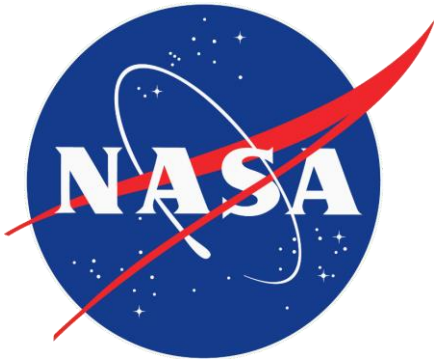


NEPP ETW 2021

MODEL- AND TESTING-BASED ASSURANCE OF CACHE MEMORY OF A SINGLE-BOARD COMPUTING SYSTEM IN RADIATION ENVIRONMENTS

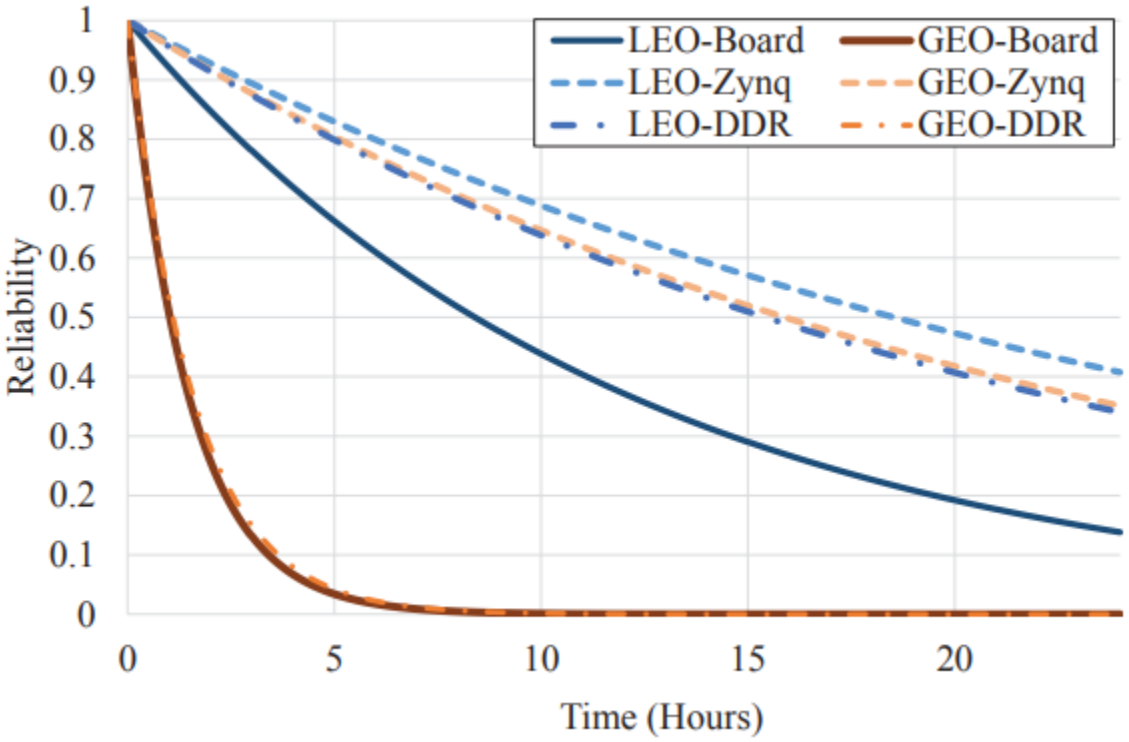
- Vanderbilt University: M. Reaz, P. Koncelik, A. F. Witulski, B. Bhuva, G. Karsai, N. Mahadevan, M. W. Rony, K. Li, B. Sierawski, A. L. Sternberg, R. A. Reed, and R. D. Schrimpf
- Alphacore Inc.: M. Turowski

Acronyms and Abbreviations

Vanderbilt Engineering

CPU	Central Processing Unit
FMECA	Fault Mode Effects & Criticality Analysis
HPC	High Performance Computing System
SEAM	System Engineering and Assurance Modeling
DFT	Dynamic Fault Tree
SysML	System Modeling Language
COTS	Commercial Off-the-shelf
FPU	Floating Point Unit
SEL	Single Event Latch-Up
SEFI	Single Event Functional Interrupt

