Embedded.Systems@UCC

MCCI Microelectronic Circuits Centre Ireland

Efficient Architectures and Implementation of Arithmetic Functions Approximation Based Stochastic Computing

Tieu-Khanh Luong[1], Van-Tinh Nguyen[2], Anh-Thai Nguyen[3] and Emanuel Popovici[1,4]

MCCI + Embedded.Systems @ University College Cork[1], Nara Institute of Science and Technology[2], Le Quy Don Technical University[3]

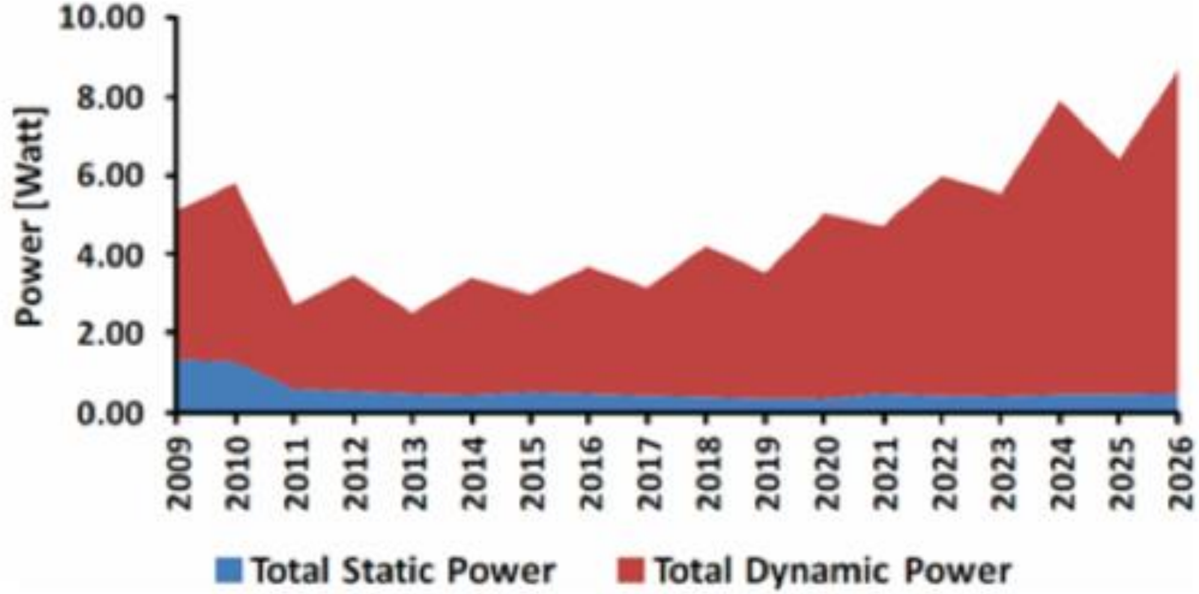
ASAP 2019

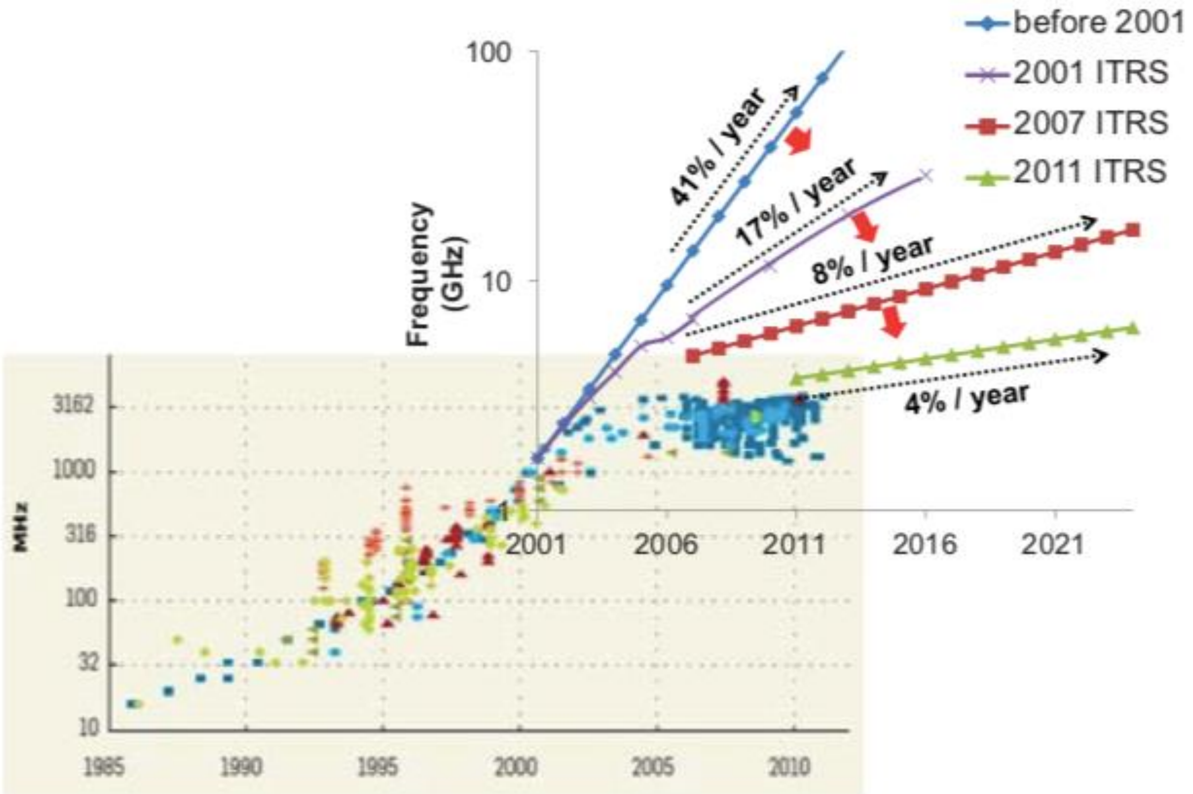
Outline

- ❑ Introduction and Previous Works
- ❑ Proposed Approach
- ❑ Hardware Architectures
- ❑ Experimental Results
- ❑ Conclusion

