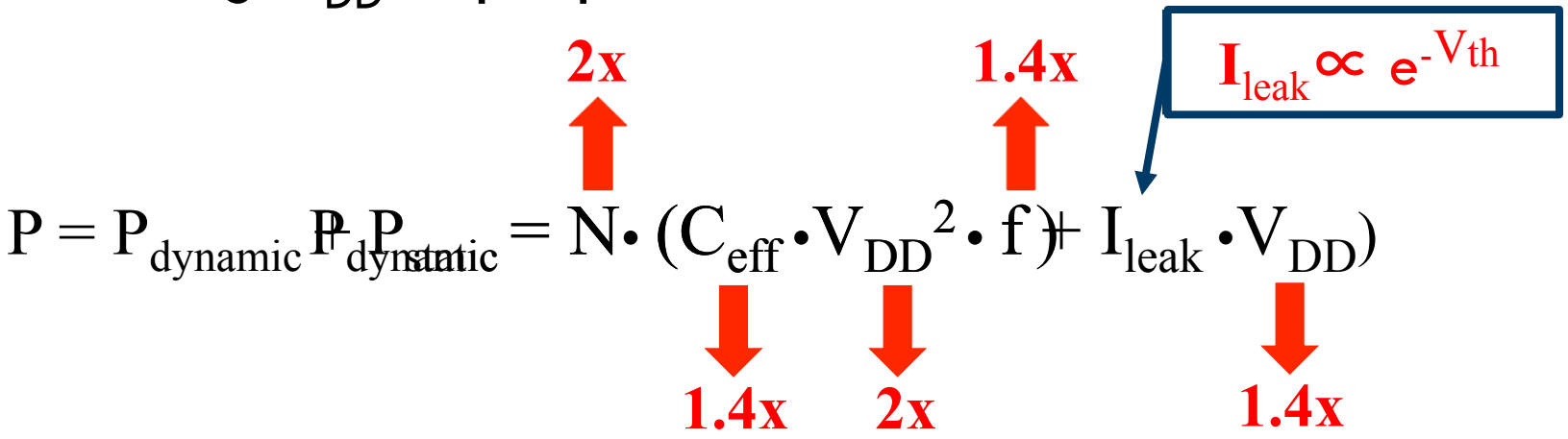
RESISTIVE COMPUTATION: AVOIDING THE POWER WALL WITH LOW-LEAKAGE, STT-MRAM BASED COMPUTING

Xiaochen Guo, Engin Ipek, and Tolga Soyata

Rochester Computer Systems Architecture Laboratory

Multicore Scaling Limited by Power

- Traditional MOSFET scaling theory relies on reducing V_{DD} in proportion to device dimensions



- V_{DD} has scaled very slowly since 90nm
- Multicore scaling severely challenged by power

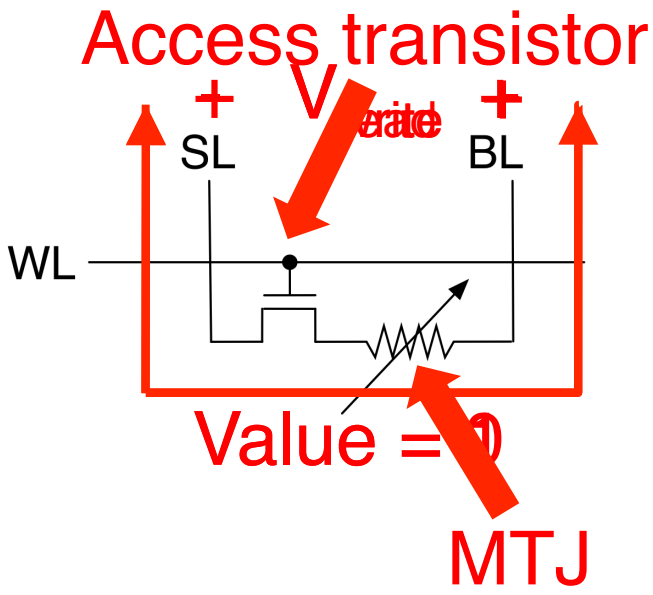
Our Approach: Resistive Computation

- Opportunity: spin-torque transfer magnetoresistive RAM (STT-MRAM)
  - Near-zero leakage power
  - Low-energy read operation

- Goal: selectively migrate on-chip storage and combinational logic to STT-MRAM to reduce power
  - On-chip storage
    - Caches, TLBs, RF, queues
  - Combinational logic
    - Lookup-table (LUT) based computing

STT-MRAM

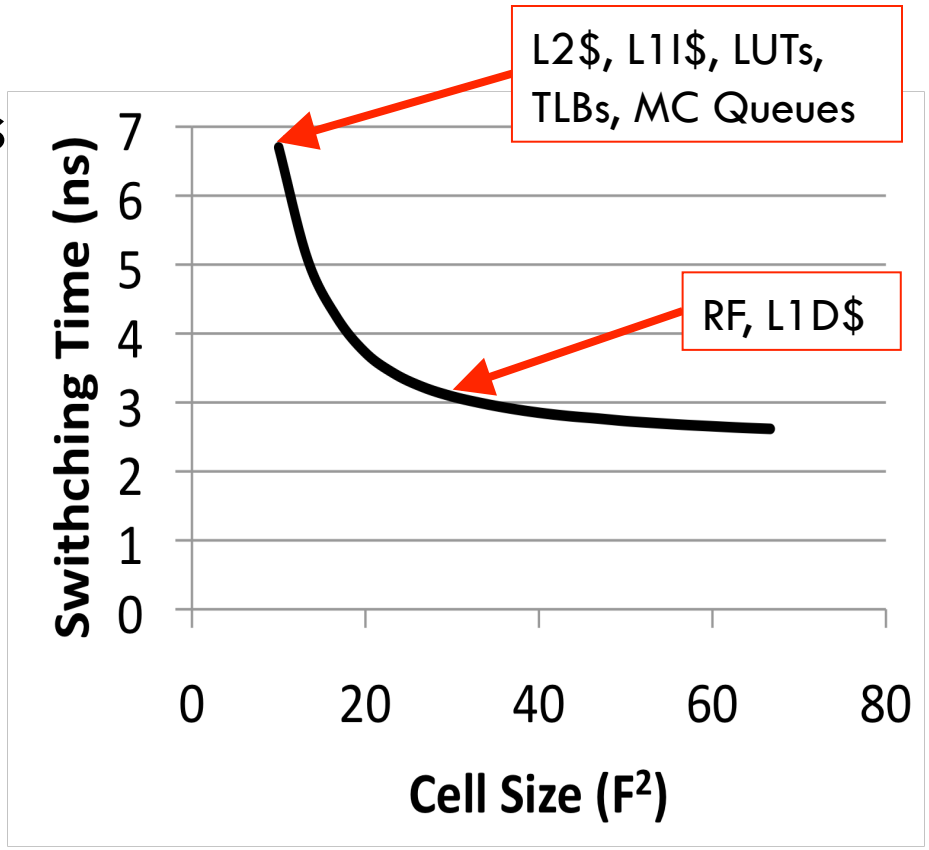
- Desirable properties
  - CMOS compatibility
  - Read speed as fast as SRAM
  - Density comparable to DRAM
  - Unlimited write endurance



