Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

In-Stream Stochastic Division and Square Root via Correlation

Di Wu and Joshua San Miguel
di.wu@ece.wisc.edu, jsanmiguel@wisc.edu

Executive Summary

- What is Stochastic Computing (SC)?
  - An ultra-low power paradigm
- What is our proposal?
  - Leveraging SC correlation for efficient nonlinear functions
  - Division and Square Root
- Design and evaluate our proposal.
  - Simultaneously higher efficiency, accuracy and speed

Outline

- ❑ **Stochastic Computing**
- ❑ Nonlinear Functions
- ❑ Proposed Stochastic Division
- ❑ Proposed Stochastic Square Root
- ❑ Performance Evaluation
- ❑ Conclusion

Data Representation in Stochastic Computing

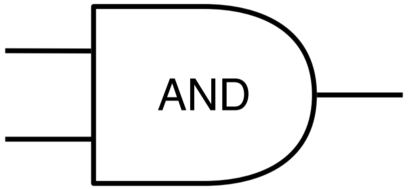
- Bernoulli Sequence (referred to as a Bit Stream)
  - The value is equal to the ratio of 1s.
  - The value is irrelevant to the position of 0s and 1s.
- Data Range
  - 0 ~ 1

A=0.5	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0
B=0.5	0	0	1	0	0	1	1	1	0	1	0	1	0	1	0	1
C=0.75	1	0	1	1	0	1	1	1	1	1	0	1	0	1	1	1

Example of Stochastic Computing Circuit

Multiplication via a single AND gate

1010010010100100 (6/16)
0101010110010110 (8/16)



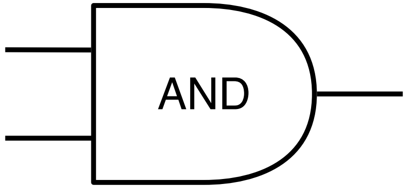
0000010010000100 (3/16)

- ➢ Extremely Simple Logic
- ➢ Ultra-Low Power
- ➢ High Resistance to Noise

Correlation in Stochastic Computing

- Zero correlation leads to Multiplication

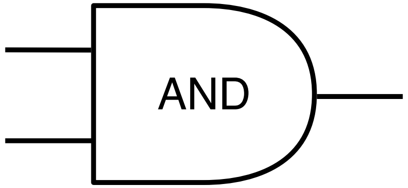
1010010010100100 (6/16)
0101010110010110 (8/16)



0000010010000100 (3/16)

- Positive correlation leads to Minimum

1010010010100100 (6/16)
Reorder 1010010010110101 (8/16)



1010010010100100 (6/16)

Number of paired 1s differs!!!

Correlation

Leveraging Correlation for SC Nonlinearity

- Zero correlation is the cornerstone of primitive SC

- Positive correlation for AND gate is
  - Harmful to Multiplication
  - Beneficial to Minimum

- Our goal is
  - To leverage SC correlation
  - For Division and Square Root
  - Achieving higher efficiency

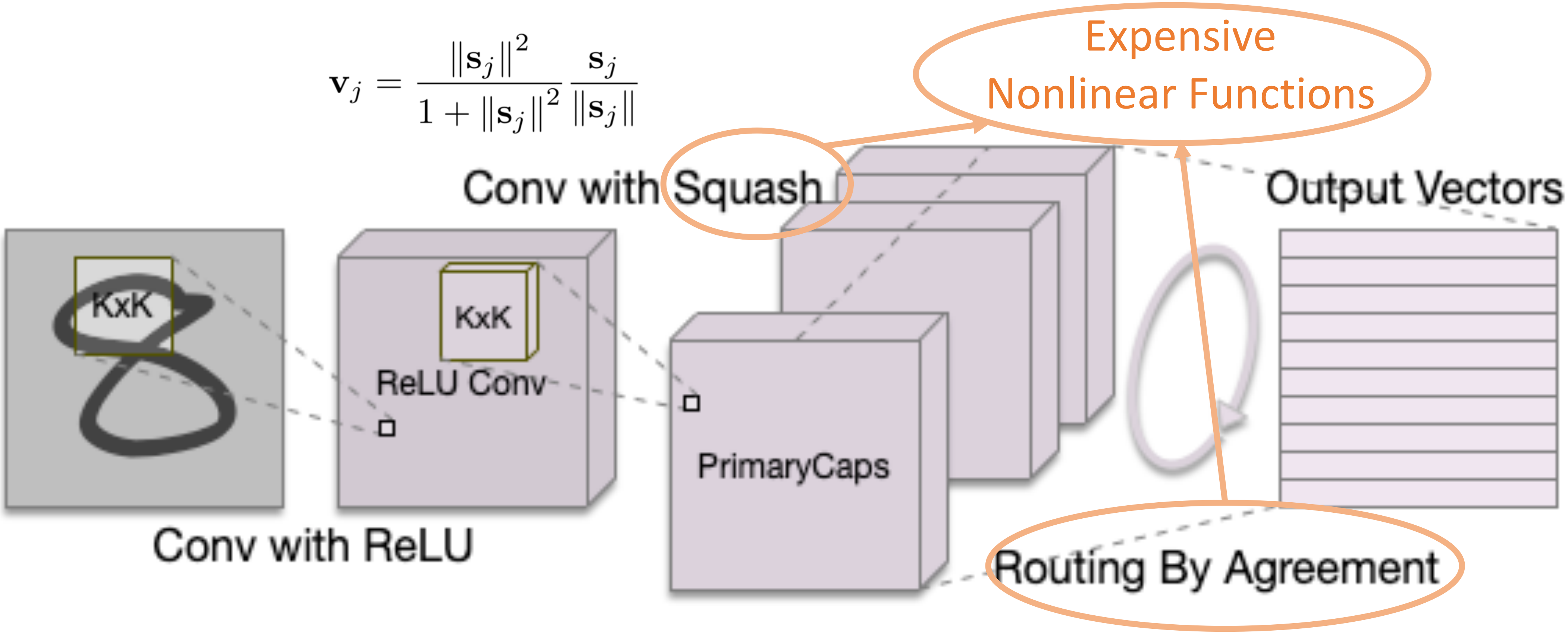
Outline

- ✓ Stochastic Computing
- ❑ **Nonlinear Functions**
- ❑ Proposed Stochastic Division
- ❑ Proposed Stochastic Square Root
- ❑ Performance Evaluation
- ❑ Conclusion

