and Introduction
- Review and Classification of Approximate Adders
- Comparison of the Approximate Adders
  - Error Characteristics
  - Circuit Characteristics
- Review and Classification of Approximate Multipliers
- Comparison of the Approximate Multipliers
  - Error Characteristics
  - Circuit Characteristics
- Conclusion
Motivation

- The physical dimensions of CMOS devices have been scaling and approaching a few nanometers.
  - Improving circuit performance of digital circuits becomes increasingly difficult.
  - Energy-efficiency is of paramount concern in digital system design.

- Computing becomes increasingly heavy with multimedia processing (audio, video, graphics, and image), recognition, search, machine learning and data mining.

- A common characteristic: a perfect result is not necessary and an approximate or less-than-optimal result is sufficient
  - Human perception is not sensitive to high frequency changes.
  - Natural noise floor due to quantization noise.
Error-Resilient Paradigms

- How can we exploit a system’s ability for imprecision-tolerance and energy reduction?

- **Approximate Computing**
  - Does not involve assumptions on the stochastic nature of any underlying processes implementing the system. Utilizes statistical properties of data and algorithms to trade quality for energy reduction.

- **Stochastic Computing**
  - Real numbers are represented by random binary bit streams that are usually implemented in series (or parallel) and in time (or space). Information is carried on the statistics of the binary streams.

- **Probabilistic Computing**
  - Exploits intrinsic probabilistic behavior of the underlying circuit fabric, most explicitly, of the stochastic behavior of a binary switch under the influence of thermal noise.

Effort in approximate computing covers a broad spectrum of research, ranging from those addressing issues at circuit and system levels, up to those at software and application levels.

We focus on approximate hardware design and, in particular, approximate arithmetic circuits of adders and multipliers, or approximately designed adders and multipliers.
Ripple-Carry Adder (RCA)

The $n$-bit ripple-carry adder.

$$s_i = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

The schematic of a full adder (FA).

Critical path: $O(n)$  
Circuit area: $O(n)$
Carry Lookahead Adder (CLA)

An n-bit carry lookahead adder.

\[ s_i = a_i \oplus b_i \oplus c_i \]
\[ g_i = a_i b_i \]
\[ p_i = a_i + b_i \]

Critical path: \( O(\log(n)) \)
Circuit area: \( O(n\log(n)) \)

The schematic of the sum, propagate and generate signal generator (SPG).
Approximate Adders: A Classification

We classify the approximate adders into four categories:

- **Speculative Adders**
  - For a 128-bit adder, the probability that the carry propagation chain is longer than 12 and 18 are 1\% and 0.01\%, respectively.*
  - Therefore, $k$ bits are used to speculate the carry for each sum bit ($k < n$).

- **Segmented Adders**
  - An $n$-bit adder is divided into a number of smaller $k$-bit sub-adders.
  - The carry may be generated by using different methods.

- **Carry-Select Adders**
  - Multiple sub-circuits are used to compute the sum for different carry values, and the result is selected by the carry of a sub-circuit.

- **Approximate Full Adders**

Speculative Adders

The almost correct adder (ACA):

The $n$-bit almost correct adder (ACA).

**Critical path:** $O(\log(k))$  \hspace{1cm} **Circuit area:** $O((n - k)k\log(k))$

Segmented Adders (1)

The equal segmentation adder (ESA):

The *n*\nobit equal segmentation adder (ESA) \((l \leq k)\).

**Critical path:** \(O(\log(k))\)  
**Circuit area:** \(O(n\log(k))\)

Segmented Adders (2)

The error-tolerant adder type II (ETAII):

The $n$-bit error-tolerant adder type II (ETAII).

**Critical path:** $O(\log(k))$  
**Circuit area:** $O(n\log(k))$

Carry Select Adders

The speculative carry selection adder (SCSA):

**Critical path:** $t_{adder} + t_{mux}$

- $t_{adder}$: $O(\log(k))$
- $t_{mux}$: delay of the multiplexer

**Circuit area:** $A_{adder} + A_{mux}$

- $A_{adder}$: $O(n\log(k))$
- $A_{mux}$: circuit area of the multiplexer

Approximate Full Adders

A general schematic:

The \(n\)-bit approximate adder using approximate full adders

**Critical path:***

\[ t_{\text{approximate adder}} + t_{\text{accurate adder}} \]

**Circuit area:***

\[ A_{\text{approximate adder}} + A_{\text{accurate adder}} \]
Approximate Mirror Adders (AMAs)

The conventional mirror adder (MA).

The mirror adder approximation 1 (AMA1).

The truth table for AMA1.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>(C_{in})</th>
<th>Sum’</th>
<th>(C_{out1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Lower-part OR Adders (LOAs)

The $n$-bit lower-part-OR adder (LOA).

