HieIM: Highly Flexible In-Memory Computing Using STT MRAM

Deliang Fan

Assistant Professor

dfan@ucf.edu

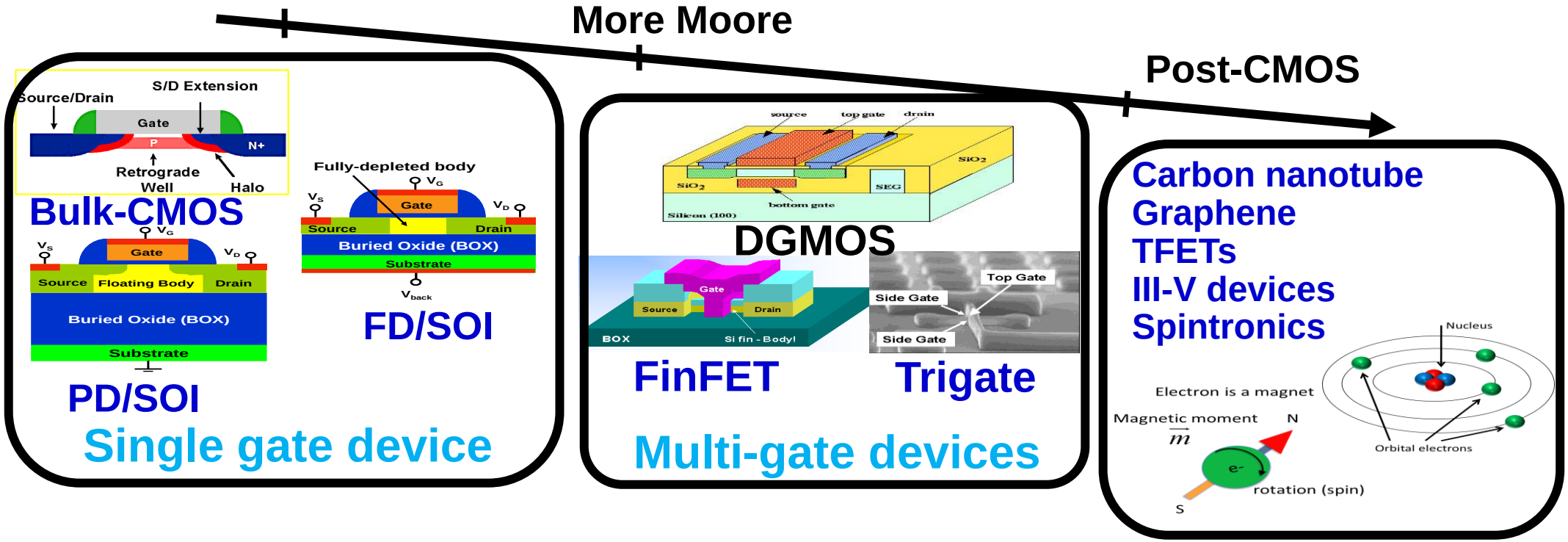
http://www.eecs.ucf.edu/~dfan/

Department of Electrical and Computer Engineering,
University of Central Florida, Orlando, FL

OUTLINE

- Motivation
- Post-CMOS Spintronic Devices
- In-Memory Processing Platform based on STT-MRAM
- Performance Evaluation
- Case Study I: In-memory Bulk Bitwise Vector Operation
- Case Study II: In-memory Data Encryption Engine

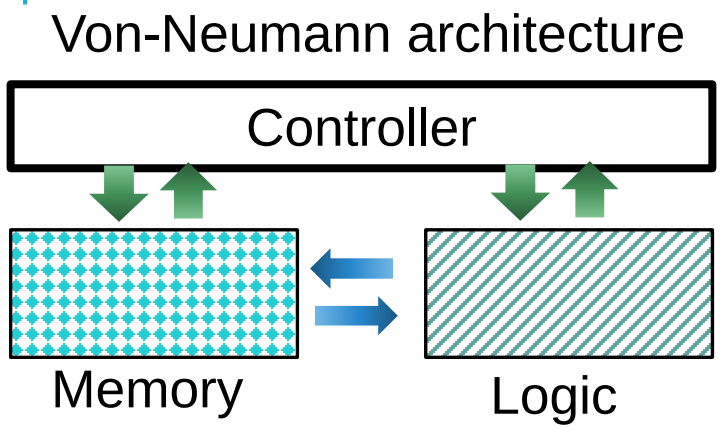
MOTIVATION (DEVICE)-TECHNOLOGY TREND



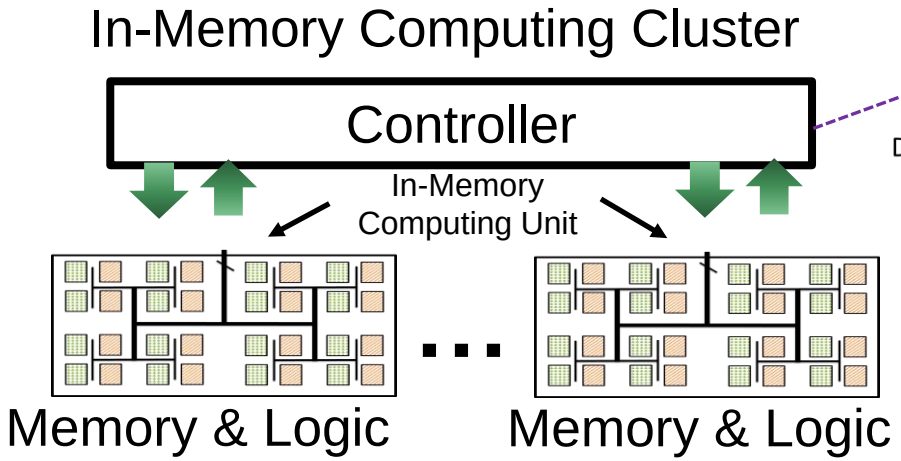
- Energy efficient and high performance computing hardware development is beginning to stall fundamentally due to limitations in both ***devices*** and ***architectures***.

