Politecnico di Torino

Department of Control and Computer Engineering

A MICRO ARCHITECTURAL EVENTS AWARE REAL-TIME EMBEDDED SYSTEM FAULT INJECTOR

ENRICO MAGLIANO, ALESSIO CARPEGNA, ALESSANDRO SAVINO, STEFANO DI CARLO

ABSTRACT:

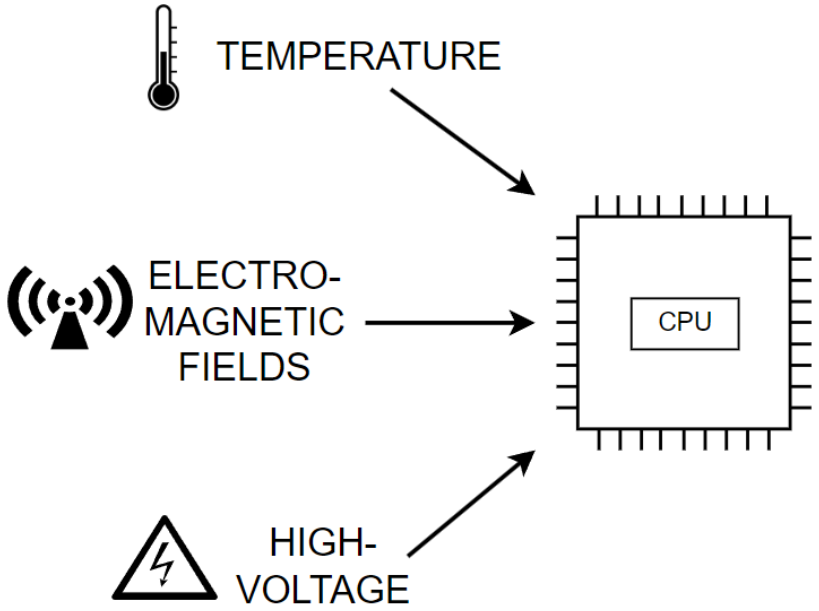
The increasing system complexity poses significant challeges to the RELIABILTY and TRUSTWORTHINESS of the Safety Critical Real-Time Embedded Systems (SACRES).

Our Fault Injector is designed with the novelty to facilitate the monitoring of micro-architectural events, harnessing the Performance Monitoring Unit (PMU)

PROBLEM:

- Vary phenomena can cause Soft Errors.

- Producing Benign or Erroneous outcomes:
  - Benign
  - Silent Data Corruption
  - Crash
  - Hang
  - Reboot