Motivation: Low Power Challenge at the Edge

- IoT and Edge devices require ultra low power solutions
- From big data in the cloud to the low power smart sensor
- Possible solution: to employ new design paradigms to overcome the challenge.





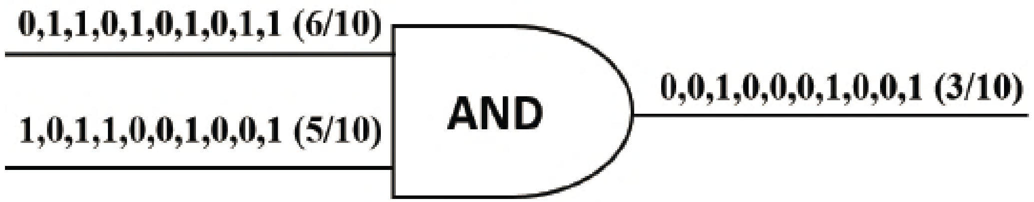
MCCI
Microelectronic Circuits Centre Ireland

New Paradigm: Stochastic Computing (SC)

- ❑ A re-emerging computing paradigm: introduced in 1969
- ❑ Has gained attention recently due its low power and error tolerance
- ❑ Logical computation on random bit streams
- ❑ Value: **probability of obtaining a one versus a zero**
  - ❖ **Unipolar [0, 1] positive**
    - Each bit has the probability X of being 1
  - ❖ **Bipolar [-1, 1] positive, negative**
    - Each bit has the probability (X+1)/2 of being one

Representation of Stochastic Numbers

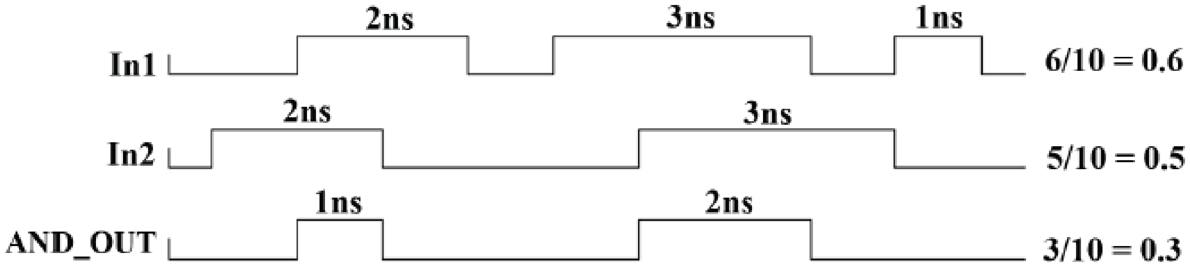
- **Digital**



$Z = X_1 \times X_2$

$3/10 = 6/10 \times 5/10$

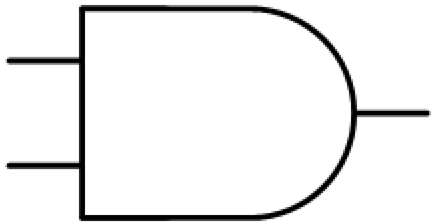
- **Analog**
  - ❑ Encoding the value as the **fraction of time the signal is high**



