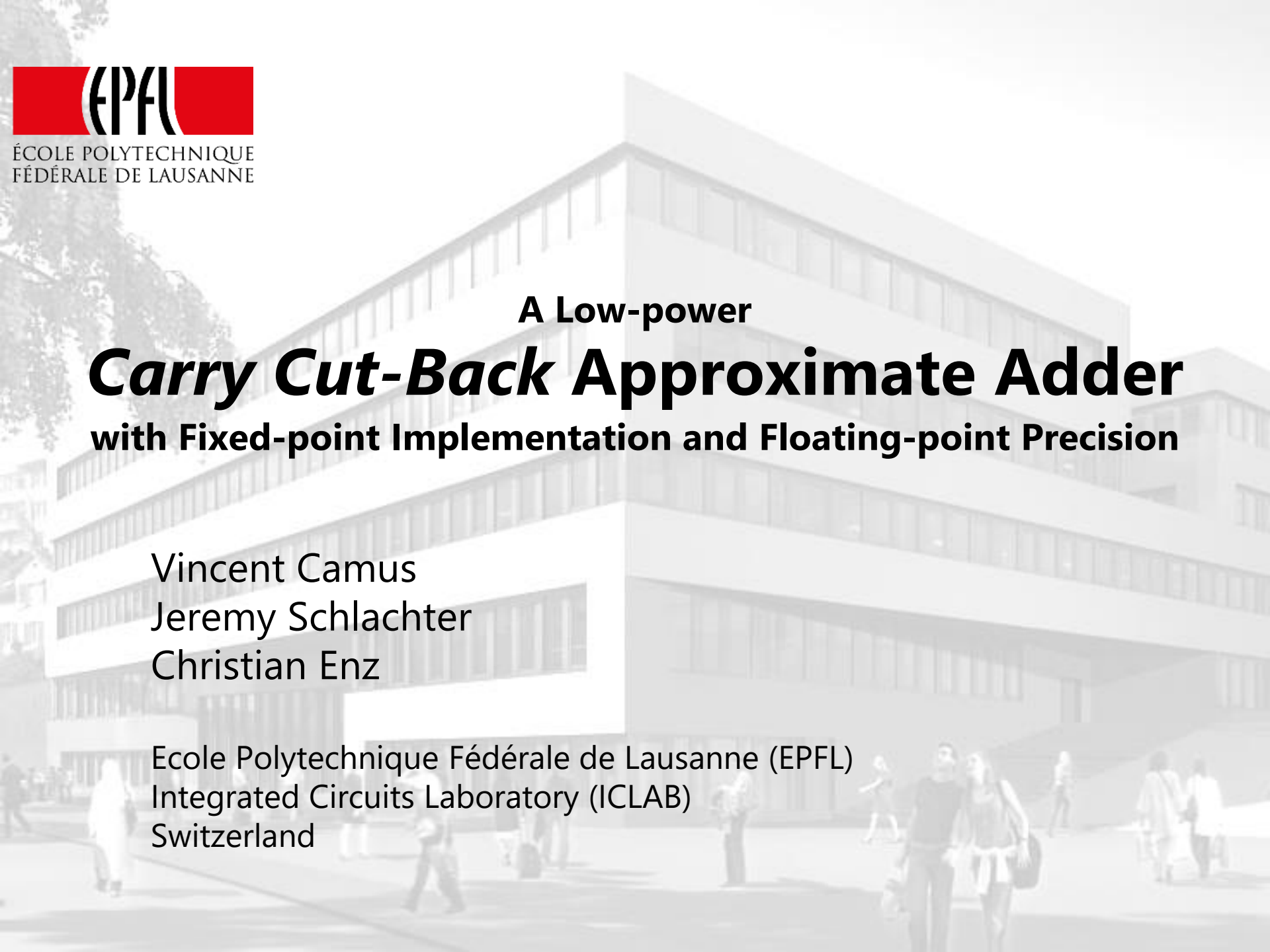


ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

A Low-power *Carry Cut-Back* Approximate Adder with Fixed-point Implementation and Floating-point Precision

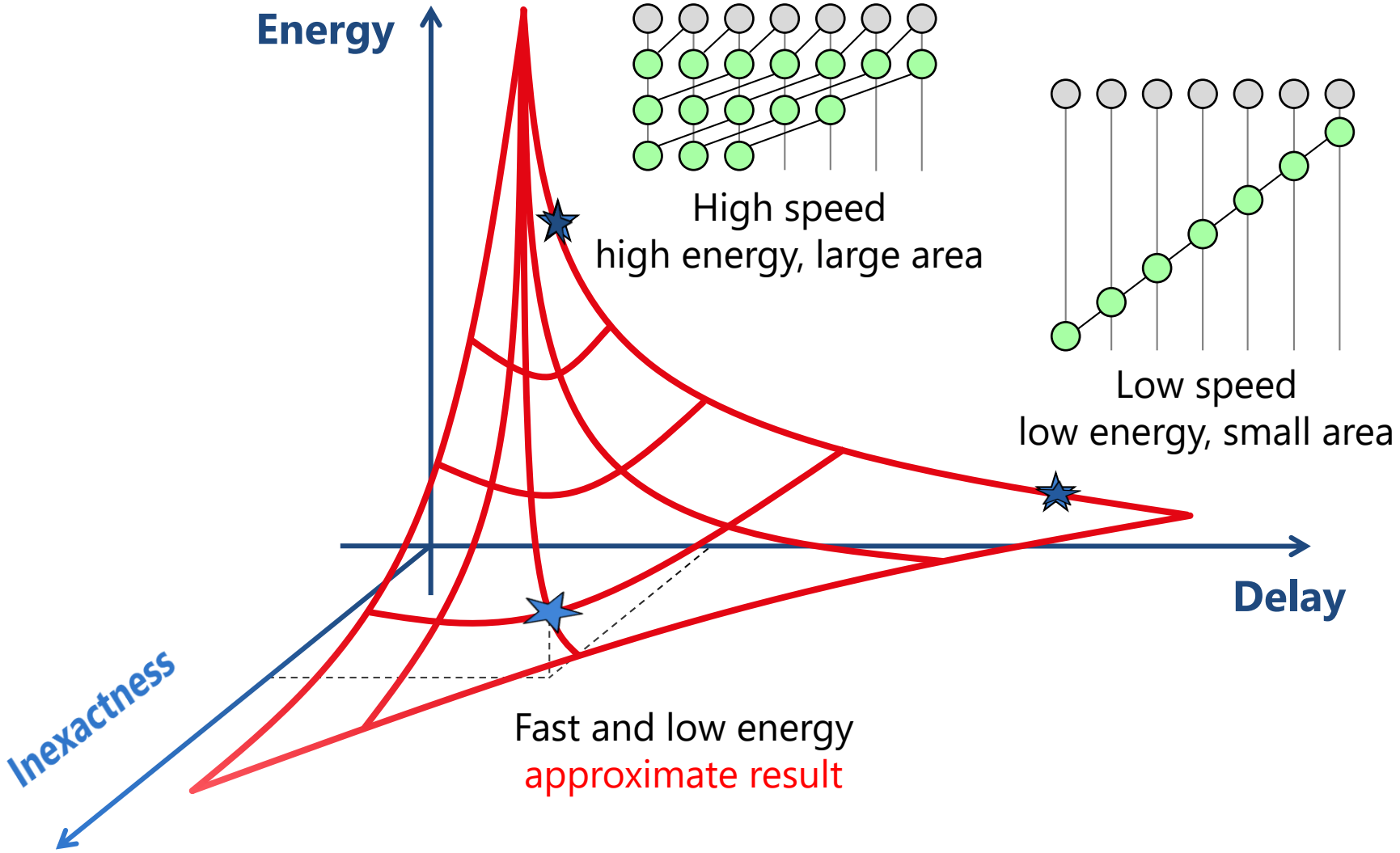
Vincent Camus
Jeremy Schlachter
Christian Enz

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Integrated Circuits Laboratory (ICLAB)
Switzerland

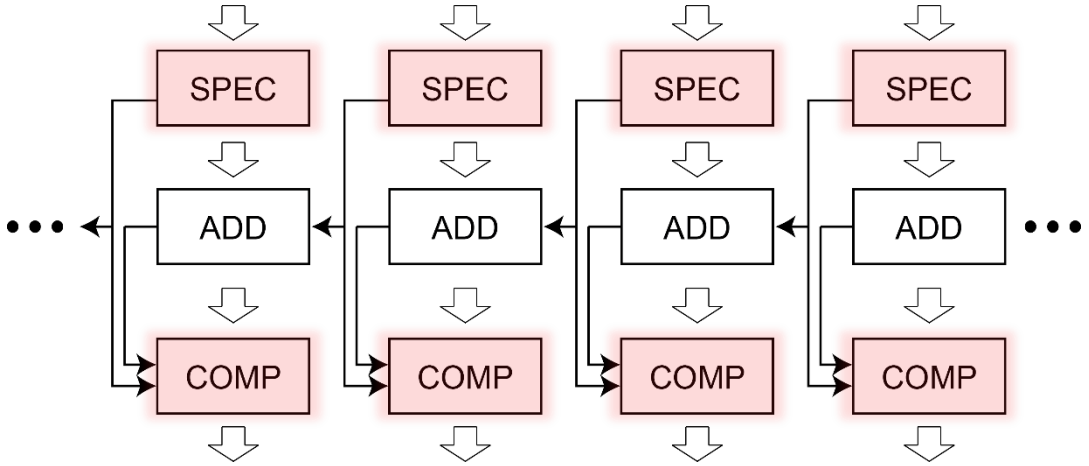
Outline

1. State-of-the-Art
2. CCB Adder Circuit
3. CCB Adder Arithmetic
4. Results and Comparison

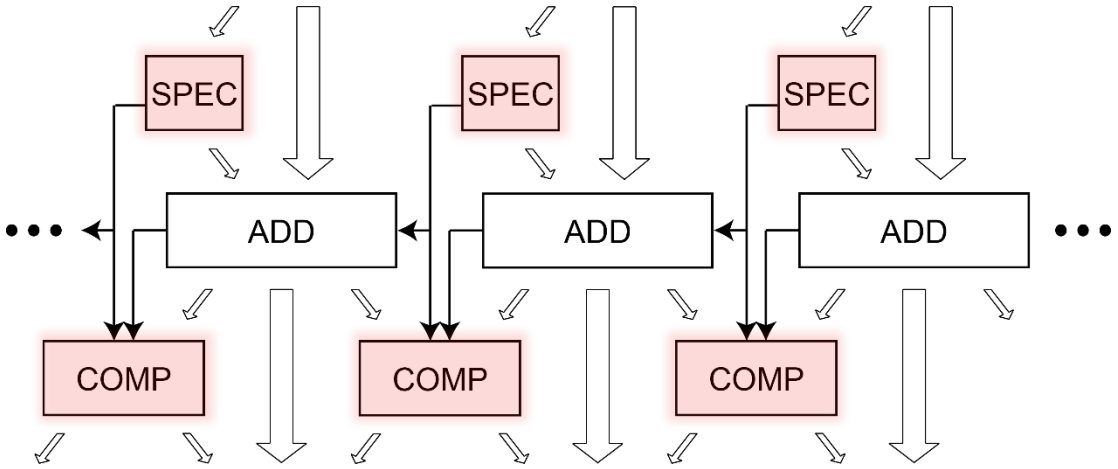
Approximate Circuits, a New Dimension



1. State-of-the-Art – Speculative Adders



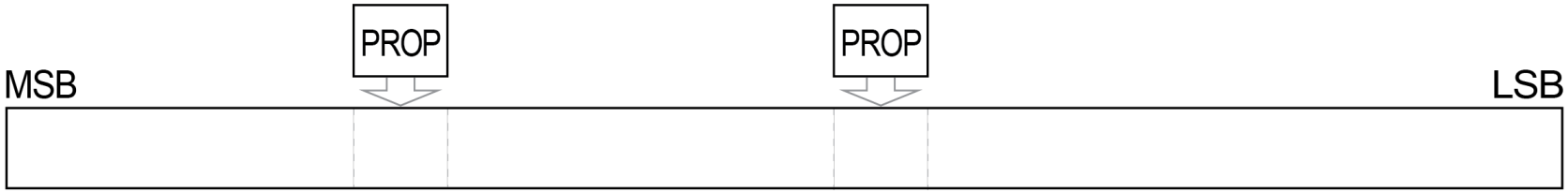
ETBA, *Weber 2013*



ISA, *Camus 2015*

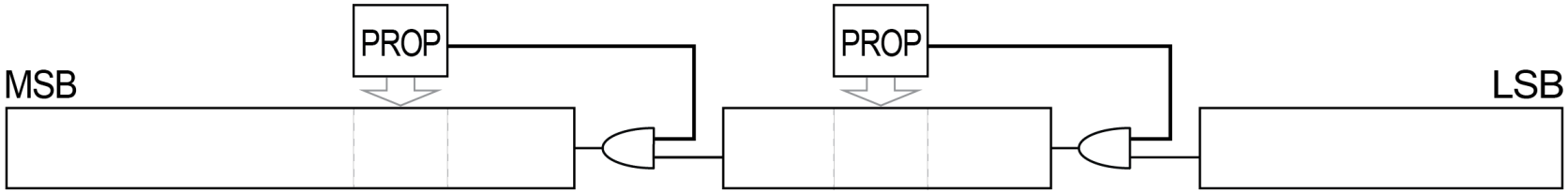
- **Principle**
  - Sliced structure
  - Speculated carry
  - Error compensation
- **Advantages**
  - High speed
  - Worst-case error control
- **Drawbacks**
  - Hardware overhead
  - Delay overhead

