IEEE Xplore® Digital Library
ISCAS 2020
2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020
CAS

Accurate and Energy-Efficient Implementation of Non-Linear Adder in Parallel Stochastic Computing Using Sorting Network

Yawen Zhang, Runsheng Wang, Yixuan Hu, Weikang Qian, Yanzhi Wang, Yuan Wang, Ru Huang

Peking University
Shanghai Jiao Tong University
Northeastern University

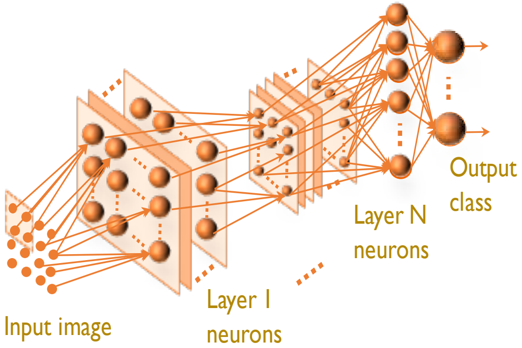
2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

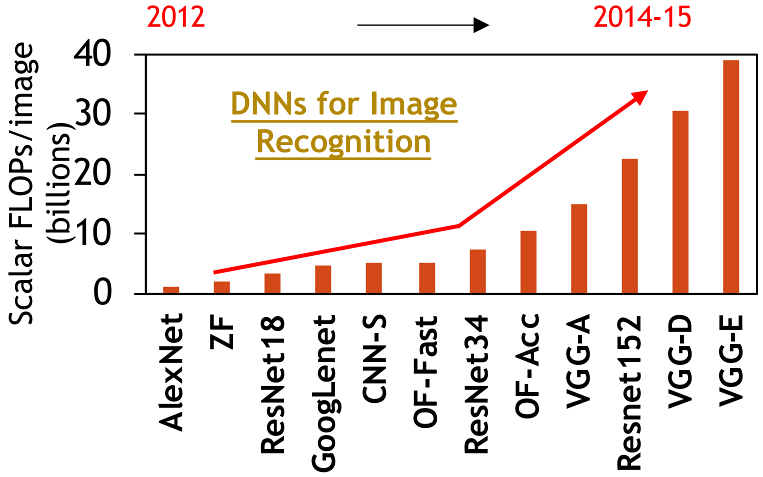
北京大學 PEKING UNIVERSITY
上海交通大學 SHANGHAI JIAO TONG UNIVERSITY
Northeastern University

Deep Neural Networks Drive Compute Demand

Deep learning achieves excellent performances on many tasks such as *images*, *videos*, *text* and *natural language*!

[Swagath Venkataramani, DAC 2018]





Challenges

- Large-scale matrix multiplication operations lead to complex hardware implementations
- Computational and memory resource requirements are becoming more stringent
  - Especially in mobile systems and the Internet of Thing (IoT) devices

Opportunity

- Quantized neural networks
  - Greatly reduce the storage and computational requirements

Low-precision neural network has relatively poor tolerance to noise!

Outline

- Background
  - Stochastic computing
  - Challenges of SC-based non-linear adder
- Parallel Non-linear Adder
- Experimental Results
- Conclusion

Stochastic Computing: What and Why?

Stochastic Computing

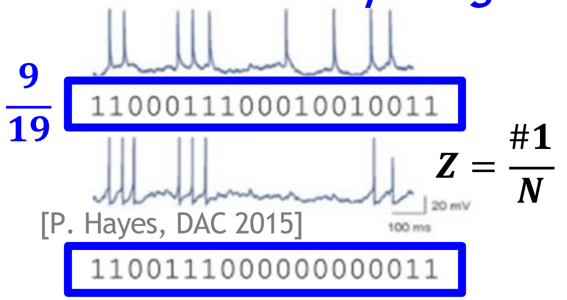
- Alternative to conventional computing
- Use random bit streams to represent operands
- Each stochastic bit stream represents a value equal to the probability of a 1 in the stream

