Spintronics based Stochastic Computing for Efficient Bayesian Inference System

Xiaotao Jia[1], **Jianlei Yang[1]**, Zhaohao Wang[1]
Yiran Chen[2], Hai (Helen) Li[2] and Weisheng Zhao[1]
1 Beihang University
2 Duke University

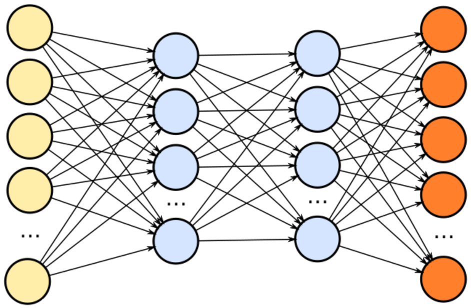
Outline

- **Background and Motivation**
- **Proposed Bayesian Inference System**
  - Spin-based stochastic bit-stream generator
  - Bayesian inference system: case studies
- **Conclusions**

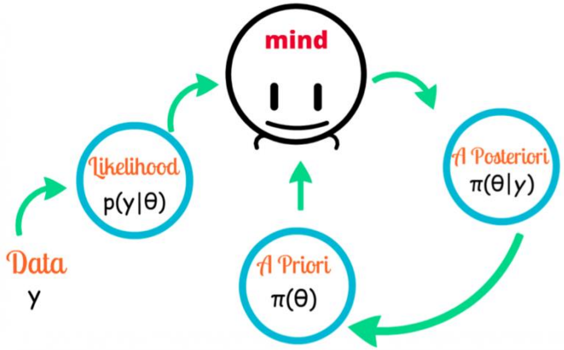
Outline

- **Background and Motivation**
- **Proposed Bayesian Inference System**
  - Spin-based stochastic bit-stream generator
  - Bayesian inference system: case studies
- **Conclusions**

Why Bayesian Inference?



- **Deep learning is everywhere**
  - But have some disadvantages:
    - Could not represent the uncertainty
    - Could not take the advantages of well-studied experience and theories
    - Require large scale training data
    - Overfitting!



- **Bayesian learning**
    - Could capture the uncertainties well
    - Could represent the casual relationships
    - More robust, closer to human mind and thinking

http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier

Bayesian Inference

- **Bayes theorem: probabilistic computing**

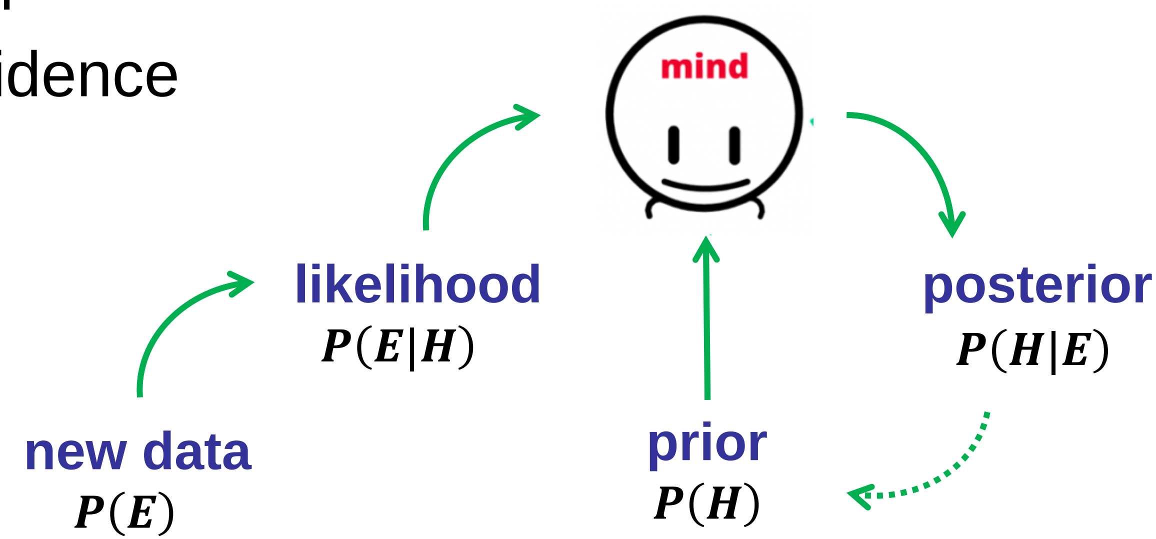
posterior probability   likelihood function   prior probability

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$

model evidence

$\boldsymbol{H}$: hypothesis

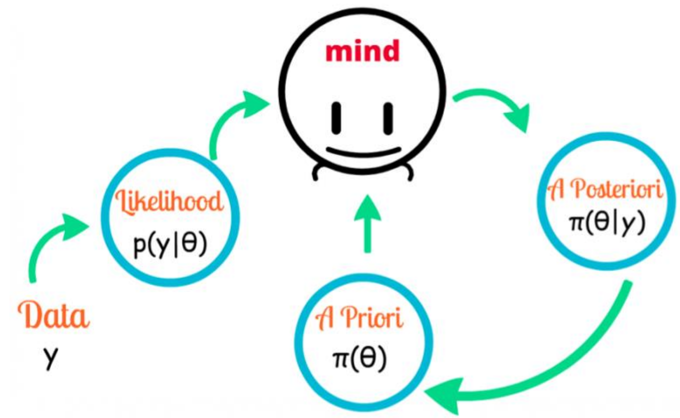
$\boldsymbol{E}$: evidence



Bayesian Learning Challenges

- **Computation intensively**
  - Kernel: probabilistic multiplication

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$



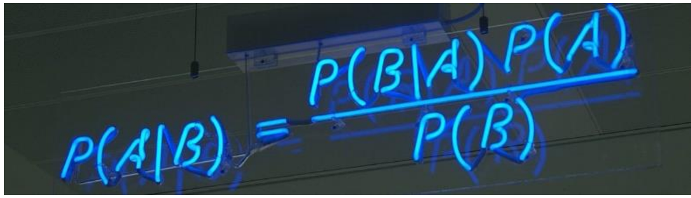
- **Challenges and opportunities**
  - Bayesian inference on FPGA [Lin, FPGA'2010]
  - Bayesian by analog CMOS [Mroszczyk, ISCAS'2014]
  - Bottlenecks of computation
    - **Float point multiplication**
    - **Random number generation**

power | **area** | **speed**

  - How to improve the efficiency of Bayesian inference?

