Deterministic Methods for Stochastic Computing using Low-Discrepancy Sequences

M. Hassan Najafi, David Lilja, and Marc Riedel

najafi@louisiana.edu

ICCAD 2018, San Diego, CA

UNIVERSITY of LOUISIANA LAFAYETTE

UNIVERSITY OF MINNESOTA
Driven to Discover®

Overview

- **Introduction to SC**
  - Advantages, weaknesses
  - Stochastic bitstream generation
- **Deterministic Approaches to SC**
  - Relatively prime length, clock division, Rotation
  - From unary-streams to pseudo-random streams
- **Low-discrepancy sequences**
  - Sobol sequences
- **Proposed LD deterministic methods**
  - Method 1, Method 2, Proposed structures
- **Evaluation**
  - Accuracy evaluation, Scalability evaluation
- **Conclusion**

- **Stochastic computing (SC)**
  - An **approximate** computing approach for many years
  - Logical computation on **random** bit-streams
  - All digits have the same weight, numbers limited to the [0, 1]
  - Value: probability of obtaining a one versus a zero

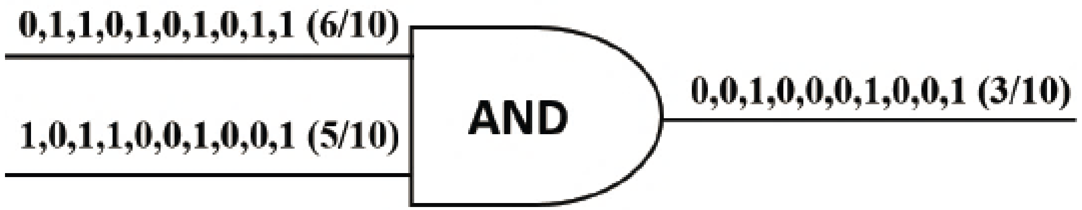
e.g. 101010, 1011011100 -> 0.6

  - **Advantages**
    - Noise tolerance e.g., 0010000011000000 3/16 -> 4/16
    - Low hardware cost e.g., multiplication using an AND gate
    - Skew tolerance [Najafi et al, TC’17]
    - Progressive precision [Alaghi et al, DAC’13]
  - **Weaknesses**
    - Random fluctuation: inaccuracy of computation
    - Long processing time -> high energy consumption

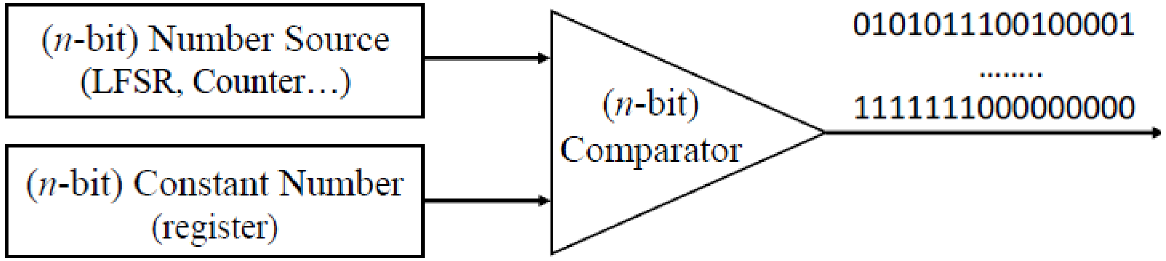
Introduction

- **Stochastic computing (SC)**
  - Common operations such as multiplication (using AND gate) or Scaled addition (using multiplexer) require **independent inputs**
    - Conventionally independence is provided by randomness

Stochastic **multiplication** using **AND** gate on random input bitstreams



  - Converting input data in binary domain into random bitstreams using random or pseudo-random constructs: e.g. LFSR



Introduction

- **Deterministic approaches to SC**
  - Recent progress in SC has revolutionized the paradigm
    [Najafi et al. TVLSI’17] [Jenson and Riedel ICCAD’16]
  - If properly structured, random fluctuation can be removed
    - Producing **deterministic and completely accurate** results
    - Improving **the processing time and hardware cost**
      - compared to conventional random-based stochastic for high accuracy
  - Logical computation is performed on **unary bit-streams**
    1111110000 -> 0.6
  - **Independence** between the input unary streams is provided by **three approaches:**
    - **1) Rel. prime stream length** **2) clock division** **3) rotation**

Deterministic Approaches to SC

[Jenson and Riedel, ICCAD'16]

- **Example. Rel. prime length method with unary bit-streams**

$$a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3$$
$$b_0\, b_1\, b_2\; b_0\, b_1\, b_2\; b_0\, b_1\, b_2\; b_0\, b_1\, b_2$$

$$\begin{aligned} 1/3 &= 100100100100 \\ 3/4 &= 111011101110 \\ \hline 3/12 &= 100000100100 \end{aligned}$$