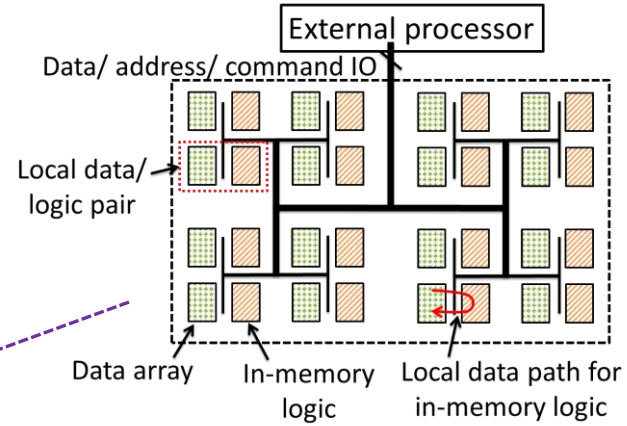
- First, the current computing platforms primarily depend on Complementary Metal Oxide Semiconductor (CMOS) technology, which is reaching its power wall

MOTIVATION (ARCHITECTURE)



VS.





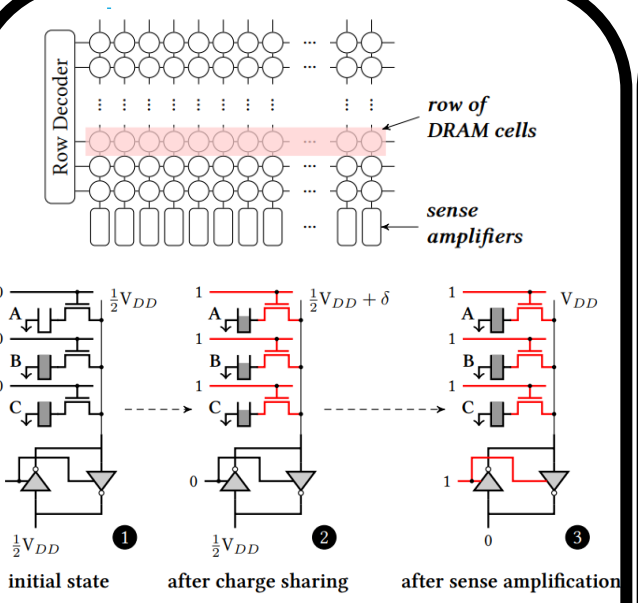
- ➢ Energy hungry data transfer
- ➢ Long memory access latency
- ➢ Limited memory bandwidth

- ✓ Parallel, local data processing
- ✓ Short memory access latency
- ✓ Ultra-low energy
- ✓ Programmable, Low cost/ area

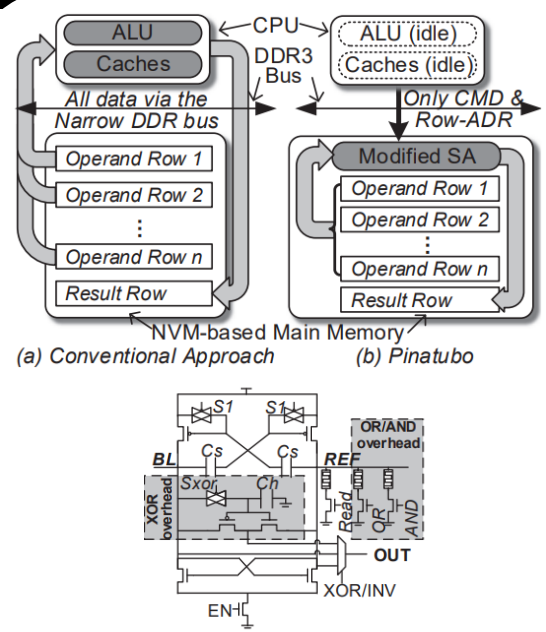
- There is an urgent need to investigate fundamentally different devices and architectures for information processing and data storage with the ability to continuously deliver energy efficient and high performance computing solutions.

[1] P. Chi et al., “Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory,” in ISCA, vol. 43, 2016.
[2] S. Li et al., “Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories,” in 2016 53nd DAC. IEEE, 2016..