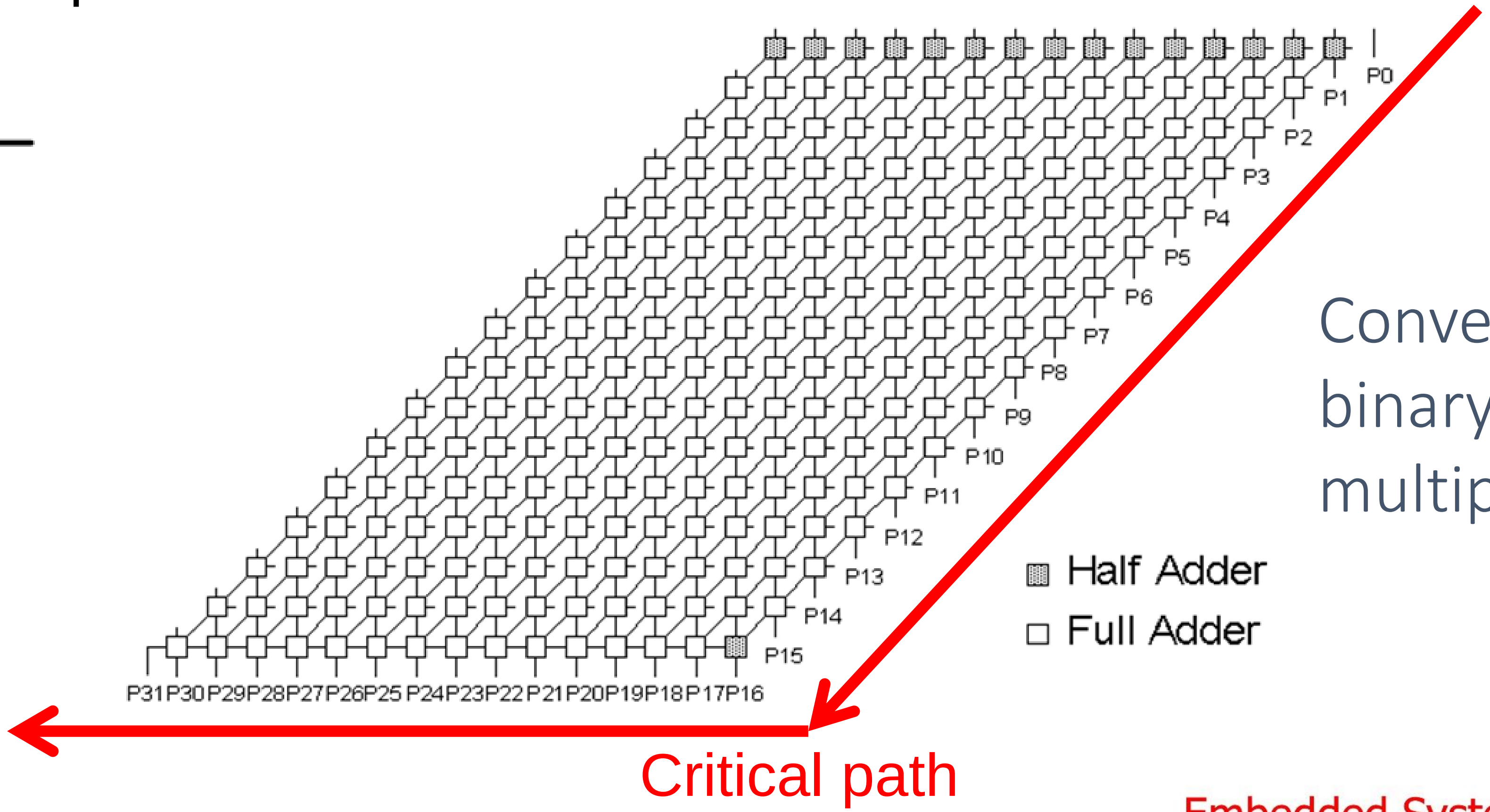
- Bit-stream length grows exponentially with precision
- Redundant representation provides error tolerance