- **Example. Clock division method with unary bit-streams**

$$a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3$$
$$b_0\, b_0\, b_0\, b_0\; b_1\, b_1\, b_1\, b_1\; b_2\, b_2\, b_2\, b_2\; b_3\, b_3\, b_3\, b_3$$

$$\begin{aligned} 1/4 &= 1000\ 1000\ 1000\ 1000 \\ 3/4 &= 1111\ 1111\ 1111\ 0000 \\ \hline 3/16 &= 1000\ 1000\ 1000\ 0000 \end{aligned}$$

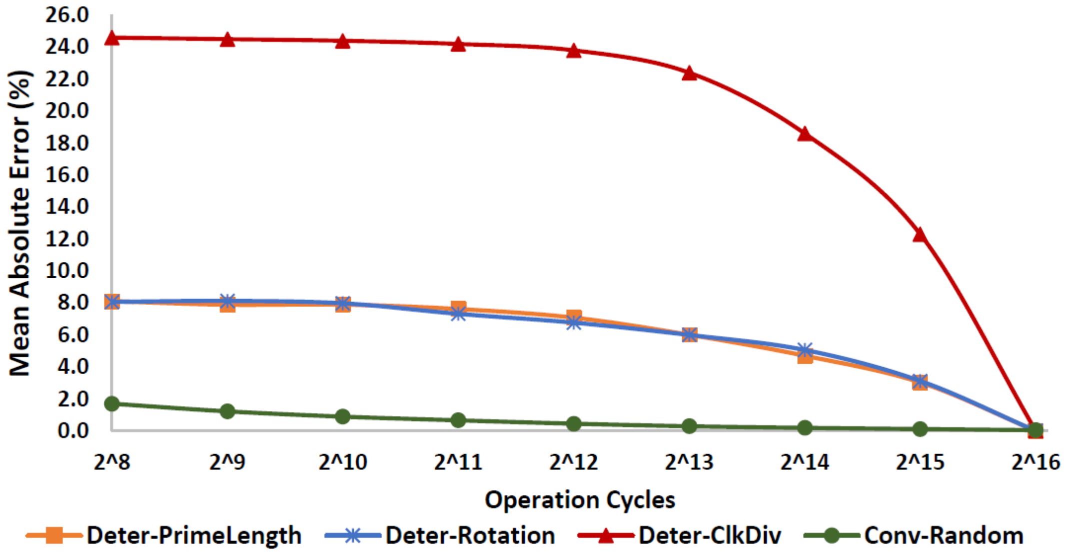
- **Example. Rotation method with unary bit-streams**

$$a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3\; a_0\, a_1\, a_2\, a_3$$
$$b_0\, b_1\, b_2\, b_3\; b_3\, b_0\, b_1\, b_2\; b_2\, b_3\, b_0\, b_1\; b_1\, b_2\, b_3\, b_0$$

$$\begin{aligned} 1/4 &= 1000\ 1000\ 1000\ 1000 \\ 3/4 &= 1110\ 0111\ 1011\ 1101 \\ \hline 3/16 &= 1000\ 0000\ 1000\ 1000 \end{aligned}$$

Deterministic Approaches to SC

- Important challenge with unary stream-based deterministic approaches
  - **Poor progressive precision**
    - **Running the operation for fewer cycles leads to a poor result**



MAE of multiplying two 8-bit precision input values

2^16 cycles:
completely accurate with deter.

2^15 cycles:
a MAE of **3.12%** for deter. rotation
a MAE of **7.98%** for deter. clk div.
a MAE of **0.11%** for prior random

2^10 cycles:
a MAE of **12.3%** for deter. rotation
a MAE of **24.4%** for deter. clk div.
a MAE of **0.89%** for prior random

Much longer processing time than random SC when **slightly inaccuracy is acceptable**

- **Energy in-efficient for many applications**

Deterministic Approaches to SC

- **Essential property of three prior deterministic methods**
  - Every bit of one bitstream pairs with every bit of the other **exactly once**

- **This property applies regardless of the distribution of the 1's and 0's in the bit streams**
  - The bit streams can in fact be randomized **[Najafi and Lilja, ICCD'17]**

- **Maximal period pseudo-random sources** can be used to generate the bit-streams accurately
  - The **period** should be equal to the **length** of bit-stream

Unary bit-stream: 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 : 8/16

Pseudo-randomized bit-stream: 1 0 0 0 1 1 0 1 0 0 0 1 0 1 1 1 : 8/16

Deterministic Approaches to SC

- Example. **Rel. prime length method** with pseudo-randomized bit-streams

$$\begin{array}{cccccccccccc} a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 \\ b_1 & b_0 & b_3 & b_1 & b_0 & b_3 & b_1 & b_0 & b_3 & b_1 & b_0 & b_3 \end{array}$$

$$\begin{array}{rl} 1/3 & = 100100100100 \\ 3/4 & = 101110111011 \\ \hline 3/12 & = 100100100000 \end{array}$$

- Example. **Clock division method** with pseudo-randomized bit-streams

$$\begin{array}{cccccccccccccccc} a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 \\ b_1 & b_1 & b_1 & b_1 & b_0 & b_0 & b_0 & b_0 & b_3 & b_3 & b_3 & b_3 & b_2 & b_2 & b_2 & b_2 \end{array}$$

$$\begin{array}{rl} 1/4 & = 0010\ 0010\ 0010\ 0010 \\ 3/4 & = 1111\ 0000\ 1111\ 1111 \\ \hline 3/16 & = 0010\ 0000\ 0010\ 0010 \end{array}$$