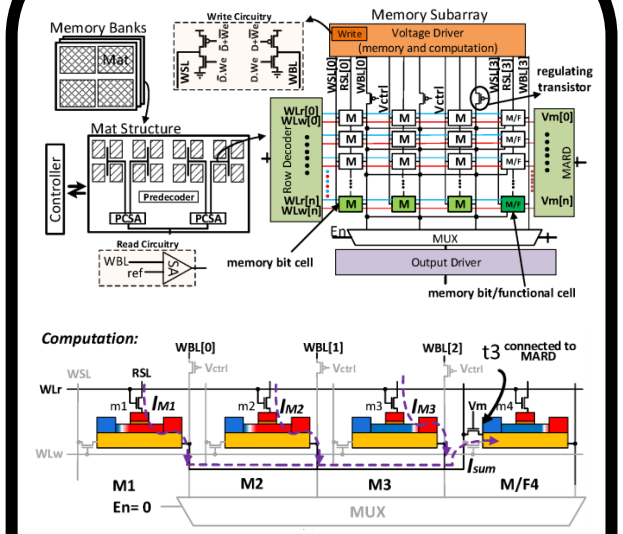
RECENT IN-MEMORY COMPUTING PLATFORMS



Ambit: DRAM-based



Pinatubo: NVM-based



RIMPA: DWM-based

HieIM: MRAM-based

- Operand locality issue
- Original data overwritten
- Multi-Cycle operations
- Low area overhead
- Hardware-friendly
- exploiting the full internal DRAM bandwidth

- Operand locality issue
- Modified SA
- Medium area overhead
- support one-step multi-row operations
- General platform

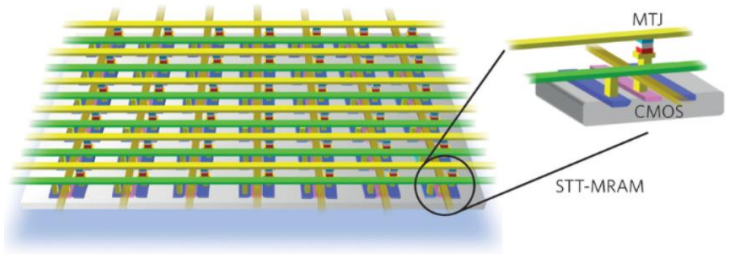
- Operand locality issue
- Large area overhead
- Fast MG-based computation
- Ultra-low power

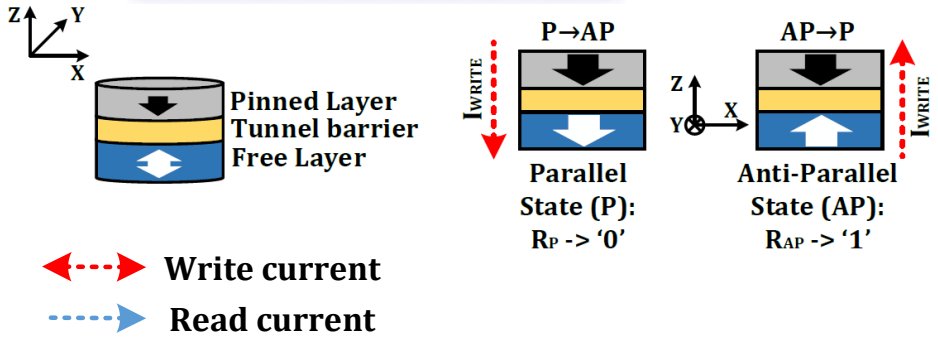
OUTLINE

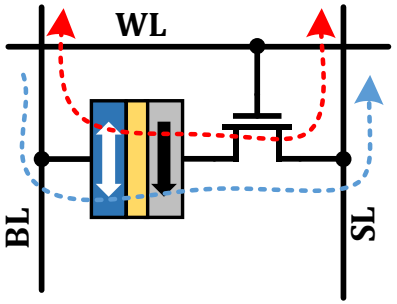
- Motivation
- Post-CMOS Spintronic Devices
- In-Memory Processing Platform based on STT-MRAM
- Performance Evaluation
- Case Study I: In-memory Bulk Bitwise Vector Operation
- Case Study II: In-memory Data Encryption Engine

POST-CMOS SPINTRONIC DEVICES

STT-MRAM

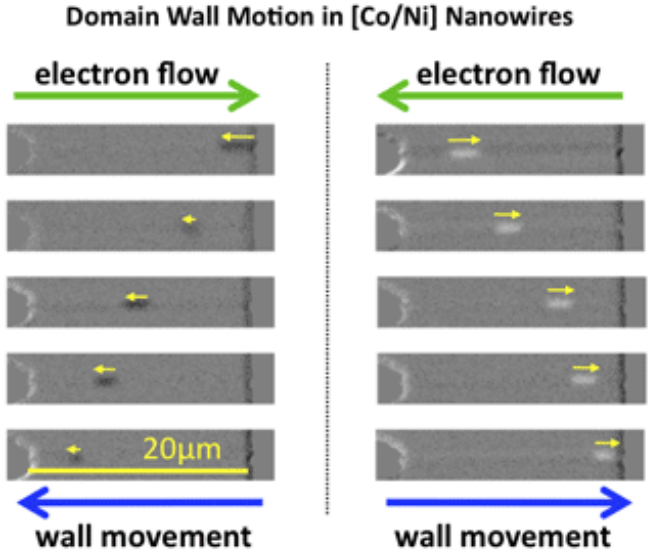


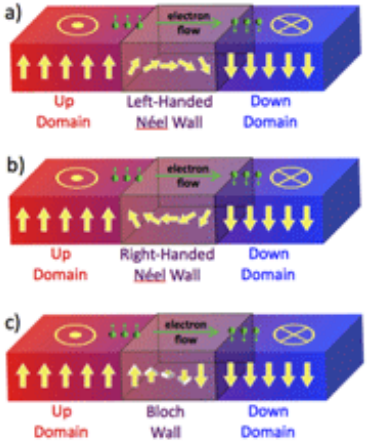




Operations	Write '1'('0')	Read
WL	V_{DD}	V_{DD}
BL	GND (V_{DD})	I_{sense}
SL	V_{DD} (GND)	GND