- Key challenge: expensive writes
  - Long switching latency (6.7ns @ 32nm)
  - High switching energy (0.3pJ/bit @ 32nm)

Switching Time vs. Cell Size

- Faster switching with wider access transistors
  - + Faster writes
  - – Slower reads
  - – Lower density
  - – Higher read energy



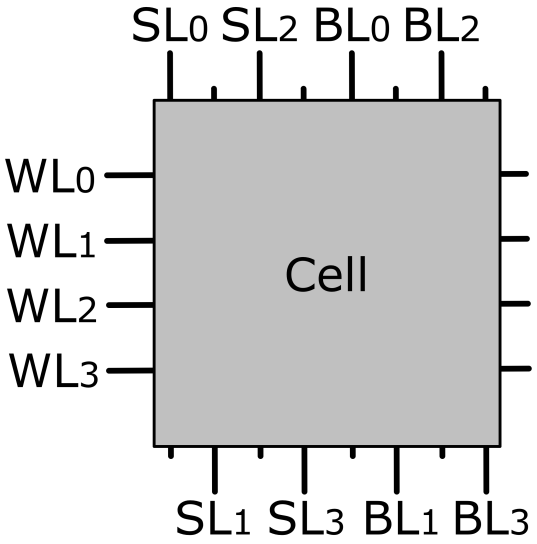
Fundamental Building Blocks

RAM Arrays and Lookup Tables

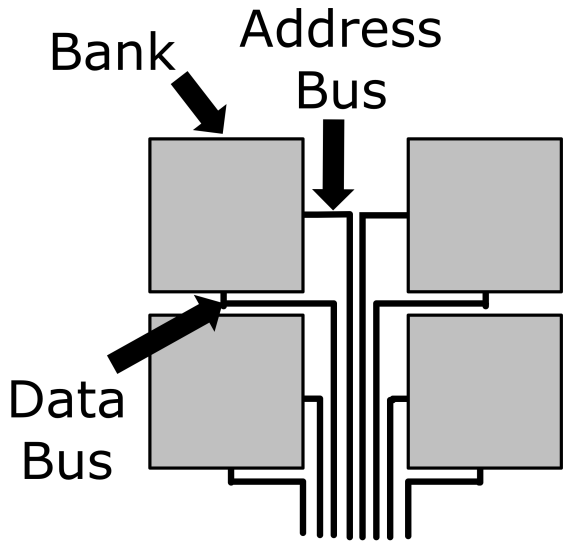
STT-MRAM Arrays

- Problem: low write throughput

Multiporting



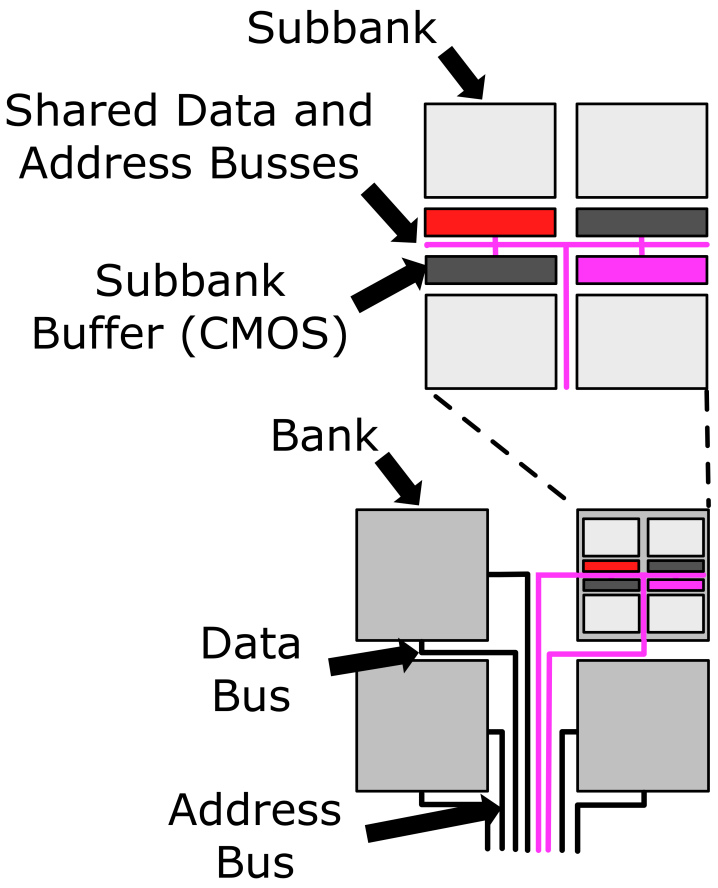
Banking



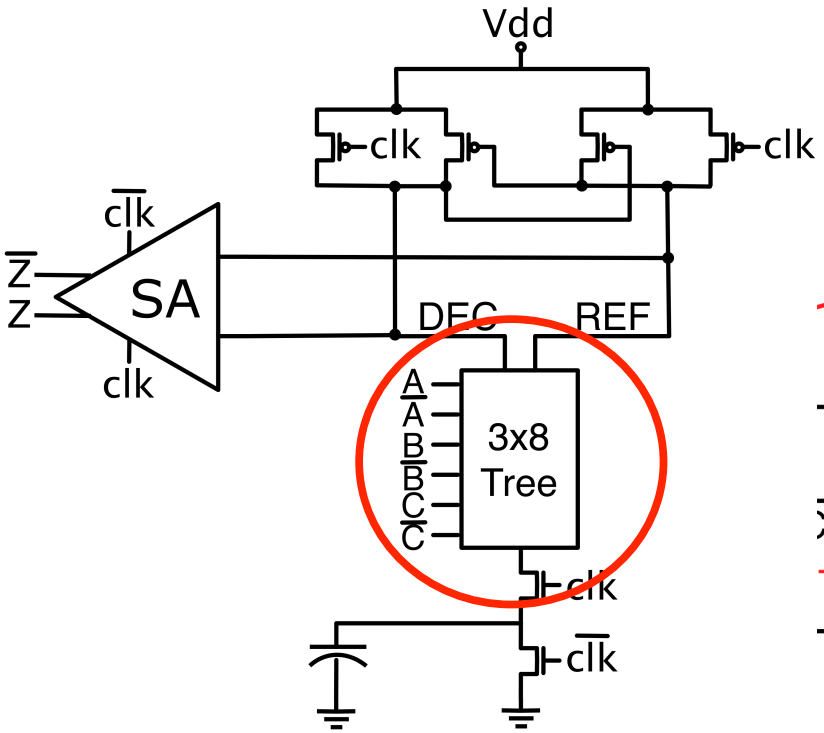
- Existing solutions incur high overheads to sustain adequate write throughput in STT-MRAM arrays

