



# Fault Propagation in Microprocessors with Configurable Cache Memory

**S. Esquer,<sup>1</sup> B. Shani<sup>1</sup>, A.F. Witulski<sup>1</sup>, B.D. Sierawski<sup>1</sup>, B.L. Bhuva<sup>1</sup>, R.A. Reed<sup>1</sup>, R.D. Schrimpf<sup>1</sup>, G. Karsai<sup>1</sup>, M. Turowski<sup>2</sup>**

**<sup>1</sup> Dept. of Electrical and Computer Engineering, Vanderbilt University**

**<sup>2</sup> Alphacore, Inc**

**Supported under subcontract to Alphacore, Inc., on NASA Grant 80NSSC21C0033**

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

# Acronyms

---



*Vanderbilt Engineering*

- COTS: Commercial off the shelf
- CPU: Central Processing Unit
- DRAM: Dynamic random-access memory
- HPC: High performance computing
- LANL: Los Alamos National Laboratory
- MWG: Mitigation working group at LANL
- SEE: Single event effect
- SEFI: Single event functional interrupt
- SEL: Single event latchup
- SEU: Single event upset
- SRAM: Static random-access memory

# Presentation Structure

---



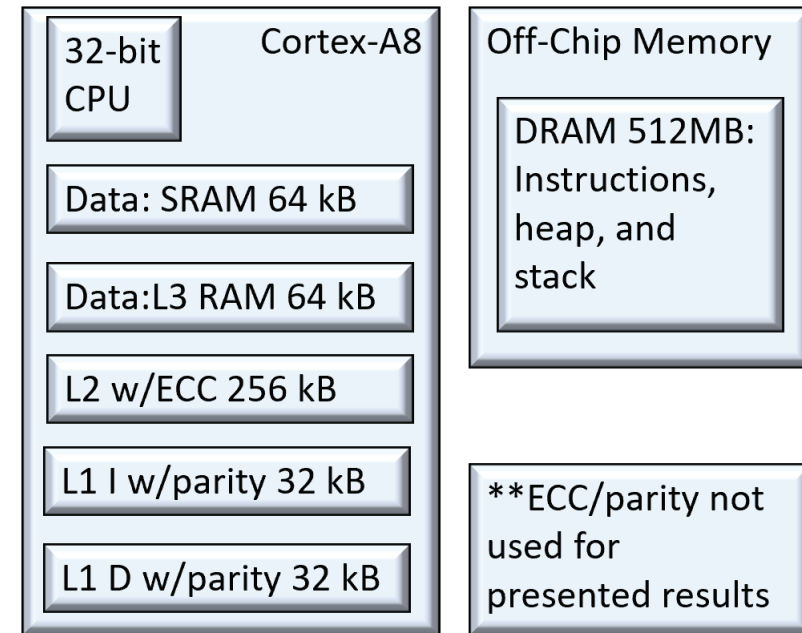
*Vanderbilt Engineering*

- **Introduction:** Purpose of this work and main findings
- **Test set-up:** Software, hardware, and alpha particle source
- **Results:** SEU propagation, and SEFIs
- **Current work:** Proton data analysis
- **Future work:** Multicore microprocessor
- **Conclusions**

# Introduction



- Overall project purpose:
  - Characterize SEEs of COTS HPC for operation in highly-shielded (habitable) space environments.
  - COTS computing provide inexpensive augmentation to spacecraft computational power<sup>1,2,3</sup>
  - What are the key critical factors that affect the SEFI and error rate? Due to ionizing radiation<sup>2</sup>
- Target device: Cortex-A8 microprocessor on the BeagleBone Black board



Cortex-A8 memory structure

# Introduction

---



Vanderbilt Engineering

- Main Findings:
  - The SEFI cross-section ( $\sigma$ ) is affected by the cache memory configuration
  - When cache-on, more halt SEFIs are experienced
  - No SEUs could be detected when isolating the cache by storing instructions, data, and stack off-chip (**key factor to analyze SEU data**)
- Purpose of this work:
  - SEFI characterization of target under alpha particles
  - Dynamic testing of target with MGW radiation benchmarks<sup>2,3,4</sup>
  - Find reasons some algorithms have higher SEFI  $\sigma$ s than others
  - Begin test methodology to classify different types of SEFIs