2. CCB Adder – Circuit



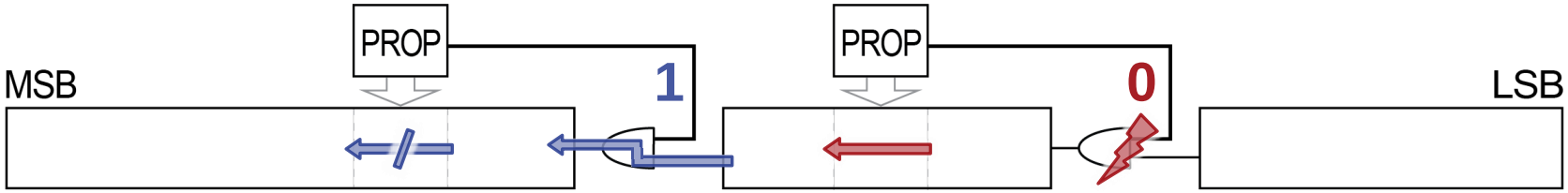
- **Principle**
  - Monitoring high-significance carry stages

2. CCB Adder – Circuit



- **Principle**
  - Monitoring high-significance carry stages
  - Cutting the carry chain at low-significance positions

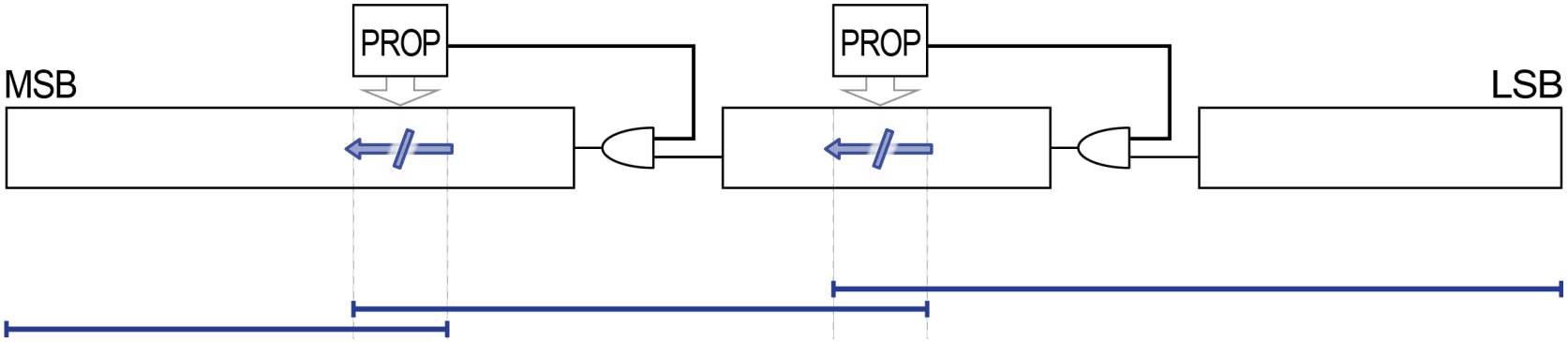
2. CCB Adder – Circuit



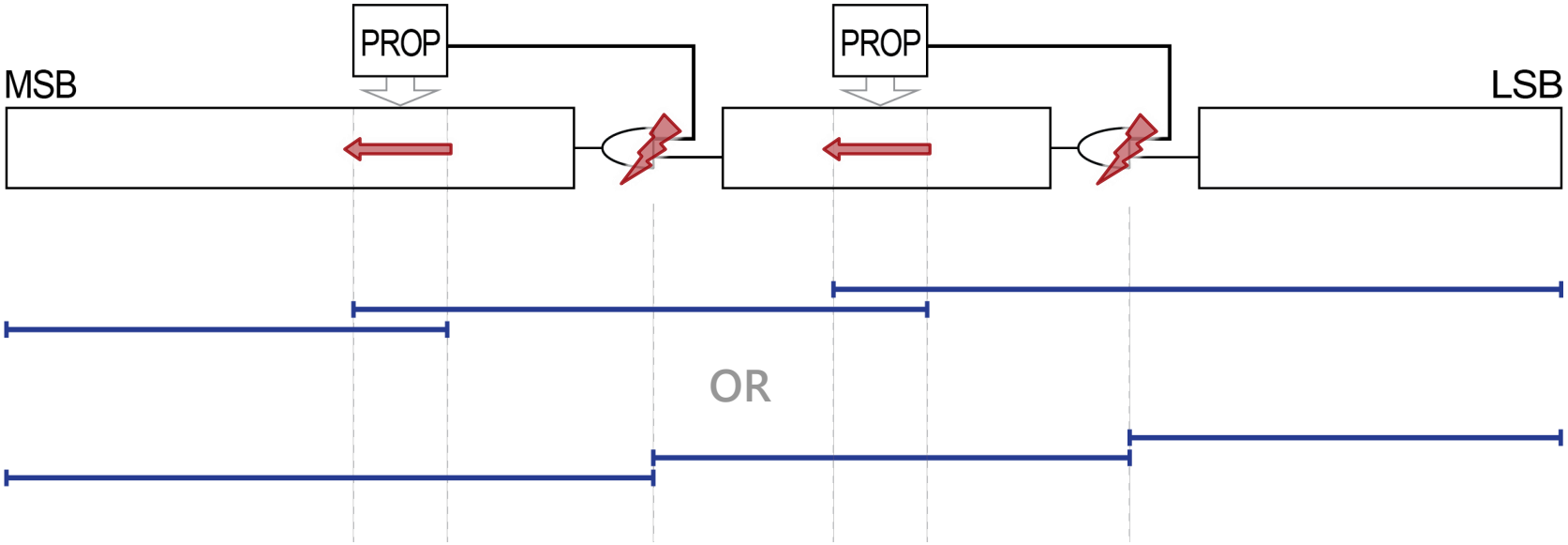
- **Principle**
  - Monitoring high-significance carry stages
  - Cutting the carry chain at low-significance positions

- **Carry propagation**
  - Naturally doesn't propagate
  - Artificially cut back