Deterministic Computing

2^4	2^3	2^2	2^1	2^0	
16	8	4	2	1	
0	0	0	0	0	00

Binary Number Representation

Stochastic Computing

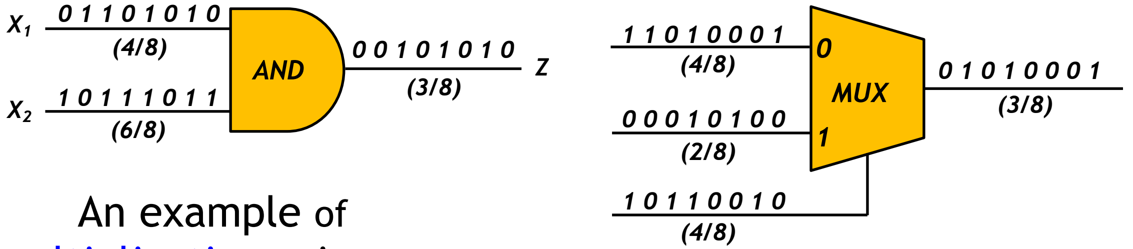


Stochastic Number Representation

SC is suitable for low-precision neural networks with poor fault tolerance

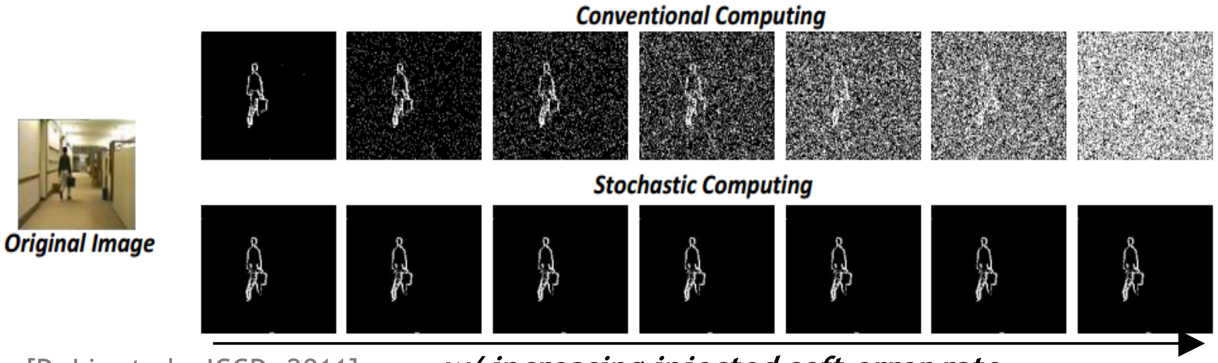
Advantages

- Complex operations performed with simple logic
- Tolerant for noise and uncertainty



