香港中文大學
The Chinese University of Hong Kong

CENG5030

Part 1-5: Approximate Computing

Bei Yu

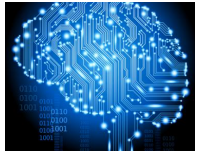
(Latest update: March 25, 2019)

Spring 2019

These slides contain/adapt materials developed by

- Animesh Jain and Parker Hill (2016). *Lecture on Approximate Computing*. Tech. rep. University of Michigan
- Jie Han and Michael Orshansky (2013). "Approximate computing: An emerging paradigm for energy-efficient design". In: *Proc. ETS*, pp. 1–6

What is Approximate Computing?



Machine learning

Data mining

Image processing

- Many applications are error tolerant
  - Application input are noisy e.g. sensors
  - Application outputs are probabilistic estimates e.g. machine learning
  - User facing application output e.g. Images
- Opportunity – skipping/inexactly performing computation results in little accuracy loss
- Goal – tradeoff little application accuracy with significant performance/energy improvements (e.g. 10% loss with 2x speedup)

Basic Approximation Techniques

- Software - Loop Perforation (e.g. PARSEC benchmarks)
  - Work skipping - Skip some iterations of the loop
  - Ideally linearly improves application performance
  - Accuracy implications changes from application to application
- Architecture - Precision Reduction
  - Applications do not require all the bits in the floating-point data elements
  - Remove bits using SW/HW techniques
- Device level – Approximate storage
  - Use new memory technology to store the data approximately

Challenges

- Where to Approximate?
- Controlling Accuracy
- Designing Good Approximation Techniques

Challenges – Where to Approximate?

- Where to approximate?
  - Code segments have some critical portions e.g. control flow
  - Approximating these portions can even lead to application crash
  - Problem – Identify approximation-amenable code segment
  - Problem – defining the interface when the application execution moves from exact to approximation to exact portions
  - Lot of programming language research in this area

Challenge – Controlling Accuracy

- Accuracy is affected at all abstraction levels
- Across applications
  - Different applications have different tolerance levels
- Within application
  - Different variables affect application accuracy differently
  - Approximation at one code segment affects later code segment accuracy
- Input sensitivity for the same application
  - Some inputs are hard to approximate as compared to others
  - Same approximation technique can lead to different accuracy for different inputs

Need to have dynamic knobs to control the accuracy in different scenarios

Challenges – Approximation Techniques

- High expectation on performance/energy improvements
- Technique might be very tightly dependent on architectural/micro-architectural specifications
  - For example – applications using vector registers need special treatment

- Overall, technique should
  - Have high performance/energy savings
  - Be flexible to adapt to application accuracy needs
  - Have minimal hardware overhead

Readling List 1

- Software Level: Loop Perforation[1]
- Architecture Level: Precision Reduction[2]
- Device Level: Approximate Memory[3]

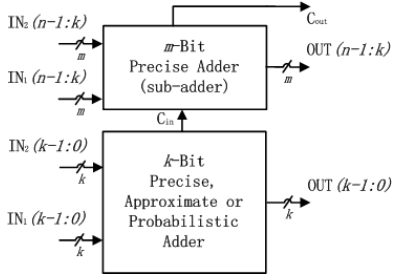
[1] Sasa Misailovic et al. (2010). “Quality of service profiling”. In: *Proc. ICSE*, pp. 25–34.

[2] Thomas Yeh et al. (2007). “The art of deception: Adaptive precision reduction for area efficient physics acceleration”. In: *Proc. MICRO*, pp. 394–406.

[3] Adrian Sampson et al. (2013). “Approximate Storage in Solid-state Memories”. In: *Proc. MICRO*, pp. 25–36.

Approximate *n*-bit Adders

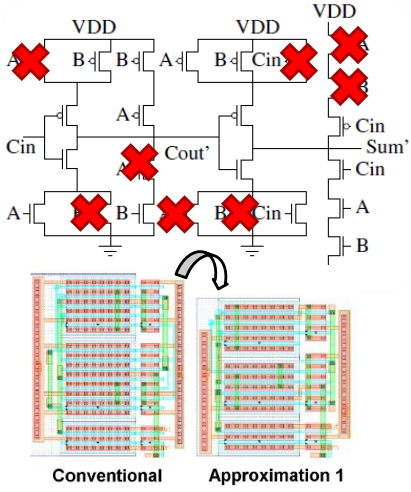
- In an approximate implementation, *n*-bit adders can be divided into two modules: the (accurate) upper part of more significant bits and the (approximate) lower part of less significant bits.
- For each lower bit, a single-bit approximate adder implements a modified, thus inexact, function of the addition.



