

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

Vincent Camus  
Jeremy Schlachter  
Christian Enz

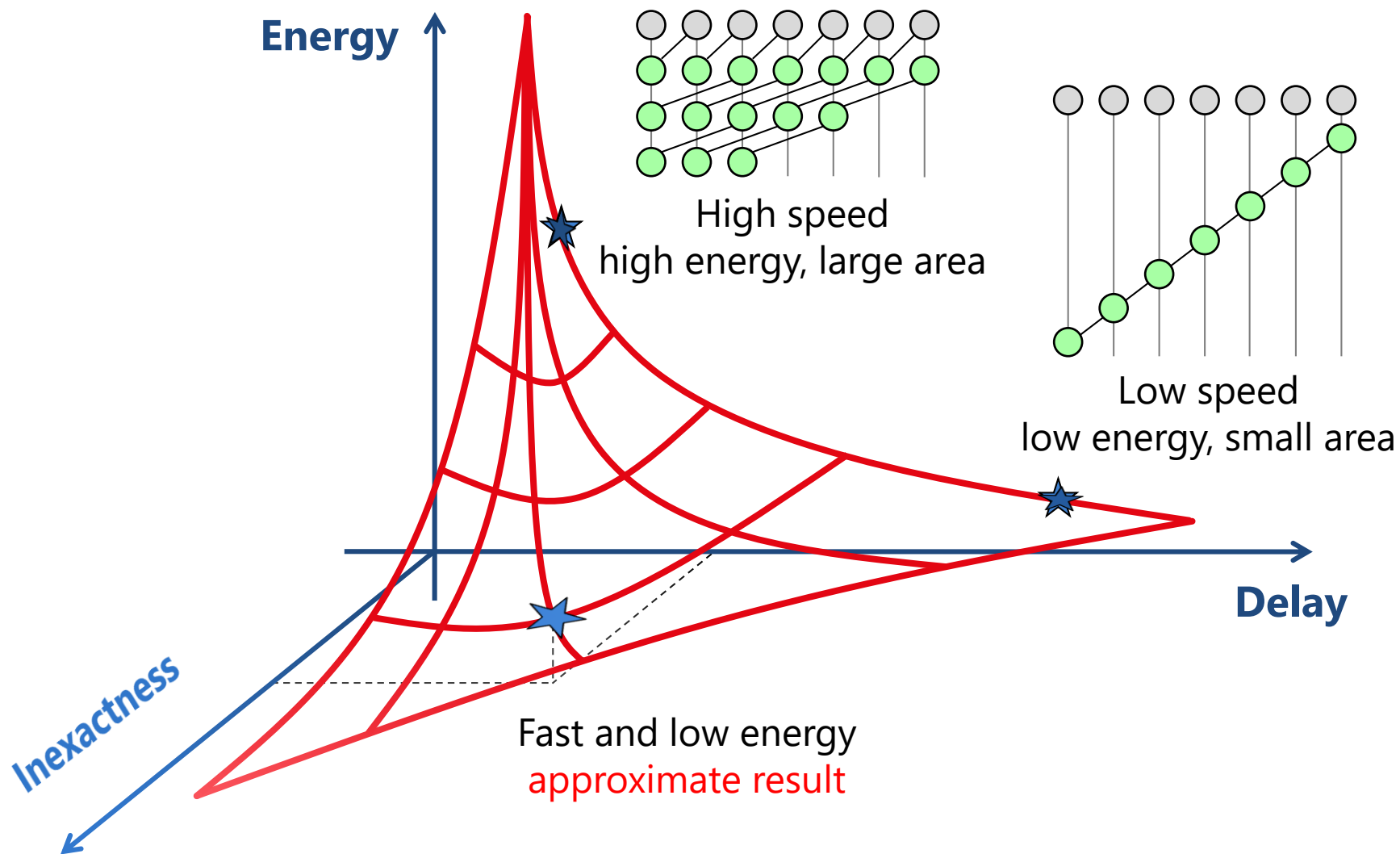
Ecole Polytechnique Fédérale de Lausanne (EPFL)  
Integrated Circuits Laboratory (ICLAB)  
Switzerland

# Outline

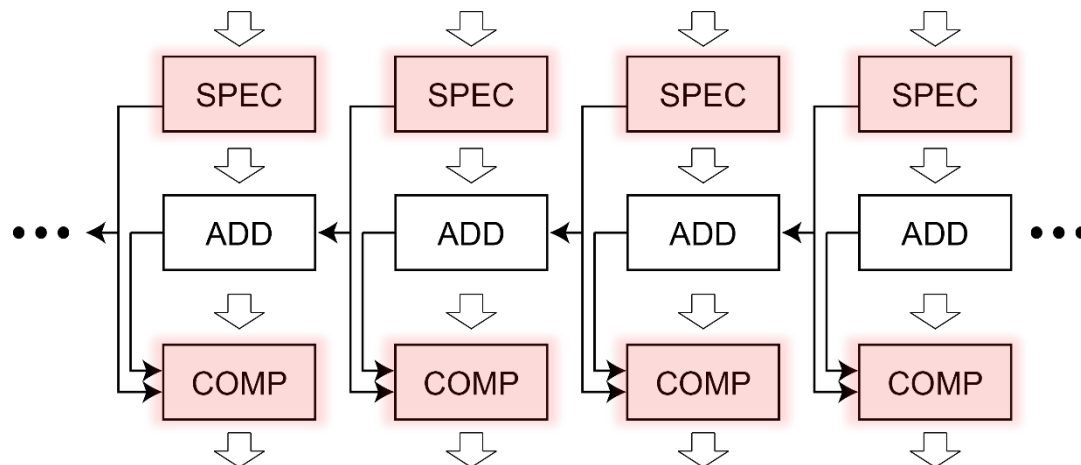
---

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

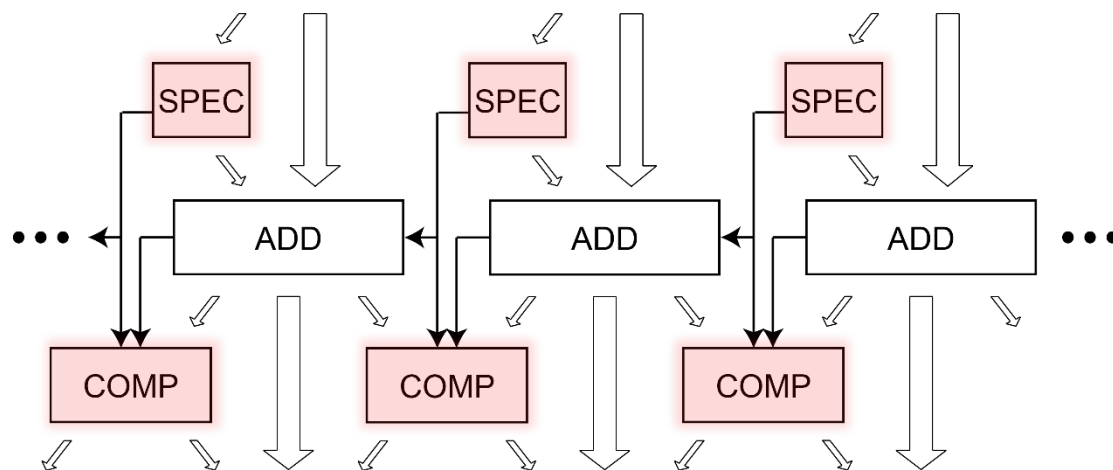
# Approximate Circuits, a New Dimension



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



ETBA, Weber 2013



ISA, Camus 2015

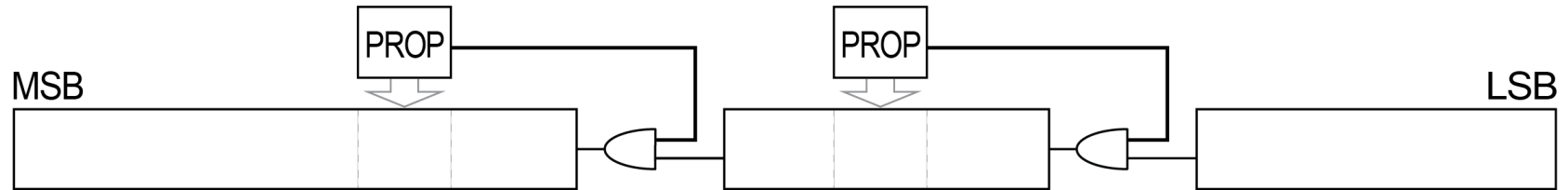
- **Principle**
  - Sliced structure
  - Speculated carry
  - Error compensation
- **Advantages**
  - High speed
  - Worst-case error control
- **Drawbacks**
  - Hardware overhead
  - Delay overhead

## 2. CCB Adder – Circuit



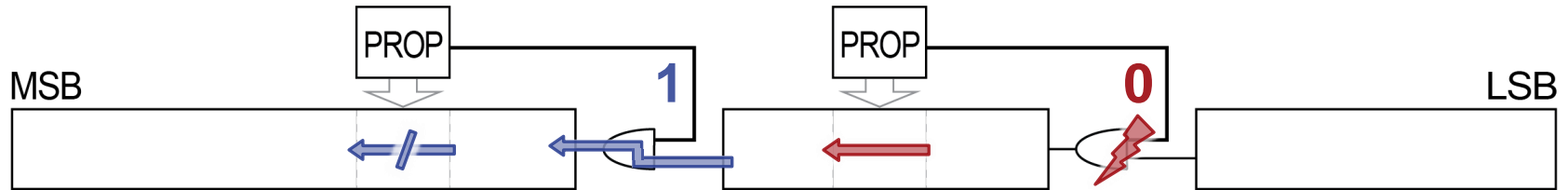
- **Principle**
  - Monitoring high-significance carry stages

## 2. CCB Adder – Circuit



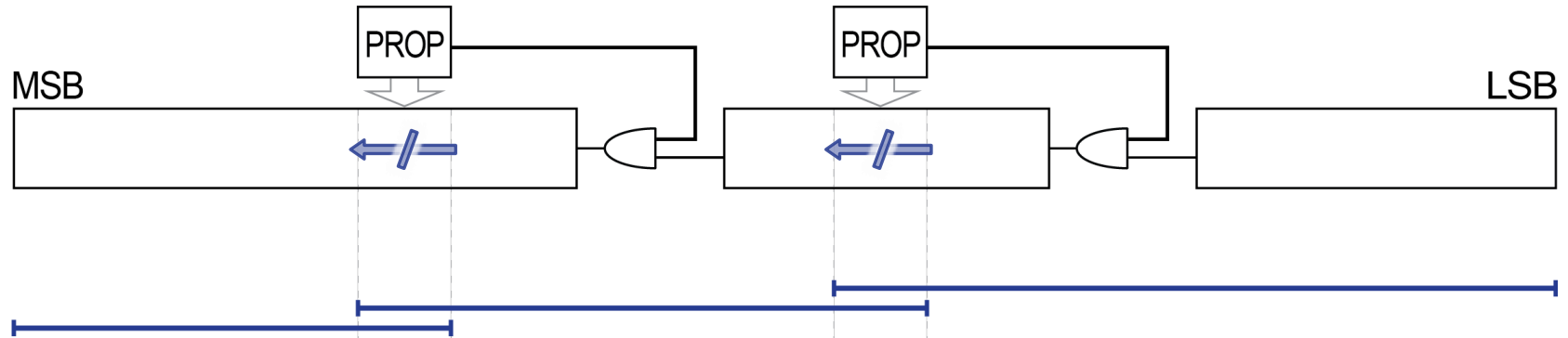
- **Principle**
  - Monitoring high-significance carry stages
  - Cutting the carry chain at low-significance positions