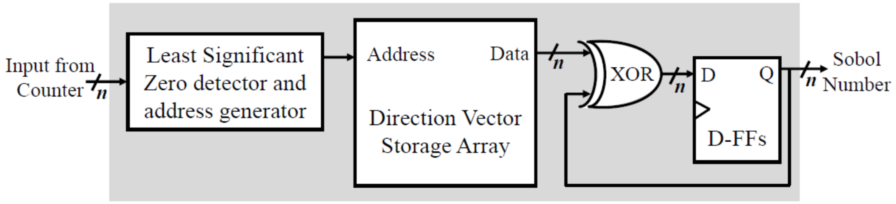
- Example. **Rotation method** with pseudo-randomized bit-streams

$$\begin{array}{cccccccccccccccc} a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 & a_0 & a_3 & a_1 & a_2 \\ b_1 & b_0 & b_3 & b_2 & b_2 & b_1 & b_0 & b_3 & b_3 & b_2 & b_1 & b_0 & b_0 & b_3 & b_2 & b_1 \end{array}$$

$$\begin{array}{rl} 1/4 & = 0100\ 0100\ 0100\ 0100 \\ 3/4 & = 1101\ 1110\ 0111\ 1011 \\ \hline 3/16 & = 0100\ 0100\ 0100\ 0000 \end{array}$$

Low-Discrepancy Sequences

- **Low discrepancy (LD)** sequences such as **Sobol** have been used in improving the speed of computation on stochastic bit-streams.
    - 1s and 0s in the bit-streams are **uniformly spaced**
        - So removing random fluctuations.
    - Bit-streams can **quickly converge** to the target value.
        - Acceptable results in a much shorter time



Sobol Sequence Generator

[Liu and Han, DATE'17]

- **The first 2^n numbers in any Sobol sequence can precisely present all possible n-bit precision numbers in the [0, 1] interval**
- e.g., simplest Sobol Seq:

0, 1/2, 1/4, 3/4, 1/8, 5/8, 3/8, 7/8, 1/16, 9/16, 5/16, 13/16, 3/16, 11/16, 7/16, 15/16

2-bit

3-bit

4-bit

Proposed LD Deterministic Method 1

- **Directly uses LD Sobol sequences**
  - The method is independent of prior deterministic methods (e.g., rotation, clk div)
- **Independence between the input bit-streams is guaranteed by**
  - Using different Sobol sequences in generating the bitstreams
- **The precision of the seq. generator should be i times the precision of the input data**
  - i = number of input data
- **Convert each input data to a stream of $2^{i.n}$ bits**
  - Comparing the input value to $2^{i.n}$ different Sobol numbers
- **Deterministic accurate output is ready after $2^{i.n}$ cycles**
  - The product of the length of the bit-streams

Proposed LD Deterministic Method 1

Sobol Seq 1	0	1/2	1/4	3/4	1/8	5/8	3/8	7/8	1/16	9/16	5/16	13/16	3/16	11/16	7/16	15/16
Sobol Seq 2	0	1/2	3/4	1/4	5/8	1/8	3/8	7/8	15/16	7/16	3/16	11/16	5/16	13/16	9/16	1/16

- **Example. Deterministic 2-bit precision multiplication: 1/4 x 3/4**

In converting to bitstream
a ‘1’ is generated if
Sobol number < target value

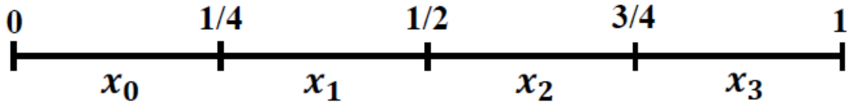
$$\begin{aligned} 1/4 &= 1000\ 1000\ 1000\ 1000 \\ 3/4 &= 1101\ 1110\ 0111\ 1011 \\ \hline 3/16 &= 1000\ 1000\ 0000\ 1000 \end{aligned}$$

- **Two important properties of the Sobol sequences:**
  - The first 2^n numbers of any Sobol sequence include all n-bit precision values in [0, 1) interval.
  - If equally split [0, 1) interval into 2^n sub-intervals, in any consecutive group of 2^n Sobol numbers starting at positions $i \times 2^n$ (i = 0,1,2, . .) there is exactly one member in each sub-interval

Proposed LD Deterministic Method 1

- We categorize consecutive groups of 2^2 numbers in the first four Sobol seq.
- Each Sobol number is **labeled** depending on its **sub-interval**

