Paper ID: 1036

ASIA SOUTH PACIFIC DAC DESIGN AUTOMATION CONFERENCE

January 22-25, 2024
Incheon, South Korea

P2LSG: Powers-of-2 Low-Discrepancy Sequence Generator for Stochastic Computing

Mehran Shoushtari Moghadam, Sercan Aygun, Mohsen Riahi Alam, and *M. Hassan Najafi

{m.moghadam, sercan.aygun, mohsen.riahi-alam, najafi}@louisiana.edu

University of Louisiana at Lafayette
School of Computing and Informatics
Lafayette, LA, USA

UNIVERSITY of LOUISIANA LAFAYETTE

Outline

- Introduction to Stochastic Computing (SC)
  - Representing numbers with Bit-streams
- Different Random sequences
  - High vs. Low Discrepancy
- Proposed design: P2LSG
- Benchmarking
  - SC Multiplication
  - SC Scaled Addition
- Applications
  - Image scaling
  - Scene merging
- Summary

SC: a re-emerging computing paradigm processing random bit-streams

$\mathbf{X} = \mathbf{4} \,/\, \mathbf{8}$ $\rightarrow$ $\boldsymbol{X}$ **(Bit-stream)**

10101010, 11110000, 11001010,…

No bit significance

$$\overset{2^0}{1}\overset{2^0}{0}\overset{2^0}{1}\overset{2^0}{0}\overset{2^0}{1}\overset{2^0}{0}\overset{2^0}{1}\overset{2^0}{0}$$

Advantages of SC:

1) **Simple execution of complex arithmetic operations (e.g., multiplication using AND)**

2) **Tolerating high rates of noise**

3) **Progressive Precision**

4) **Clock and data skew tolerant**

SC: a re-emerging computing paradigm processing <u>random bit-streams</u>

$\mathbf{X} = \mathbf{4}$ / 8 $\rightarrow$ $\boldsymbol{X}$ (Bit-stream)

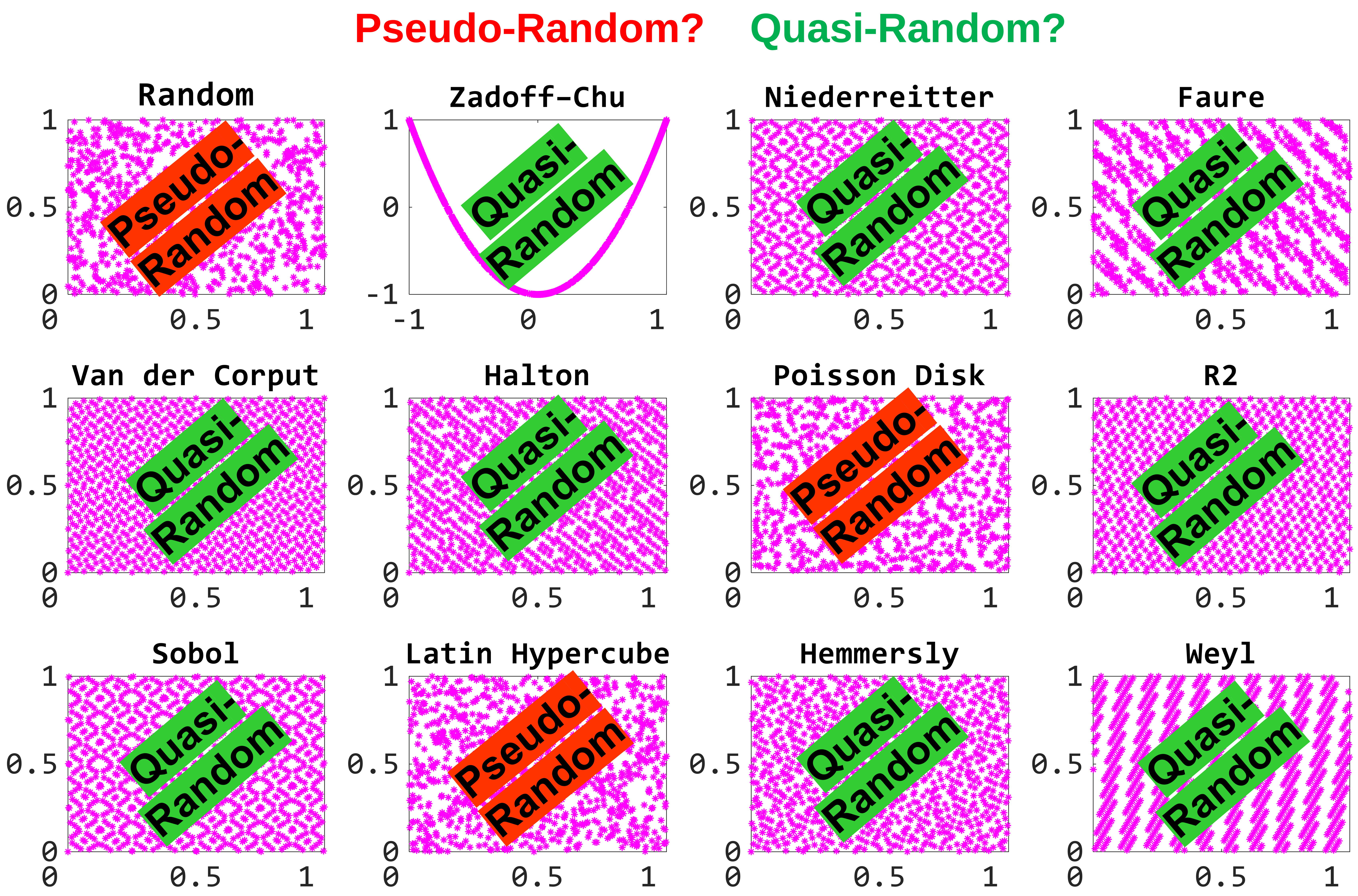
But how can we generate stochastic bit-stream?

compare

random #
scalar X $\Rightarrow$ $\begin{cases} 1 & X > \text{random \#} ♠ \\ 0 & X \leq \text{random \#} ♠ \end{cases}$

10101010, 11110000, 11001010,…

To generate a bit-stream of N bits, we need N random numbers!