## 2. CCB Adder – Circuit



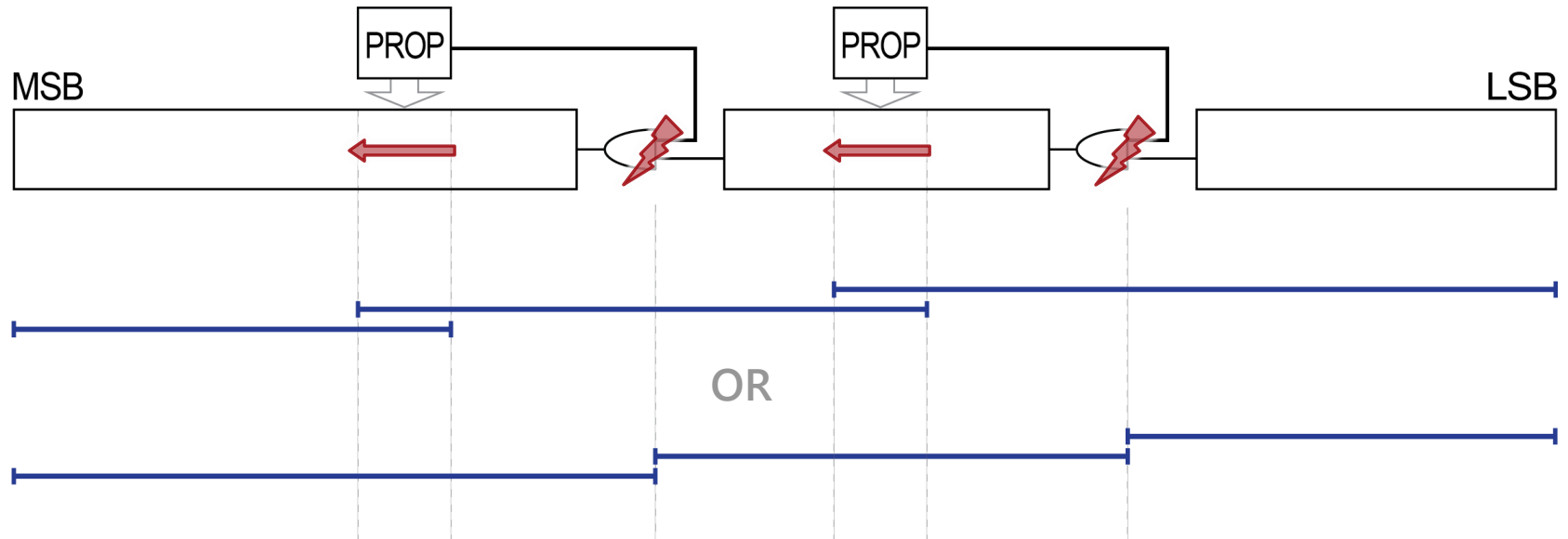
- **Principle**
  - Monitoring high-significance carry stages
  - Cutting the carry chain at low-significance positions
- **Carry propagation**
  - Naturally doesn't propagate
  - Artificially cut back

