Forget Failure: Exploiting SRAM Data Remanence for Low-overhead Intermittent Computation

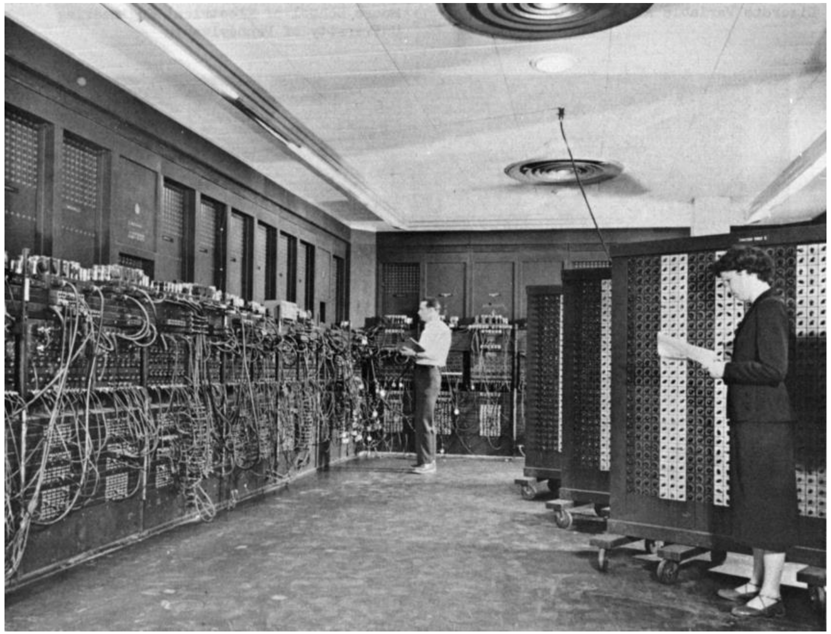
Harrison Williams
hrwill@vt.edu

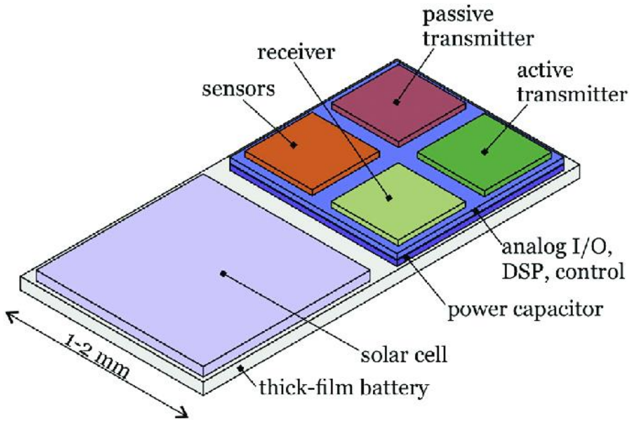
Xun (Steve) Jian
xunj@vt.edu

Matthew Hicks
mdhicks2@vt.edu

COMPUTER SCIENCE
VIRGINIA TECH™

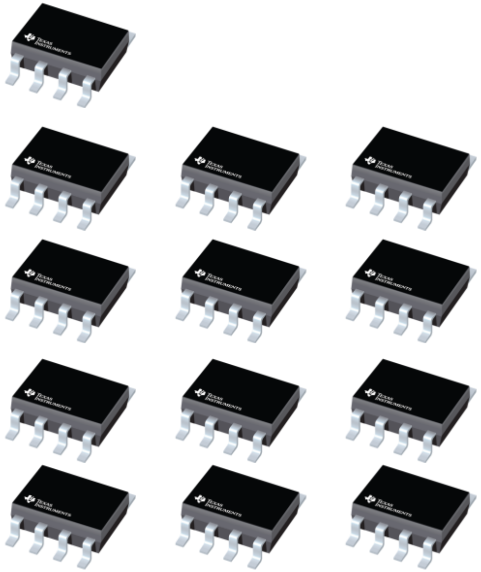
Processors are small

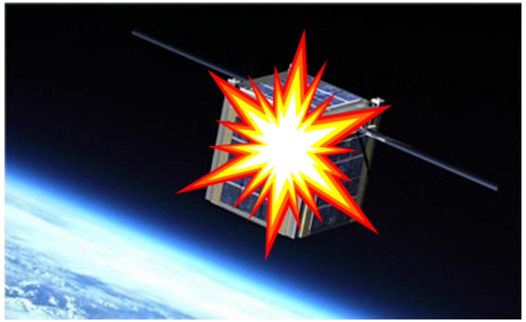


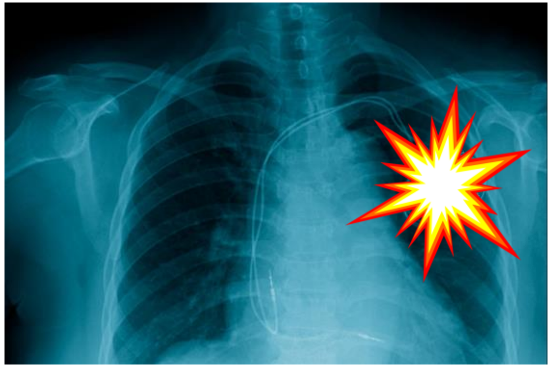


Batteries are burdensome

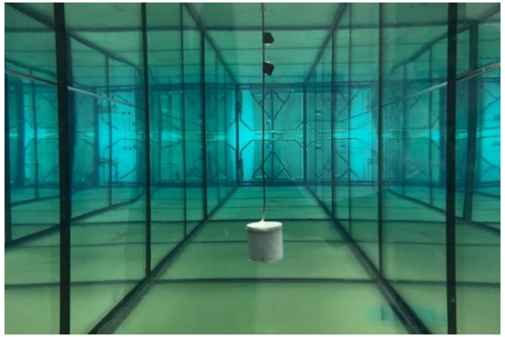
=





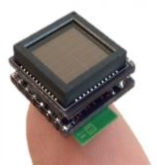