Improving Inference Efficiency

- **Bottlenecks of computation**
  - Float point multiplication
  - Random number generation

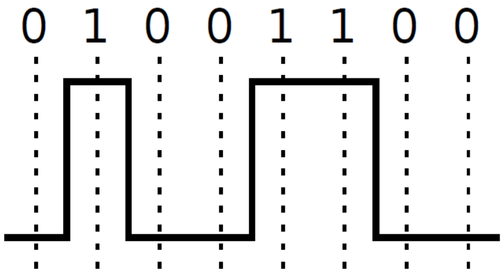


- **Our solution**
  - Stochastic computing for FP multiplications
  - Efficient random number generator by emerging spintronics device and circuit

TARGET

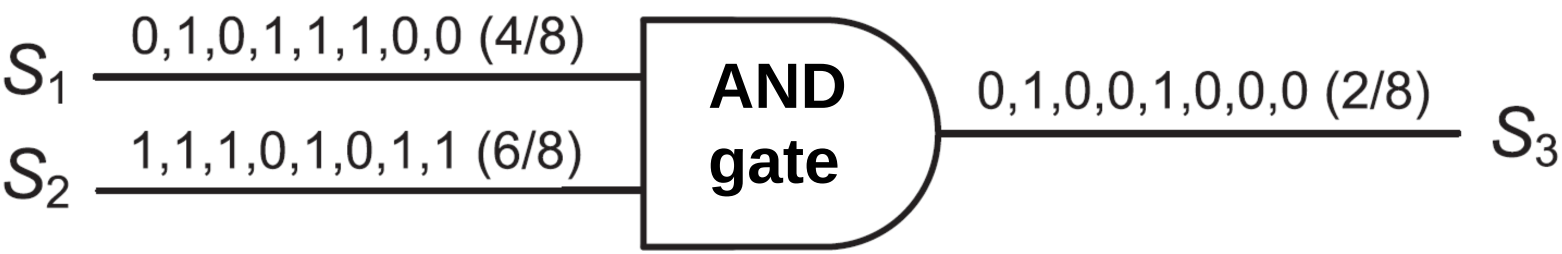
exploiting **non-conventional computing**
with **emerging technologies**
for efficient Bayesian learning

Stochastic Computing



- **Basic concepts**
  - FP numbers are represented by random bit-streams
  - By the ratio of ‘1’: 5/8 (01101101, 10111001, 10101011)
  - Complex computations could be realized by simple bit-wise operations on the bit-streams
    - **AND** for multiplication **MUX** for scaled addition
- **Stochastic multiplier**

input random bit-streams output random bit-streams



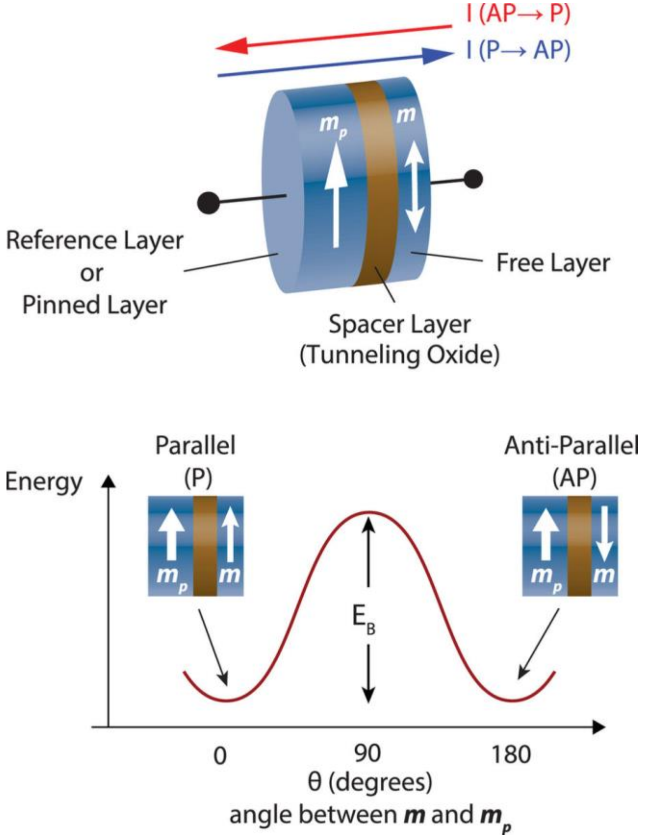
$4/8 * 6/8 \approx 2/8$

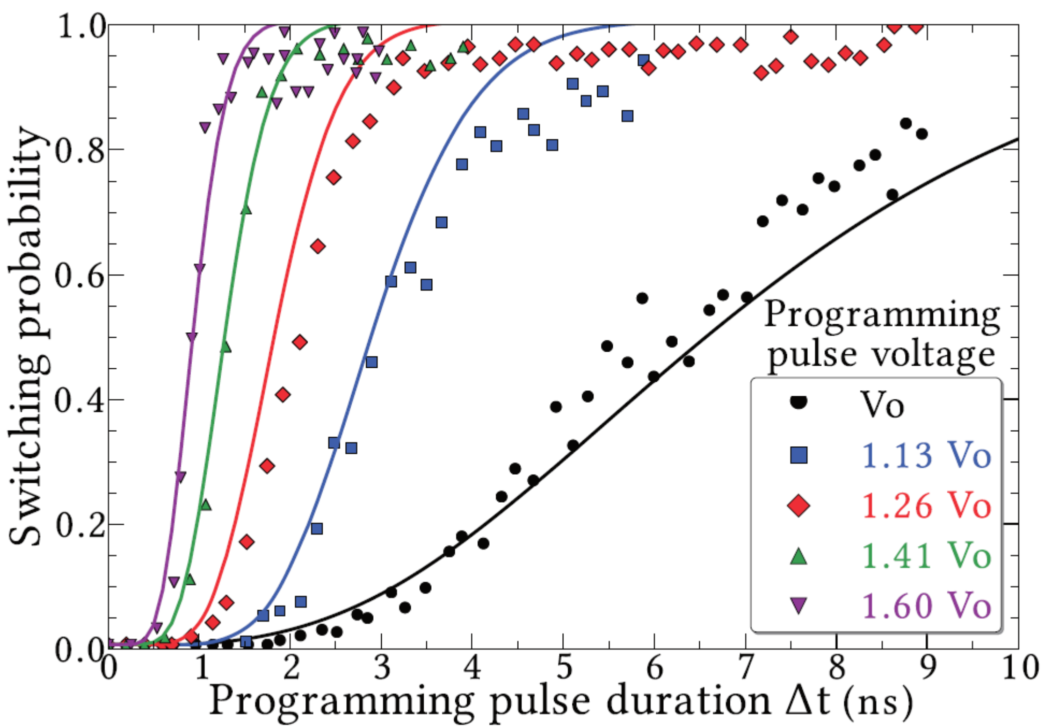
- Simple arithmetic operation
- Low cost, low latency, low power
- Error tolerance

Q: RNG? Comparator?

Randomness Representation

- **Magnetic tunnel junction (MTJ)**
  - For memory use: **deterministic** switching
  - For randomness: **stochastic** switching





Inherent randomness!