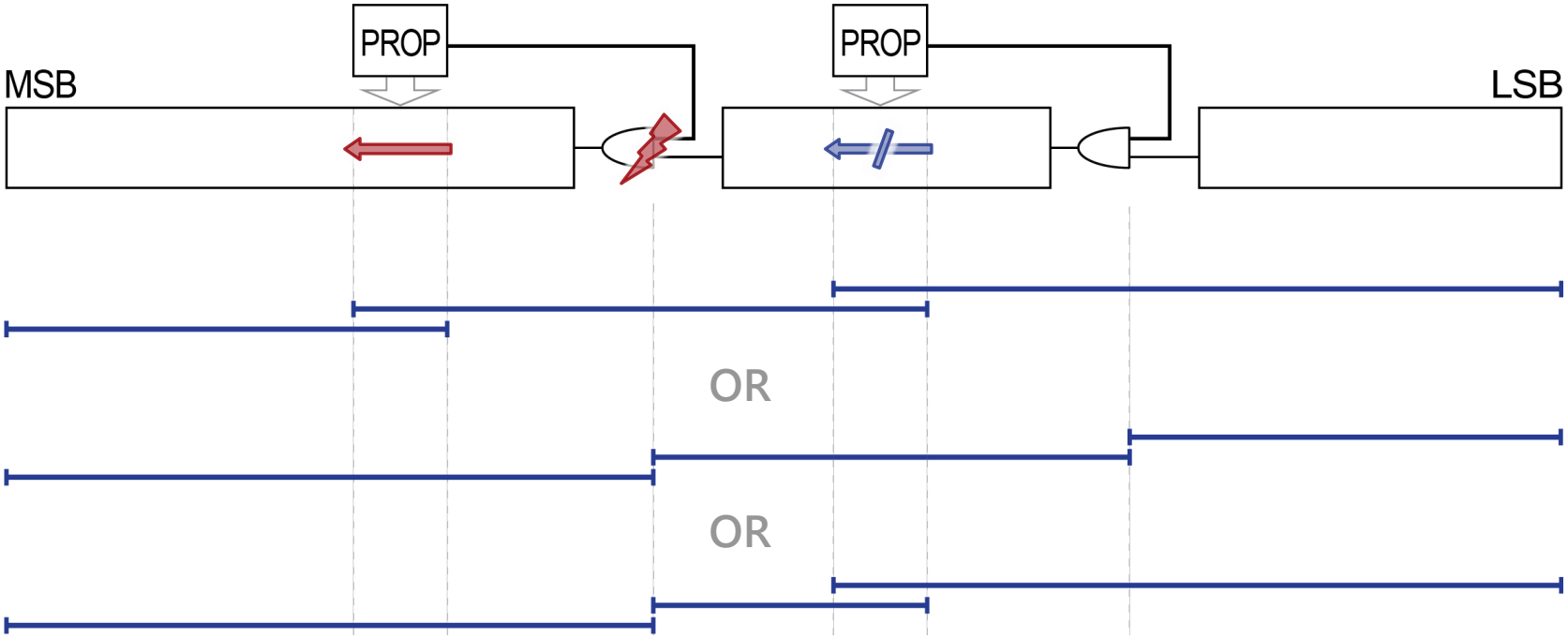
2. CCB Adder – Circuit



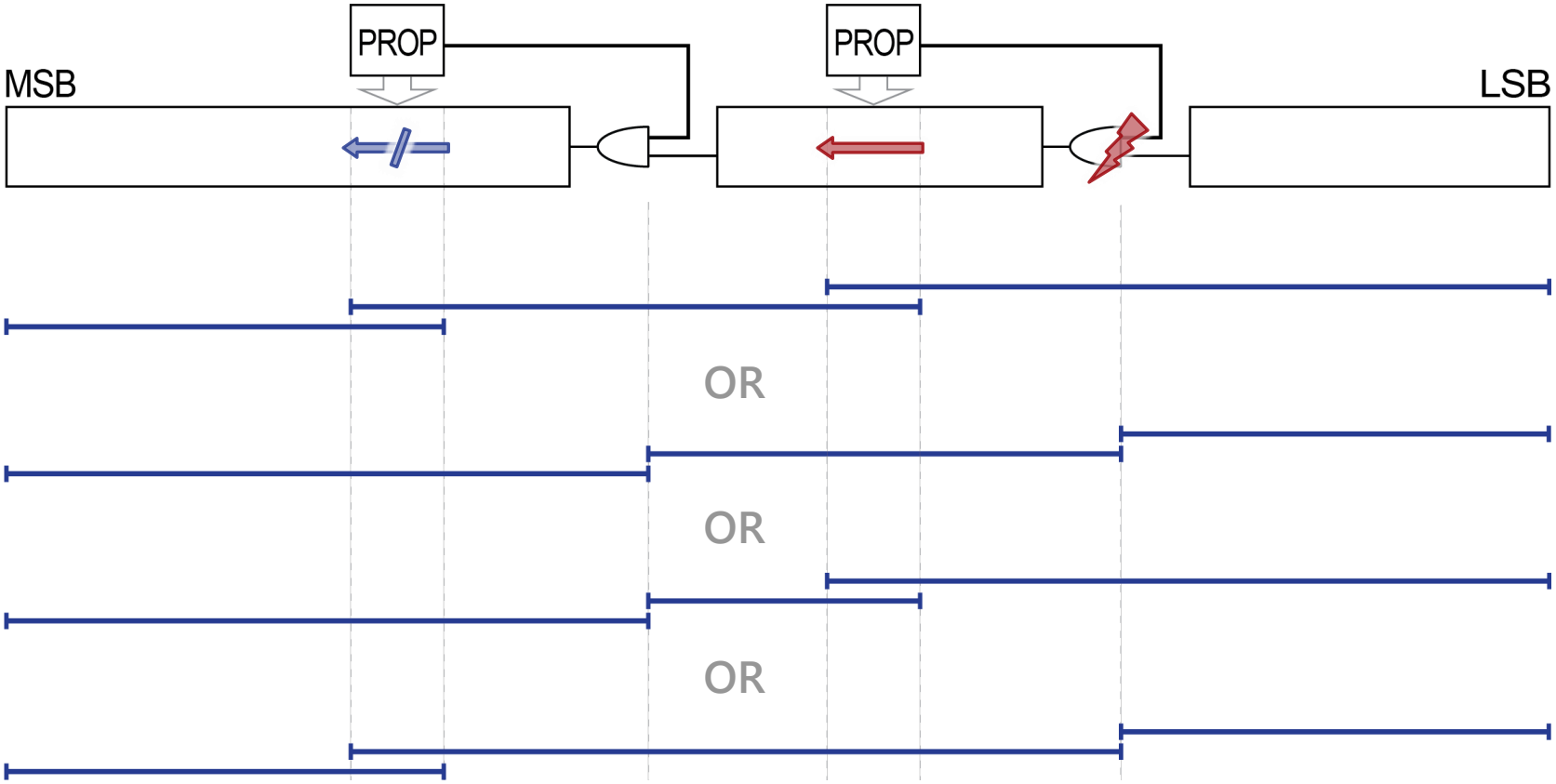
2. CCB Adder – Circuit



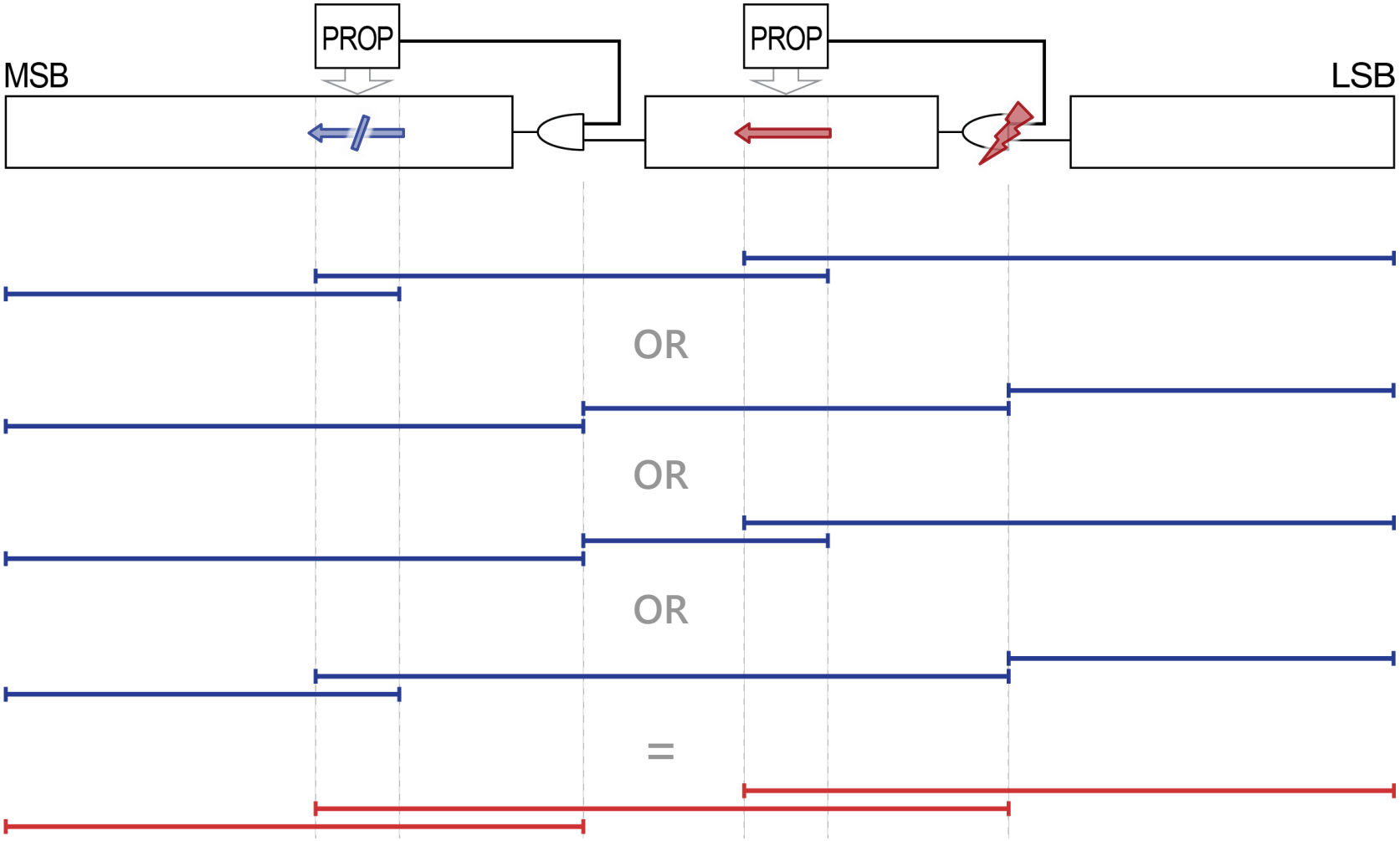
2. CCB Adder – Circuit



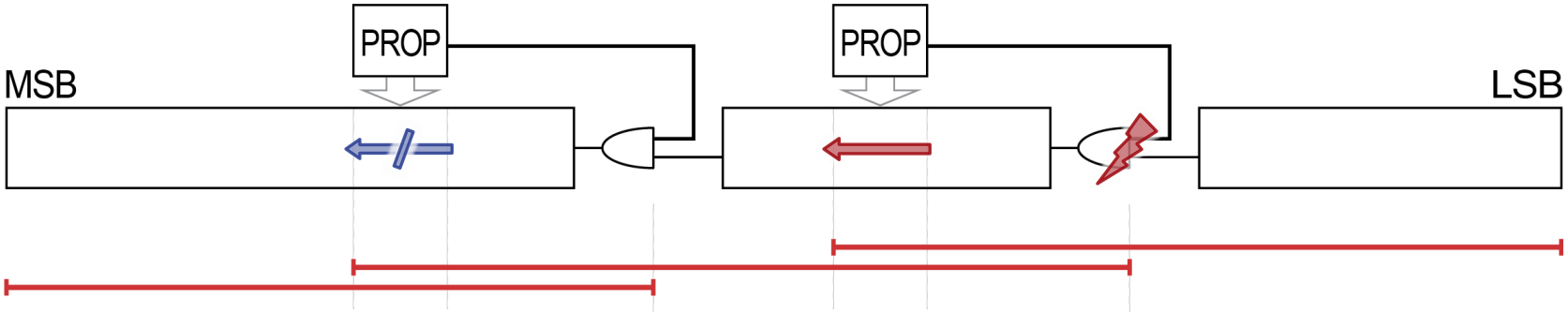
2. CCB Adder – Circuit



2. CCB Adder – Circuit



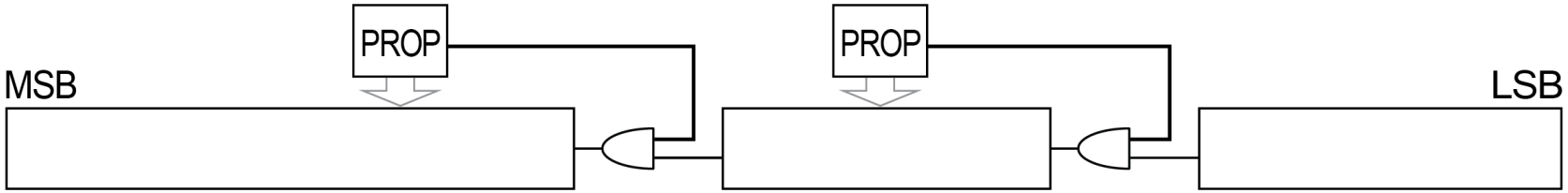
2. CCB Adder – Circuit



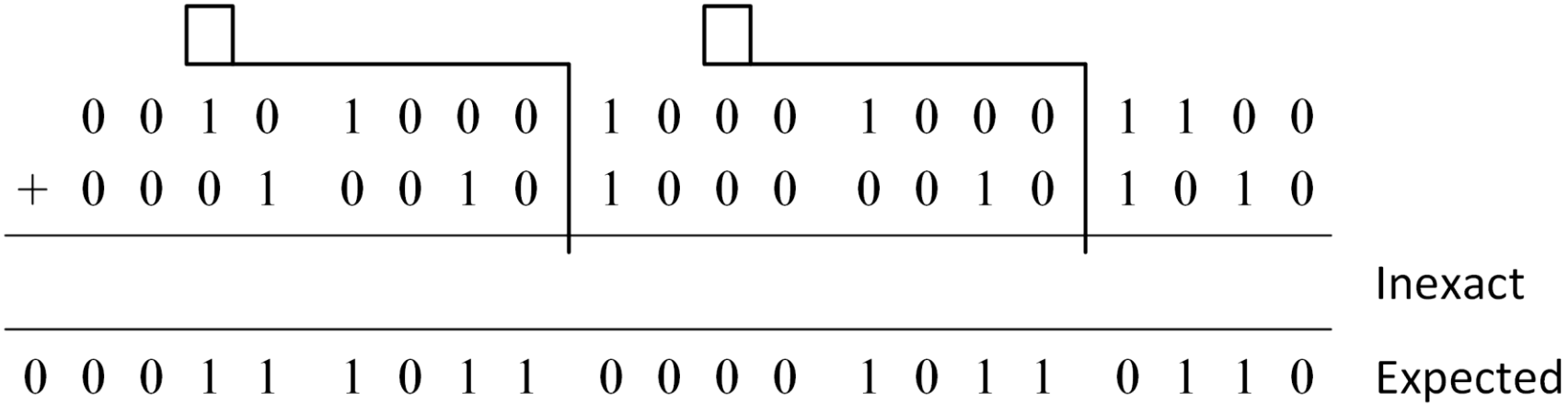
The <u>full</u> carry chain exists… but is never stimulated !

- **Carry chain transformed into false-path**
  - Strong timing relaxation
  - Improved performance and efficiency
- **False-path engineering**
  - Non-recognized by regular timing analysis
  - Requires timing exception script