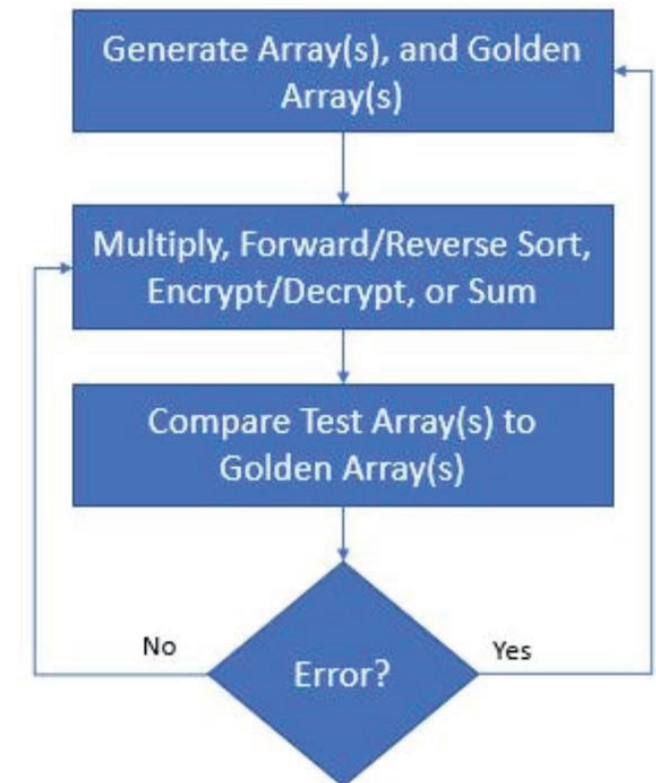
# Test Set-Up: Algorithms



Vanderbilt Engineering

- **Algorithms:** Mitigation Working Group at Los Alamos National Laboratory LANL radiation benchmarks<sup>3</sup>
- **Selected benchmarks:** Matrix multiply (MM), and sort algorithm (Q-Sort)
- **Execution:** One completed loop of the flow chart is one benchmark cycle
  - Benchmarks were run on bare-metal (no operating system used)
- **Memory during testing:**
  - Instructions and stack in the off-chip DRAM
  - Data in the on-chip SRAM
  - ECC and parity **always** off

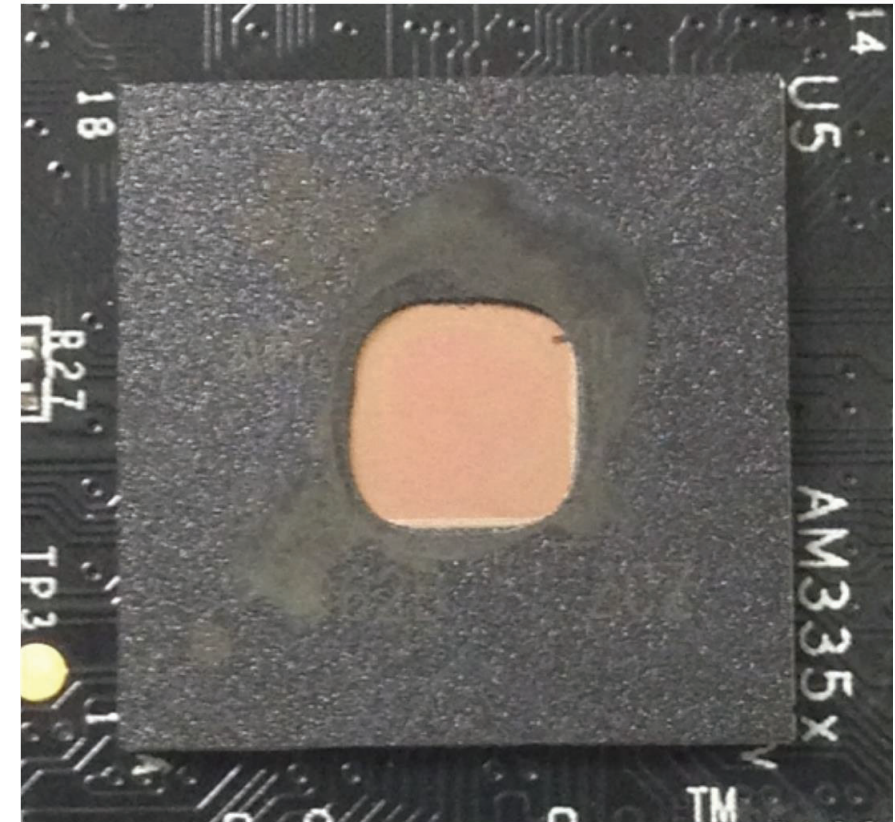
## Benchmarks





# Test Set-Up: Source & Target

- 10 and 0.1  $\mu$ -Ci Am-241 with 4-MeV alpha particles
- Flux of 10  $\mu$ -Ci =  $1,000 \frac{\text{particles}}{\text{mm}^2 * \text{s}}$
- Flux of 0.1  $\mu$ -Ci =  $10 \frac{\text{particles}}{\text{mm}^2 * \text{s}}$
- The Cortex-A8 had a  $3.95 \times 3.95 \text{ mm}^2$  decapsulated opening
- LET is  $0.7 \left( \text{MeV} - \frac{\text{cm}^2}{\text{mg}} \right)^5$





# Test Set-Up: SEFI Capture



A SEFI occurs when an ion strike triggers the control circuitry of one of the subsystems. The system could enter an undefined state until a reset or sometimes a power cycle is performed.<sup>6,7</sup>