## 2. CCB Adder – Circuit

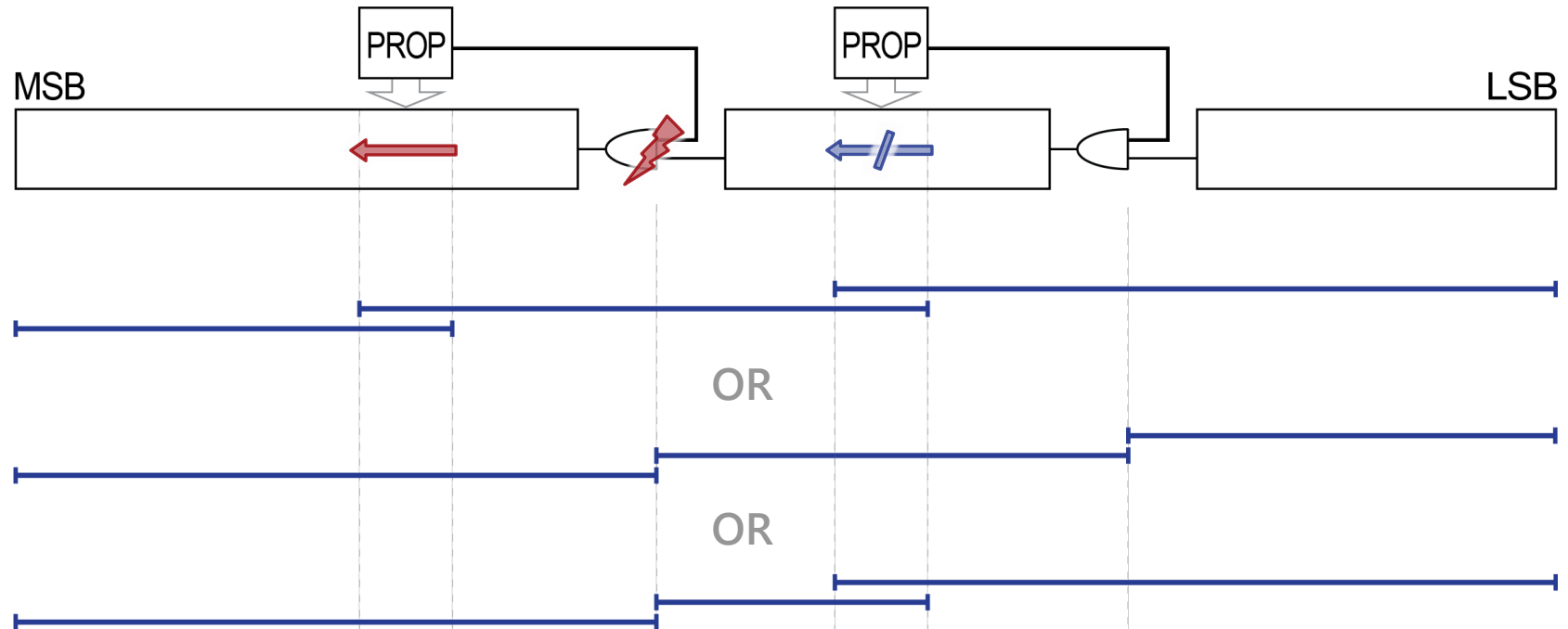




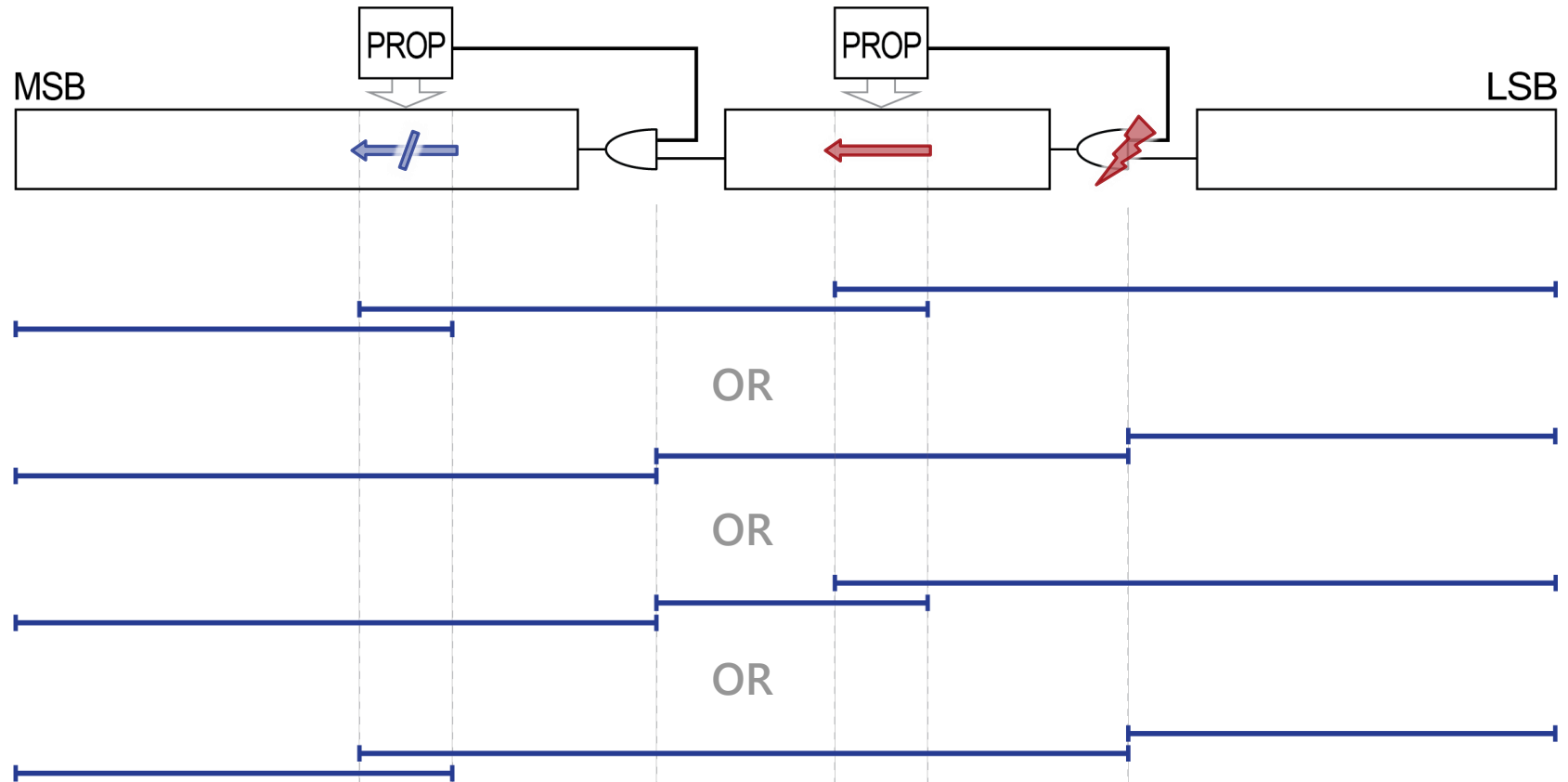
## 2. CCB Adder – Circuit



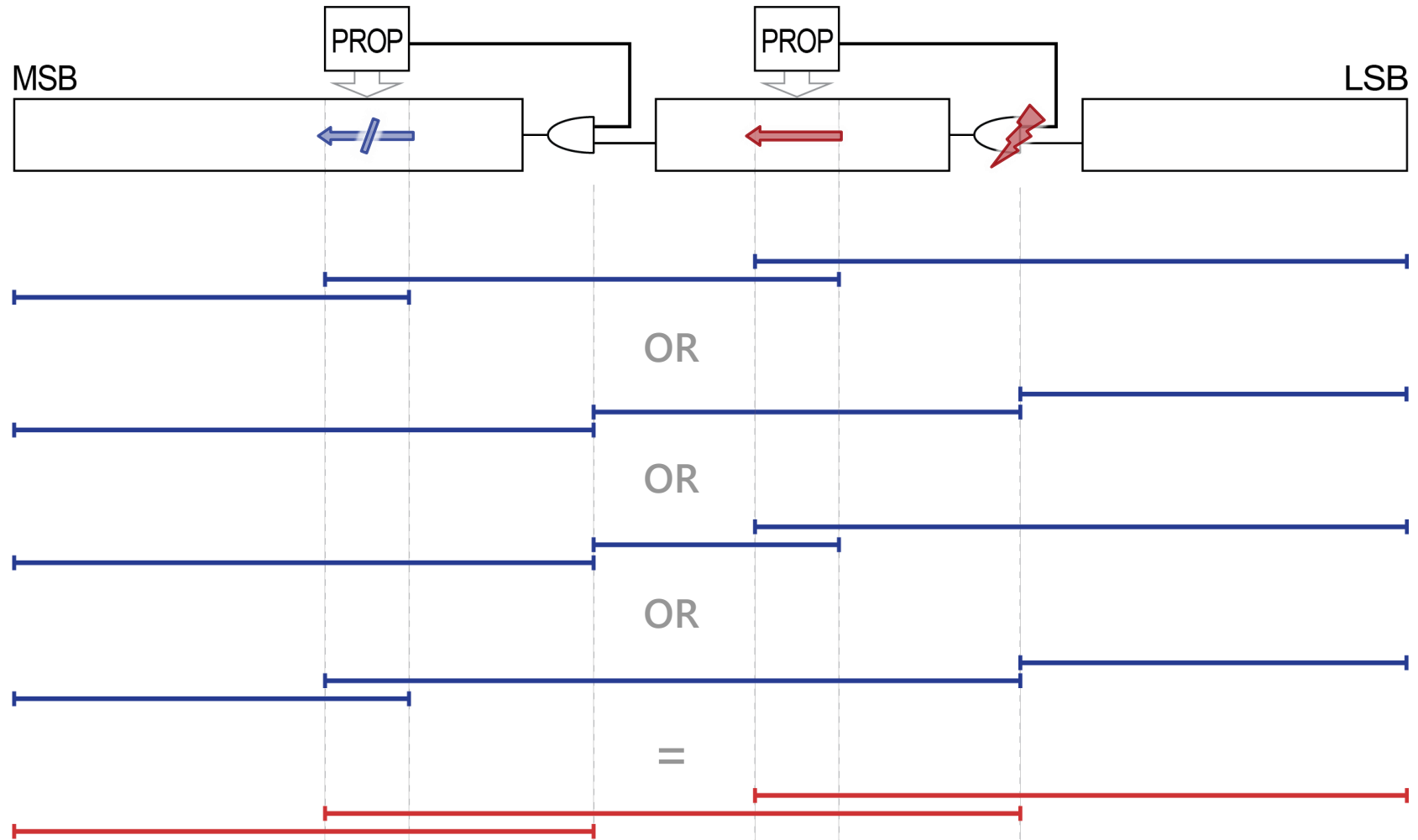
## 2. CCB Adder – Circuit



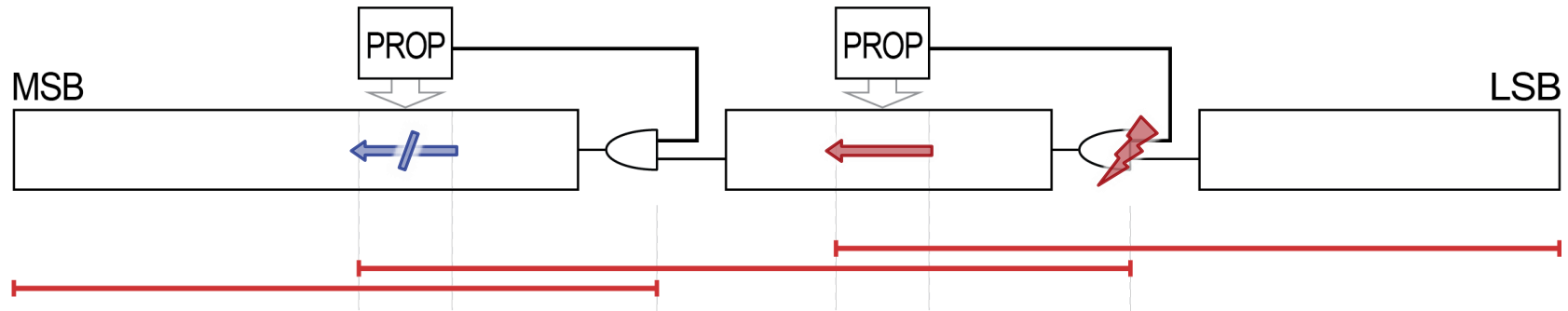
## 2. CCB Adder – Circuit



## 2. CCB Adder – Circuit



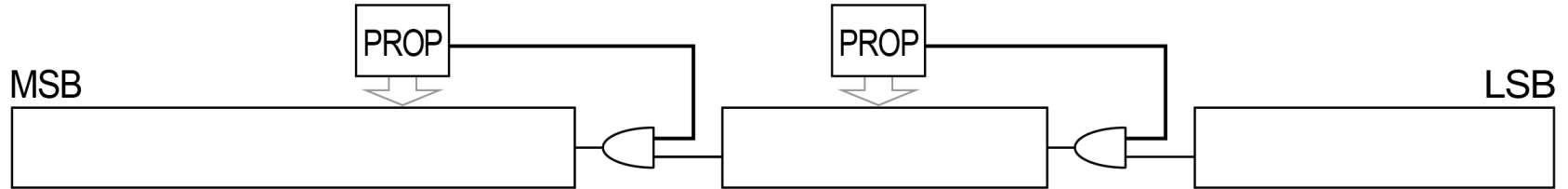
## 2. CCB Adder – Circuit



*The full 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

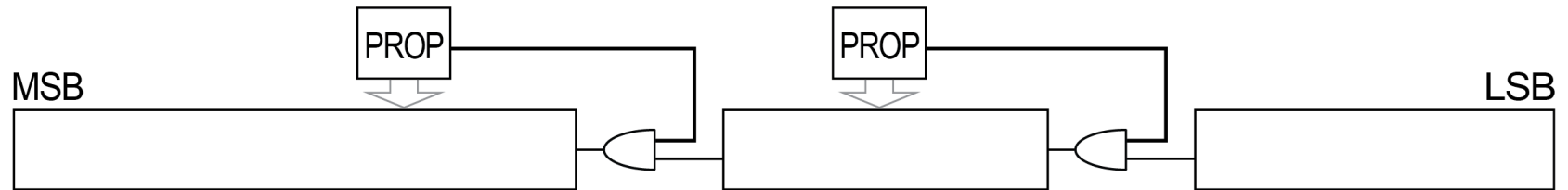
### 3. CCB Adder – Arithmetic



- **Arithmetic principle**

[illegible]

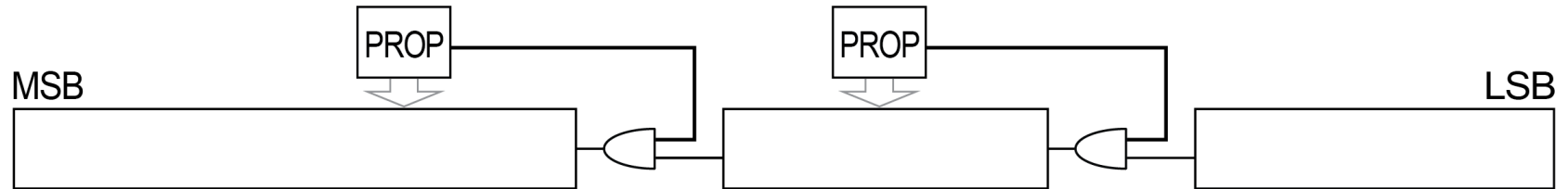
### 3. CCB Adder – Arithmetic



- **Arithmetic principle**

Diagram illustrating the effect of a carry-in on the carry propagation of a 16-bit ripple-carry adder. The diagram shows two 16-bit numbers being added. The first number is 00101000 (8) and the second is 00010010 (2). The sum is 00111010 (10). The carry-in is 1. The carry propagation is shown with a solid line for the first 8 bits and a dashed line for the next 8 bits. The carry-out is 1. The diagram is labeled "Inexact" and "Expected".

### 3. CCB Adder – Arithmetic

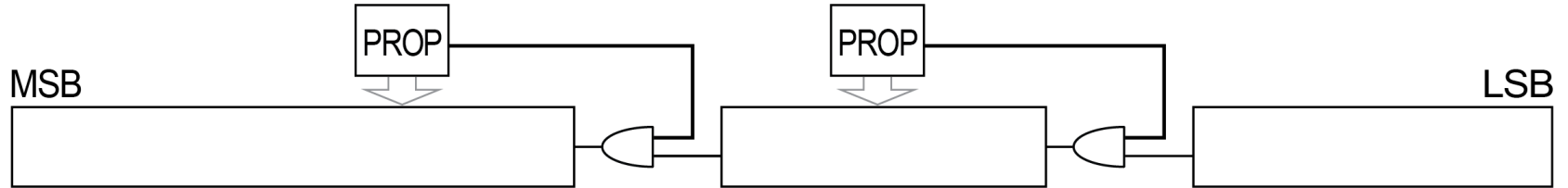


- **Arithmetic principle**

Diagram illustrating the propagation of a carry bit  $P$  through a chain of adders. The diagram shows two rows of 16-bit numbers being added. The first row is 00101000 + 00010010. The second row is 00000000 + 00000000. The result of the first addition is 00111010, which is labeled 'Inexact'. The result of the second addition is 00000000, which is labeled 'Expected'. A carry bit  $P$  is shown propagating from the first addition to the second. A carry bit  $K$  is shown propagating from the second addition to the third. The diagram uses solid lines for the first addition and dashed lines for the second addition.