COTS component in Space



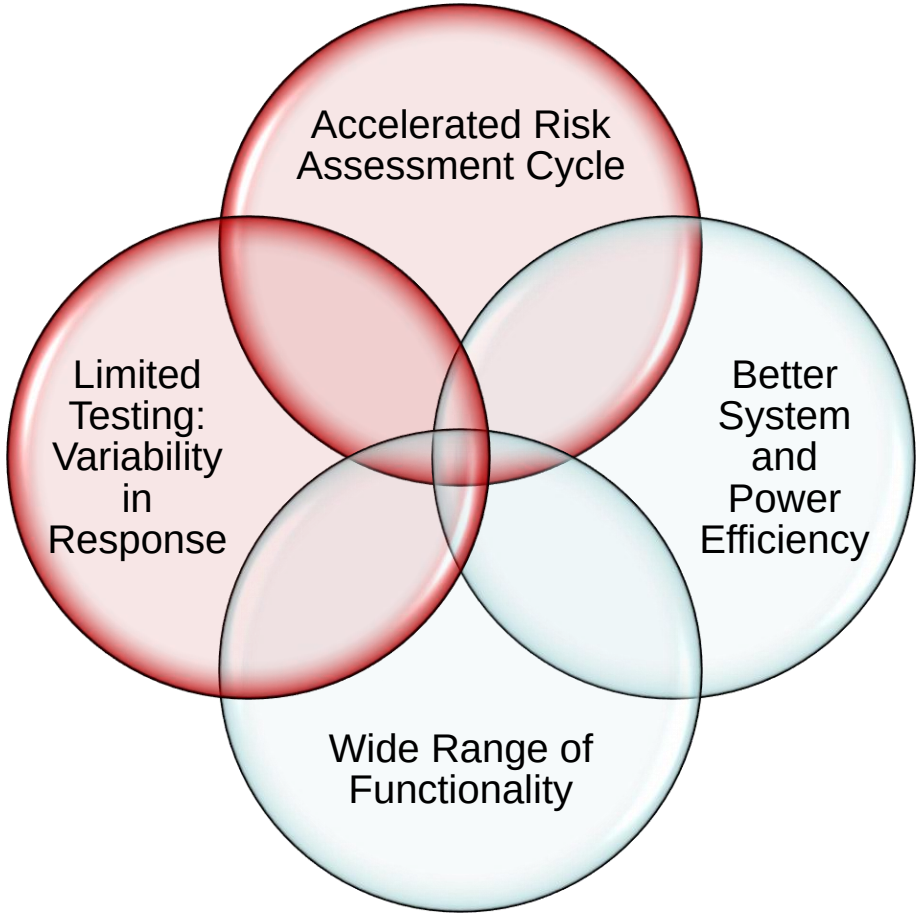
In some orbits, or with sufficient shielding, commercial electronics are almost as reliable as rad-hard electronics

NASA Goddard Space Flight Center SpaceCubev1.5 in CubeSat-like form factor featuring commercial Xilinx technology

C. Wilson & A. George, IEEE Aerospace Conf., 2016

Performance: COTS Components under Radiation in Space

Vanderbilt Engineering



Failure Mode Effects & Criticality Analysis

- Dynamic fault tree model
- Nested function-component dependence
- Identify the components with unknown criticality
- Experiment to measure the failure cross-section

- ❑ What is the minimal change in design/ component that would improve the reliability of a system with COTS component in space?
- ❑ Known failure rate for DDR and FLASH memory
- ❑ The failure rate for SRAM and CPU/GPU subsystems is not known especially within the context of a specific architecture, hardware, and OS configuration

Radiation tolerance is expected to vary depending on the hardware selection and mitigation by software design

STTR Phase II : Testing of COTS Systems in Space Radiation Environments

Vanderbilt Engineering

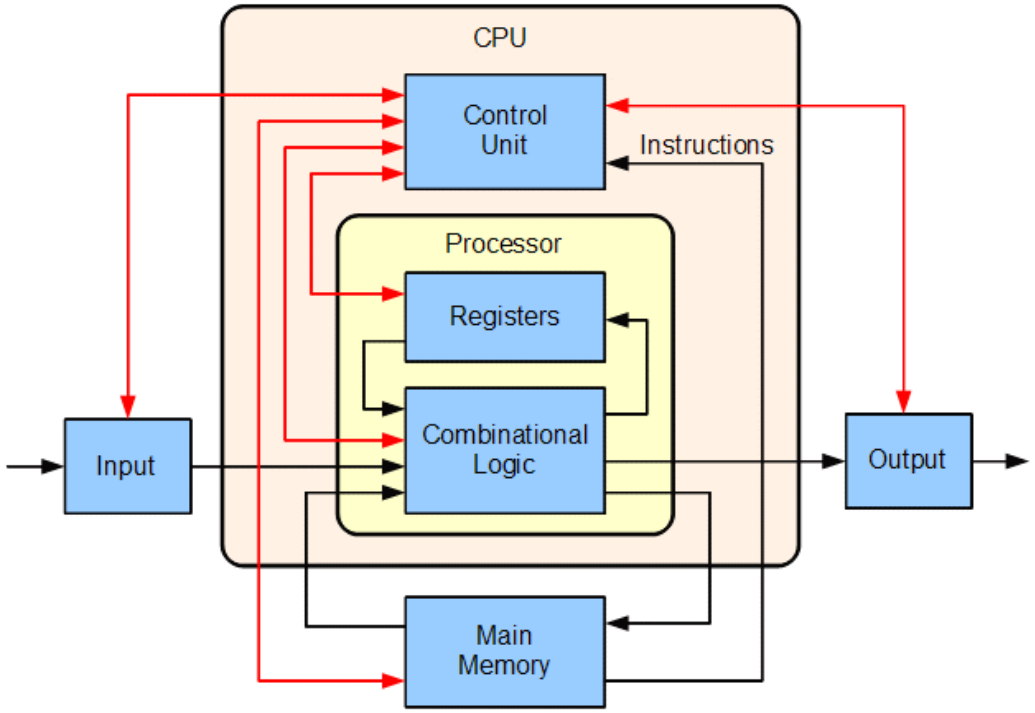
- NASA STTR 2019 Phase II Solicitation from Langley Research Center
- T6.05 Testing of COTS Systems in Space Radiation Environments

From the RFP: Investigate the feasibility of COTS electronics for *High Performance Computing (HPC)* in *space environments which are already heavily shielded*. It seeks strategies *based on a complete system analysis of HPC COTS* that include, but not limited only to, *failure modes* to mitigate radiation induced impacts to potential HPC systems in those highly shielded space environments.