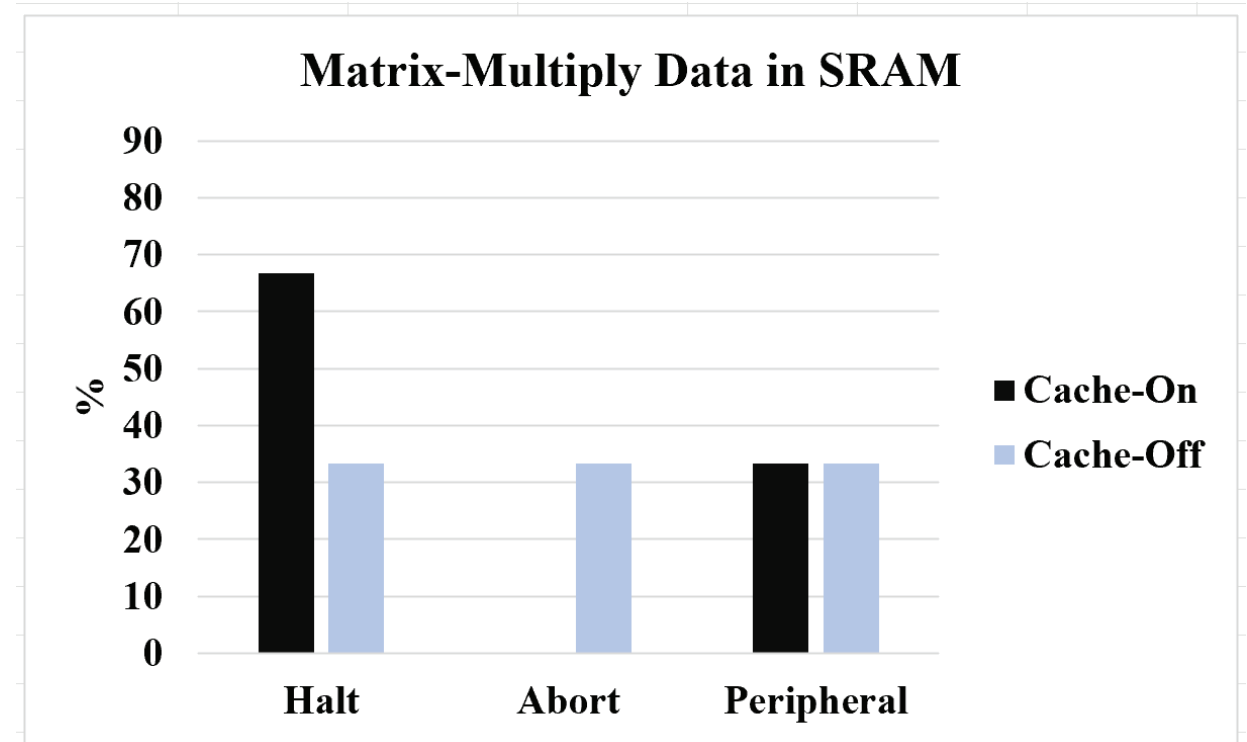
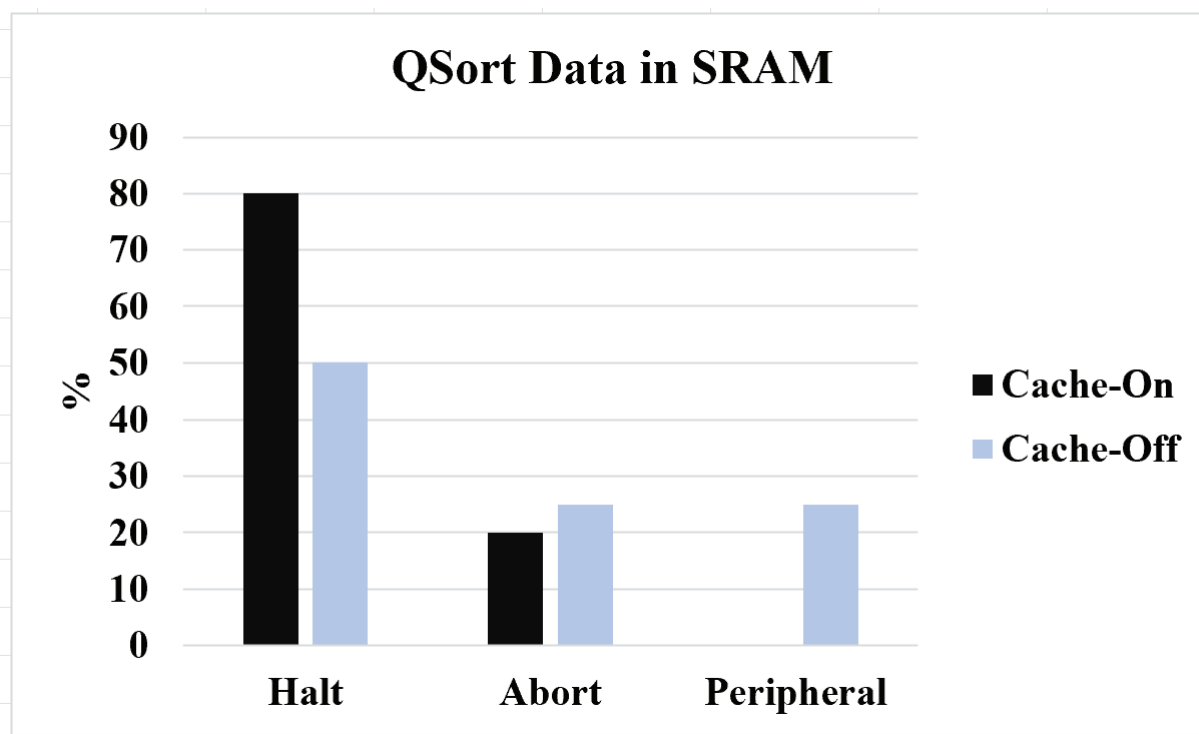
SEFI	Mechanism
Reset	The microprocessor resets on its own. <sup>7</sup> <b>For the Cortex-A8 instructions/data is erased (volatile memory) and it is hard to differentiate from crash SEFIs</b>
Crash	The processor gets into an undefined state. Fetch and execute cycles are halted <sup>7</sup> Trigger: Direct/indirect ionization in special purpose registers Trigger: Access to invalid instruction or data memory-> entrance to <b>“abort exception handler”</b> <sup>8</sup> Trigger: Access to invalid instruction or data memory-> entrance to <b>“undefined exception handler”</b> <sup>8</sup>
Peripheral	When the system does not operate the peripherals as desired <sup>7</sup> Trigger: Direct ionization in the peripheral’s control registers



