Highlights from ASP-DAC 2014

Abde Ali Kagalwalla

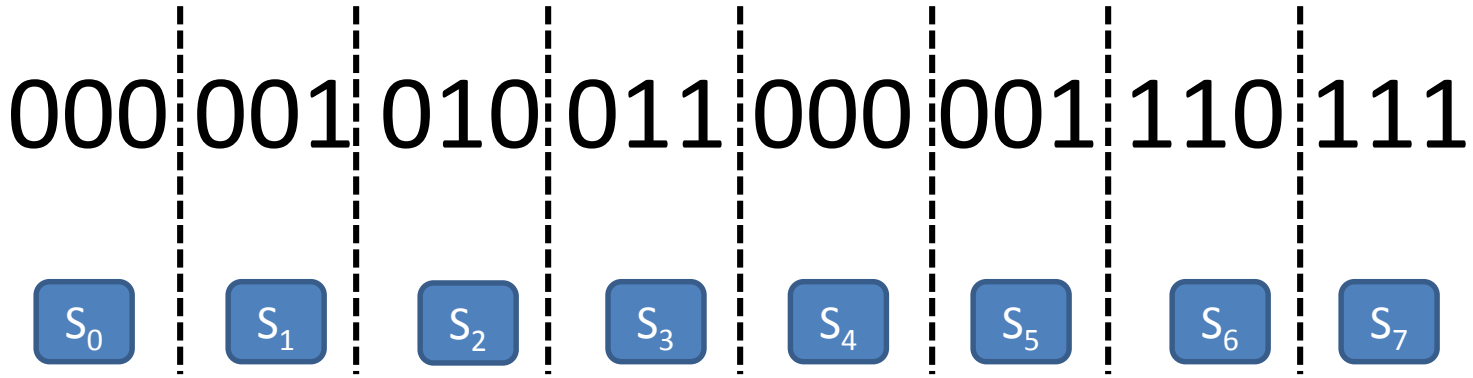
Outline

- **Data Compression via Logic Synthesis**
  - L. Amaru, P.-E. Gaillardon, A. Burg and G. D. Micheli (EPFL, Switzerland)
- Applying VLSI EDA to Energy Distribution System Design
  - Sani Nassif, Gi-Joon Nam, Jerry Hayes (IBM Research) and Sani Fakhouri (UC Irvine)
- Energy Efficient In-Memory Machine Learning for Data Intensive Image-Processing by Non-volatile Domain-Wall Memory
  - H. Yu, Y. Wang, S. Chen, W. Fei (NTU Singapore), C. Weng, J. Zhao and Z. Wei (Huawei Shannon Lab, China)

Lossless Compression

- Two main steps in most data compression approaches
  - Data decorrelation: Reduce autocorrelation of data. E.g. DCT (Discrete Cosine Transform), dictionary
  - Entropy encoding: Compress the decorrelated data. E.g. Huffman coding, arithemetic coding
- Logic synthesis also compresses a binary sequence
  - Can it be used efficiently in compression ?
  - Prior work less efficient compared to conventional compression

Example of Compression using Logic Synthesis



Define 3 output Boolean function G

$$G_0 = BR(6) + BR(7) = I_0 I_1 \bar{I}_2 + I_0 I_1 I_2 = I_0 I_1$$

Objective: Find minimum number of gates to represent the 3-bit function G

Additional heuristic: Stimulated by BR(i) iff stimulated by

Overall Method for Compression

- Input: Binary string B
- Output: Compressed string C
- Summary of method
  1. Split B in L-bit substrings
  2. Find sum of products expression for L-bit function G corresponding to substrings of B
  3. Boolean minimization of every bit of G to circuit K
  4. If G_i too big or minimization too slow, use entropy encoding to compress
  5. Resource sharing of circuit K