Sobol Seq 1	0	1/2	1/4	3/4	1/8	5/8	3/8	7/8	1/16	9/16	5/16	13/16	3/16	11/16	7/16	15/16
	a_0	a_2	a_1	a_3	a_0	a_2	a_1	a_3	a_0	a_2	a_1	a_3	a_0	a_2	a_1	a_3
Sobol Seq 2	0	1/2	3/4	1/4	5/8	1/8	3/8	7/8	15/16	7/16	3/16	11/16	5/16	13/16	9/16	1/16
	b_0	b_2	b_3	b_1	b_2	b_0	b_1	b_3	b_3	b_1	b_0	b_2	b_1	b_3	b_2	b_0
Sobol Seq 3	0	1/2	1/4	3/4	7/8	3/8	5/8	1/8	11/16	3/16	15/16	7/16	5/16	13/16	1/16	9/16
	c_0	c_2	c_1	c_3	c_3	c_1	c_2	c_0	c_2	c_0	c_3	c_1	c_1	c_3	c_0	c_2
Sobol Seq 4	0	1/2	3/4	1/4	7/8	3/8	1/8	5/8	7/16	15/16	11/16	3/16	9/16	1/16	5/16	13/16
	d_0	d_2	d_3	d_1	d_3	d_1	d_0	d_2	d_1	d_3	d_2	d_0	d_2	d_0	d_1	d_3



- Each group of 2^n numbers includes all labels from 0 to $2^n - 1$
- The difference is only in the **order of labels**

Proposed LD Deterministic Method 1

- The result of multiplying two bit-streams was deterministic and accurate if
  - **Every bit of one bit-stream meets every bit of the other stream exactly once**

- As shown in the figure, for any pair of two different Sobol sequences,

  every label u (u=0,1,2,3) in $x_u(x = a, b, c, d)$ meets
  every label t (t=0,1,2,3) in $y_t(y = a, b, c, d)$ exactly once.

  So, the result of multiplication by ANDing the two bitstreams is deterministic and completely accurate.

- The argument can be extended to multiplication of $\boldsymbol{i}$ $\boldsymbol{n}$**-bit** precision numbers

- The generated bitstreams can be divided into groups of $\mathbf{2^n}$ bits with different groups of a bit-stream representing same n-bit precision

Proposed LD Deterministic Method 2

- **Rotating LD Sobol sequences**
  - The method depends on prior deterministic methods
- **Independence between the input bit-streams is guaranteed by**
  - Rotating the bit-streams by stalling on powers of the stream lengths
- **The precision of the seq. generator is equal to the precision of the input data**
  - In contrast to the first method that depends on the number of inputs i
- **Convert each input data to a stream of 2^n bits and repeat**
  - Comparing the input value to 2^n different Sobol numbers
- **Deterministic accurate output is ready after $2^{i.n}$ cycles**
  - The product of the length of the bit-streams

Proposed Deter. Method 2

Sobol Seq 1	0	1/2	1/4	3/4	1/8	5/8	3/8	7/8	1/16	9/16	5/16	13/16	3/16	11/16	7/16	15/16
Sobol Seq 2	0	1/2	3/4	1/4	5/8	1/8	3/8	7/8	15/16	7/16	3/16	11/16	5/16	13/16	9/16	1/16

- **Example. Deterministic 2-bit precision multiplication: 2/4 x 3/4**

Sobol source 1 with a period of 2^2 and no rotation:

0, 1/2,1/4, 3/4 0,1/2,1/4,3/4, 0,1/2,1/4,3/4, 0,1/2,1/4,3/4

Sobol source 2 with a period of 2^2 and inhibiting after every 2^2 cycles:

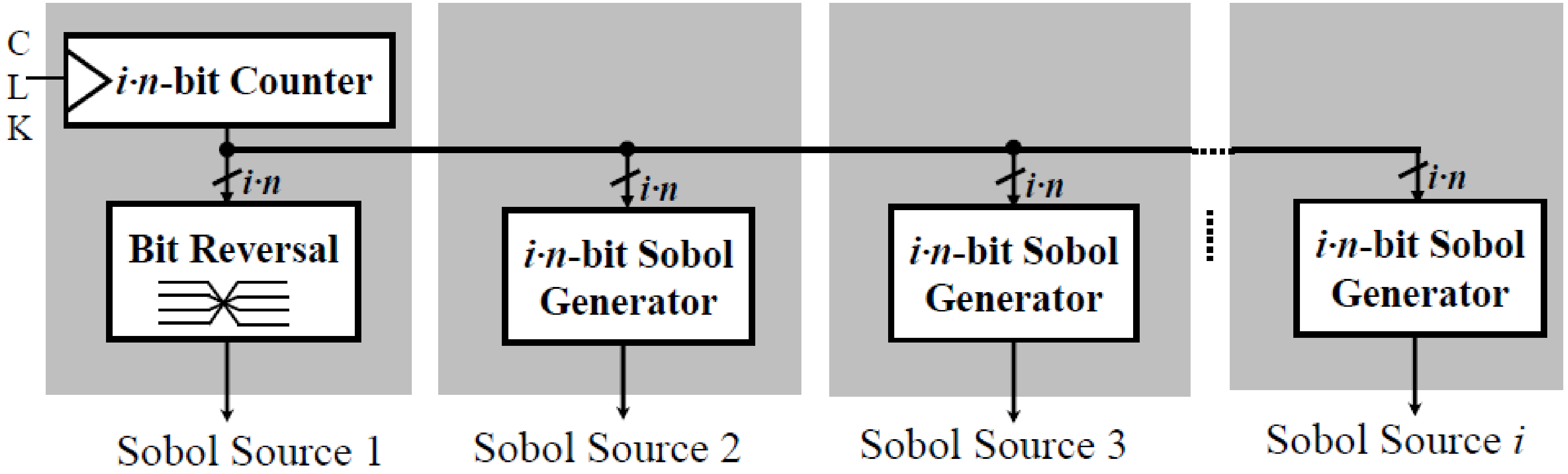
0,1/2,3/4,1/4, 1/4,0,1/2,3/4, 3/4,1/4,0,1/2 1/2,3/4,1/4,0