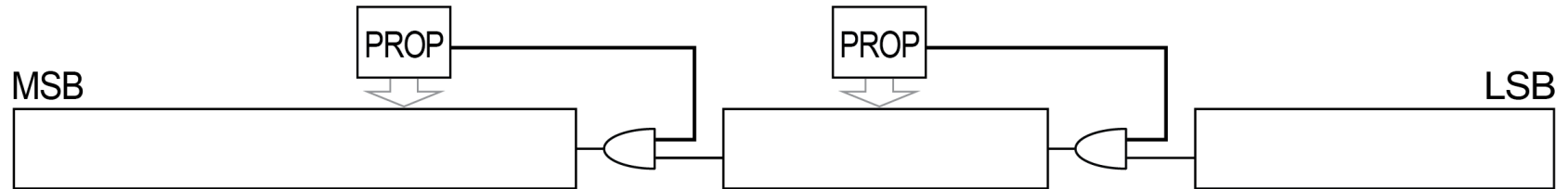
### 3. CCB Adder – Arithmetic



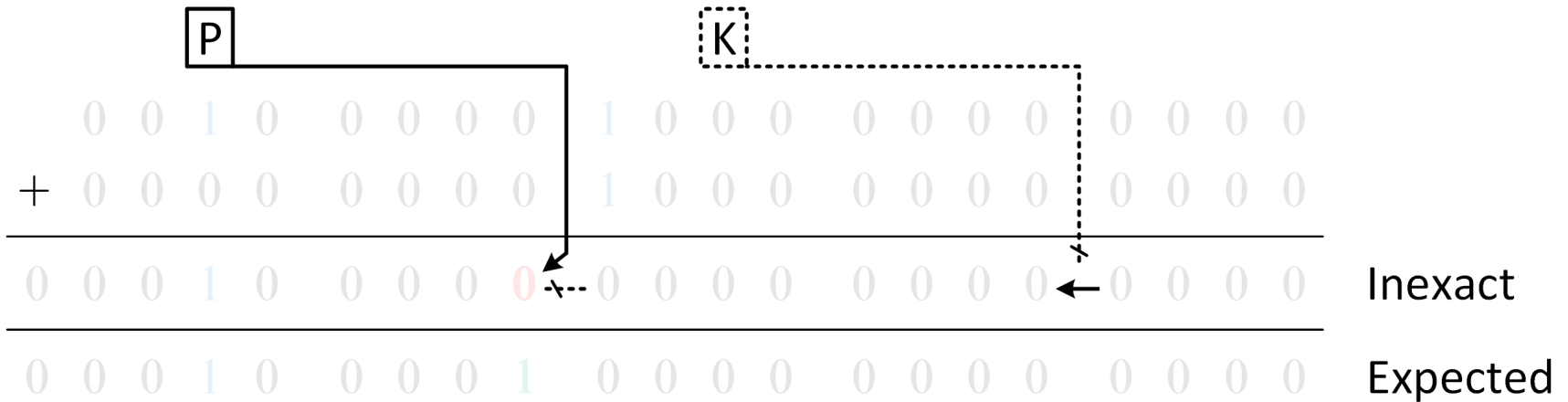
- **Arithmetic principle**

Diagram illustrating a carry chain in a 16-bit addition. The first 8-bit result (P) has a carry-out of 1 (red '0' with a dot). This carry propagates through the second 8-bit result (K), which also has a carry-out of 1 (red '0' with a dot). The final 16-bit result has a carry-out of 0 (green '1' with a dot). The labels 'P' and 'K' are above the first and second 8-bit results respectively. Arrows indicate the carry propagation from the first 8-bit result to the second, and from the second to the final 16-bit result.

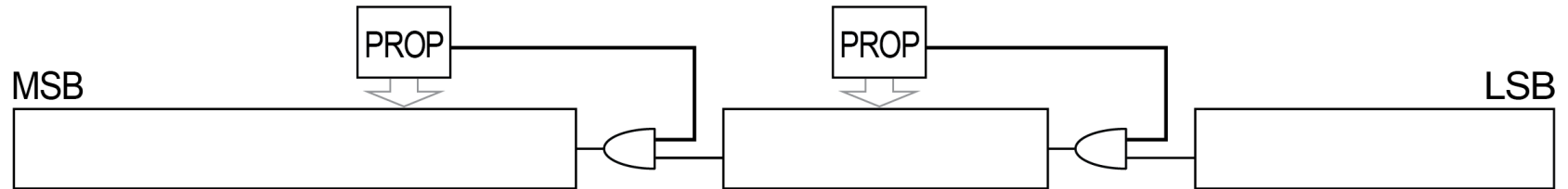
### 3. CCB Adder – Arithmetic



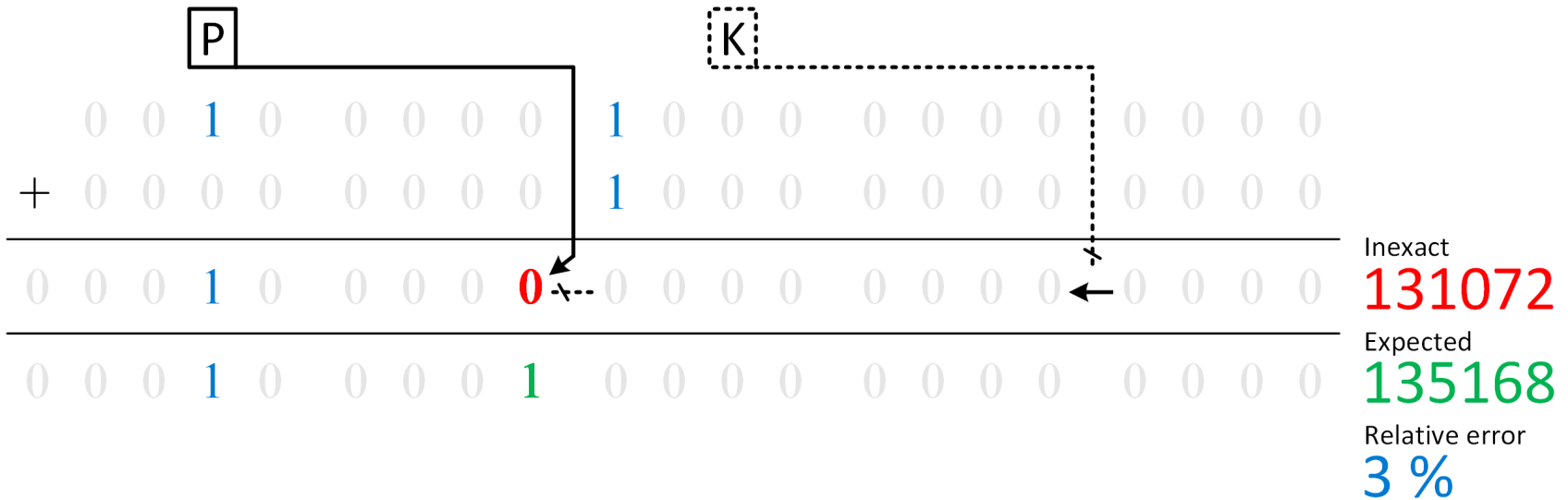
- **Worst-case error**



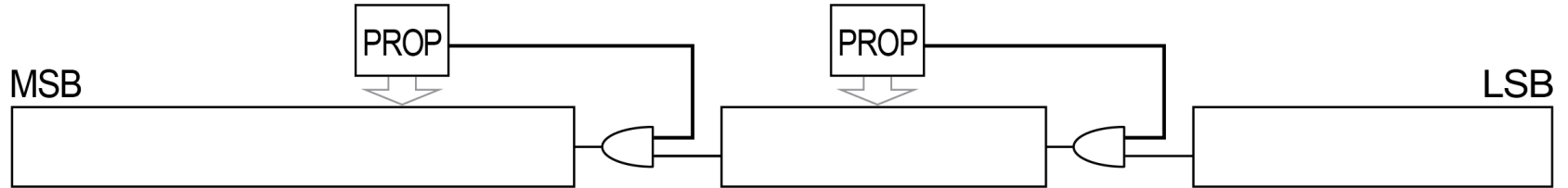
### 3. CCB Adder – Arithmetic



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



### 3. CCB Adder – Arithmetic



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

Diagram illustrating the effect of a 3% relative error on a 16-bit floating-point number. The diagram shows two addition operations, each involving a carry bit (P) and a carry-in (C).

**Operation 1 (Left):**

- Carry bit (P): 1
- Carry-in (C): 0
- Input 1: 0 0 1 0 0 0 0 0
- Input 2: 0 0 0 0 0 0 0 0
- Result (Inexact): 0 0 0 1 0 0 0 0 (Red)
- Expected Result: 0 0 0 1 0 0 0 1 (Green)

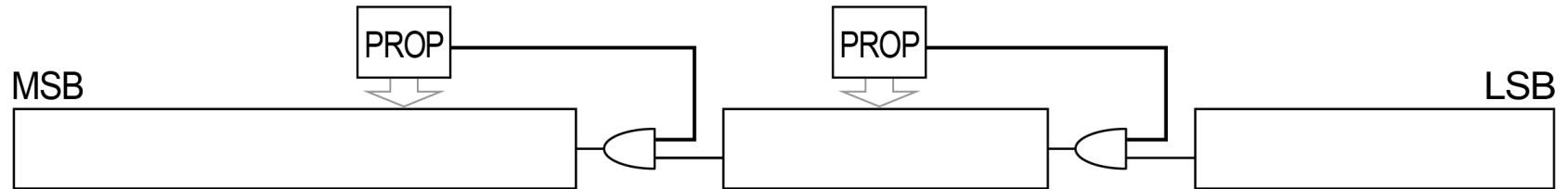
**Operation 2 (Right):**

- Carry bit (P): 1
- Carry-in (C): 1
- Input 1: 1 0 1 0 0 0 0 0
- Input 2: 1 0 0 0 0 0 0 0
- Result (Inexact): 1 0 1 0 0 0 0 0 (Red)
- Expected Result: 1 0 1 0 0 0 0 1 (Green)

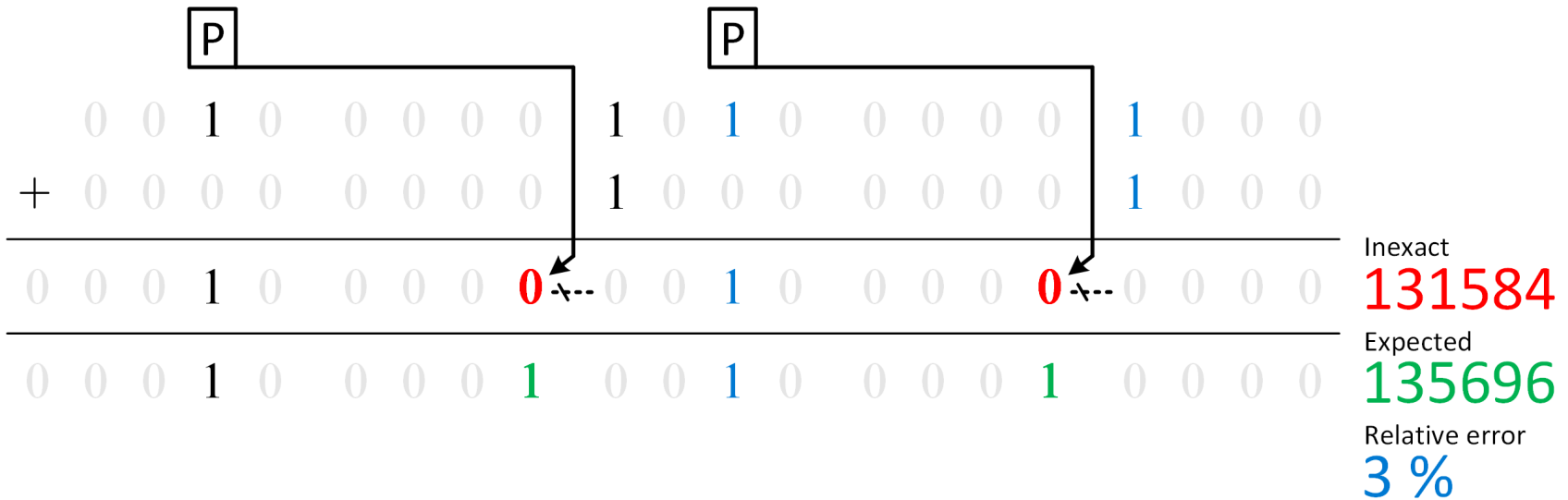
**Summary:**

- Inexact: 131584 (Red)
- Expected: 135696 (Green)
- Relative error: 3% (Blue)