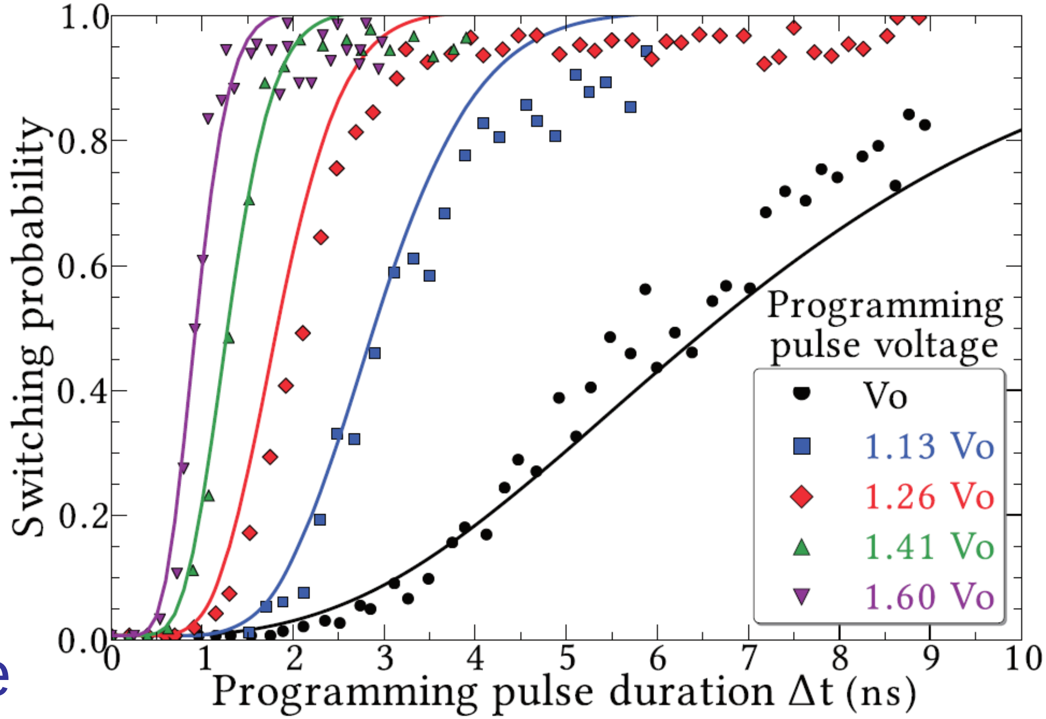
Outline

- **Background and Motivation**
- **Proposed Bayesian Inference System**
    - Spin-based stochastic bit-stream generator
    - Bayesian inference system and case studies
- **Conclusions**

MTJ Stochastic Switching

- **MTJ states**
  - High resistance (AP) or low resistance (P)
- **Stochastic behaviors**
  - Applying bias voltage/current for switching

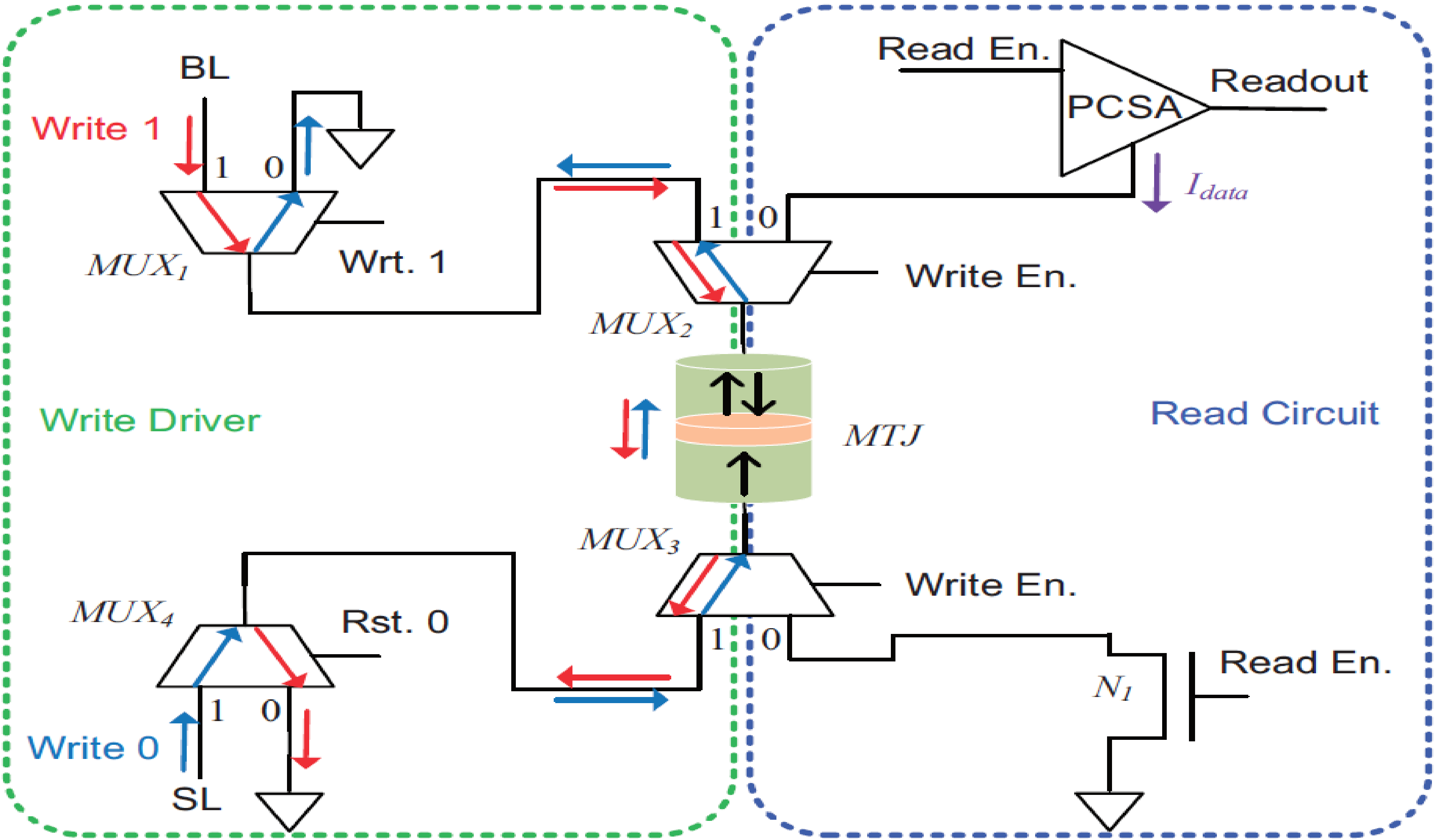
Scaled probabilities for representing FP numbers



varying bias or duration time

Stochastic Bit-stream Generator (SBG)