An example of multiplication using an AND gate

An example of scaled addition using a MUX unit

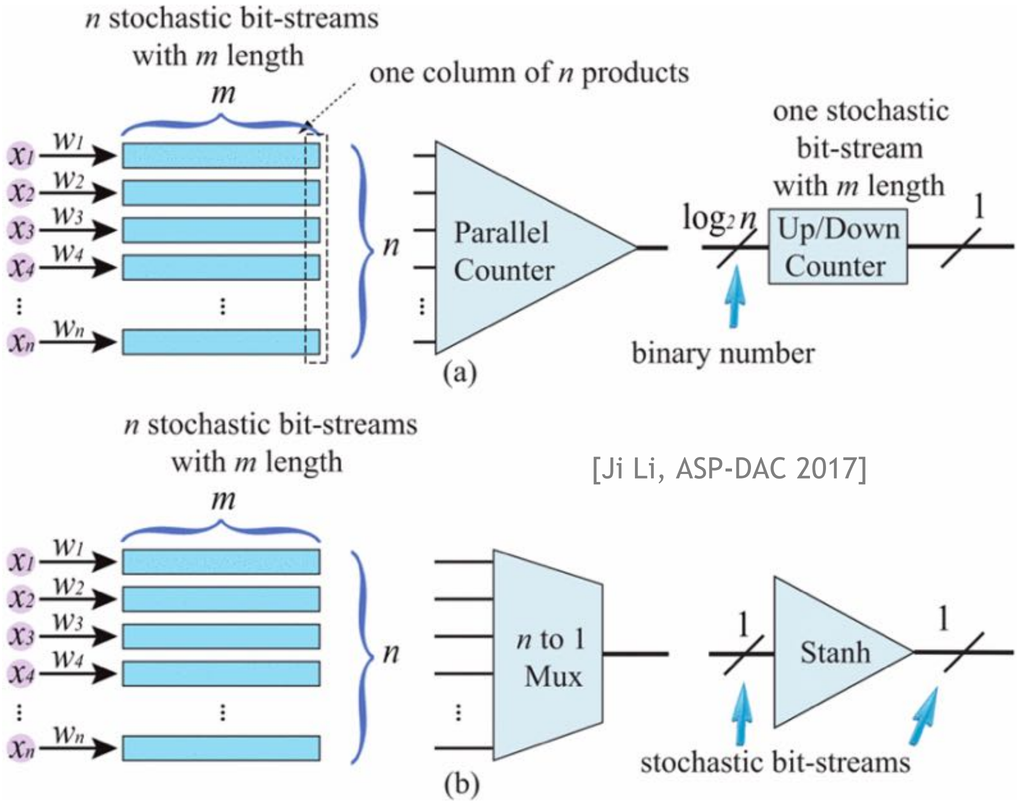


[P. Li, et al., ICCD, 2011] *w/ increasing injected soft error rate*

Traditional SC-based Nonlinear Adder

Accumulation

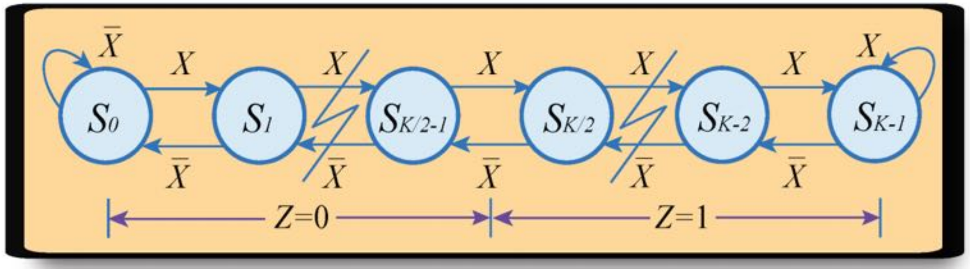
- Approximate parallel counter (APC)
- MUX-based scaled adder



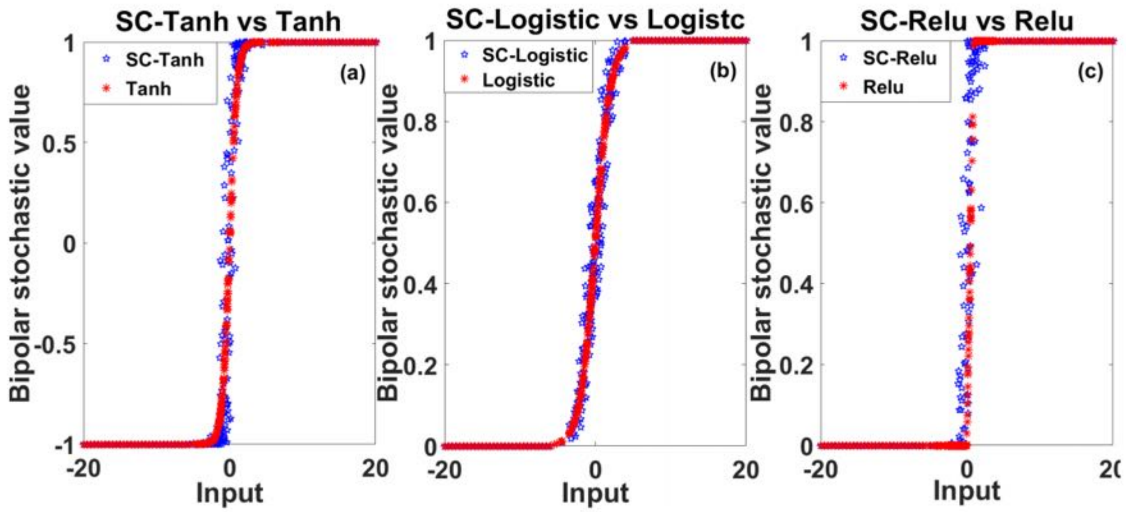
[Ji Li, ASP-DAC 2017]

Activation Function

- Up/down counter



[Ji Li, IJCNN 2017]