Nonlinear Functions

'Dynamic Routing via Capsules' by Hinton (*NeurIPS'17*)

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$$



$$softmax(x_j) = \frac{exp(x_j)}{\sum exp(x_k)}$$

Nonlinear Functions

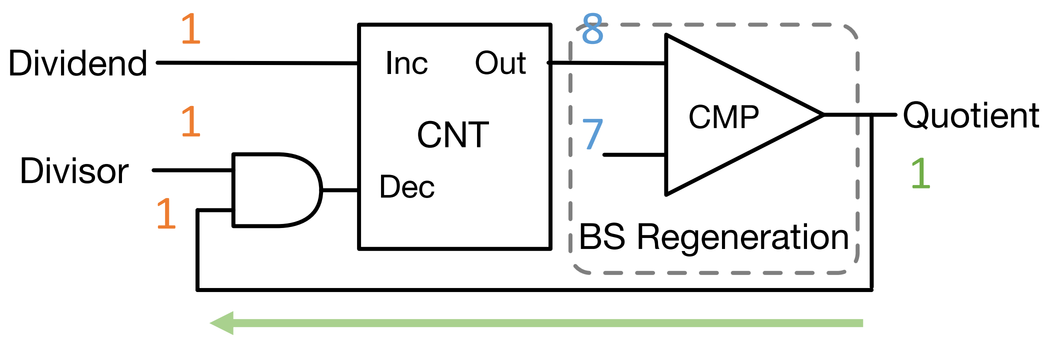
- Statistics from emerging neural networks

NN	Operation
CNN	Conv, FC, Pooling, ReLu
LSTM	*, Div, Exp, Tanh
Graph CNN	*, Div, Exp, Log, Sqrt
CapsNet	*, Div, Exp, Log, Sqrt

- Stochastic computing for better hardware efficiency

Existing Stochastic Divisions

- Gaines Division (GDIV) (*Gaines, 1969*)

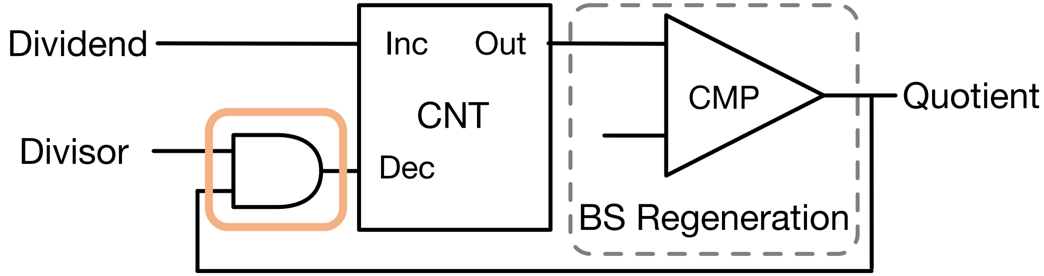


System reaches equilibrium:

$$P_{Dividend} = P_{Quotient} \cdot P_{Divisor}$$

Existing Stochastic Divisions

- Gaines Division (GDIV) (*Gaines, 1969*)

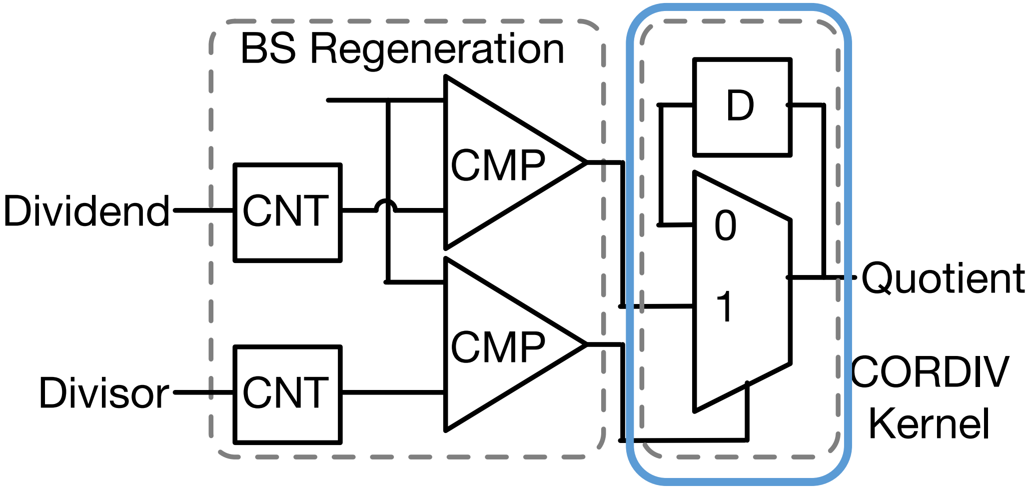


Requiring zero correlation

System reaches equilibrium:

$$P_{Dividend} = P_{Quotient} \cdot P_{Divisor}$$

- Correlated Division (CORDIV) (*Chen, IVLSI 2016*)