**Critical path:**

$O(\log(n - l))$

$A_{adder}: O((n - l)\log(n - l))$

**Circuit area:**

$A_{adder} + (l \times A_{OR})$

$A_{OR} :$ circuit area of the OR gate.

## A Brief Summary

Analysis of delay and circuit complexity of approximate adders.

<table>
<thead>
<tr>
<th>Adder Type</th>
<th>Adder Name</th>
<th>Delay</th>
<th>Circuit Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional Adders</td>
<td>RCA</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td></td>
<td>CLA</td>
<td>$O(log(n))$</td>
<td>$O(n\cdot log(n))$</td>
</tr>
<tr>
<td>Speculative Adders</td>
<td>ACA [3]</td>
<td>$O(log(k))$</td>
<td>$O((n - k)\cdot k\cdot log(k))$</td>
</tr>
<tr>
<td></td>
<td>ESA [6]</td>
<td>$O(log(k))$</td>
<td>$O(n\cdot log(k))$</td>
</tr>
<tr>
<td></td>
<td>ETAII [4]</td>
<td>$O(log(k))$</td>
<td>$O(n\cdot log(k))$</td>
</tr>
<tr>
<td></td>
<td>ACAA [5]</td>
<td>$O(log(k))$</td>
<td>$O((n - k)\cdot log(k))$</td>
</tr>
<tr>
<td>Segmented Adders</td>
<td>SCSA [7]</td>
<td>$t_{adder} + t_{mux}$</td>
<td>$A_{adder} + A_{mux}$</td>
</tr>
<tr>
<td></td>
<td>CSA [8]</td>
<td>$O(log(k))$</td>
<td>$A_{adder} + A_{carry}$</td>
</tr>
<tr>
<td></td>
<td>CSPA [10]</td>
<td>$t_{adder} + t_{mux}$</td>
<td>$A_{adder} + A_{mux} + A_{carry}$</td>
</tr>
<tr>
<td></td>
<td>CCA [11]</td>
<td>$t_{adder} + t_{mux}$</td>
<td>$A_{adder} + A_{mux}$</td>
</tr>
<tr>
<td></td>
<td>GCSA [12]</td>
<td>$O(log(k))$</td>
<td>$O(n\cdot log(k))$</td>
</tr>
<tr>
<td>Approximate Full Adders</td>
<td>LOA [13]</td>
<td>$O(log(n - l))$</td>
<td>$A_{loa} + (l \times A_{OR})$</td>
</tr>
</tbody>
</table>

$t_{adder}: O(log(k))$  
$A_{adder}: O(n\cdot log(k))$  
$A_{loa}: O((n - l)\cdot log(n - l))$  
$A_{carry}: circuit area of the carry prediction circuit$

- ESA has the smallest delay and circuit area.
- ETAII, ACAA and SCSA have the same accuracy; ETAII is the most efficient design among them.
Error Metrics

- **Error rate (ER)** is the probability of producing an incorrect result.

- **Error distance (ED)** is the arithmetic distance between an approximate result and the accurate result.

  If $M'$ and $M$ are the approximate and accurate results, $ED = |M' - M|$. 

- **Relative error distance (RED)** is used to evaluate the relative difference between an approximate result and the accurate result.

  For $M'$ and $M$, $RED = \frac{ED}{M}$. 
Error Metrics (cont’d)

- **Mean error distance (MED)** considers the average error distance for multiple inputs.
  - The MED increases exponentially with the number of approximate bits in an adder.

- **Normalized mean error distance (NMED)** is the normalization of MED by the maximum output value.
  - The NMED is a nearly invariant metric independent of the size of an adder.

- **Mean relative error distance (MRED)** assesses the average relative error distance for multiple inputs.

Simulation Results of Approximate Adders

- 16-bit adders are simulated for all approximate designs.

- Each adder’s name is followed by the value of its parameter $k$.
  - $k$ is the size of the sub-adder for ACA, ETAII, ESA, CSA, CSPA, CCA and GCSA.
  - $k$ is the size of the less significant adder for LOA.

- 100,000,000 random input combinations are simulated by MATLAB.

- The NMED and MRED show the same trend.
- ETAII, SCSA and ACAA have the same error characteristics (ER, NMED and MRED) due to the same carry propagation chain for each sum bit.
Error Characteristics of Approximate Adders

- LOA has a rather small MRED but very large ER.
- CSA-5 and CSA-4 is the most accurate.
- GCSA-5 and GCSA-4 are the second most accurate.
- The information used to predict each carry in ESA, CSPA and ACA is rather limited, so the MRED and ER of ESA are the largest, followed by CSPA and ACA, when the same value of k is considered.
- CCA, ETAII, SCSA and ACAA shows moderate MRED and ER.