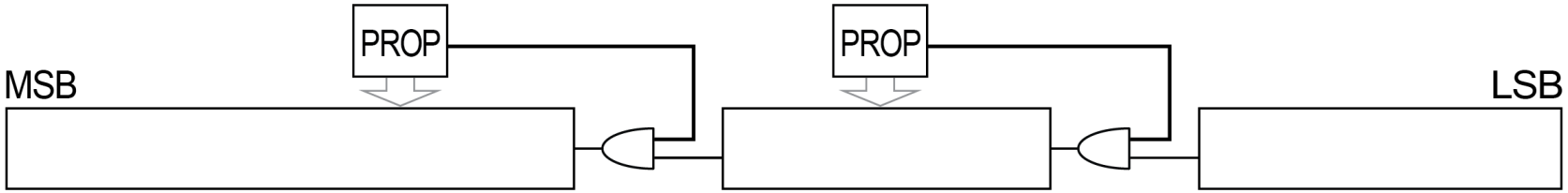
3. CCB Adder – Arithmetic



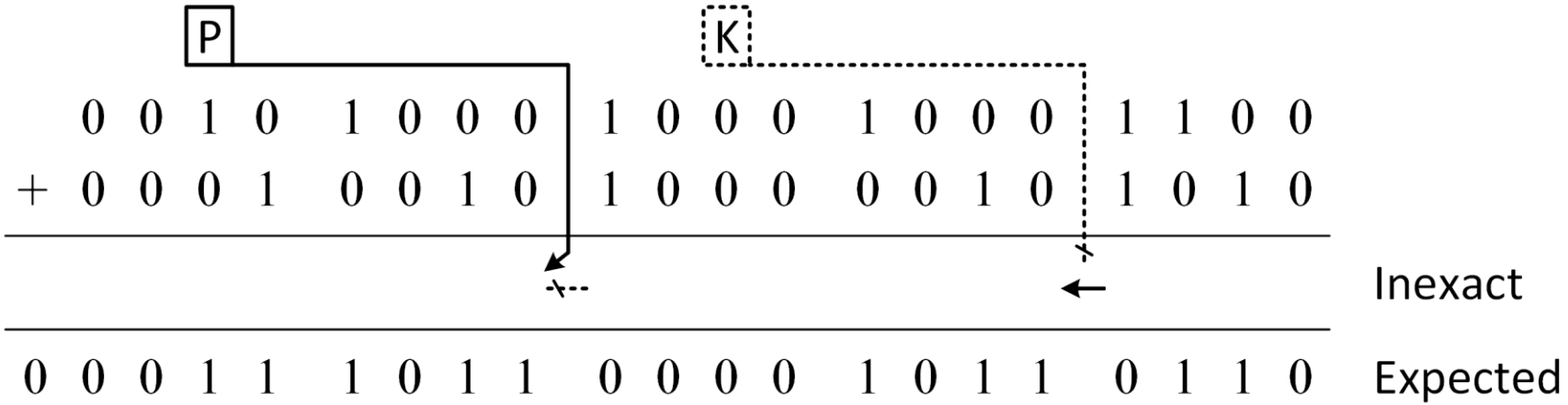
- **Arithmetic principle**



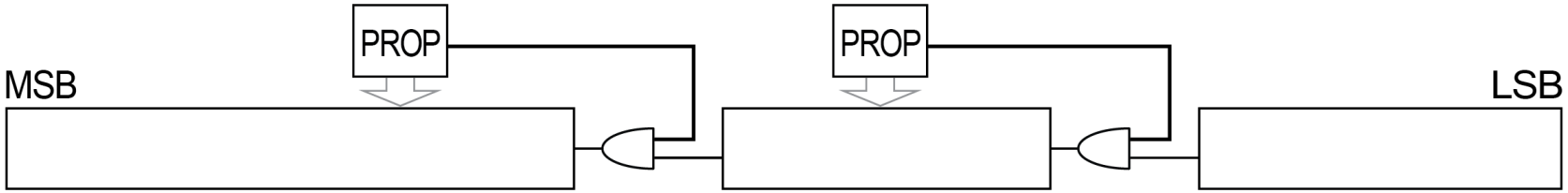
3. CCB Adder – Arithmetic



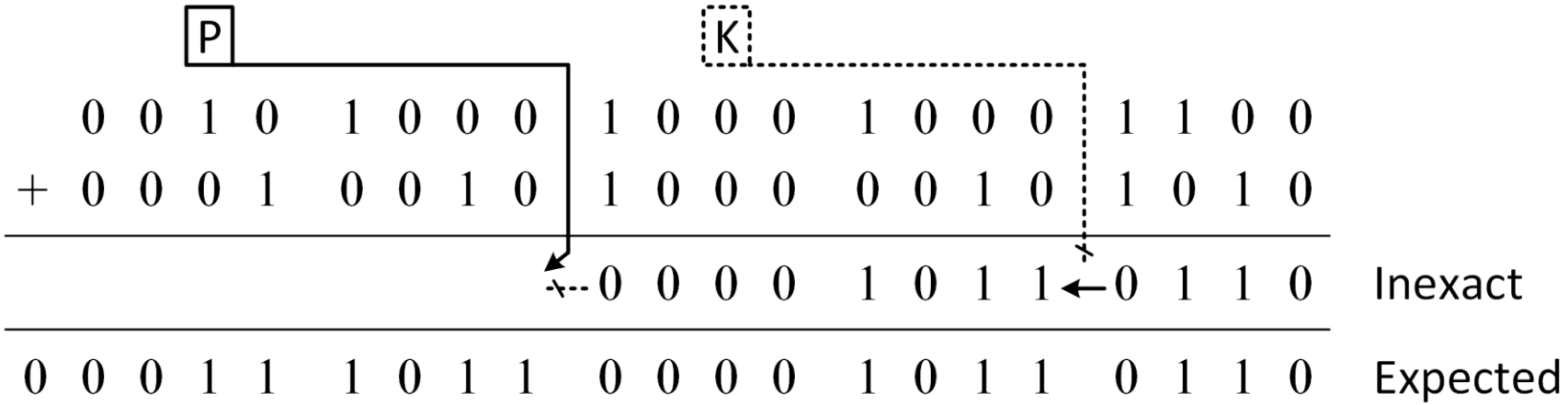
- **Arithmetic principle**



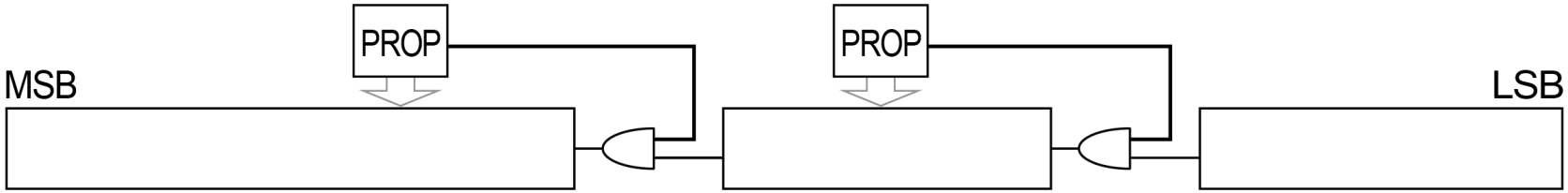
3. CCB Adder – Arithmetic



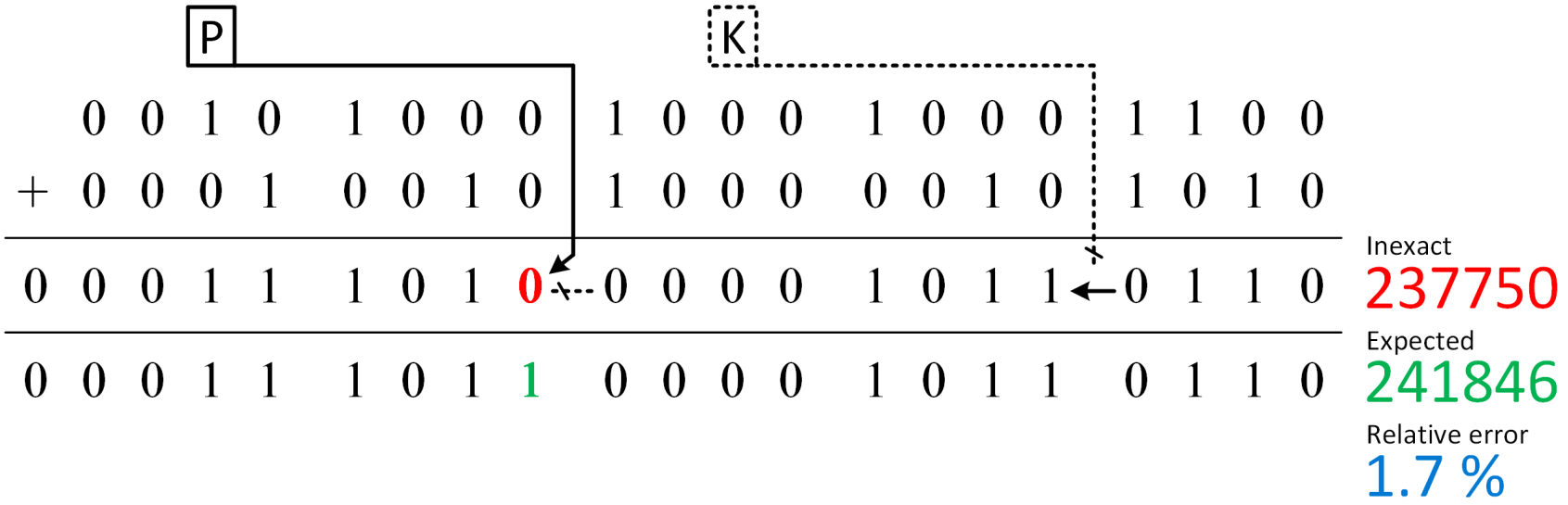
- **Arithmetic principle**



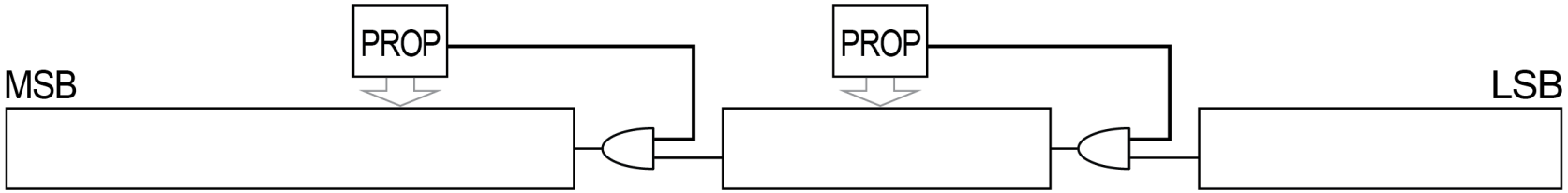
3. CCB Adder – Arithmetic



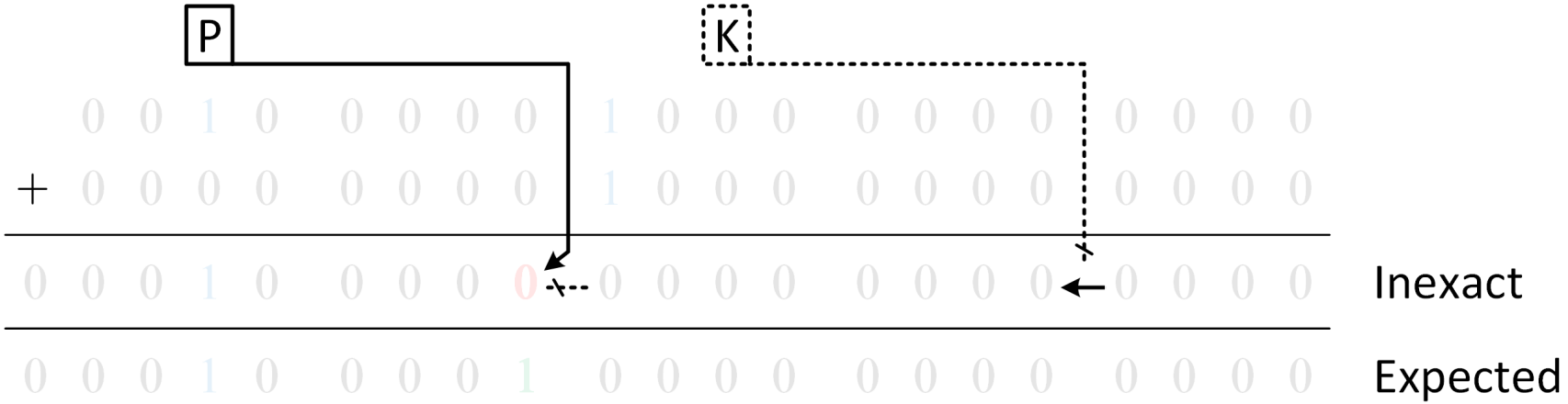
- **Arithmetic principle**



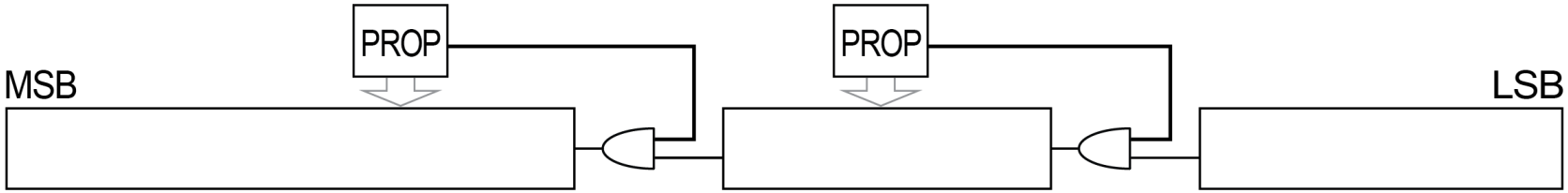
3. CCB Adder – Arithmetic



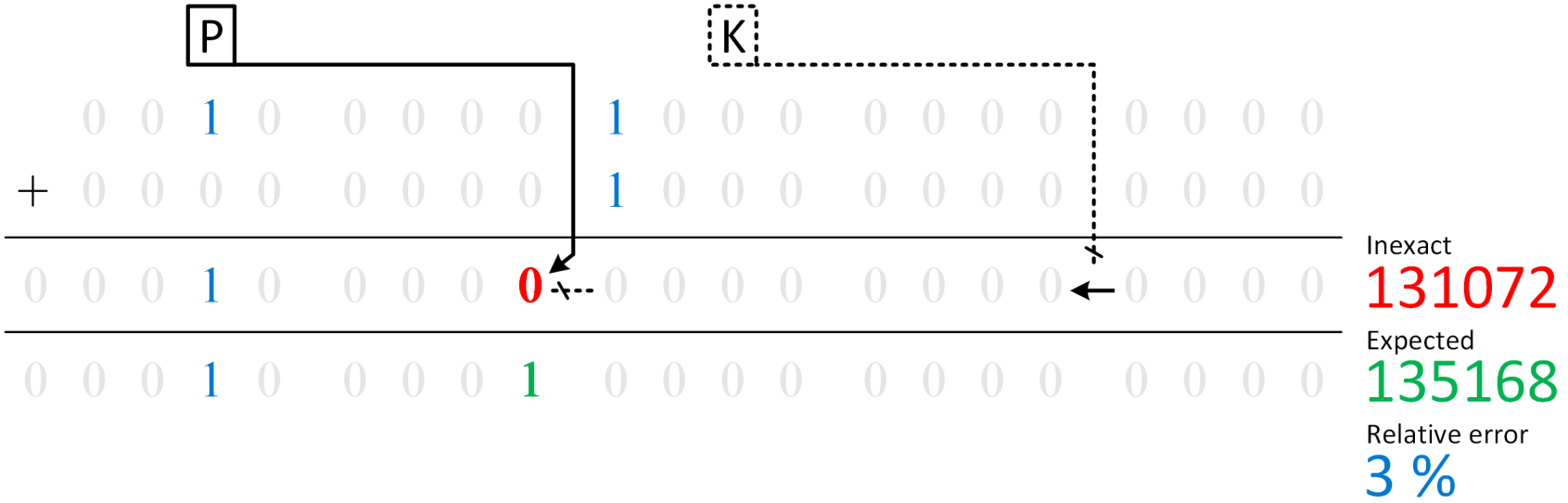
