B1-26: Fault-Tolerant Techniques for Heterogenous Computing Architectures

Mission-Critical Computing
NSF CENTER FOR SPACE, HIGH-PERFORMANCE, AND RESILIENT COMPUTING (SHREC)

SHREC Annual Workshop (SAW25-26)

University of Pittsburgh
BYU BRIGHAM YOUNG UNIVERSITY
VIRGINIA TECH
UF UNIVERSITY of FLORIDA

January 13-14, 2026

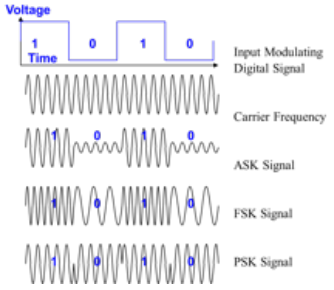
Dr. Mike Wirthlin & Dr. Jeff Goeders

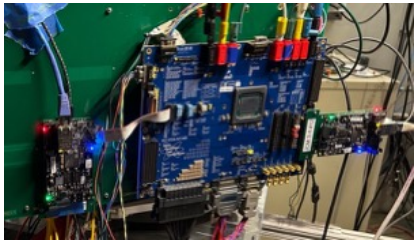
Garrett Smith, MS
Jacob Brown, MS
Alan Howe, MS
Scott May, MS
Brigham Morgan, BS
Callin Schuring, BS
John Rowberry, BS
Derek Harker, BS
Taeook Kim, BS

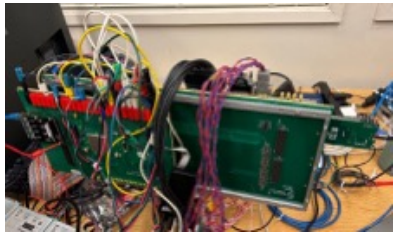
Number of requested memberships ≥ 4

Project Tasks

- **Task 1**: Reliable Deep Learning
  - Add fault tolerance to existing deep learning implementations
  - Explore alternative ML architectures to improve reliability
  - Exploit emerging implementation tools for ML applications
- **Task 2**: Versal Reliability
  - Apply proven techniques to Versal 2nd Generation
  - Demonstrate NoC SEU mitigation techniques
  - Develop Versal distributed recovery approaches
- **Task 3**: Radiation Testing of Complex SoC
  - Perform radiation tests on complex SoC devices
  - Develop new radiation testing infrastructure
  - Fault analysis from existing radiation test data







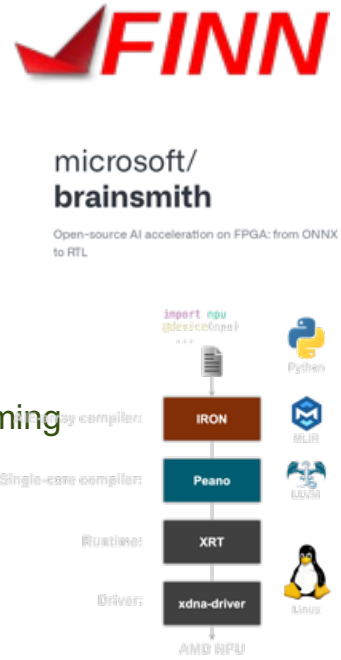
Task 1: Reliable Deep Learning

- Deep learning on Versal
  - Improves on-board data processing, reducing network bandwidth
  - Heterogeneous computing provides high performance at lower power compared to GPU
- Improve custom YOLO design to better rival DPU IP
  - Reduce utilization, increase throughput
  - Move more compute from PL to AIE
- Explore different model types and applications
  - VGG, SSD, MobileNet CNNs
  - UNet for image segmentation
  - LSTM for anomaly detection
- Apply same testing methodology to new applications
  - Performance evaluation, fault injection testing, etc.

Segmentation example

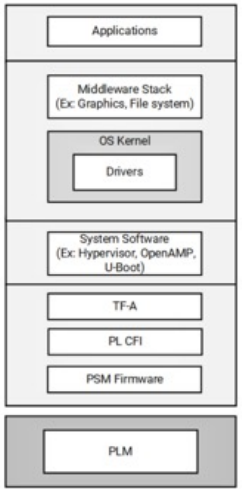
Tools and Reliability Approaches

- Evaluate existing tools for DL on Versal (non-DPU)
  - FINN – AMD open-source tool for converting models to hardware
  - Brainsmith – Microsoft tool built on FINN that enables design space exploration
  - IRON – AIE programming using MLIR
  - Manual – manual PL and AIE design using Vitis HLS and AIE API
- Improve Deep Learning reliability on Versal
  - Develop approaches for recovery from DPU timeout without reboot
    - May require building DPU into a Dynamic Function Exchange (DFX) design to enable reprogramming
  - Explore TMR variations on manual YOLO, AMC designs
    - Vary primitives selected for TMR
    - TMR PL-AIE interface tiles
    - Adjust design before TMR (e.g. remove floating point operations used in re-scaling)



