Fused Three-Input SORN Arithmetic

Moritz Bärthel[1] Chen Yuxing[1] Nils Hülsmeier[1] Jochen Rust[2] Steffen Paul[1]

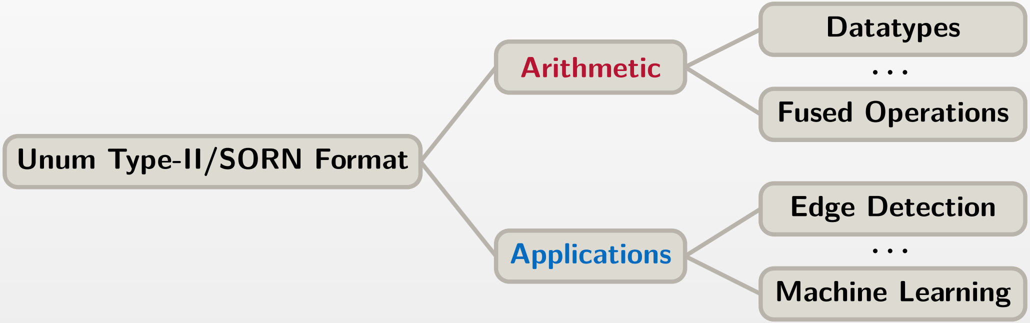
[1]Institute of Electrodynamics and Microelectronics (ITEM.me), University of Bremen, Germany

[2]DSI Aerospace Technologie GmbH, Bremen, Germany

2023 Conference on Next Generation Arithmetic (CoNGA)
Singapore, Singapore, March 1-2, 2023

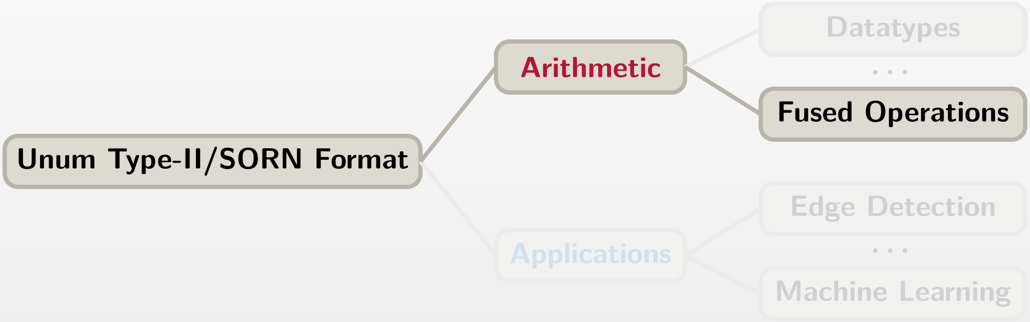
Sets-Of-Real-Numbers (SORN)

- interval-based number format derived from unum type-II
- low hardware complexity due to LUT-based arithmetic
- application specific due to low precision



Sets-Of-Real-Numbers (SORN)

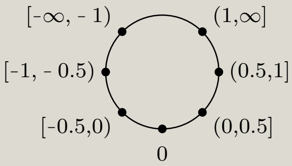
- interval-based number format derived from unum type-II
- low hardware complexity due to LUT-based arithmetic
- application specific due to low precision



SORN Design Flow

- application-specific SORN datatype encoded as binary representation

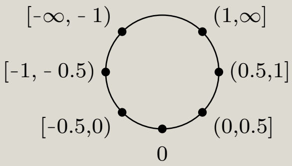
Datatype



SORN Design Flow

- application-specific SORN datatype encoded as binary representation
- arithmetic operations realized as lookup-tables (LUTs)

Datatype



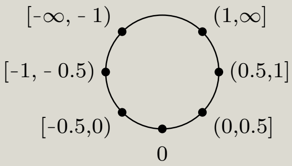
LUT

×	[-∞, - 1) 1000000	...	(1,∞] 0000001
[-∞, - 1) 1000000	0000001	...	1000000
⋮	⋮	⋱	⋮
(1,∞] 0000001	1000000	...	0000001

SORN Design Flow

- application-specific SORN datatype encoded as binary representation
- arithmetic operations realized as lookup-tables (LUTs)
- implementation of arithmetic LUTs on RTL/gate level with simple logic circuits