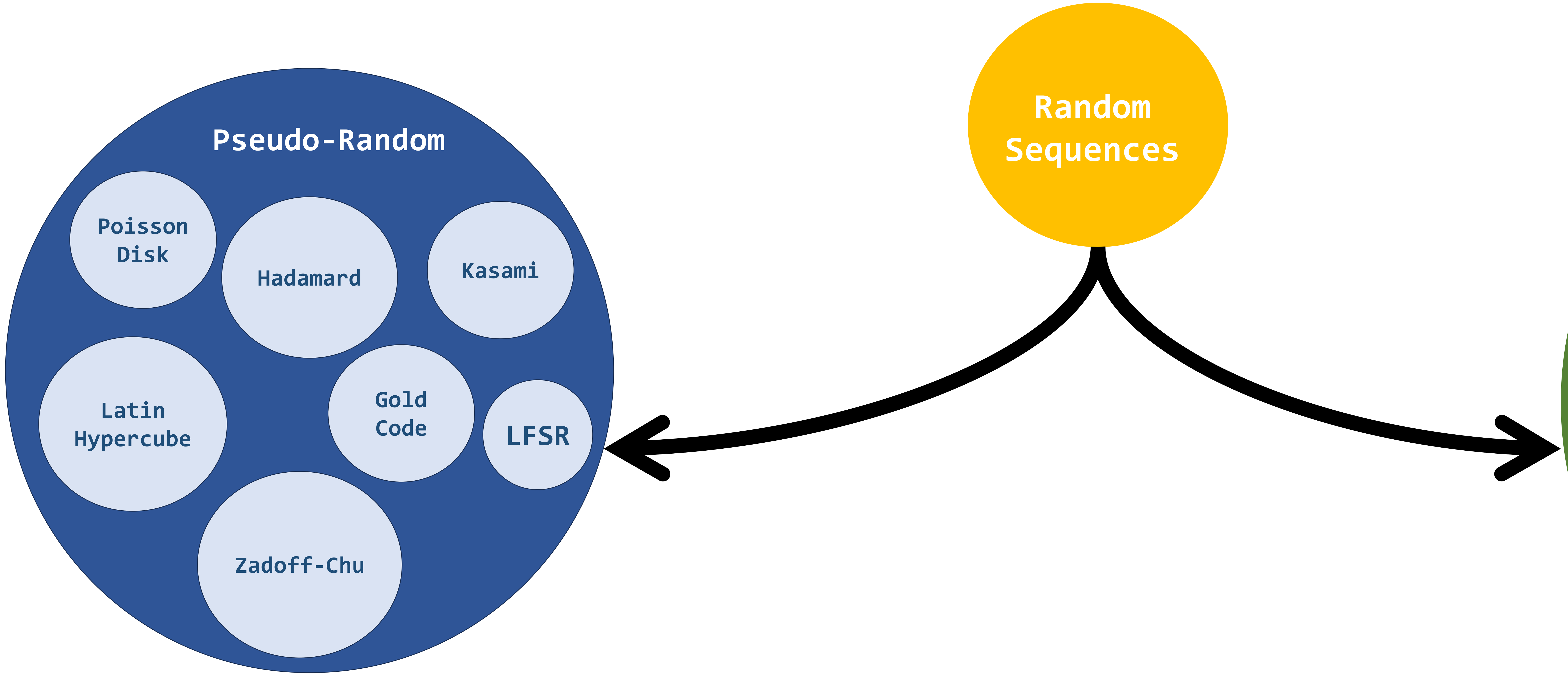


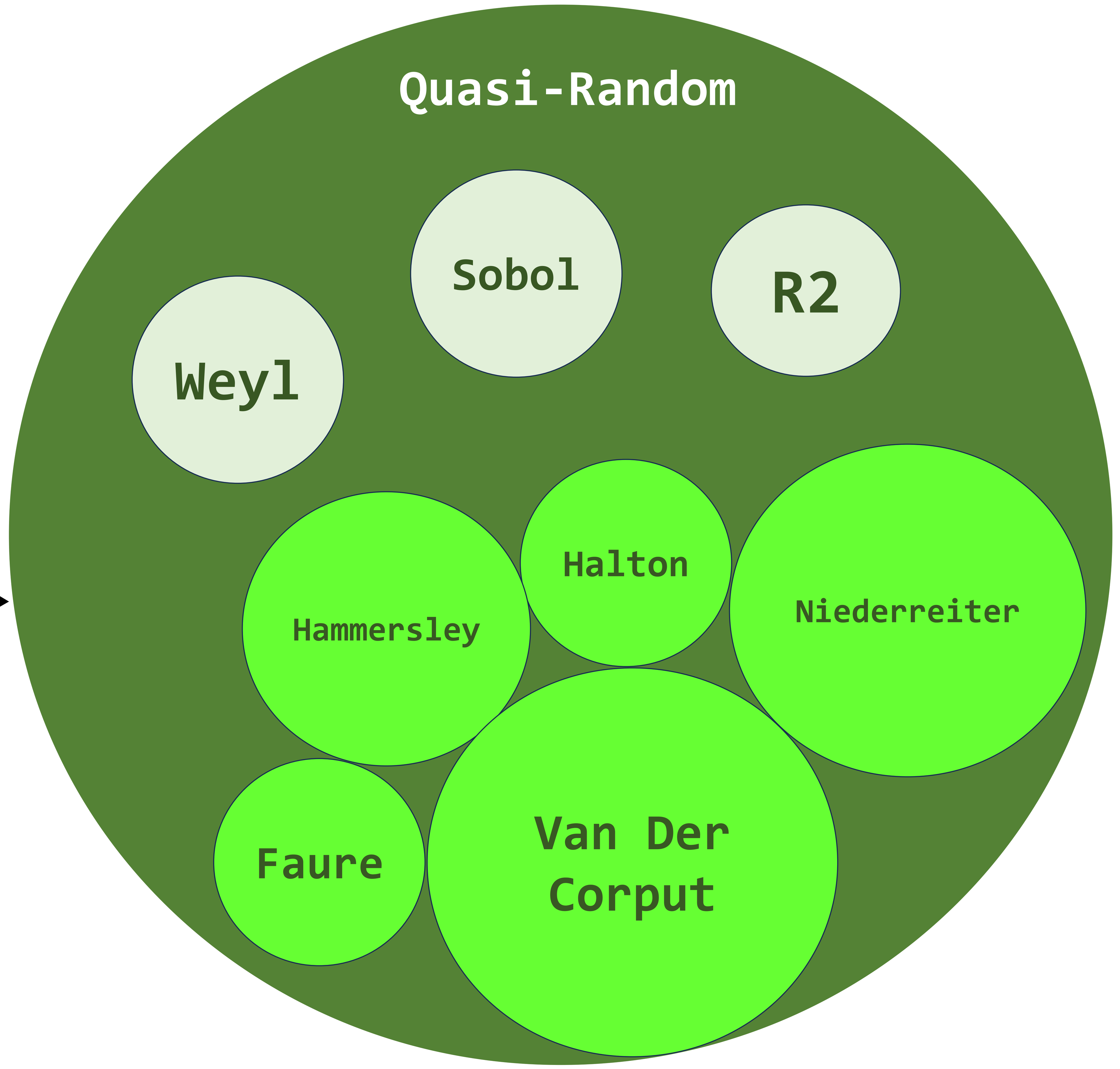
Sequence 1 versus Sequence 2 plots

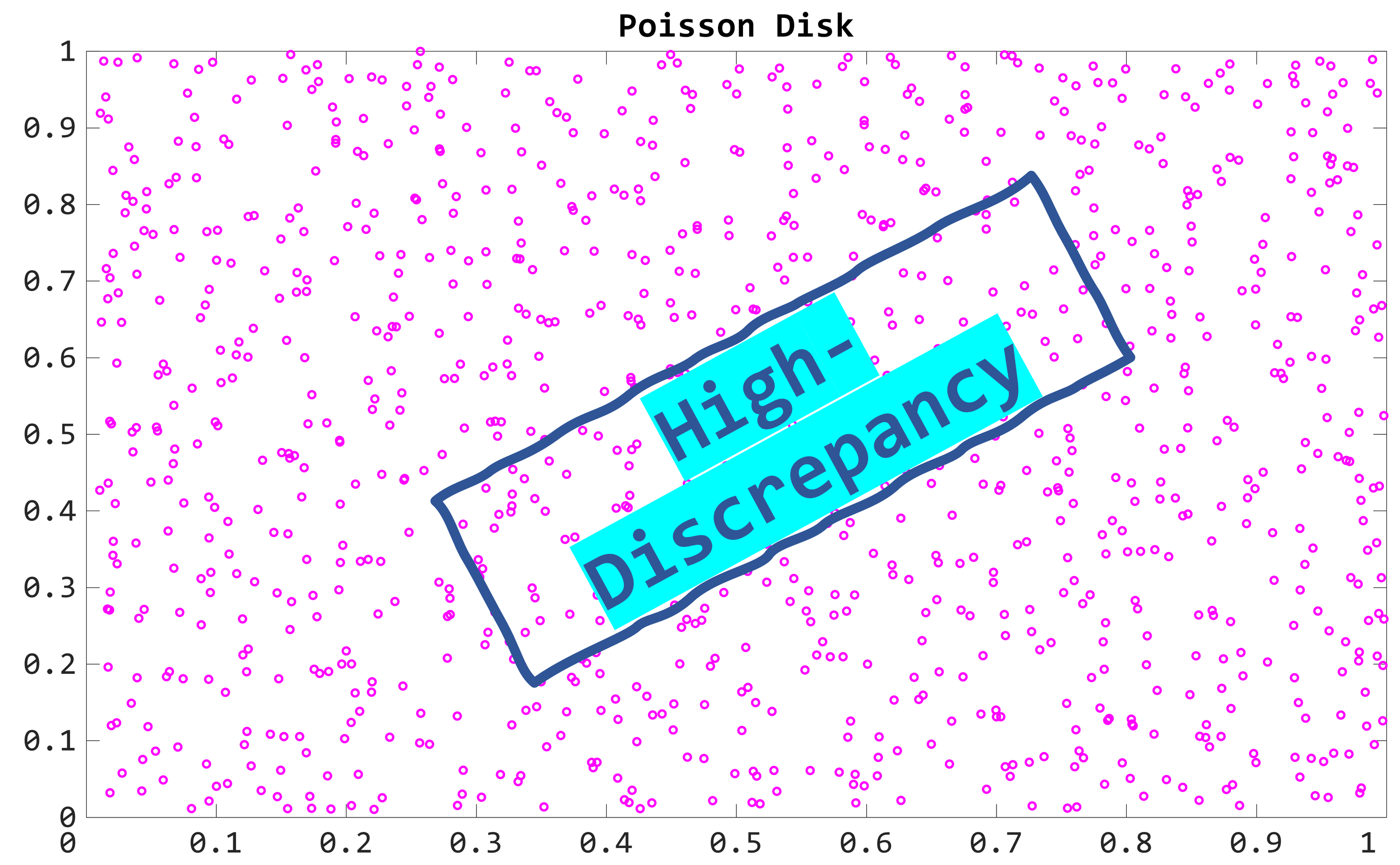
Various Random Sequences including Pseudo-Random and Quasi-Random characteristics