- **Worst-case error**



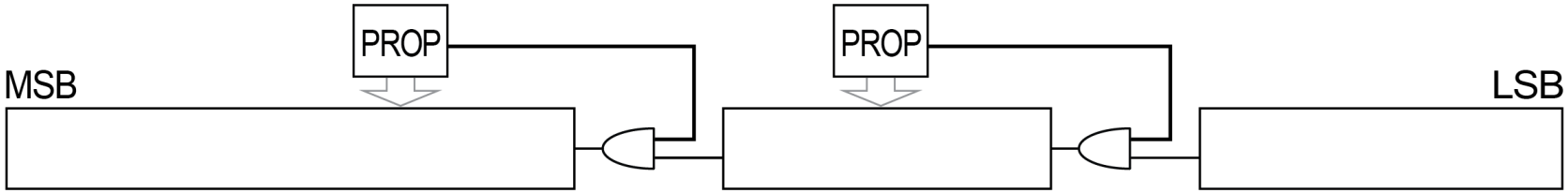
3. CCB Adder – Arithmetic



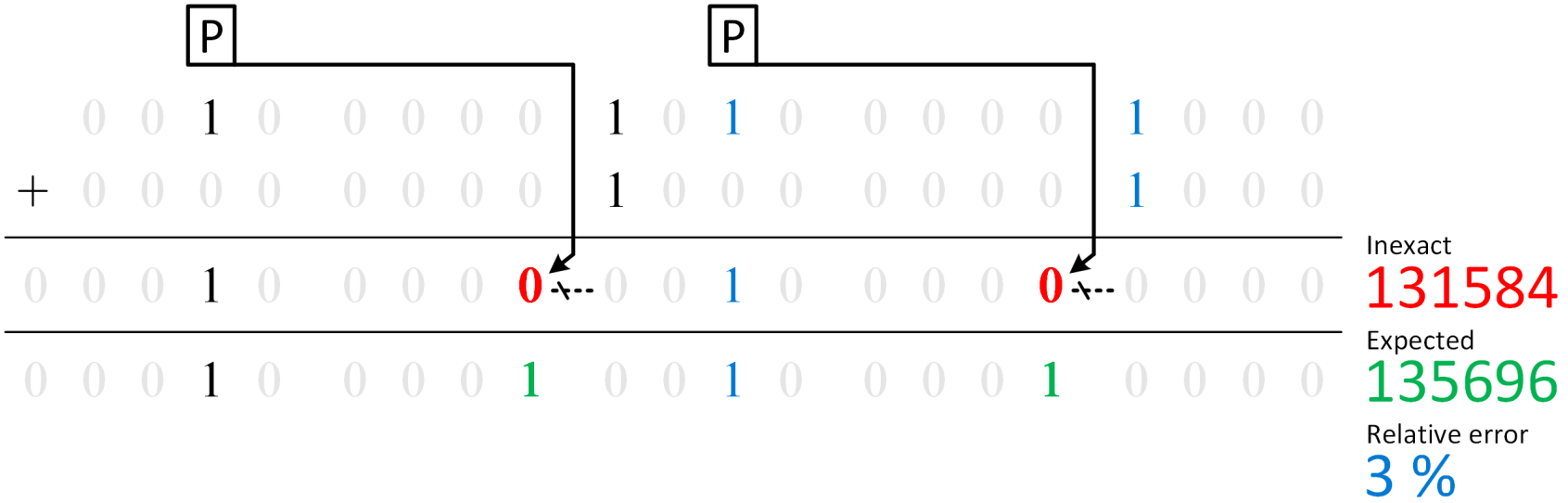
- **Worst-case error (very low thanks to the feed-back)**



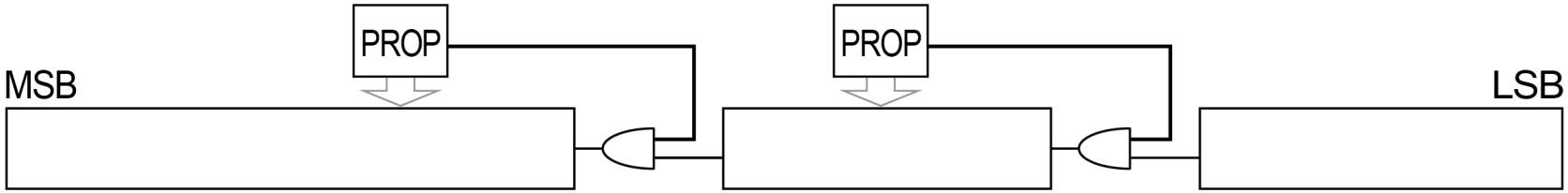
3. CCB Adder – Arithmetic



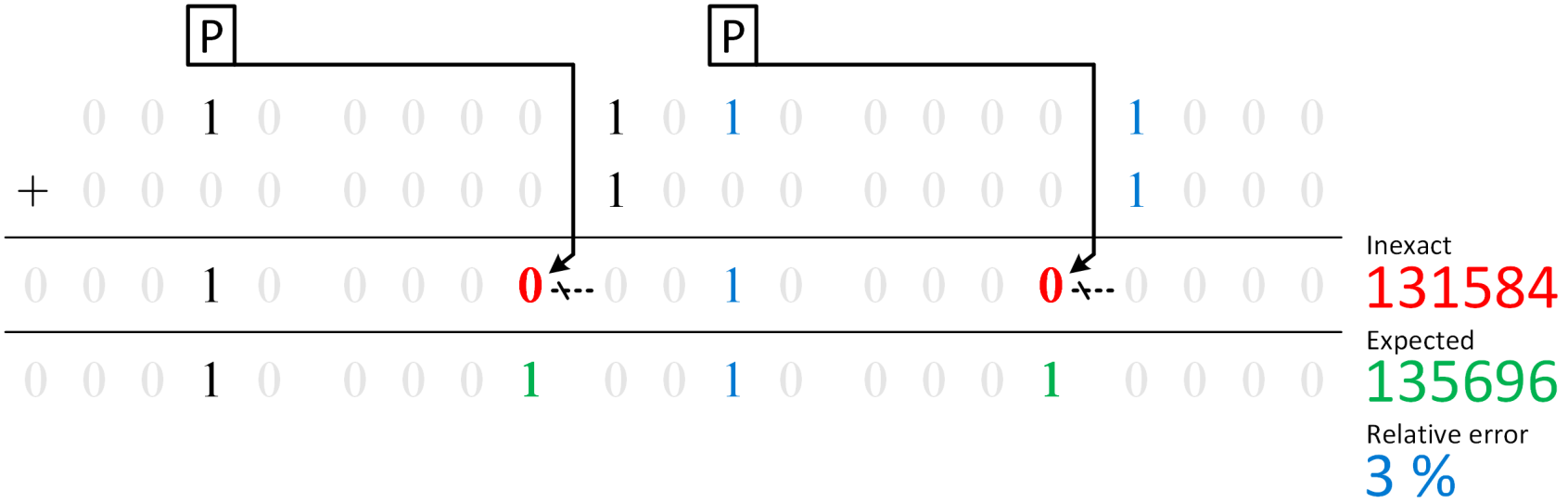
- **Multiple errors (do not increase worst-case error)**



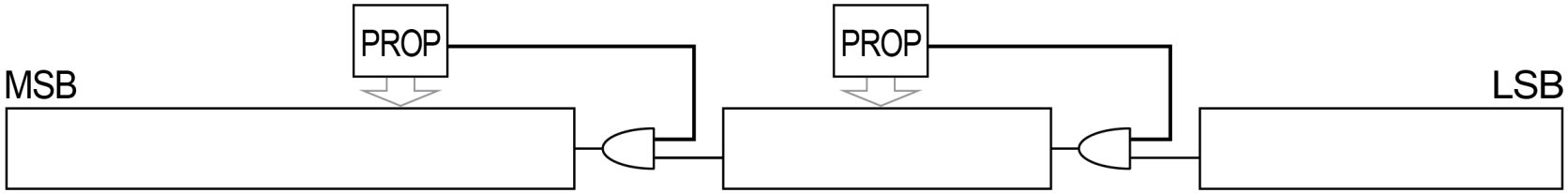
3. CCB Adder – Arithmetic



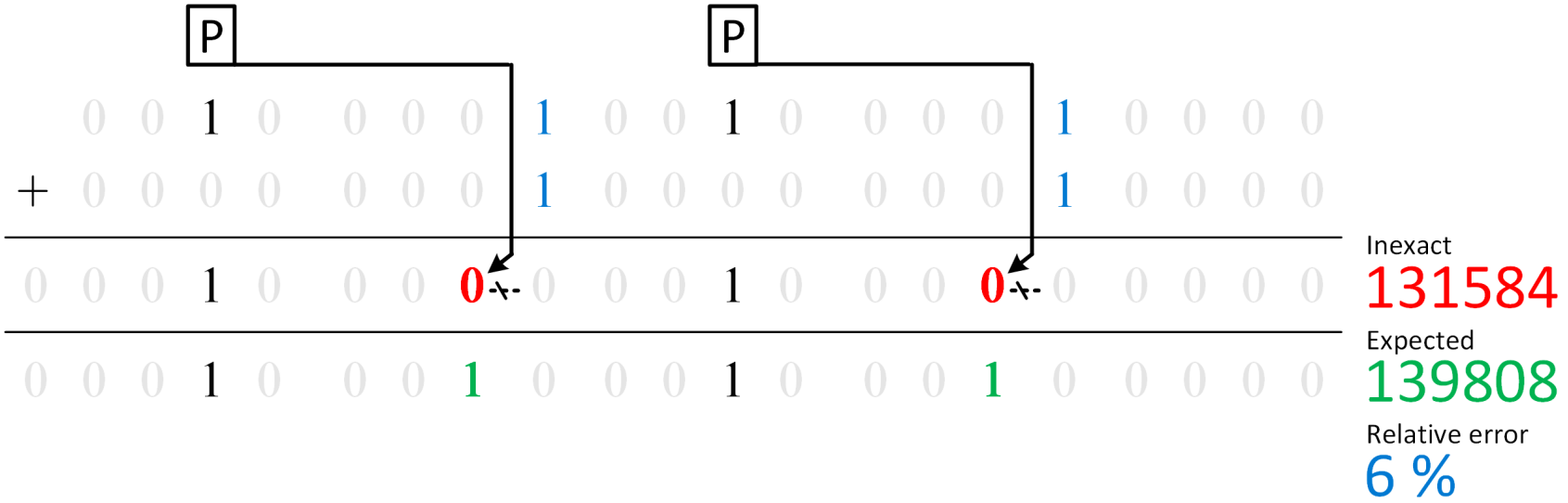
- **Error control: *cut-back* length: 5 bits**



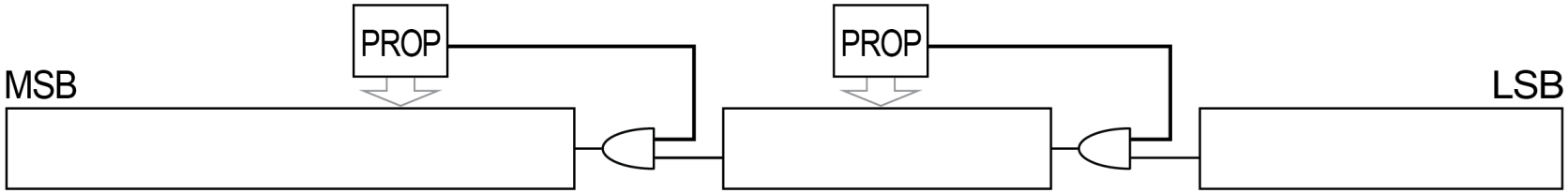
3. CCB Adder – Arithmetic



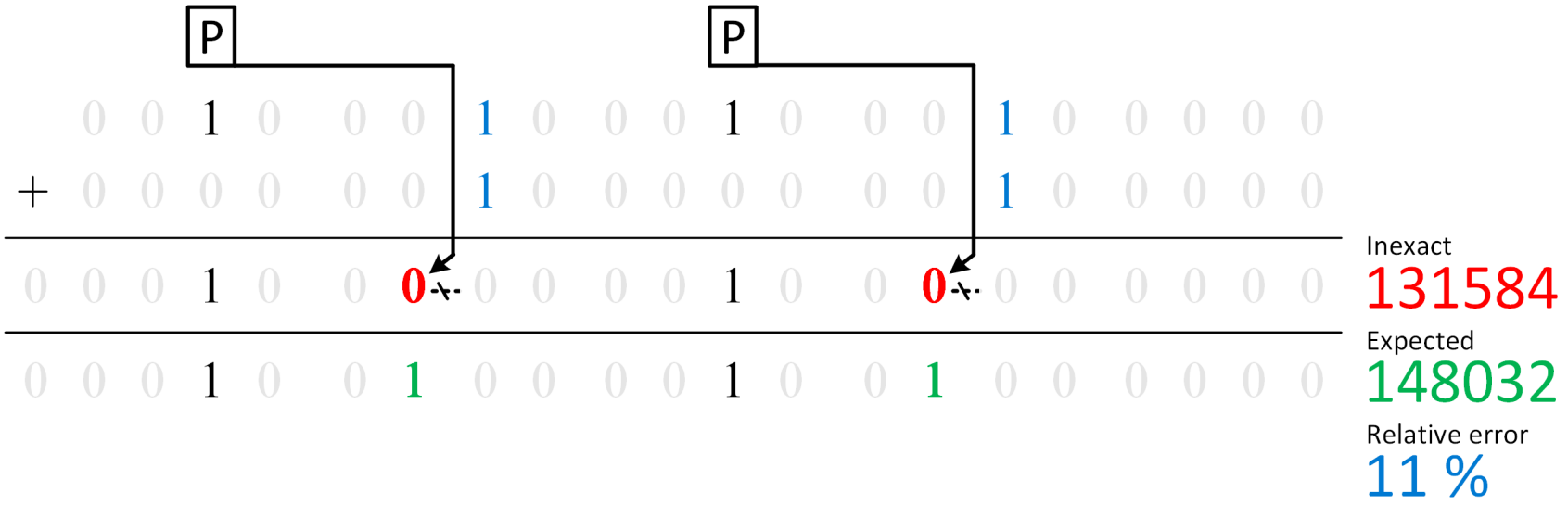
- **Error control: *cut-back* length: 4 bits**



