Security Challenges and Opportunities for Processing-In-Memory

Tanvir Arafin: Morgan State University
Xueyan Wang: Beihang University
Zhaojun Lu: Huazhong University of Science and Technology
Qian Xu and **Gang Qu**: University of Maryland, College Park

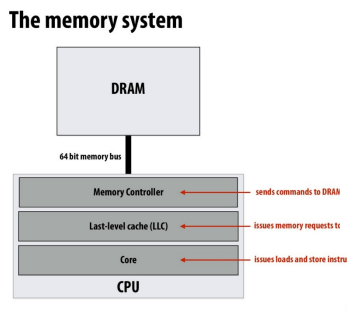
Electronic Design Process Symposium (EDPS)
November 4, 2021

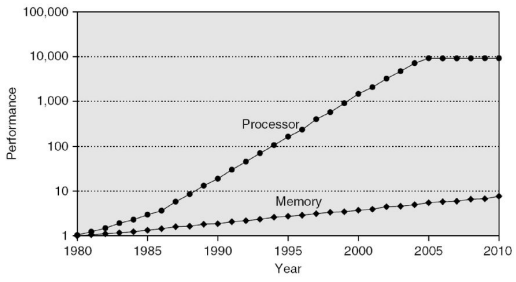
MeshSec Lab

HSTTC
IEEE Hardware Security & Trust Technical Committee

The Walls in High Performance

- Memory/Bandwidth wall (1990's)





Processor mem accesses/sec vs DRAM accesses/sec

https://slideplayer.com/slide/14465936/

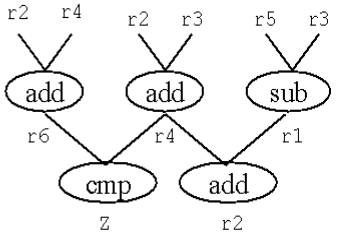
MeshSec Lab

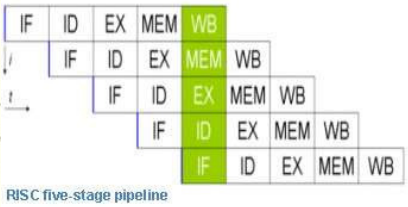
Dr. Gang Qu (gangqu@umd.edu) 2

The Walls in High Performance

- Memory/Bandwidth wall (1990's)
- ILP wall

```
add   r2,r4,r6
add   r2,r3,r4
sub   r5,r3,r1
add   r4,r1,r2
cmp   r6,r4
```





http://www.cs.cmu.edu/afs/cs/academic/class/15828-s98/lectures/0112/sld012.htm

https://www.edn.com/future-of-computing-part-3-the-ilp-wall-and-pipelines/

3

The Walls in High Performance

- Memory/Bandwidth wall (1990's)
- ILP wall
- Power wall
- Big data
- AI/ML
- IoT
- ...

Dynamic power ∝ capacitive load × voltage² × frequency

Static power: transistors burn power even when inactive due to leakage

High power = high heat

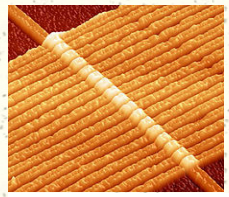
Power is a critical design constraint in modern processors

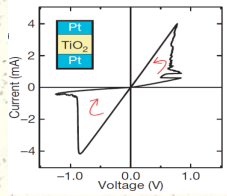
	TDP
Intel Core i7 (in this laptop):	45W
Intel Core i7 2700K (fast desktop CPU):	95W
NVIDIA GTX 780 GPU	250W
Mobile phone processor	½ - 2W
World's fastest supercomputer	megawatts
Standard microwave oven	700W

http://15418.courses.cs.cmu.edu/spring2017/lecture/whyparallelism/slide_035

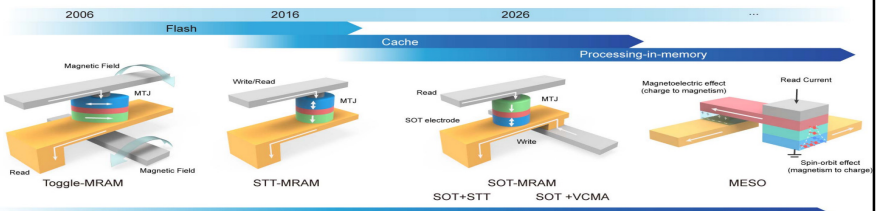
MeshSec Lab

Non-Volatile Memory

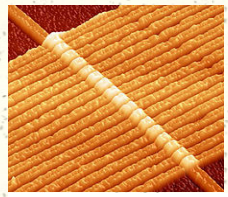


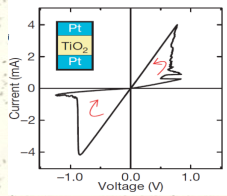


- NVM: retains stored information after power is removed.
  - Magnetic-core memory, flash memory
  - Ferroelectric RAM; Magnetoresistive RAM, spin-transfer torque (STT), memristive; Phase-Change RAM, FeFET memory …