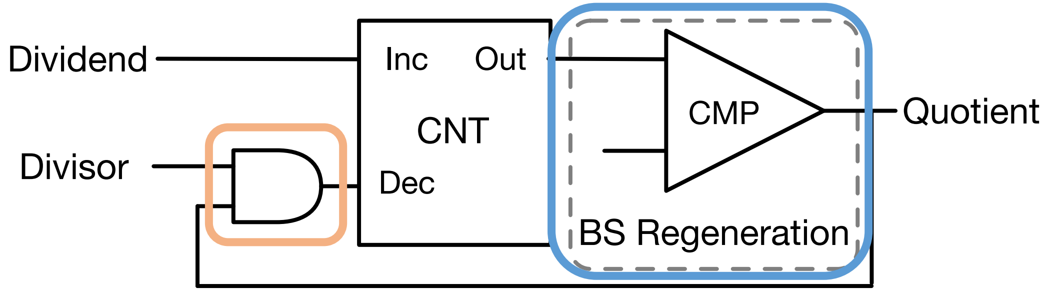


BS inputs to CORDIV Kernel are positively correlated

$$P_{Quotient} = P_{Dividend} / P_{Divisor}$$
$$= \frac{\#(1)_{Dividend}}{\#(1)_{Divisor}}$$

Existing Stochastic Divisions

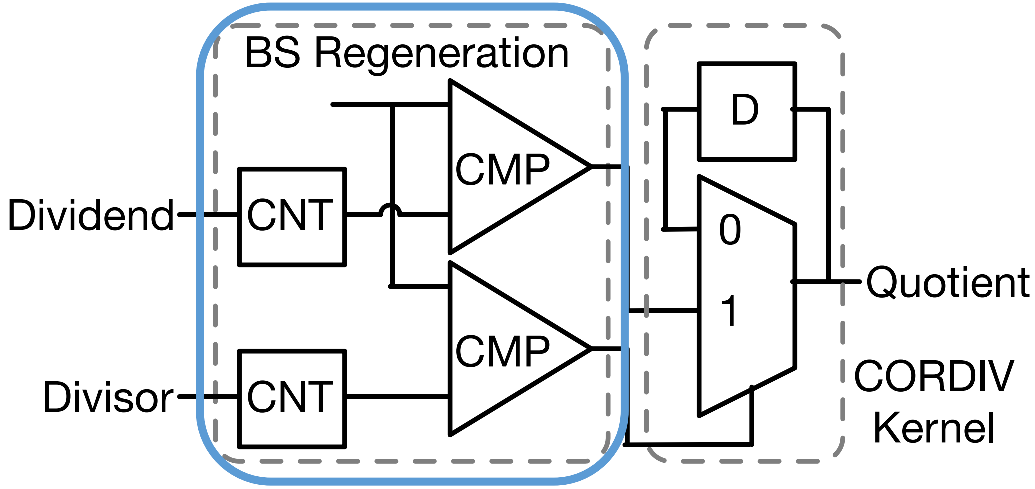
- Gaines Division (GDIV) (*Gaines, 1969*)



System reaches equilibrium:

$$P_{Dividend} = P_{Quotient} \cdot P_{Divisor}$$

- Correlated Division (CORDIV) (*Chen, IVLSI 2016*)

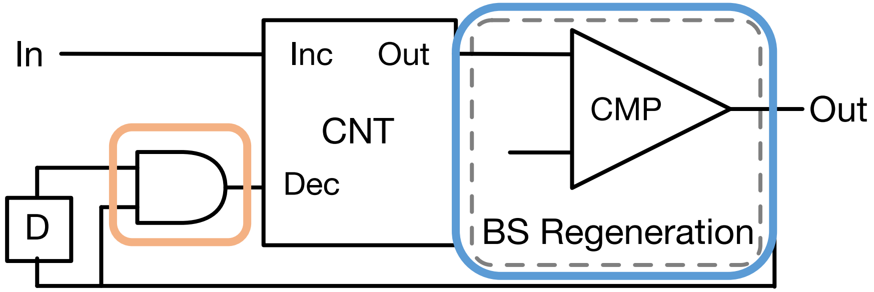


BS inputs to CORDIV Kernel are positively correlated

$$P_{Quotient} = P_{Dividend} / P_{Divisor}$$
$$= \frac{\#(1)_{Dividend}}{\#(1)_{Divisor}}$$

Existing Stochastic Square Root

- Gaines Square Root (GSQRT) (*Gaines, 1969*)



System reaches equilibrium:

$$P_{In} = P_{Out} \cdot P_{Out}$$

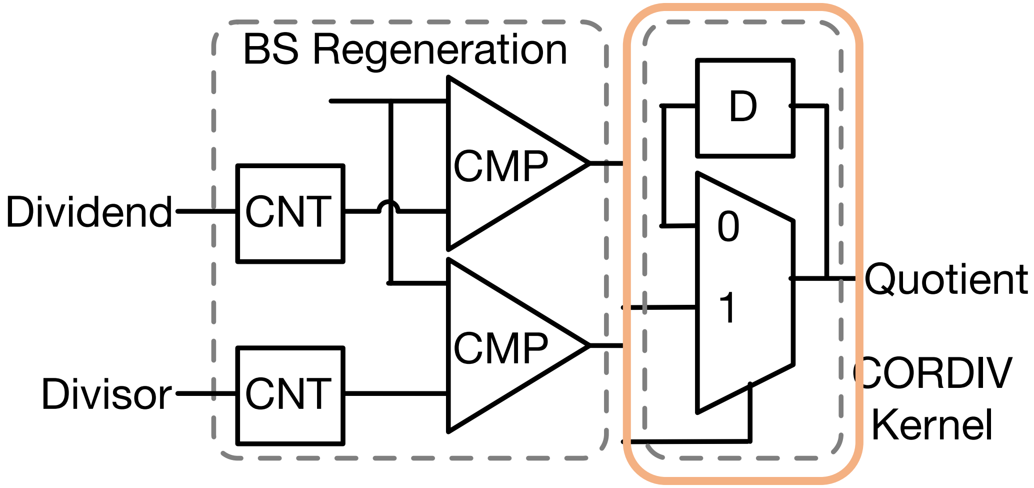
- Problems in existing stochastic Division and Square Root
  - Highly relying on near zero correlation
    - Requiring extensive manipulation on Random Number Generators
  - Expensive Bit Stream Regeneration
    - Need comparators, and even additional registers