Results for Benchmarks generated from Causal Process

TABLE I
DATA COMPRESSION RESULTS

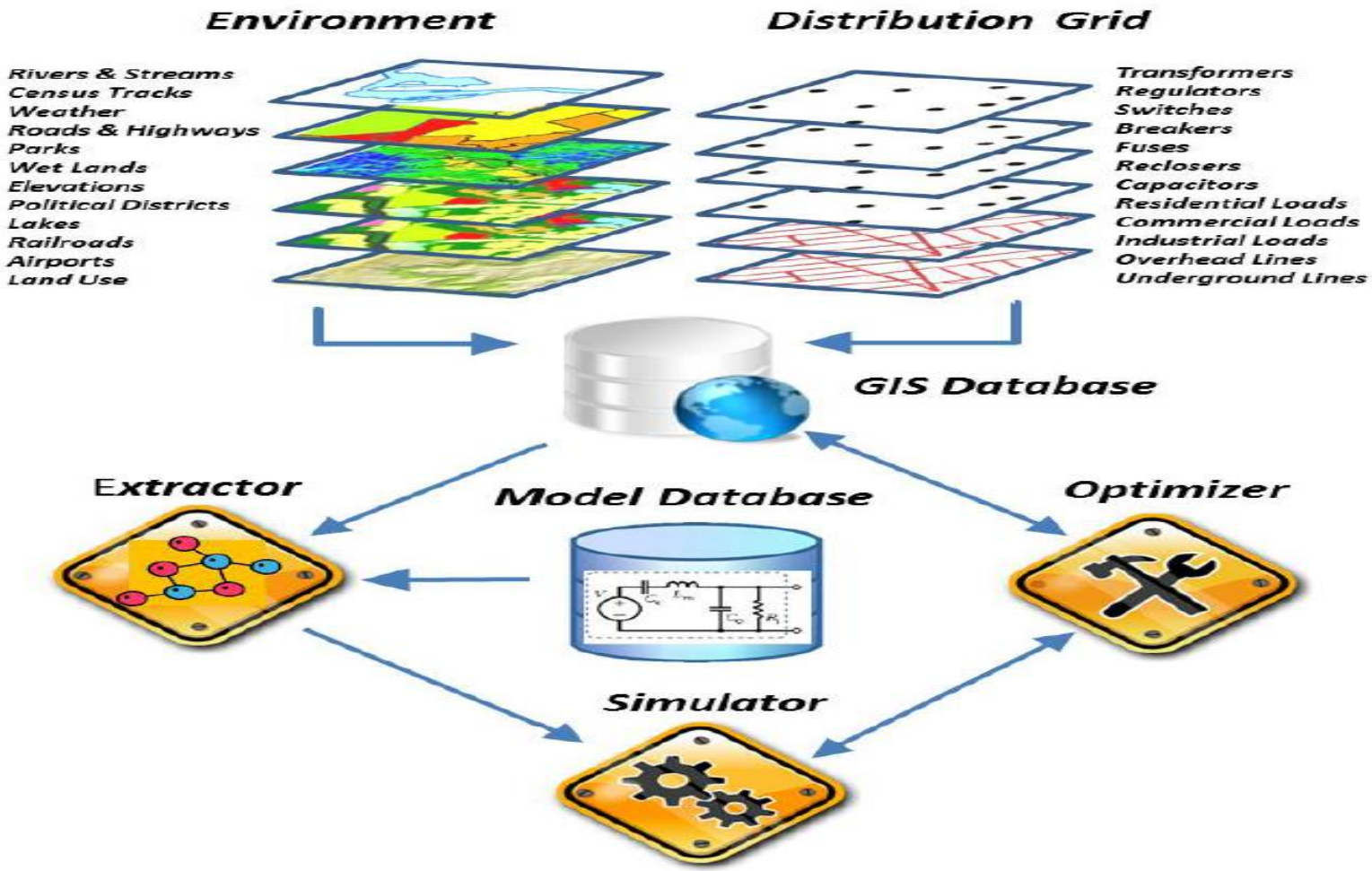
Benchmark	Original Data Size	ZIP	DCT+ZIP	bzip2	7zip	Our Approach	ZIP Runtime
Linear Data	2.2 MB	208 KB	868 KB	316 KB	60 KB	8 KB	0.3 s
	25 MB	2.1 MB	8.3 MB	3.1 MB	888 KB	8 KB	2.1 s
	287 MB	21 MB	81 MB	31 MB	3.4 MB	302 KB	32 s
Linear Data + Noise	2.2 MB	264 KB	872 KB	258 KB	212 KB	80 KB	0.4 s
	25 MB	2.7 MB	8.4 MB	2.6 MB	2.4 MB	700 KB	3.0 s
	287 MB	27 MB	84 MB	30 MB	23 MB	7.1 MB	43 s
Quadratic Data	3.3 MB	484 KB	816 KB	532 KB	272 KB	8 KB	1.0 s
	39 MB	5.3 MB	7.6 MB	6.1 MB	3.3 MB	16 KB	6.1 s
	449 MB	59 MB	71 MB	67 MB	40 MB	566 KB	64 s
Random (XOR-intensive network)	1.6 MB	116 KB	304 KB	124 KB	44 KB	8 KB	0.1 s
	20 MB	1.2 MB	3.2 MB	1.5 MB	796 KB	8 KB	1.2 s
	230 MB	12 MB	31 MB	15 MB	3.8 MB	234 KB	10 s
Average runtime (normalized to ZIP)	–	1	–	1.5 x	8 x	12 x	–