Domain Wall Motion Device





Possible wall structures in a magnetic nanowire

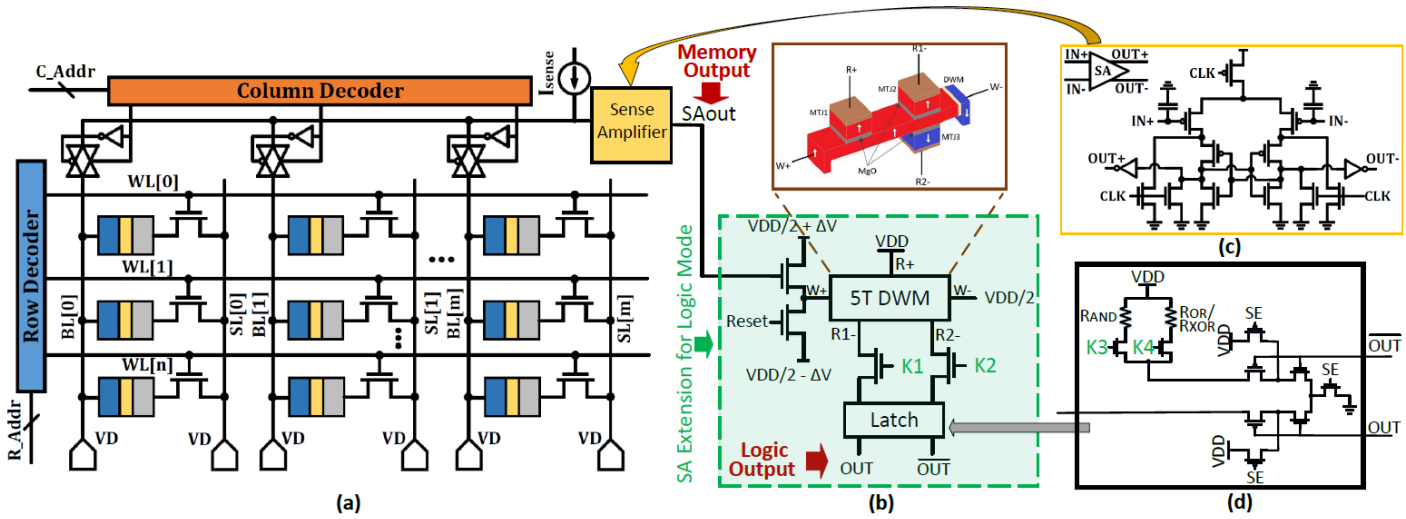
OUTLINE

- Motivation
- Post-CMOS Spintronic Devices
- In-Memory Processing Platform based on STT-MRAM
- Performance Evaluation
- Case Study I: In-memory Bulk Bitwise Vector Operation
- Case Study II: In-memory Data Encryption Engine

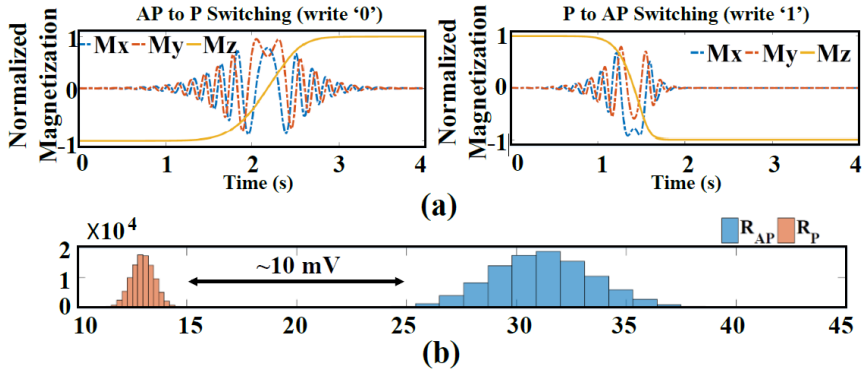
IN-MEMORY PROCESSING PLATFORM

Operations	Write '1'('0')	Read
WL	V_{DD}	V_{DD}
BL	GND (V_{DD})	I_{sense}
SL	V_{DD} (GND)	GND

- Dual mode architecture that perform both memory read-write and in-memory logic (AND/NAND, OR/NOR, XOR/XNOR).
- ***Memory Write***: To write data in a memory cell, the corresponding WL is activated using the row decoder. Then appropriate voltage difference is applied to the corresponding BL and SL using the voltage drivers.
- ***Memory Read***: The corresponding WL is activated using the row decoder and the corresponding BL is connected to the sense amplifier (SA) using the column decoder.
- ***Computing Mode***: We propose a sensing circuit design using 5T DWM device [1], as an extension to SA of memory array, to implement complete Boolean logic functions between any two cells in the memory array.



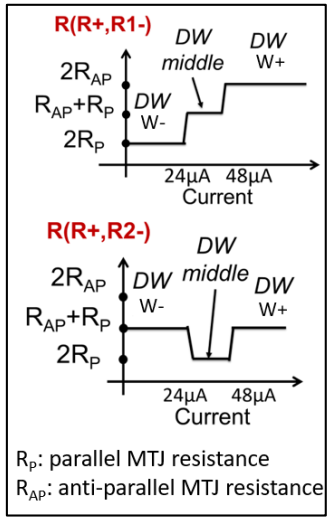
(a) HielM, (b) proposed sensing scheme, (c) Memory sense amplifier, (d) Differential Latch.