The MRED and ER of approximate adders, sorted by ER.

Circuit Characteristics of Approximate Adders

- All adders and sub-adders in the approximate designs are implemented as CLA.
- Synthesized by Synopsys Design Compiler based on an STM 28-nm process with a supply voltage of 1.0V at a temperature of 25 ºC.

The power and area of approximate adders, sorted by power.

- A circuit with larger area is likely to consume more power except for CSA with low power dissipation but large area.
- This is due to its short critical path and enhanced carry select scheme and thus, complex wiring.
Circuit Characteristics of Approximate Adders

Among ETAII, SCSA and ACAA (with the same accuracy), SCSA albeit being the fastest, incurs the largest power dissipation and area, and ACAA is the slowest because of its long critical path \((2k)\). ETAII has a shorter delay than ACAA and consumes less power and requires a smaller area than SCSA.

The accurate CLA has the longest delay, but not the highest power dissipation.

LOA is very slow, but it is the most power and area efficient.

Except for LOA and for the same \(k\), ACA is the fastest and power-wasting, ESA is pretty fast and power and area efficient, ACAA is the slowest, and CCA is the most power and area consuming scheme.

Both CSPA and GCSA have moderate power dissipations, but CSPA is faster and GCSA is more area efficient.
Considering Both Accuracy and Hardware

• For the adders with the same accuracy, ETAII, SCSA and ACAA, ETAII has the lowest PDP (except for ETAII-6), while SCSA has the highest.
• CSA shows the best performance with very small PDP, ER and MRED.
• ESA has a rather small PDP but a considerably large ER and MRED.
• ACA has very small PDP and slightly lower ER and MRED than ESA.
• CCA has the largest PDP and moderate ER and MRED.
• LOA shows small PDPs, and its MREDs are moderate but with the highest ER.
Conclusion on Approximate Adders

- In general, approximate speculative adders show moderate accuracy and relatively small PDPs.
- The approximate adders using approximate full adder in the LSBs are slow, but they are power efficient with high ERs (because of the approximate LSBs) and moderate NMED and MRED values because of the accurate MSBs).
- The error and circuit characteristics of the segmented and carry select adders vary with the predictions of carry signals.
Outline

- Motivation and Introduction
- Review and Classification of Approximate Adders
- Comparison of the Approximate Adders
  - Error Characteristics
  - Circuit Characteristics
- Review and Classification of Approximate Multipliers
- Comparison of the Approximate Multipliers
  - Error Characteristics
  - Circuit Characteristics
- Conclusion
Multiplier: Wallace tree

The basic arithmetic operation of a 4 x 4 bit unsigned multiplier

Critical path: $O(\log(n))$
Multiplier: Carry-Save Adder Array

The partial product accumulation structure for a 4 x 4 bit unsigned multiplier using a carry-save adder array

Critical path: $O(n)$
Approximate Multipliers: A Classification

We classify the approximate multipliers into four categories:

- **Approximation in Generating Partial Products**
  - Using simpler structure to generate partial products.

- **Approximation in the Partial Product Tree**
  - Omitting some partial products.
  - Dividing partial products into several sections and applying approximation in the less significant sections.

- **Using Approximate Counters or Compressors in the Partial Product Tree**
  - Approximating adders, counters or compressors

- **Approximate Booth Multipliers**
Approximation in Generating Partial Products

The Underdesigned Multiplier (UDM):

K-Map for the 2 x 2 bit underdesigned multiplier block

<table>
<thead>
<tr>
<th></th>
<th>B₁B₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>A₁A₀</td>
<td>00</td>
</tr>
<tr>
<td>00</td>
<td>000</td>
</tr>
<tr>
<td>01</td>
<td>000</td>
</tr>
<tr>
<td>11</td>
<td>000</td>
</tr>
<tr>
<td>10</td>
<td>000</td>
</tr>
</tbody>
</table>

A 4 x 4 bit multiplier built on 2 x 2 bit block.

Approximation in the Partial Product Tree (1)

The Broken-Array Multiplier (BAM):

Structure of the Broken-Array Multiplier.

Approximation in the Partial Product Tree (2)

The Error-Tolerant Multiplier (ETM):

Architecture of a 16 x 16 bit Error-Tolerant Multiplier.

Approximation in the Partial Product Tree (3)

Approximate Wallace Tree Multiplier (AWTM):

Architecture of an Approximate Wallace Tree Multiplier.