Operation and Memory Access Flow of Computers

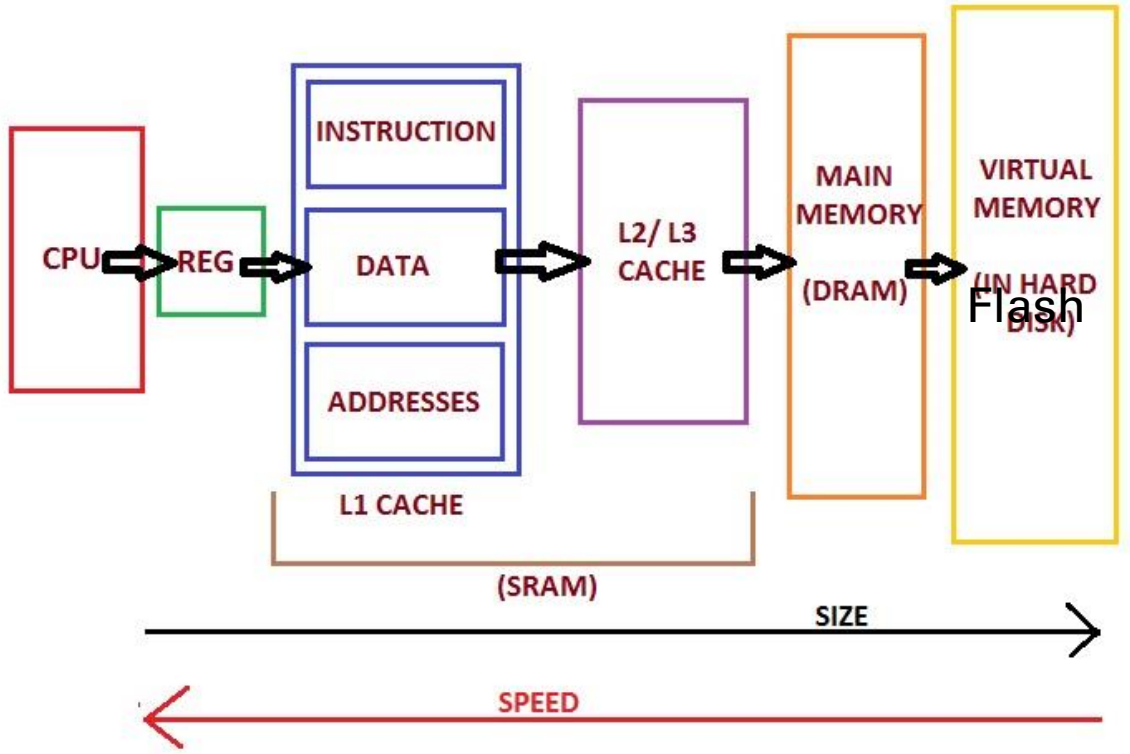
Vanderbilt Engineering

Uniprocessor CPU



Black lines —— data flow
Red lines —— control flow

Control Flow is more complex than data flow



Radiation Reliability of CPU & Cache Memory are largely unknown

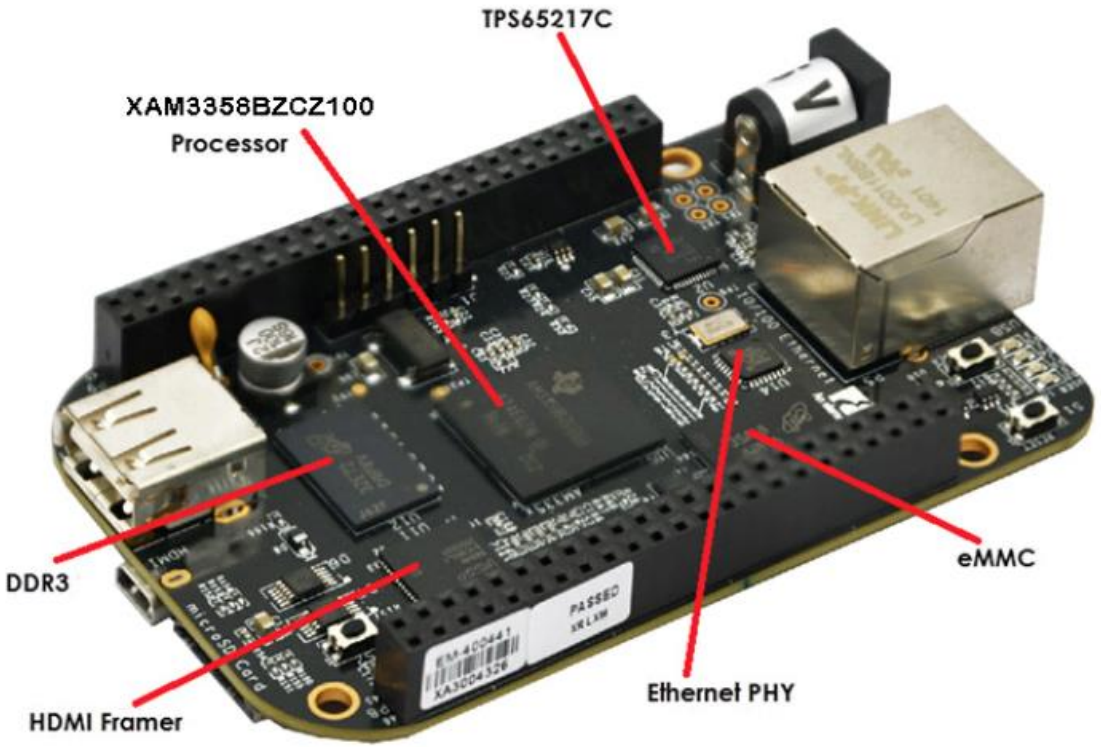
Computing Board Under Study: BeagleBone Black (BBB)