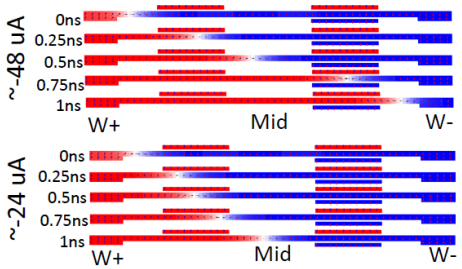


(a) Magnetization switching of STT-MRAM,
(b) The Monte-Carlo simulation result of memory read

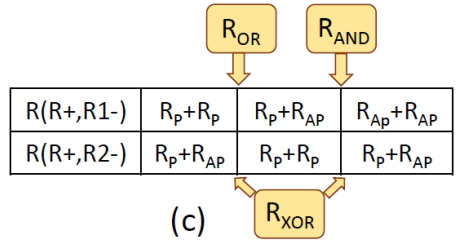
[1] F. Parveen et al. Hybrid polymorphic logic gate with 5-terminal magnetic domain wall motion device. In ISVLSI, pages 152–157. IEEE, 2017.

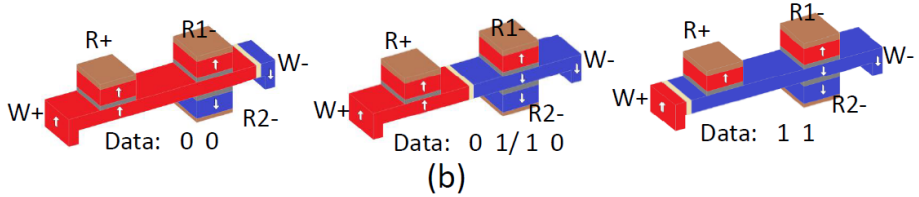
IN-MEMORY PROCESSING PLATFORM





Data	0 0	0 1/ 1 0	1 1
DW position	W-	Mid	W+



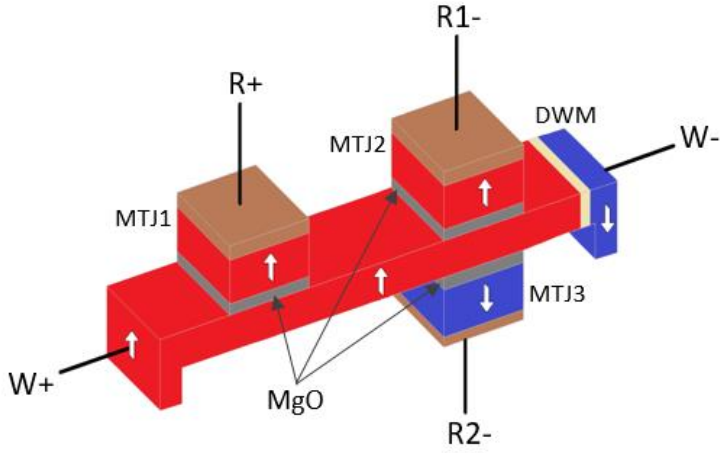


Data		1 1	0 1/ 1 0	0 0
DW Position		W+	Mid	W-
AND Keys: K1-K4=1010	R(R+,R1-)	$R_{AP}+R_{AP}$	R_P+R_{AP}	R_P+R_P
	compare to R_{AND}*	>	<	<
	OUT	1	0	0
OR Keys: K1-K4=1001	R(R+,R1-)	$R_{AP}+R_{AP}$	R_P+R_{AP}	R_P+R_P
	compare to R_{OR}**	>	>	<
	OUT	1	1	0
XOR Keys: K1-K4=0101	R(R+,R2-)	$R_{AP}+R_P$	R_P+R_P	R_P+R_{AP}
	compare to R_{XOR}**	>	<	>
	OUTbar	0	1	0

* R_{AND} = Between $2R_{AP}$ and R_P+R_{AP}

** R_{OR}/R_{XOR} = Between $2R_P$ and R_P+R_{AP}

- For a complete Boolean operation, the SA extension needs 3 subsequent stages- Reset, Compute and Sense.
- In Reset stage (Reset=1), the reset transistor is turned on for 1ns. A current of 48uA flows from W- to W+ terminals, which sets the DW back to its initial position at W- side.
- In Compute stage, two operands stored in the memory array are read in two consecutive cycles using the SA and applied to DWM device.
- In Sense stage, a small sensing current is injected through DWM device from R+ to R1- or from R+ to R2- terminals based on required logic implementation.



OUTLINE

- Motivation
- Post-CMOS Spintronic Devices
- In-Memory Processing Platform based on STT-MRAM
- Performance Evaluation
- Case Study I: In-memory Bulk Bitwise Vector Operation
- Case Study II: In-memory Data Encryption Engine

PERFORMANCE EVALUATION

Device to System Level Simulations:

Device Level:

Verilog-A model of 5T DWM device was developed to co-simulate with the interface CMOS circuits in SPICE to validate the functionality and evaluate performance of the proposed design. The STT-MRAM is simulated by solving LLG equation to model dynamics of MTJ free layer.

Circuit Level:

45nm North Carolina State University (NCSU) Product Development Kit (PDK) [1] library is used in SPICE to verify the proposed design and evaluate the performance.

System Level:

We employ the modified self-consistent NVSim [2] along with an in-house developed C++ code to verify the performance of memory.

[1] www:eda.ncsu.edu/wiki/FreePDK45
[2] X. Dong et al., ”NVSim: A circuit-level performance, energy, and area model for emerging non-volatile memory,” Springer, 2014, pp. 15-50.