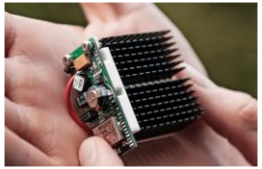
Harvesting energy from the environment



$\longrightarrow$

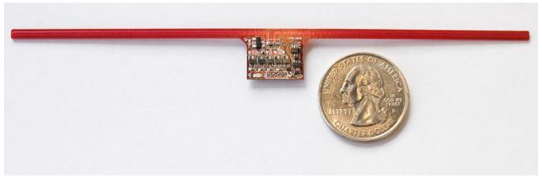
🙂



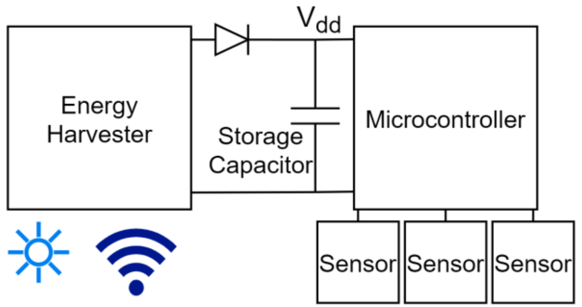


$\longrightarrow$

🙁

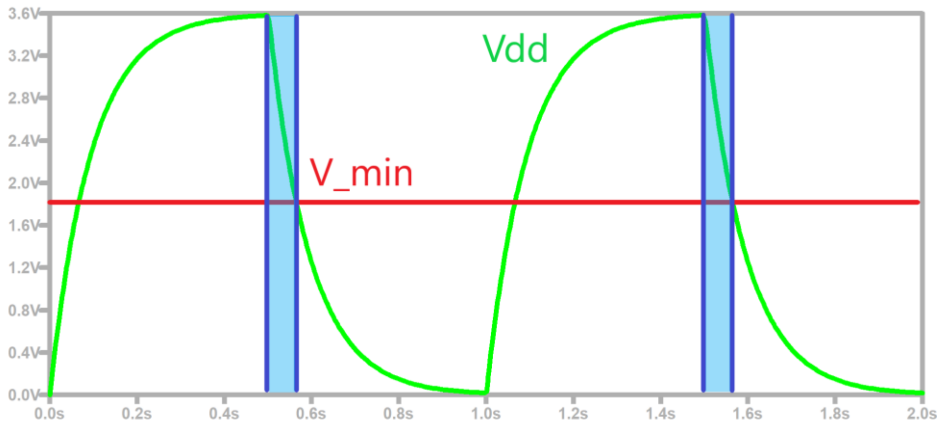


Intermittent power means intermittent computation

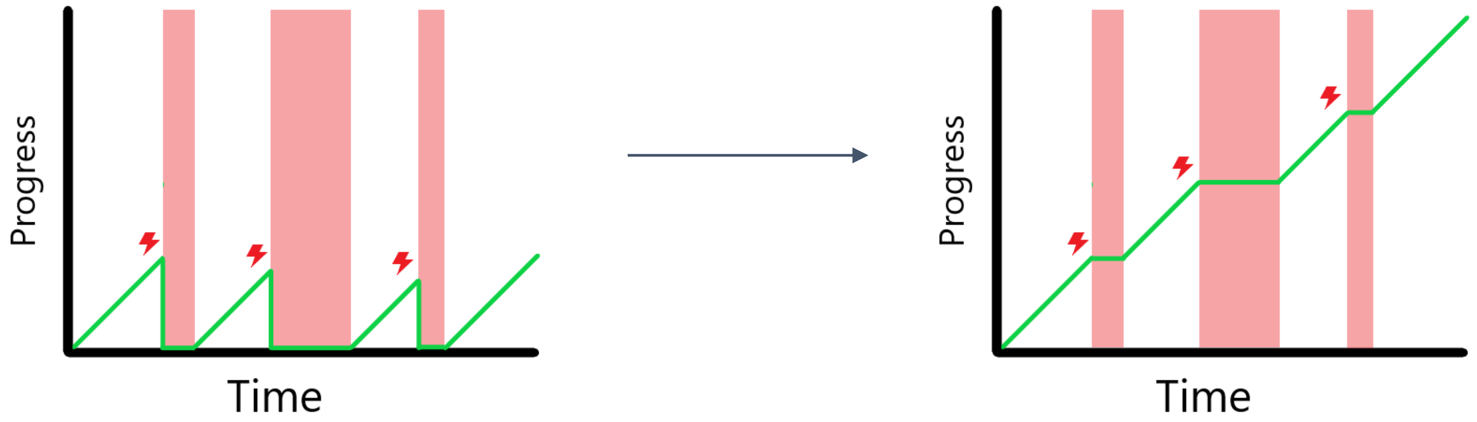


Common-case failure

- Time limit
- Energy limit
- Correctness issues



What can we do?



Challenge: Preserve volatile state

Solution: Checkpoint to Non-Volatile Memory (NVM) before failure [Ransford '11]

Memory choice determines performance