Outline

- Data Compression via Logic Synthesis
  - L. Amaru, P.-E. Gaillardon, A. Burg and G. D. Micheli (EPFL)
- **Applying VLSI EDA to Energy Distribution System Design**
  - Sani Nassif, Gi-Joon Nam, Jerry Hayes (IBM Research) and Sani Fakhouri (UC Irvine)
- Energy Efficient In-Memory Machine Learning for Data Intensive Image-Processing by Non-volatile Domain-Wall Memory
  - H. Yu, Y. Wang, S. Chen, W. Fei (NTU Singapore), C. Weng, J. Zhao and Z. Wei (Huawei Shannon Lab, China)

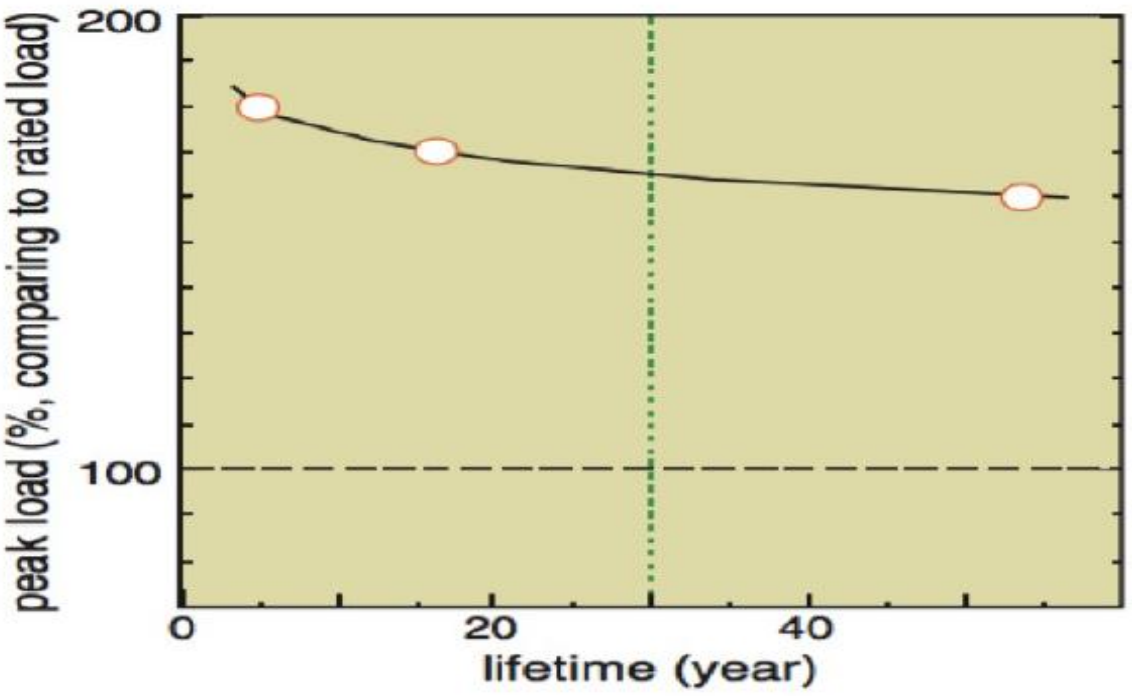
Motivation

- Using rigorous modeling, analysis and optimization (as done in EDA) for solving problems in power distribution