STT-MRAM Arrays

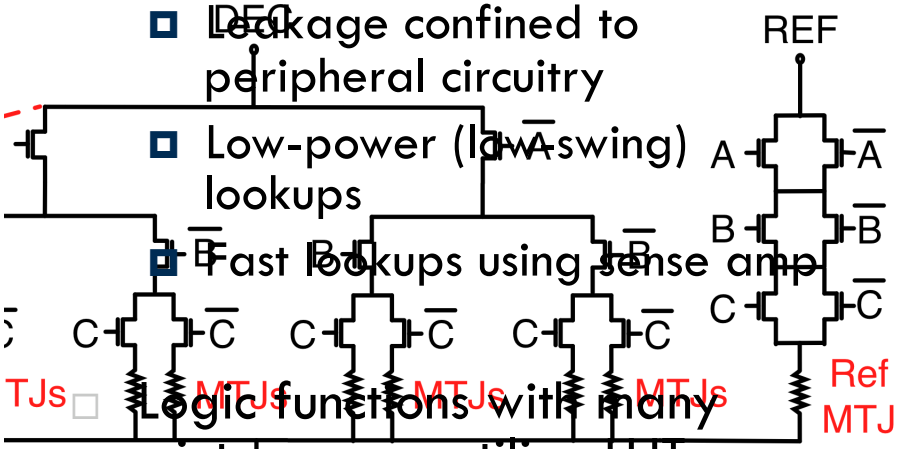
- CMOS subbank buffers
  - Latch in addr/data and release H-tree; complete write locally
  - Allow forwarding from ongoing writes
  - Facilitate local differential writes
- Reads access subbank via exclusive read port



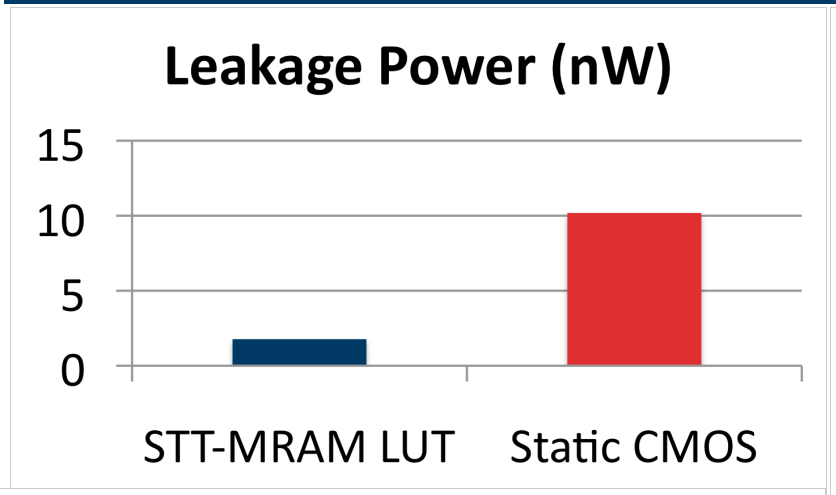
STT-MRAM LUTs [Suzuki09, Matsunaga08]

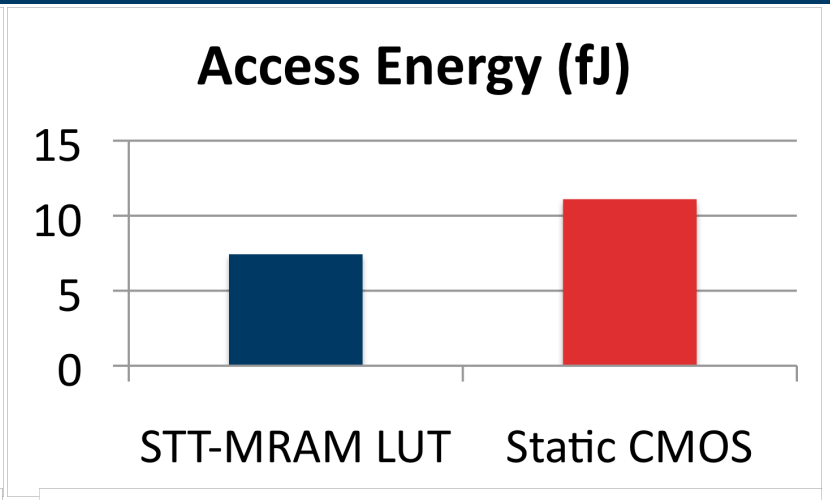


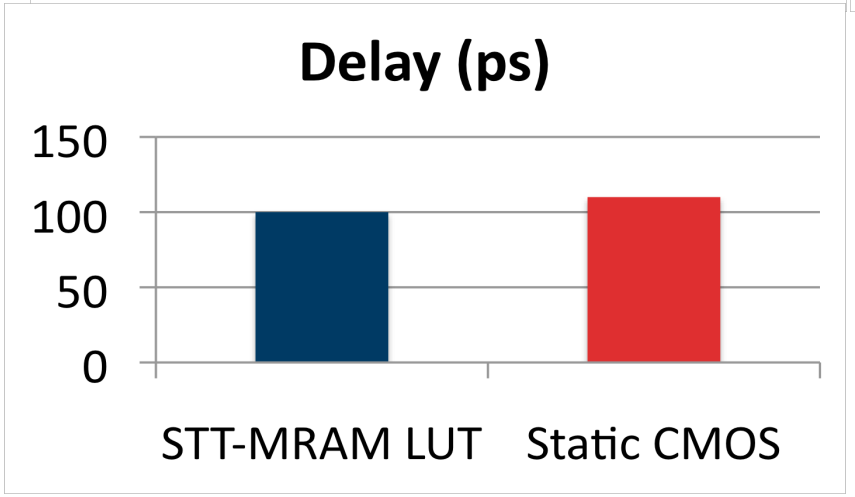
- Store truth tables of logic functions directly in STT-MRAM
- Benefits
  - Leakage confined to peripheral circuitry
  - Low-power (low-swing) lookups
  - Fast lookups using sense amp
- Logic functions with many minterms can utilize LUTs effectively