Area, Computation Efficiency and Delay

SC: smaller area, longer computation latency, and shorter critical path



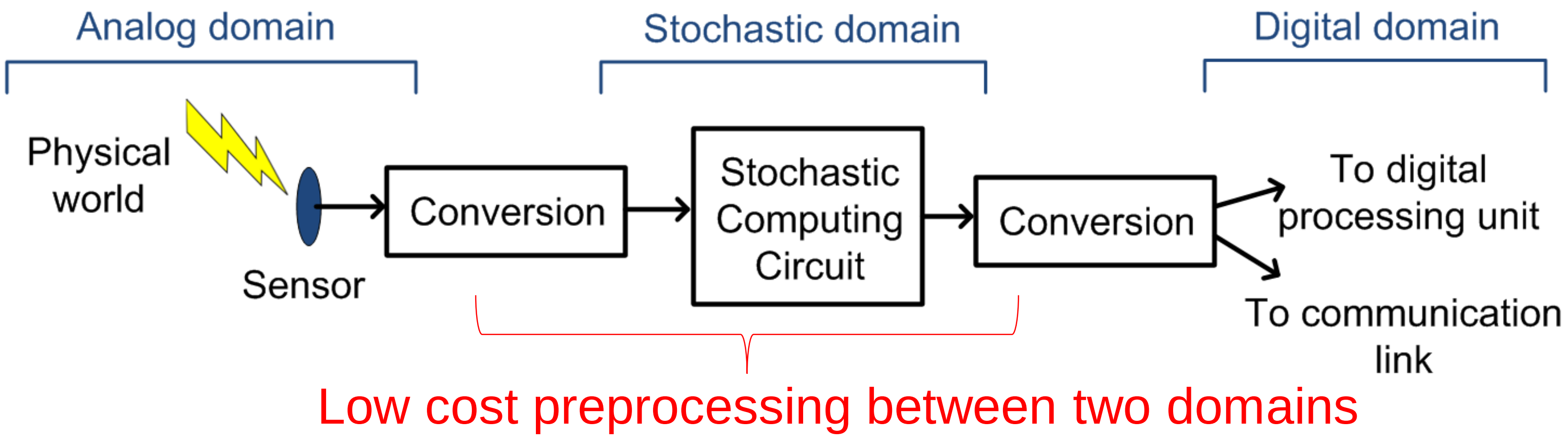
Stochastic multiplier

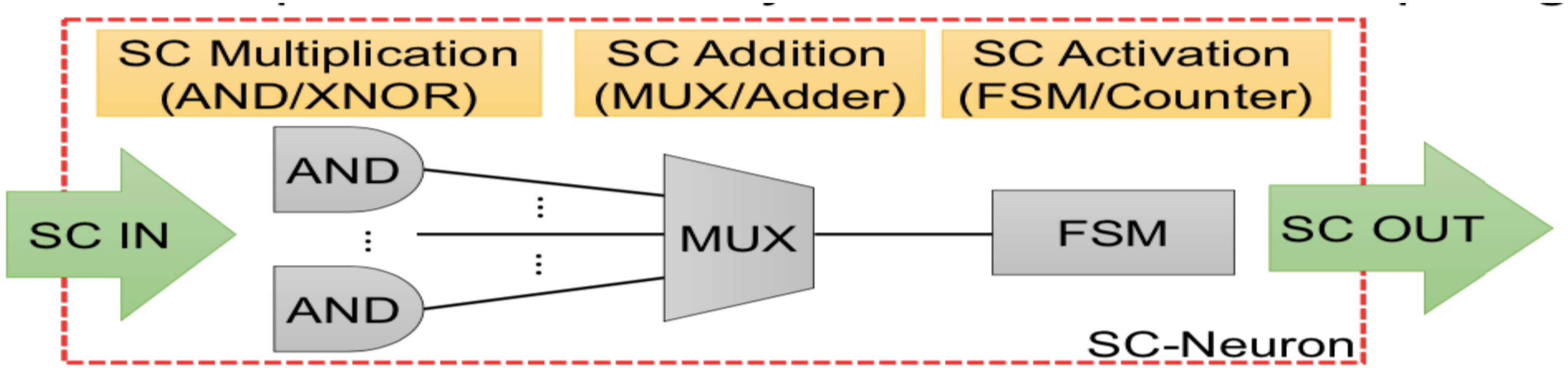


Conventional binary multiplier

Application Context of SC

- Stochastic computing circuit performs cheap pre-processing; saves resources





Embedded.Systems@UCC

Advantages and Weaknesses

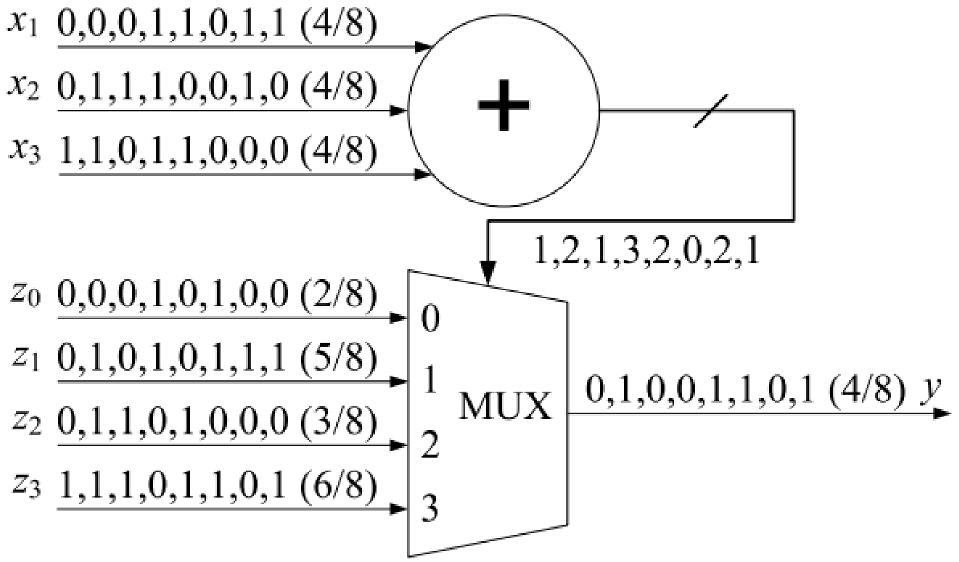
- **Advantages**
  - ❑ **Simple hardware** for complex operations
    - ❖ Multiplication: AND (Unipolar), XNOR (Bipolar)
    - ❖ Scaled Addition: MUX
  - ❑ Gracefully **tolerate noise**
    - ❖ Redundant representation provides error tolerance
    - ❖ Stochastic: 0010000011000000 (3/16) => 4/16
    - ❖ Binary: 0.0011 = 0.1875 => 0.1011 = 0.68
  - ❑ **Skew tolerance**

- **Main Weakness**
  - ❑ High accuracy ⬄ Long stochastic streams
  - ❑ Long computation time ➔ high energy consumption
    - ❖ Much slower
    - ❖ More energy consumption than conventional binary design

Previous Works

- **Bernstein Polynomial**
  - ❑ A function $f(x) \in [0,1]$ given $x \in [0,1]$ can be implemented using Bernstein polynomial method.

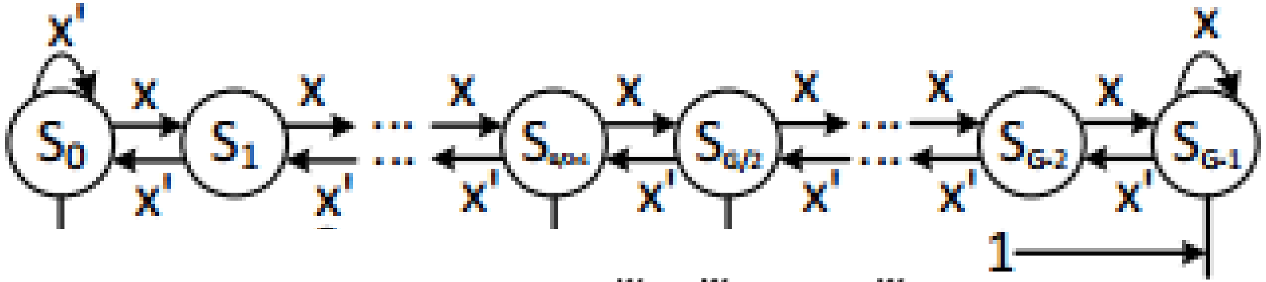
    The target function: $f(x) = \sum_{i=0}^{n} \beta_i B_{i,n}(x)$ where $B_{i,n}(x) = \binom{n}{i} x^i (1-x)^{n-i}$
  - ❑ Increasing hardware complexity as $x_i's$ and $z_i's$ required SNGs.