We explore some new random sequences for the first time in SC

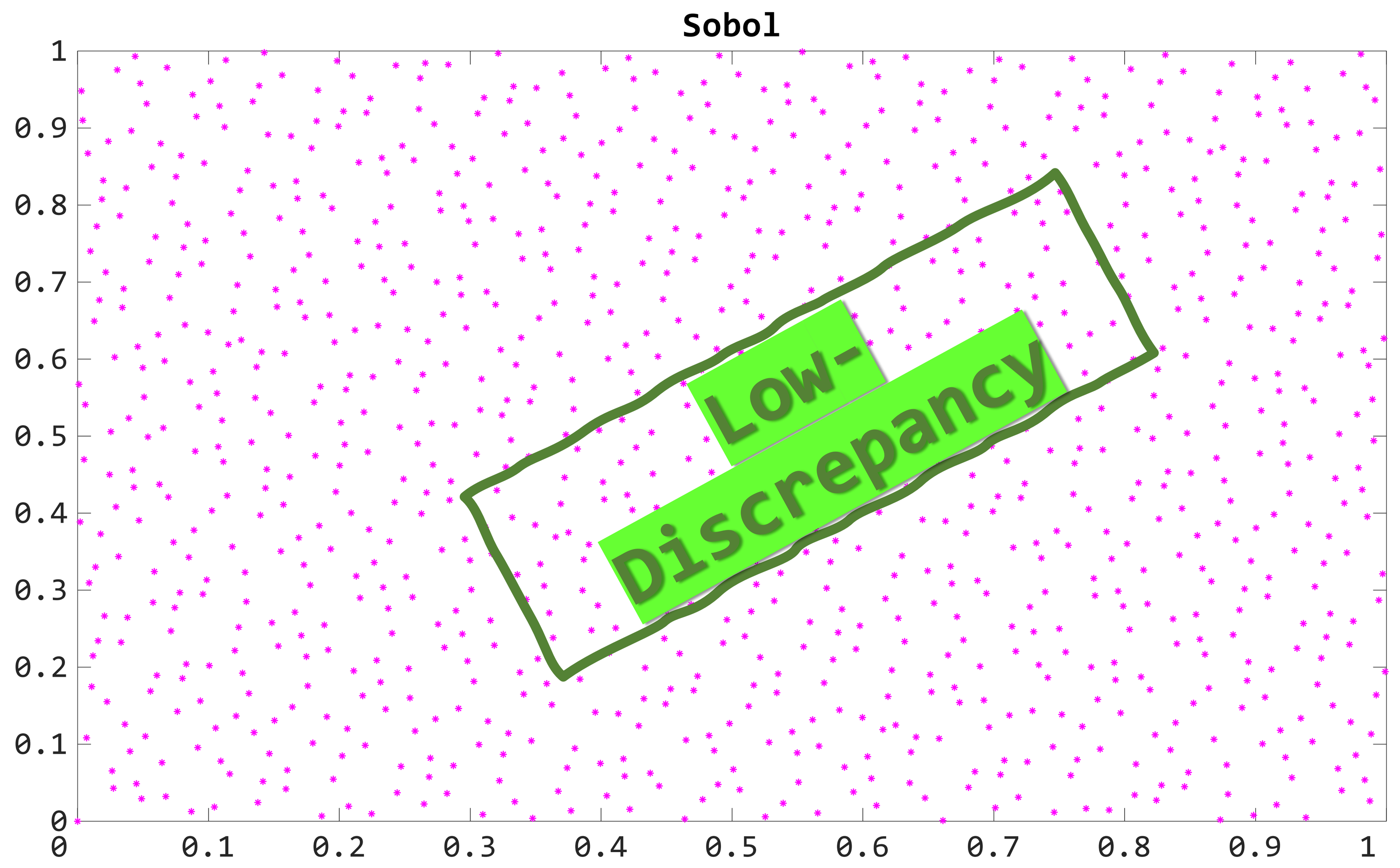
Random Sequences







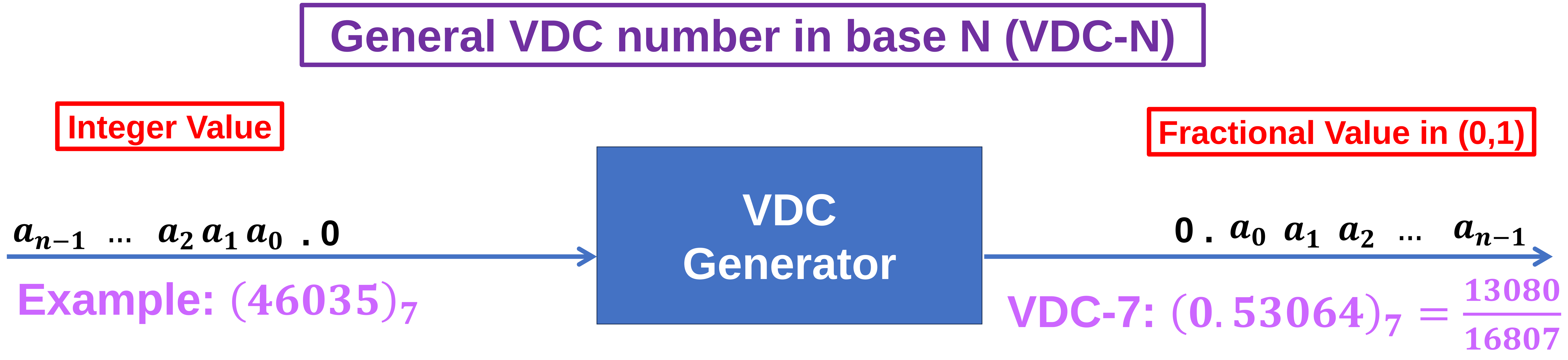
- ***High-discrepancy***
  - Unequally distributed in multi-dimensional space
- ***Low-discrepancy***
  - Evenly distributed in multi-dimensional space



Van Der Corput (VDC) Sequence Generation

- **A high-quality random sequence SC: Van Der Corput (VDC)**

- **General Algorithm to generate a base-B VDC sequence**
  - Generate an **integer** number
  - **Convert** the integer number to **base-B representation**
  - **Reverse** the base-B representation
  - **Convert** the reversed base-B representation to a binary number



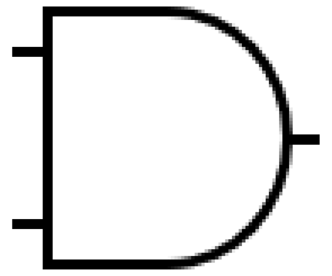
Complexity of Hardware Design depends on the chosen Base