Flash

- ➕ Most common NVM
- ✖ Fast reads, ***slow*** writes
- ✖ **Low endurance** (lifespan: 1 day)

Emerging memories (FRAM)

- ➕ High checkpoint performance
- ✖ Lower execution performance
- ✖ **Limited options**

Memory choice determines performance

Static RAM (SRAM)

➕ Highest performance

➕ Ubiquitous

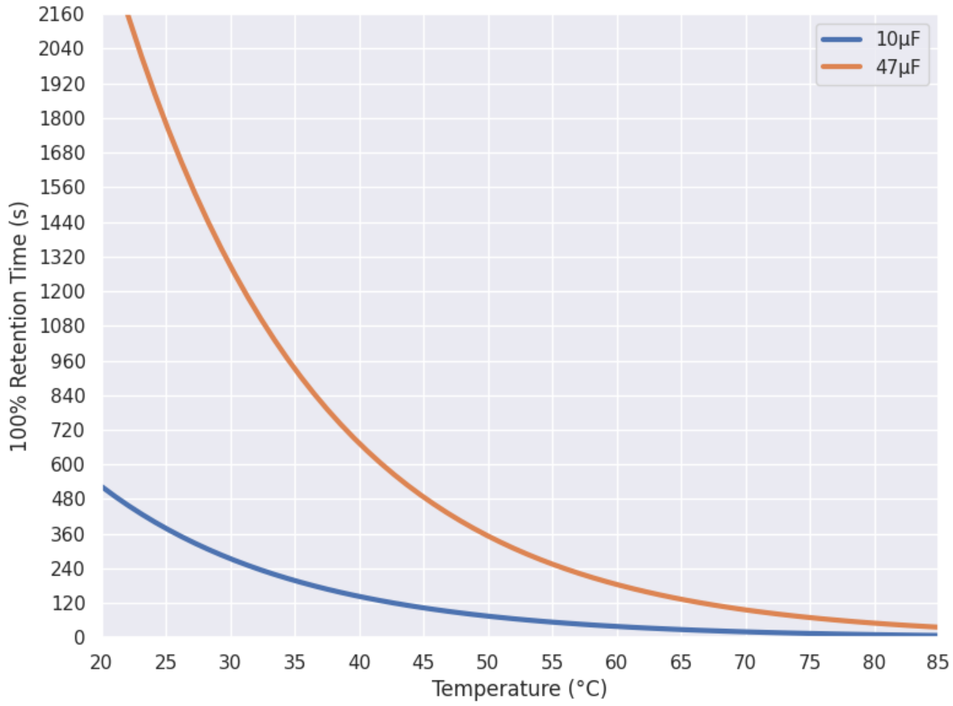
➕ No transfer overhead

✖ Volatile

SRAM has time-dependent volatility

Full data retention for > 5 minutes

Past work: ~2 second retention with no added capacitor! [Rahmati '12]



COMPUTER SCIENCE
VIRGINIA TECH.