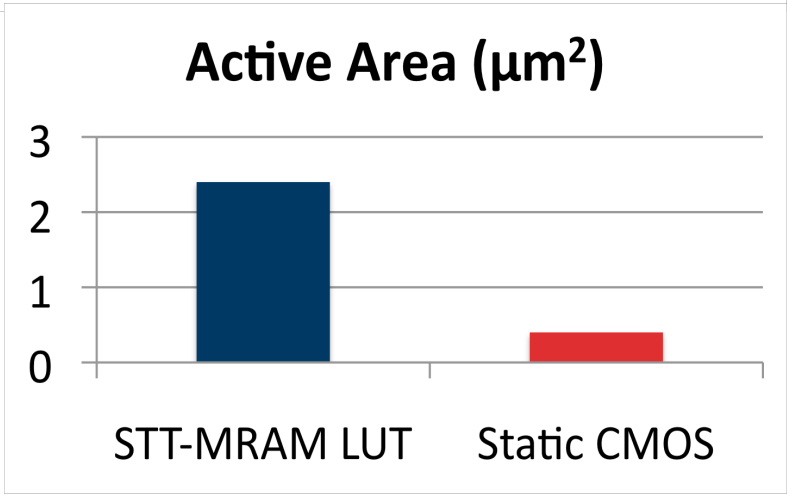


Case Study: 3-bit Adder







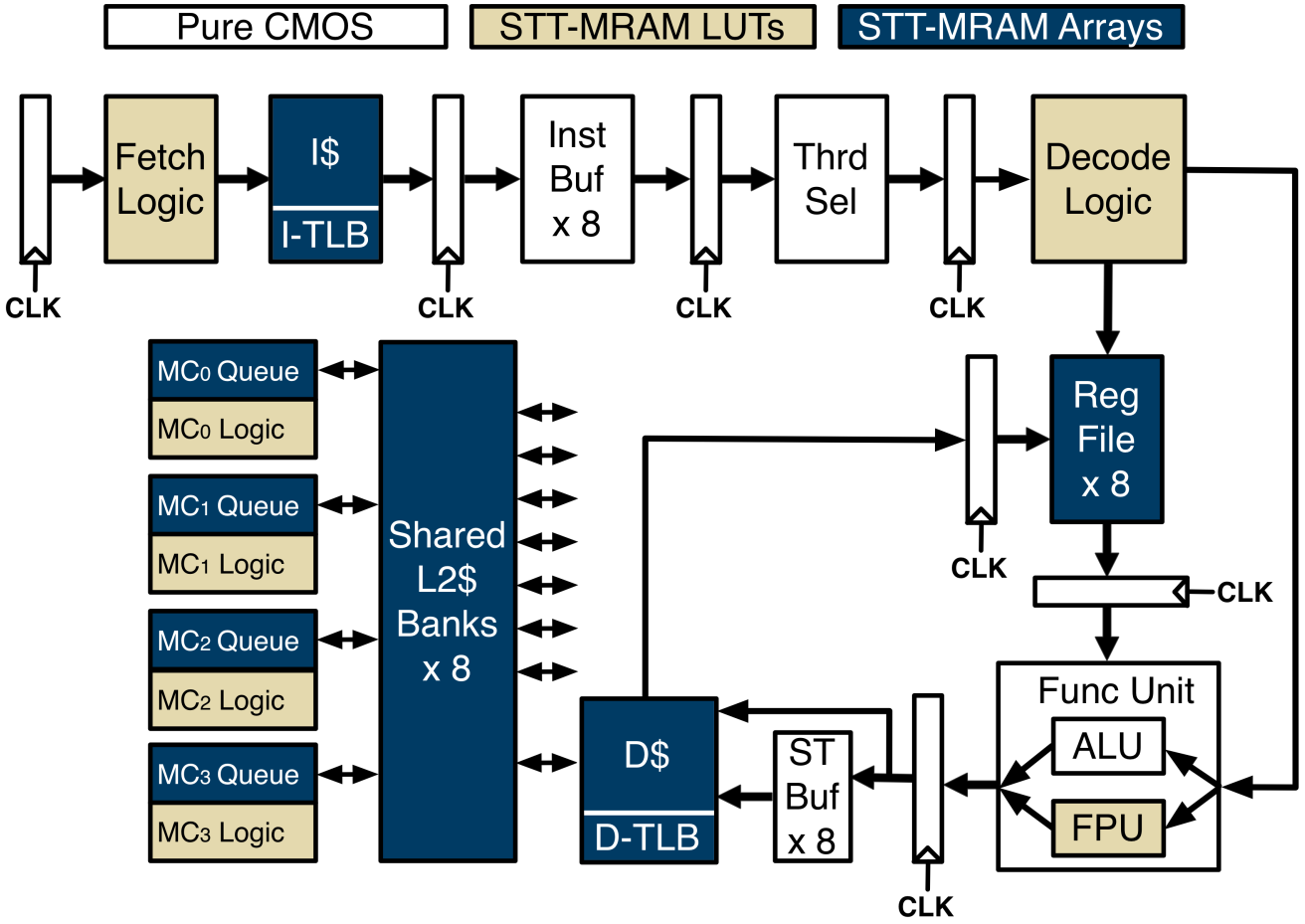


Pipeline Organization

Hybrid CMT Pipeline

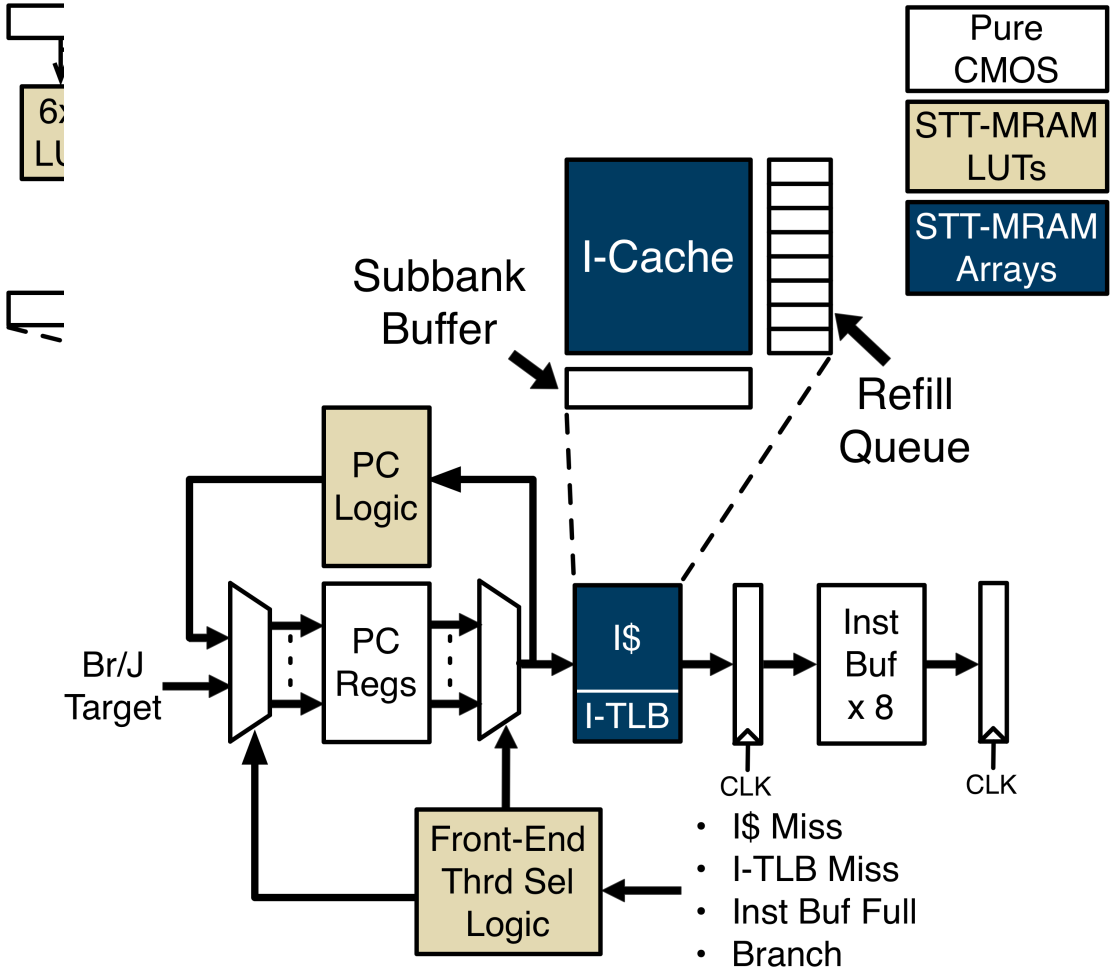
Small arrays and simple logic in CMOS

Large arrays and complex logic in STT-MRAM