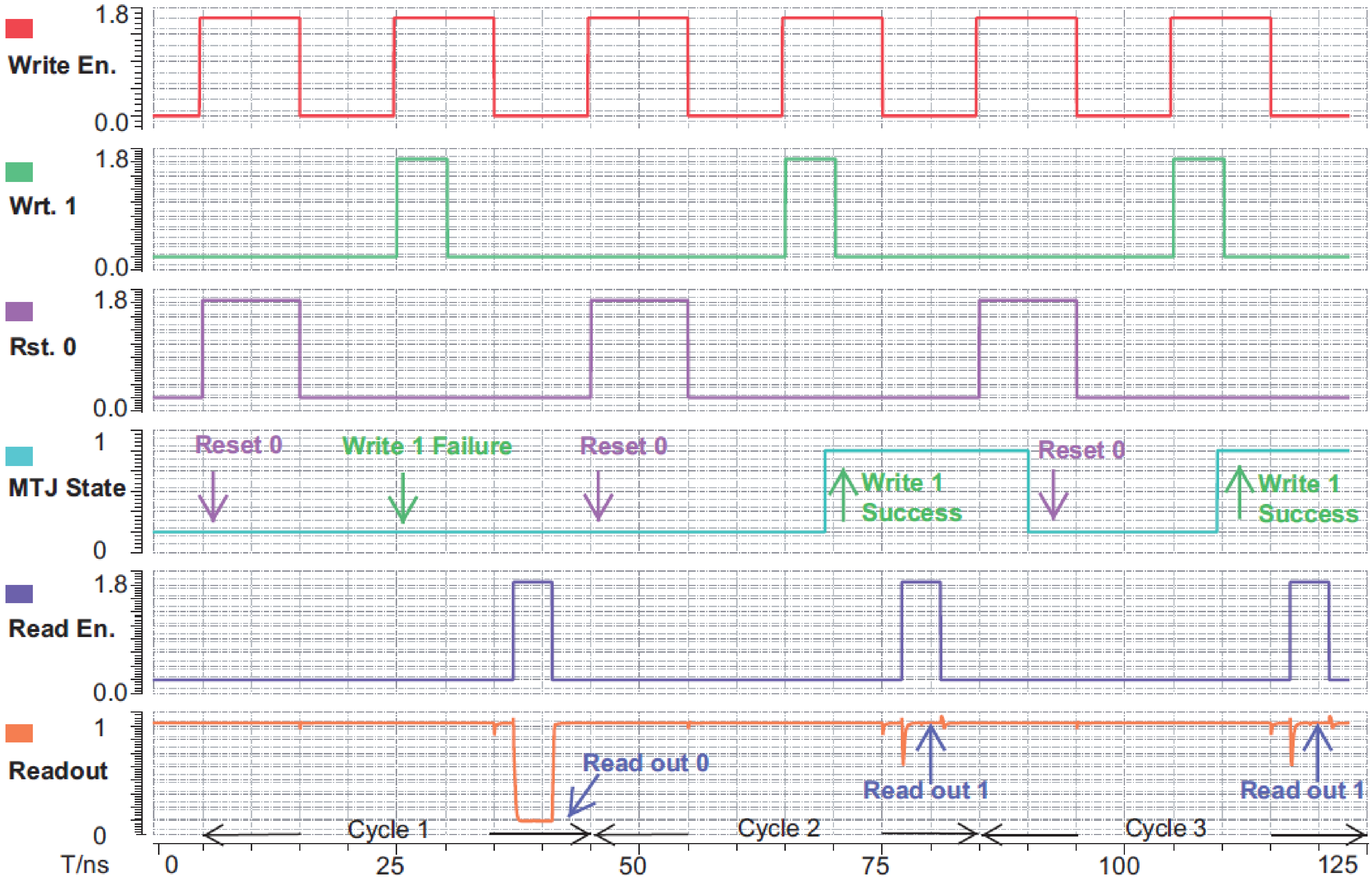
bias voltage is applied between BL and SL



MTJ/CMOS Hybrid Design

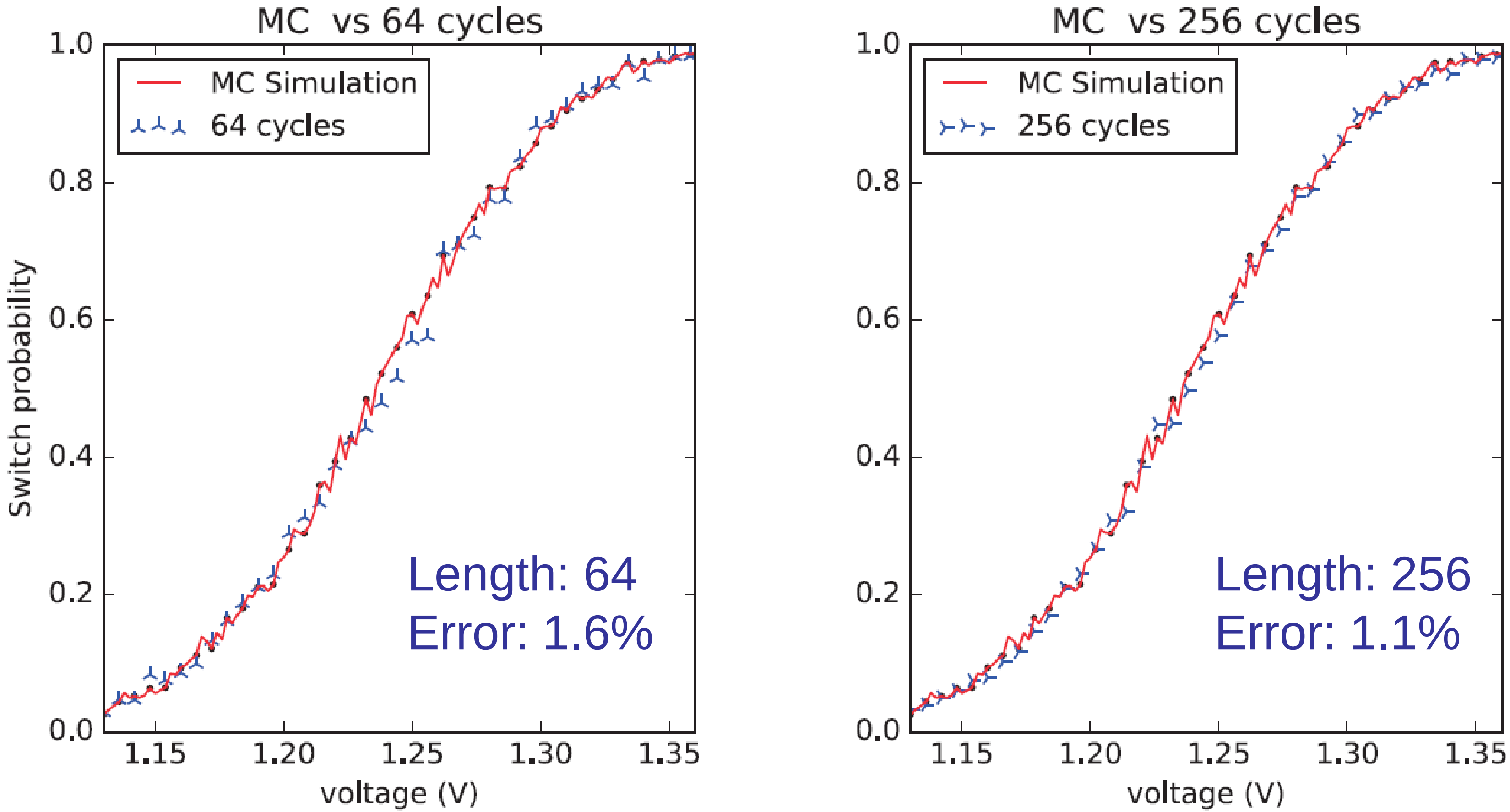
- **Design setup**
  - PDK: 45 nm CMOS and 40 nm MTJ fab.
- **Simulations**
  - Fixed duration time
  - Varying bias voltage (represents probabilities)
  - AP->P (write ‘1’) duration time: 5ns
  - P->AP (reset, write ‘0’) duration time: 10ns
  - Each cycle generates one random bit
    - reset (write ‘0’) first
    - then write ‘1’ (throw the dice)
    - read out (check the MTJ state)