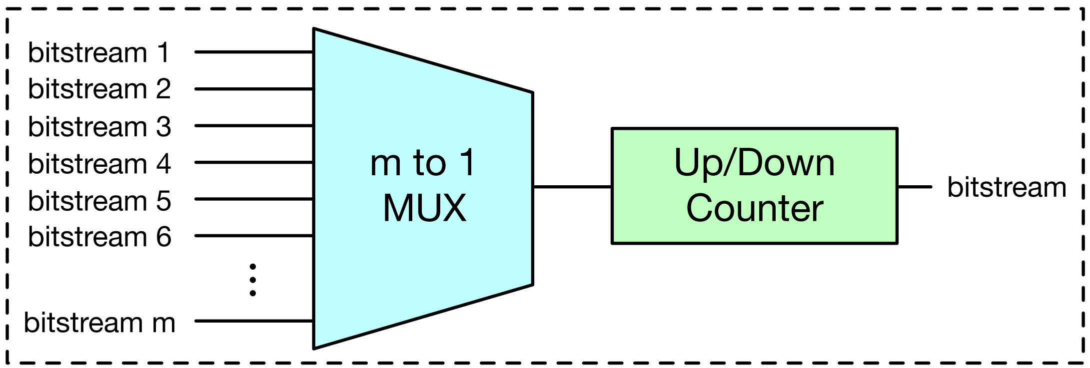


Accurately and efficiently realize the non-linear addition

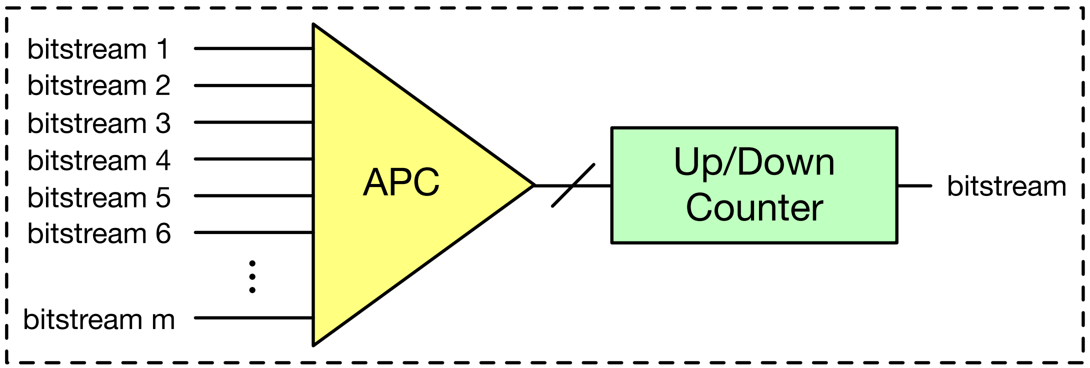
Challenges

Problem

- **MUX-based scaled scaling adder**
  - Low accuracy: randomness in the selecting signal of the MUX
  - Long latency: need long bit stream to improve the accuracy



- **APC-based non-linear scaling adder**
  - Imprecise
  - Need an additional module to convert the bitstreams into a binary number
  - Eliminate the advantages of SC including simple circuitry and high fault tolerance
  - Depend on the randomness of the input bitstream



Accurately and efficiently realize the non-linear addition

Challenges

Problem

- **APC-based non-linear scaling adder**
  - Imprecise
  - Need an additional module to convert the bitstreams into a binary number
  - Eliminate the advantages of SC including simple circuitry and high fault tolerance
  - Depend on the randomness of the input bitstream
- **MUX-based scaled scaling adder**
  - Low accuracy: randomness in the selecting signal of the MUX
  - Long latency: need long bit stream to improve the accuracy

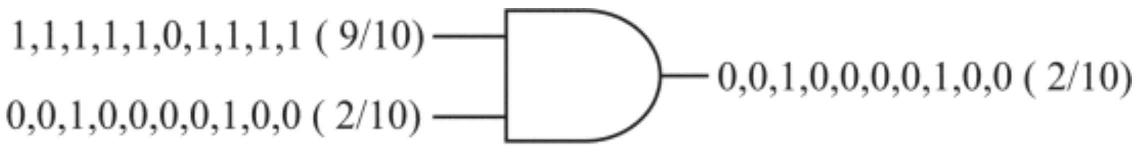
A new deterministic coding method is used to redesign the non-linear scaling addition in stochastic computing

Outline

- Background
- **Parallel Non-linear Adder**
  - **Parallel Thermometer Coding**
  - **Bitonic Sorting Network**
  - **Selective Interconnect**
- Experimental Results
- Conclusion

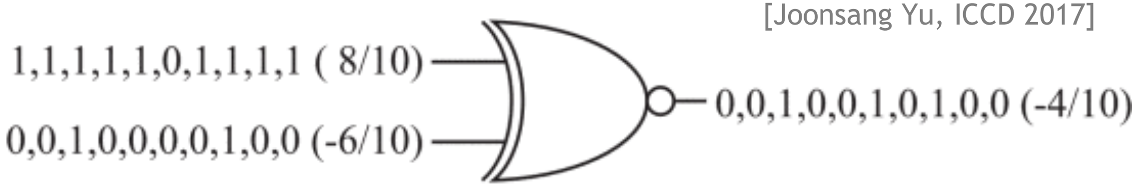
Parallel Thermometer Coding

- Stochastic bitstreams in two formats
  - Bipolar and unipolar
- Bipolar coding --> cover the negative numbers



Multiplication in unipolar format

[Joonsang Yu, ICCD 2017]



Multiplication in bipolar format

- Deterministic coding format makes the calculations accurate

$a_0\, a_1\, a_2\, a_3\, a_0\, a_1\, a_2\, a_3\, a_0\, a_1\, a_2\, a_3$

$b_0\, b_1\, b_2\, b_0\, b_1\, b_2\, b_0\, b_1\, b_2\, b_0\, b_1\, b_2$

[Davon Jenson, ICCAD 2016]