# Test Set-Up: SEFI Capture



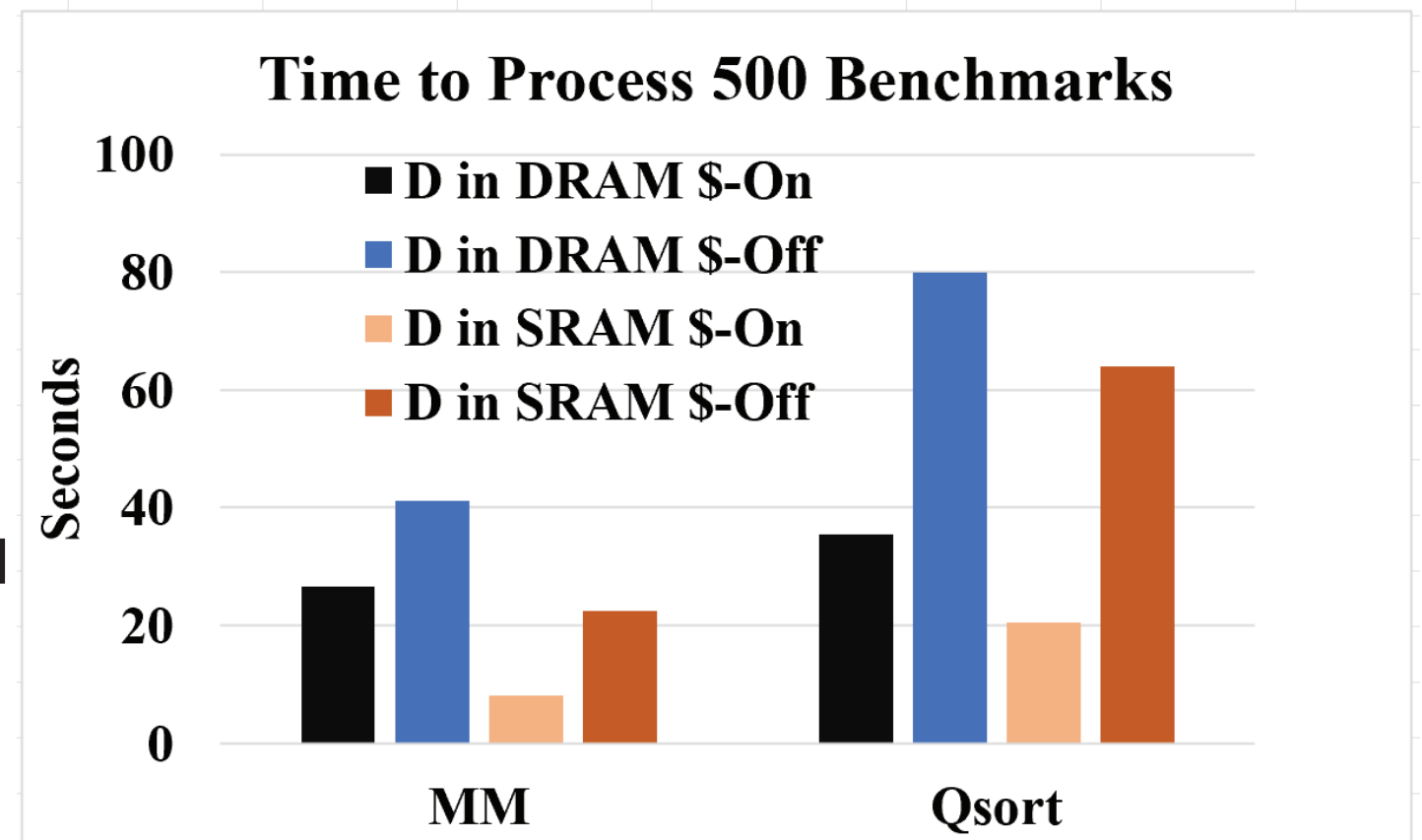
- **Cache-On:** Algorithms experienced higher occurrences of halt SEFIs
- SEUs in the cache control circuitry could be obstructing fetch of instructions
- **Halt SEFI:** Is either a reset/or crash SEFI (due to volatile memory)
- SEFIs events: MM cache on/off (3)/(3), Q-Sort cache on/off (5)/(4)