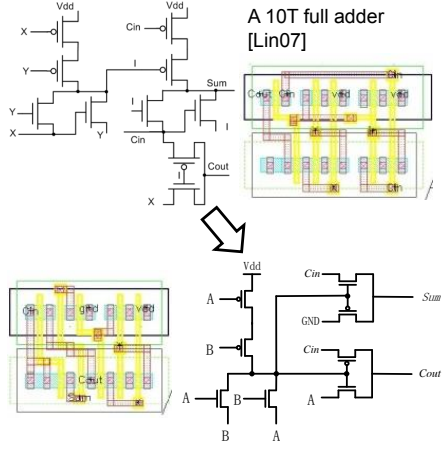
A general architecture for an approximate adder divided into two modules: the accurate MSBs and approximate LSBs.

Approximate Full Adders

- Approximate Mirror Adders (AMAs) [Gupta13]



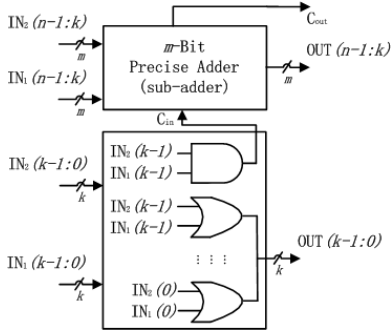
- Approximate XOR/XNOR Adders (AXAs) [Yang13]



Approximate and Probabilistic Adders

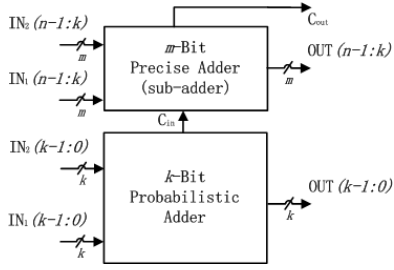
Approximate logic design

- Lower-part OR adder [Mahdiani10]



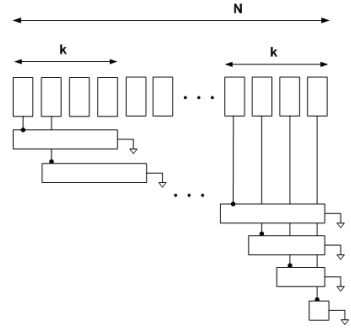
Probabilistic adder

- PCMOS-based design [Cheemalavagu05]



Approximate Speculative Adders (1)

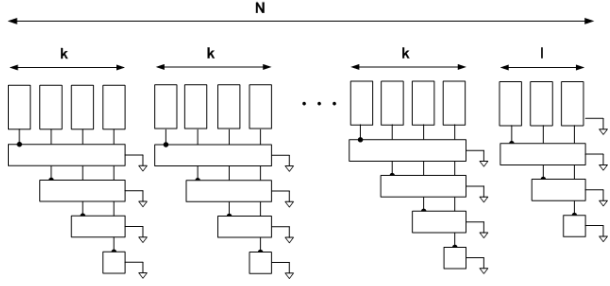
- The critical path delay of a parallel adder (such as a carry look ahead) is asymptotically proportional to log(*N*) for an *N*-bit adder.
- Sub-logarithmic delays can however be achieved by the so-called speculative adders [Lu04, Verma08].
- A speculative adder exploits the fact that the typical carry propagation chain is significantly shorter than the worst-case carry chain by using a limited number of previous input bits to calculate the sum (e.g. look-ahead *k* bits) [Lu04].
- It can be developed into a reliable variable latency speculative adder (VLSA) with error detection and recovery [Verma08].



A speculative adder as an almost correct adder (ACA).