I) Non-Powers-of-2 Bases

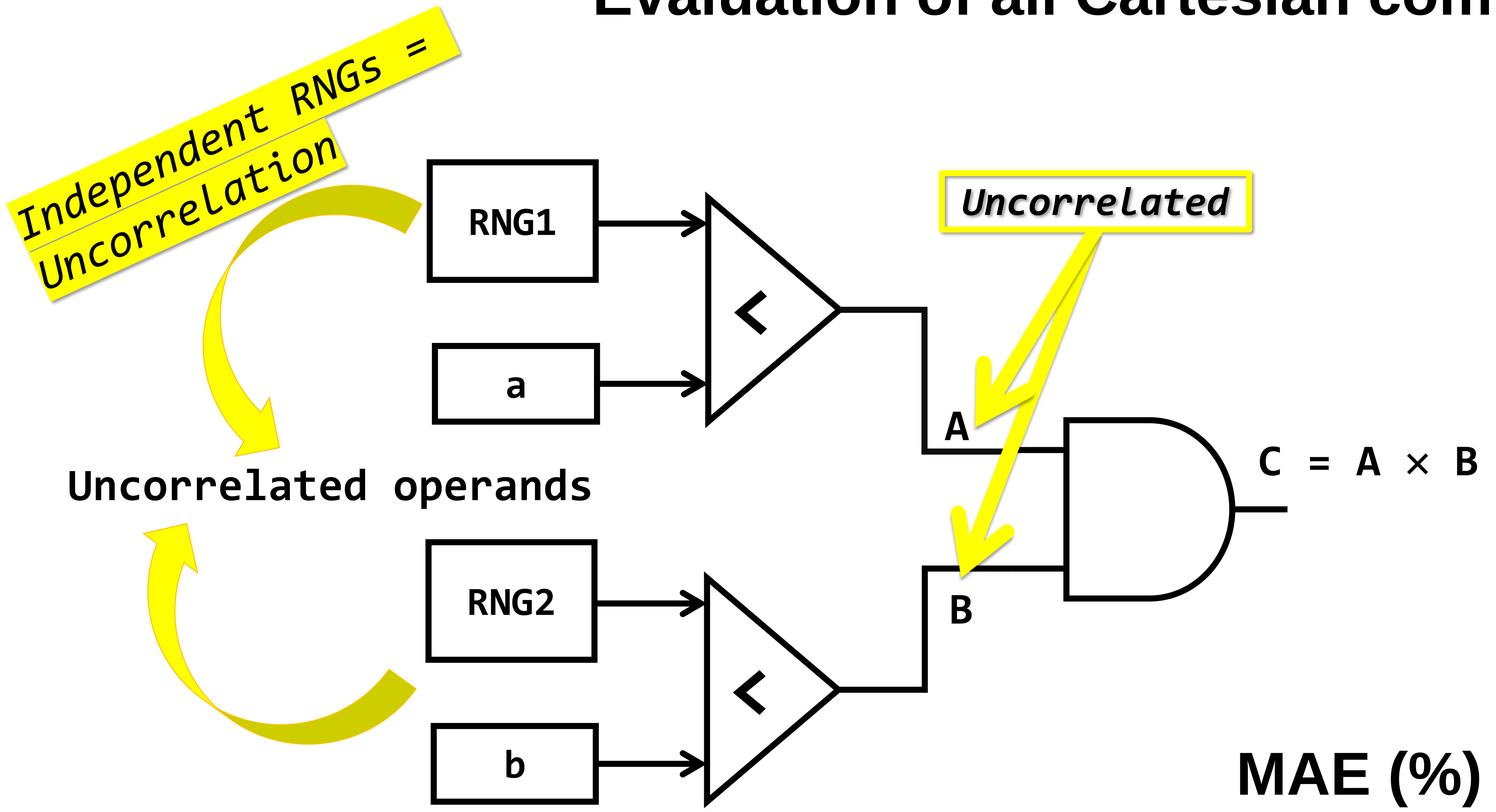
II) Powers of 2 Bases

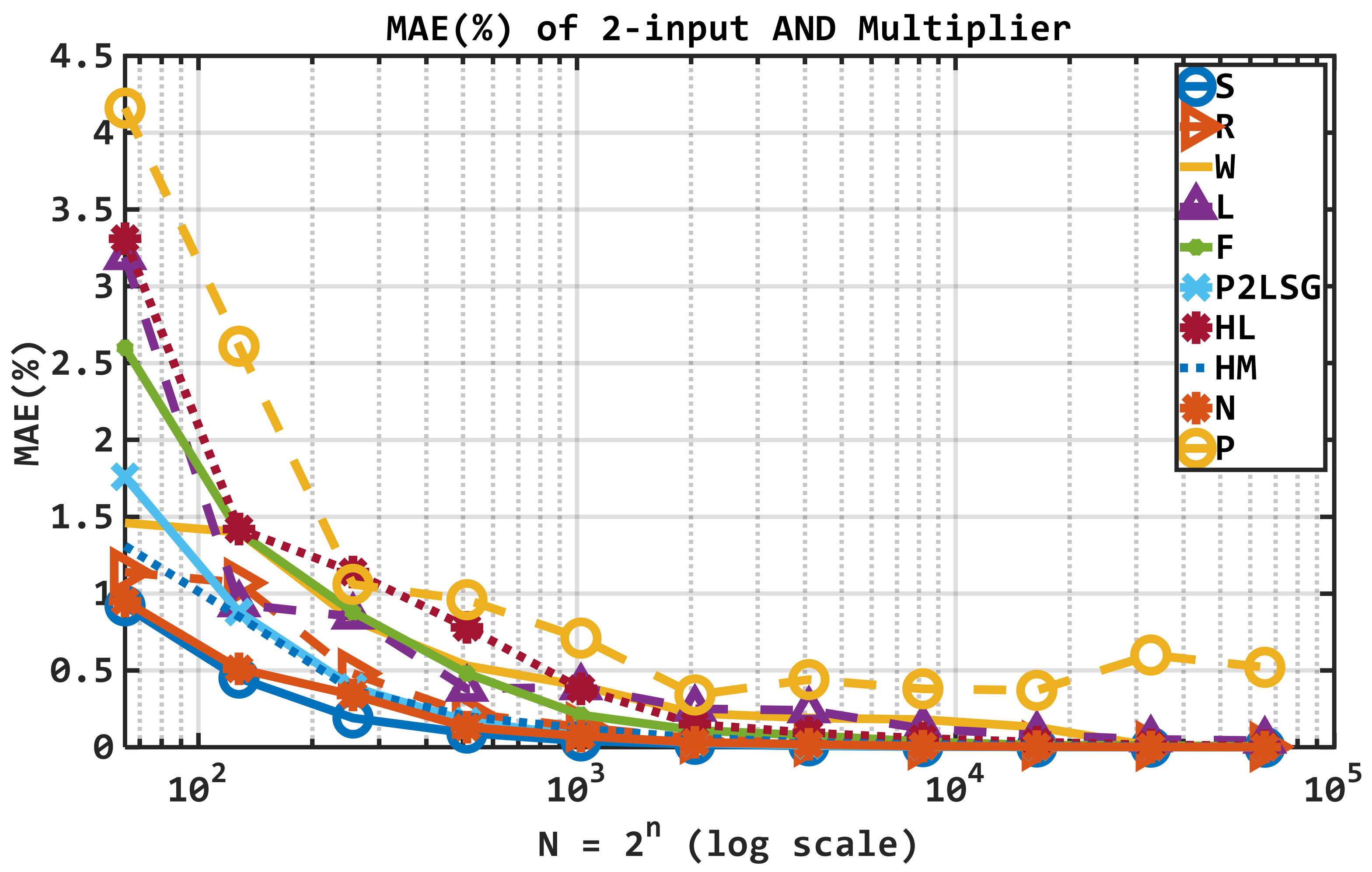
Benchmark 1

Evaluation of all Cartesian combinations of inputs in 8-bit precision



Multiplication

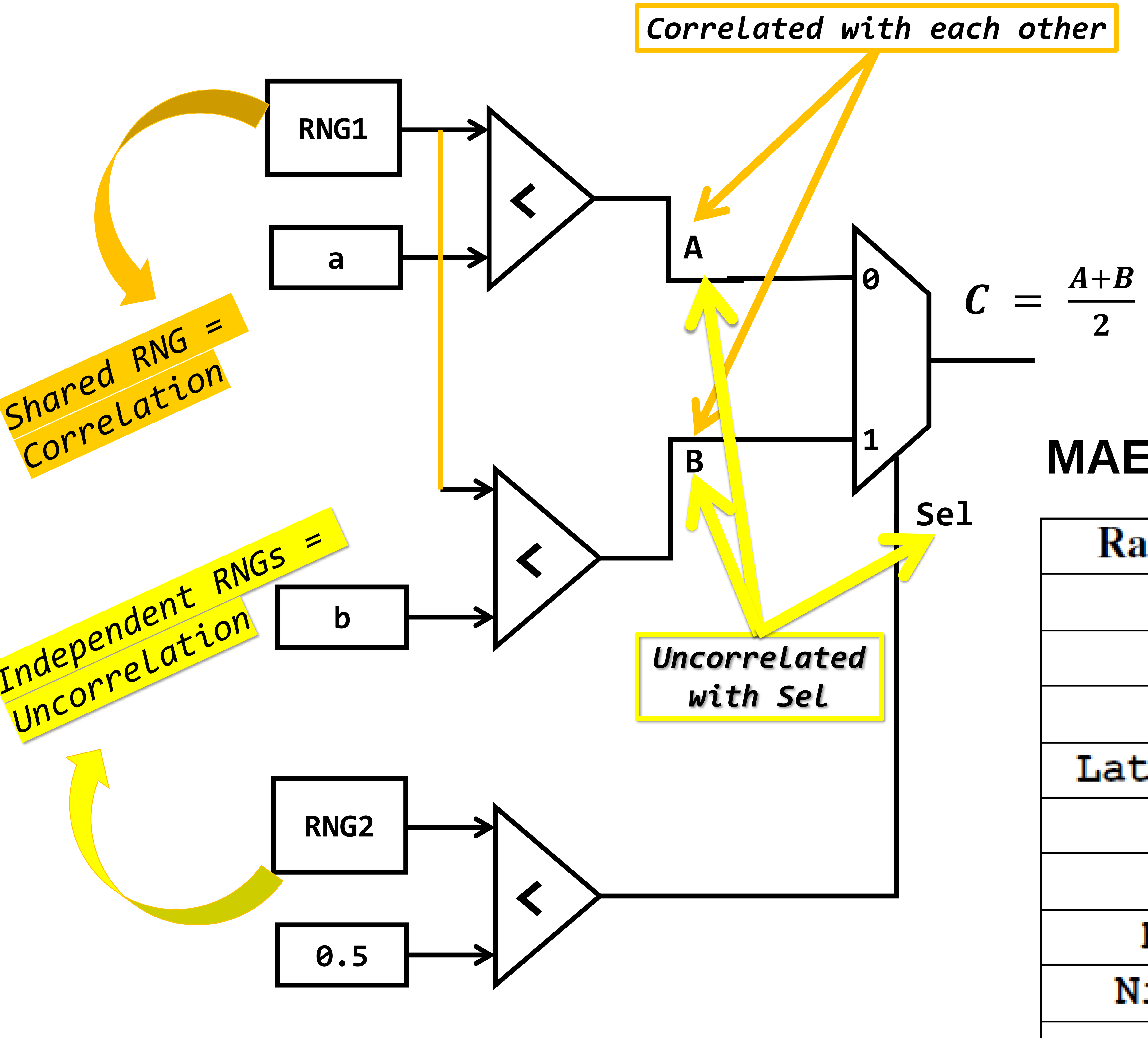




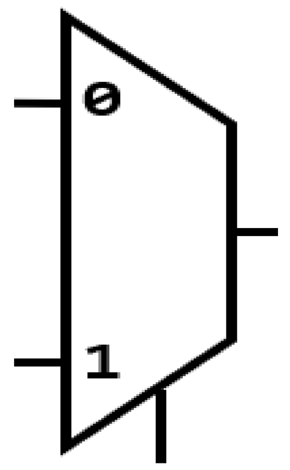
Best accuracy necessitates
Uncorrelation

MAE (%) comparison of multiplying two 8-bit precision bit-streams

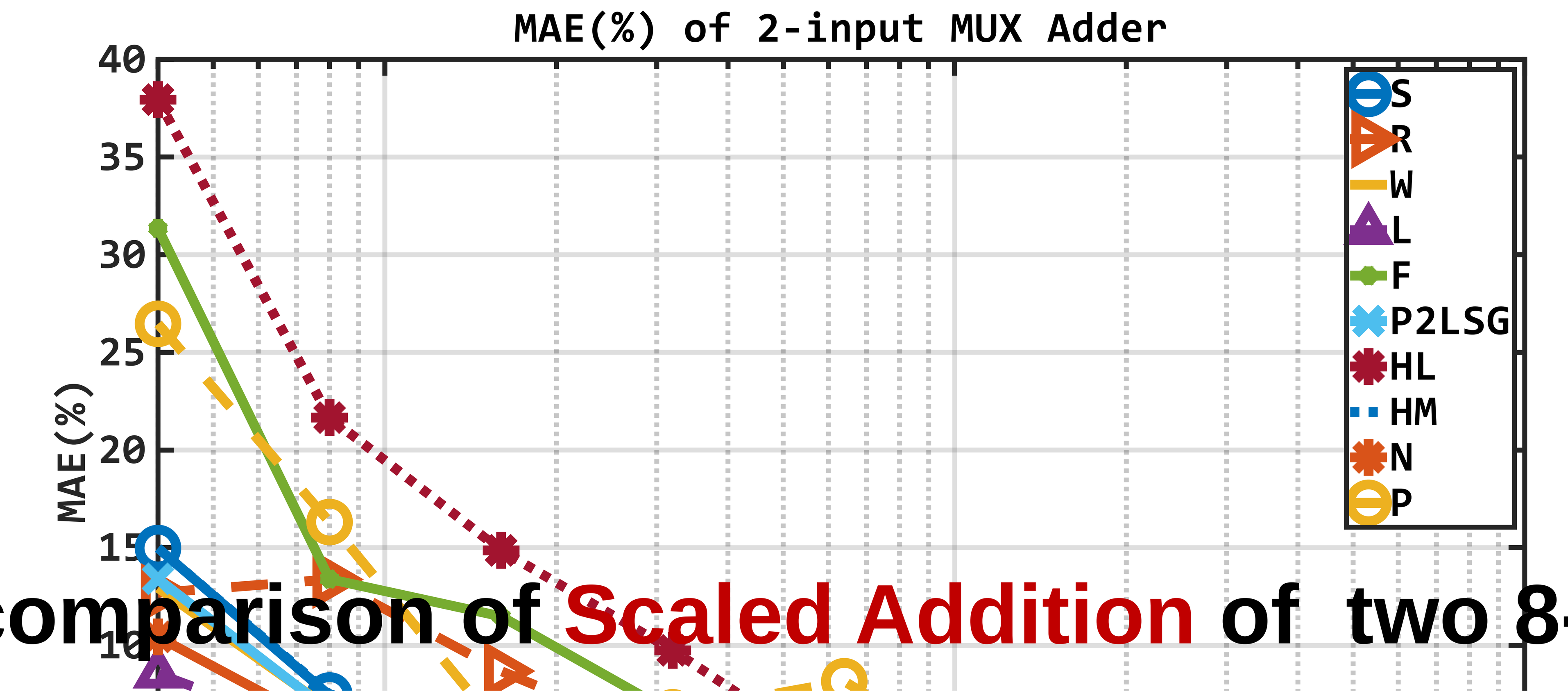
Random Sequence	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
Sobol	0.92	0.45	0.19	0.092	0.041	0.019	0.009	0.0035	0.0013	0.0003	0.0000
R2	1.14	1.07	0.48	0.220	0.130	0.055	0.037	0.0164	0.0099	0.0078	0.0024
Weyl	1.46	1.40	0.83	0.530	0.400	0.220	0.190	0.1800	0.1300	0.0090	0.0065
Latin Hypercube	3.19	0.93	0.85	0.380	0.390	0.250	0.240	0.1200	0.0795	0.0508	0.0424
Faure	2.60	1.40	0.88	0.480	0.210	0.110	0.077	0.0360	0.0136	0.0113	0.0040
Halton	3.31	1.42	1.14	0.780	0.380	0.150	0.093	0.0570	0.0330	0.0150	0.0083
Hammersley	1.31	0.85	0.37	0.200	0.120	0.061	0.030	0.0170	0.0098	0.0043	0.0019
Niederreiter	0.95	0.51	0.34	0.130	0.072	0.032	0.019	0.0067	0.0039	0.0015	0.0011
Poisson Disk	4.16	2.61	1.06	0.960	0.710	0.340	0.440	0.3800	0.3700	0.6000	0.5200
P2LSG	1.76	0.88	0.39	0.170	0.073	**0.030**	**0.012**	**0.0045**	**0.0015**	**0.0003**	**0.0000**