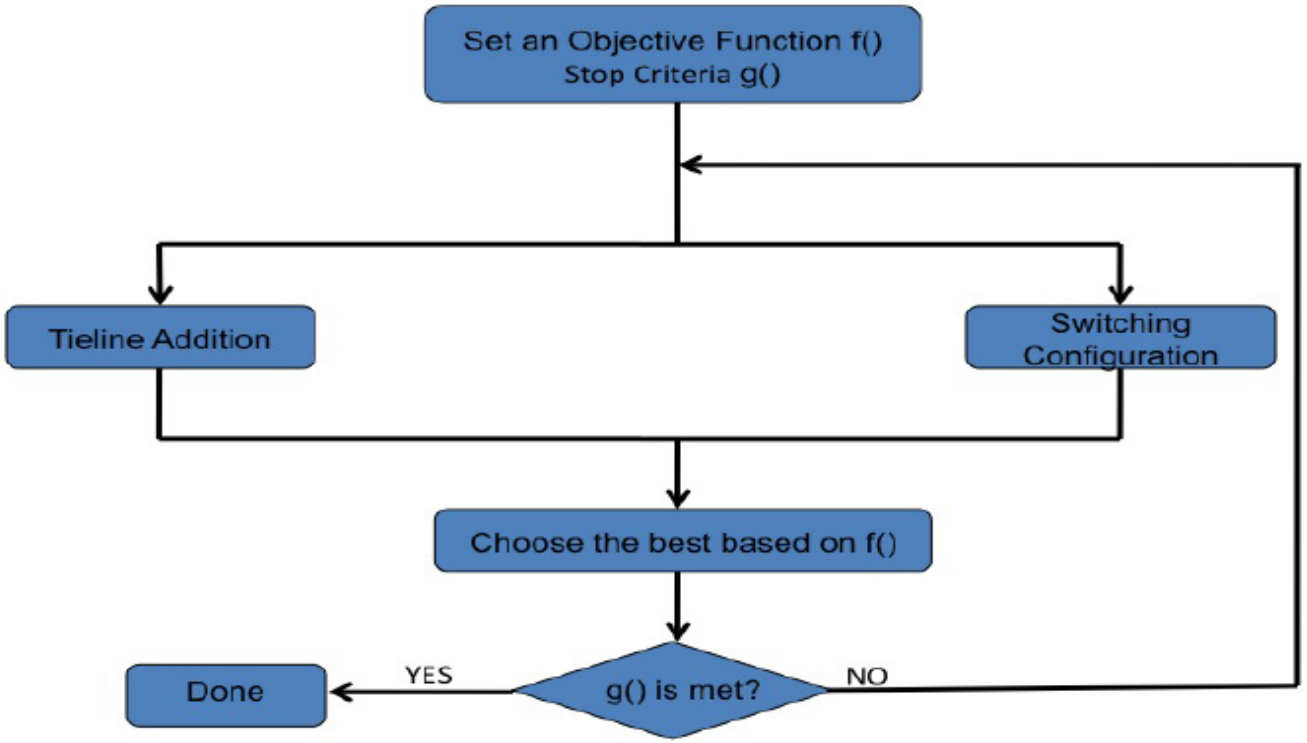
Load Balancing for Reliability

- Lifetime of costly components (like transformers) depends on usage level relative to rating
- Two techniques: Adding tieline or reconfigure switches



Load Balancing Optimization Method

- Key idea is to construct grid graph and define slack (rating - load) for each graph edge
- Simple greedy algorithms. E.g. for tieline addition, connect vertices with high load differentials and less physical distance