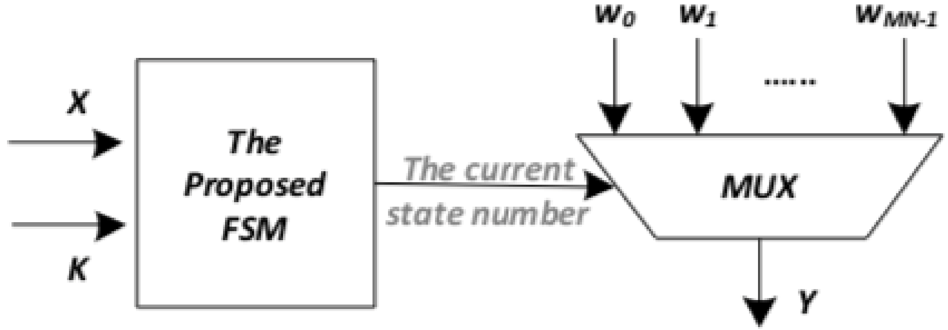


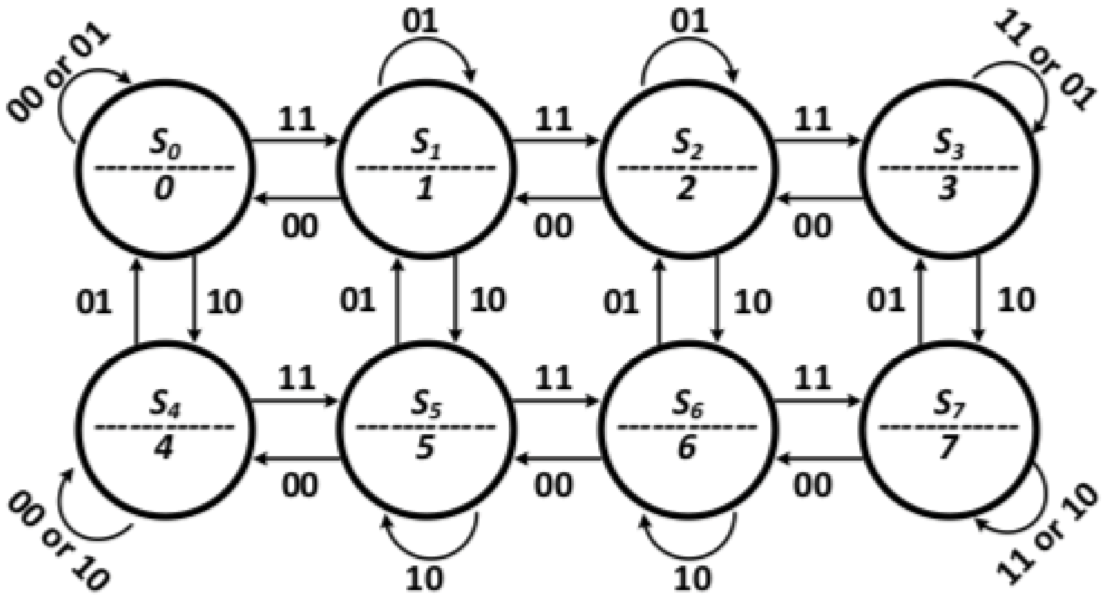
$$f_1(x) = \frac{2}{8}B_{0,3}(x) + \frac{5}{8}B_{1,3}(x) + \frac{3}{8}B_{2,3}(x) + \frac{6}{8}B_{3,3}(x)$$

Previous Works

- **Finite-state-machine based approach**
  - The method was proposed by Brown and Card using to implement tangent hyperbolic and exponential functions
  - The linear FSM topology cannot be used to synthesize more sophisticated functions [1].
  - Extra inputs to synthesize more sophisticated functions increase hardware complexity







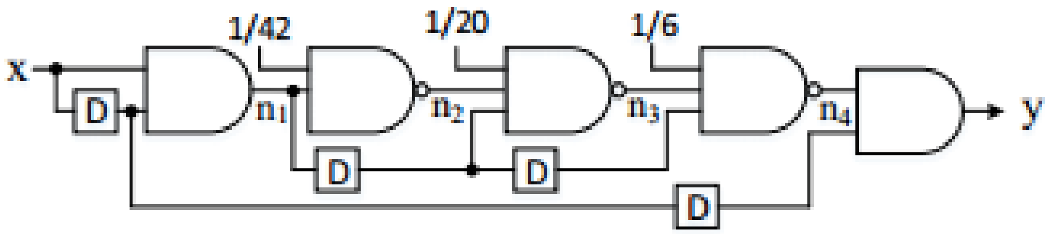
MCCI Microelectronic Circuits Centre Ireland