$$
\begin{array}{l}
2/4 = 1010\ 1010\ 1010\ 1010 \\
3/4 = 1101\ 1110\ 0111\ 1011 \\
\hline
6/16 = 1000\ 1010\ 0010\ 1010
\end{array}
$$

Proposed Structures

- **Structures** of the sources of generating Sobol sequences for the proposed methods

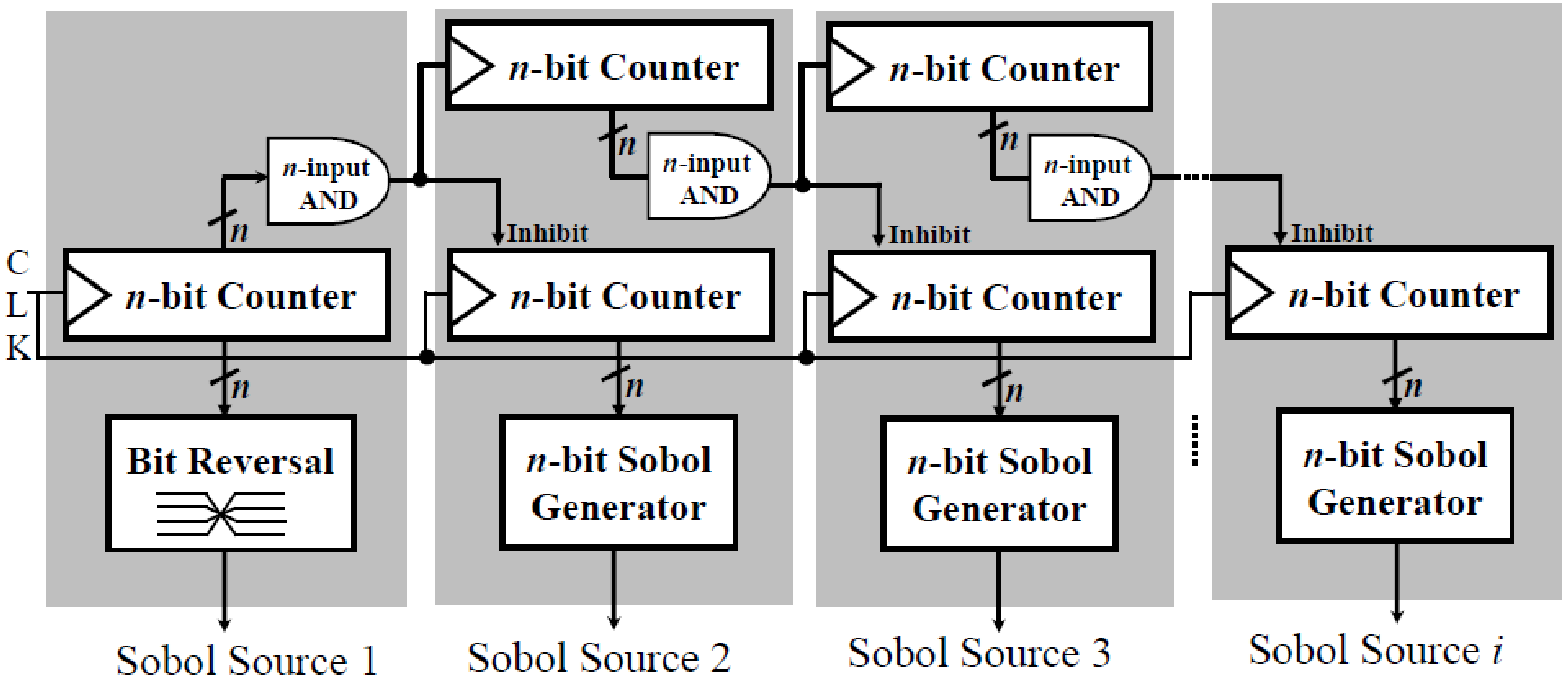
- **Method 1**



Proposed Structures

- **Structures** of the sources of generating Sobol sequences for the proposed methods

- **Method 2**



Accuracy Evaluation

- Exhaustively tested multiplication of two 8-bit precision input data in [0,1]