Datatype



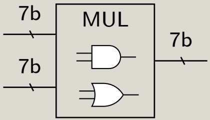
→

LUT

×	[-∞, - 1) 1000000	…	(1,∞] 0000001
[-∞, - 1) 1000000	0000001	…	1000000
⋮	⋮	⋱	⋮
(1,∞] 0000001	1000000	…	0000001

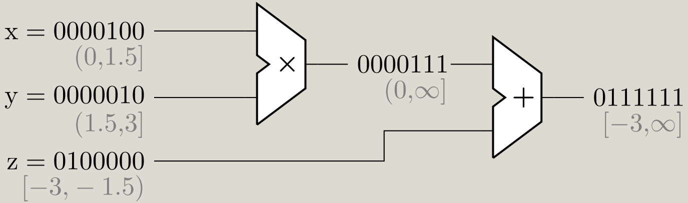
→

Logic



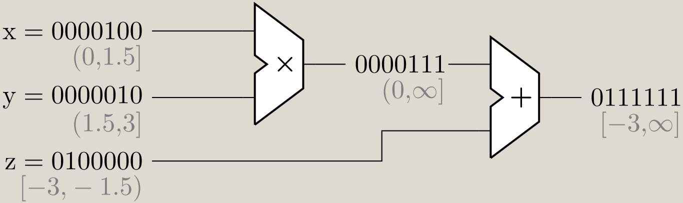
Expanding SORN Intervals

Multiply-Add $(x \times y) + z$ *Datatype:* $[-\infty,-3]$ $(-3,-1.5]$ $(-1.5,0]$ 0 $(0,1.5]$ $(1.5,3]$ $(3,\infty]$



Expanding SORN Intervals

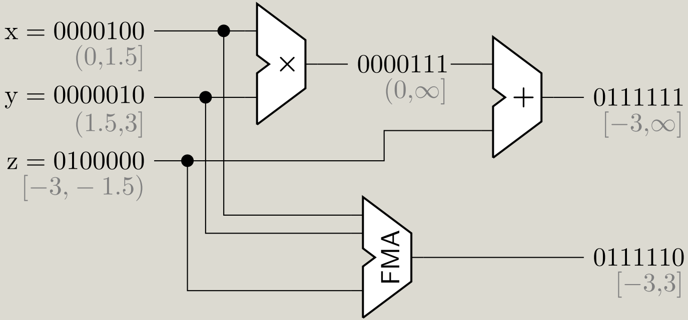
Multiply-Add $(x \times y) + z$ ***Datatype:*** $[-\infty,-3]$ $(-3,-1.5]$ $(-1.5,0]$ 0 $(0,1.5]$ $(1.5,3]$ $(3,\infty]$



$\Rightarrow$ intermediate rounding after multiplication: $(0,4.5] \rightarrow (0,\infty]$

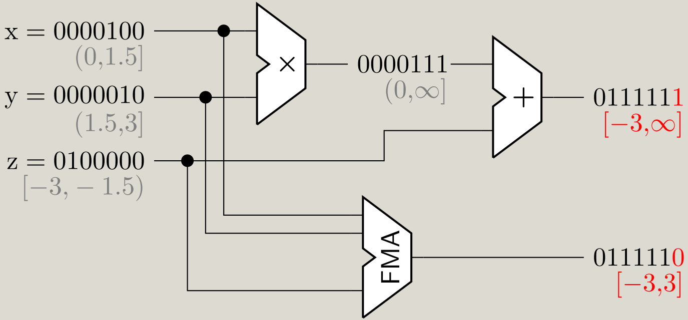
Expanding SORN Intervals

Multiply-Add $(x \times y) + z$ *Datatype:* $[-\infty,-3]$ $(-3,-1.5]$ $(-1.5,0]$ 0 $(0,1.5]$ $(1.5,3]$ $(3,\infty]$



Expanding SORN Intervals

Multiply-Add $(x \times y) + z$ *Datatype:* $[-\infty,-\mathbf{3}]$ $(-\mathbf{3},-\mathbf{1.5}]$ $(-\mathbf{1.5},\mathbf{0}]$ $\mathbf{0}$ $(\mathbf{0},\mathbf{1.5}]$ $(\mathbf{1.5},\mathbf{3}]$ $(\mathbf{3},\infty]$