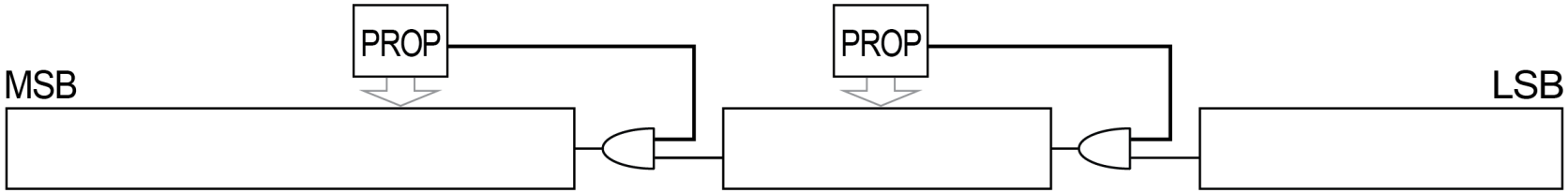
3. CCB Adder – Arithmetic



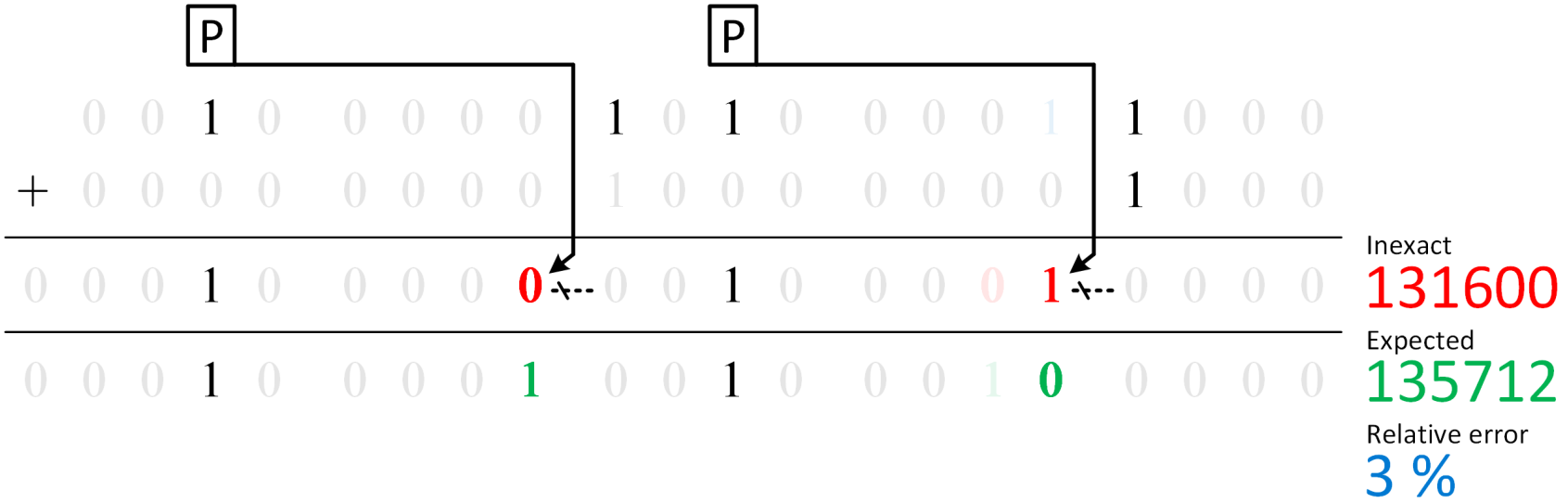
- **Error control: *cut-back* length: 3 bits**