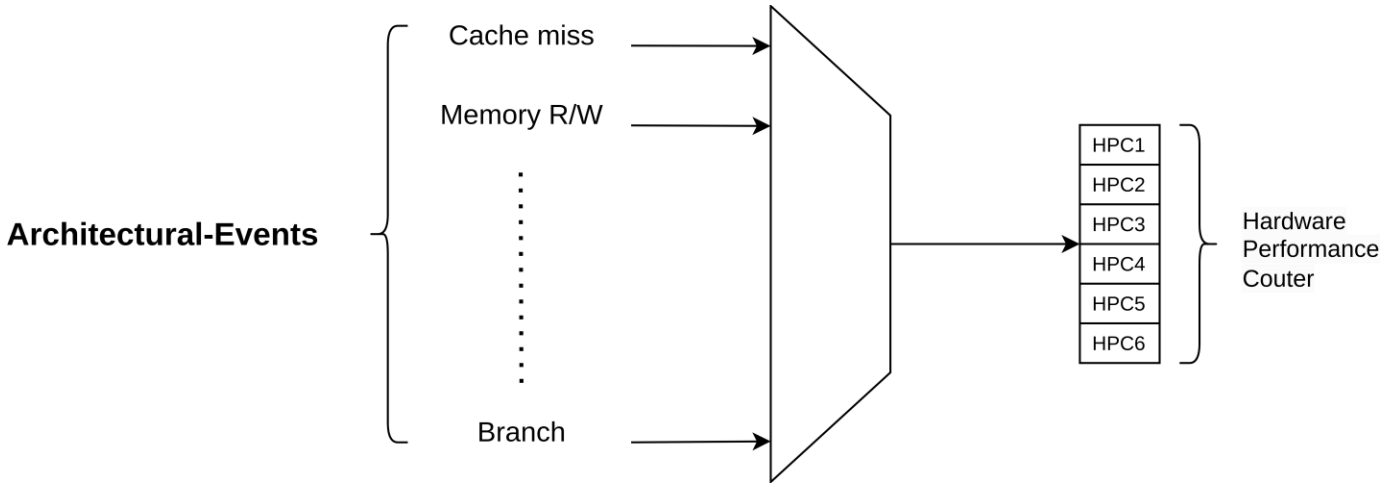


FAULT INJECTOR

FI tailored for SACRES capable of inject bit-flips in real embedded hardware, by exploiting the debug unit of the modern CPUs.

It guarantees:

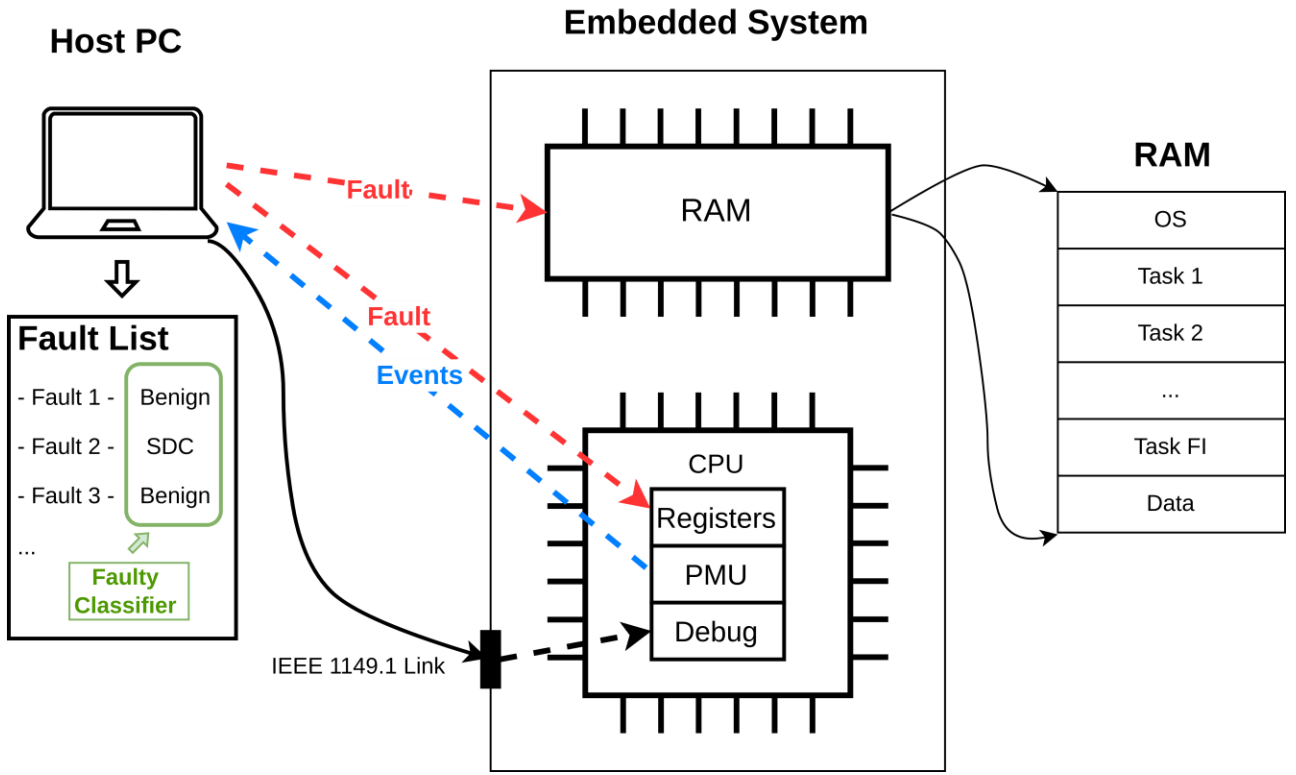
- Controllability
- Observability
- Repeatability



Repeat many times the same fault to collect all the available micro-architectutal events
Since the number of available HPC is less than the number of events.