Approximate Speculative Adders (2)

- An error tolerant adder truncates the carry propagation chain by dividing the adder into several sub-adders (ETAII); its accuracy can be improved by connecting carry chains in a few most significant sub-adders (ETAIIM) [Zhu09].
- An alternating carry select process can be used in the sub-adder chain to enhance the design (ETAIV) [Zhu10].



A general architecture of an error tolerant adder (ETA).

Approximate Speculative Adders (3)

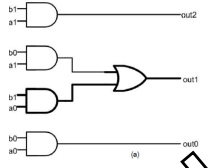
- A reliable variable latency carry select adder (VLCSA) employs carry chain truncation and carry select addition as a basis in a speculative adder [Du12].
- An accuracy-configurable adder (ACA) enables an adaptive operation, either approximate or accurate, that is configurable at runtime [Kahng12].
- In a dithering adder, subsequent additions produce opposite-direction errors such that the final result has a smaller overall error variance [Miao12].
- More details discussed later.

Approximate Multipliers (1)

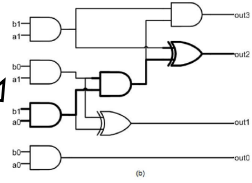
- A multiplier usually consists of three stages: partial product generation, partial product accumulation and a carry propagation adder at the final stage.
  - The use of speculative adders in an approximate multiplier to compute the sum of partial products is not efficient in terms of trading off accuracy for energy and area savings [Lu04, Huang12].
- In [Kulkarni11], inaccurate 2 × 2 multiplier blocks are used to compute approximate partial products that are accumulated using accurate adders.

A_1A_0 \ B_1B_0	00	01	11	10
00	000	000	000	000
01	000	001	011	010
11	000	011	111	110
10	000	010	110	100

Truth table for the approximate 2 × 2 multiplier



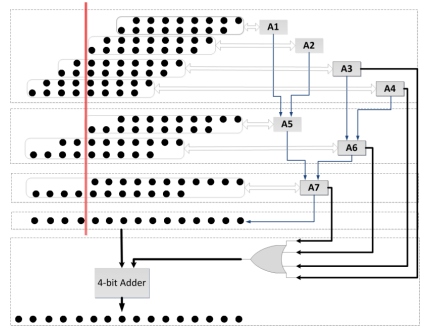
(a) Inaccurate 2 × 2 multiplier



(b) Accurate 2 × 2 multiplier

Approximate Multipliers (2)

- A significant design aspect is to reduce the critical path delay in an approximate multiplier.
- A high-performance approximate multiplier with configurable partial error recovery is proposed for DSP applications [Liu13].
  - This multiplier leverages a newly-designed approximate adder that limits its carry propagation to the nearest neighbors for fast partial product accumulation.
  - Different levels of accuracy can be achieved through a configurable error recovery by using different numbers of MSBs for error reduction.
  - Similar performance as exact multipliers in image processing applications.



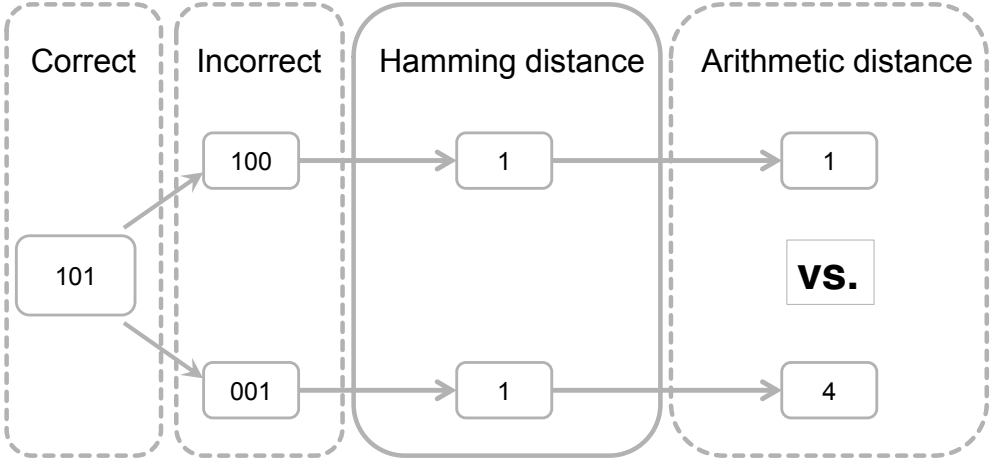
An approximate multiplier with partial error recovery