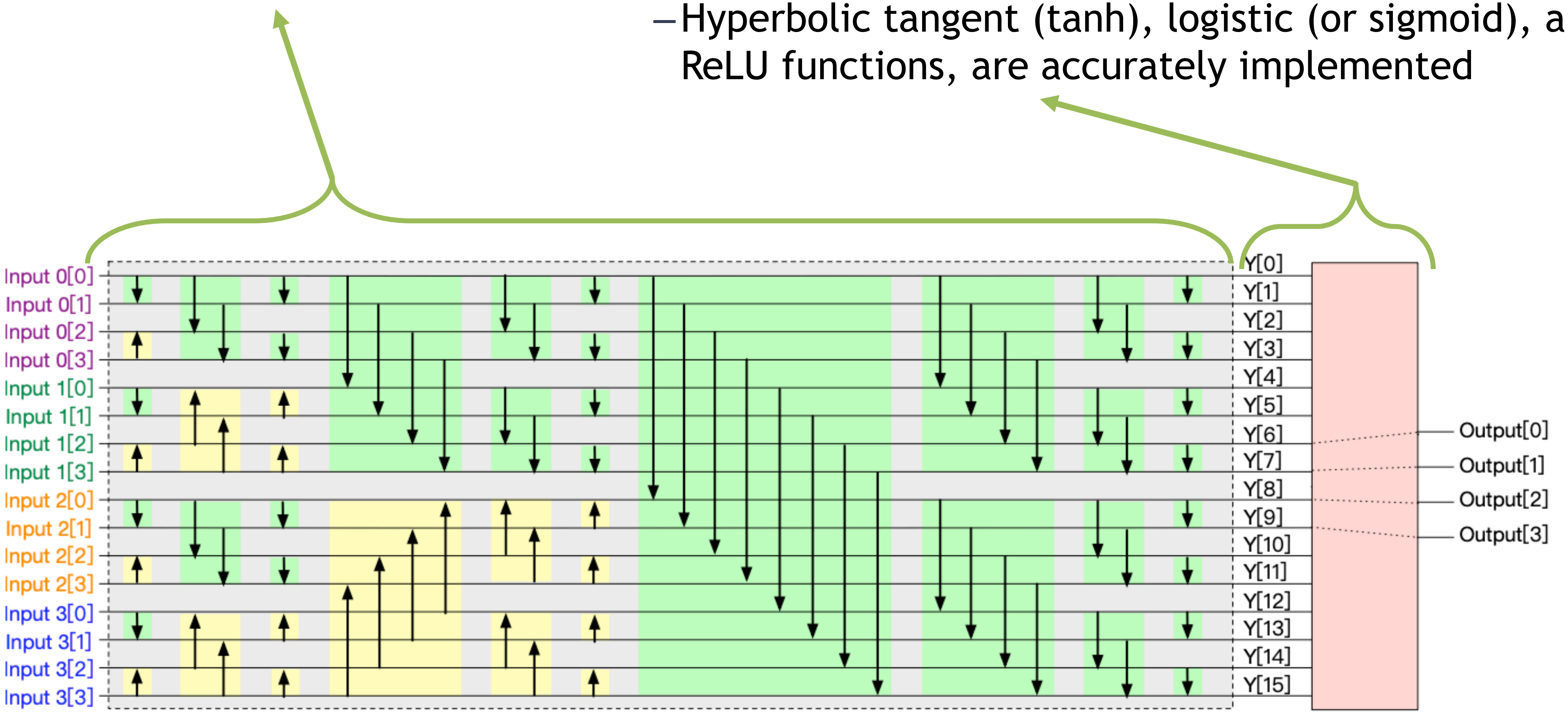
Parallel Thermometer Coding

- Thermometer coding is a type of unary coding
- Continuous sequence of 1s followed by continuous sequence of 0s
- All the bits of a stream are simultaneously input

Decimal	*Binary*			*Thermometer Coding*						
	A_0	A_1	A_2	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	1	1	1
5	1	0	1	0	0	1	1	1	1	1
6	1	1	0	0	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1

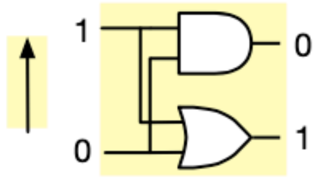
Novel Non-linear Adder: Overview

- **Bitonic sorting network**
  - Transform multiple parallel bitstreams into one
- **Selective interconnect**
  - Selecting different outputs of the sorting network, three widely-used activation functions
  - Hyperbolic tangent (tanh), logistic (or sigmoid), and ReLU functions, are accurately implemented