PERFORMANCE EVALUATION

Memory Mode:

- The proposed STT-MRAM memory model shows the least write dynamic energy in comparison to other designs.
- It reduces the total leakage power compared to SRAM.
- It shows longer average latency compared to SRAM due to the longer write latency of magnetic memory storage.
- Its area overhead is 29.1% more than DRAM but still 37.51% less than SRAM design.

SRAM, DRAM AND PROPOSED STT-MRAM MEMORY MODEL VALIDATION AND COMPARISON FOR A SAMPLE 4MB MEMORY

	SRAM		DRAM		STT-MRAM	
Metrics	Write	Read	Write	Read	Write	Read
Average Latency (ns)	1.76		2.7		2.67	
Dynamic Energy (pJ)	1213	1483	917.8	967	826.15	870.042
Leakage Power (mW)	5316		185.5		830.847	
Area (mm^2)	10.613		4.702		6.632	

Computing Mode:

- The in-memory AND operation shows 65.3% and 81.32% lower energy consumption than Domain-Wall (DW) Racetrack based and MTJ based in-memory non-volatile AND gate implementations.
- Our design requires longer latency to compute the logic result than other designs

PERFORMANCE EVALUATION AND GATES

Performance	HieIM	DW Racetrack based [27]	MTJ based [28]	DW Racetrack based [29]	CMOS
Energy (fJ)	23.5	67.72	125.85	504.36	6.69
Speed (ns)	4	1.12	1.18	2.14	0.062

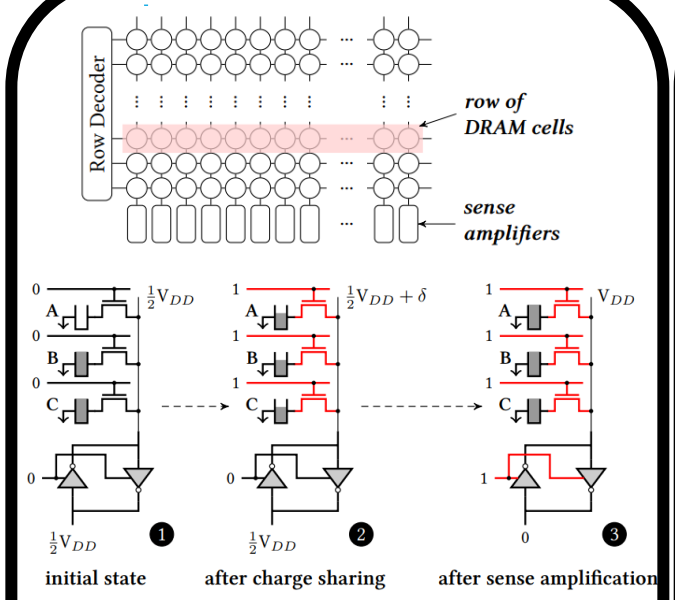
- The in-memory FA implementation is comparable to that of LPM based FA design. However, our design requires longer delay due to the read-and-write-back overhead of the intermediate results.

PERFORMANCE EVALUATION OF FA CELLS

Parameters	HieIM	HSM [30]	LPM [30]	Diode-GSHE [31]	CMOS [31]
Average Power (μW)	91.93	1354	85	15.6	49.4
Delay (ps)	26,000	269	877	10,000	1000

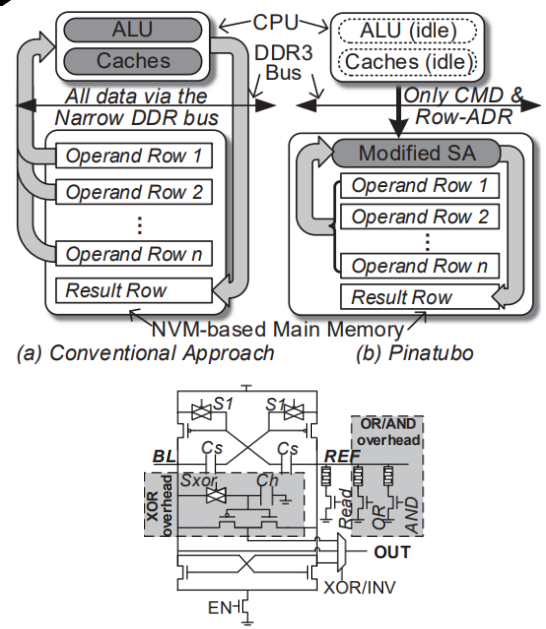
[27] K. Huang et al. Magnetic domain-wall racetrack memory-based nonvolatile logic for low-power computing and fast run-timereconfiguration. 2016.
[28] K. Huang et al. Stt-mram based low power synchronous non-volatile logic with timing demultiplexing. In NANOARCH, pages 31–36. ACM, 2014
[29] H.-P. Trinh et al. Magnetic adder based on racetrack memory. IEEE TCAS I, 60(6):1469–1477, 2013.
[30] A. Roohi et al. A tunable majority gate-based full adder using currentinduced domain wall nanomagnets. IEEE Trans. Magn., 52(8):1–7, 2016.
[31] Y. Zhang et al. Giant spin hall effect (gshe) logic design for lowpower application. In DATE, pages 1000–1005, 2015.