Approximate Counters or Compressors (1)

Inaccurate Counter based Multiplier (ICM):

K-Map for the inaccurate 4:2 counter for 4 x 4 bit Wallace multiplier

<table>
<thead>
<tr>
<th>X_3X_4</th>
<th>X_1X_2</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>CS</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>00</td>
<td>01</td>
<td>10</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>01</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Larger Multipliers are implemented by the inaccurate 4 x 4 bit multiplier.

Approximate Counters or Compressors (2)

Approximate Compressor based Multiplier (ACM):

Approximate 4-2 compressor design 1 (AC1): altering 12 out of 32 K-Map entries.

Approximate 4-2 compressor design 2 (AC2): altering 4 out of 16 K-Map entries.

ACM-3: AC1 in LSBs and accurate compressors in MSBs in a Dadda multiplier. ACM-4: AC2 in LSBs and accurate compressors in MSBs in a Dadda multiplier.

Approximate Counters or Compressors (3)

Approximate Multiplier (AM) with Configurable Partial Error Recovery and Truncated AM (TAM):

- Two error accumulation trees are designed for AM1 and AM2.
- TAM1 and TAM2 are obtained by truncating some LSBs.

The approximate adder cell.

The approximate multiplier with 4-bit error recovery.

C. Liu, J. Han, and F. Lombardi, “A low-power, high-performance approximate multiplier with configurable partial error recovery,” DATE, 2014.
Simulation Results of Approximate Multipliers

- 16 x 16 bit multipliers are simulated for all approximate designs.
- Each multiplier’s name is followed by the value of its parameter $k$.
  - $k$ is the number of MSBs used for error reduction in AM1, AM2, TAM1 and TAM2.
  - $k$ is the number of LSBs in the inaccurate part for ETM.
  - $k$ is the mode number in AWTM and ACM.
  - It is the vertical broken length for BAM.
- 100,000,000 random input combinations are simulated by MATLAB.

The ER and NMED of approximate multipliers, sorted by NMED.

- **ICM** has a very low ER of 5.45%, because it uses just one approximate compressor in a 4 × 4 bit sub-multiplier with an error rate of only 1/256.
- Most of the designs, especially those with truncation, have large ERs (nearly 100%).
Error Characteristics of Approximate Multipliers

- ICM, AM2-15 and TAM2-16 have close NMED values, however ICM has the smallest MRED while the MRED of TAM2-16 is the largest.
- Multipliers with truncation (TAM2-16 and BAM-18) tend to have larger MREDs when NMEDs are similar.
- ACM-4, ACM-3 and AWTM-4 achieve very low NMEDs because only LSBs are approximated in them.
- ETM and BAM have relatively large MREDs due to truncation.

The MRED and NMED of approximate multipliers, sorted by MRED.
Circuit Characteristics of Approximate Multipliers (1)\textsuperscript{36}

- 16 x 16 bit multipliers are implemented in VHDL for all approximate designs.

- Synthesized by Synopsys Design Compiler based on an STM 28-nm process with a supply voltage of 1.0V at a temperature of 25 °C.

- A multiplier with larger area is likely to consume more power.

The power and area of approximate multipliers, sorted by power.
Circuit Characteristics of Approximate Multipliers (2)  

- ArrayM is the slowest.
- WallaceM is the most power consuming.
- AM1, TAM1, AM2 and TAM2 have smaller delays even with a 16-bit error reduction.
- BAM is significantly slow due to its array structure.
- AWTM, UDM, ICM and ACM have larger delays than the other multipliers.
- BAM consumes very low power, the power consumption of AWTM and ACM are in the middle range, while UDM and ICM incur relatively high power consumption.

ETM, TAM1/TAM2 and BAM are among the most power efficient designs.

Truncation is a useful approach to reduce power and area.
Considering Both Accuracy and Hardware

- TAM1-13, TAM-16, TAM2-13 and BAM-18 have both small PDPs and MREDs.
- ICM and ACM incur a very low error, but their PDPs are very high.
- BAM-22 has the smallest PDP but largest MRED.
- ETM-8 has the smallest PDP but significantly large MRED.
- UDM shows poor performance in both PDP and MRED.
- ETM has a small PDP and a relatively large MRED.
- Most BAM configurations have small PDPs, their delays are generally large.
Conclusion on Approximate Multipliers

- Truncation on part of the partial products is an effective methodology to save hardware. However, it incurs a large ER but moderate NMED and MRED.

- Approximate multipliers implemented by smaller approximate sub-multiplicants may have very low ERs (e.g., UDM and AWTM-3), but their NMEDs and MREDs are usually large because of the large errors that may occur in the more significant part of the multiplier. Moreover, they usually have rather high PDPs.
References


References


References


References


Thanks for your attention. Questions?