Mean Absolute Error (%) for different operation cycles

Design Approach	Area(μm^2)	2^{16}	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}
Conv. Approx. SC [5], [11]	781	0.05	0.15	0.26	0.39	0.58	0.79	1.20	1.67	2.32	3.32	4.72
Deter. Rotation Unary [6]	492	0.00	3.10	4.84	6.15	7.08	7.66	7.99	8.17	8.26	33.1	51.8
Deter. Rotation Pseudo-Random [9]	536	0.00	0.09	0.16	0.24	0.35	0.47	0.60	0.71	0.82	2.56	4.26
This work 1- Deter. Sobol	3361	0.00	0.0003	0.0013	0.0035	0.009	0.019	0.041	0.092	0.190	0.451	0.921
This work 2- Deter. Rotation Sobol	1277	0.00	0.0013	0.0033	0.0075	0.014	0.031	0.059	0.112	0.190	0.451	0.921

- Both proposed methods could produce **completely accurate results**
- **A significantly lower MAE** when truncating the streams
  - E.g., When running for 2^{15}cycles, a MAE
    - 100X lower than the MAE of the deter. pseudo-random rotation method
    - 3000X lower than the MAE of the deter. unary rotation approach

Scalability Evaluation

- An important challenge
  - **Limited Scalability**
    - **Hardware cost** significantly increases with the number of inputs

Hardware Area Cost (μm^2)of the Bitstream Generators (N=Input data precision, I=Number of inputs)

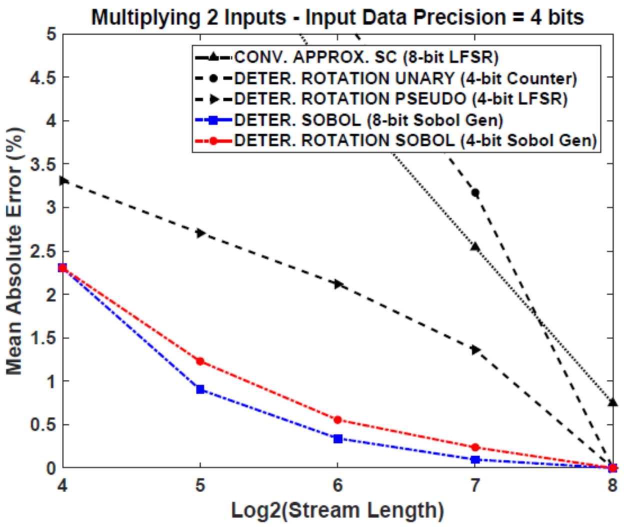
Design Approach	N=4 i=2	N=4 i=3	N=4 i=4	N=8 i=2	N=8 i=3	N=8 i=4
Conv. Approx. SC [5], [11]	397	821	1394	781	1622	2799
Deter. Rotation Unary [6]	224	342	459	492	754	1016
Deter. Rotation Pseudo [9]	262	411	560	536	832	1127
This work 1- Deter. Sobol	1005	3740	9127	3361	13193	32406
This work 2- Deter. Rotation Sobol	456	806	1156	1277	2324	3371

	Converging Speed	Hardware Cost	Cost increase from I=2 to I=4
Rotation Unary	**Very Slow**	**Lowest**	**Lowest increase rate (2X)**
Prop. Method 1	**Very Fast**	**Highest**	**Highest increase rate (9X)**
Prop. Method 2	**Very Fast**	**Medium**	**Low increase rate (2.5X)**

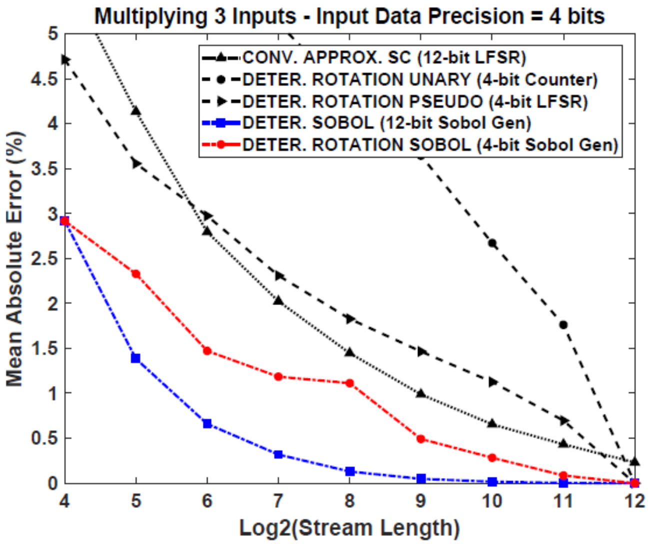
Scalability Evaluation

- Mean Absolute Error (%) of the implemented **4-bit precision** multipliers

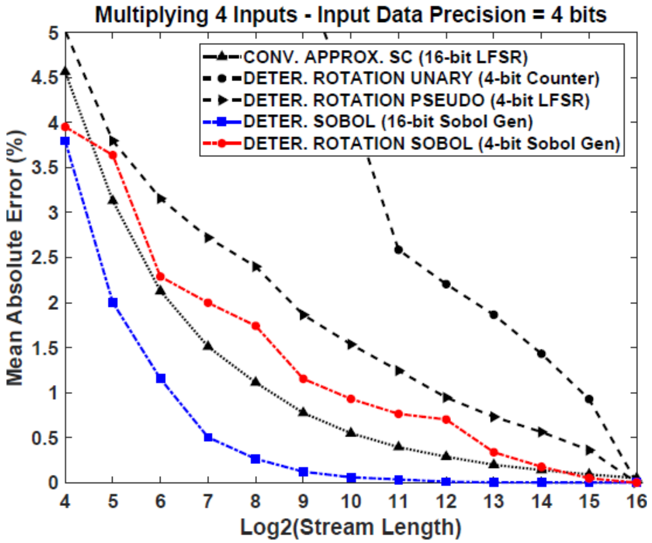
Blue = First method   Red = Second method