Task 2: Versal Reliability

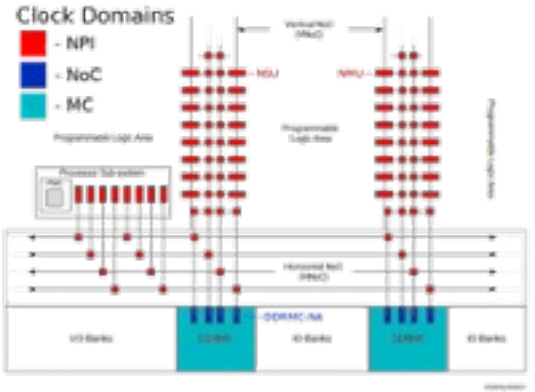
- Versal 2nd Generation
  - Apply techniques used in Versal 1st generation to 2nd Generation
  - Extend techniques to new 2nd gen features (L3 cache, processors, etc.)
- Improve Linux Reliability Techniques
  - Fault tolerant drivers (programmable scrub rates)
  - Additional CoreSight trace support
  - Integrate RPU booting within Linux
- New Firmware approaches
  - Programmable firmware scrub rates
  - Integrate latest firmware version
  - Bus recovery approaches
  - Improved build process to migrate firmware versions

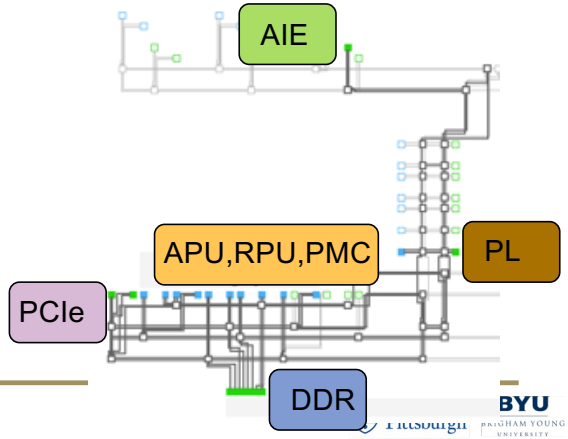


B1

Network on Chip (NoC) Reliability

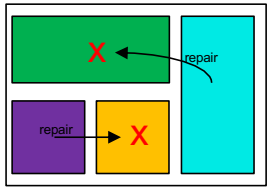
- NoC Functionality Essential for Reliable Device Operation
  - NoC failures appear to be primary cause of APU/Linux hangs
    - Symptoms?
  - Limited feedback on NoC failures
- NoC Radiation Testing
  - Full register monitoring and dumping
  - High-bandwidth test architectures
- NoC Register Scrubbing
  - Implement active NoC register scrubbing (outside of APU)
    - Internal: RPU, PLM, or PSM
    - External (JTAG/DAP/PCIe)
  - Incorporate “smart” register scrubbing
- NoC Recovery Approaches
  - “Safe” register repair
  - NoC restart sequences

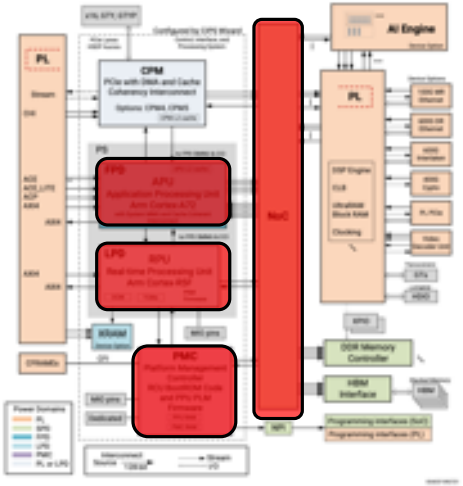




Distributed Recovery Approaches

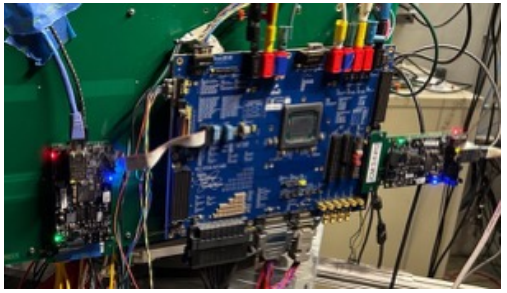
- Observations From Radiation Tests
  - Device failures occur within local regions
  - No global failure modes observed
    - Other “active” regions operational during regional failure
    - Other regions unaffected by most regional failures
- Approach: Exploit operational regions to repair failed region
  - Avoid full device reconfiguration/repower by repairing locally
  - Distributed repair functionality spread among active regions
- Recovery regions of interest
  - APU/Linux: ability to reboot Linux on APU without reconfiguration
  - RPU: embedded real-time processor restart
  - PLM (PSM): restart PLM functionality when hung
  - NoC: repair NoC without rebootingAPU0/Linux recovery or reboot?
  - Interconnect: recover from blocking interconnect failures