$$C = \frac{A+B}{2}$$

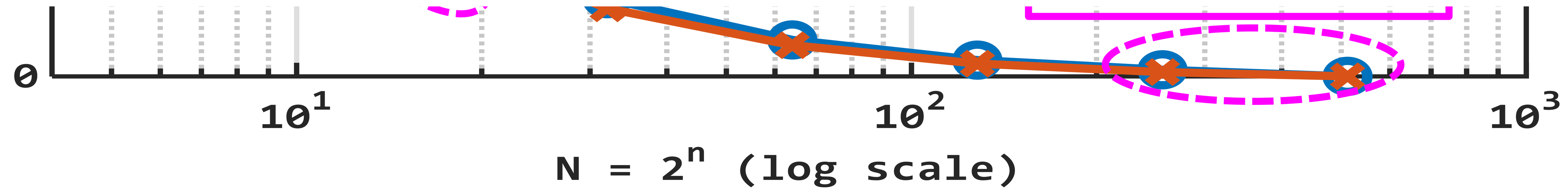


Scaled Addition

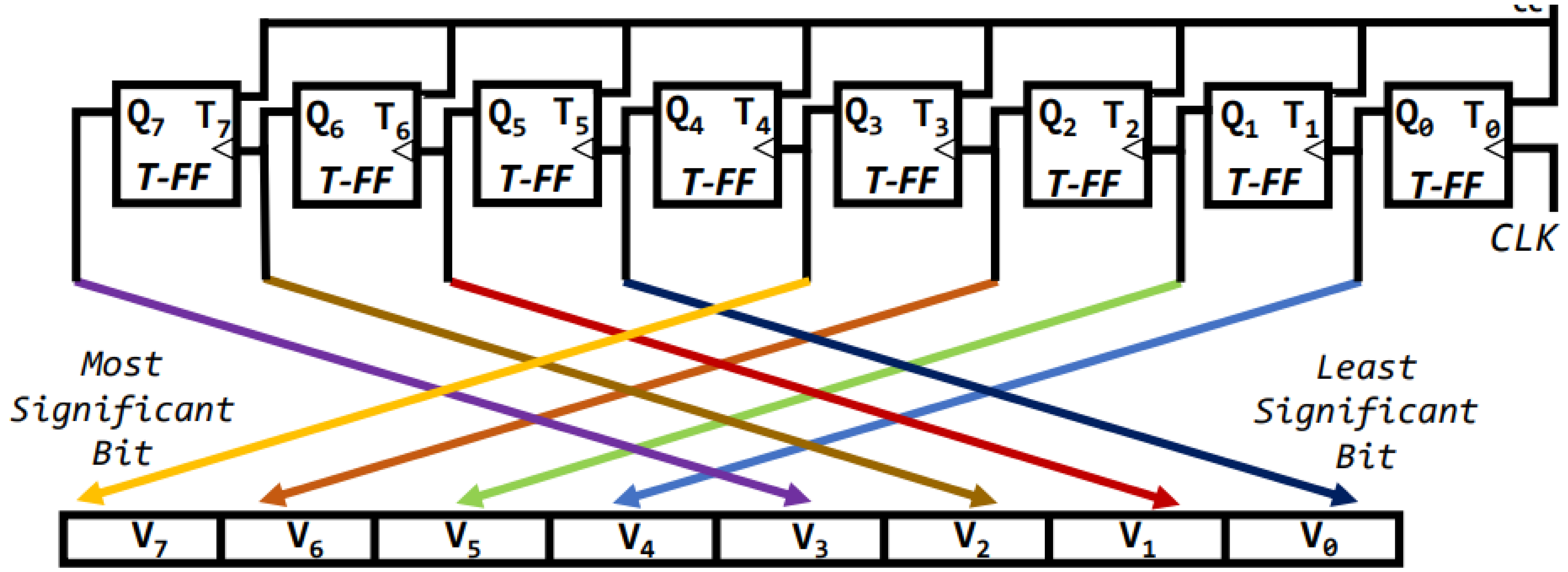


MAE (%) comparison of Scaled Addition of two 8-bit precision bit-streams

Random Sequence	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9
Sobol	14.98	7.43	3.66	1.77	0.83	0.37	0.15	0.00
R2	12.69	13.37	8.68	5.04	1.91	1.66	0.60	0.25
Weyl	12.94	6.74	4.52	3.07	2.19	1.17	0.92	0.79
Latin Hypercube	8.54	5.30	3.92	2.95	1.81	0.97	1.02	0.86
Faure	31.35	13.39	11.48	6.52	2.96	2.59	1.62	0.78
Halton	37.94	21.66	14.86	9.74	3.95	2.09	1.58	0.75
Hammersley	14.98	7.49	5.55	2.49	1.19	0.85	0.29	0.18
Niederreiter	10.43	5.74	3.18	1.65	0.81	0.36	0.15	0.00
Poisson	26.46	16.31	5.66	6.57	8.16	2.69	1.53	1.65
P2LSG	13.40	6.63	3.24	**1.55**	**0.71**	**0.29**	**0.097**	**0.00**



Sequential Design



P2LSG using 8-bit counter

$b_n b_{n-1} b_{n-2}$ … … … $b_3 b_2 b_1 b_0$ by-$\log_2(B)$ grouping

$b_3 b_2 b_1 b_0$ … … … $b_n b_{n-1} b_{n-2}$ Signif. inversion

Examples of structuring an 8-bit counter to generate P2LSG-4, P2LSG-8, and P2LSG-16

by-$\log_2(4)$ grouping: $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$
Significance inversion: $b_1 b_0 b_3 b_2 b_5 b_4 b_7 b_6$

by-$\log_2(8)$ grouping: $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$
Signif. inversion: $b_2 b_1 b_0 b_5 b_4 b_3 0 b_7$

by-$\log_2(16)$ grouping: $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$
Signif. inversion: $b_3 b_2 b_1 b_0 b_7 b_6 b_5 b_4$

Hardware cost for generating two different 8-bit random sequences

SNG	Area (μm^2)	Power (μW)	CPL (ns)
Sobol#2 & Sobol#3 [2]	2×781	2×45.15	0.68
Halton#1 & Halton#2	$130 + 450$	$15.15 + 35.30$	1.06
✅ **P2LSG-4 & P2LSG-16**	**163**	**16.05**	**0.49**