LETS FILL IT



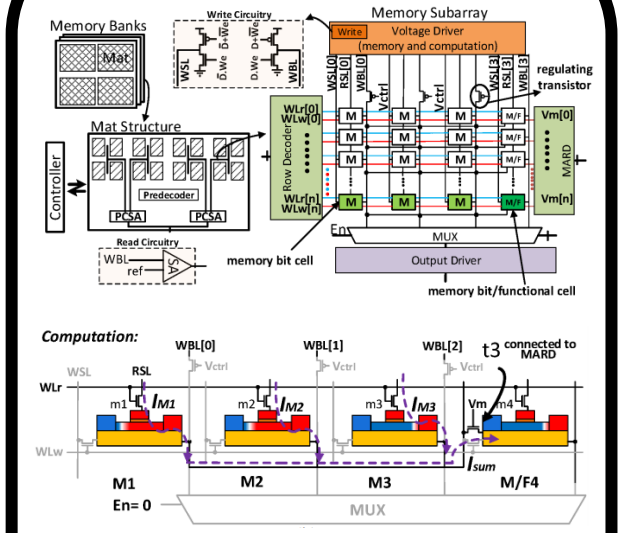
Ambit: DRAM-based

- Operand locality issue
- Original data overwritten
- Multi-Cycle operations
- Low area overhead
- Hardware-friendly
- exploiting the full internal DRAM bandwidth



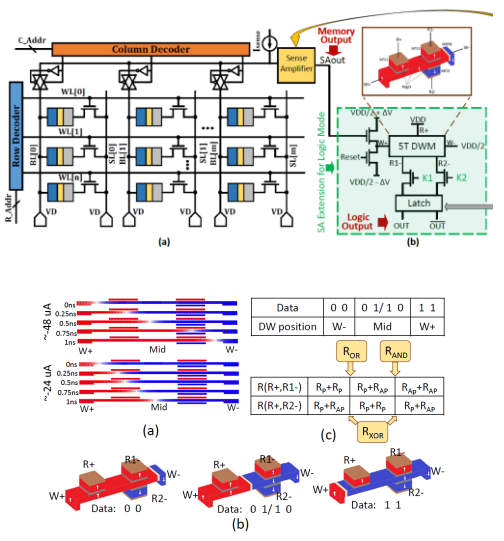
Pinatubo: NVM-based

- Operand locality issue
- Modified SA
- Medium area overhead
- support one-step multi-row operations
- General platform



RIMPA: DWM-based

- Operand locality issue
- Large area overhead
- Fast MG-based computation
- Ultra-low power



HieIM: MRAM-based

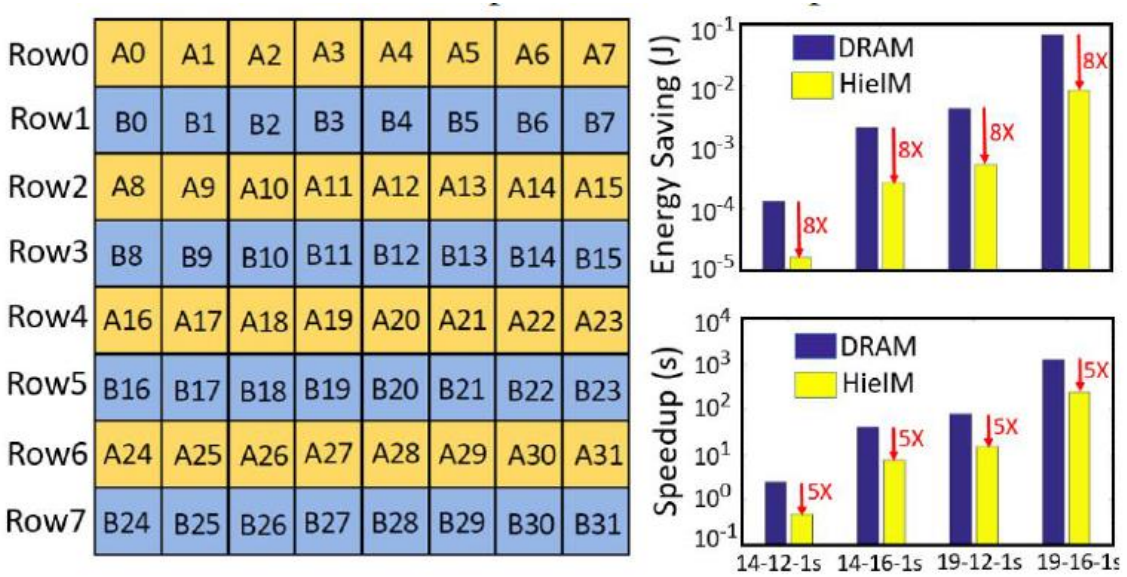
- Long Latency
- Modified SA
- Medium area overhead
- Ultra-low power
- No operand locality issue

OUTLINE

- Motivation
- Post-CMOS Spintronic Devices
- In-Memory Processing Platform based on STT-MRAM
- Performance Evaluation
- Case Study I: In-memory Bulk Bitwise Vector Operation
- Case Study II: In-memory Data Encryption Engine

CASE STUDY I: IN-MEMORY BULK BITWISE VECTOR OPERATION

- Four different vector datasets [1] have been used. Here, a dataset '19-16-1s' refers to a vector dataset with vector length=2^{19}, number of vectors=2^{16}, and AND/OR operation is done between2^{1} rows.
- Each compute (AND/OR) operation has been carried out using 4 consecutive clock cycles (1ns each).
- HieIM offers ~8× energy saving and ~5× speed up compared to that using Ambit-DRAM based in-memory computing platform [2].