HIGH LEVEL ARCHITECTURE

Host machine creates the fault list, then using the debug unit controls the execution to perform the injection in the CPU registers and in the RAM of the Embedded System.
Then collects all the architectural events from the PMU.



FAULT REPRESENTATION

(Location, Line of Code)

Compose by location (CPU/RAM), address or registers, and bit position.

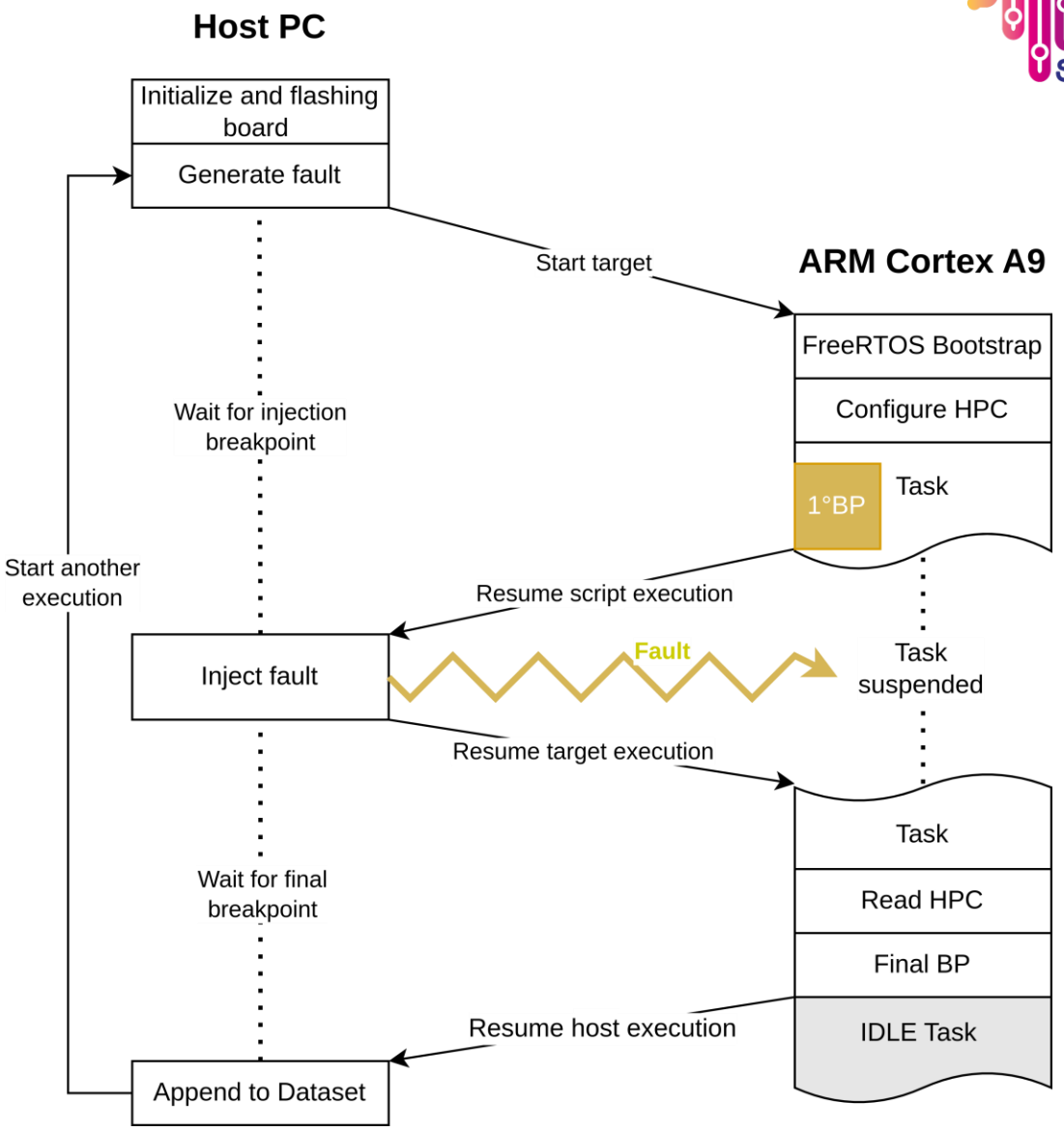
Address	Hex dump	ASCII
0028FEA4	00 00 00 00 C8 00 00 00	È
0028FEAC	64 00 00 00 30 15 48 00	d 0§H
0028FEB4	19 00 00 00 84 FF 28 00	├ „ÿ(
0028FEBC	EE 13 40 00 01 00 00 00	î‼@
0028FEC4	70 15 48 00 18 1F 48 00	p§H ↑ H
0028FECC	00 00 00 00 00 00 00 00	
0028FED4	00 00 00 00 00 00 00 00	