Front End

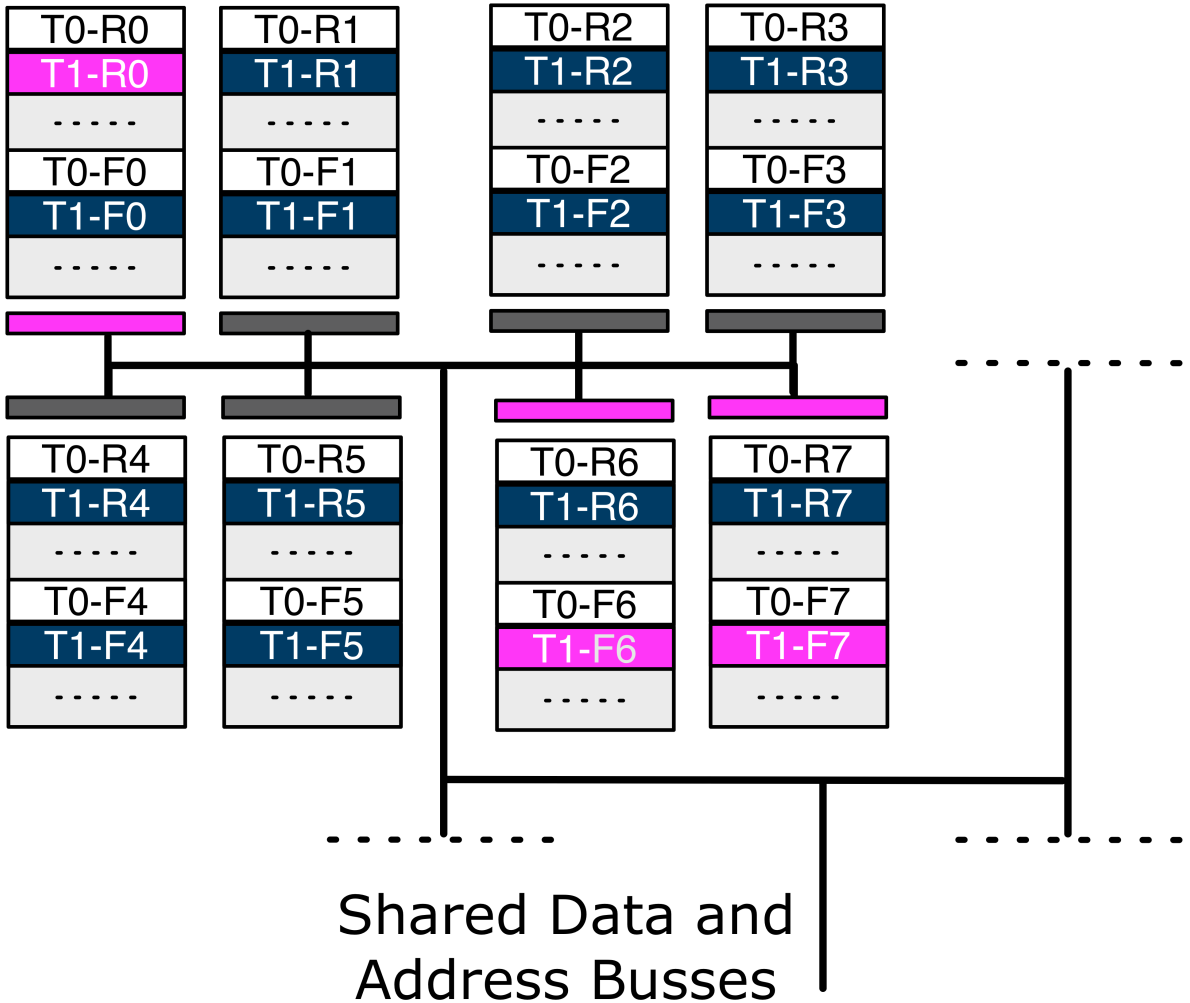
- LUT-based carry-select adder to compute PC+4
- LUT-based front-end thread selection logic
- SRAM-based refill queue to avoid I$ conflicts
- Predecode and back-end thread selection with MRAM-related stall conditions