SORN Datatypes in Evaluation

- 6 different SORN datatypes with
  - 7b, 9b and 11b
  - linear interval spacing
  - small value ranges (label A) and large value ranges (label B)

label	config
lin-9-A	$[-\infty, -1)\ \ldots\ [-1/3,0)\ 0\ (0,1/3]\ (1/3,2/3]\ (2/3,1]\ (1,\infty]$
lin-9-B	$[-\infty, -120)\ \ldots\ [-40,0)\ 0\ (0,40]\ (40,80]\ (80,120]\ (120,\infty]$

Evaluation: Three-Input Addition, Multiplication and Multiply-add

- RTL designs for 6 different SORN datatypes
- accuracy vs. hardware complexity (CMOS 28nm post-synthesis)
- non-fused vs. fused

Evaluation: Three-Input Addition, Multiplication and Multiply-add

- RTL designs for 6 different SORN datatypes
- accuracy vs. hardware complexity (CMOS 28nm post-synthesis)
- non-fused vs. fused

Definition: SORN accuracy

The accuracy of a SORN value is measured by the **width of the represented interval**, which is equivalent to the **number of consecutive one-bits** in a SORN value.

Evaluation: Three-Input Addition, Multiplication and Multiply-add

- RTL designs for 6 different SORN datatypes
- accuracy vs. hardware complexity (CMOS 28nm post-synthesis)
- non-fused vs. fused

Definition: SORN accuracy

The accuracy of a SORN value is measured by the **width of the represented interval**, which is equivalent to the **number of consecutive one-bits** in a SORN value.

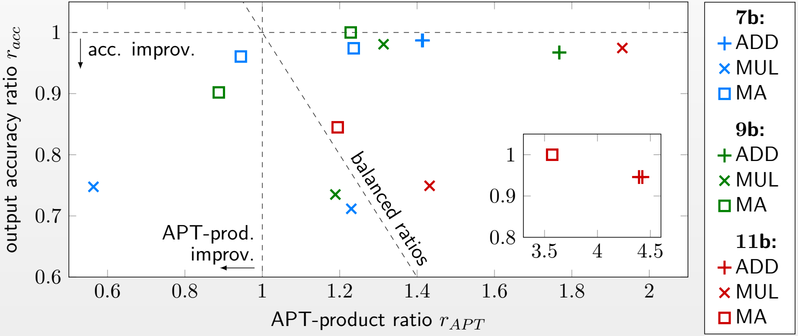
output accuracy ratio

$$r_{acc} = \frac{\text{fused mean out width}}{\text{non-fused mean out width}}$$

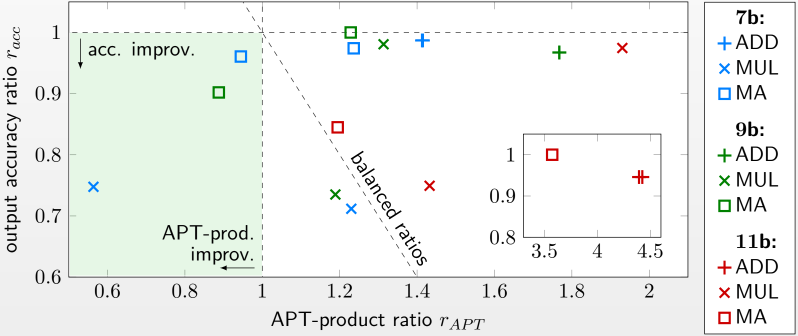
APT-product ratio

$$r_{APT} = \frac{\text{fused APT } [\mu\text{m}^2 \times \mu\text{W} \times \text{ns}]}{\text{non-fused APT } [\mu\text{m}^2 \times \mu\text{W} \times \text{ns}]}$$

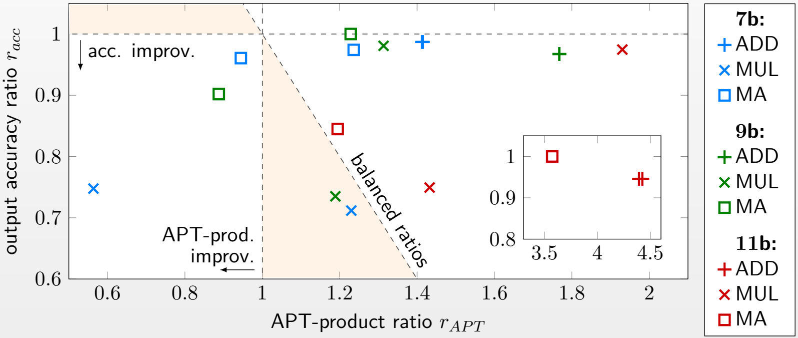
Evaluation: Three-Input Addition, Multiplication and Multiply-add