write ‘1’ and read out for each cycle



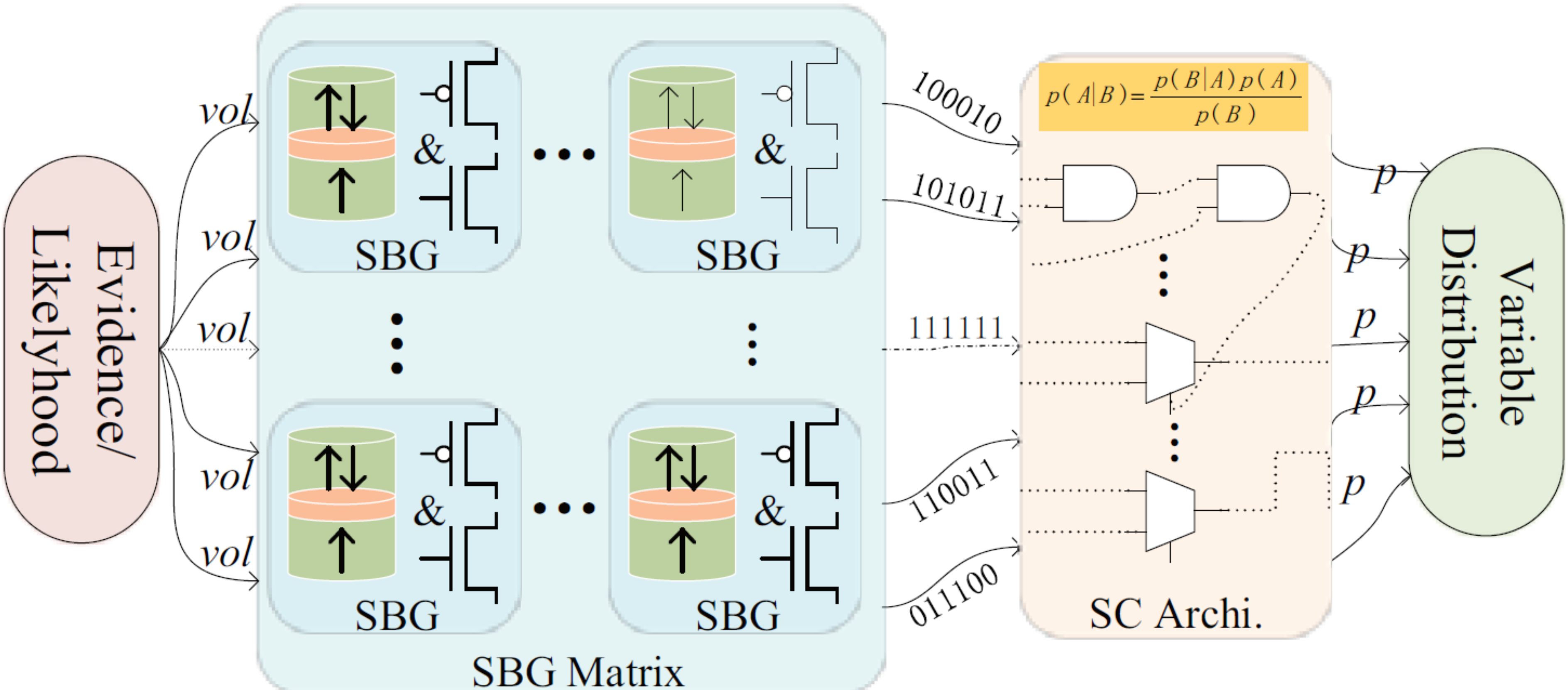
Switching Probabilities

- **Accuracy** (compared with MC simulation)
  - Improved by increasing the stream length/cycles



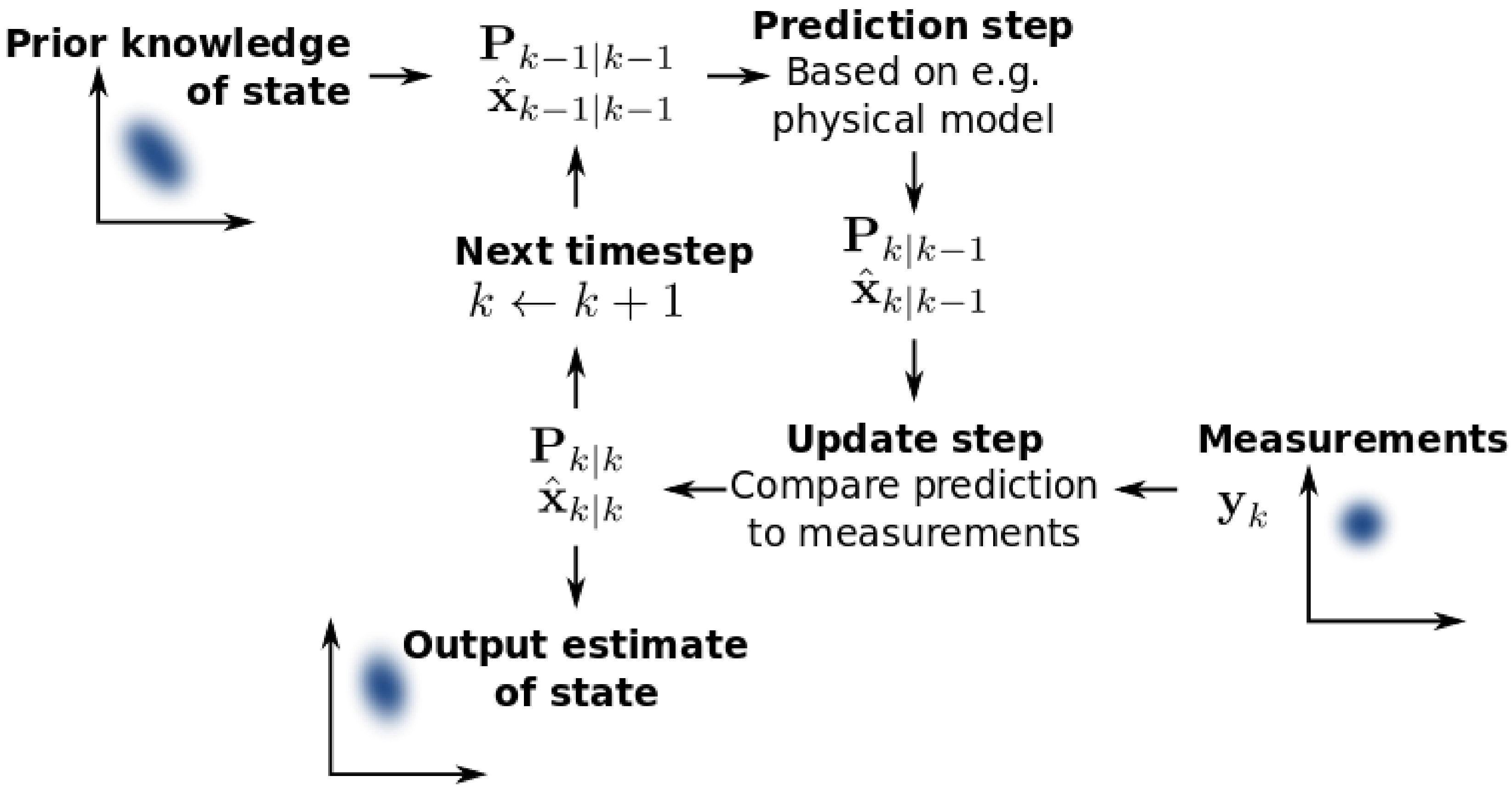
Inference System Diagram

- **Architectures: SBG and SC**
  - Input: evidence and likelihood function
  - Output: variable distribution