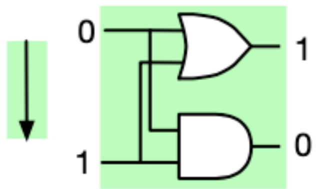


Part I: Bitonic Sorting Network

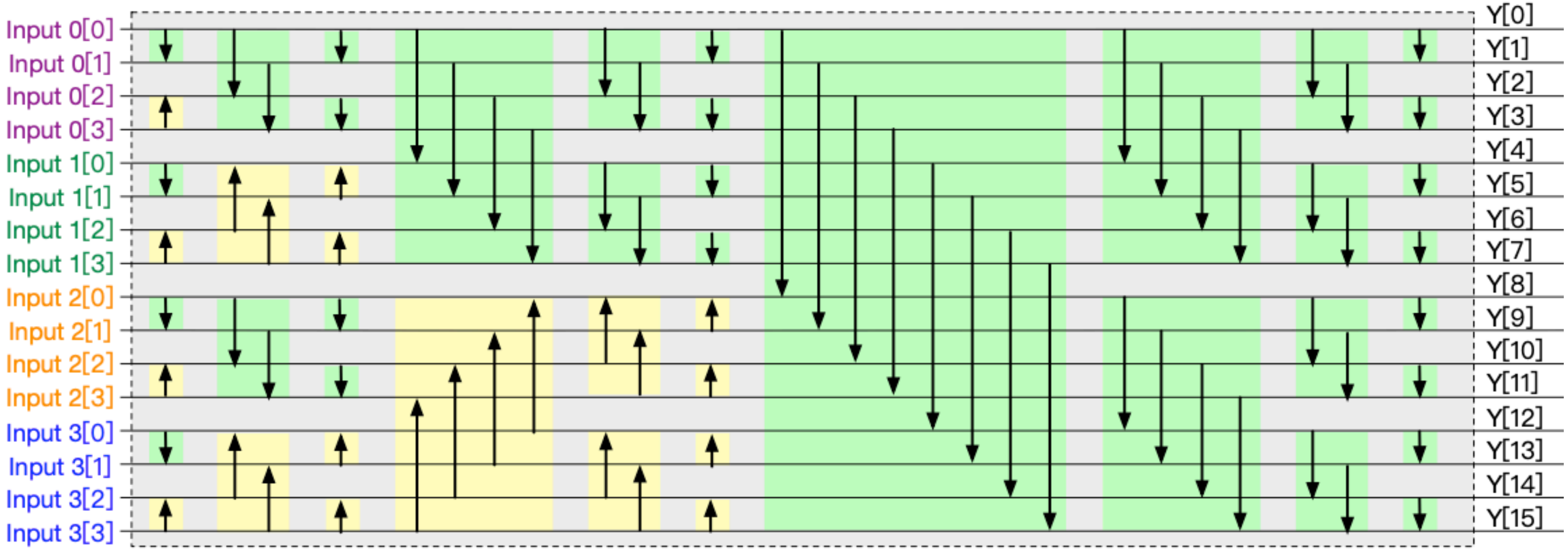
- **Arrow modules**
  - Ascending sorting module & Descending sorting module
  - Consists of an AND gate and an OR gate
- **Process**
  - Step1: An unordered sequence of length L is transformed into an ascending and a descending sequence of length L/2 by merge sorting
  - Step2: These two sequences are transformed into a monotonic sequence