Outline

- ✓ Stochastic Computing
- ✓ Nonlinear Functions
- ❑ **Proposed Stochastic Division**
- ❑ Proposed Stochastic Square Root
- ❑ Performance Evaluation
- ❑ Conclusion

In-Stream Correlation Based Division (ISCBDIV)

Leveraging CORDIV Kernel, which requires positive correlation



BS inputs to CORDIV Kernel are positively correlated

$$P_{Quotient} = P_{Dividend}/P_{Divisor}$$
$$= \frac{\#(1)_{Dividend}}{\#(1)_{Divisor}}$$

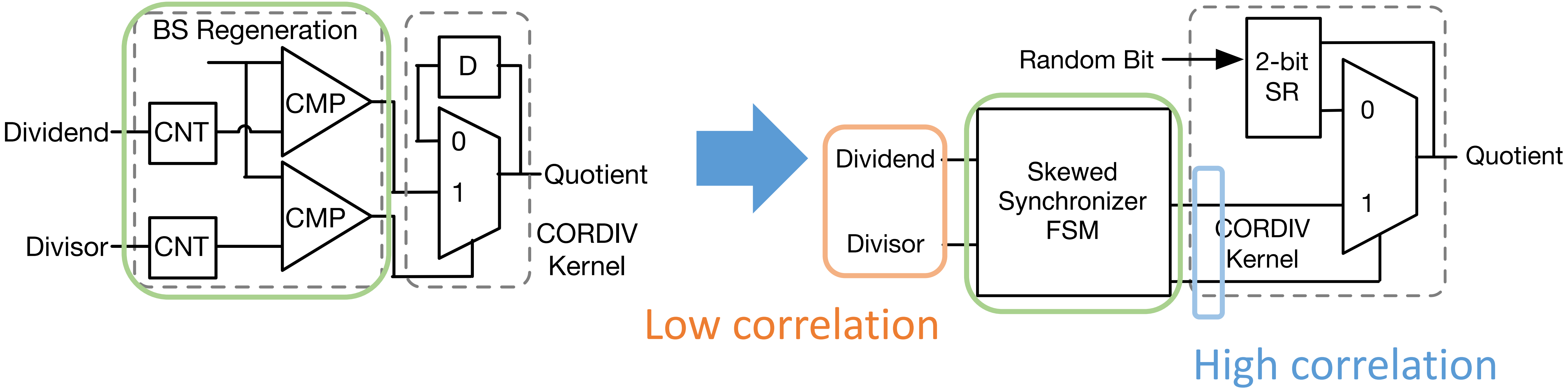
Pair up 1s in the input BSs

In-Stream Correlation Based Division (ISCBDIV)

Leveraging CORDIV Kernel, which requires positive correlation

- ➢ Maximize correlation via Skewed Synchronizer

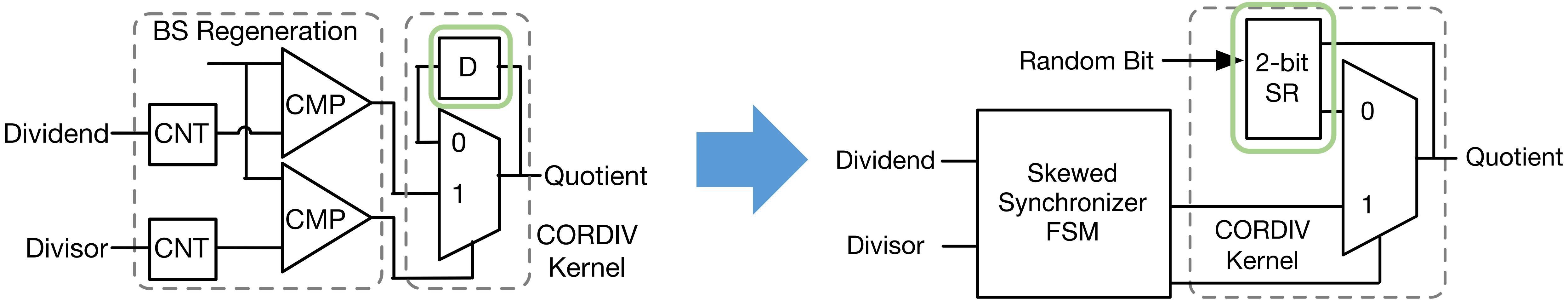
Based on the fact:
SC dividend cannot be larger than the divisor.



