Architecting for Causal Intelligence at Nanoscale*

Santosh Khasanvis
Senior Research Scientist
BlueRISC Inc., Amherst, MA

*Research part of PhD at University of Massachusetts Amherst
(Directed by Prof. C. Andras Moritz)
Contact: santosh@bluerisc.com, andras@ecs.umass.edu
Introduction

- Emerging opportunities
  - Personalized medicine, big data analytics, cyber-security, etc.
  - Cognitive computing frameworks such as Bayesian networks (BNs) may be helpful

- Challenges
  - High computational complexity; require persistence
  - Implementation on CMOS Von Neumann microprocessors inefficient
    - Layers of abstraction, emulation on deterministic Boolean logic, rigid separation of memory and computation

- Rethink computing from the ground-up leveraging emerging nanotechnology
  - Architecting with Physical Equivalence – as direct mapping as possible of conceptual framework to physical layer
  - Disruptive technology: Potential for orders of magnitude efficiency
  - This talk: Architecting for probabilistic reasoning with BNs
Bayesian Networks (BNs)

- Probabilistic modeling of domain knowledge for reasoning under uncertainty
- Graphical representation of a domain
  - Structure: Directed Acyclic Graph; Nodes → domain variables (w/ several states); Edges → relationships/dependence between variables

Bayesian Networks are graphs, representing domain knowledge using probabilities and involve probability computations for inference

Adapted from Slides by Irina Rish, IBM – “A Tutorial on Inference and Learning in Bayesian Networks”
Overview of Approach: Architecting for Causal Intelligence

Architectural Approach
- Reconfigurable Bayesian Cell Architecture to map Bayesian Networks

Information Encoding
- Probabilities tied to physical layer, encoded in electrical signals/S-MTJ resistances used in circuits

Circuit Framework
- Mixed-signal hybrid circuits (S-MTJ + CMOS)
- Direct computation on probabilities (memory in-built)
- Bayesian Cells incorporate these circuits

Physical Layer
Non-volatile Straintronic magnetic tunneling junctions (S-MTJs) + CMOS

\[ P = \begin{pmatrix} p_1 & p_2 & \cdots & p_n \end{pmatrix} \]
\[ p_i \in \{0, 1\} \]

\[ V_{\text{out}} \text{ or } I_{\text{out}} \propto (p_1 + p_2 + p_3 + \cdots + p_n) \]
Outline

- Technology Overview: Nanoscale Straintronic MTJs (S-MTJs)
- Physically Equivalent Intelligent System for Reasoning with BNs
  - Data Encoding: Mapping probabilities in physical layer
  - Circuit Framework: Mixed-signal circuits operating on probabilities for Bayesian computations
  - Reconfigurable Bayesian Cell Architecture for BN Mapping
- Evaluation
- Summary
Non-Volatile Straintronic-MTJ (S-MTJ)

A. K. Biswas, Prof. Bandyopadhyay, Prof. Atulasimha, *Virginia Commonwealth Univ.*

- Voltage-controlled magneto-electric devices
- Stacked nanomagnets separated by spacer layer: Resistance depends on relative magnetization orientation of nanomagnets
- Strain-based switching

Outline

- Technology Overview: Nanoscale Straintronic MTJs

- Physically Equivalent Intelligent System for Reasoning with BNs
  - Data Encoding: Mapping probabilities physically using S-MTJs
  - Circuit Framework: Mixed-signal circuits operating on probabilities for Bayesian computations
  - Reconfigurable Bayesian Cell Architecture for BN Mapping

- Evaluation

- Summary
Encoding Probability

- Represented as non-Boolean flat probability vector of spatially distributed digits

\[
[\begin{array}{cccccc}
p_1 & p_2 & p_3 & \ldots & p_n \\
\end{array}]
\]

\[
P = \frac{1}{n} \left( \sum_{i=1}^{n} p_i \right)
\]

Resolution = 1/n; where \( n \): #digits

- **Physical Equivalence**: Direct correlation to S-MTJ resistances and electrical signals
- E.g. Using 10 digits, \( p_i \in \{0, 1\} \leftrightarrow \text{Resistance } r_i \in \{R_{\text{OFF}}, R_{\text{ON}}\} \leftrightarrow \text{Voltages } V_{i1}, V_{i2} \in \{0V, 40mV\}