The Line of Code (LoC) where the breakpoint is set to stop the execution and inject faults.

```
9   int main(){
10      int fd = open("myPipe", O_RDONLY);
11      char c;
12      int res = 1;
13      while (res > 0){
14          res = read(fd, &c, 1);
15          printf("%c", c);
16      }
17      close(fd);
18      return 0;
19  }
```

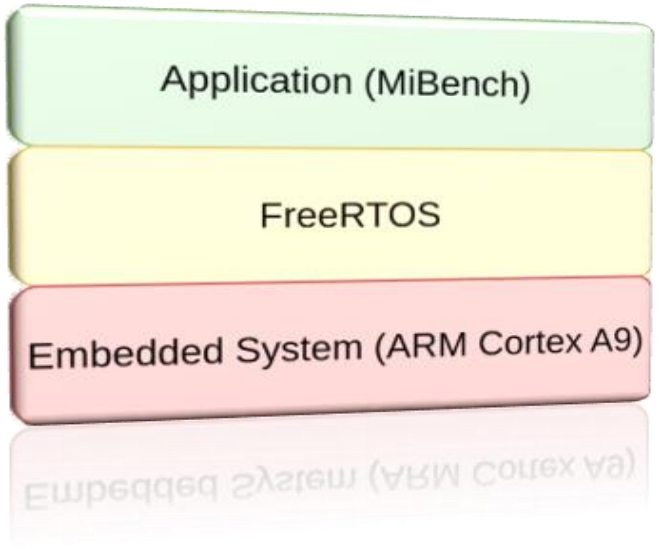
IMPLEMENTATION DETAILS

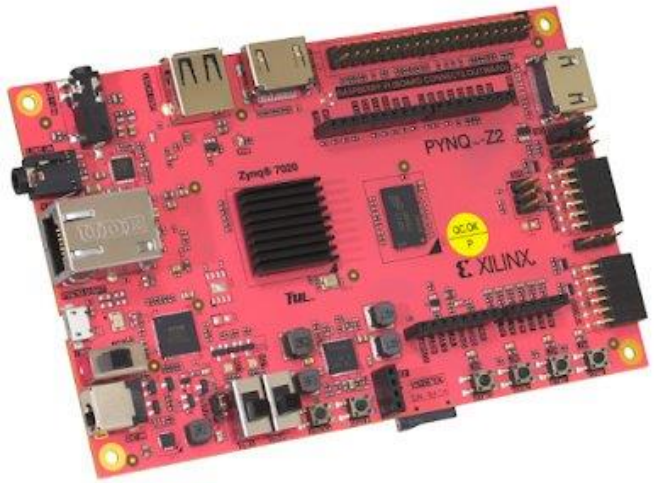
Python script able to control target execution thanks to XSCT commands. The script sets a random breakpoint, configures PMU, and performs a bit-flipping (reading and writing CPU registers). This is repeated many times to collect a large Dataset of executions.