In-Stream Correlation Based Division (ISCBDIV)

Leveraging CORDIV Kernel, which requires positive correlation

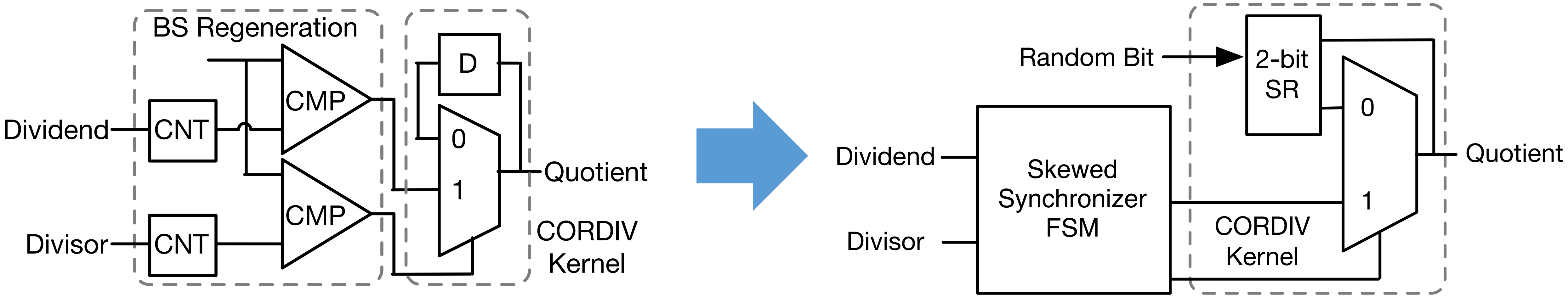
- Maximize correlation via Skewed Synchronizer
- Extend original DFF to 2-bit Shift Register (SR)



In-Stream Correlation Based Division (ISCBDIV)

Leveraging CORDIV Kernel, which requires positive correlation

- Maximize correlation via Skewed Synchronizer
- Extend original DFF to 2-bit Shift Register (SR)



Given 256 bit length BS, number of registers is reduced from 17 to 4

Outline

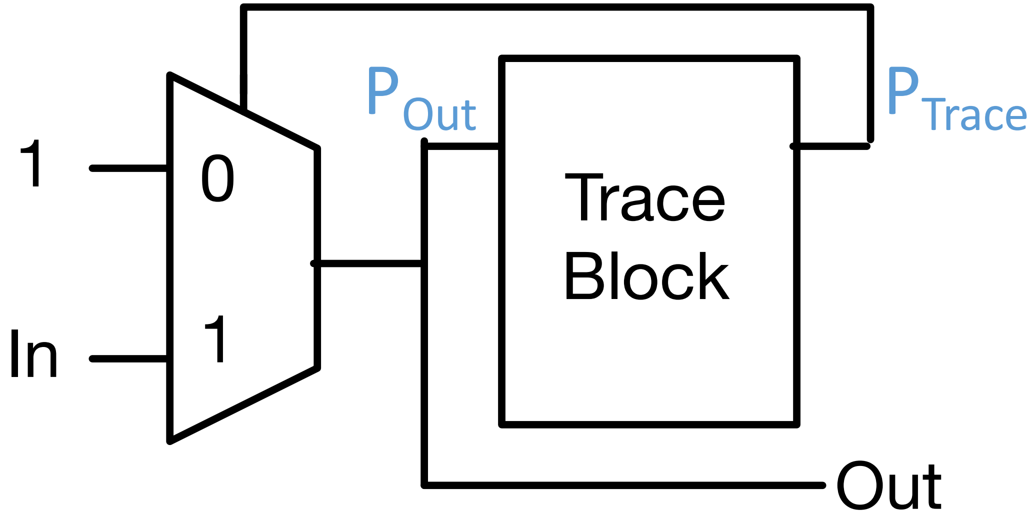
- ✓ Stochastic Computing
- ✓ Nonlinear Functions
- ✓ Proposed Stochastic Division
- ❑ **Proposed Stochastic Square Root**
- ❑ Performance Evaluation
- ❑ Conclusion

Bit Inserting Square Root (BISQRT)

New Computing Scheme

$\sqrt{0.81} = 0.9 > 0.81$

- Output of Stochastic Square Root is always larger than input
- Inserting 1s into input properly leads to correct output



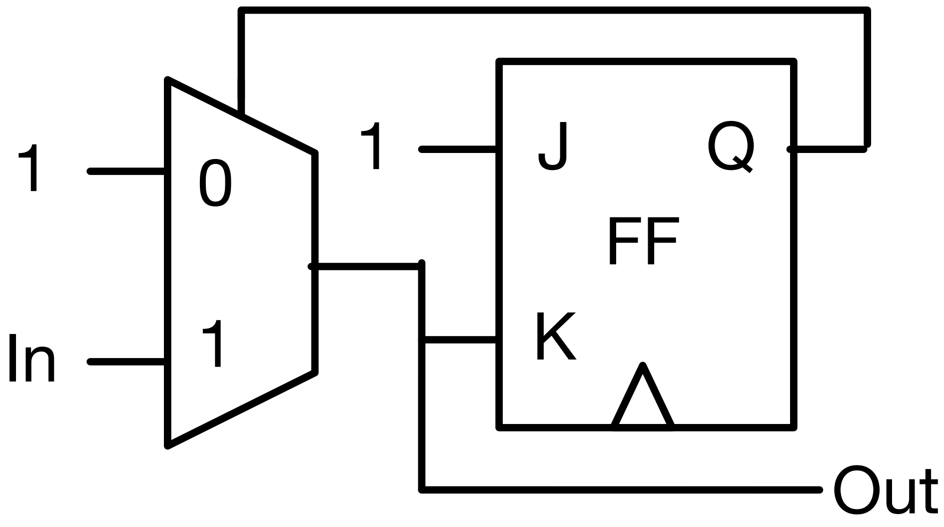
Functionality is guaranteed if

$$P_{Trace} = 1/(1 + P_{Out})$$

Bit Inserting Square Root (BISQRT)