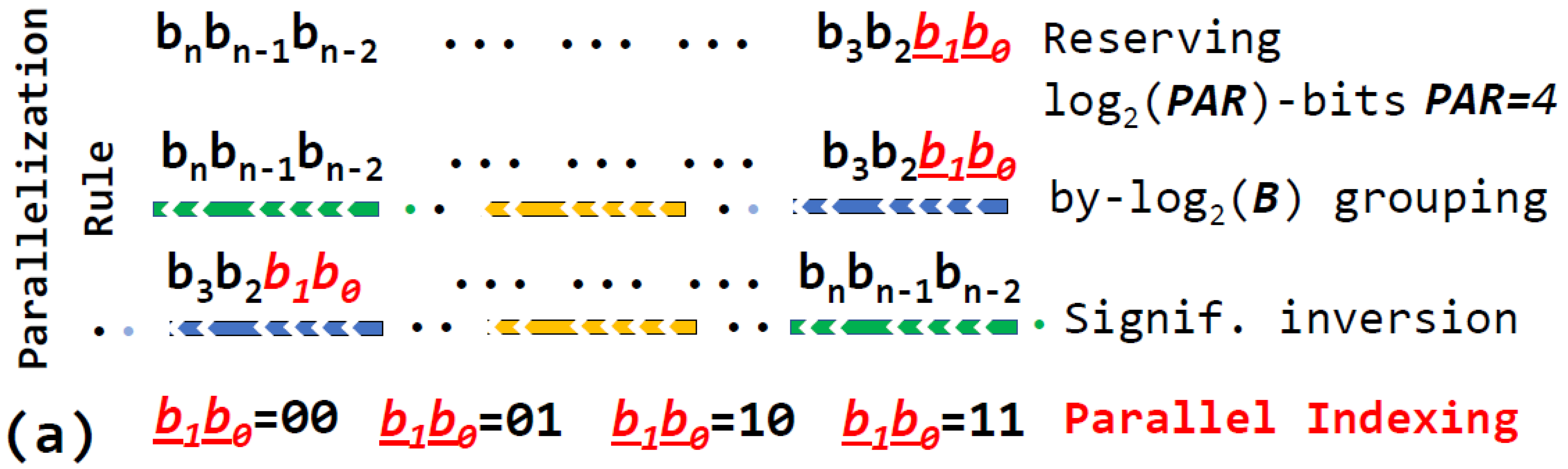
👍 **Improvement compared to Sobol**

- Area ≈ 9.5 ×
- Power ≈ 5.62×

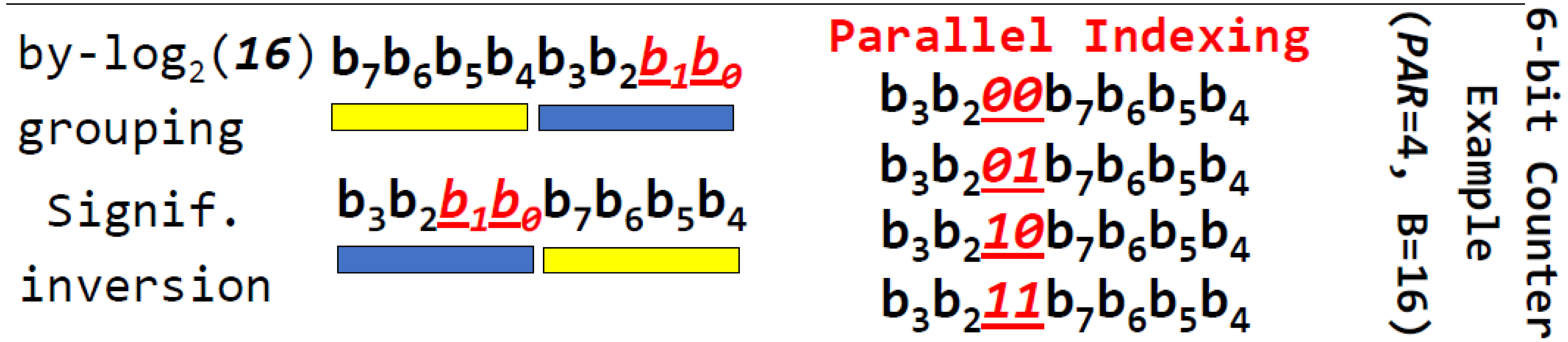
Parallel Design

- We can also implement a **parallel** P2LSG to **generate more than one number** (i.e., *PAR* numbers) of the sequence at any cycle

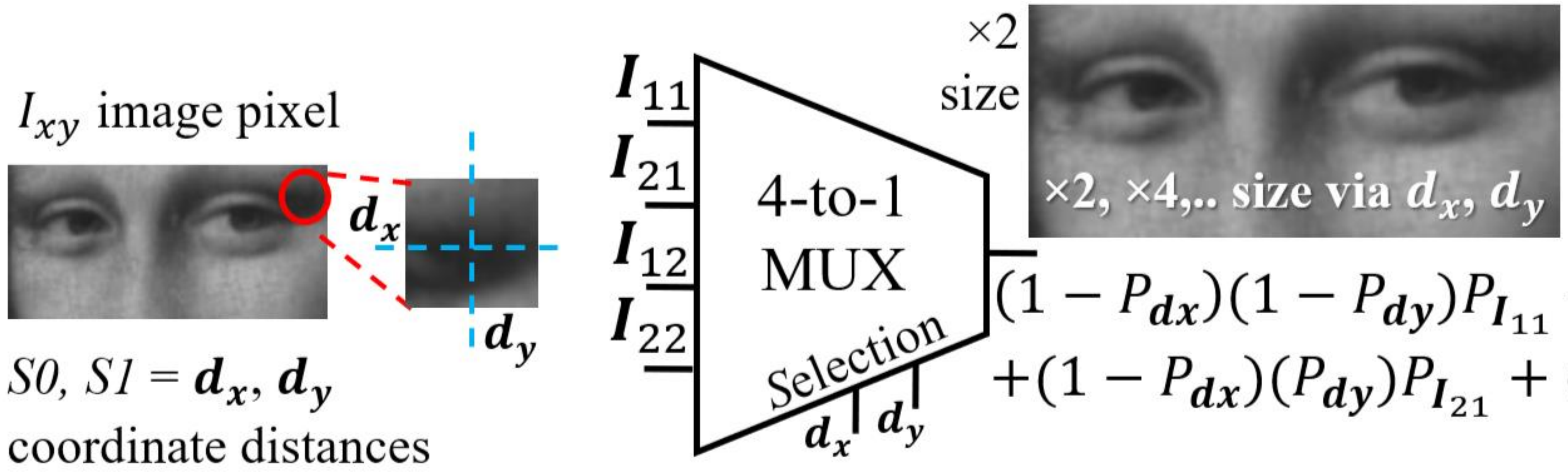
- The **general rule** to assign parallel indexing bits



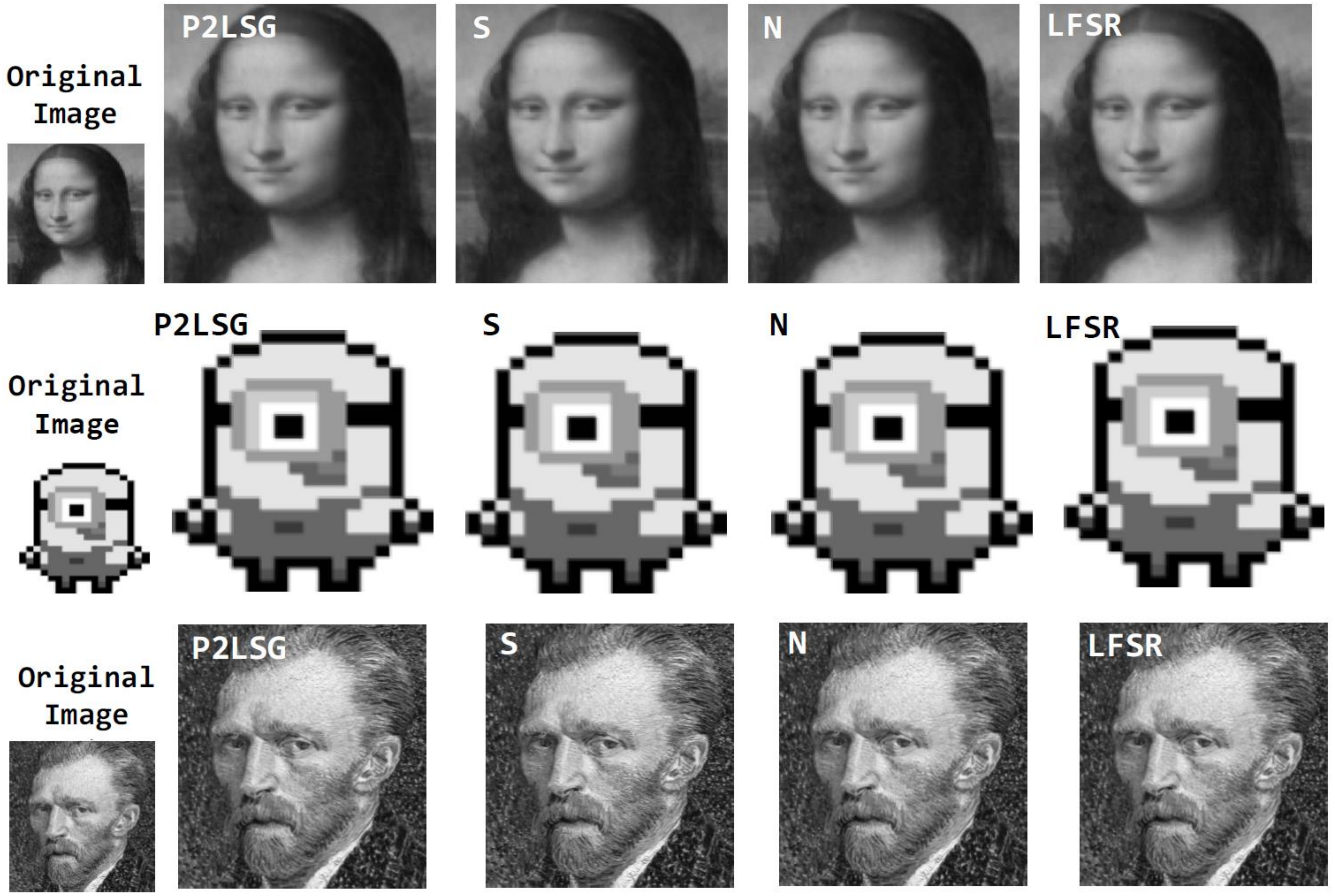
- P2LSG-16 example with *PAR=4* concurrent number generation.