Intermittent off times are short

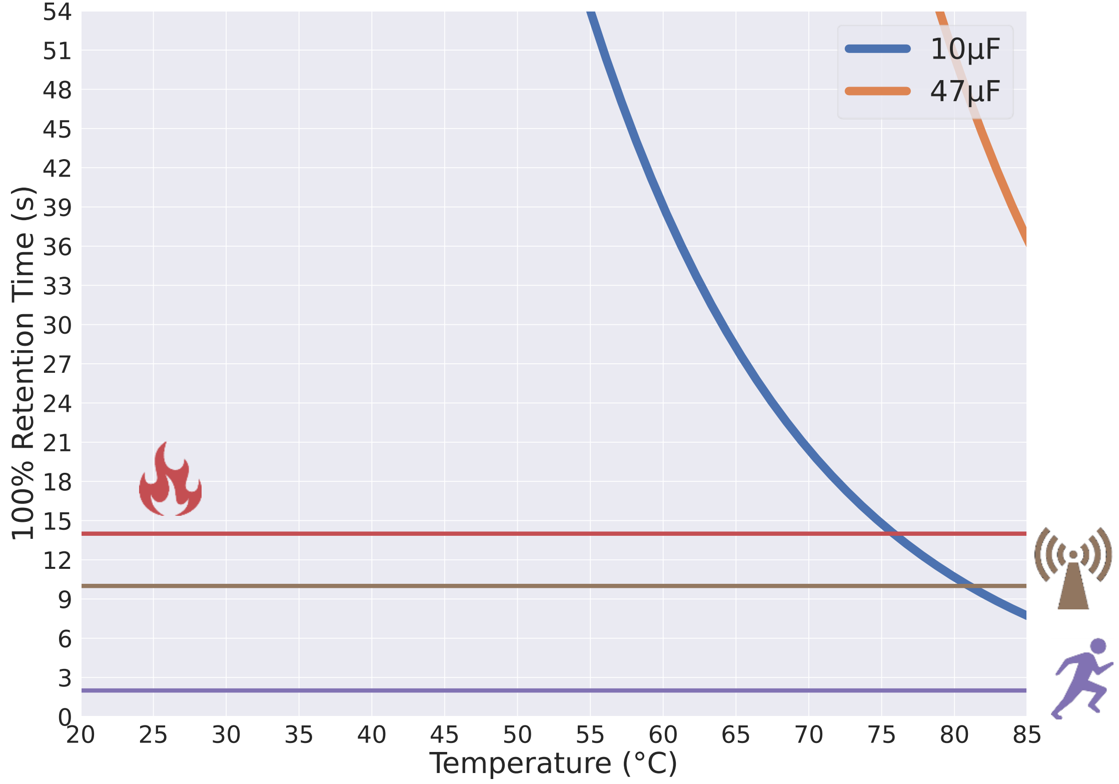
Off times ≈ on times (short)

Retention at 20° C:

- Flash: 100 years
- SRAM: 6 minutes

SRAM retention fits!

Why use NVM at all?



COMPUTER SCIENCE
VIRGINIA TECH.

Some off times are long

Long off times = predictable **Long off times = irrelevant**

TotalRecall

Avoid all NVM-writes by taking checkpoints that reside entirely in SRAM.

Goals:

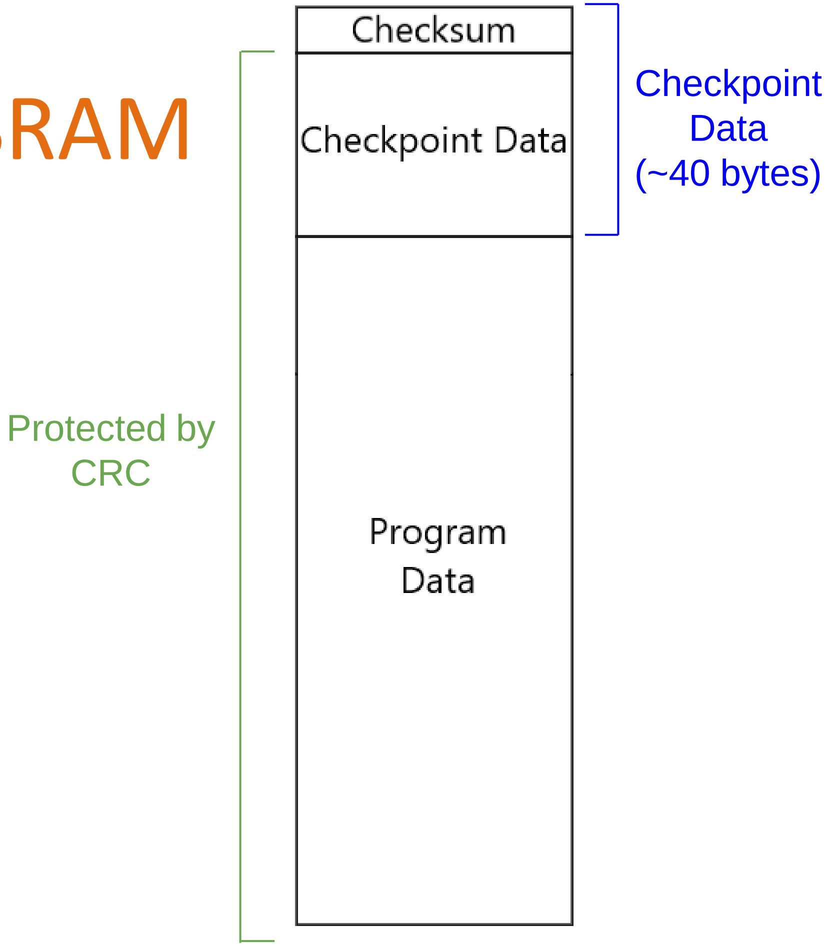
1. Efficiency - take advantage of common case short off times