2-input



3-input

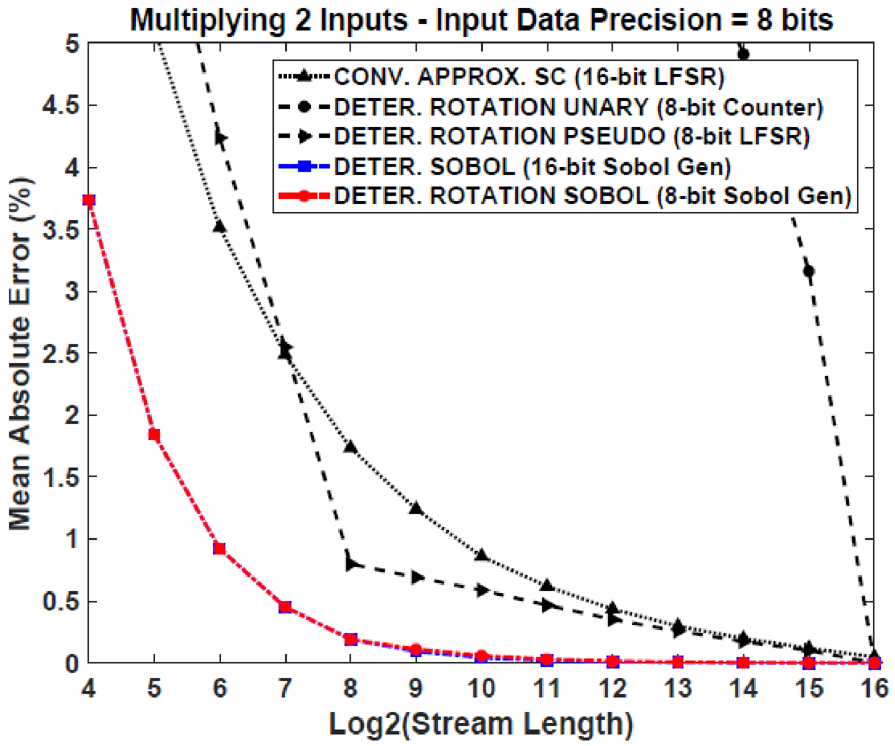


4-input

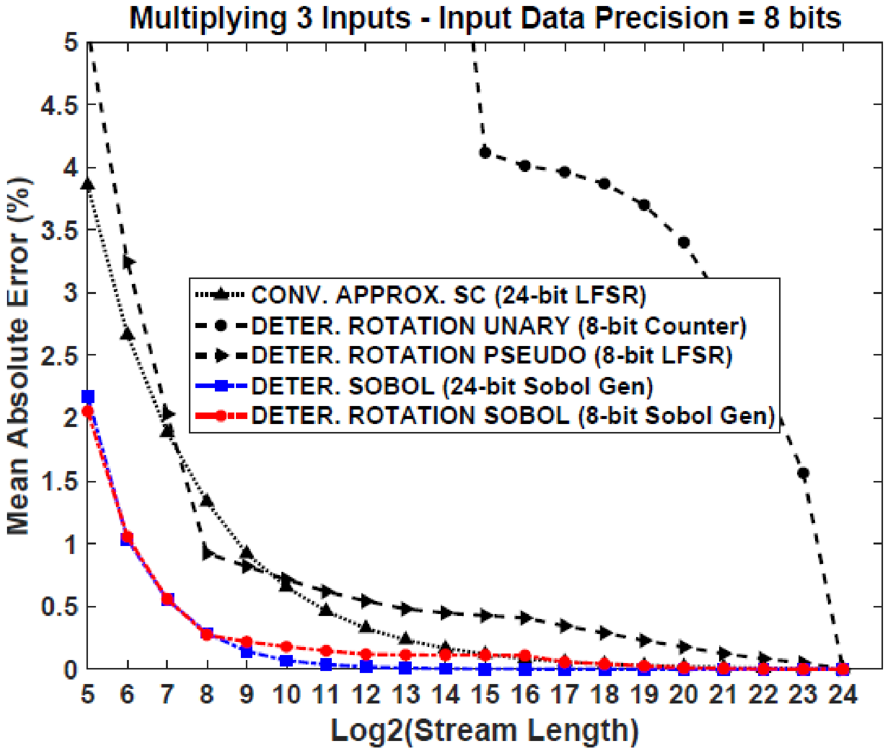
- The computation accuracy of the proposed methods scales with increasing the number of inputs