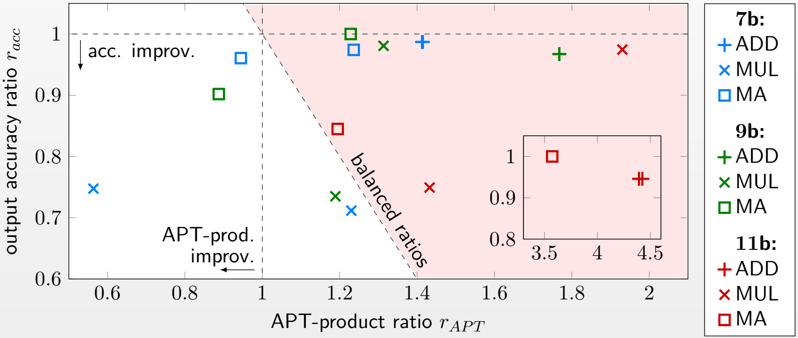
Evaluation: Three-Input Addition, Multiplication and Multiply-add



Evaluation: Three-Input Addition, Multiplication and Multiply-add

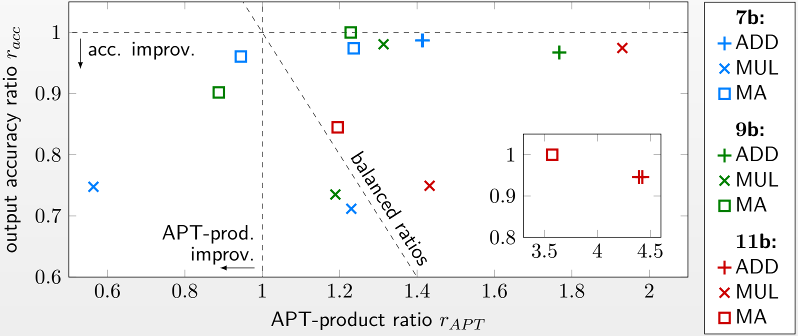


Evaluation: Three-Input Addition, Multiplication and Multiply-add



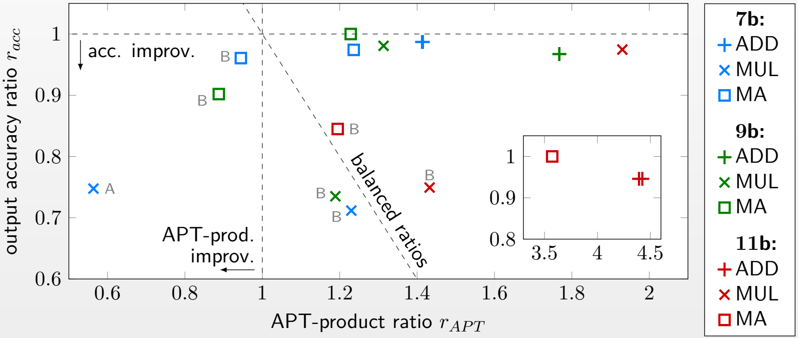
Evaluation: Three-Input Addition, Multiplication and Multiply-add

- accuracy improvements for multiply-add (up to 15%) and multiplication (up to 29%)
- both ratios < 1 for 7b and 9b multiplication and multiply-add
- addition with high APT-product increase



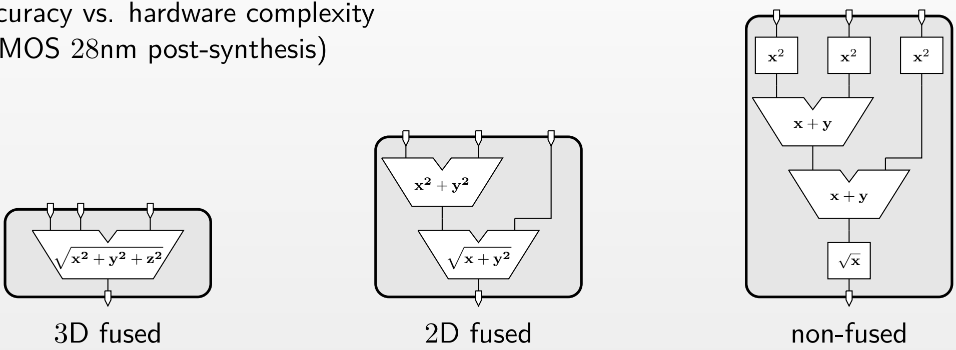
Evaluation: Three-Input Addition, Multiplication and Multiply-add