B1

Task 3: Radiation Testing of Complex SoC Devices



- Radiation testing essential in improving SoC reliability
  - Validate mitigation techniques
  - Identify failure modes
  - Improve radiation testing efficiency
- Texas A&M (Heavy Ion)
  - Versal NoC Testing
  - Validate SoC mitigation approaches
  - Demonstrate novel fault analysis
  - High-speed data extraction (HSDP)

TEXAS A&M UNIVERSITY
Cyclotron Institute

- UC Davis (Proton)
  - Flux-dependent processor testing
  - High-flux SEFI testing
  - Deep Learning Application Testing

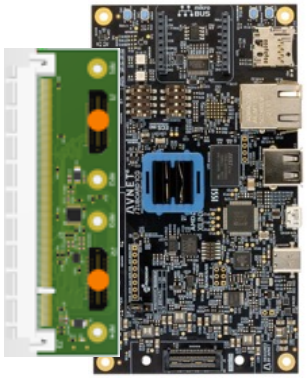
UCDAVIS
CROCKER NUCLEAR LABORATORY

B1

SoC Radiation Testing Infrastructure

- Improved infrastructure essential for complex SoC Testing
  - Extract data more quickly/efficiently
  - Support multiple, high-speed data interfaces
- ZUB JCM PCIe
  - Replace PolarFire PCIe infrastructure with MPSoC based PCIe
  - Higher speed, consistent software infrastructure
- High-Speed Debug Port (HSDP)
  - High-speed serial I/O port on AMD for debugging
    - High-speed extraction of traces, memory, and internal state
    - Requires custom hardware (SmartLynq+)
  - Demonstrate successful use during radiation testing
  - Explore custom HSDP adapter for JCM
- JCM Software Improvements
  - Support AMD/Xilinnx Sysmon (GUI support for real-time analysis)
  - Improved CoreSight trace (DAP)
  - DAP improvements (error handling, concurrent execution)
- Custom Testing Interfaces
  - XRTC & Sandia Test Boards

ZUB PCIe



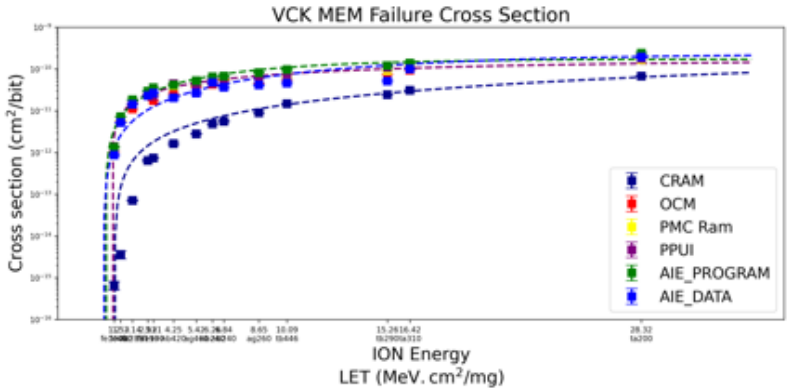
AMD SmartLynq+

JCM w/XRTC
SMAP

Versal Test Data Analysis

B1

- Significant Versal Radiation Test Data
  - 9 distinct experiments
  - ~44,000 log files
  - 95 GB (5 GB text, 90 GB binary)
- Data Analysis Questions and Tasks
  - Multi-cell upsets (MCU): as a function of LET, tilt, and ion
  - Flux-rate dependencies: identify linear vs. polynomial failure rate
  - Comprehensive list of failure types (from all tests)
  - Finalize cross-section and LET curves
  - Run-time environment specific failures (Linux vs. Bare Metal)
- Higher Level Failure Analysis
  - Identify common failure cause and effect
    - Analyze symptoms from multiple sources
    - Generate time-line of symptoms
    - Categorize into specific failure profile
  - Create failure hierarchy for all failures
    - Many failures contain common symptoms
    - Failure symptoms appear “hierarchal”
  - Post-failure register analysis
    - Comprehensive review of all logged registers
    - Automatic categorization of failure based on register signature

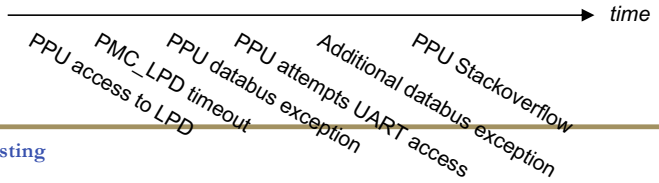


Fault Analysis Example:

Primary Symptoms:
- PPU Hangs w/o Error
- DAP UART crashes w/o Error

Secondary Symptoms:
- PPU stack overflow (PPU register)
- PSM/PLD/FPD register inaccessible
- Timeout flag on PMC_LPD interconnect
- APU/RPU responsive to DAP debug requests

Tertiary Symptoms:
- PPU exception address points to Exception handler
- PPU r5 and r22 indicate databus exception



Questions?

B1