Vanderbilt Engineering

- CPU – Integer Core
- GPU – Neon Core (Floating point calculation, vectorized data processing)
- Memory – Cache, SRAM, EEPROM, FLASH, MMC, SD, ECC
- Interface – µSD, µHDMI, Ethernet, JTAG, GPIO, PWM, Serial, SPI, and I2C

BBB ~ smallest building block of high-performance computing

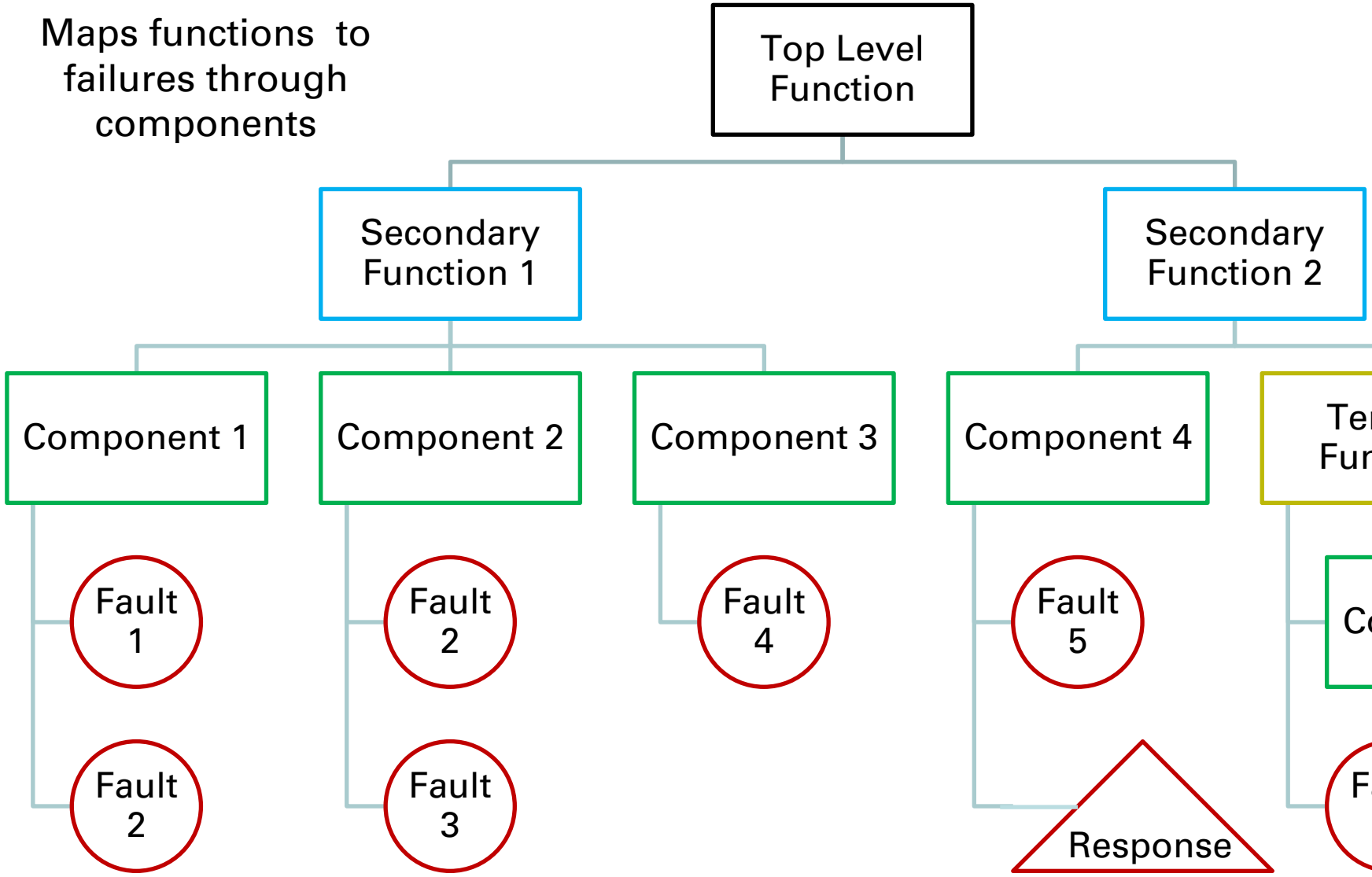
- Open-source community
- No heatsink on processor-SE Tests
- Good availability
- Low Power – Arm Instruction