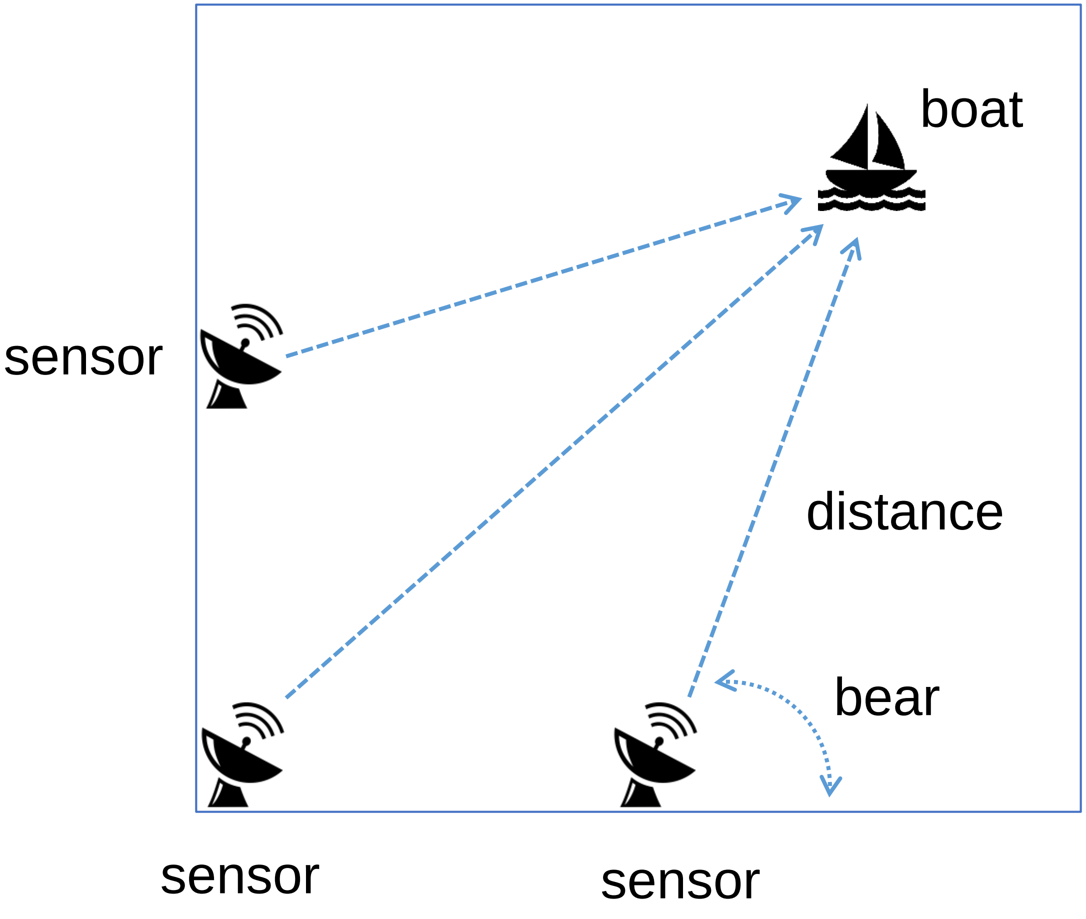
Case Study: Sensor Fusion

- **Representation of a Kalman filter**



Case Study: Sensor Fusion

- **Example: locating a target with 3 sensors**

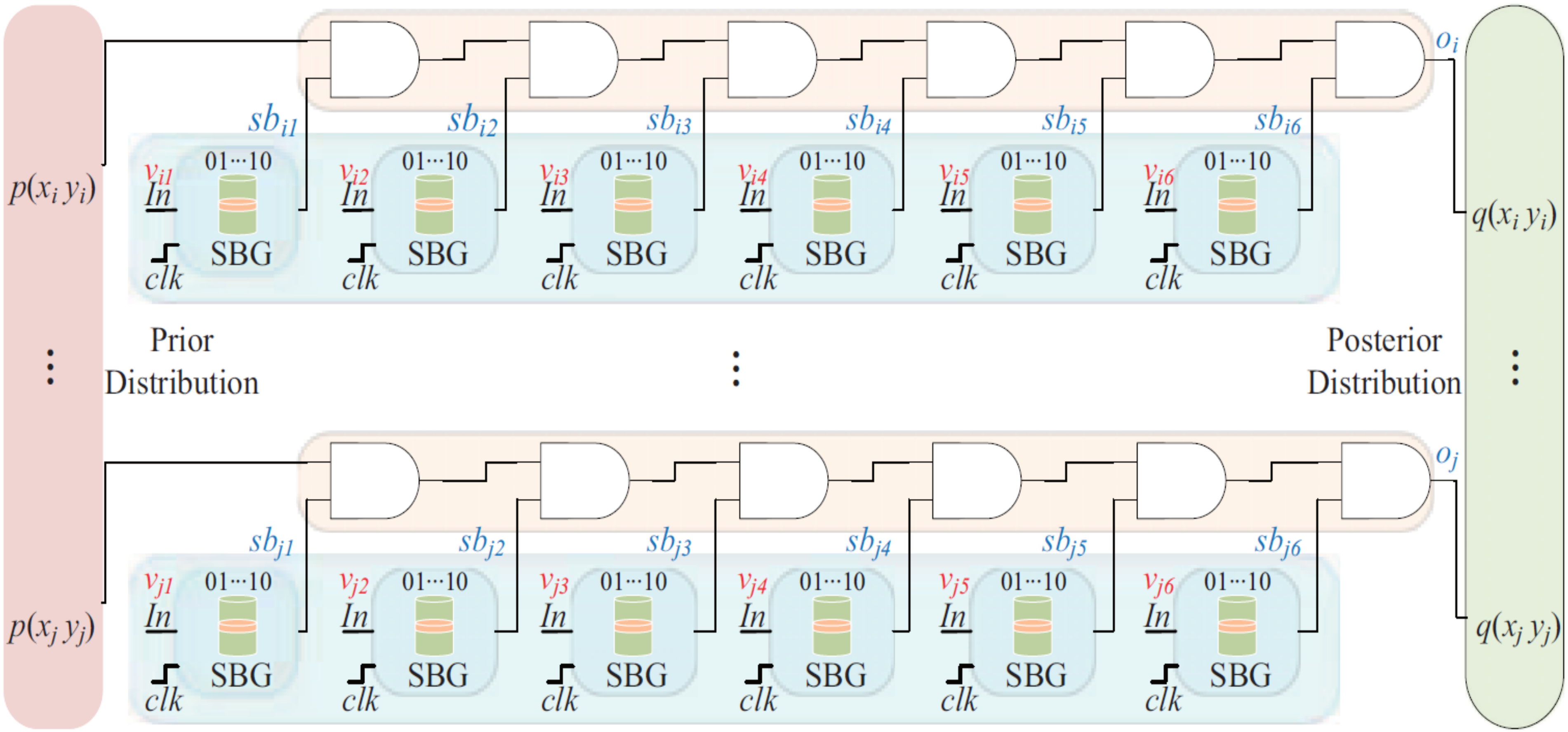


- two types of data
  - distance
  - bear
- inference procedure
  - update location with the observations
- kernel computing
  - Bayesian inference