Two approaches to build the Trace Block

- ➢ JKDIV (*Gaines, 1969*) based BISQRT



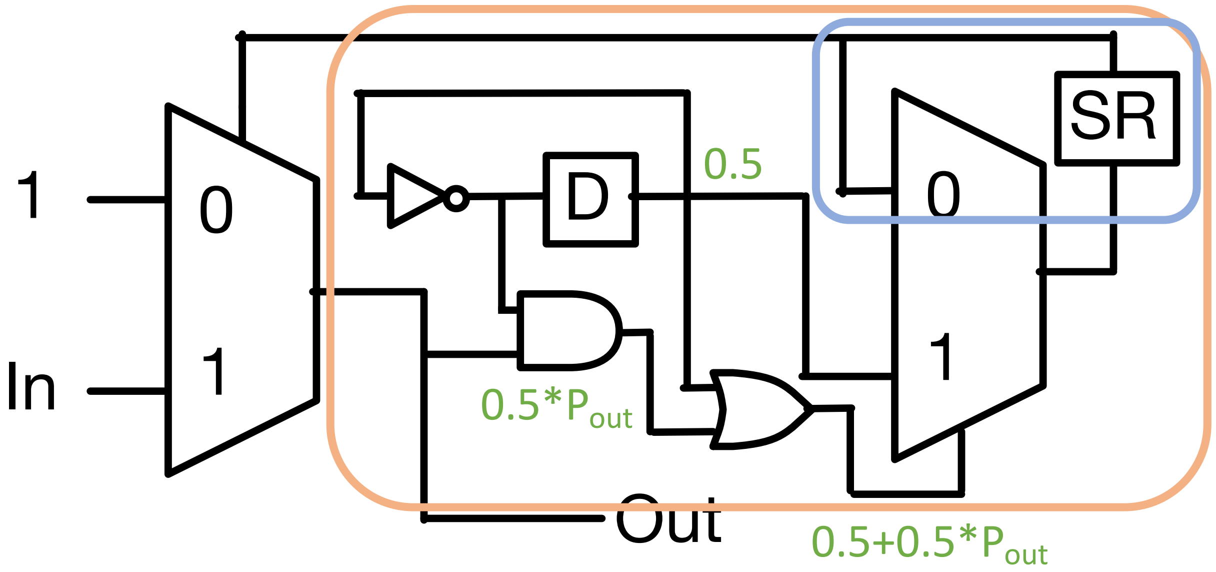
Connecting port J to 1 leads to

$$P_Q = P_J/(P_J + P_K) = \frac{1}{1 + P_K}$$

Bit Inserting Square Root (BISQRT)

Two approaches to build the Trace Block

- ➢ JKDIV (*Gaines, 1969*) based BISQRT
- ➢ ISCBDIV based BISQRT



Simplified ISCBDIV

Output of CORDIV Kernel comes from the SR for isolation

$$P_{Quotient} = 0.5/(0.5 + 0.5 \cdot P_{Out}) = \frac{1}{1 + P_{Out}}$$

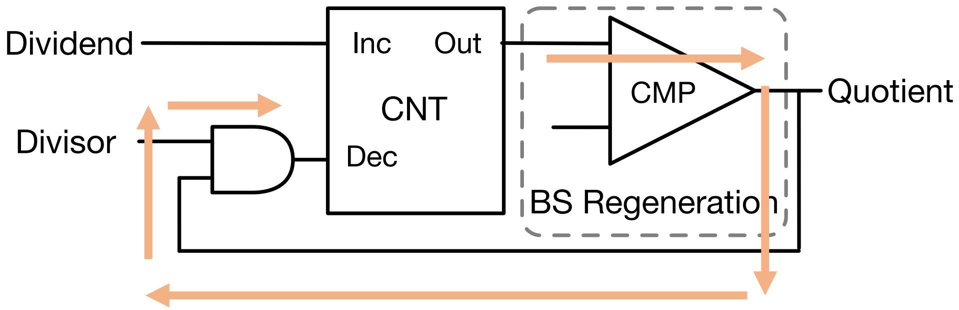
Outline

- ✓ Stochastic Computing
- ✓ Nonlinear Functions
- ✓ Proposed Stochastic Division
- ✓ Proposed Stochastic Square Root
- ❑ **Performance Evaluation**
- ❑ Conclusion

Evaluating Metrics

Feedback loops with Counters

- Counters incur fluctuation in output Bit Stream
- Before reaching equilibrium, the BS error rate is inaccurate