Dynamic Fault-Tree

Vanderbilt Engineering

Maps functions to failures through components



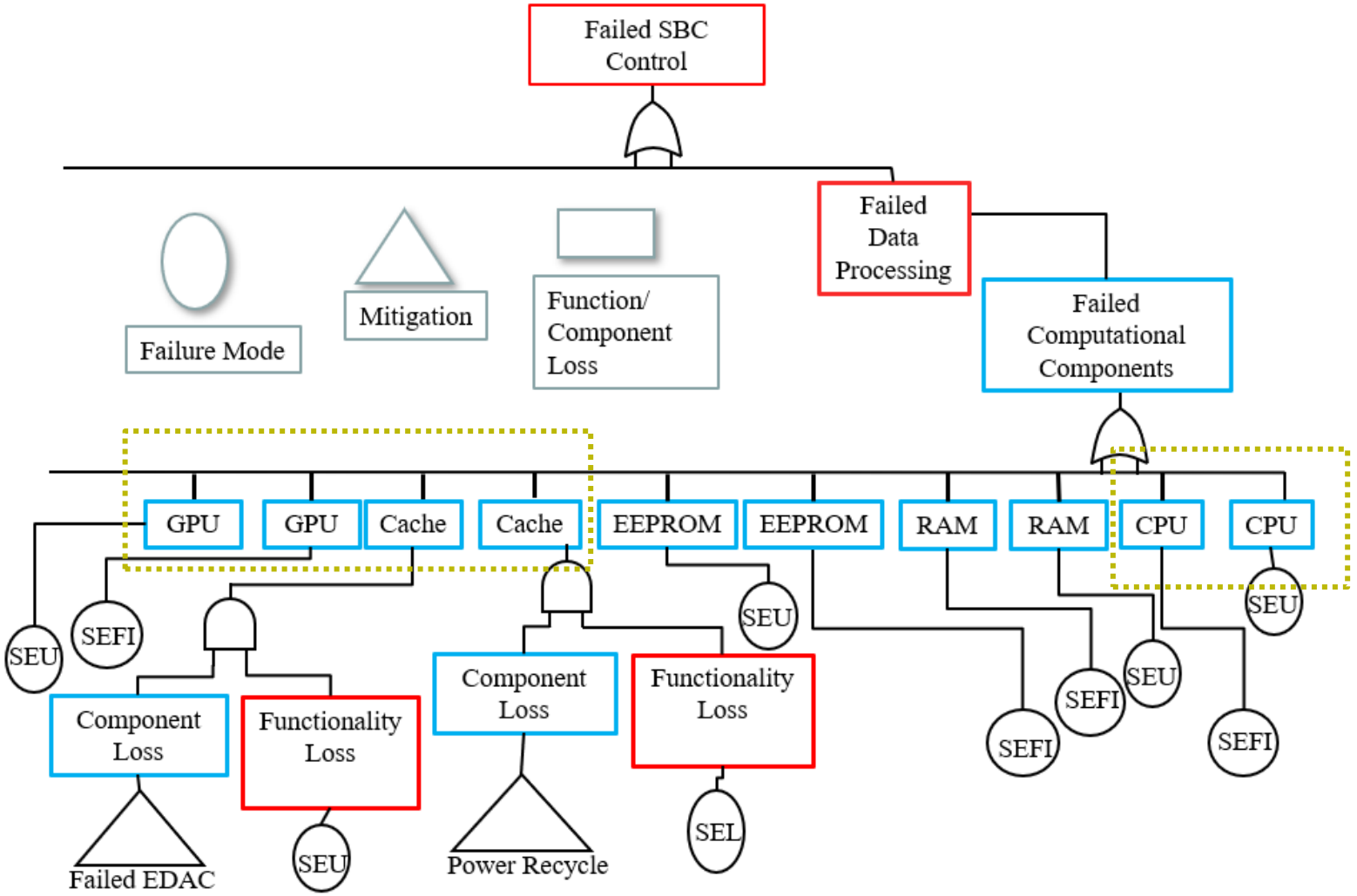
- Failures in uPs can be induced from a sequence of events
- Hierarchical function-component dependance captures the sequence-dependent behavior

Simplified diagram shown here for brevity. Derived behavior are possible to be nested at any order.