Register File

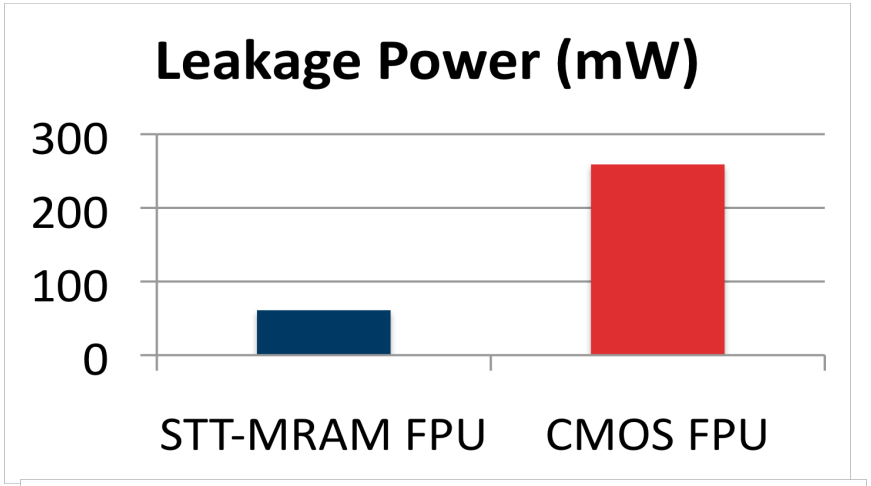
Architectural registers of all threads aggregated in a unified STT-MRAM array to amortize subbank buffers

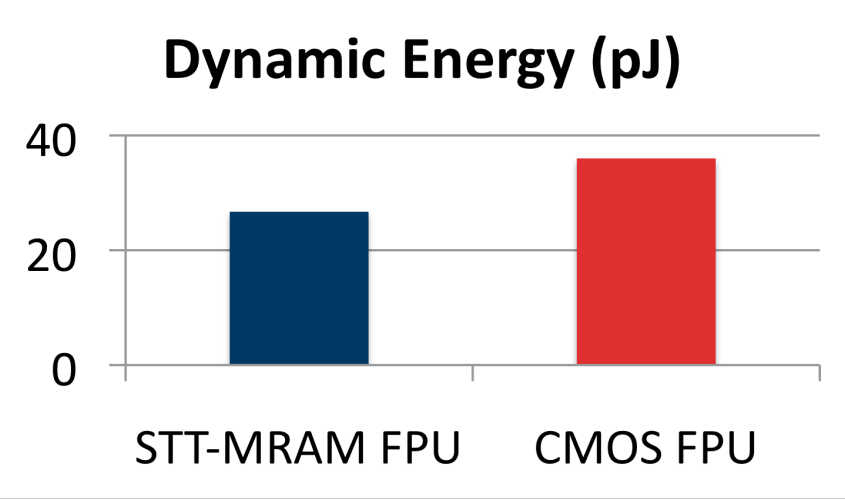
Registers of a single thread striped across subbanks to reduce subbank buffer conflicts

