Computer Architecture

Keivan Navi

Cal Poly Pomona University

knavi@cpp.edu

Office hours: Tu/Th 5:25 Pm to 6:55 Pm

Office: 8-49

Computer Architecture

Approximate Computation

Approximate Computation

- Approximate computing is a design paradigm that uses less accurate functions to improve performance and reduce power consumption. It's based on the idea that many applications can produce results that are more accurate than needed, wasting resources.
- Approximate computing can be used in situations where an approximate result is sufficient, such as search engines or video applications. For example, in a search engine, there may not be an exact answer to a search query, so many answers may be acceptable. In a video application, dropping some frames occasionally may go undetected due to human perceptual limitations.

Approximate Computation (continued)

- Some techniques used in approximate computing include: Unreliable hardware, Spy Fly, Sensor networks, Numerical approximations, Mobile ATM and Neural network accelerators.
- Approximate computing can be used to improve the energy efficiency of computing systems at different abstraction levels. For example, in the k-means clustering algorithm, allowing a 5% loss in classification accuracy can result in a 50 times energy saving.

Approximate Half Adder

- The Truth table of a Half Adder follows:
- C=AB
- S=A XOR B

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- The Truth table of approximate Half Adder follows:
- C=A XNOR B
- S=A XOR b

A	B	C	S
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

Approximate HA+

- The Truth table of HA+ follows:
- C= A+B
- S= A XOR B

A	B	C	S
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	1

- The Truth table of Approximate HA+ Follows:
- C= A XOR B
- S= A XNOR B

A	B	C	S
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Approximate HA-

- The Truth table of HA- follows:

A	B	C	S
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	1

- C= (A+B)’=$\overline{A + B}$
- S= A XNOR B
- The Truth table of Approximate HA- follows:

A	B	C	S
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

- C= A XNOR B
- S= A XNOR B

Approximate HA- -

- The Truth table of HA- - follows:

A	B	C	S
0	0	1	0
0	1	1	1
1	0	1	1
1	1	0	0

- C= (AB)’ = $\overline{AB}$
- S= A XOR B
- The truth table of approximate HA - - Follows:

A	B	C	S
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

- C= A XOR B
- S= A XOR B

Approximate Full Adder (Version 1)

- The Truth table od a FA follows:
- C= AB+AC+BC= Majority(A,B,C)
- S= A XOR B XOR C

a	b	c	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- The truth table of approximate FA follows:
- C= AB+AC+BC = Majority(A,B,C)
- S =(AB+AC+BC)'= Minority(A,B,C)

a	b	c	c	s
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Approximate Full Adder (Version 2)

- The Truth table od a FA follows:
- C= AB+AC+BC= Majority(A,B,C)
- S= A XOR B XOR C

a	b	c	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- The truth table of approximate FA follows:
- C= A XNOR B XNOR C
- S = A XOR B XOR C

a	b	c	c	s
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Approximate FA- - (Version 1)

- The Truth table of FA - - follows:

a	b	c	c	s
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

- C= (AB+AC+BC)’=$\overline{AB + AC + BC}$
- S= A XOR B XOR C
- The truth table of approximate FA - - follows:
- C= A XOR B XOR C
- S= A XOR B XOR C

a	b	c	c	s
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Approximate FA- - (Version 2)

- The Truth table of FA - - follows:
- C= (AB+AC+BC)’=$\overline{AB + AC + BC}$
- S= A XOR B XOR C

a	b	c	c	s
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

- The truth table of approximate FA - - follows:
- C= (AB+AC+BC)’=$\overline{AB + AC + BC}$
- S= (AB+AC+BC)’=$\overline{AB + AC + BC}$

a	b	c	c	s
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Most significant bit(s) or least significant bit(s) ?

- Imagine that a this is a bank account in binary representation:
- 1000000000000000000000000000000000000001
- What will happen if the least significant bit is changed to 0?
- What will happen if the most significant bit is changed to 0?
- Which bit (s) do you prefer to be approximated?

Most significant bits or least significant bits?

- Imagine that a this is my bank account in binary representation:
- 00000000000000000000000000000000
- What will happen if the least significant bit is changed to 1?
- What will happen if the most significant bit is changed to 1?
- Which bit (s) do you prefer to be approximated?