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
</table>

\[
\frac{V_h}{V_1} = 0, \frac{V_h}{V_2} = 0, \frac{V_h}{V_1} = 0, \frac{V_h}{V_2} = 0, \frac{V_h}{V_1} = 0, \frac{V_h}{V_2} = 0, \frac{V_h}{V_1} = 0, \frac{V_h}{V_2} = 0, \frac{V_h}{V_1} = 0, \frac{V_h}{V_2} = 0
\]

\[
P = 0.4
\]

- Equivalent Digital Voltages
- Equivalent S-MTJ Resistances

- Digit \( p_i \) related to S-MTJ resistance \( r_i \) as follows

\[
r_i = \frac{\beta}{p_i + \epsilon}
\]

\( \beta \) and \( \epsilon \) are constants

\[
\epsilon = \frac{1}{R_{\text{OFF}} - R_{\text{ON}}} \quad \beta = \epsilon \cdot R_{\text{OFF}}
\]
Circuit Framework

- Unconventional magneto-electric mixed-signal circuit framework

- **Physical Equivalence:** Directly implements *Bayesian computations on probabilities* using underlying circuit principles in analog domain
  - Input: Digital; Output: Analog

- **Approach**
  - Operating on spatial probability digital vectors that are converted into an analog representation of single probability value → this is referred to as *Probability Composer*
  - Probability Addition, Multiplication Composers internally use *Probability Composers*
  - Cascade computational blocks for Bayesian functions: Enabled by *Decomposers*

Probability Composer Circuit

- Needed to convert spatial probability representation (digital) → analog quantity representing total probability value in current/voltage domain
- Parallel topology of S-MTJs; effective resistance encodes probability
  - Individual S-MTJ resistances set using digital voltages as shown earlier

**Probability Composer:** Collection of S-MTJs
- Probability value encoded in $1/R_{PC}$
- Read-out in current/voltage
Elementary Arithmetic Composer Circuits

Addition Composer Circuit

Current Addition

\[ I_{out} = \frac{nV_{REF}}{\beta} [P_A + P_B] \quad R_L \ll R_{PC} \]

\[ V_{out} = I_{out} \cdot R_L \]

Multiplication Composer Circuit

Ohm's law

\[ I_{out} = \frac{V_{in1}}{R_{in2}} = V_{in1} \cdot P_{in2} \]

Input \( P_A \): Voltage domain
Input \( P_B \): S-MTJ Resistance

Simulated Output Characteristics (HSPICE)
Combining Elementary Composers: Add-Multiply

- Example: \( P_{\text{out}} = P_a \cdot P_b + P_c \cdot P_d \); typical in BN inference computations
  \[ \text{ADD}\{ \text{MUL}(P_a, P_b), \text{MUL}(P_c, P_d)\} \]; two levels of hierarchical instantiation
- Elementary Composers = MUL, arranged in topology self-similar to ADD (Dominator Composer)
Outline

- Technology Overview: Nanoscale Straintronic MTJs
- Physically Equivalent Intelligent System for Reasoning with BNs
  - Data Encoding: Mapping probabilities in physical layer
  - Circuit Framework: Mixed-signal circuits operating on probabilities for Bayesian computations
    - Elementary Arithmetic Composers
    - Inference in BNs: Belief Propagation Algorithm Overview
    - Composers for BN Inference Operations
    - Reconfigurable Bayesian Cell Architecture for BN Mapping
- Evaluation
- Summary
Bayesian Inference: Pearl’s Belief Propagation

- Compute belief $P(X_i | E)$ based on evidence $E$ using local computations and message propagation
- Each node maintains
  - Conditional probability tables (CPTs): $\text{CPT}_{jk}(X_i) = P(X_i = j | \text{Pa}(X_i) = k)$
  - Likelihood $\lambda(X_i) = P(E^- | X_i)$ and Prior $\pi(X_i) = P(X_i | E^+)$
  - Belief Vector $\text{BEL}(X_i) = P(X_i | E)$
- Local node computations using messages from neighbors
  - $\lambda$ messages from child to parent to compute $\lambda(X_i)$
  - $\pi$ messages from parent to child nodes for $\pi(X_i)$
  - $\text{BEL}(X_i) = \lambda(X_i) \cdot \pi(X_i)$
- Applicable to trees and poly-trees