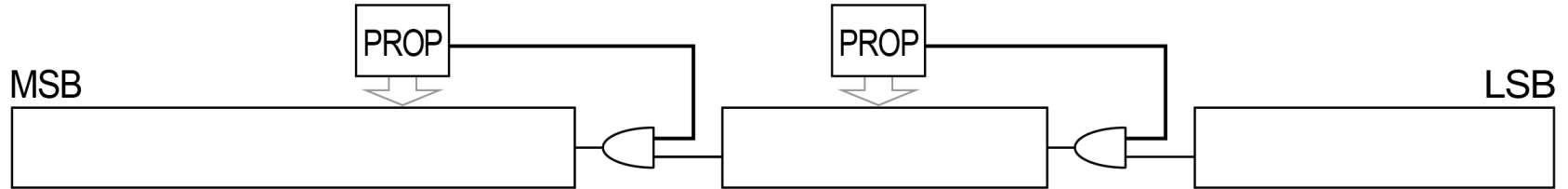
### 3. CCB Adder – Arithmetic



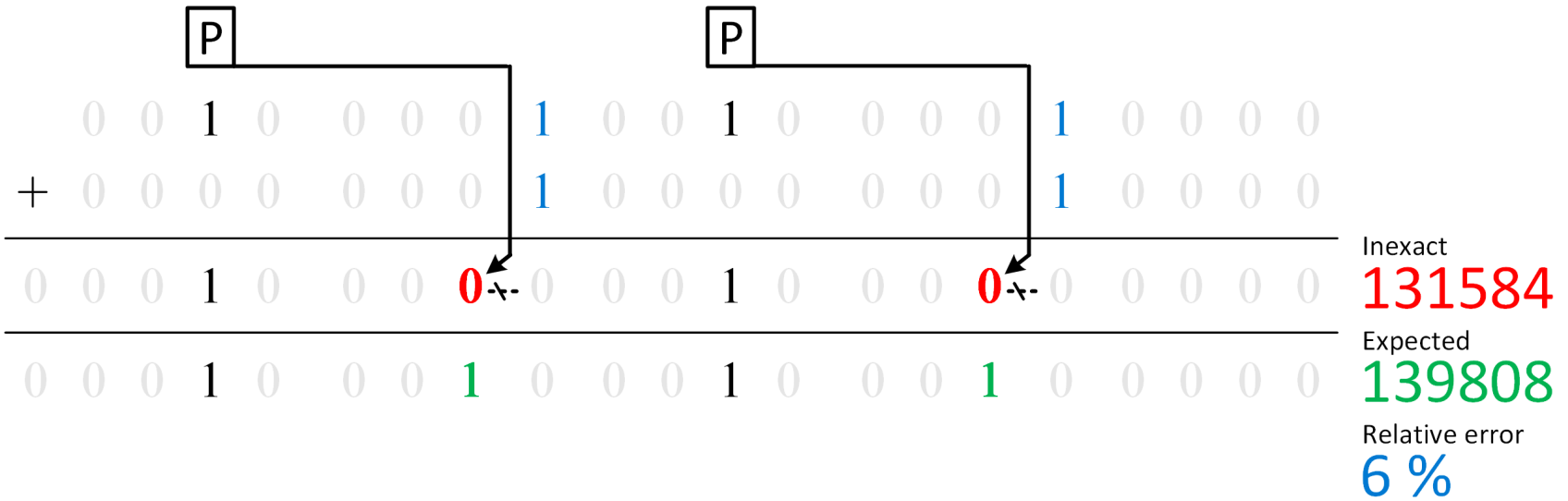
- **Error control: *cut-back* length: 5 bits**



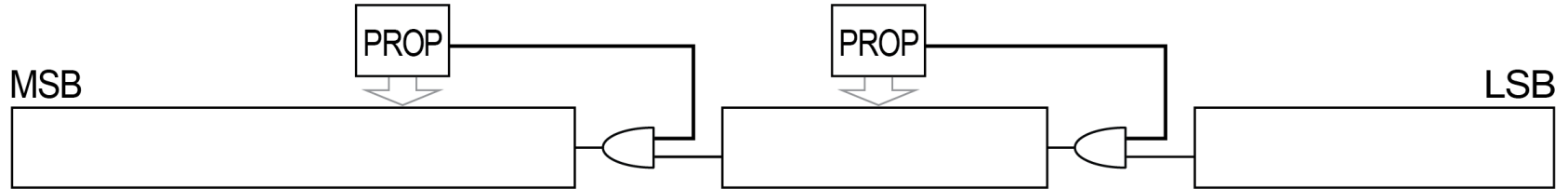
### 3. CCB Adder – Arithmetic



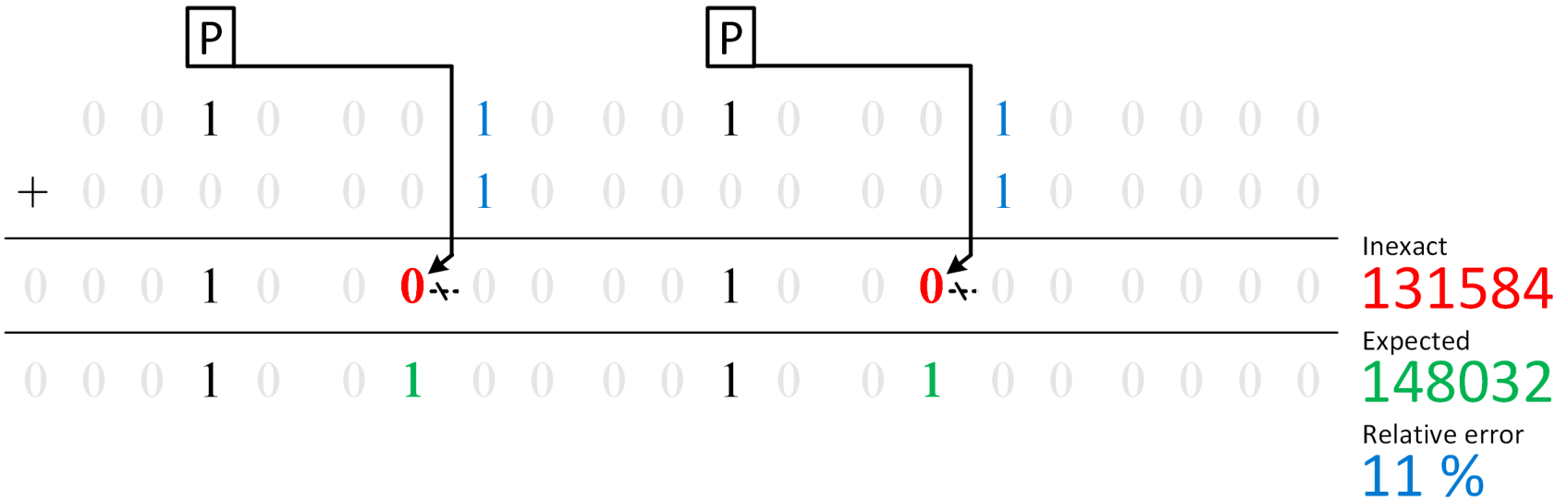
- **Error control: *cut-back* length: 4 bits**



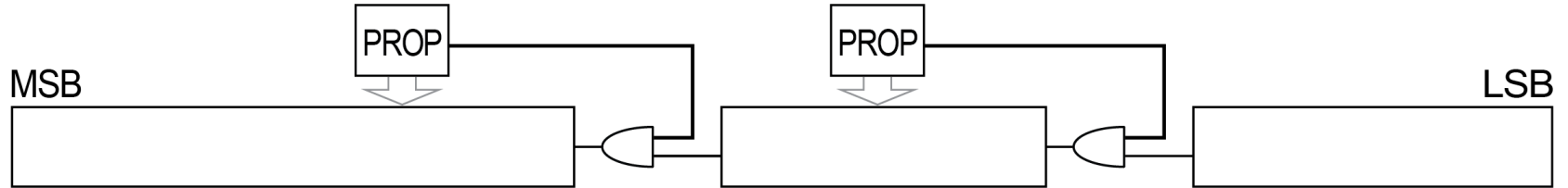
### 3. CCB Adder – Arithmetic



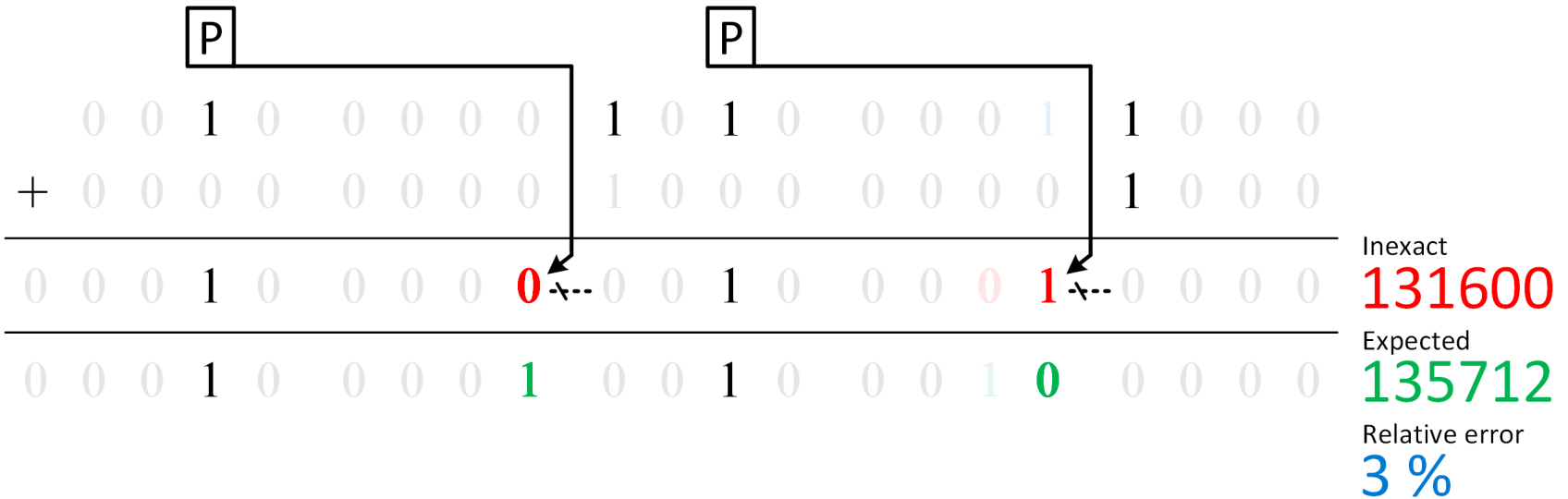
- **Error control: *cut-back* length: 3 bits**