New Metrics for Approximate Circuits

- The traditional metric of reliability is defined as the probability of correct circuit function:
  - Reliability of any approximate circuit is 0 for some inputs.
- New metrics are needed to assess the reliability of approximate circuits.
  - *Error rate* (ER) or *error frequency* is the fraction of incorrect outputs out of a total number of inputs in an approximate circuit [Breuer04].
  - *Error significance* (ES) refers to the degree of error severity due to the approximate operation of a circuit [Breuer04], as
    - the numerical deviation of an incorrect output from a correct one [Shin10],
    - the Hamming distance of the two vectors [Kahng12],
    - the maximum *error magnitude* of circuit outputs [Miao12].
  - A composite quality metric is the product of ER and ES [Shin11, Chong06].
  - Other common metrics include the relative error, average error and error distribution.

Error Distance for Approximate Circuits

- Error distance is defined as the arithmetic distance between an inexact output and the correct output for a given input [Liang13].

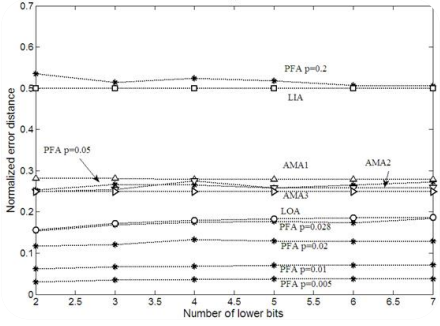


Mean and Normalized Error Distances

- *Mean error distance* (MED) considers the averaging effect of multiple inputs.
  - The MED is useful in measuring the implementation accuracy of a multiple-bit adder, but its value increases exponentially with the number of approximate bits in an adder.
- *Normalized error distance* (NED) is the normalization of MED for multiple-bit adders.
  - The NED is a nearly invariant metric independent of the size of an adder, so it is useful when characterizing the reliability of a specific design of full adders.
- MED or NED can be used with power or energy for evaluating the tradeoff between power consumption and precision in an approximate circuit.

NED as a Metric for Approximate adders

- The normalized error distance (NED) is almost independent of the number of approximate bits.

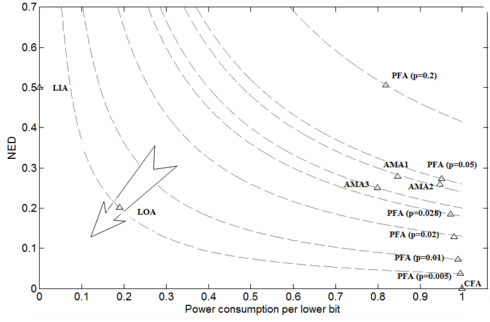


Normalized error distance (NED) vs. the number of approximate bits in an adder.

- It provides an effective alternative to an application-specific metric such as the peak signal-to-noise ratio (PSNR).

Power and Accuracy Tradeoffs

- The product of power and NED can be utilized for evaluating the tradeoff between power consumption and precision.
  - To emphasize the significance of a particular metric (such as the power or precision), a different measure with more weight on this metric can be used for a better assessment of a design according to the specific requirement of an application.



Power and precision tradeoffs: the product of power per bit and NED is shown by a dashed curve. The arrow points to the direction for a better design with a more efficient power and accuracy tradeoff.

Approximate Circuit: Readling List

- Shih-Lien Lu (2004). “Speeding up processing with approximation circuits”. In: *Computer* 37.3, pp. 67–73
- Vaibhav Gupta et al. (2013). “Low-power digital signal processing using approximate adders”. In: *IEEE TCAD* 32.1, pp. 124–137
- Jinghang Liang, Jie Han, and Fabrizio Lombardi (2013). “New metrics for the reliability of approximate and probabilistic adders”. In: *IEEE Transactions on Computers* 62.9, pp. 1760–1771
- Rong Ye et al. (2013). “On reconfiguration-oriented approximate adder design and its application”. In: *Proc. ICCAD*, pp. 48–54