Case Study: Sensor Fusion

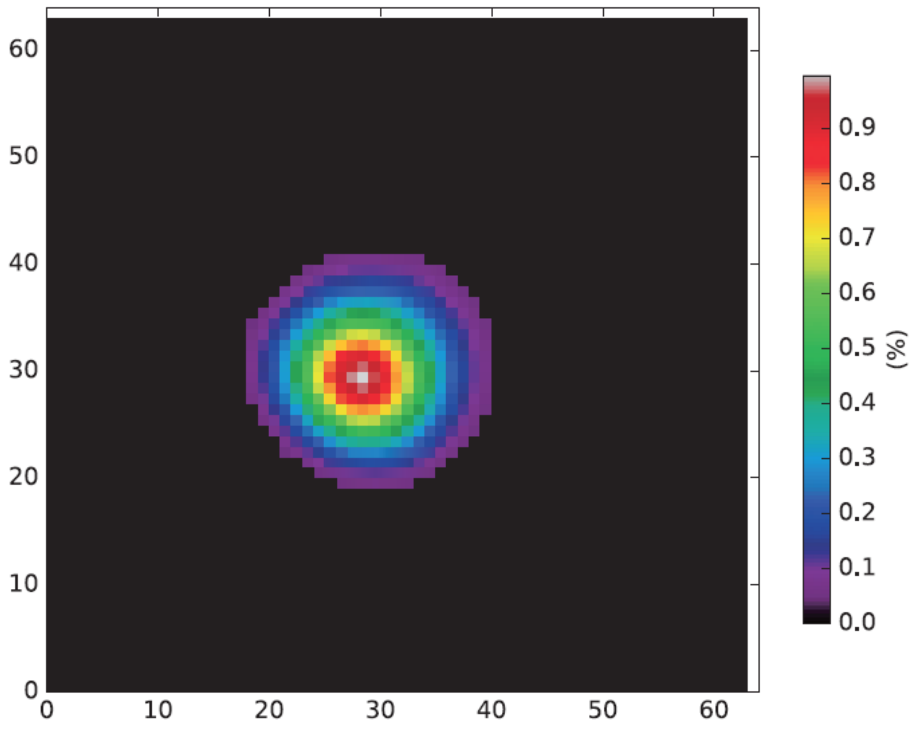
- **Example: locating a target with 3 sensors**

$$p(x\ y|D_1\ B_1\ D_2\ B_2\ D_3\ B_3) \propto p(x\ y) * \prod_i p(B_i|x\ y)p(D_i|x\ y)$$

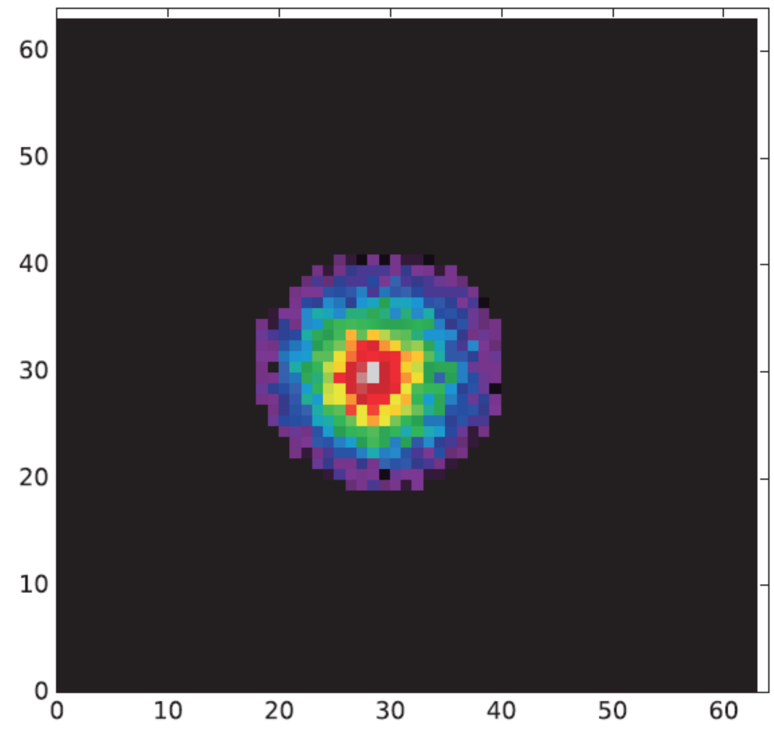


Case Study: Sensor Fusion

- **Example: locating a target with 3 sensors**
- Probability distribution comparison



Ground truth



Bayesian fusion w/ 64 cycles

Case Study: Sensor Fusion

- **Example: locating a target with 3 sensors**
- **Accuracy analysis**
  - Kullback-Leibler divergence (**KL**)
  - Ground truth v.s. Bayesian fusion

Grid size	Bit-stream length		
	64	128	256
16 x 16	0.0090	0.0043	0.0018
32 x 32	0.0086	0.0041	0.0019
64 x 64	0.0080	0.0035	0.0011