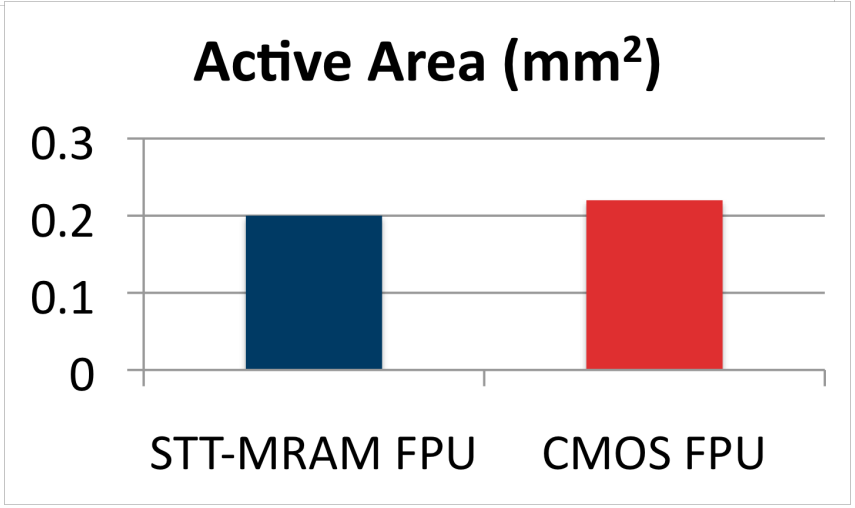


Floating-Point Unit

	STT-MRAM FPU	CMOS FPU
Add, Sub, Mult	24 cycles	12 cycles
Div	64 cycles	64 cycles





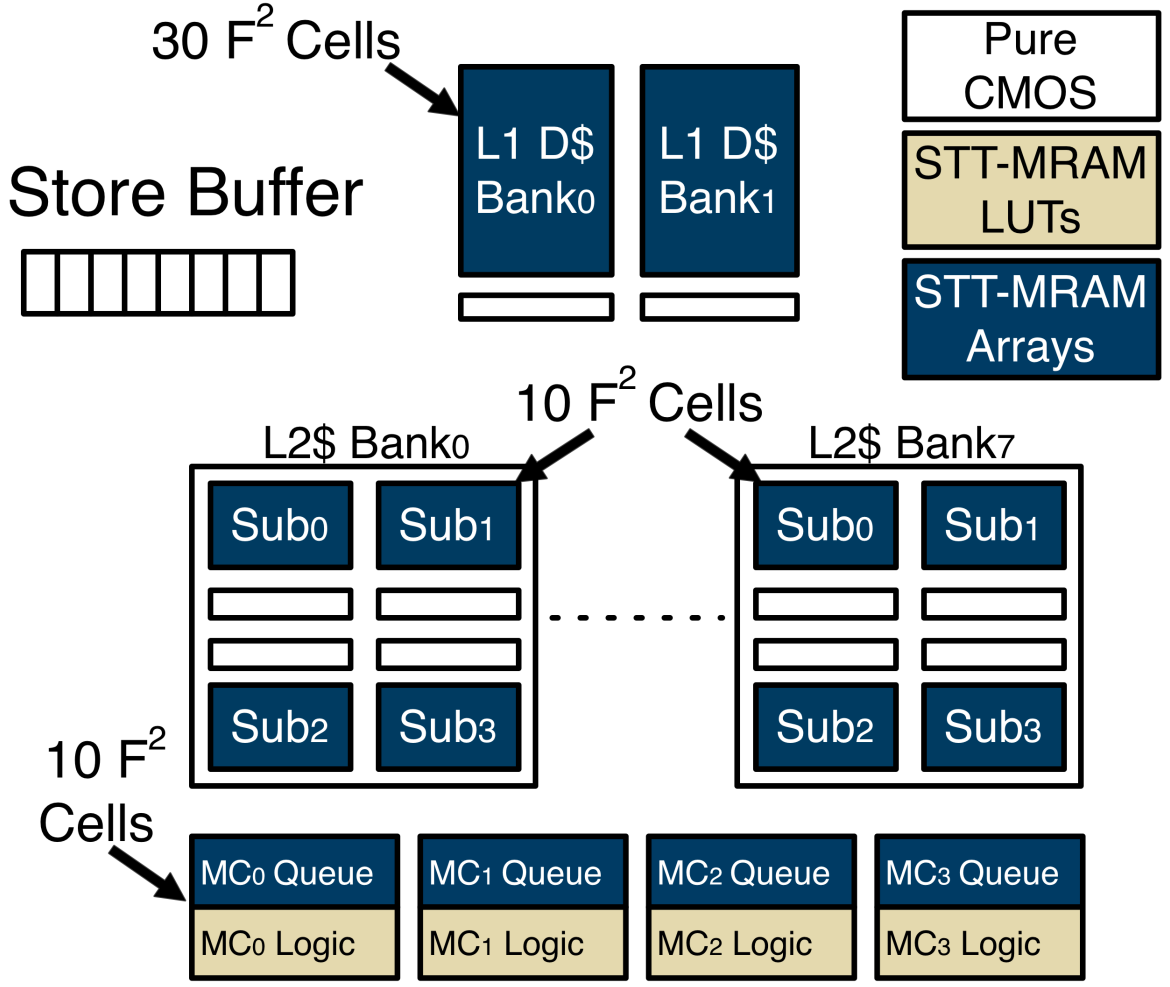


Memory System

Use store buffers to avoid L1 D$ subbank conflicts

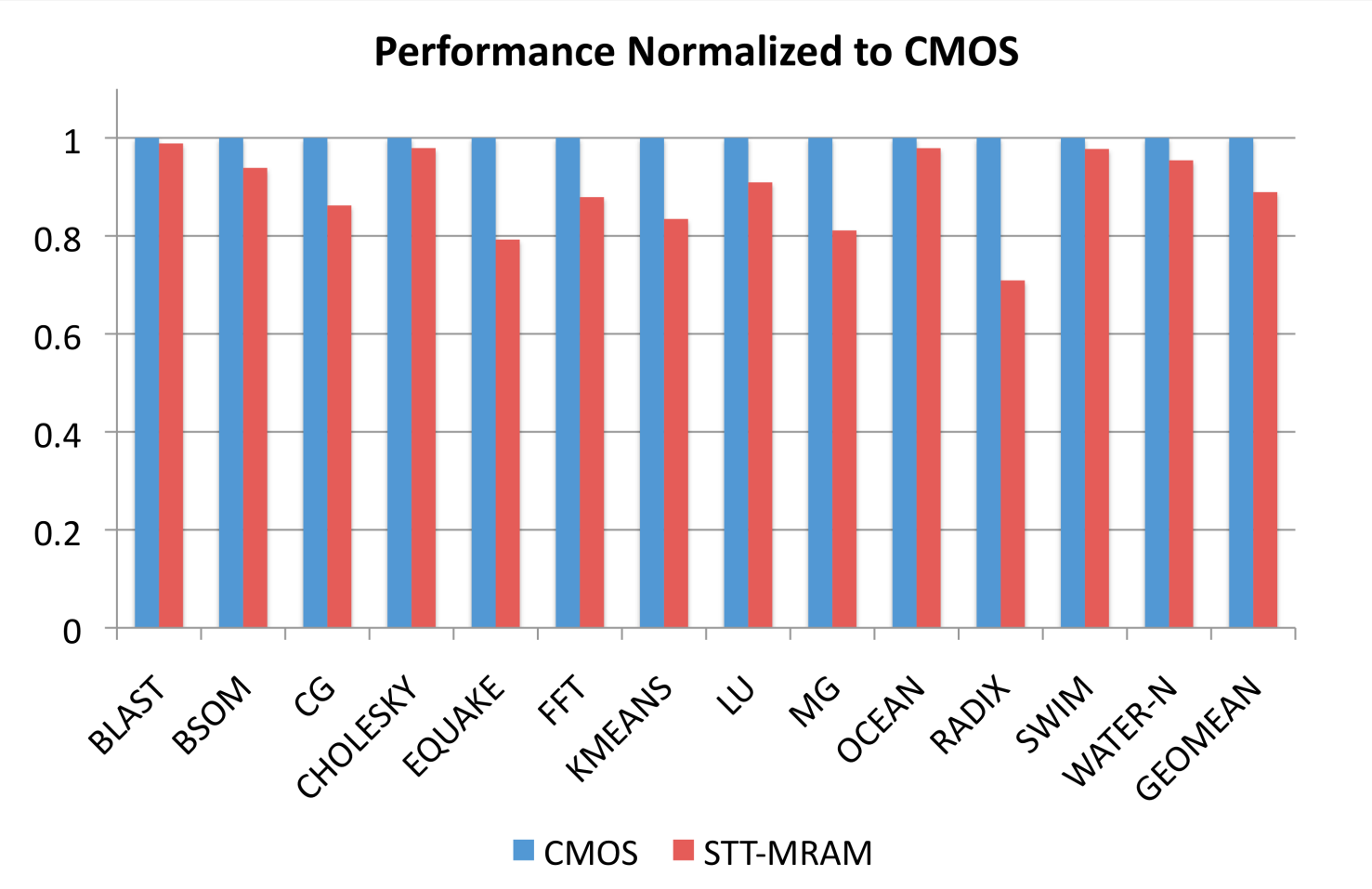
L1s optimized for fast writes using $30F^2$ cells