Descending sorting module



Ascending sorting module



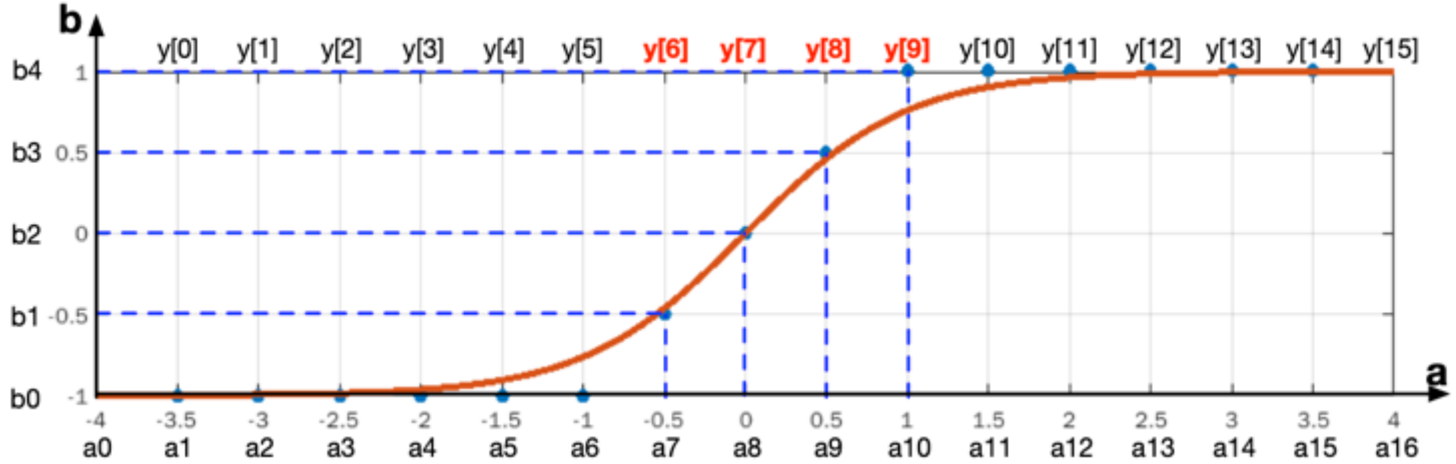
Part II: Selective Interconnect

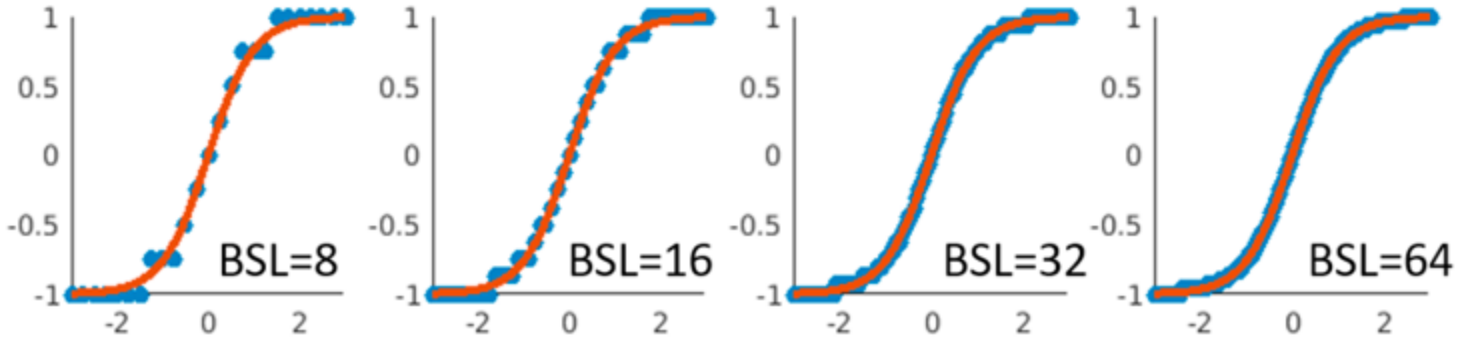
- Choose N outputs from the MN outputs y[i] of the sorting network as the final outputs to implement the non-linear activation function
- As the BSL increases, the accuracy of the proposed non-linear adder improves

Algorithm 1: Non-Linear Addition

Input: M: The number of input bitstreams;
N: The length of bitstream;
$f(x)$: Non-linear activation function (tanh, sigmoid, ReLU)
Output: *output*

```
calculate the sorting results y;
start = −M × N;
for i = 1; i < N; i++ do
    while Σ_{j=start} f′(−M + (2×j)/N) < 1 do
        j = j + 1;
    end
    end = j;
    start = end;
    output[i] = y[end];
end
return output
```