[1] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. Riedel, "The synthesis of complex arithmetic computation on stochastic bit streams using sequential logic," in Proceedings of the International Conference on Computer-Aided Design, pp. 480–487, ACM, 2012

Previous Works

- **Maclaurin based approach**
  - ❑ Complex arithmetic functions were implemented by using Horner’s rule for Maclaurin expansions and factorization is considered in some arithmetic functions .
  - ❑ This approach is suited for low power application [1]

$$sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$$
$$= x(1 - \frac{x^2}{6}(1 - \frac{x^2}{20}(1 - \frac{x^2}{42})))$$



- AND gate is used to implement SC multiplication.
- NOT gate is used to implement (1-x)
- One bit delay and AND gate is used to implement x^2

[1] K. Parhi and Y. Liu, "Computing Arithmetic Functions Using Stochastic Logic by Series Expansion," IEEE Trans. on Emerging Topics in Computing, pp. 1-13, Oct. 2016.

Proposed Approach

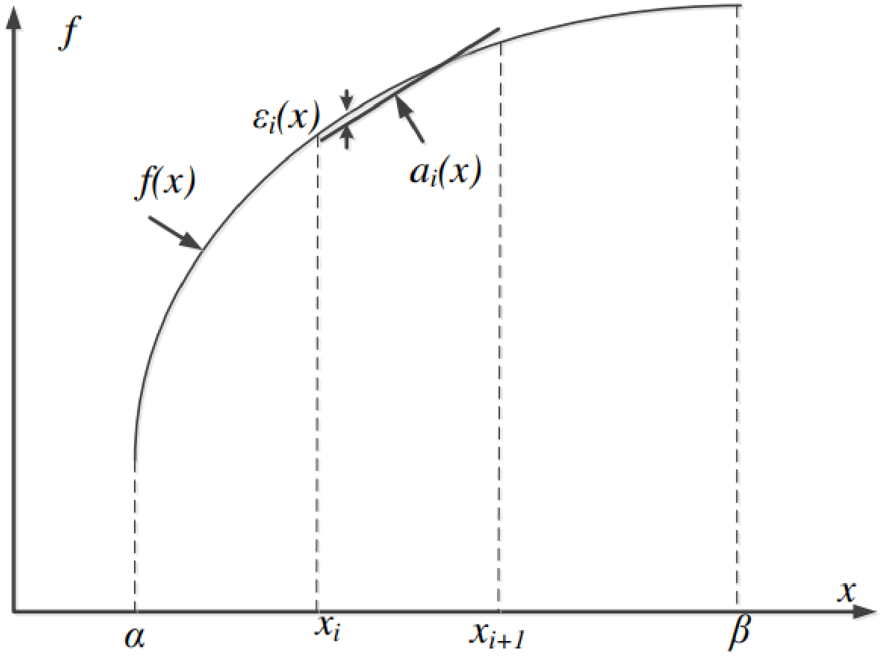
- **Piecewise linear approximation**
  - A complex arithmetic $f(x)$ is approximated by segments.
  - The domain of $x \in (\alpha, \beta)$ could be divided into s equal segments.
  - In the i^{th} segment, the function f(x) can be written as:

$$f(x) \approx a_i x + b_i, \qquad \frac{i}{s}(\beta - \alpha) \le x \le \frac{i+1}{s}(\beta - \alpha)$$

$$i = 0 \to s - 1$$

  - The error in i^{th} segment:

$$\varepsilon_i = f(x) - (a_i x + b_i)$$



Proposed Approach

- **Lagrange interpolation approximation**

  ❑ The optimized coefficients a_i, b_i in each segment can be found by using Lagrange interpolation approximation using Chebyshev nodes.

$$f(x) = \sum_{i=0}^{n} L_i(x)$$

$$L_i(x) = f(x_i). \prod_{j=0, i \neq j}^{n} \frac{x - x_i}{x_i - x_j}$$

  ❑ Fitting points on $f(x)$ to find the optimal polynomial.

$$c_0 = cos\left(\frac{\pi}{2n+2}\right), \ldots, c_n = cos\left(\frac{(2n+1)\pi}{2n+2}\right)$$

  ❑ **The shorter** the approximation interval, **the closer** to linear the function => lower degree polynomial => decrease hardware complexity

Hardware Architecture

- The Hardware Designs of f(x)= $e^{-x}, cos(x)$

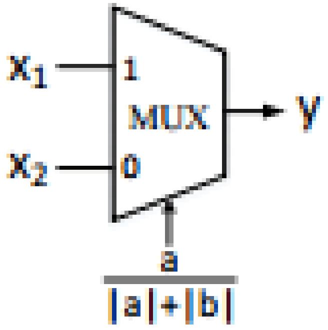
  - The function can be approximated as: $f(x) \approx -a_i x + b_i$

e^{-x}	
a_i	b_i
-962 x 2^{-10}	1005 x 2^{-10}
-849 x 2^{-10}	988 x 2^{-10}
-748 x 2^{-10}	926 x 2^{-10}
-661 x 2^{-10}	859 x 2^{-10}
-582 x 2^{-10}	773 x 2^{-10}
-517 x 2^{-10}	723 x 2^{-10}
-453 x 2^{-10}	682 x 2^{-10}
-394 x 2^{-10}	611 x 2^{-10}