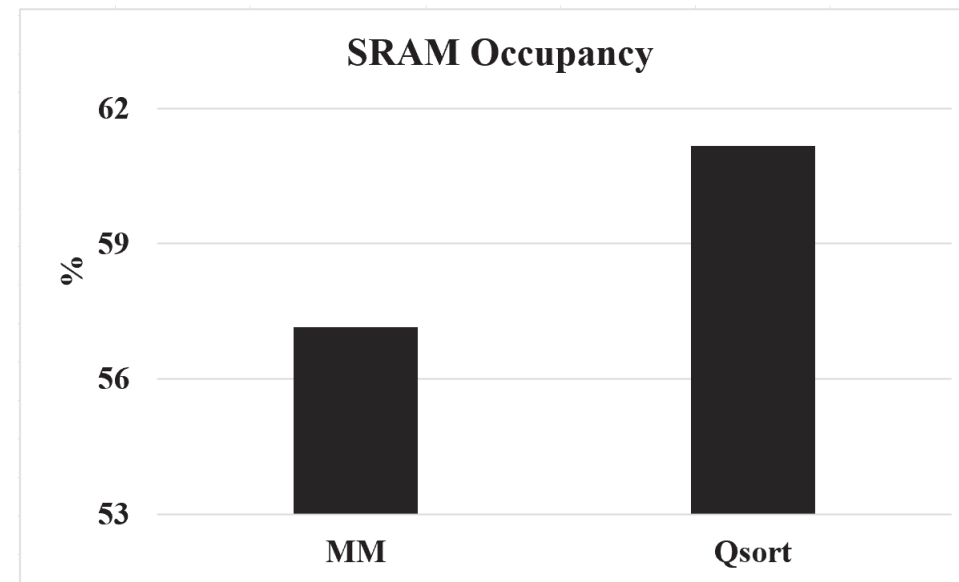
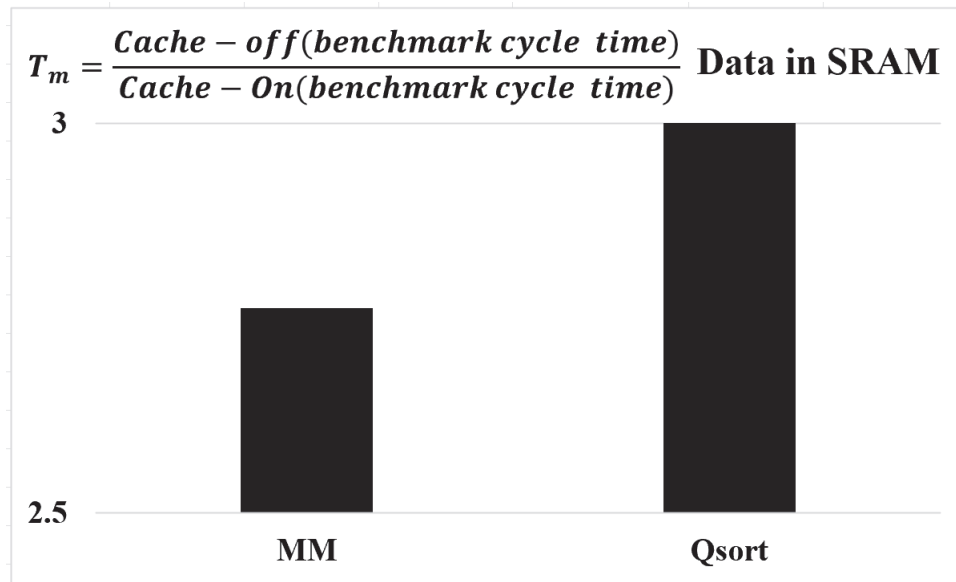
# Test Set-Up: Memory

- Cache (\$) SEU  $\sigma$  test:
  - No SEUs, after isolating \$ by storing data in off-chip DRAM
  - Single entrance to undefined instruction handler indicates *possible SEU* in L1 inst. cache
  - MM runs 2.5x faster cache-on, and 2.8x faster cache-off when data in SRAM compared to Q-Sort



D=Data. Different settings demonstrate proper use of cache, and cache isolation

# Test Set-Up: Memory



- Time ratio ( $T_m$ ):
  - MM is 2.7x faster cache-on vs. cache-off. Q-Sort is 3.13x faster cache-on vs. cache-off (higher performance) with data in SRAM
  - Cache hit/miss ratios could provide reason for Q-Sort's higher performance gain

# Results: Benchmark Cycle Errors with Alpha Source



Vanderbilt Engineering

## $(n \times n)$ Matrix Example

Case 1:  
SEU in 2<sup>nd</sup> matrix

$$\begin{bmatrix} n_{11} & \dots & n_{1n} \\ \vdots & \ddots & \vdots \\ n_{n1} & \dots & n_{nn} \end{bmatrix} \times \begin{bmatrix} n_{11} & \dots & n_{1n} \\ \vdots & \ddots & \vdots \\ n_{n1} & \dots & n_{nn} \end{bmatrix} = \begin{bmatrix} n_{11} & \dots & x \\ \vdots & \ddots & \vdots \\ n_{n1} & \dots & x \end{bmatrix}$$

Results Matrix

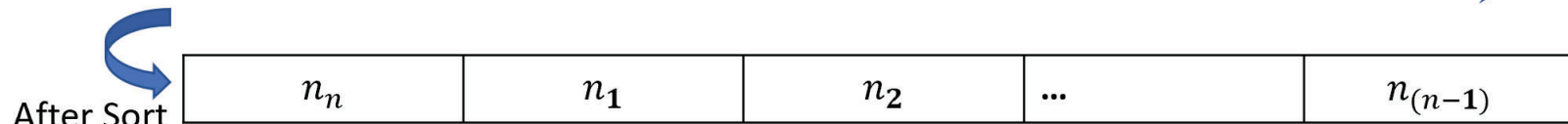
Case 2:  
SEU in 1<sup>st</sup> matrix

$$\begin{bmatrix} n_{11} & \dots & n_{1n} \\ \vdots & \ddots & \vdots \\ n_{n1} & \dots & n_{nn} \end{bmatrix} \times \begin{bmatrix} n_{11} & \dots & n_{1n} \\ \vdots & \ddots & \vdots \\ n_{n1} & \dots & n_{nn} \end{bmatrix} = \begin{bmatrix} x & \dots & x \\ \vdots & \ddots & \vdots \\ n_{n1} & \dots & n_{nn} \end{bmatrix}$$