FMECA Analysis: Dynamic Fault Tree Model for BBB

Vanderbilt Engineering

Rate, origin, and/or type of radiation induced failures are less known for cache, CPU, and GPU/FPU



Modeled in SEAM
@https://modelbasedassurance.org/

Candidate Computing Board: BeagleBone Black (BBB)

- Configurable CPU and GPU via open-source tool
- Allow placement of data and execution in user-defined core and memory units
- Parallel read access of CPU memory through execution cycle
- JTAG interfaces enable real-time debug
- Configurable ECC and parity mode for L2 cache

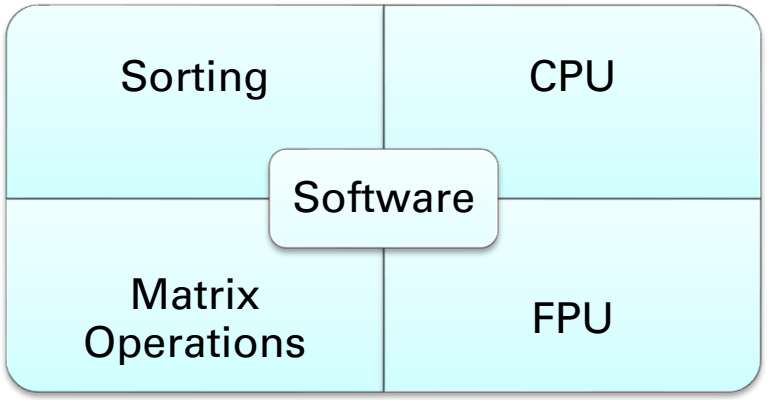
Access to failure monitor flags (control-registers) via JTAG interface.

Goal : Distinguish the type and **origin** of faults in the µP chip including Cache memory.

Challenge : Several SEU-induced faults are not traceable due to system complexity (CPU hang-ups). ECC can mask the origin of the failures.

Proposed Solution : Distinguish memory R/W and access faults to those in control registers.

Benchmark Software Tests



Entire SOC is irradiated including cache, and execution units:

- ECC Correction only in L2 Cache (Data + Instruction)
- No Bit Interleaving
- Sorting in CPU
- Matrix Multiplication in GPU (Neon Coprocessor)
- Data and code is placed in CPU cache (irradiated)
- HEAP, STACK is placed in DDR (unirradiated)
- Transferrable algorithms written in C
- System controls in ARM assembly (unique to AM335x μP)
- Tests are repeatable for other computing hardwire system.

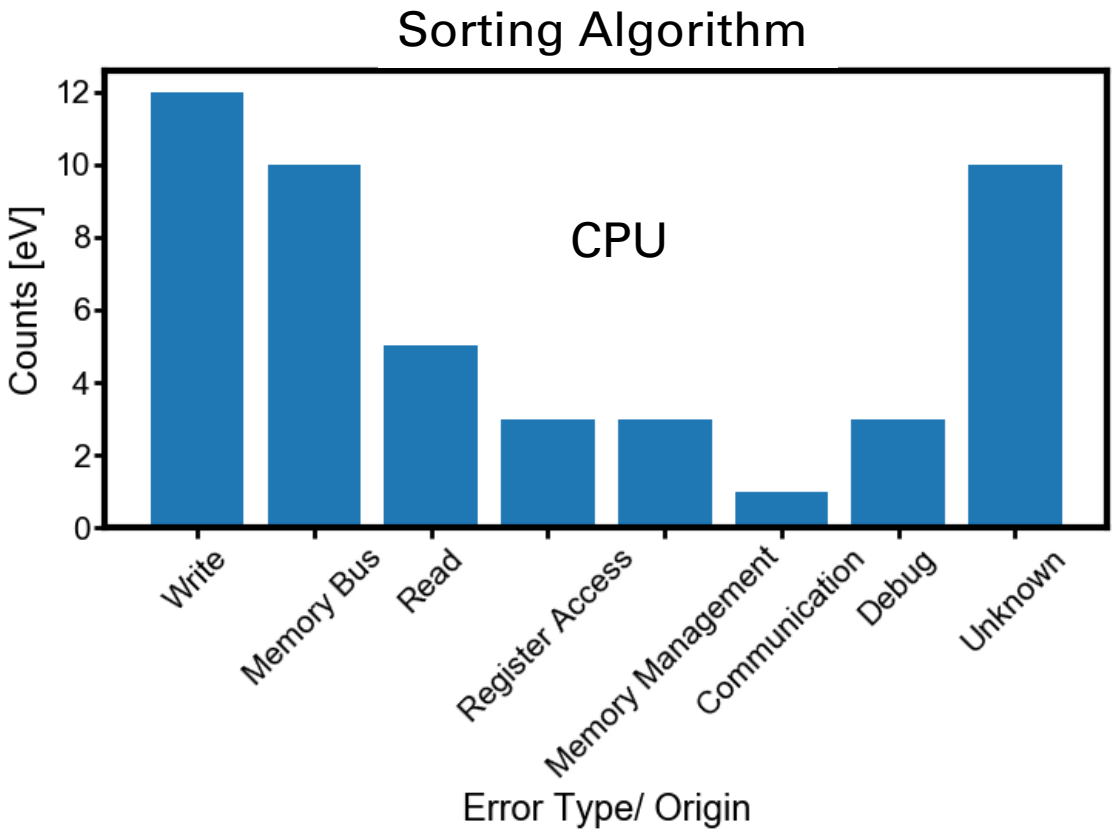
Hardware configurations and software implementations are done by minimalistic open-sourced real-time operating system by vendor (Texas Instruments).