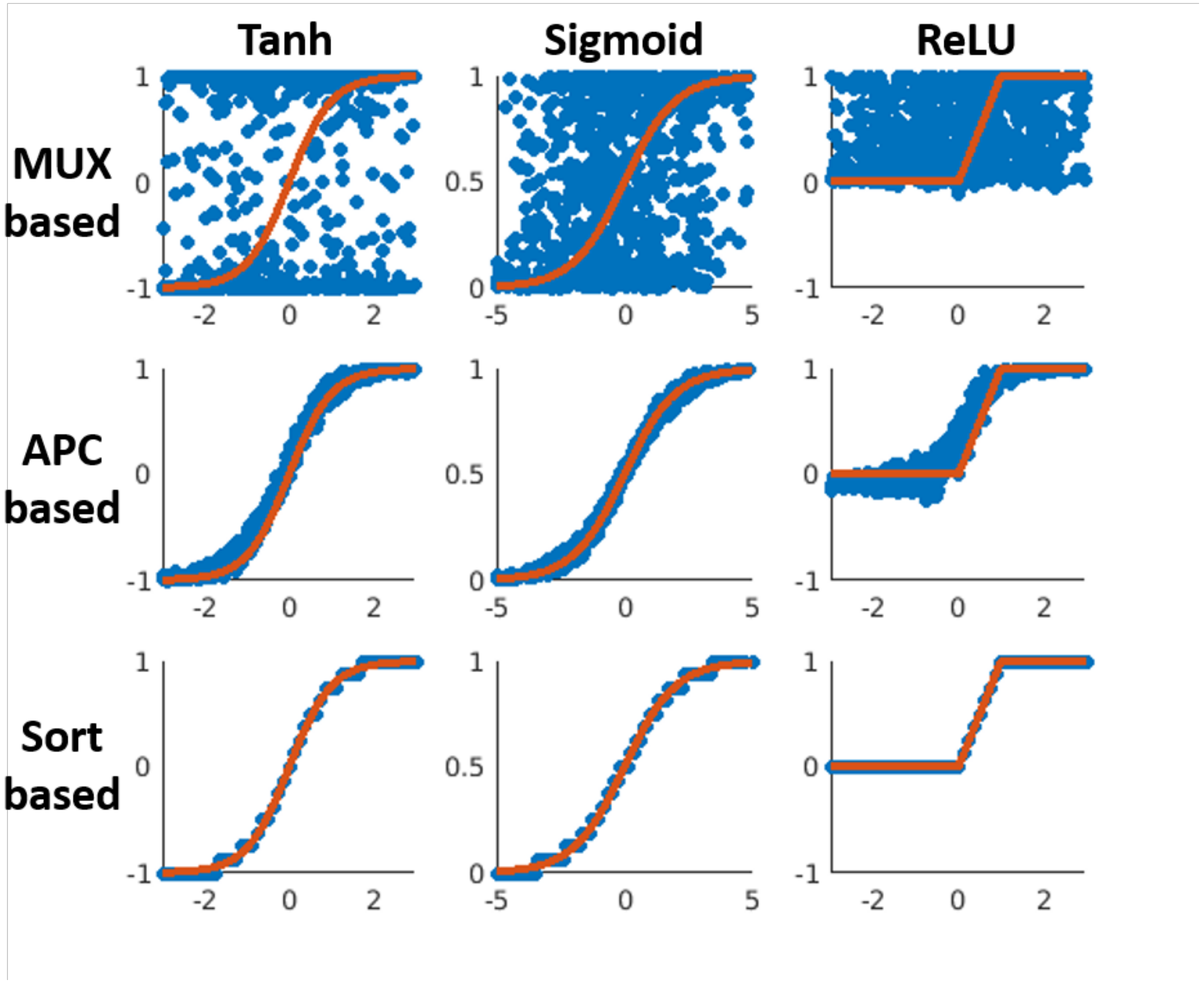


Outline

- Background
- Parallel Non-linear Adder
- **Experimental Results**
- Conclusion

Accuracy

- **MUX-based non-linear addition**
  - One of the M input bitstreams is randomly selected as the output, while the information carried by the other $M - 1$ bitstreams is lost
  - error caused by the information loss
- **APC-based non-linear addition**
  - Counter-based implementation of the non-linear activation function needs a large BSL to achieve relatively accurate results
- **The proposed design**
  - Accurate and deterministic
  - Accuracy loss is only due to the insufficient BSL



Hardware Implementation

- **Synopsys Design Compiler with TSMC 40nm technology (100MHz)**
- **The proposed design**
  - Reduce the variance by more than three orders of magnitude and improves the accuracy
  - Significantly reduce the latency from BSL clock cycles to only one cycle
  - With the BSL of 8 achieves at least 44.5× and 87.4× energy improvement over the MUX-based one and the APC-based one, respectively

	Non-Linear Function	Variance (%)	Area (μm^2)	Power (μW)	Latency/Operation (μs)	Energy/Operation (fJ)
MUX-based (1024 BSL)	Tanh	105.45	135.65	18.7	10.24	191.5
	Sigmoid	17.65	131.41	18.9	10.24	193.5
	ReLU	22.51	115.01	11.3	10.24	115.7
APC-based (1024 BSL)	Tanh	0.35	261.25	25.4	10.24	260.1
	Sigmoid	0.5	106.37	18.7	10.24	191.5
	ReLU	1.78	253.31	22.2	10.24	227.3
This work (16 BSL)	Tanh	0.08	5607.58	701	0.01	7.0
	Sigmoid	0.04	5602.11	609	0.01	6.1
	ReLU	0	5354.62	693	0.01	6.9
This work (8 BSL)	Tanh	0.29	2082.93	250	0.01	2.5
	Sigmoid	0.13	2009.73	210	0.01	2.1
	ReLU	0	1981.15	258	0.01	2.6

Outline

- Background
- Parallel Non-linear Adder
- Experimental Results
- Conclusion

Conclusion

- We propose a new non-linear adder based on parallel stochastic computing
  - Bitonic sorting network: transform multiple parallel bitstreams into one
  - Selective interconnect: implement activation functions
- Advantages
  - Accurate and deterministic with only rounding errors
  - For a short BSL, such as 8 or 16, the accuracy improves by more than three orders of magnitude compared with the traditional designs
  - ReLU function has no error
  - Achieve at least 44.5× and 87.4× energy consumption improvement compared with the MUX-based one and the APC-based one, respectively.


Thank You !