Data mapping for performing vector operation between two 32 bit vectors using an 8*8 STT-MRAM array

[1] S. Li et al. Pinatubo: a processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories. In DAC, pages 1– 6. IEEE, 2016.
[2] V. Seshadri et al. Fast bulk bitwise and and or in dram. IEEE Computer Architecture Letters, 14(2):127–131, 2015.

OUTLINE

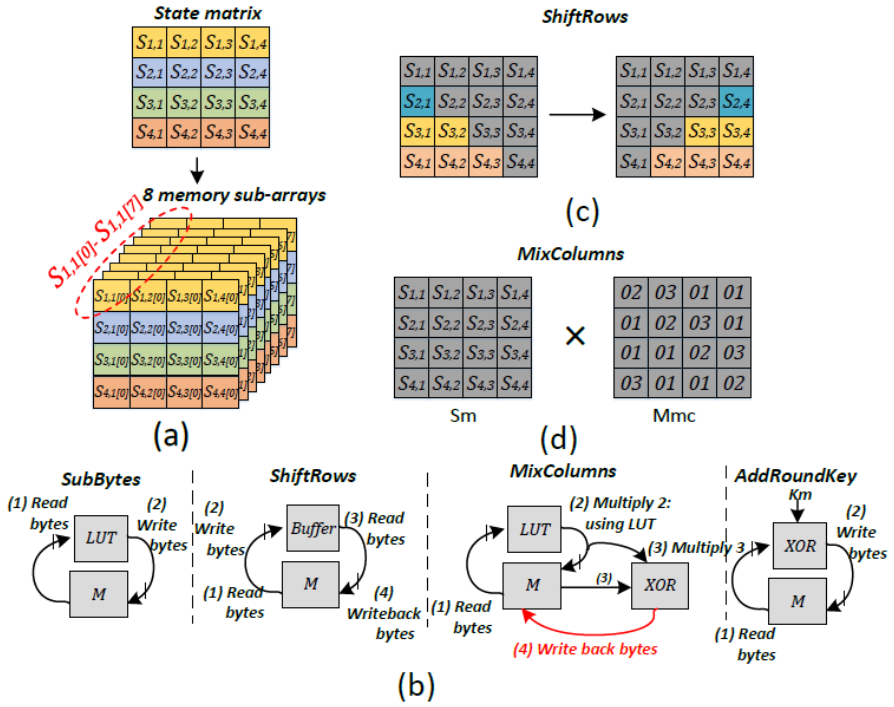
- Motivation
- Post-CMOS Spintronic Devices
- In-Memory Processing Platform based on STT-MRAM
- Performance Evaluation
- Case Study I: In-memory Bulk Bitwise Vector Operation
- Case Study II: In-memory Data Encryption Engine

CASE STUDY II: IN-MEMORY DATA ENCRYPTION ENGINE

- Advanced Encryption Standard (AES) has been used to employ in-memory data encryption engine using HieIM.
- HieIM can achieve 51.5% and 68.9% lower energy consumption compared to CMOS-ASIC and CMOL based implementations, respectively.
- HieIM occupies ~3.5× less area compared to baseline DW-AES.
- Note that, Baseline DW AES [36] requires lower number of cycles due to intrinsic shift operation and multi-bit data storage of DWM racetrack devices.

AES PERFORMANCE

Platforms	Energy (nJ)	Cycles	Area (μm^2)
GPP [37]	460	2309	2.5e+6
ASIC [41]	6.6	336	4400
CMOL[42]	10.3	470	320
Baseline DW [36]	2.4	1022	78
Pipelined DW [36]	2.3	2652	83
Multi-issue DW [36]	2.7	1320	155
HieIM	3.2	1620	21.8



(a) Data Organization, (b) Data Mapping of four AES transformations, (c) ShiftRows transformation, (d) MixColumn transformation.

[36] Y. Wang et al. Dw-aes: a domain-wall nanowire-based aes for high throughput and energy-efficient data encryption in non-volatile memory.IEEE TIFS, 11(11):2426–2440, 2016.
[37] K Malbrain. Byte-oriented-aes: a public domain byte-oriented implementation of aes in c, 2009.
[41] S. Mathew et al. 340 mv–1.1 v, 289 gbps/w, 2090-gate nanoaes hardware accelerator with area-optimized encrypt/decrypt gf (2 4) 2 polynomials in 22 nm tri-gate cmos. IEEE JSSC, 50(4):1048–1058, 2015.
[42] Z Abid et al. Efficient cmol gate designs for cryptography applications. IEEE TNANO, 8:315–321, 2009.

CONCLUSION

- In this work, we develop a new in-memory processing architecture based on STT-MRAM called HeiIM, which could be used as both non-volatile memory and reconfigurable in-memory logic.
- HeiIM offers several significant features as non-volatility, in-memory logic, operation with high data mapping flexibility, low dynamic power consumption and high packing density.
- The in-memory AND operation itself shows 65.3% and 81.32% lower energy consumption than Domain-Wall (DW) Racetrack based and MTJ based in-memory non-volatile AND implementations.
- In-memory bulk bitwise Boolean vector logic (AND/OR) operation for different vector datasets ~8× energy saving and ~5× speed up compared to that using DRAM based in-memory computing platform.
- We further have employed in-memory data encryption engine using AES algorithm, which shows 51.5% and 68.9% lower energy consumption compared to CMOS-ASIC and CMOL-based implementations, respectively.

THANKS