Scalability Evaluation

- Mean Absolute Error (%) of the implemented **8-bit precision** multipliers



2-input

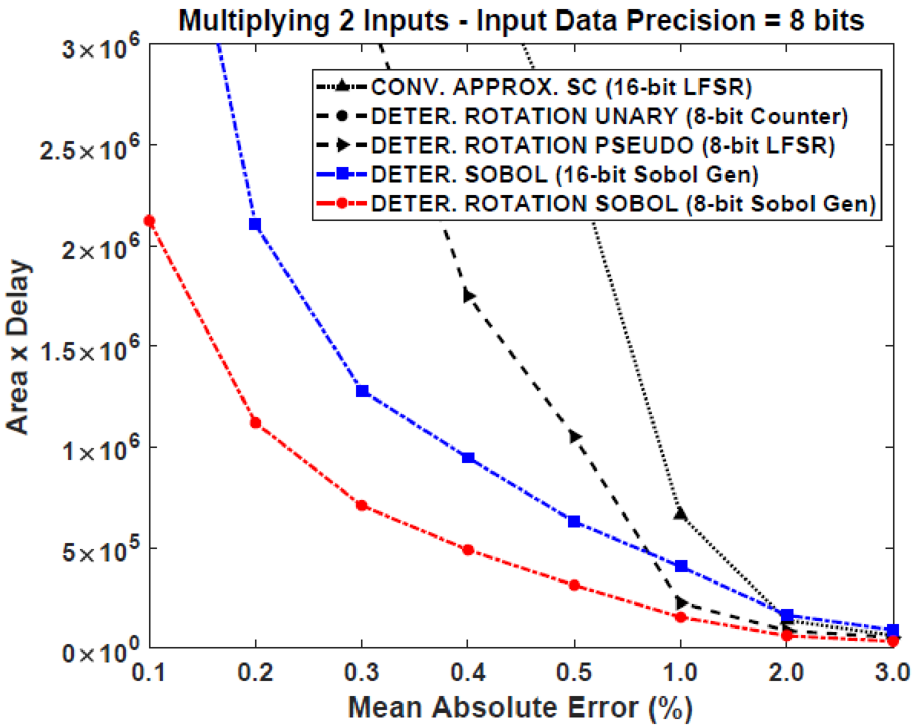


3-input

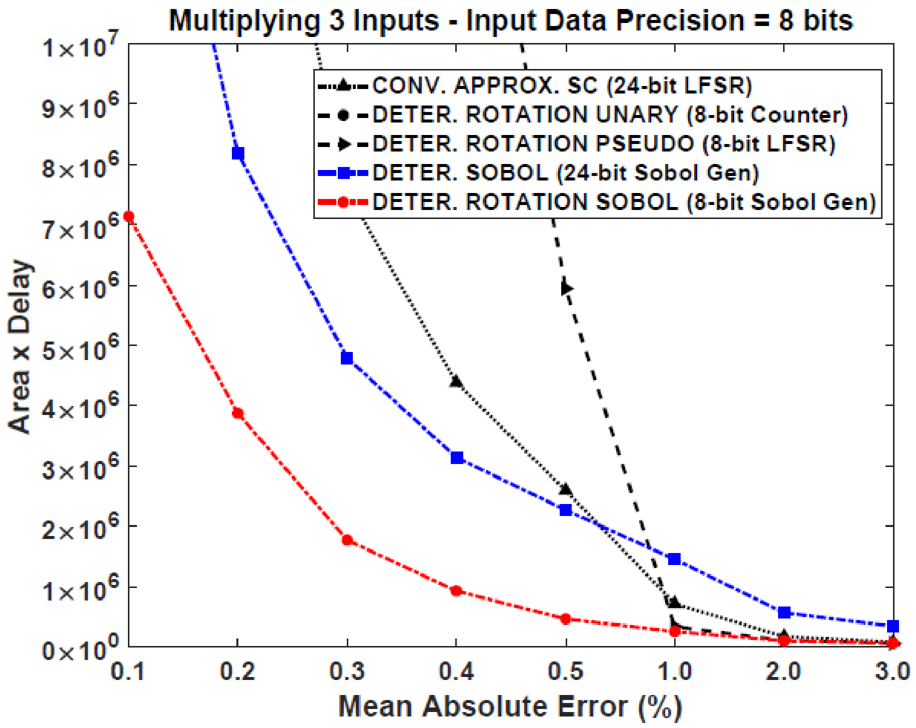
- **We achieved the best accuracy performance by using the two proposed methods**

Scalability Evaluation

- Area x Delay of the implemented **8-bit precision** multipliers for different MAEs



2-input



3-input

- **The second proposed method (red lines) has the lowest area delay product**

Summary

- **Two main challenges** with the recently developed deterministic methods of processing bitstreams
  - **Poor progressive precision**
  - **Limited scalability**

- We proposed **two fast-converging scalable deterministic approaches** for processing bit-streams based on LD sequences
  - **First method**: best accuracy for a fixed processing time
  - **Second method**: lowest area x delay product

- Both methods can produce **completely accurate results**
- A **higher hardware area cost** than prior methods, but **a significantly better progressive precision** makes them a better choice for applications that can tolerate slight inaccuracy
  - e.g., image processing, neural networks

Thank you

Questions?

M. Hassan Najafi
najafi@louisiana.edu