2. Correctness - handle uncommon case long off times

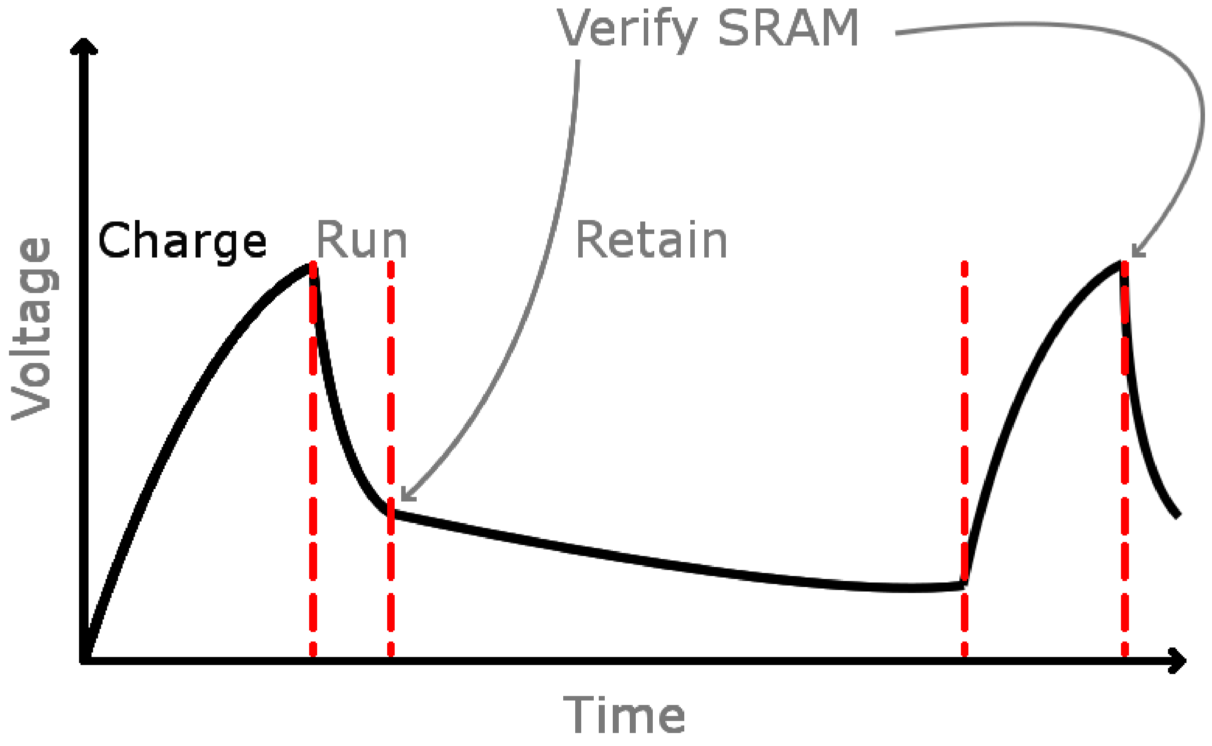
TotalRecall checkpoints in SRAM

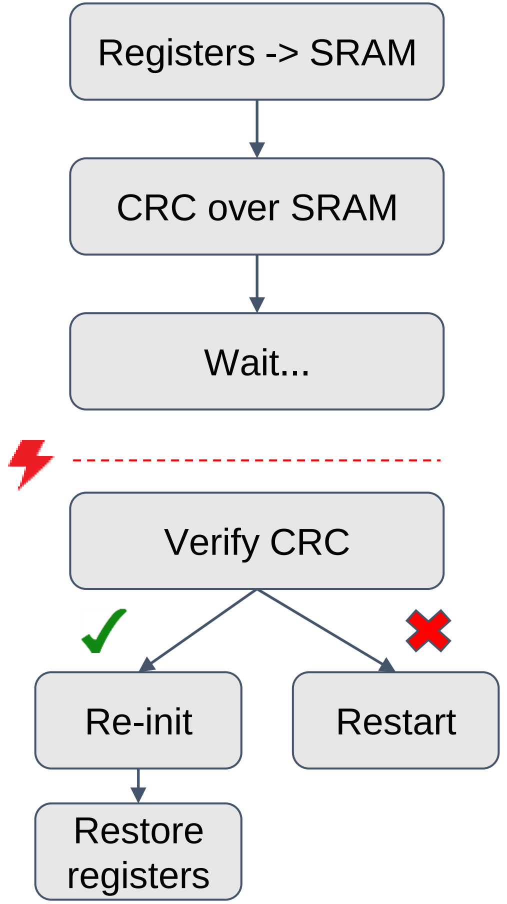
Cyclic Redundancy Check (CRC)

- ✚ Simple
- ✚ Fast
- ✚ Hardware support



Using remanence safely





COMPUTER SCIENCE
VIRGINIA TECH.

How did we evaluate it?