- better results for DTs with large value range (B)

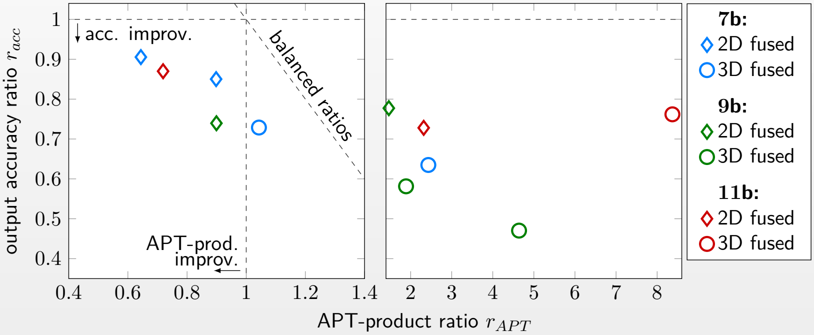


Evaluation: Three-Input Hypot Function $\sqrt{x^2 + y^2 + z^2}$

- fused operations with two inputs (2D) and three inputs (3D)
- RTL designs for 6 different SORN datatypes
- accuracy vs. hardware complexity (CMOS 28nm post-synthesis)

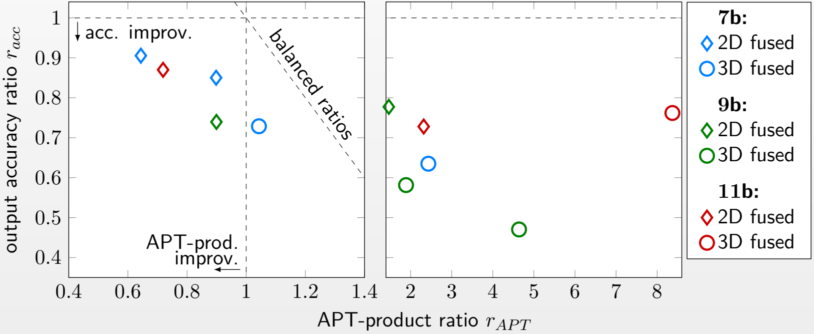


Evaluation: Three-Input Hypot Function $\sqrt{x^2 + y^2 + z^2}$



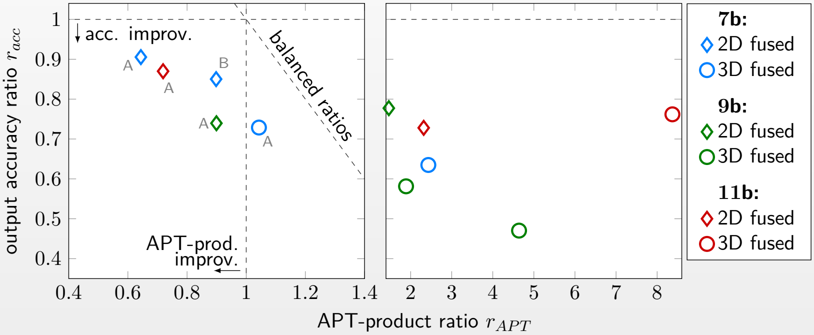
Evaluation: Three-Input Hypot Function $\sqrt{x^2 + y^2 + z^2}$

- up to 27% accuracy improvement for 2D and up to 53% for 3D designs
- very high APT-product increase for most 3D designs
- both ratios < 1 for 2D designs only



Evaluation: Three-Input Hypot Function $\sqrt{x^2 + y^2 + z^2}$

- better results for DTs with small value range (A)



Conclusion

accuracy 👍

- fused three-input SORN LUTs reduce interval growth and improve accuracy
- except addition, all operations show significant accuracy improvements

hardware complexity 👎

- reduced APT-product for some multiplication, multiply-add and 2D hypot designs
- strong complexity increases for addition and most 3D hypot designs

datatypes

- multiplication and multiply-add: better performance for large range DTs (B)
- hypot: better performance for small range DTs (A)

THANK YOU FOR YOUR ATTENTION!