In general:

Assume  $n_n$  is flipped to any integer smaller than any other member within the array

## Forward Sort Example (Q-Sort). Unidimensional Array



Entire array is shifted one position to the right

In general:  
1 SEU leads to errors  $\leq$  array size per benchmark cycle

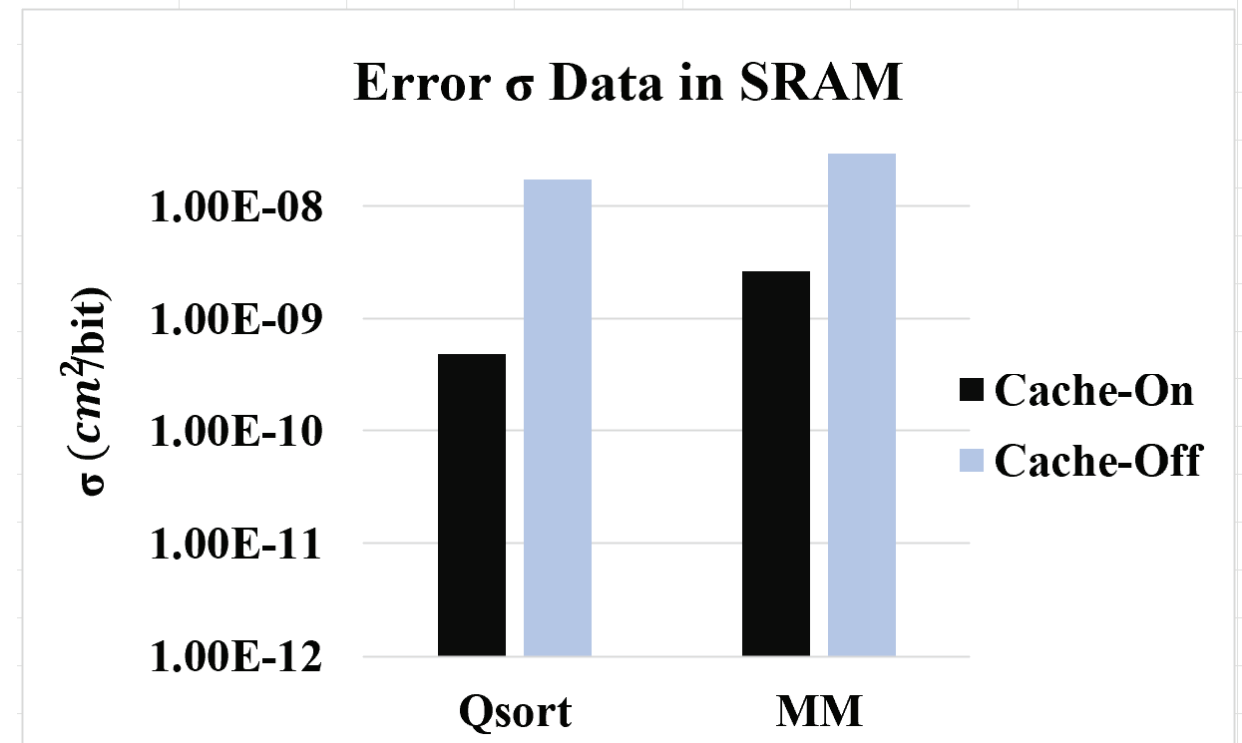
- 1 SEU in MM might be propagating faster into errors ( 2.5x faster \$-on and 2.8x faster \$-off vs. Q-Sort)
- SEUs  $\neq$  errors
- More errors could accumulate with higher fluxes