- Convergence time
  - Cycle count to achieve expected stable-state accuracy

Experimental Configuration

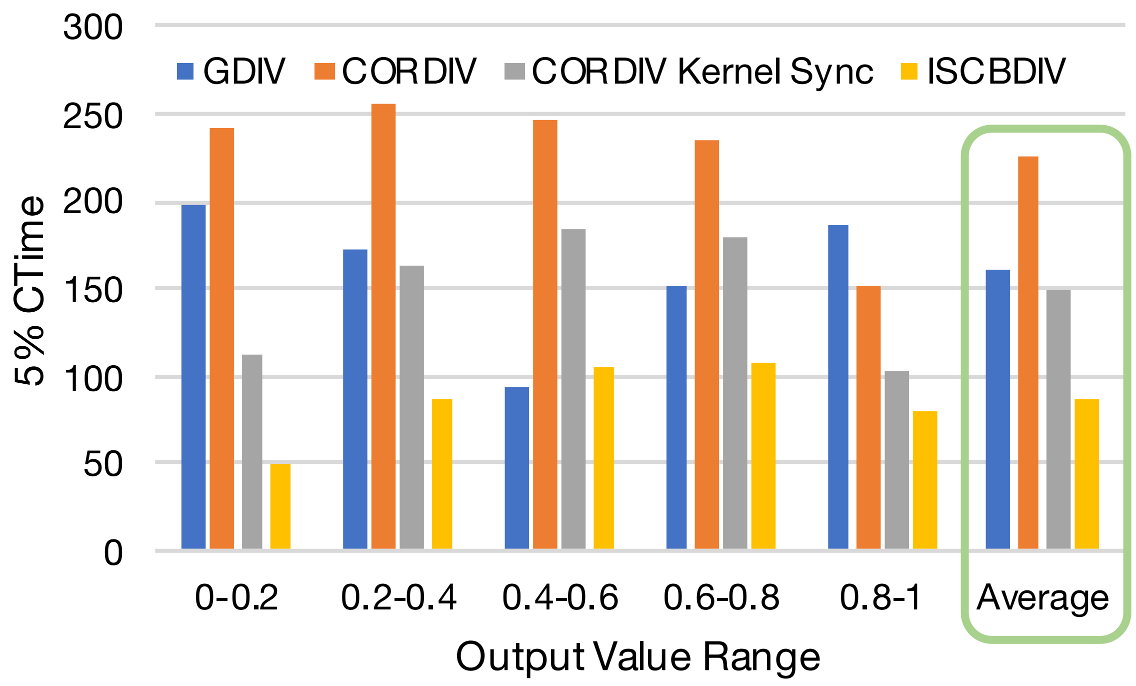
Focus on influence of input correlation to the results

- Vary output value range
  - Different output value ranges imply different input correlation
- Fix input length
- Apply high quality Sobol RNGs (*Li, DATE’17*)

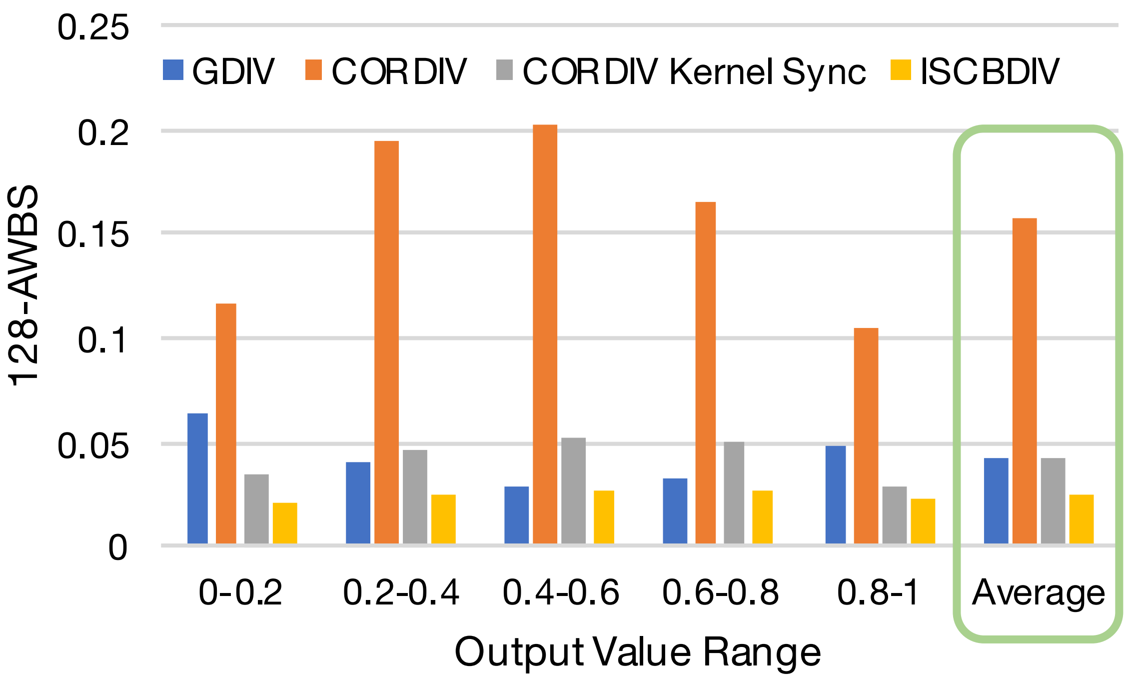
ISCBDIV Performance

- Converges 46.3% faster.
- Reduces error by 43.3%.

Convergence time (Cycle)