Platforms

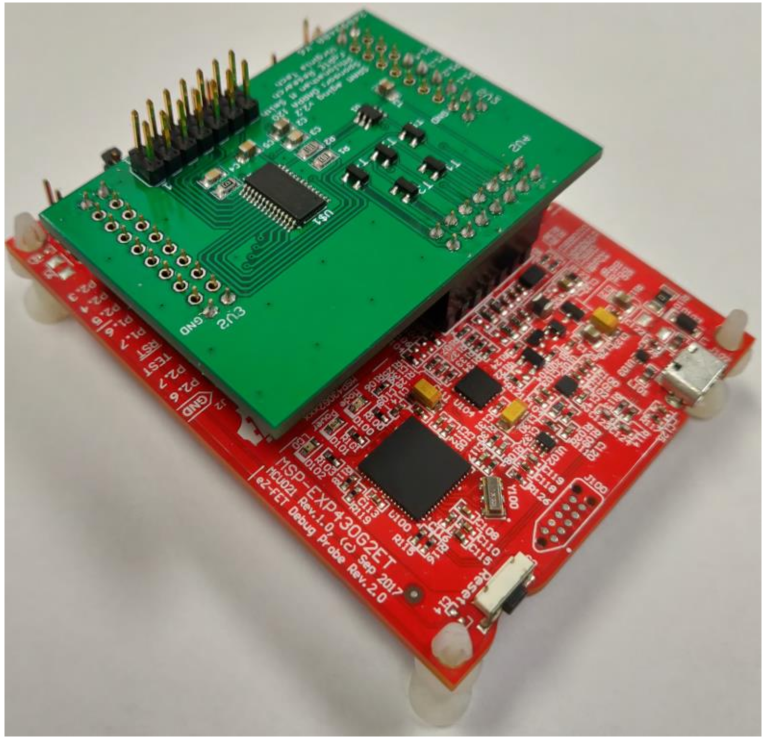
- MSP430G2553 (Flash)
- MSP430FR6989 (FRAM)
  - CRC engine

Benchmarks

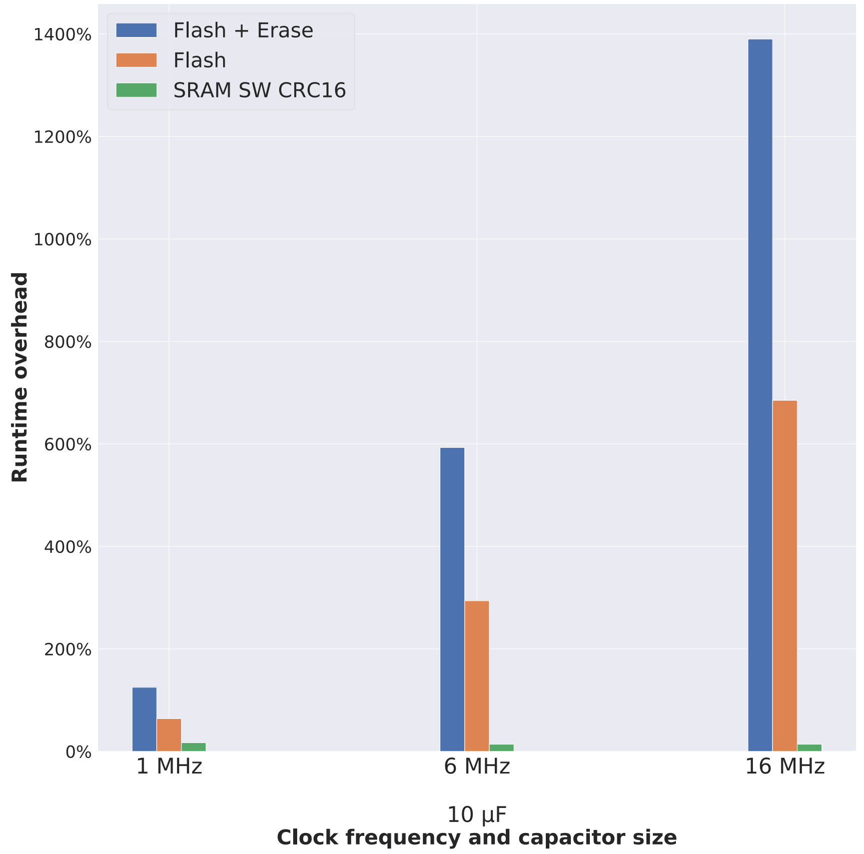
- DSP, math, sorting
- Benchmark-agnostic