## Extra slide – Arithmetic

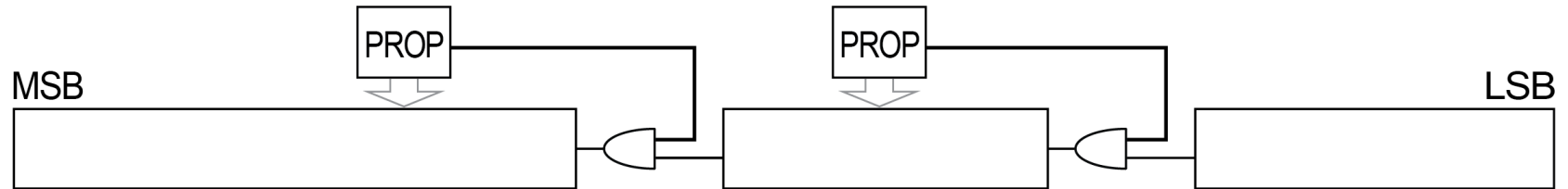


- **Propagating error: 1 bit**

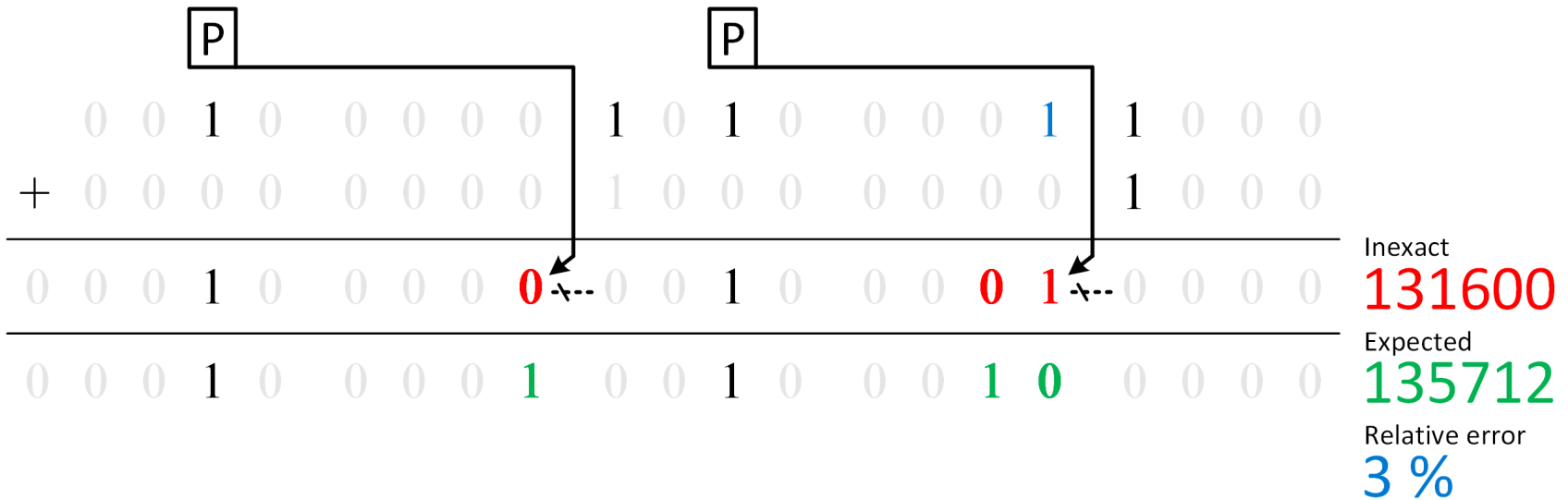




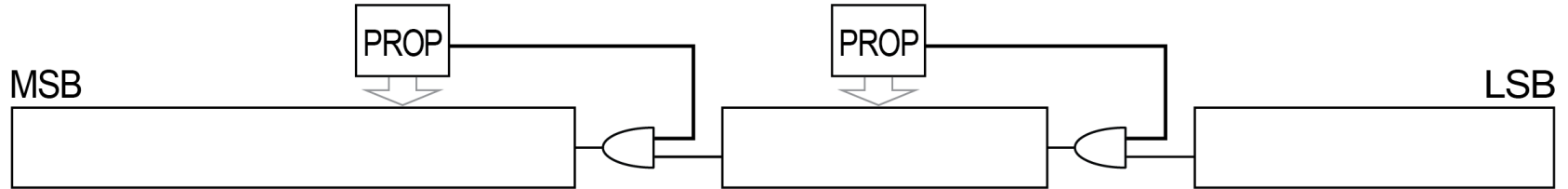
## Extra slide – Arithmetic



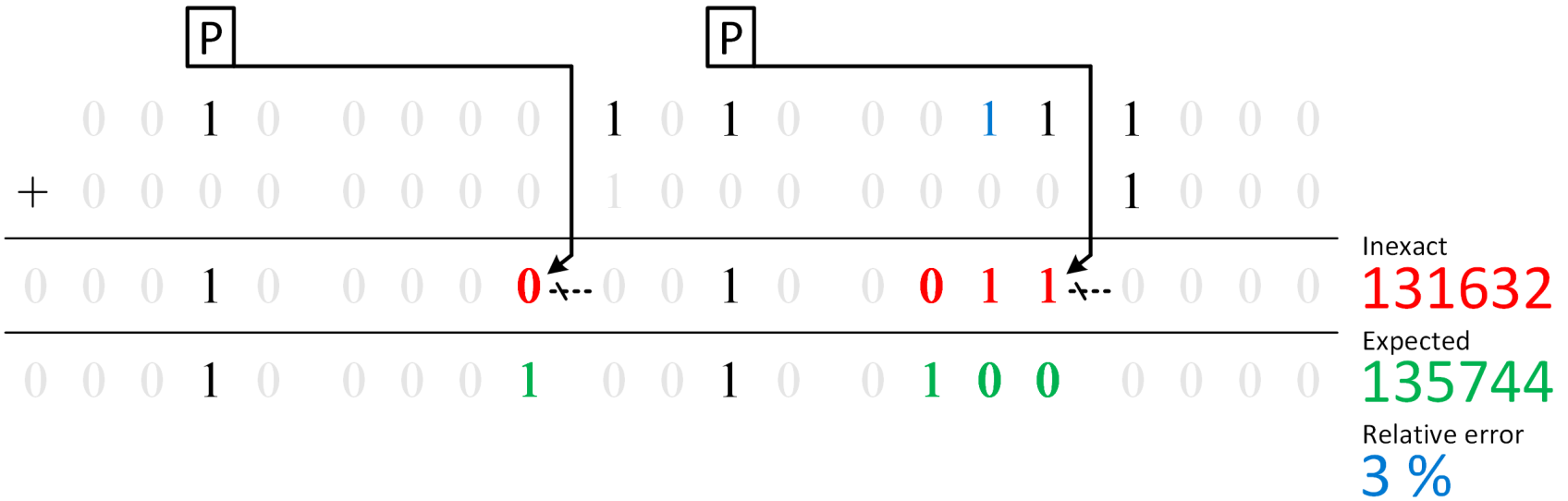
- **Propagating error: 2 bits**



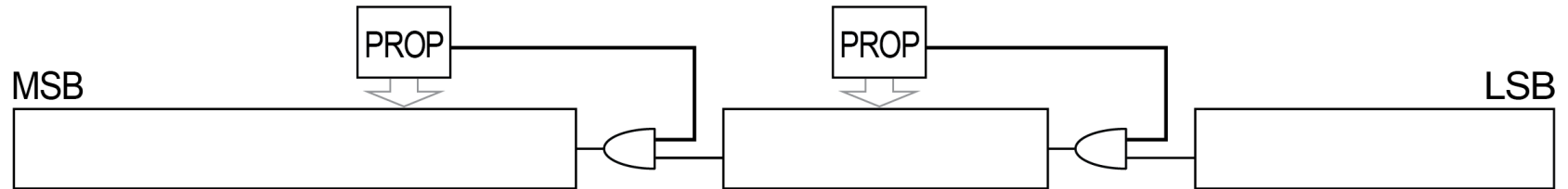
## Extra slide – Arithmetic



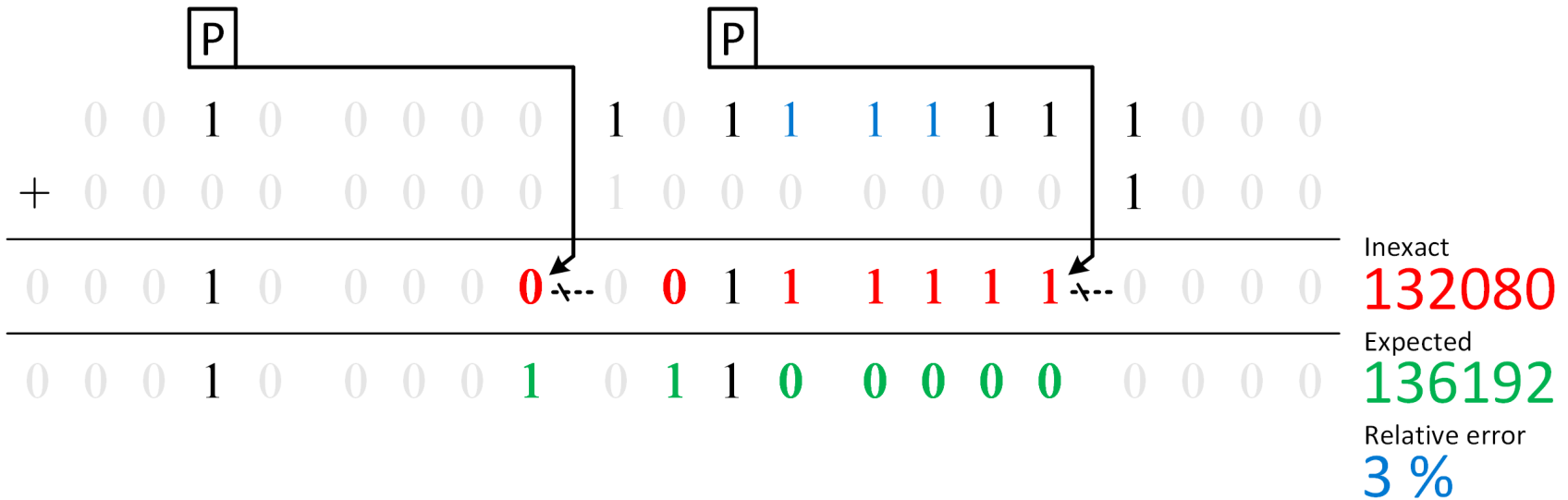
- **Propagating error: 3 bits**



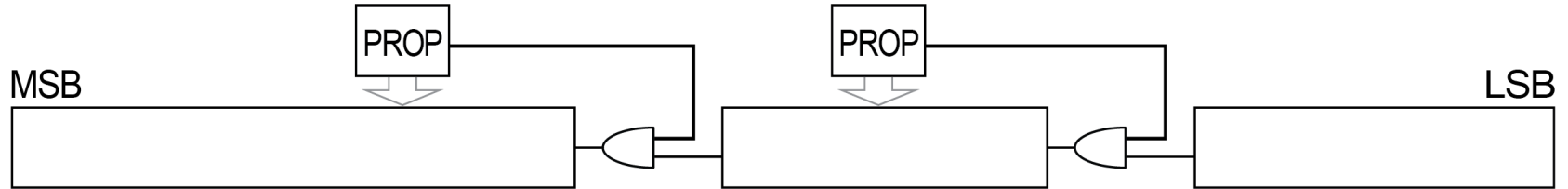
## Extra slide – Arithmetic



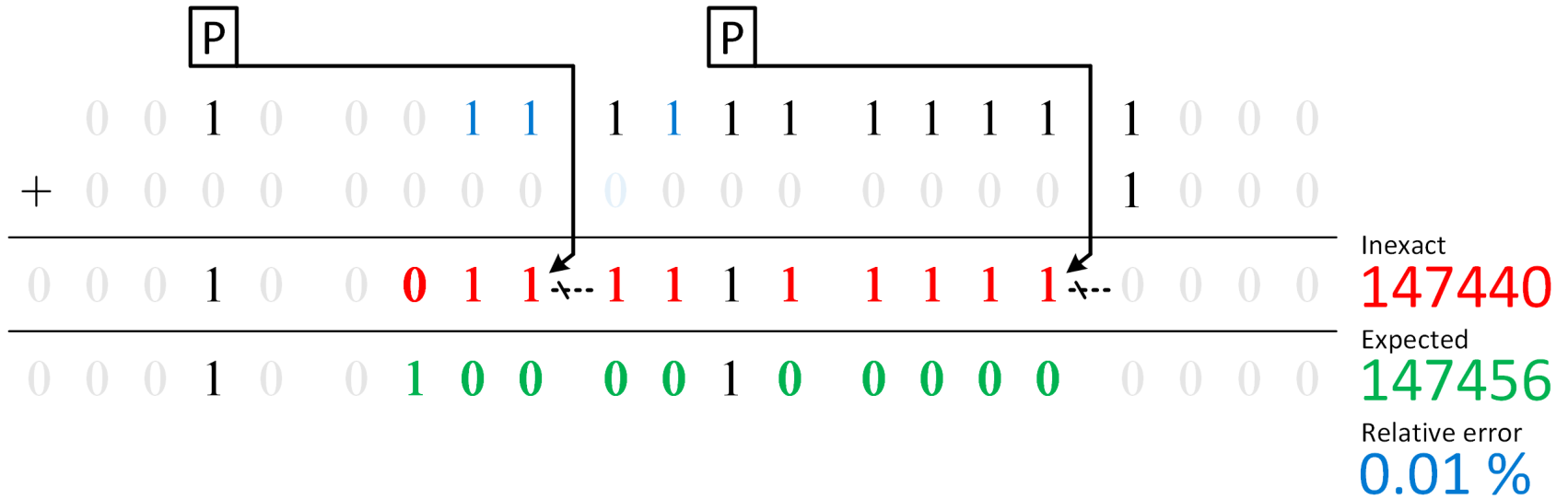
- **Propagating error: many bits, without increasing errors**