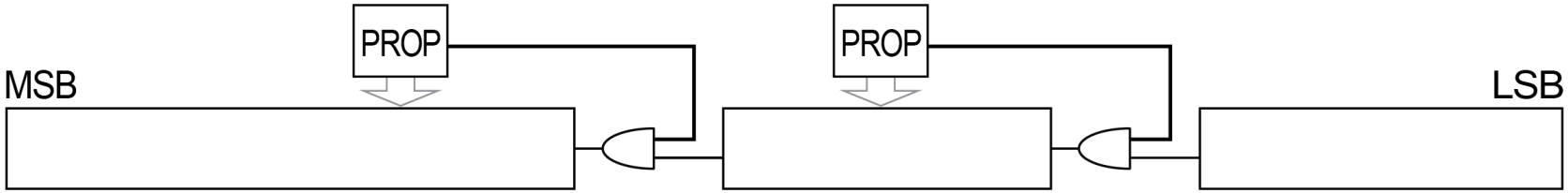
Extra slide – Arithmetic



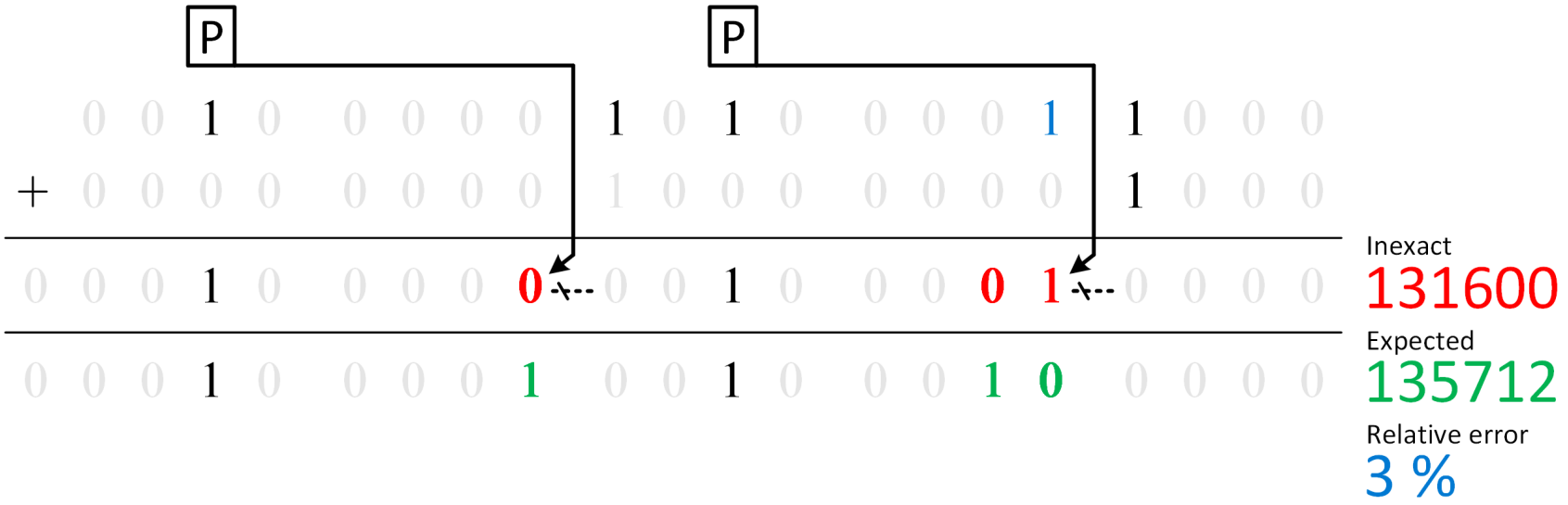
- **Propagating error: 1 bit**



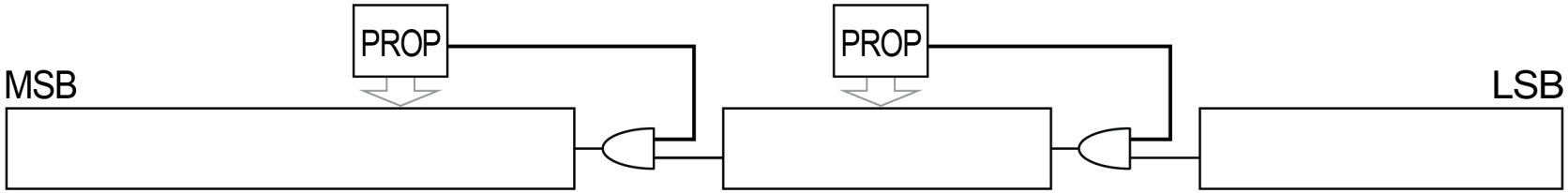
Extra slide – Arithmetic



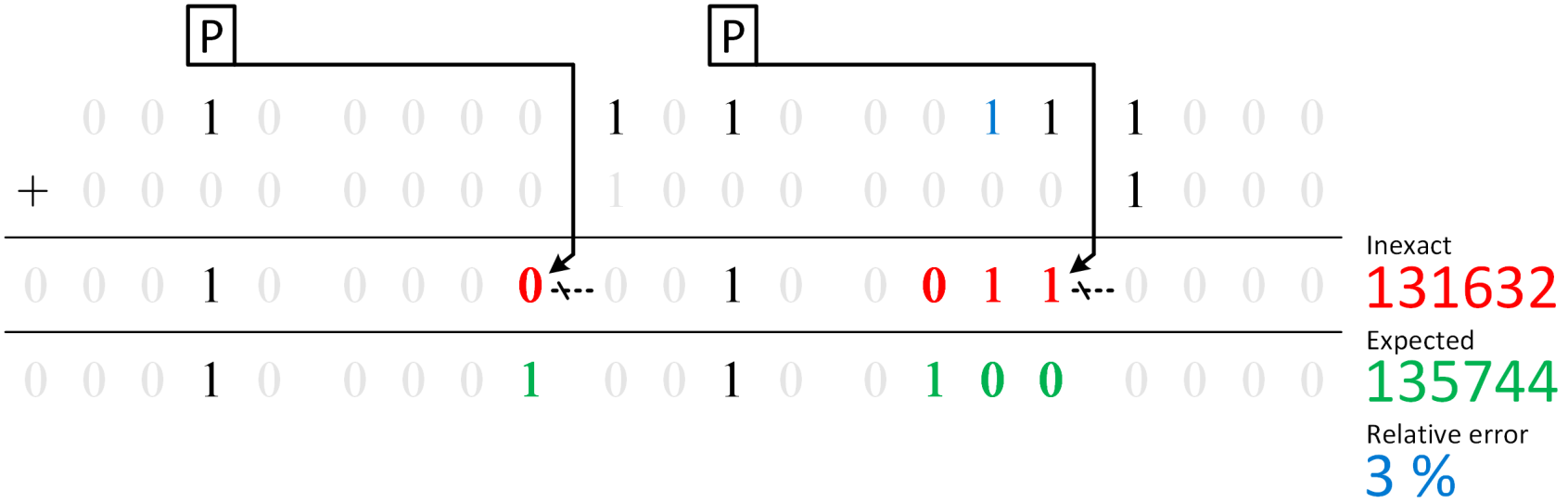
- **Propagating error: 2 bits**



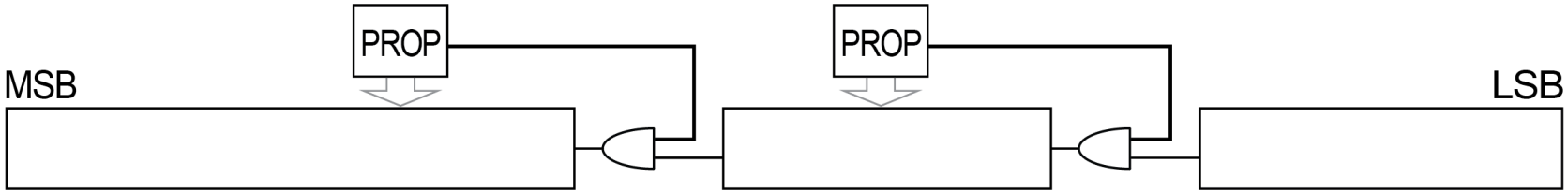
Extra slide – Arithmetic



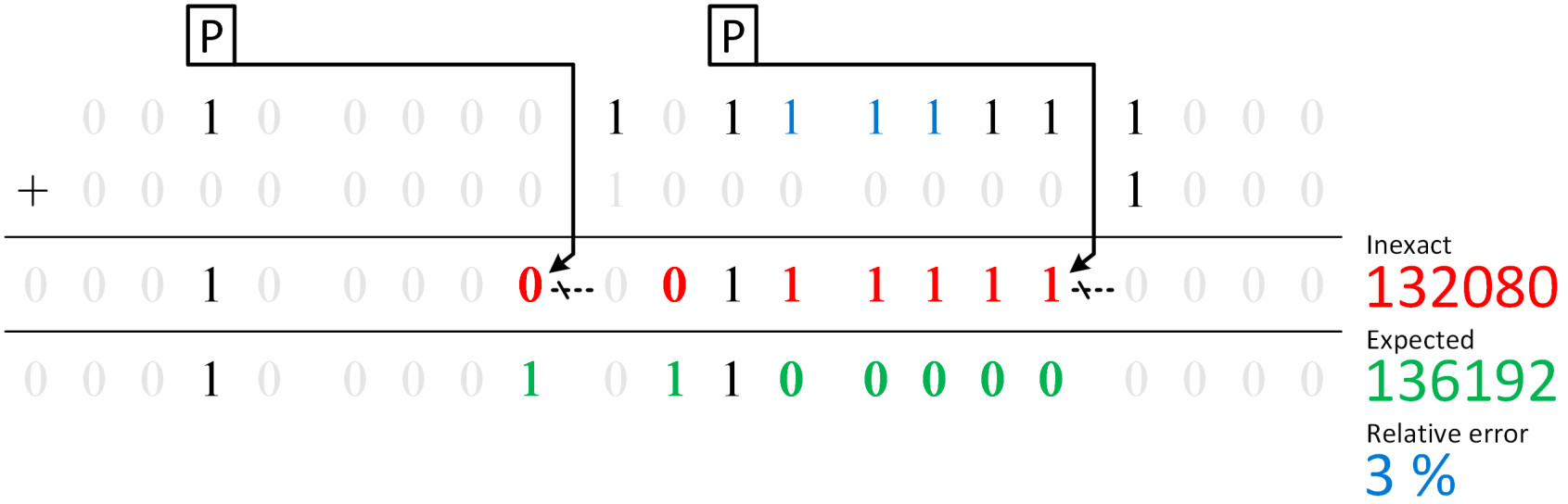
- **Propagating error: 3 bits**



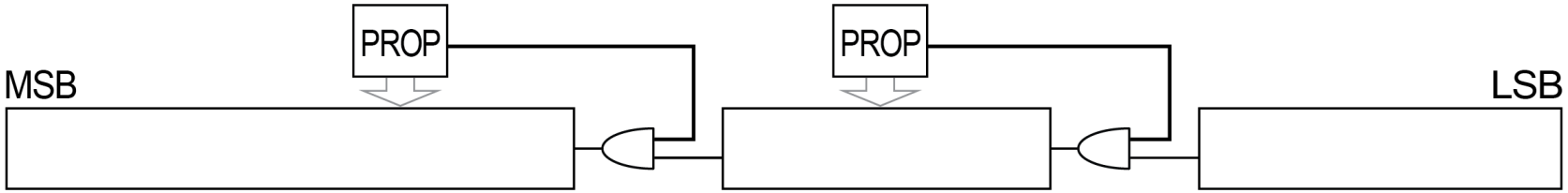
Extra slide – Arithmetic



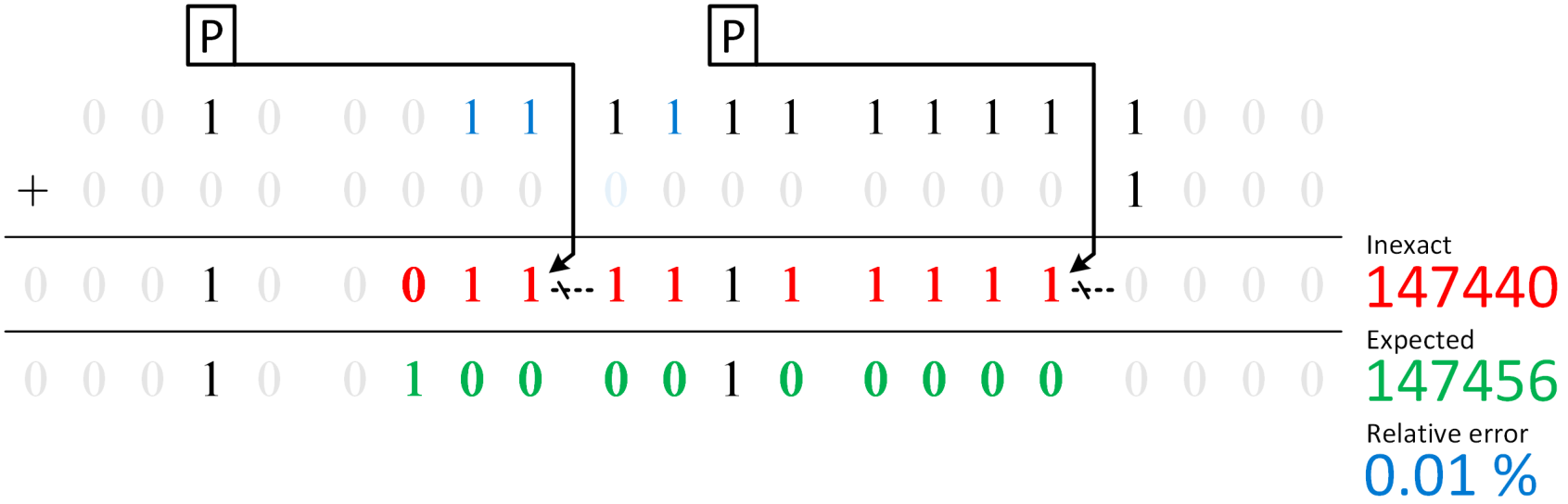
- **Propagating error: many bits, without increasing errors**



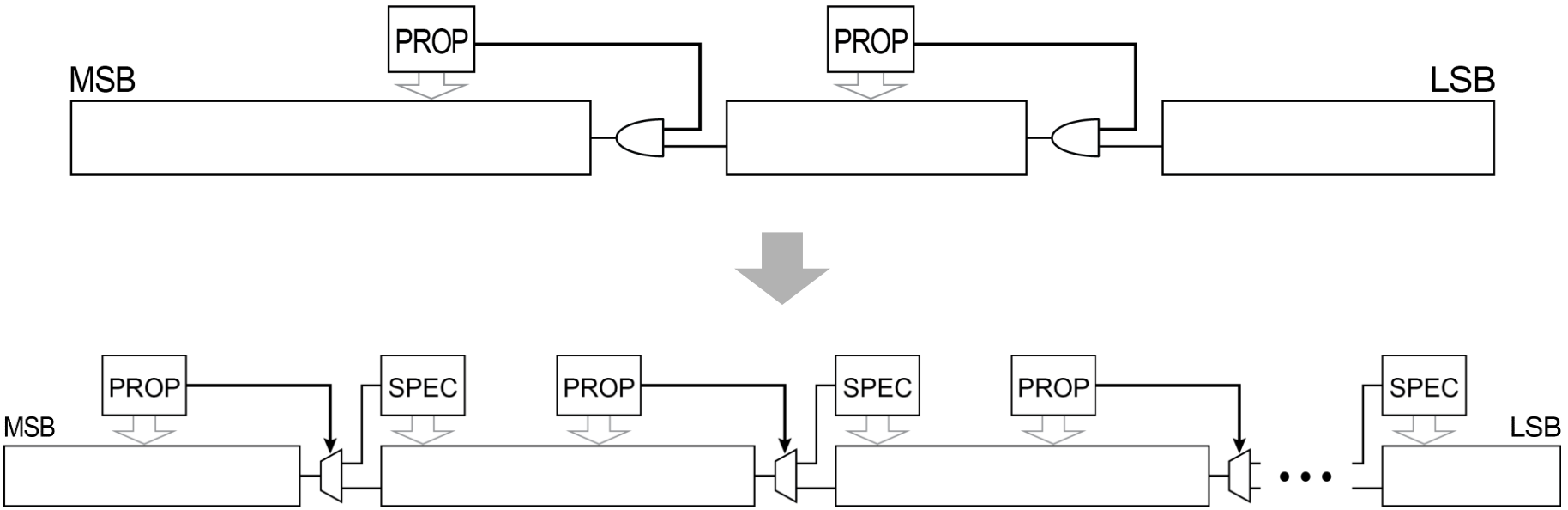
Extra slide – Arithmetic



- **Propagating error: many bits, without increasing errors**



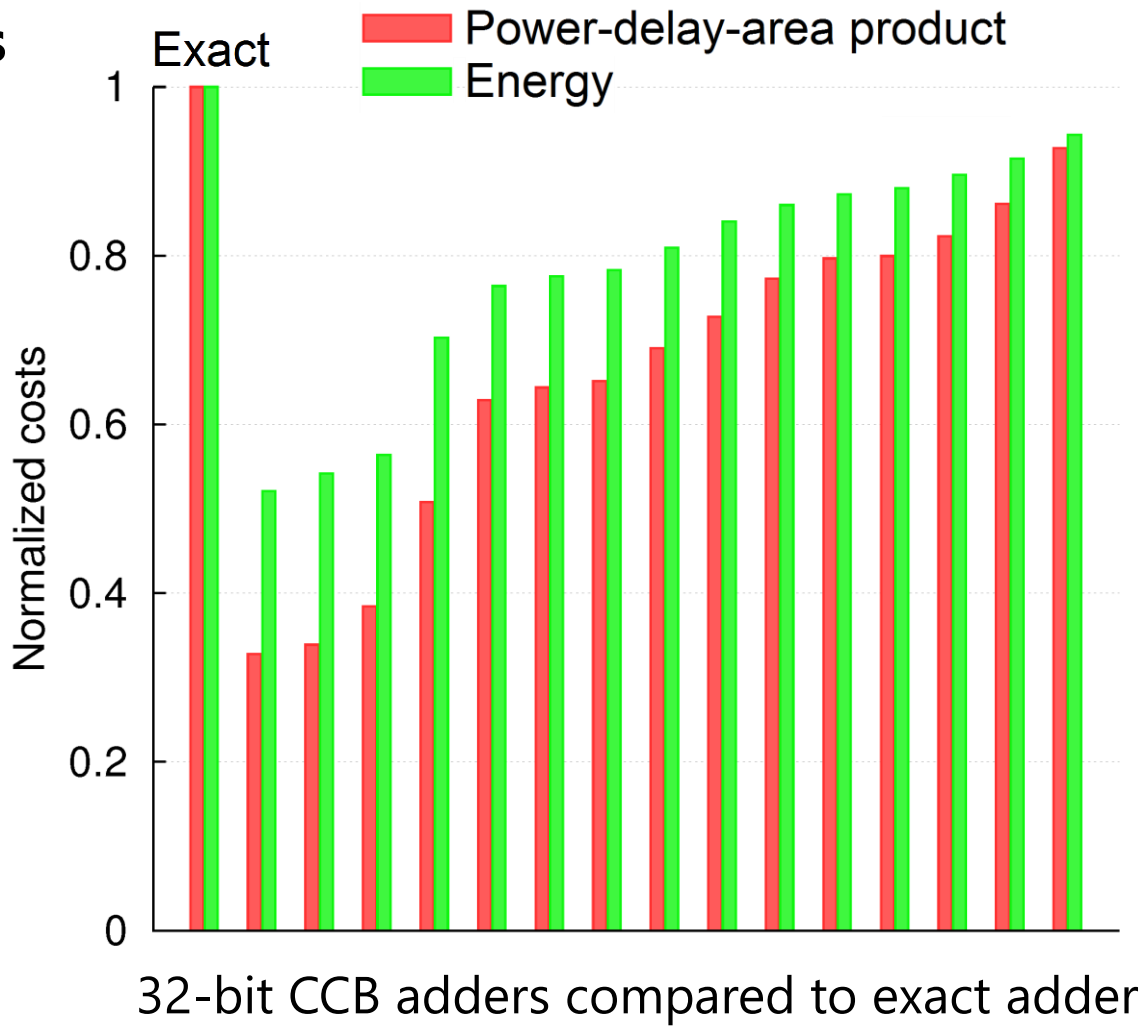
4. CCB Adder – Implementation



- **General architecture**
  - Better error control
  - Delay optimization
- **High-level description**
  - Design flexibility
  - Optimized compilation