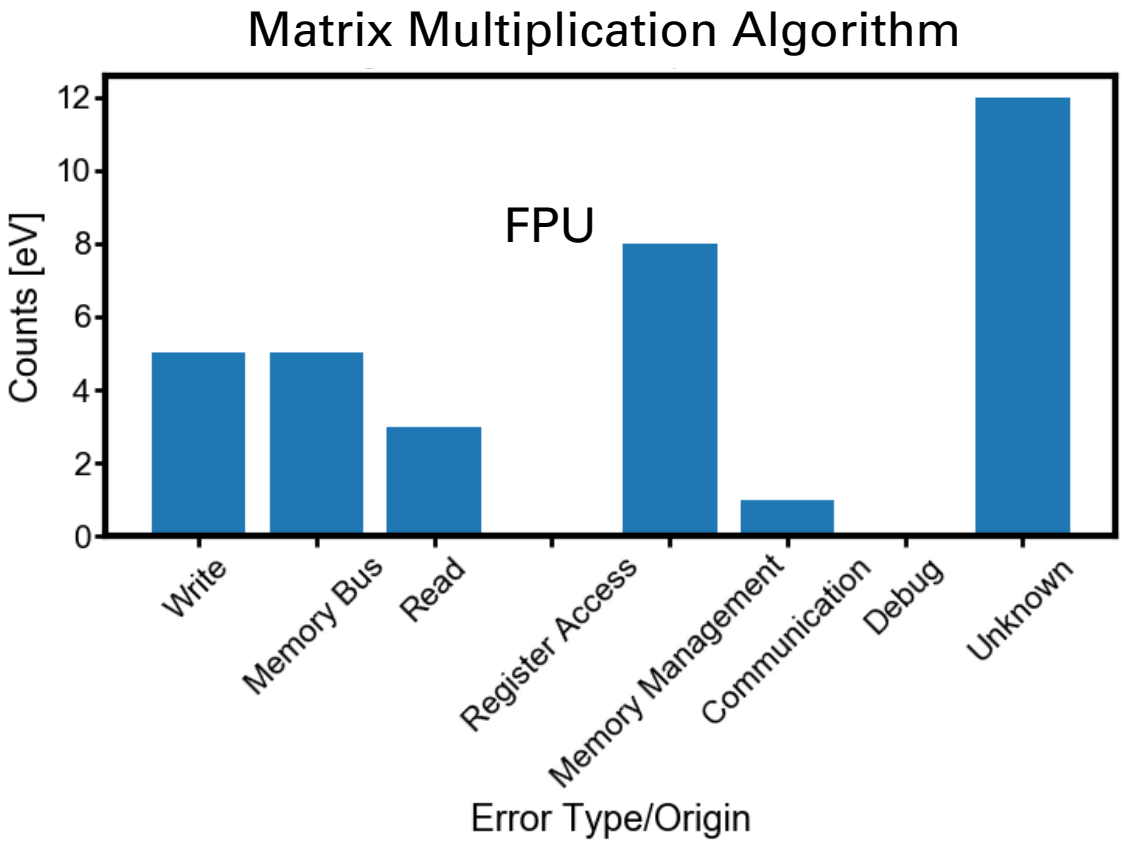
Guidelines followed from
H. Quinn, et al., TNS, 2015

Results (Work in Progress)

Vanderbilt Engineering

Radiation Source: Americium
Flux: 1000 alphas/mm2-sec
Source to target distance in Air: 2 cm