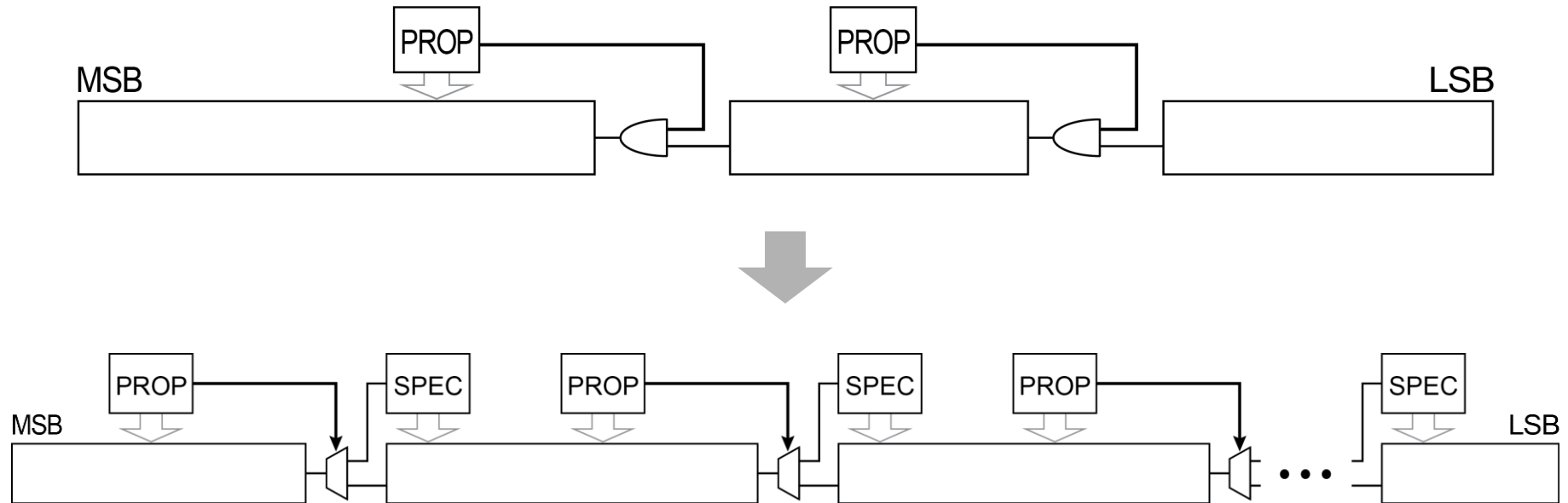
## Extra slide – Arithmetic



- **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

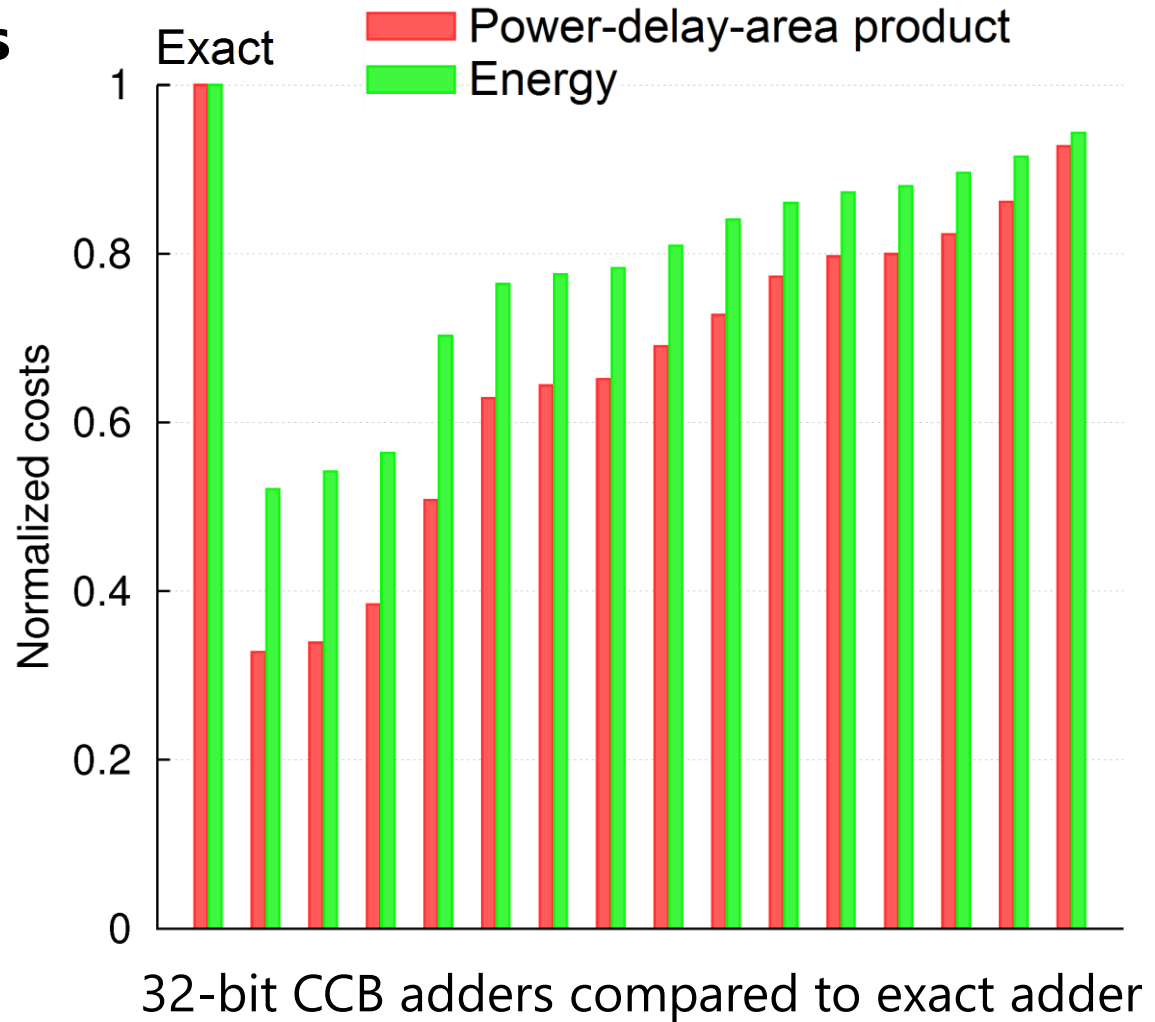
## 4. CCB Adder – Results

- **Implementations**

- 32 bits
- 65 nm techno
- 800 MHz

- **Metrics**

- Energy
- PDAP



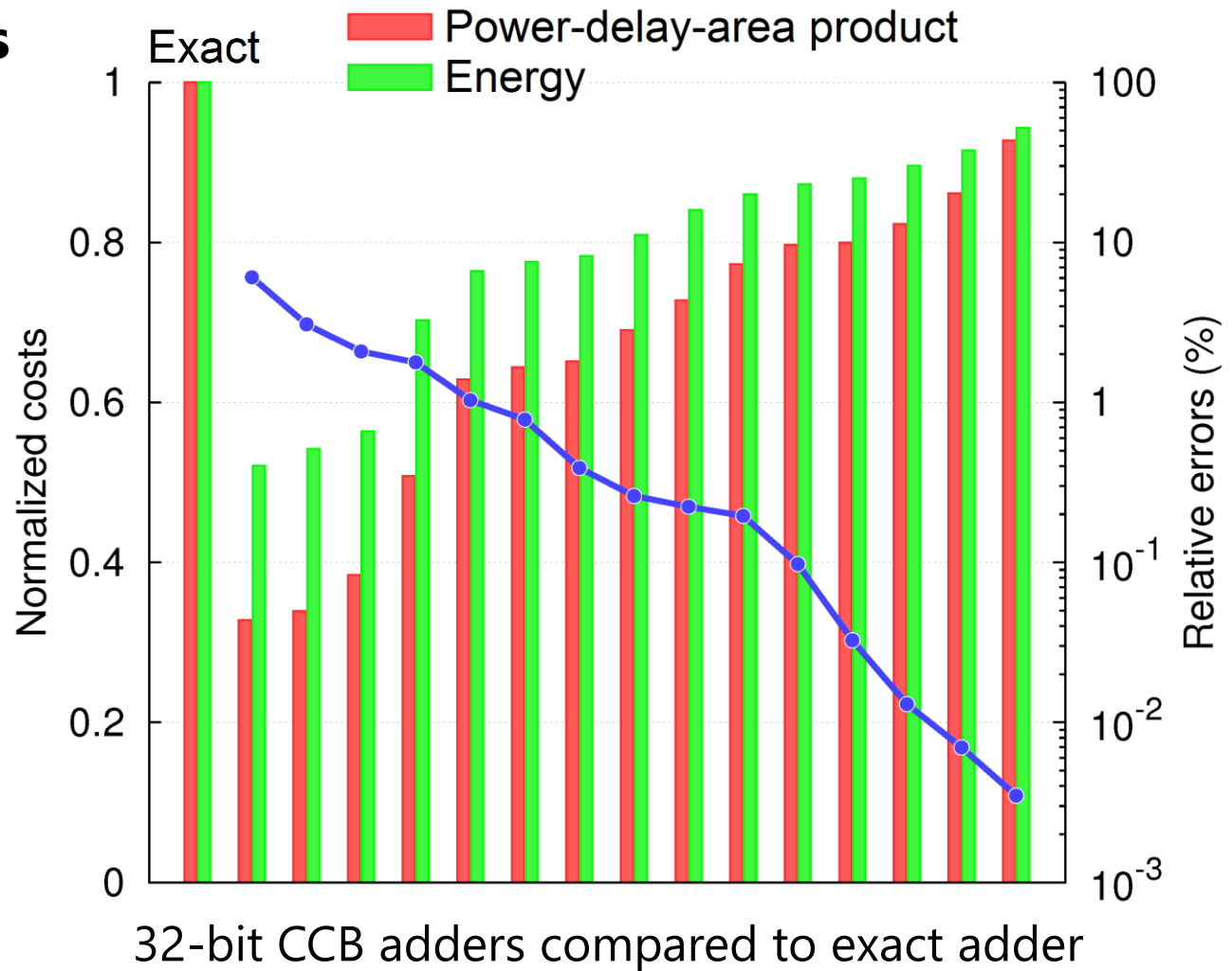
## 4. CCB Adder – Results

- **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**

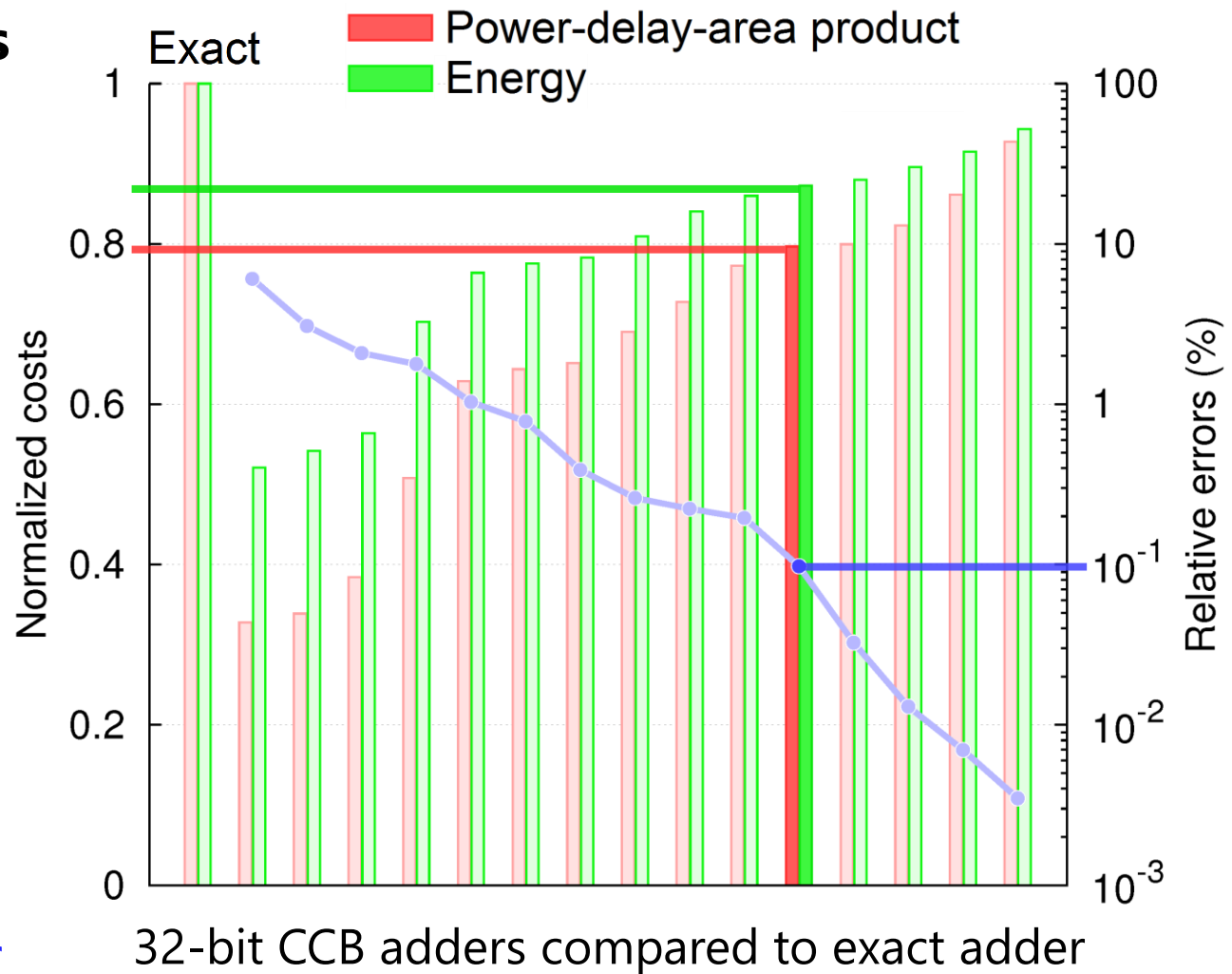
- Energy
- PDAP