Repeated application of Bayes Rule

\[
\lambda(X) = \begin{pmatrix}
\lambda_1(X) \\ \\
\lambda_2(X) \\ \\
\lambda_3(X) \\ \\
\lambda_4(X)
\end{pmatrix} \quad \pi(X) = \begin{bmatrix} \pi_1(X) & \pi_2(X) & \pi_3(X) & \pi_4(X) \end{bmatrix}
\]

\[
\text{BEL}(X) = \begin{bmatrix} \text{BEL}(X=1) & \text{BEL}(X=2) & \text{BEL}(X=3) & \text{BEL}(X=4) \end{bmatrix}
\]

\[
\text{CPT}(X|A) = \begin{pmatrix}
P(X=1 | A=1) & P(X=1 | A=2) & P(X=1 | A=3) & P(X=1 | A=4) \\
P(X=2 | A=1) & P(X=2 | A=2) & P(X=2 | A=3) & P(X=2 | A=4) \\
P(X=3 | A=1) & P(X=3 | A=2) & P(X=3 | A=3) & P(X=3 | A=4) \\
P(X=4 | A=1) & P(X=4 | A=2) & P(X=4 | A=3) & P(X=4 | A=4)
\end{pmatrix}
\]

Composer Circuits for BN Inference Operations

- Uses either elementary arithmetic composers or combines them
  - Likelihood Estimation: $\lambda(X) = \lambda_Y(X) \times \lambda_Z(X)$
  - Prior Estimation: $\pi(X) = \pi_X(A) \otimes CPT(X|A)$
  - Belief Update: $BEL(X) = \pi(X) \times \lambda(X)$
  - Diagnostic Support to Parent: $\lambda_X(A) = CPT(X|A) \otimes \lambda(X)$
  - Predictive Support to Child nodes: $\pi_Y(X) = \alpha \pi(X) \times \lambda_Z(X)$
  $\pi_Z(X) = \alpha \pi(X) \times \lambda_Y(X)$

Add-Multiply Composers for Prior Estimation, Diagnostic Support

Multiplication Composers for Likelihood Estimation, Belief Update, Predictive Support
Outline

- Technology Overview: Nanoscale Straintronic MTJs

- **Physically Equivalent Intelligent System for Reasoning with BNs**
  - Data Encoding: Mapping probabilities in physical layer
  - Circuit Framework: Mixed-signal circuits operating on probabilities for Bayesian computations
  - Reconfigurable Bayesian Cell Architecture for BN Mapping

- Evaluation

- Summary
Physically Equivalent Architecture for BNs

- **Physical Equivalence**: *Every node in DAG mapped to a Bayesian Cell* in H/W; incorporates non-volatile Arithmetic Composers for Bayesian computations
- **Reconfigurable links using Switch Boxes** (similar to FPGAs) to map any BN structure
- Persistence in configuration + computation through non-volatile Composers; no need for external memory
Outline

- Technology Overview: Nanoscale Straintronic MTJs
- Physically Equivalent Intelligent System for Reasoning with BNs

Evaluation

- Methodology
- System-level Evaluation for BN Inference using Physically Equivalent Framework
- Analytical Modeling of BNs Inference Performance on CMOS Multi-core Processors and Comparison

Summary
Example Bayesian Graph to Estimate System-level Performance

- Assuming a balanced binary tree structure for system level performance estimation
  - Each parent has 2 child nodes; each node has 4 states (applications like gene expression networks require 3*)
  - All leaf nodes are treated as evidence variables
- Total number of nodes scaled from ~100 to ~1 million

BN inference execution time estimated based on critical path delay ($T_{BC}$) in each BC and Switch Box communication delay ($T_{SB}$) for worst-case
- For Bayesian Network with $n$ levels; (active nodes in a time-step operate in parallel)
  $$T_{exec} = (2n-1) \times T_{BC} + T_{comm}$$

Evaluation Methodology for BN Composer Circuits

- Delay, power measured using HSPICE simulations
  - HSPICE behavioral macromodels built for S-MTJs
- Area determined by number of S-MTJs + CMOS support
  - Accounting for S-MTJ spacing to minimize magnetic interactions

### Dipole Coupling

Low coupling energy implies minimal magnetic interaction

### S-MTJ Center-Center Distance

Collaboration: Data provided by VCU group (Prof. Atulasimha, Prof. Bandyopadhyay)