# Results: Campaign Benchmark Error Cross-Section ( $\sigma$ )



Vanderbilt Engineering

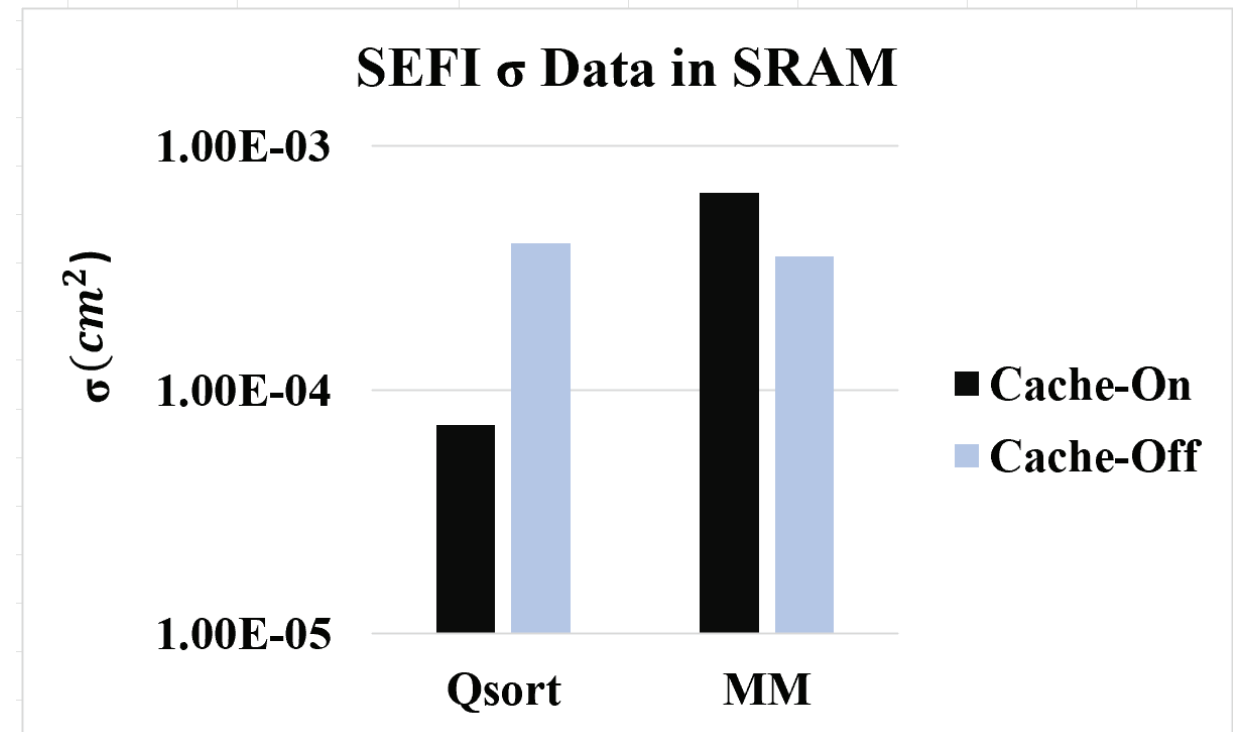
- Low  $\sigma$  for cache-on:
- Preliminary cache isolation demonstrated SEUs occur mainly in main memory
- Algorithmic dependency in Error  $\sigma$ :
- MM always had higher error  $\sigma$  due to propagation of SEUs



# Results: SEFIs



- MM higher SEFI  $\sigma$ , cache-on :
  - Cache hit/miss ratios could provide reason
- Higher  $\sigma$  for Q-Sort, cache-off:
  - Factors other than build-up of stack frames in the stack (stack in off-chip memory) makes recursive algorithms vulnerable to SEFIs

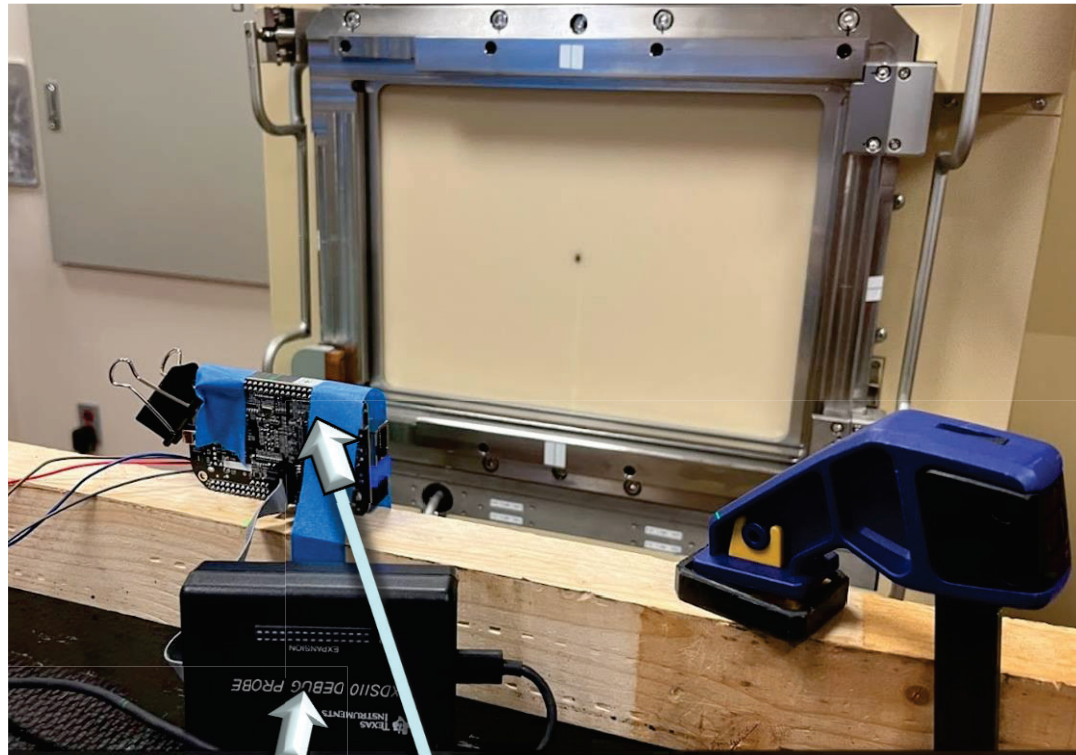


- For these test conditions the SEFI  $\sigma$  is affected by the cache configurations