Contact

Moritz Bärthel

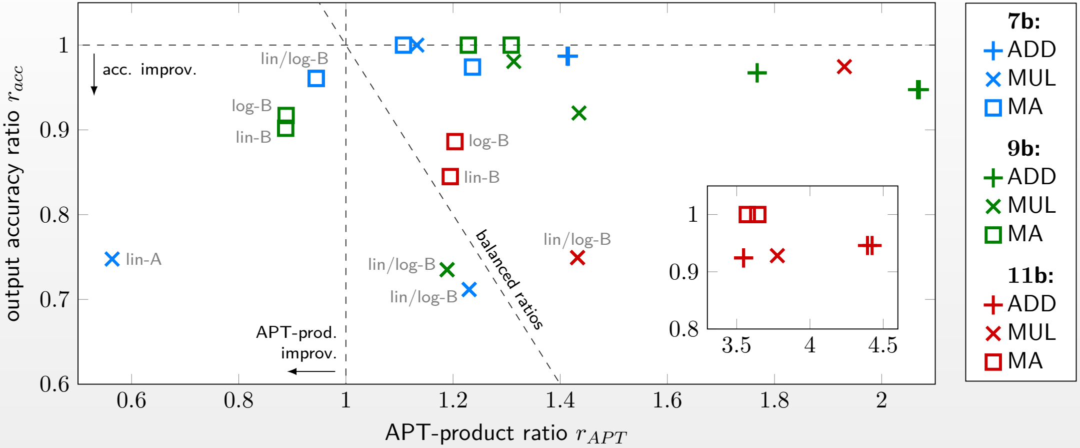
University of Bremen, Institute of Electrodynamics and Microelectronics, Department of Communication Electronics (ITEM.me)