<table>
<thead>
<tr>
<th>Module</th>
<th>Critical Path Delay (ns)</th>
<th>Area (μm²)</th>
<th>Worst-case Power (μW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood Estimation</td>
<td>144</td>
<td>20</td>
<td>4.57</td>
</tr>
<tr>
<td>(Multiplication Composersx4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Belief Update</td>
<td>144</td>
<td>20</td>
<td>4.57</td>
</tr>
<tr>
<td>(Multiplication Composersx4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior Estimation</td>
<td>137</td>
<td>50</td>
<td>11.24</td>
</tr>
<tr>
<td>(Add-multiply Composersx4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostic Support</td>
<td>137</td>
<td>50</td>
<td>11.24</td>
</tr>
<tr>
<td>(Add-multiply Composersx4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior Support</td>
<td>144</td>
<td>40</td>
<td>9.14</td>
</tr>
<tr>
<td>(Multiplication Composersx8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decomposer</td>
<td>132.9</td>
<td>240</td>
<td>11.37</td>
</tr>
<tr>
<td>(x60)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMOS Op-Amp</td>
<td>100</td>
<td>95.4</td>
<td>89.32</td>
</tr>
<tr>
<td>(x176)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch Box</td>
<td>10</td>
<td>398.8</td>
<td>0.85</td>
</tr>
</tbody>
</table>
Path Delays within Bayesian Cell for Inference

<table>
<thead>
<tr>
<th>Path Label</th>
<th>Total Path Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>746.8</td>
</tr>
<tr>
<td>2</td>
<td>754.2</td>
</tr>
<tr>
<td>3</td>
<td>998.2</td>
</tr>
<tr>
<td>4</td>
<td>991.2</td>
</tr>
</tbody>
</table>

Worst-case Delay $T_{BC}$
Implementation of BNs on Multi-core Processors

- Hardware platform: Multi-core processor (100 cores) based on TILEPro from Tilera Corp.*
- **Lower bound execution time** analytically estimated based on computation + memory requirements for inference using Belief Propagation algorithm
  - Maximum idealized parallelism and operation cost, no network contention, no synchronization cost
- Power and area from specifications

Comparison vs. Multi-Core Processors

Delay Comparison for Bayesian Inference

Log-scale

1.0E+00
1.0E-01
1.0E-02
1.0E-03
1.0E-04
1.0E-05
1.0E-06
1.0E-07
1.0E-08

Inference Runtime (Seconds)

Size of Binary Tree Bayesian Network (No. of Variables)

100 Core  PEAR  100 Core  PEAR  100 Core  PEAR  100 Core  PEAR  100 Core  PEAR
127  1023  16383  131071  1048575

Legend:

CMOS Multicore Processors
- Blue: Arithmetic Execution + Communication
- Red: Memory Overhead

Physically Equivalent Architecture (PEAR)
- Orange: Arithmetic Execution + Communication

Speedup over 100-Core Processors

12x  80x  8686x
Comparison vs. Multi-Core Processors (contd.)

**Power Comparison**

Log-Scale

100-Core Processor Power

4788x Efficiency (Power x Delay)

Size of Binary Tree Bayesian Network (No. of Variables)

PEAR Power (Size = No. of variables)

**Area Comparison**

Log-Scale

100-Core Processor Area

Size of Binary Tree Bayesian Network (No. of Variables)

PEAR Area (Size = No. of variables)
Summary

- Physically equivalent intelligent system for probabilistic reasoning using Bayesian Networks (BNs)
  - Architected from ground-up and enabled by emerging nanotechnology
  - Probability encoding based mixed-signal magneto-electric circuit framework
  - Reconfigurable Bayesian Cell architecture

- Up to 8686x inference speed-up, 4788x lower energy for BNs with ~1M nodes for resolution 0.1 vs. 100-core processor

- Reasoning/learning tasks on complex problems with million variables made feasible

- Embed real-time intelligence capabilities at smaller scale (100s of variables) everywhere
Thank you

Acknowledgements

- Collaboration with Prof. Atulasimha, Prof. Bandyopadhyay, VCU
- Sponsored by National Science Foundation (CCF-1407906, ECCS-1124714, CCF-1216614, CCF-1253370)