# Current Analysis: Proton Test Campaigns at the Mayo Clinic



Vanderbilt Engineering



JTAG

Beagle Bone Black  
with Cortex-A8

Proton beam campaign hardware set-up

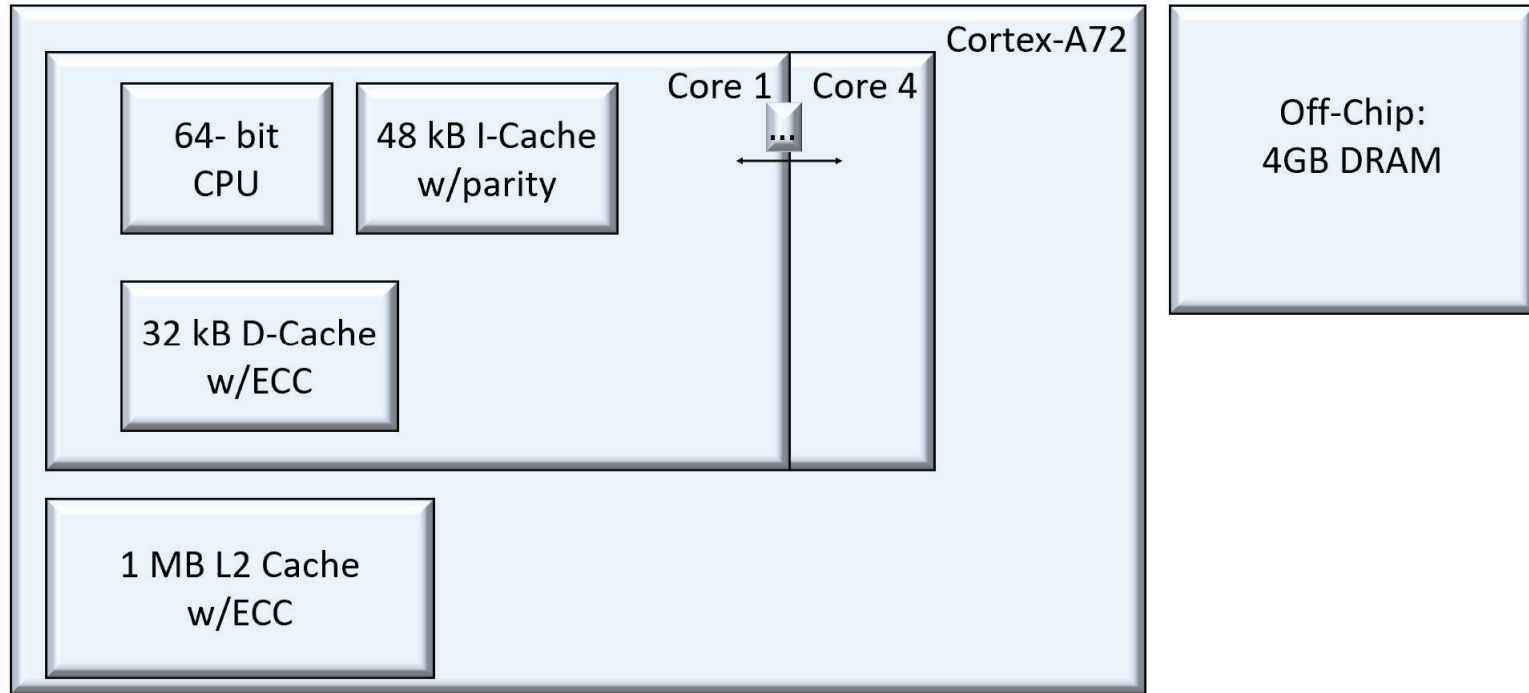
- **Used benchmarks:**
  - Advanced encryption standard AES
  - Cache test (summation algorithm)
- **Purpose:**
  - SEFI dependence to cache configuration
  - Compare stack influence between higher control flow dependence (AES) and a lower control flow dependence (Summation)
- Currently doing data analysis



# New Target: Cortex-A72 on the Raspberry Pi 4 Model B



Vanderbilt Engineering



Cortex-A72 memory structure

- Interest:
  - Mimic multiple nodes of a high-performance computer
  - Run bare-metal source code
  - Compare benchmarks SEFI results with single core Cortex-A8

# Conclusions and Future Work



*Vanderbilt Engineering*

- **Conclusions:**
  - SEFI  $\sigma$  is affected by the cache memory configuration
  - Time ratios  $T_m$  demonstrated that Q-Sort has a higher performance gain when cache-on
  - Cache hit/miss ratios could provide explanation for Q-Sort's lower SEFI  $\sigma$  when cache-on
  - When cache-on more halt SEFIs were experienced
  - Lower “campaign benchmark error  $\sigma$ ” is due to SEUs mainly occurring in main memory (ECC was off)
- **Future work:**
  - Proton data analysis to further clarify SEFI mechanisms in Cortex-A8
  - SEFI characterization on multicore architecture platform

# References

---



- <sup>1</sup>J.Mee et al., IEEE Aerospace conference, 2021
- <sup>2</sup>H.Quinn, SELSE workshop, 2019
- <sup>3</sup>H.Quinn et al., IEEE TNS VOL.62, NO.6, 2015
- <sup>4</sup>F.Irom, Guideline for ground radiation testing of microprocessors in the space radiation environment, 2008
- <sup>5</sup>NIST.gov accessed 12 May 22
- <sup>6</sup>P. E. Dodd and L. W. Massengill, IEEE TNS, VOL. 50, NO.3, 2003
- <sup>7</sup>H.Quinn et al., IEEE REDW, 2014
- <sup>8</sup>J.Carreira et al., IEEE TSE, VOL. 24, NO.2, 1998