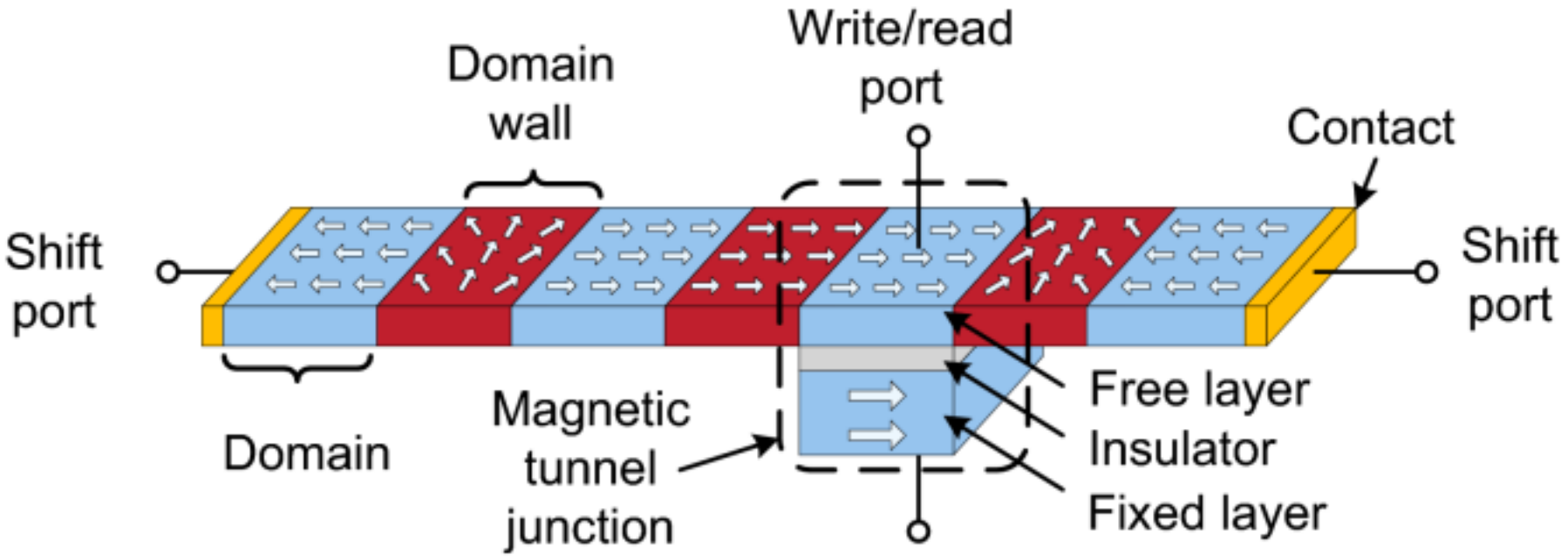
Other Problems Mentioned

- Static simulation of entire grid → Use techniques commonly deployed in power grid analysis of VLSI designs
- Contingency planning → Add redundancies to network to account for geography, weather, societal need (e.g. hospitals)
- Component upgrade
- Loss minimization