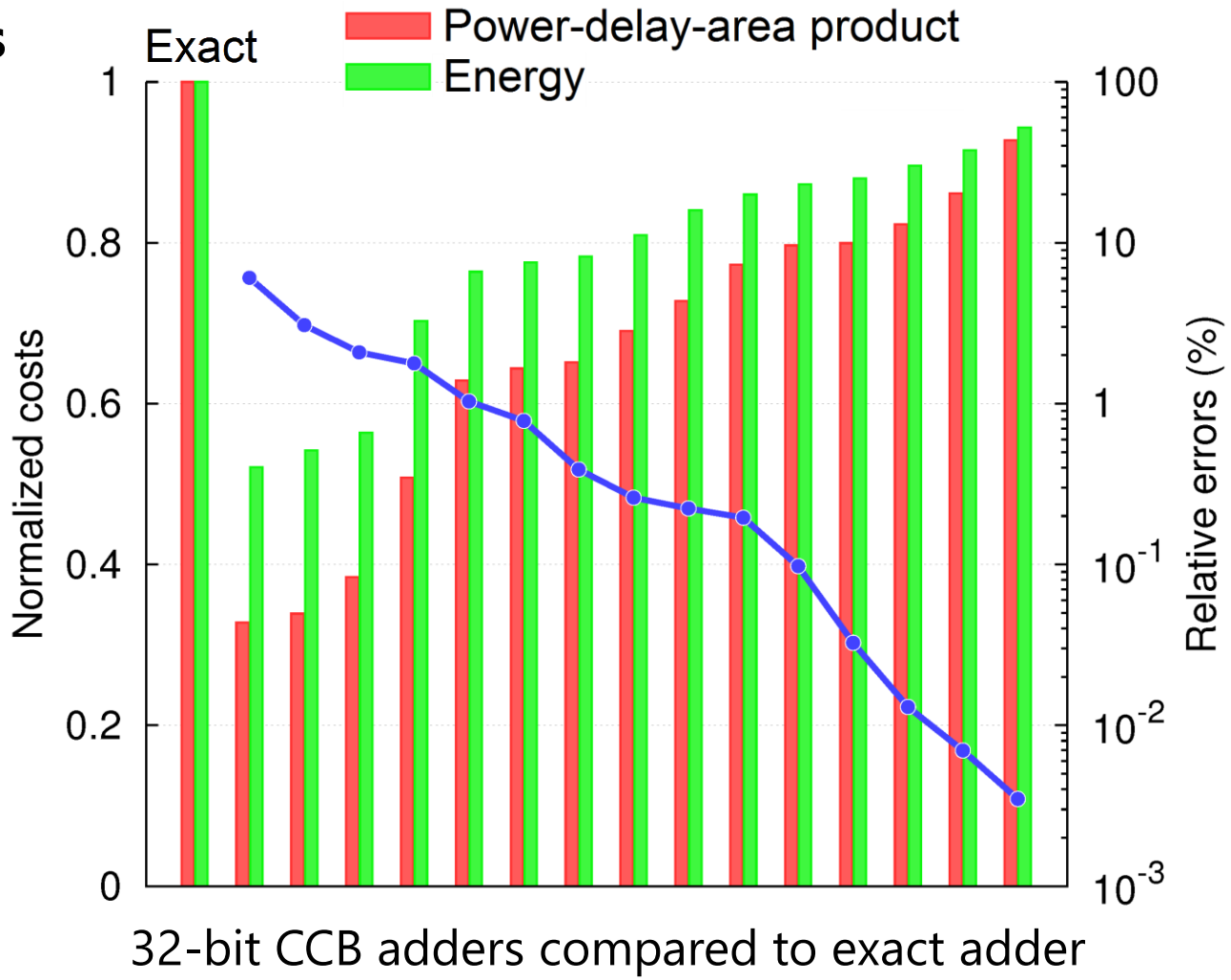
4. CCB Adder – Results

- **Implementations**
  - 32 bits
  - 65 nm techno
  - 800 MHz
- **Metrics**
  - Energy
  - PDAP



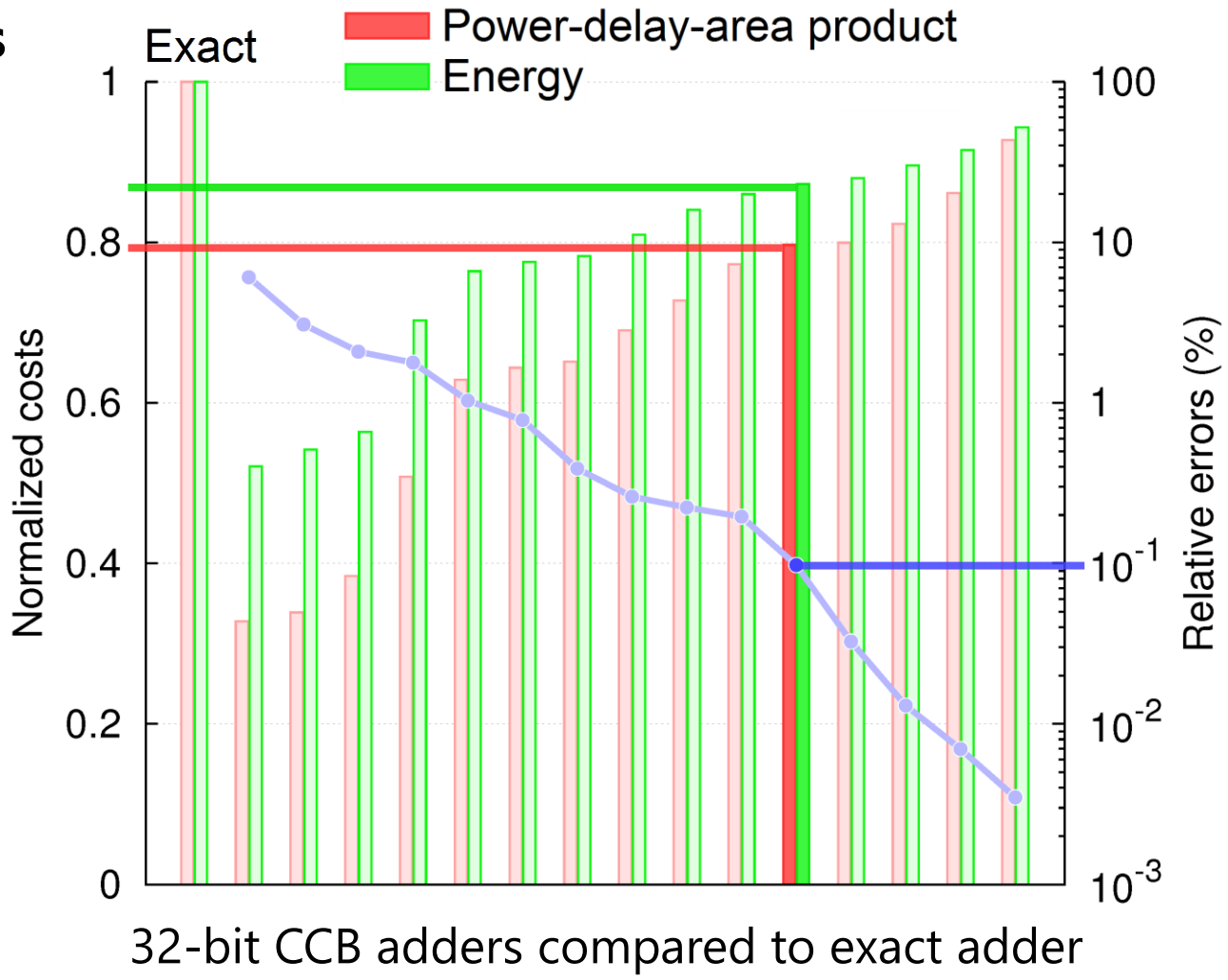
4. CCB Adder – Results

- **Implementations**
  - 32 bits
  - 65 nm techno
  - 800 MHz
- **Metrics**
  - Energy
  - PDAP
  - Max error



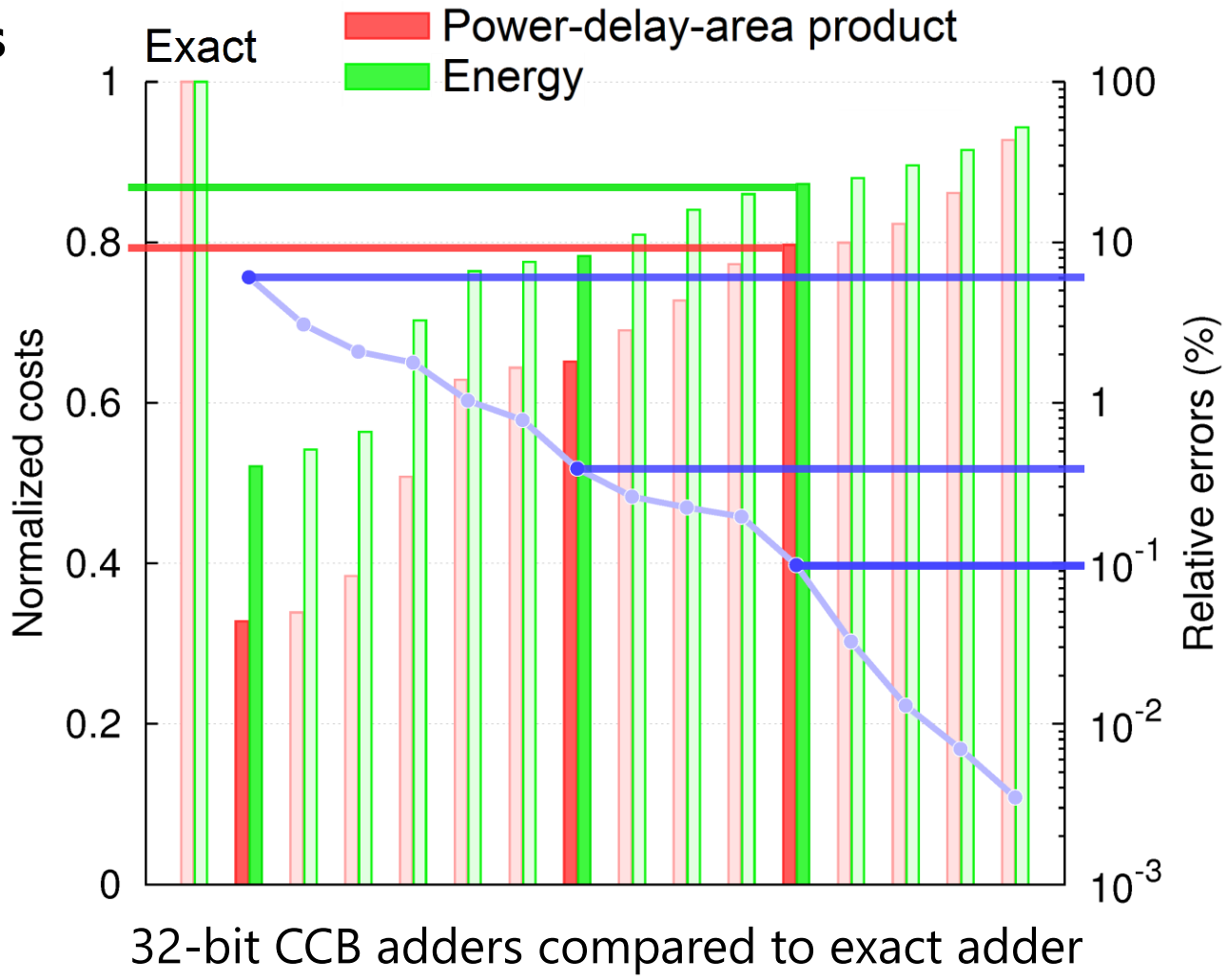
4. CCB Adder – Results

- **Implementations**
  - 32 bits
  - 65 nm techno
  - 800 MHz
- **Metrics**
  - Energy
  - PDAP
  - Max error
- **Savings**
  - -14 % energy
  - -22 % PDAP
  - 0.1 % max error



4. CCB Adder – Results

- **Implementations**
  - 32 bits
  - 65 nm techno
  - 800 MHz
- **Metrics**
  - Energy
  - PDAP
  - Max error
- **Savings**
  - -14 % energy
  - -22 % PDAP
  - 0.1 % max error



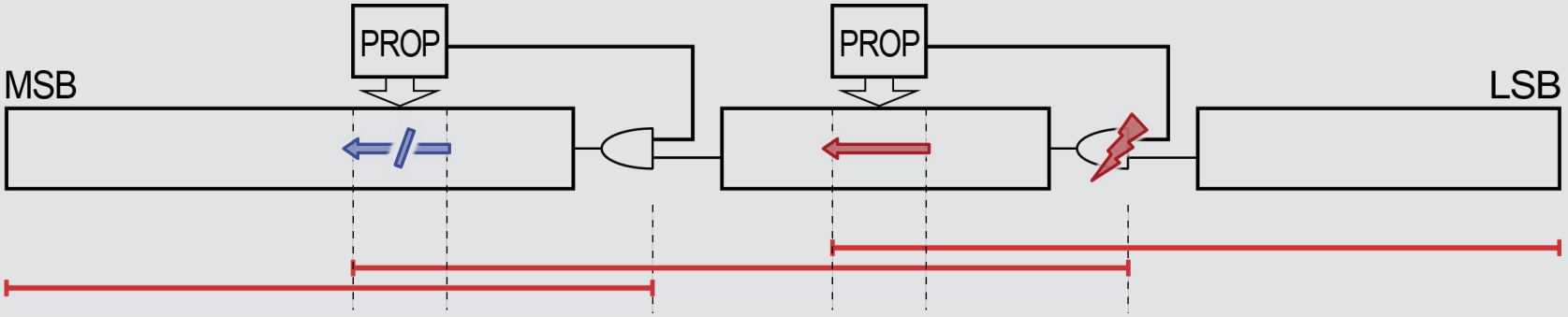
4. CCB Adder – Comparison

Comparison of 32-bit adders (800 MHz, 65 nm techno)

Architecture	Max error (%)	Energy (fJ)	PDAP
Exact	0	79	100
ETBA	6	50	59
ISA		49 (-18%)	47 (-45%)
CCB adder		41 (-16%)	33 (-30%)
ETBA	0.4	82	117
ISA		69 (-24%)	83 (-42%)
CCB adder		62 (-10%)	68 (-18%)
ISA	0.2	78	99
CCB adder	0.1	**68** (-13%)	**78** (-21%)

- **State-of-the-Art**
  - Medium savings
  - Inefficient for low errors (high overhead)
- **CBB adder**
  - High savings
  - Decent savings for low errors (marginal overhead)

CCB Adder – Summary



- **Principle**
  - *Carry Cut-Back* technique
  - False-path carry chain
  - Floating-point-type precision
- **Results**
  - 30-45 % better than existing
  - 22 % savings 0.1 % error (IEEE half-precision equivalent)

New concept

Circuit functionality and timing co-designed by introduction and exploitation of false paths


Thank you!