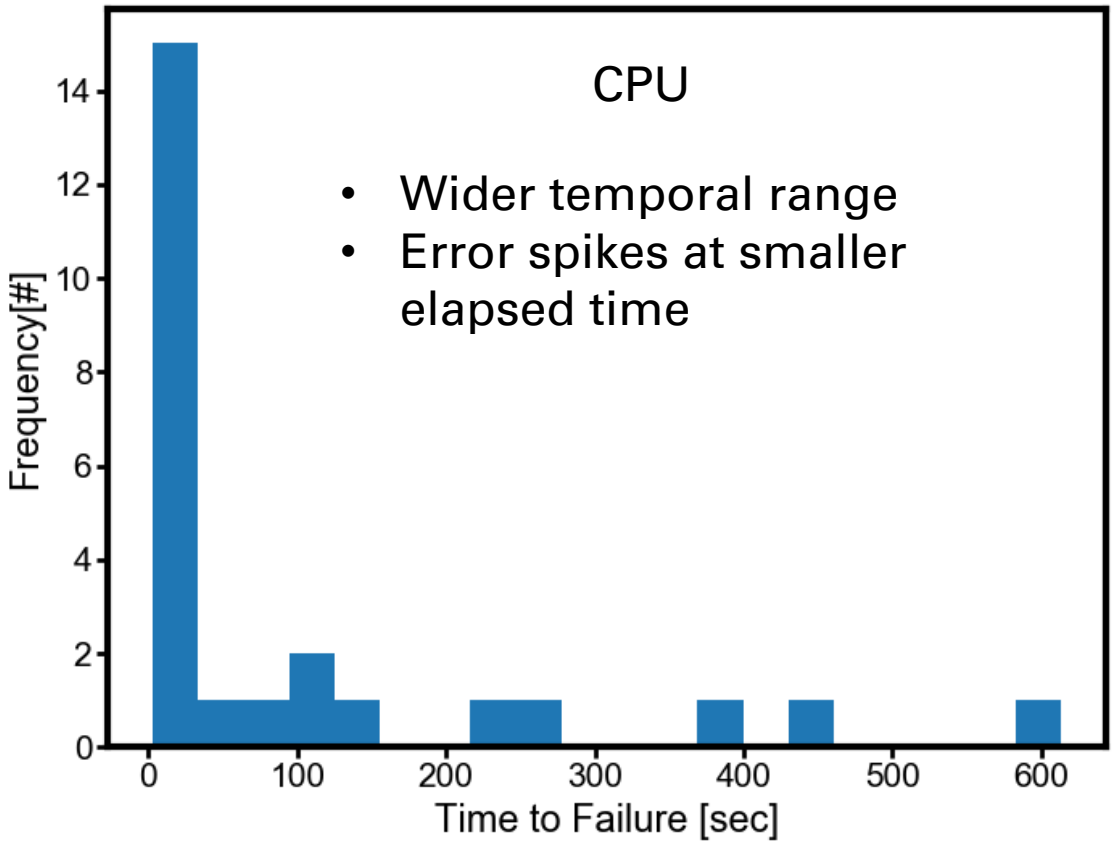




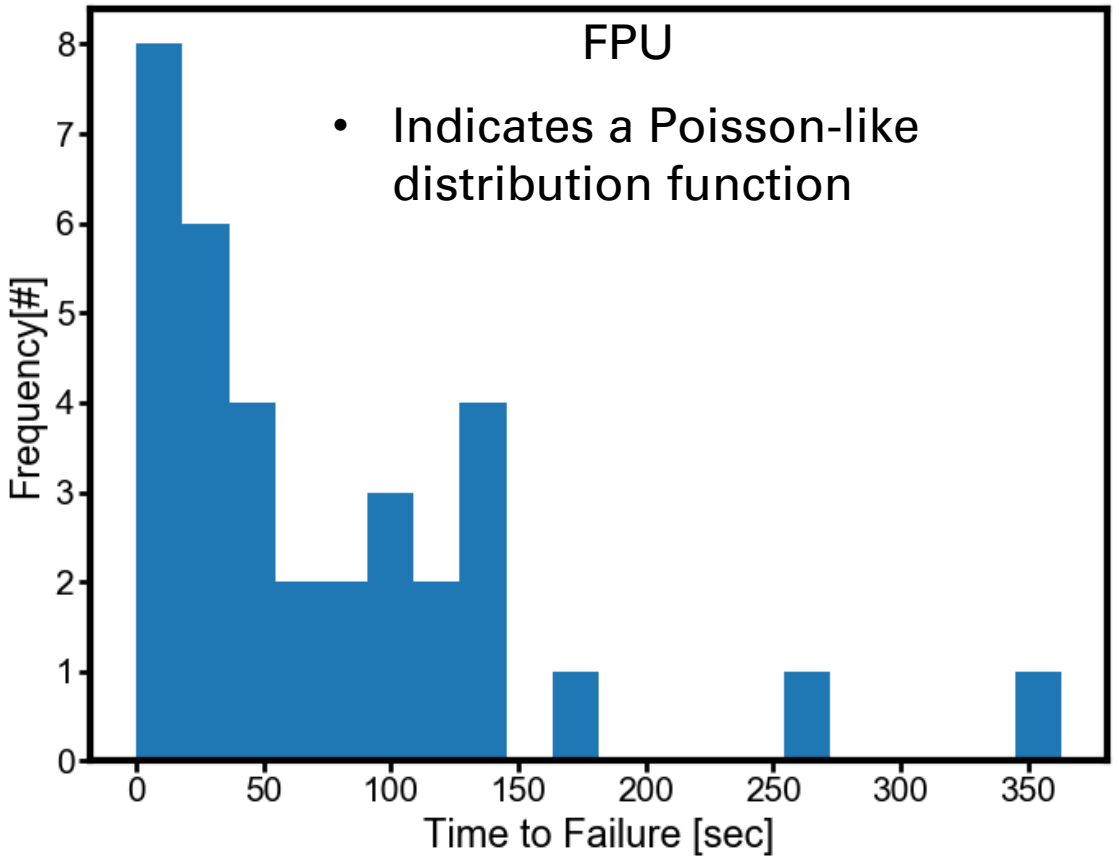
Results (Work in Progress)

Vanderbilt Engineering

Sorting Algorithm



Matrix Multiplication Algorithm



L2 Cache Error: Data and Instruction Memory (Work in Progress)

Algorithm	Data Fault in L2	Instruction Fault in L2	Total Number of Faults in IC
Sorting	4	2	45
Matrix Multiplication	4	4	35

SE-induced failures in L2 cache is a significant fraction to that of the total failures in µP IC

To monitor the L1 cache during static tests would be interesting ~ work in progress

Summary

Vanderbilt Engineering

Dynamic Fault Tree analysis provide the critical components of functionality——helps identify subsystems for testing

Derived method for independent measurement of the multi-core µP subsystems

Preliminary results show promise for differentiating errors originated in CPU and FPU.