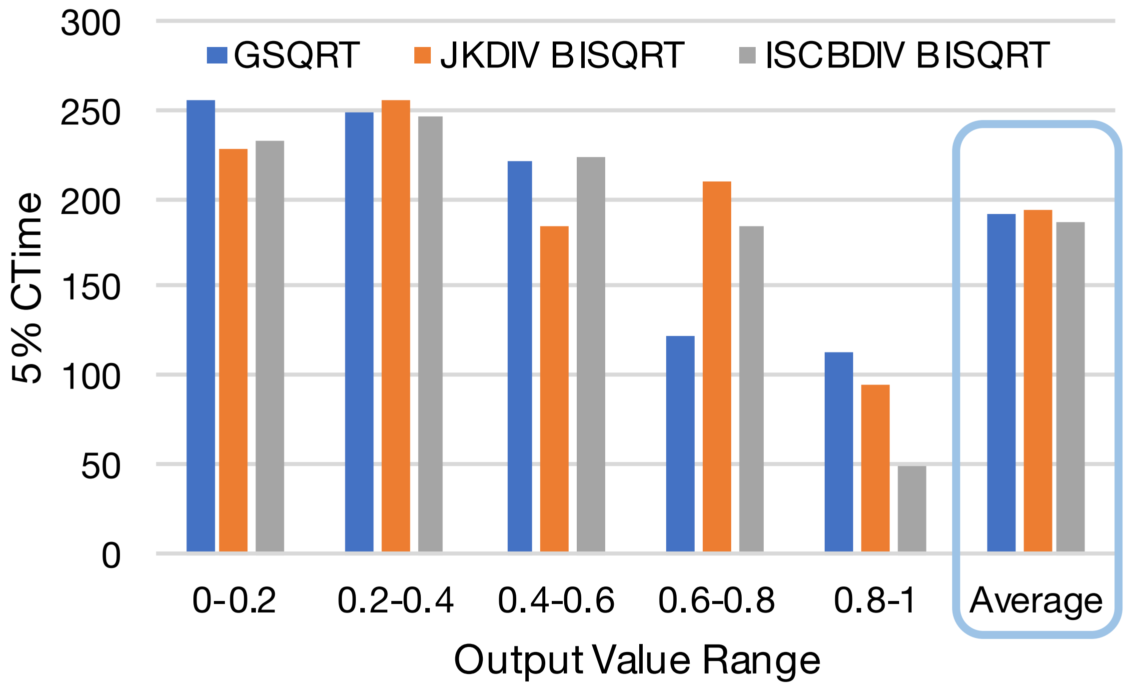
Stable error rate (100%)



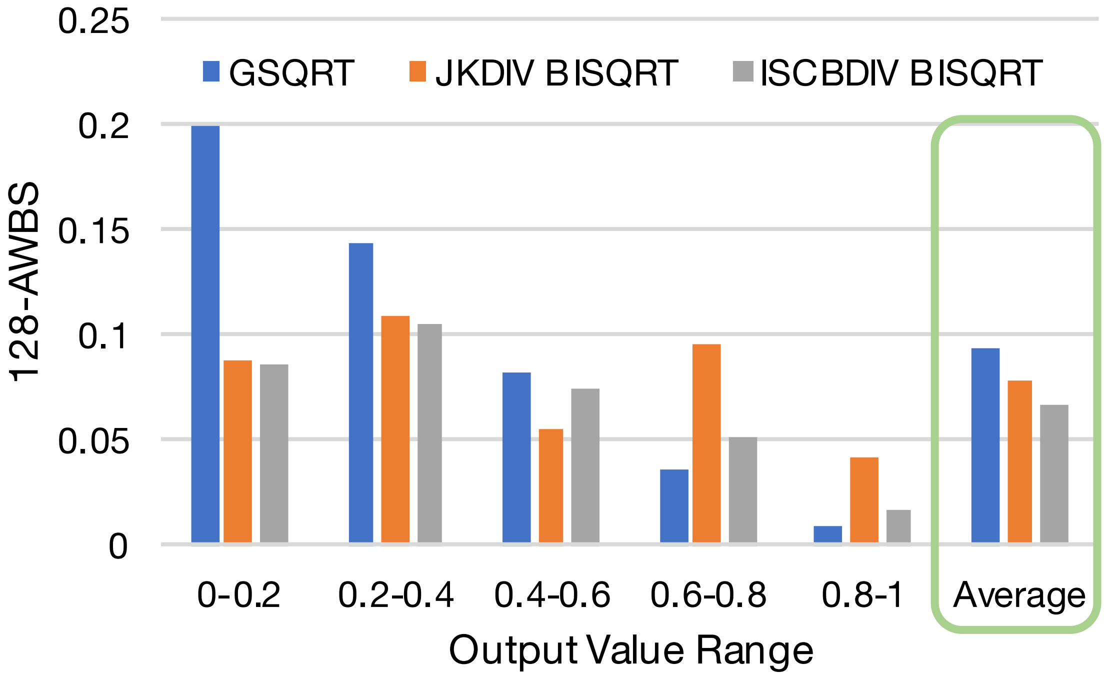
BISQRT Performance

- JKDIV-BISQRT reduces error by 16.8%.
- ISCBDIV-BISQRT reduces error by 29.0%.

Convergence time (Cycle)



Stable error rate (100%)



Hardware Implementation

Synopsys Design Compiler with TSMC 45nm @ 400MHz

- Area, Power, Throughput Per Area (TPA)

$$frequency/(latency \times area)$$

For DIV, energy reduction is 67.6%.

Design	*Area* (μm^2)	*Power* (μW)	*Latency* $(cycles)$	*TPA* $(1/(\mu m^2 \cdot s))$
GDIV(Depth-5)	74.3	21.0	158	34,073
CORDIV	211.2	60.9	226	8,384
Proposed ISCBDIV	**40.4**	**12.5**	**86**	**115,128**
GSQRT(Depth-5)	78.3	23.5	192	26,607
Proposed JKDIV BISQRT	**11.3**	**6.3**	195	**181,529**
Proposed ISCBDIV BISQRT	25.4	12.6	**187**	84,214

For JKDIV SQRT, energy reduction is 72.8%.

Outline

- ✓ Stochastic Computing
- ✓ Nonlinear Functions
- ✓ Proposed Stochastic Division
- ✓ Proposed Stochastic Square Root
- ✓ Performance Evaluation
- ❑ **Conclusion**

Conclusion

Our proposal outperforms state-of-the-art nonlinear functions

- General benefits
  - SC is inherently more hardware efficient than conventional binary computing
- Enhancement to the state-of-the-art in SC
  - Lower area/power
  - Higher accuracy
  - Faster convergence speed

Department of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

Thank you
Q & A

Di Wu and Joshua San Miguel
di.wu@ece.wisc.edu, jsanmiguel@wisc.edu