L2 and memory controllers optimized for density using $10F^2$ cells

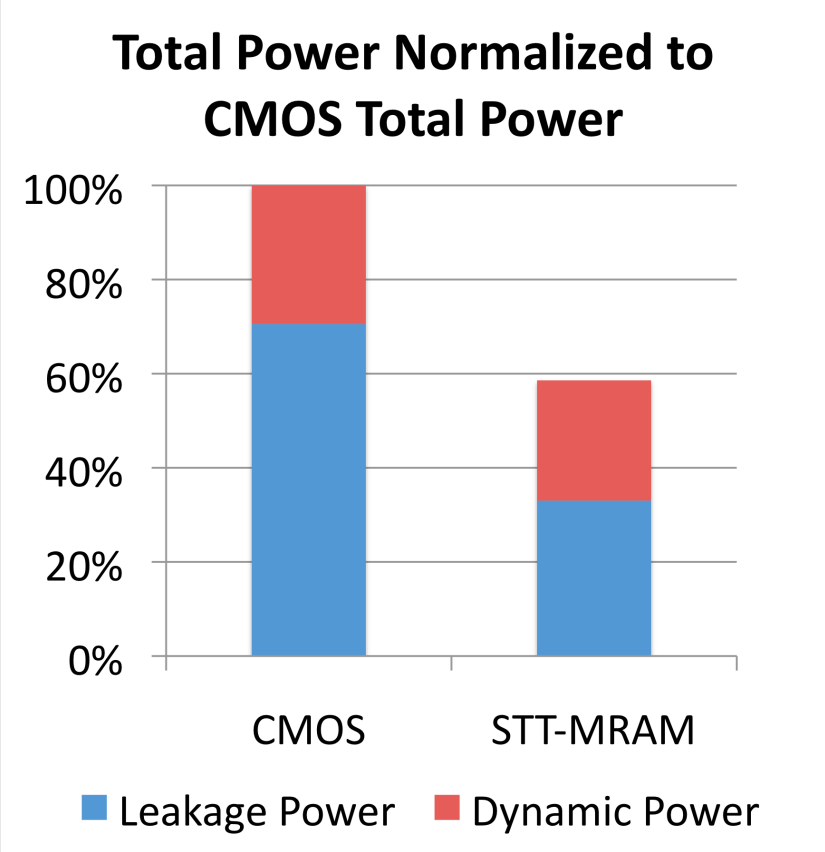


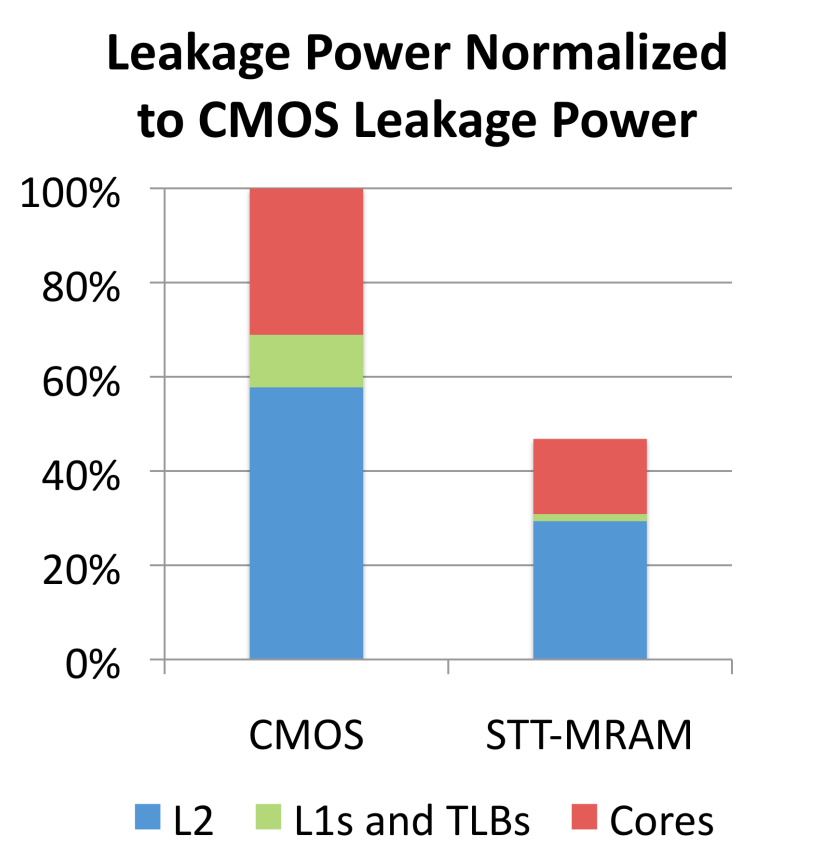
Evaluation

Performance



Power


Total Power Normalized to CMOS Total Power
100%
80%
60%
40%
20%
0%
CMOS
STT-MRAM
Leakage Power
Dynamic Power


Leakage Power Normalized to CMOS Leakage Power
100%
80%
60%
40%
20%
0%
CMOS
STT-MRAM
L2
L1s and TLBs
Cores

Contributions and Findings

- New technique to reduce leakage and dynamic power in a deep-submicron microprocessor
  - Selectively migrate on-chip storage and combinational logic from CMOS to STT-MRAM
  - Use subbank buffers to alleviate long write latency

- STT-MRAM is an attractive low-power solution beyond 32nm
  - Dramatically lower leakage power
  - Modest loss in performance