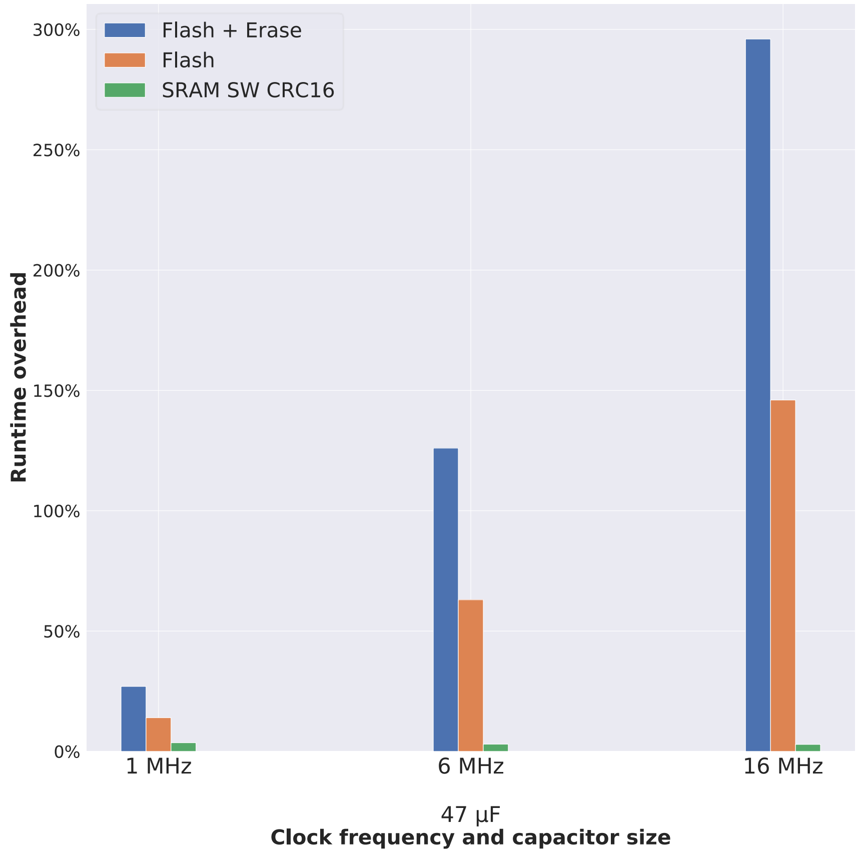
Baseline

- One-time checkpoints to NVM



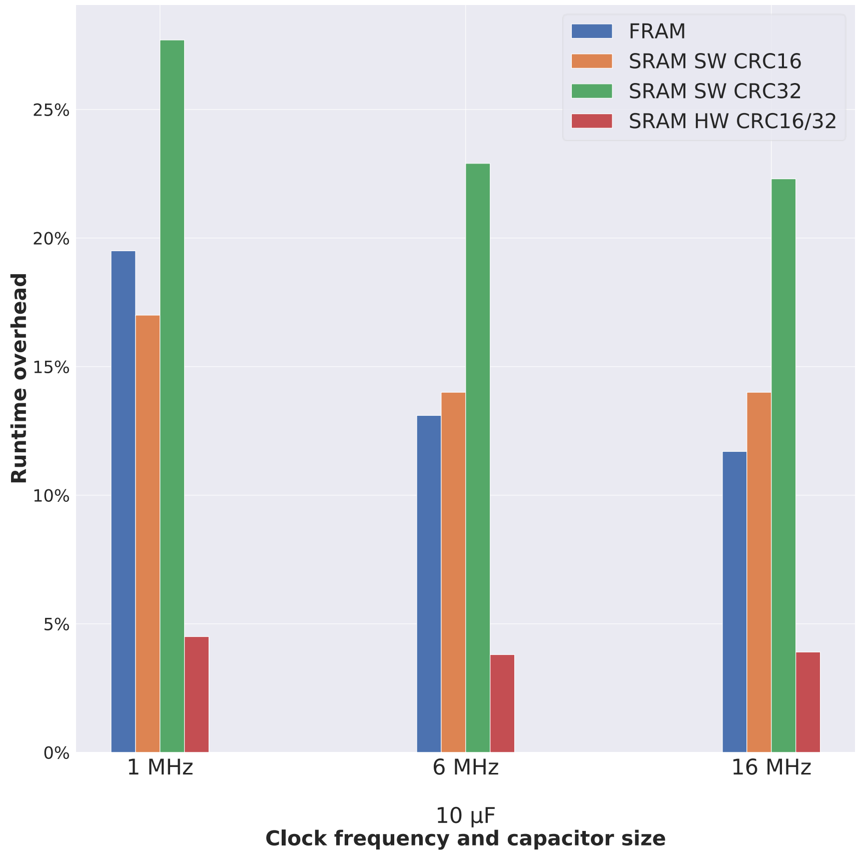
Flash: high overhead in all cases

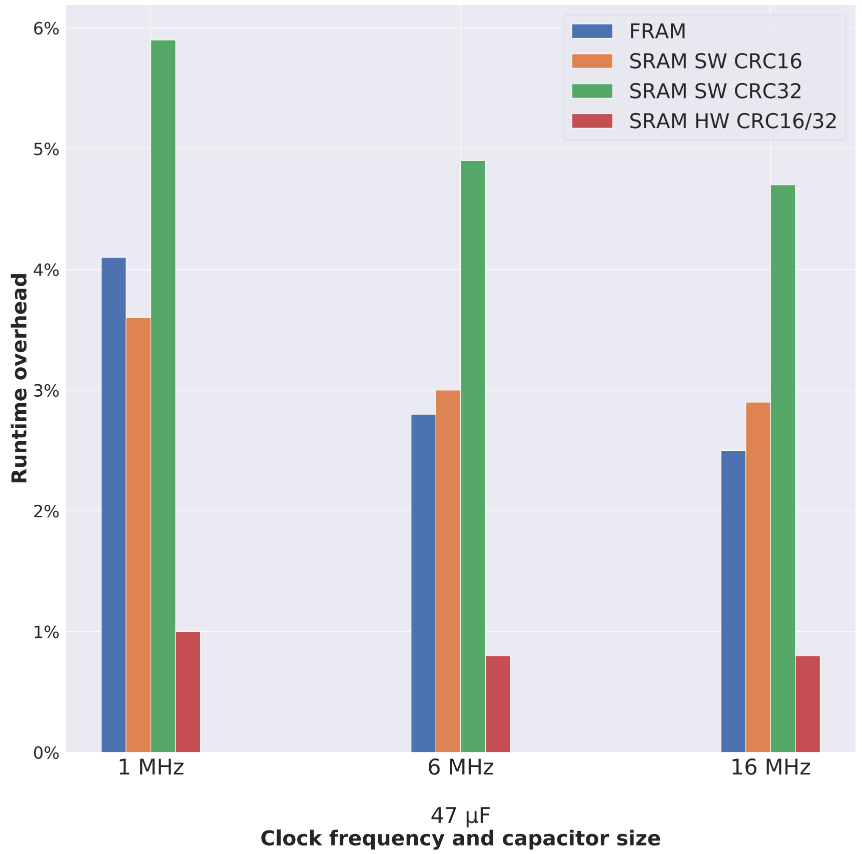




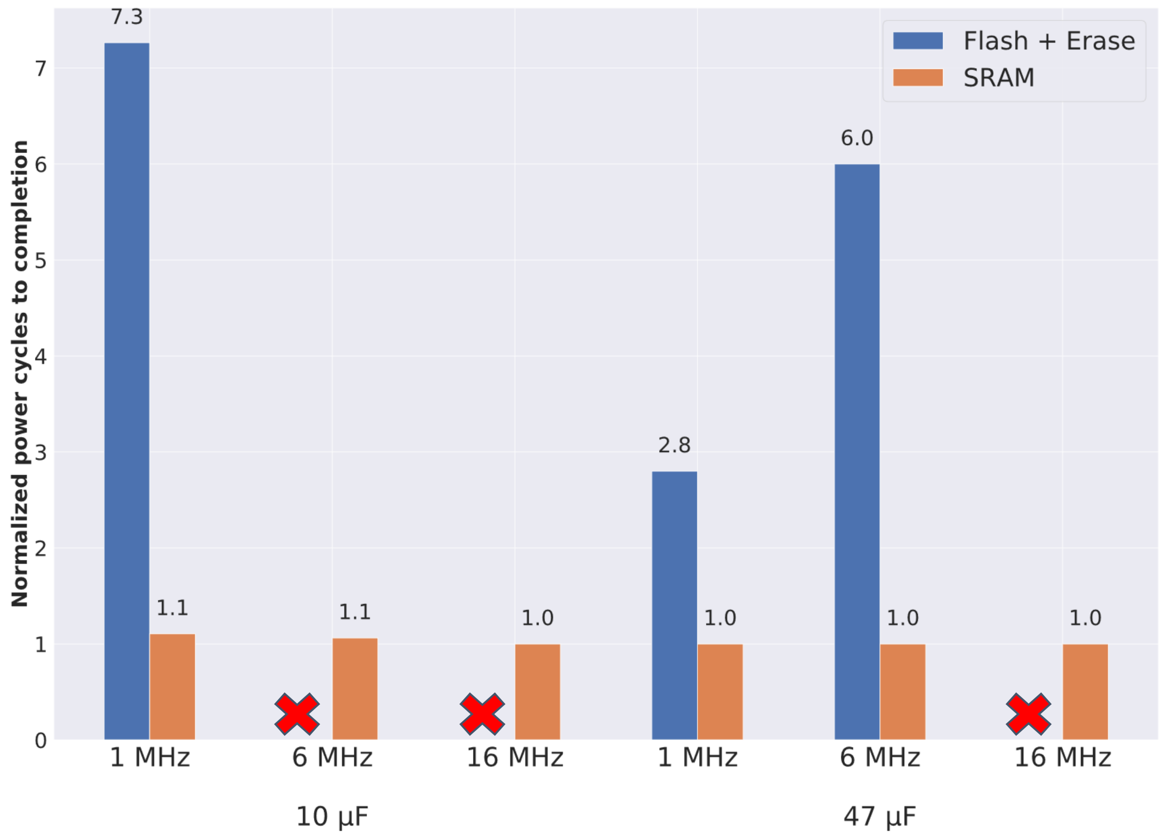
COMPUTER SCIENCE
VIRGINIA TECH.

TotalRecall: FRAM-level performance anywhere





Benchmarks complete in ideal time



TotalRecall Summary

- ✓ Preserves program data using SRAM remanence - not costly NVM
- ✓ Protects against data loss with a simple, quick integrity check
- ✓ Outperforms state-of-the-art NVM with a software update
- ✓ Result: efficient intermittent computation on ***any*** platform

https://github.com/FoRTE-Research/TotalRecall-artifact