$a_i < 0$

$|a_i| < |b_i|$

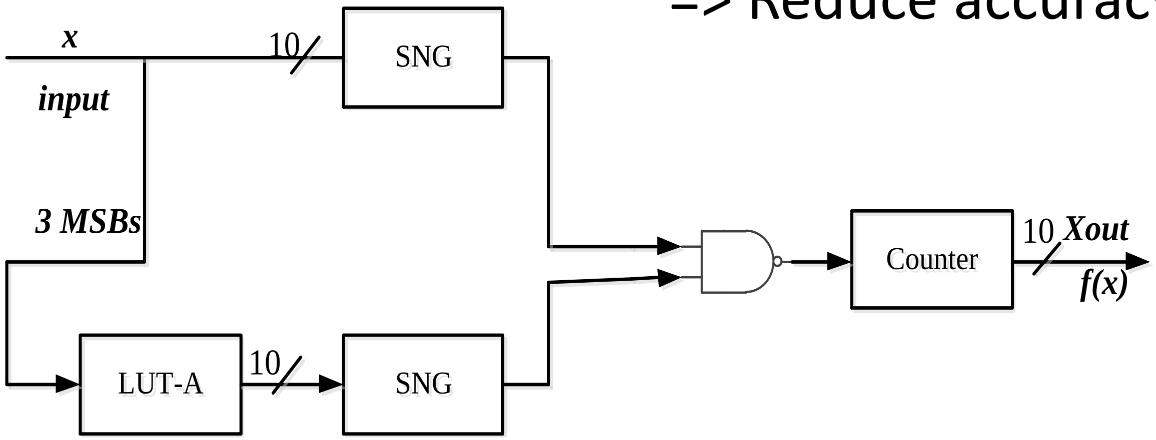
$\left|\frac{a_i}{b_i}\right| \in (0,1)$



=> Reduce accuracy

  - The function can be re-written as

$$f(x) = 1 - \frac{a_i}{b_i} x, \; i = 0:7$$



Hardware Architectures

- The Hardware Designs of f(x)= $ln(1+x),\ tanh(x),\ sigmoid(x), sin(x)$
  - ❑ The function can be approximated as: $f(x) \approx a_i x + b_i$

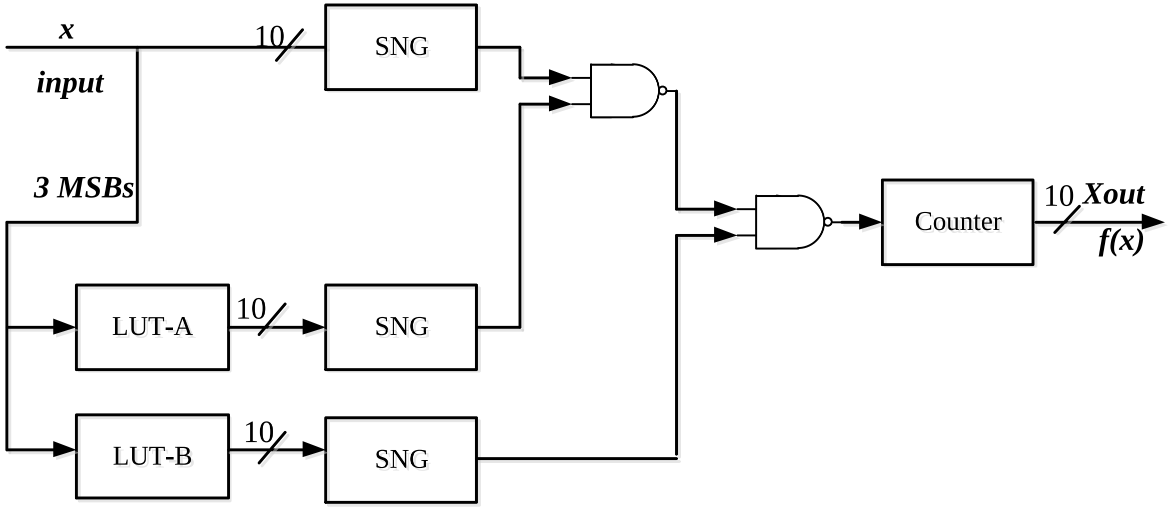
$ln(1+x)$	
a_i	b_i
964 x 2^{-10}	1 x 2^{-10}
861 x 2^{-10}	14 x 2^{-10}
780 x 2^{-10}	34 x 2^{-10}
713 x 2^{-10}	60 x 2^{-10}
655 x 2^{-10}	88 x 2^{-10}
606 x 2^{-10}	118 x 2^{-10}
565 x 2^{-10}	150 x 2^{-10}
529 x 2^{-10}	181 x 2^{-10}

$a_i > 0$

$|a_i| > |b_i|$

Avoid using addition:

$b_i = 1 - c_i$



  - ❑ The function can be re-written as:

$$f(x) = 1 - c_i + a_i x = 1 - c_i\left(1 - \frac{a_i}{c_i} x\right)$$

Hardware Architectures

- The Hardware Designs of f(x)= e^{-2x}

❑ The function can be approximated as: $f(x) \approx a_i x + b_i$

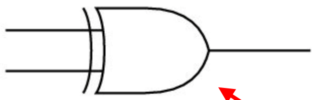
e^{-2x}	
a_i	b_i
-1809×2^{-10}	1023×2^{-10}
-1409×2^{-10}	970×2^{-10}
-1097×2^{-10}	893×2^{-10}
-855×2^{-10}	802×2^{-10}
-665×2^{-10}	708×2^{-10}
-518×2^{-10}	616×2^{-10}
-403×2^{-10}	530×2^{-10}
-314 x 2-10	452×2^{-10}