- Max error

- **Savings**

-14 % energy

-22 % PDAP

0.1 % max error





## 4. CCB Adder – Results

- **Implementations**

- 32 bits
- 65 nm techno
- 800 MHz

- **Metrics**

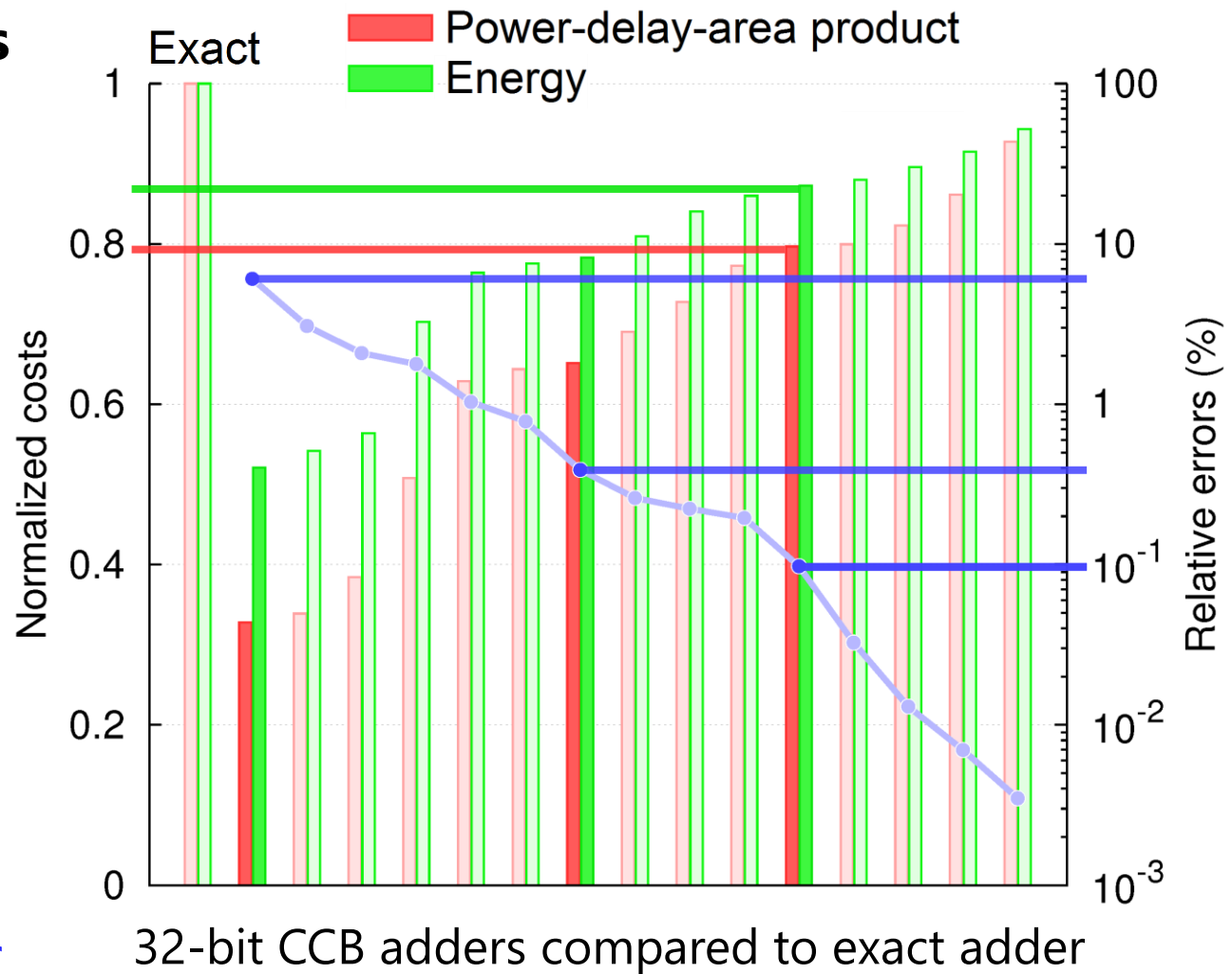
- Energy
- PDAP
- Max error

- **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
<b>CCB adder</b>	0.1	<b>68</b>  -13 %	<b>78</b>  -21 %

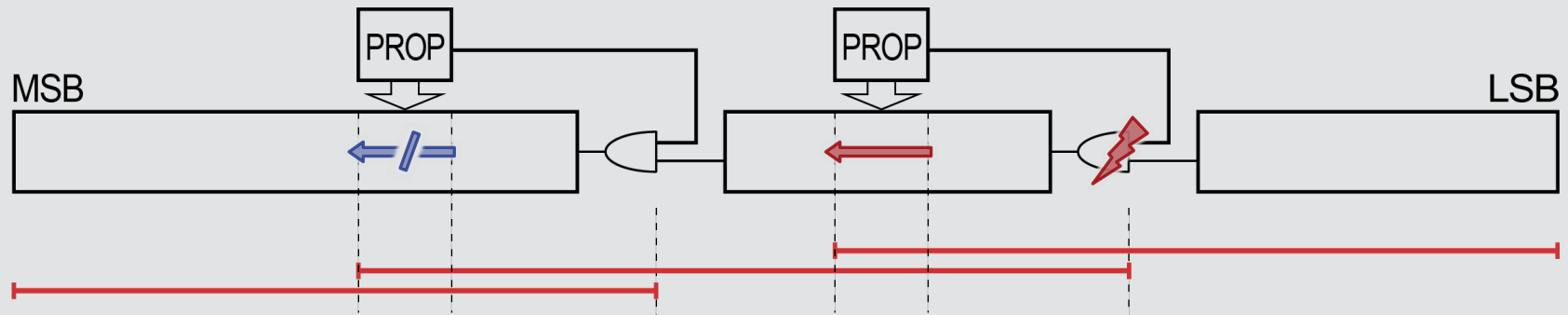
- **State-of-the-Art**

- 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!**