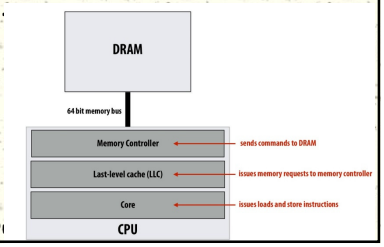


Non-Volatile Memory



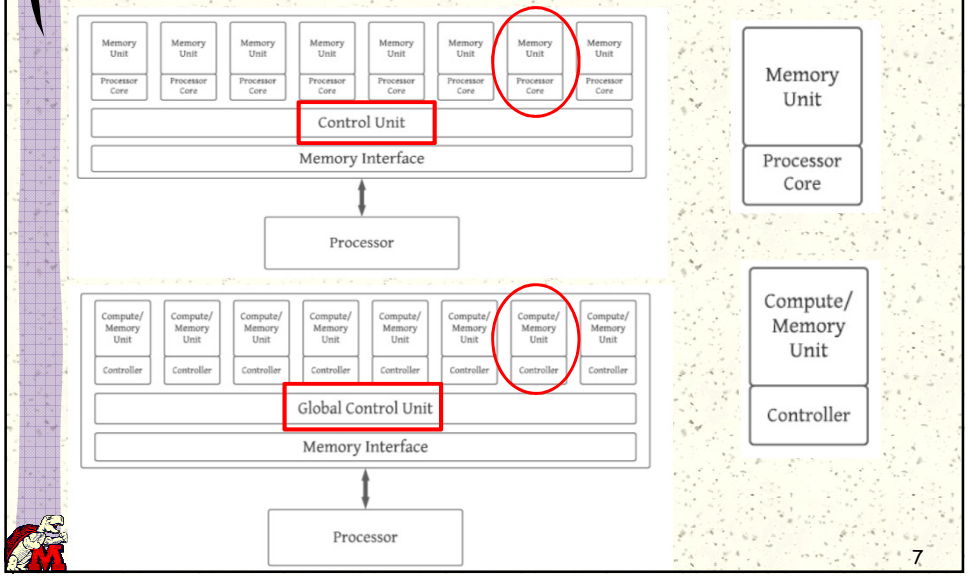


- NVM: retains stored information after power is removed.
  - Magnetic-core memory, flash memory
  - Ferroelectric RAM; Magnetoresistive RAM, spin-transfer torque (STT), memristive; Phase-Change RAM, FeFET memory …
- Addressing the power wall
- What about the memory/ bandwidth wall?
- What about the ILP wall?



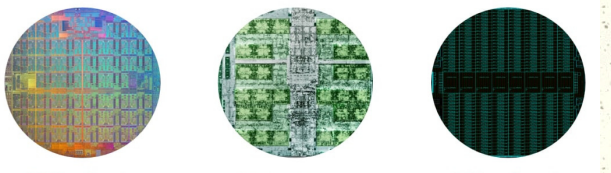
MeshSec Lab
Dr. Gang Qu (gangqu@

Processing Near-/In-Memory



7

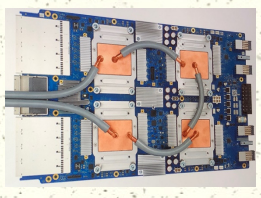
PIM-based IPUs



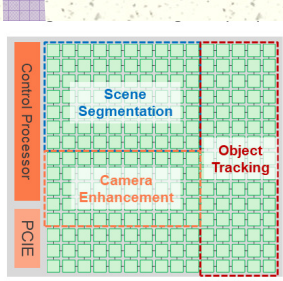
CPU = Scalar
Designed for office apps
Has evolved for the web

GPU = Vector
Designed for graphics
Has evolved for HPC

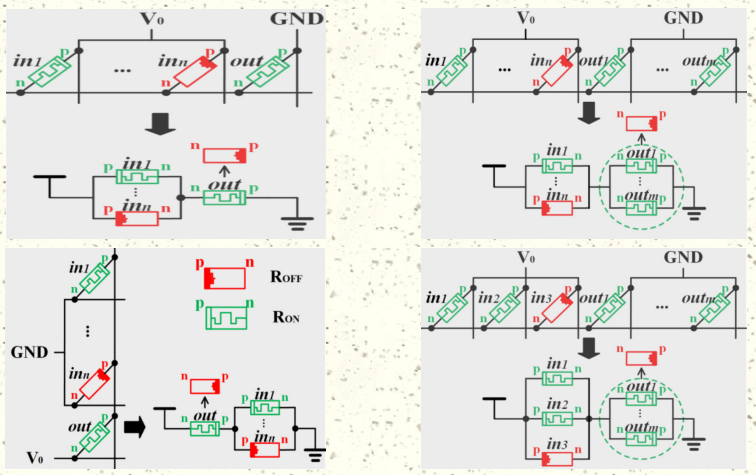
IPU = Graph
Designed for intelligence
The future of computing