$a_i \in [-2, 0]$

$b_i \in [0,1]$

$|a_i| > |b_i|$ In first four values

$|a_i| < |b_i|$ In second four values



XOR gate for unipolar subtract

❑ Considering first four values:

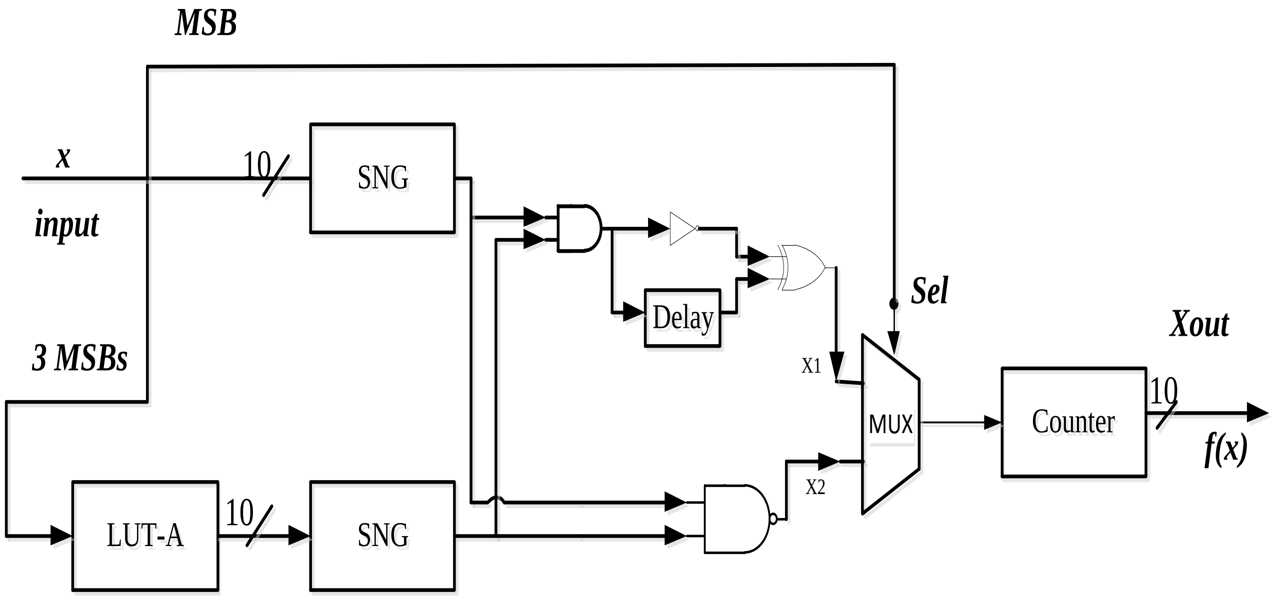
$\left|\frac{a_i}{b_i}\right| \in [1, 2]$ Cannot convert to SC => applying factorization: $f(x) = 1 - \frac{a_i}{2b_i}x - \frac{a_i}{2b_i}x$

Hardware Architectures

- The Hardware Designs of f(x)= e^{-2x}

❑Considering second four values:

$$f(x) = -a_i x + b_i = 1 - \frac{a_i}{b_i} x, \ i = 4,5,6,7$$



Simulation Results

- Accuracy

❑ The Monte Carlo simulation was used to evaluate Mean Absolute Error (MAE)

- Improvement of **2.5 times** on average comparing to Maclaurin based method
- Improvement of **8.5 times** on average comparing to FSM based method

Function		Proposed method	Horners rule	FSM -based
sin(x)	Order	-	7	8 states
	Error	0.0013	0.0034	0.0025
$\ln(1+x)$	Order	-	7	8 states
	Error	0.0026	0.0081	0.0186
$\tanh(x)$	Order	-	7	8 states
	Error	0.0012	0.0140	0.0351
$sigmoid(x)$	Order	-	7	8 states
	Error	0.0043	0.0046	0.0198

Microelectronic Circuits Centre Ireland

Embedded.Systems@UCC

Simulation Results

- Proposed hardware architectures were synthesized in 65 nm CMOS library

Function		Proposed method	Honners rule	FSM-based
$\ln(1+x)$	Total Cell Area	607	763	1269
	Power (μW)	16.73	21.42	34.79
	Delay (ns)	2.92	2.89	2.78
$\tanh(x)$	Total Cell Area	603	674	1270
	Power (μW)	16.62	18.82	35.5213
	Delay (ns)	2.97	2.89	2.71
$sigmoid(x)$	Total Cell Area	600	758	1489
	Power (μW)	16.542	21.15	41.23
	Delay (ns)	2.86	2.79	3.09

Conclusions

- We proposed an approach to customize arithmetic functions based stochastic computing in which the Mean Absolute Error is significantly improved comparing to previous methods.
- The experiment results show area and power consumption improvement over previous works

- Future Work
  - Neural Networks
  - LDPC decoders

MCCI Microelectronic Circuits Centre Ireland

Embedded.Systems@UCC

Thank you

E-mail: tieu@ue.ucc.ie

Microelectronic Circuits Centre Ireland

Embedded.Systems@UCC