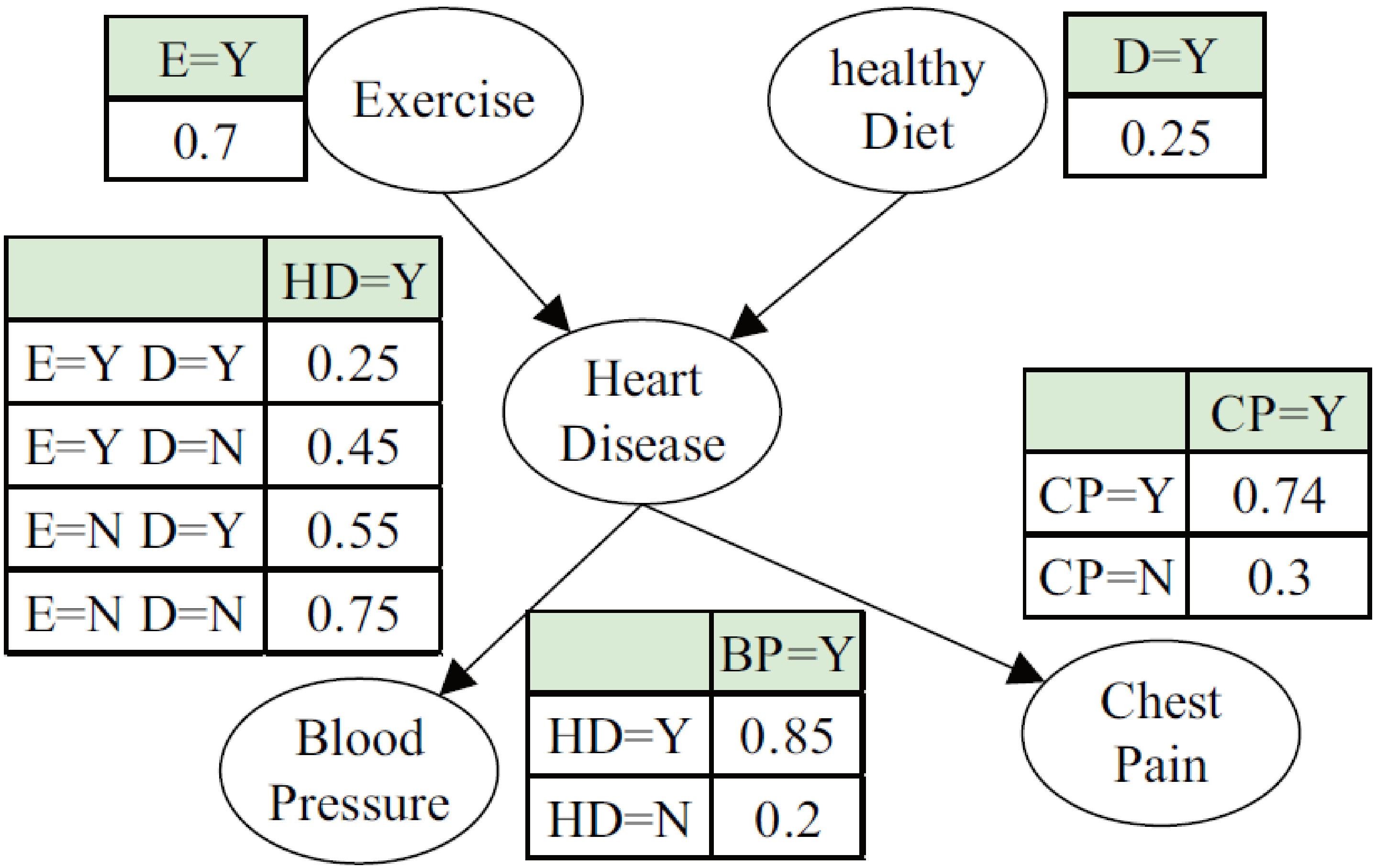
Case Study: Sensor Fusion

- **Example: locating a target with 3 sensors**
- **Inference efficiency analysis**
  - FPGA implementation* v.s. MTJ-based SC
  - 32x32 grids
  - Achieve the same accuracy (KL divergence)
  - Bit-stream length
    - FPGA-based BIS requires 10^5 bits
    - We only use 256 bits
  - Speed: FPGA (10^5 * 20 ns) v.s. MTJ (256 * 40 ns)
  - Power: FPGA (0.29 mJ) v.s. MTJ (<0.01 mJ)

* Bayesian Sensor Fusion with Fast and Low Power Stochastic Circuits, DATE 2016.

Case Study: Bayesian Belief Network

- **Example: heart disaster prediction**
  - Probabilistic graphical model



E=Y
0.7

D=Y
0.25

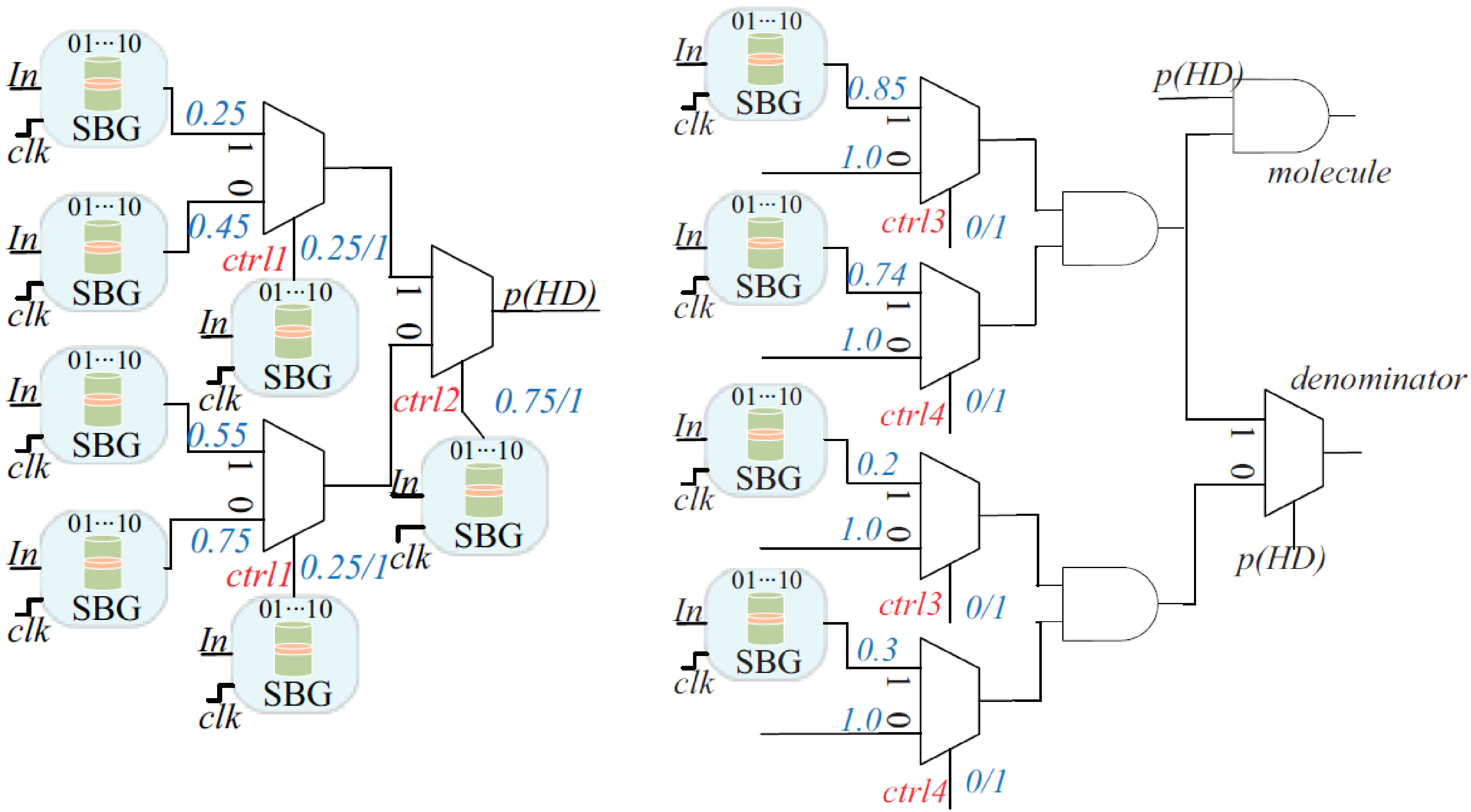
	HD=Y
E=Y D=Y	0.25
E=Y D=N	0.45
E=N D=Y	0.55
E=N D=N	0.75

	CP=Y
CP=Y	0.74
CP=N	0.3

	BP=Y
HD=Y	0.85
HD=N	0.2

Case Study: Bayesian Belief Network

- **Example: heart disaster prediction**
  - Probabilistic graphical model



Case Study: Bayesian Belief Network

- **Example: heart disaster prediction**
- Accuracy analysis
  - Compared with software results

Probability	(ctrl1, ctrl2, ctrl3, ctrl4)	Ref.*	SC
p(HD|BP)	(0.25, 0.75, 1.00, 0.00)	0.803	0.805
p(HD|D,E,BP)	(1.00, 1.00, 1.00, 0.00)	0.586	0.592
p(HD|E,BP)	(0.25, 1.00, 1.00, 0.00)	0.687	0.694
p(HD|D,E,BP,CP)	(1.00, 1.00, 1.00, 1.00)	0.777	0.742
p(HD|CP)	(0.25, 0.75, 0.00, 1.00)	0.703	0.700

* Pythonic bayesian belief network framework
https://github.com/eBay/bayesian-belief-networks

Outline

- **Background and Motivation**
- **Proposed Bayesian Inference System**
  - Spin-based stochastic bit-stream generator
  - Bayesian inference system and case studies
- **Conclusions**

Conclusions

- Build Bayesian inference system with non-conventional computing and emerging technologies
- Stochastic switching of spin device is well exploited for realizing inherent randomness for stochastic computing
- Applications have shown that our spin-based stochastic computing could improve the inference efficiency with lower design cost.

Thanks!

Q & A?