Outline

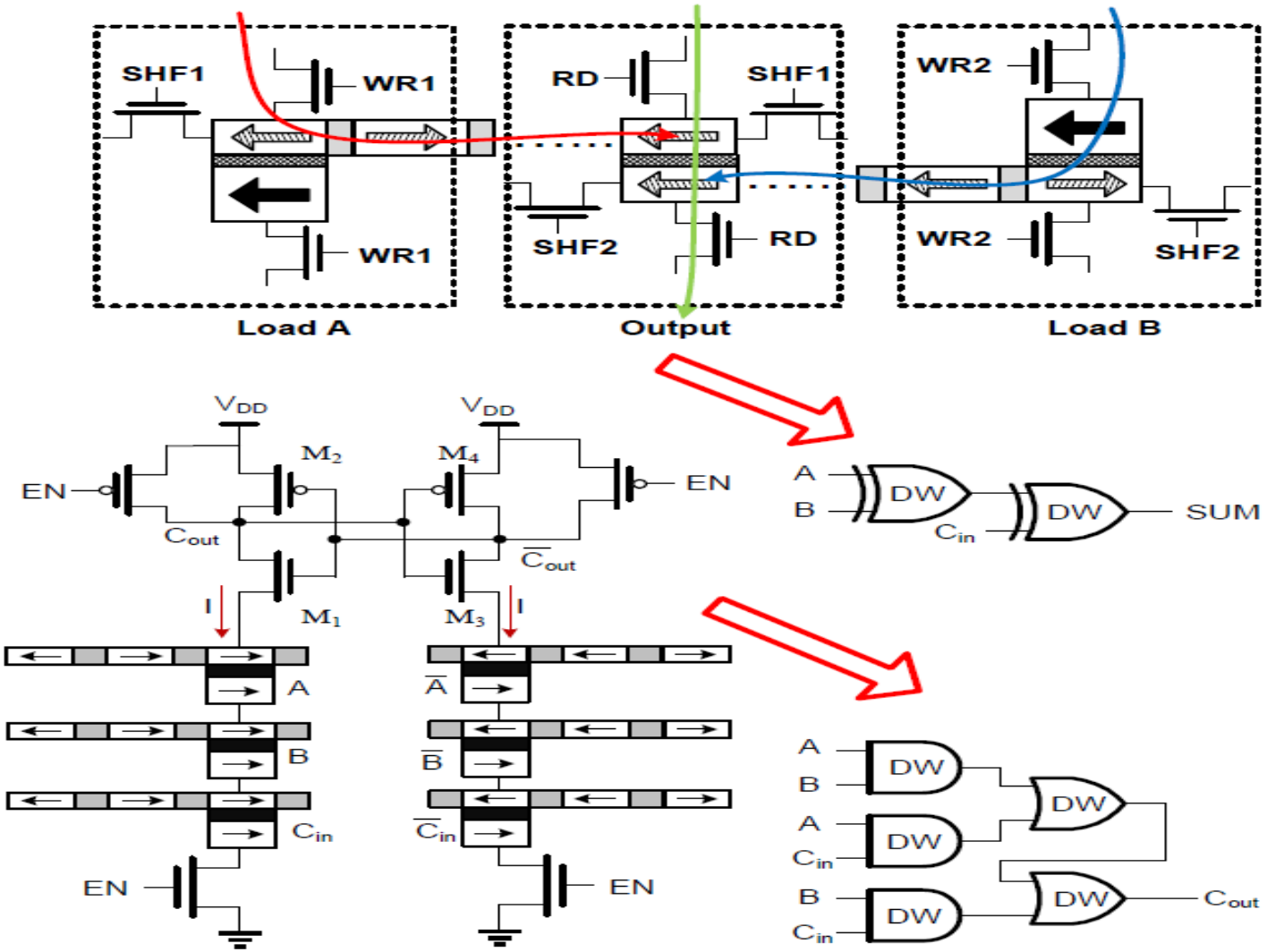
- Data Compression via Logic Synthesis
  - L. Amaru, P.-E. Gaillardon, A. Burg and G. D. Micheli (EPFL)
- Applying VLSI EDA to Energy Distribution System Design
  - Sani Nassif, Gi-Joon Nam, Jerry Hayes (IBM Research) and Sani Fakhouri (UC Irvine)
- **Energy Efficient In-Memory Machine Learning for Data Intensive Image-Processing by Non-volatile Domain-Wall Memory**
  - H. Yu, Y. Wang, S. Chen, W. Fei (NTU Singapore), C. Weng, J. Zhao and Z. Wei (Huawei Shannon Lab, China)