baerthel@me.uni-bremen.de

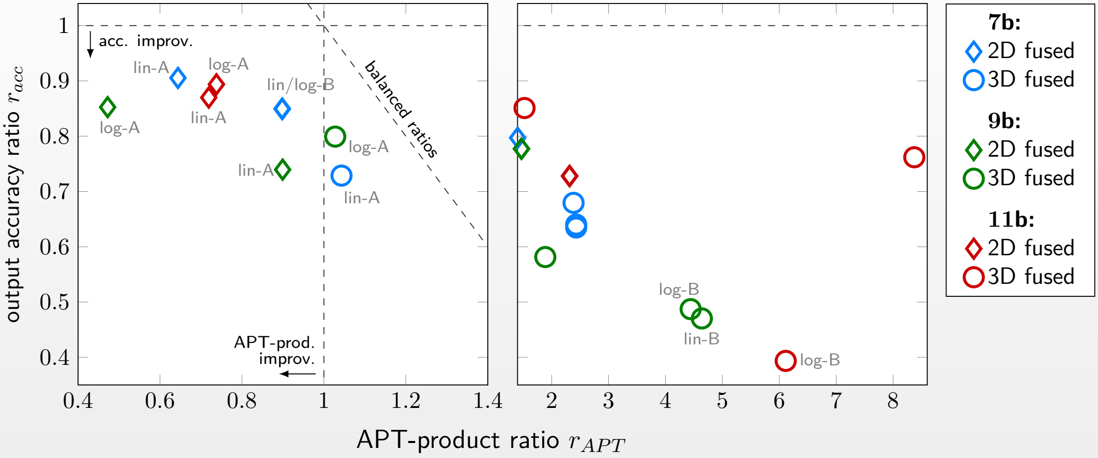
SORN Datatypes

label	config
lin-7-A	$[-\infty,-3)$ $[-3,-1.5)$ $[-1.5,0)$ 0 $(0,1.5]$ $(1.5,3]$ $(3,\infty]$
lin-7-B	$[-\infty,-100)$ $[-100,-50)$ $[-50,0)$ 0 $(0,50]$ $(50,100]$ $(100,\infty]$
log-7-A	$[-\infty,-1)$ $[-1,-0.5)$ $[-0.5,0)$ 0 $(0,0.5]$ $(0.5,1]$ $(1,\infty]$
log-7-B	$[-\infty,-64)$ $[-64,-32)$ $[-32,0)$ 0 $(0,32]$ $(32,64]$ $(64,\infty]$
lin-9-A	$[-\infty,-1)$ $\ldots$ $[-1/3,0)$ 0 $(0,1/3]$ $(1/3,2/3]$ $(2/3,1]$ $(1,\infty]$
lin-9-B	$[-\infty,-120)$ $\ldots$ $[-40,0)$ 0 $(0,40]$ $(40,80]$ $(80,120]$ $(120,\infty]$
log-9-A	$[-\infty,-2)$ $\ldots$ $[-0.5,0)$ 0 $(0,0.5]$ $(0.5,1]$ $(1,2]$ $(2,\infty]$
log-9-B	$[-\infty,-128)$ $\ldots$ $[-32,0)$ 0 $(0,32]$ $(32,64]$ $(64,128]$ $(128,\infty]$
lin-11-A	$[-\infty,-1)$ $\ldots$ $[-0.25,0)$ 0 $(0,0.25]$ $(0.25,0.5]$ $(0.5,0.75]$ $(0.75,1]$ $(1,\infty]$
lin-11-B	$[-\infty,-200)$ $\ldots$ $[-50,0)$ 0 $(0,50]$ $(50,100]$ $(100,150]$ $(150,200]$ $(200,\infty]$
log-11-A	$[-\infty,-2)$ $\ldots$ $[-0.25,0)$ 0 $(0,0.25]$ $(0.25,0.5]$ $(0.5,1]$ $(1,2]$ $(2,\infty]$
log-11-B	$[-\infty,-256)$ $\ldots$ $[-32,0)$ 0 $(0,32]$ $(32,64]$ $(64,128]$ $(128,256]$ $(256,\infty]$

Evaluation with Datatypes (1)



Evaluation with Datatypes (2)



Edge Detection with SORN Arithmetic [1]

Edge Detection

Figure: Highway grayscale image

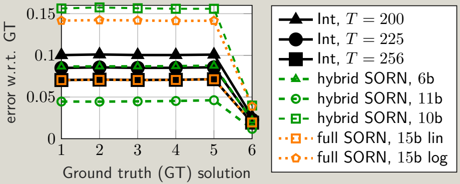
Figure: Edge Detection result

Sobel Operator on grayscale image A

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \mathbf{A}_{3\times 3} \qquad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}_{3\times 3}$$

$$G = \sqrt{{G_x}^2 + {G_y}^2} \qquad \Rightarrow \text{Edge if } G > T$$

Evaluation on BSDS500



Hardware Performance

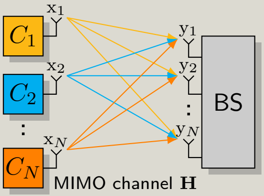
- Lower Area/Power/Runtime for all SORN designs
- Up to 47% reduction in Area utilization for Hybrid SORN Design
- Up to 80% reduction in Power consumption for Full SORN Design

Reference

[1] Moritz Bärthel, Nils Hülsmeier, Jochen Rust, Steffen Paul, "On the Implementation of Edge Detection Algorithms with SORN Arithmetic", Proceedings of the Conference for Next Generation Arithmetic (CoNGA) 2022, Singapore, March 2022.

Sphere Decoding with SORN Arithmetic [2]

MIMO Scenario



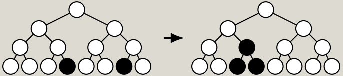
Maximum-Likelihood-Estimation

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

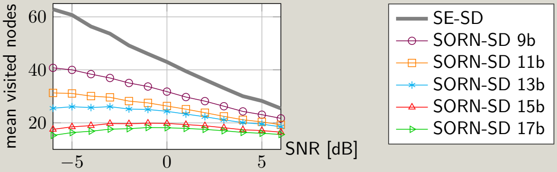
$$\Leftrightarrow \hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{S}^N}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2$$

SORN Preprocessing

$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$ as exhaustive search using SORNs



Reduction of Visited Nodes



Reference

[2] M. Bärthel, S. Knobbe, J. Rust and S. Paul, "Hardware Implementation of a Latency-Reduced Sphere Decoder With SORN Preprocessing," in IEEE Access, vol. 9, pp. 91387-91401, 2021, doi: 10.1109/ACCESS.2021.3091778