Tensor Processing Unit
AI accelerator



RRAM-based Logic Operations

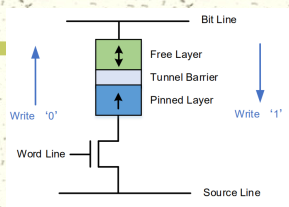


MeshSec Lab

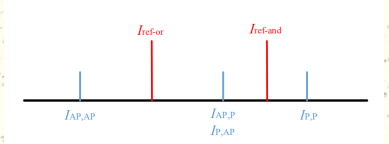
Z. Lu et al. RIME: A Scalable and Energy-Efficient Processing-In-Memory Architecture for Floating-Point Operations. *ASP-DAC 2020.*

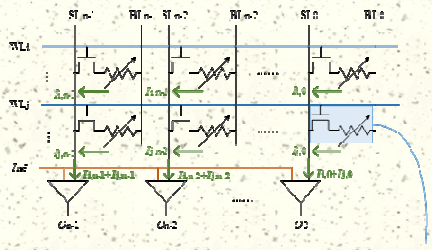
PIM with STT-MRAM

- A typical STT-MRAM cell



- With different reference current, logics (AND, OR, …) can be implemented.





MeshSec Lab

Dr. Gang Qu (gangqu@umd.edu) 10

Common PIM Operations

Type	Operation
Memory	Bulk Load, Store Row-clone [27]
Arithmetic	Integer addition, multiplication [39] Floating point addition [1] Integer increment [1] Integer min [1, 8]
Matrix Operation	Dot product, Euclidean Distance [1] Convolution [7, 30]
Logic	AND, OR, NOT [29, 36] IMPLY [38] Minority [32]
Algorithmic	Hash table manipulation [1] Pointer Chasing [10]

MeshSec Lab

T. Arafin and Z. Lu: Security Challenges of processing-in-memory systems, *GLSVLSI 2020.*

Hardware Security Lessons

- Cache/memory side-channel
- Fault injection
  - Rowhammer
  - Clkscrew
  - VoltJockey
- Memory leak
  - ThunderClap

Security must be considered during PIM design and implementation

MeshSec Lab

Dr. Gang Qu (gangqu@umd.edu) 12

Vulnerabilities in PIM

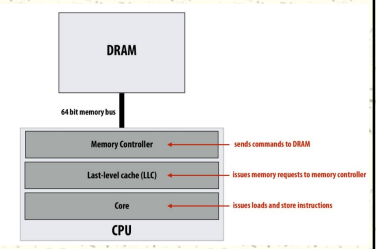
- Memory/storage
  - Information leak: malicious read, physical feature, non-volatile
  - Memory corruption: exploit vulnerable code to access out-of-bound memory.
  - Denial of service: use contention to occupy the PIM controller
- PIM
  - Cache/EM side-channel, fault injection, information leak …

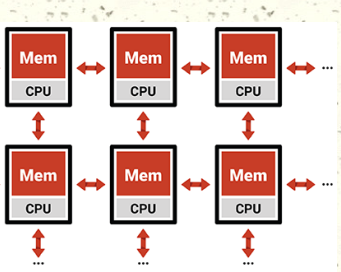
MeshSec Lab

T. Arafin et al.: A Survey on memristor modeling and security applications. ISQED 2015.

Security Opportunities by PIM

- Data/process privacy
- Secure "enclave"
- Reduced side-channel leakage
- Security related operations
  - Accelerator (e.g. PQC)
  - Lightweight security primitive