Domain Wall Nanowire

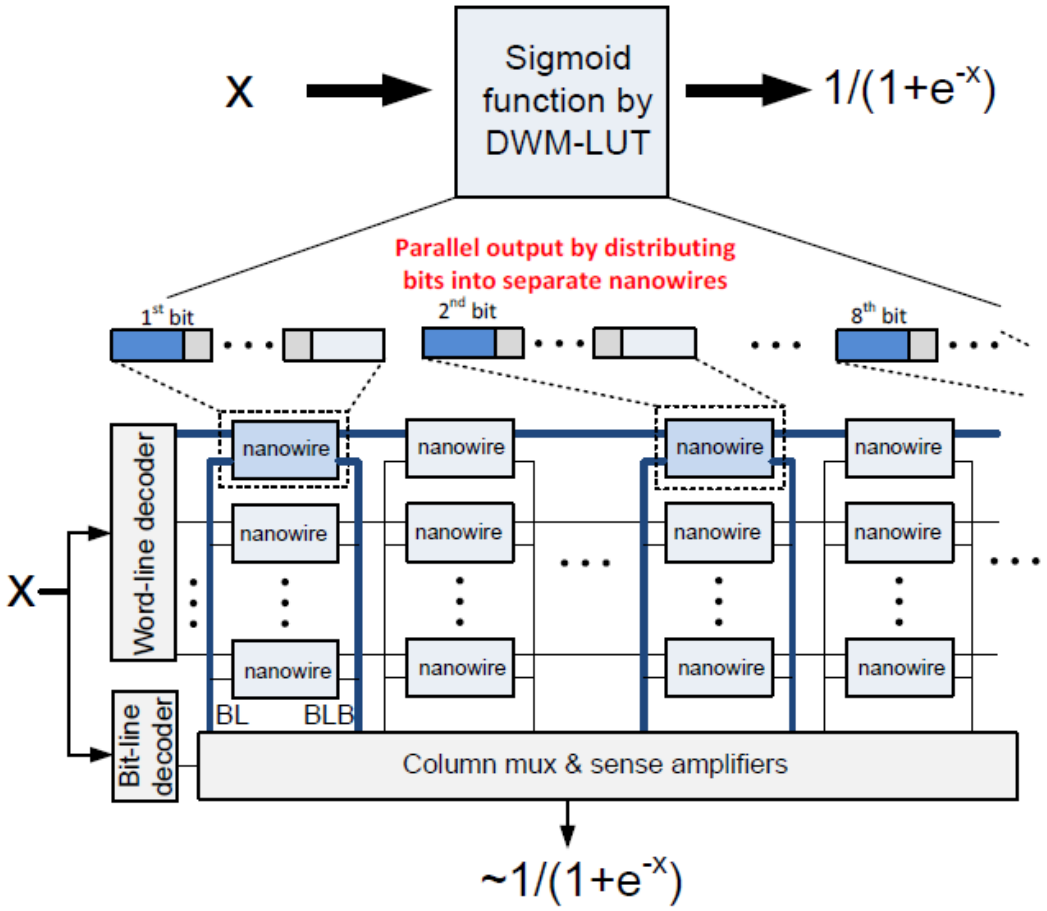


- Antiparallel spin direction has higher resistance (GMR effect)
- Spin based non-volatile memory → No standby power
- Can be used directly as part of in-memory logic
- Goal: Use this technology to implement neural network in hardware

Binary sum using DW-nanowire



LUT Based Sigmoid Function



SPICE Simulation Results at 32nm

Domain wall nanowire device			
Operation	Speed (cycles)	Energy (pJ)	
read	1	0.5	
write	1	0.1	
shift	1	0.3	
Domain wall nanowire logic			
Logic	Speed (cycles)	Energy (pJ)	Area (um^2)
8-bit full adder	54	40	2.6
8-bit multiplier	163	308	18.9
8-bit sigmoid (LUT)	2	116	31.8

CACTI Simulation Results for an Image Processing Application

Platform	DW-NN	GPP (with on-chip memory)	GPP (with off-chip memory)
Computational resources utilized	1×Processor 7714×DW-ADDER 7714×DW-MUL 551×DW-LUT 1×controller	1×Processor	1×Processor
Area of computational units	18 mm^2 (processor) + 0.5 mm^2 (accelerators)	18 mm^2	18 mm^2
Power (Watt)	10.1	12.5	12.5
Throughput (GBytes/s)	108MBytes/s	9.3MBytes/s	9.3MBytes/s
Energy efficiency (nJ/bit)	7	389	642