Case Study 1: Interpolation & Image Scaling

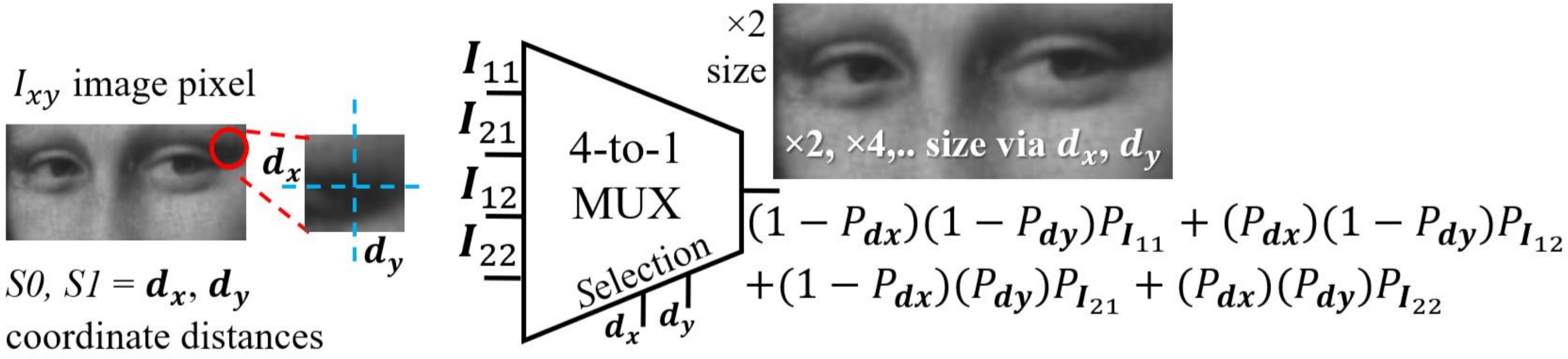


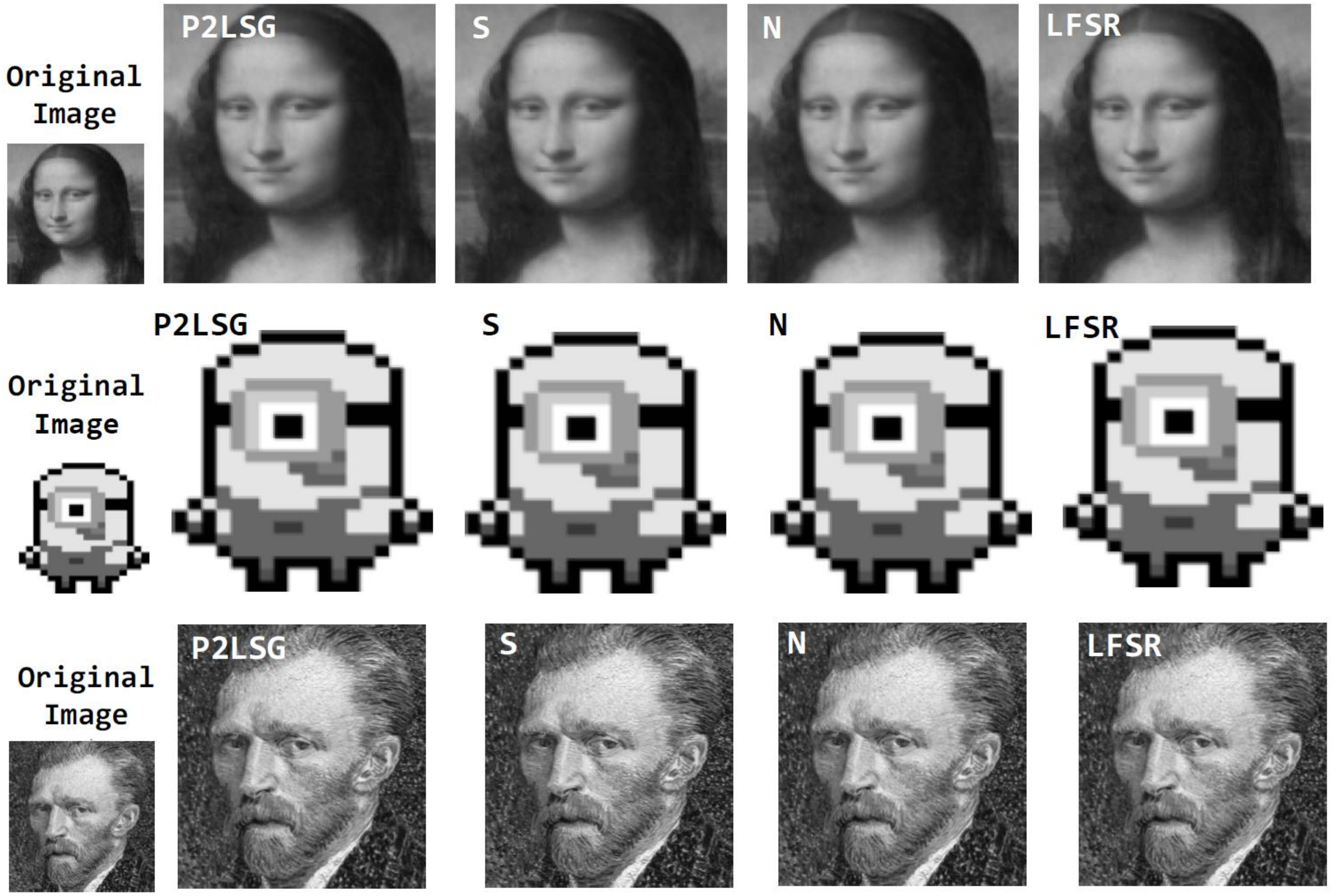
$$(1-P_{dx})(1-P_{dy})P_{I_{11}} + (P_{dx})(1-P_{dy})P_{I_{12}}$$
$$+(1-P_{dx})(P_{dy})P_{I_{21}} + (P_{dx})(P_{dy})P_{I_{22}}$$



SC IMAGE SCALING WITH DIFFERENT RANDOM SEQUENCES.

Image Scaling (*Scale Factor*=2)			
✅ P2LSG	Sobol (S)	Niederreiter (N)	LFSR
Mona Lisa			
PSNR: **46.62dB** SSIM: **0.9958**	PSNR: 46.25dB SSIM: 0.9948	PSNR: 46.37dB SSIM: 0.9949	PSNR: 44.63dB SSIM: 0.9892
Minion			
PSNR: **39.36dB** SSIM: **0.9975**	PSNR: 39.26dB SSIM: 0.9926	PSNR: 39.30dB SSIM: 0.9958	PSNR: 38.50dB SSIM: 0.9964
Van Gogh			
PSNR: **43.10dB** SSIM: **0.9958**	PSNR: 42.91dB SSIM: 0.9921	PSNR: 42.99dB SSIM: 0.9923	PSNR: 41.65dB SSIM: 0.9880





Hardware cost comparison in 45nm technology when processing Mona Lisa image (107 x 104 size)

N=256	Area (μm^2)	*Energy (pJ)	*CPL (ns)	Total Energy (μJ)	Runtime
	Image Scaling				
Sobol [2]	2017	17.55	366	0.781	16.3 ms
Parallel 4× Sobol	4548	16	91.5	0.713	4.07 ms
P2LSG	**715**	**5.6**	**317**	**0.25**	**14.12 ms**
Parallel 4× P2LSG	**2040**	**2.4**	**71**	**0.107**	**3.16 ms**

Improvements (Parallel Design)

Energy ≈ 6.66×

Area ≈ 2.23×

Case Study 2: Scene Merging (Video Processing)

Merged Pixel =
Background Pixel × (1 − alpha) +
Foreground Pixel × alpha

P2LSG:
PSNR: 48.23 dB
SSIM: 0.999

Sobol:
PSNR: 48.13 dB
SSIM: 0.9999

Hardware cost comparison of scene merging

N=256	Area (μm^2)	*Energy (pJ)	*CPL (ns)	Total Energy (μJ)	Runtime
	Scene Merging				
Sobol [2]	1787	16.6	353	1568	33.3 s
Parallel 4× Sobol	3628	15.1	88	1424	8.33 s
P2LSG	**485**	**4.7**	**304**	**443**	**28.7 s**
Parallel 4× **P2LSG**	**1120**	**1.48**	**68**	**140**	**6.4 s**

*Energy and *CPL are for producing each output pixel. Bit-stream Length (N) is 256.

Improvements (Parallel Design)

Energy ≈ 10.17×

Area ≈ 3.24×

Summary

- ✓ **Powers-of-2 Low-Discrepancy Sequence Generator (P2LSG):**

  Cost-efficient and accurate random bit-stream generation in SC

- ✓ **Efficient & ultra light-weight hardware design**

- ✓ **Parallel processing opportunities**

- ✓ **Explored a new set of deterministic sequences for SC**

- ✓ **Evaluated practicality in *image scaling & scene merging***

- ✓ **Potential for other computing paradigms: *Hyperdimensional Computing (HDC)***

Thank you for your Attention!

najafi@louisiana.edu