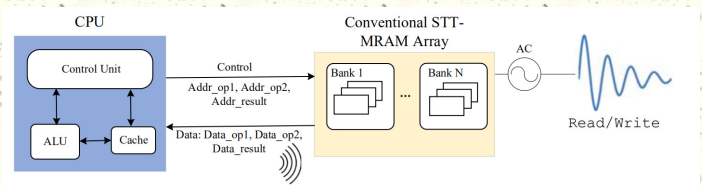


MeshSec Lab

Dr. Gang Qu (gangqu@umd.edu) 14

Reduced Side-Channel Leakage

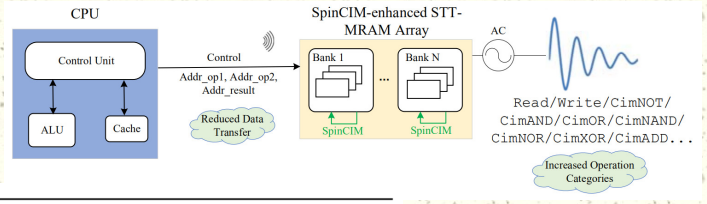
- Traditional data transfer pattern



Operation	Delay (*ns*)	Energy (*fJ*)
Read '1'	0.6	8.611
Read '0'	0.6	7.669
Write '1'	4.4	233.300
Write '0'	3.3	191.400

Reduced Side-Channel Leakage

- Data transfer pattern in SpinPIM

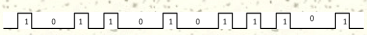


Operation	Delay (*ns*)	Energy (*fJ*)
Read '1'	0.63	22.69
Read '0'	0.67	23.85
Write '1'	4.40	244.64
Write '0'	3.30	202.70
CimNOT	0.60	22.20
CimAND	0.55	22.30
CimOR	0.53	22.90
CimNAND	0.45	18.89
CimNOR	0.45	21.00
CimXOR	0.53	26.34
CimADD	0.53	26.32

Increased operation types
Reduced data transfers

Memristor based Authentication

- Key Concepts
  - Pulse-train
    - Shared secret is (converted to) a sequence of small/short pulses



  - Registration
    - Bob choses a random sequence of pulses (pulse-train)
    - Bob applies this pulse-train to a device Alice own
    - Alice saves the device's response for later authentication
  - Verification
    - Alice verifies Bob by matching the device's response to an applied pulse-train

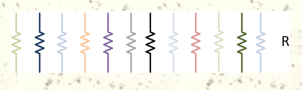
MeshSec Lab

M.T. Arafin and G. Qu. Memristors for secret sharing-based lightweight authentication, ICCAD'15, TVLSI'18

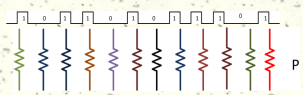
Memristor based Authentication

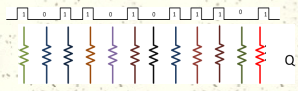
Single user (Bob) authentication (ICCAD'15)

- Registration
  - For the i^{th} cycle in the pulse train (Bob's password), store it in RRAM P_i
    - Assign R_i a random initial resistance level
    - **Copy State** (R_i, P_i)
    - Apply the i^{th} cycle in the pulse train to P_i



- Verification
  - For the i^{th} cycle in the pulse train
    - **Copy State** (R_i, Q_i)
    - Apply the i^{th} cycle in the pulse train to Q_i
    - Q_state = **Read State** (Q_i)
    - P_state = **Read State** (P_i)
    - If (P_state and Q_state are the same) the i^{th} cycle in the pulse train is correct
    - Else the i^{th} cycle in the pulse train does not match Bob's
    - **Pre-condition** (P_i, SPC-P_state)





MeshSec Lab

Dr. Gang Qu (gangqu@umd.edu)

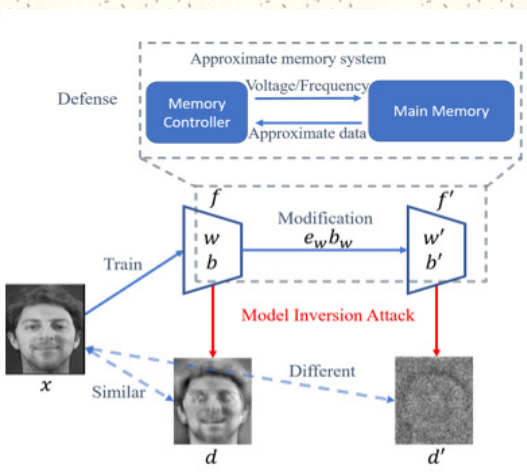
What is Model Inversion Attack?

M. Fredrikson et al, Model Inversion Attacks that Exploit Confidence I. Information and Basic Countermeasures, CCS, 2015.

MeshSec Lab

MIDAS Approach

Model Inversion Defense with Approximate memory system



Q. Xu et al. Model Inversion Defense using an Approximation Memory System. AsianHOST 2020.

Conclusions

- High performance
- Low power
- Security, privacy, trust

- Non-volatile memory and PIM
  - New attacking surface
  - New security primitives
  - PIM is not for all applications

Thank you!

Tanvir Arafin: Morgan State University
Xueyan Wang: Beihang University
Zhaojun Lu: Huazhong UST
Qian Xu: UMD