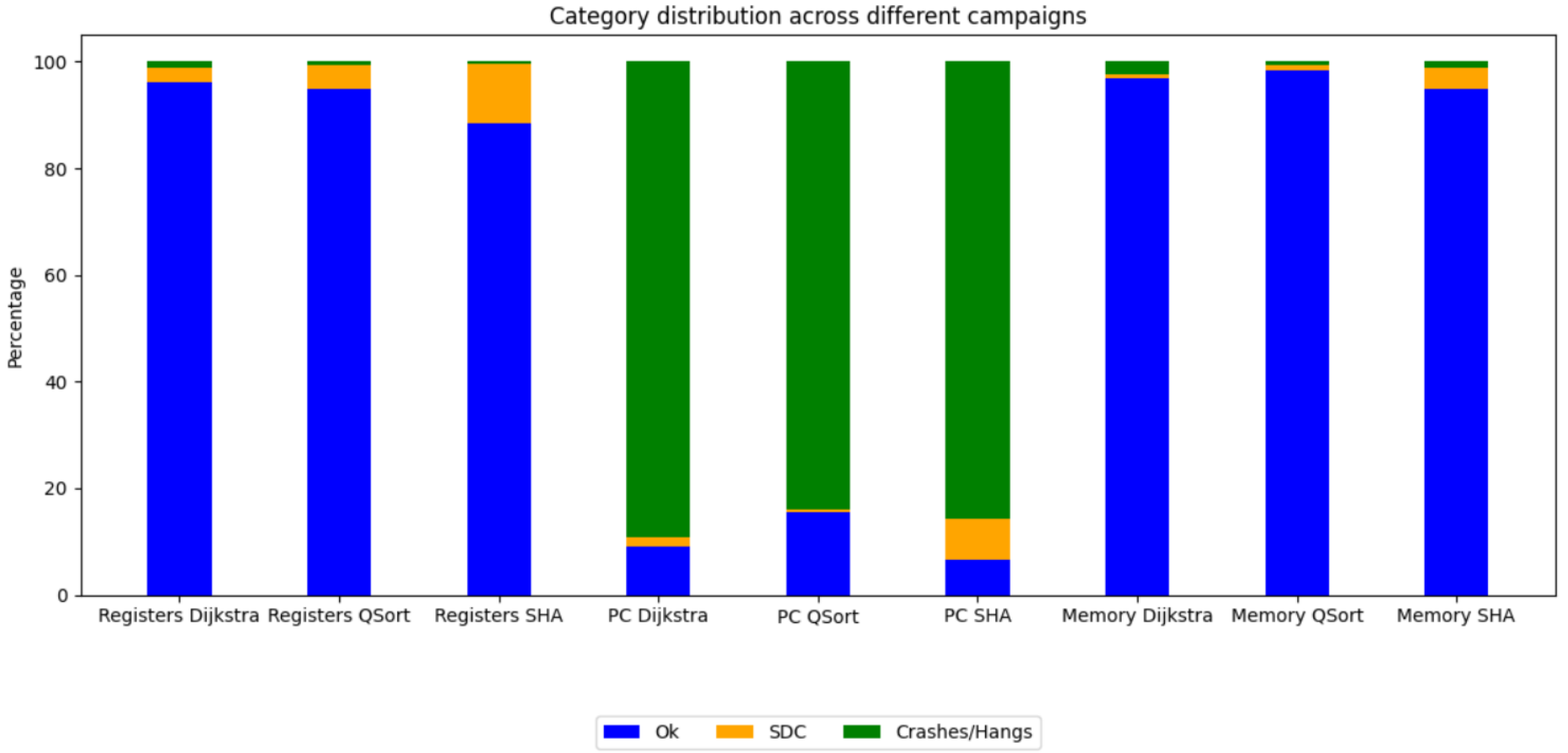
EXPERIMENTAL SETUP

- **Hardware:** Xilinx Pynq z2 board with on board Arm Cortex A9.
- **Software:** FreeRTOS real-time operating system and MiBench benchmark applications (Dijkstra, QuickSort, SHA).

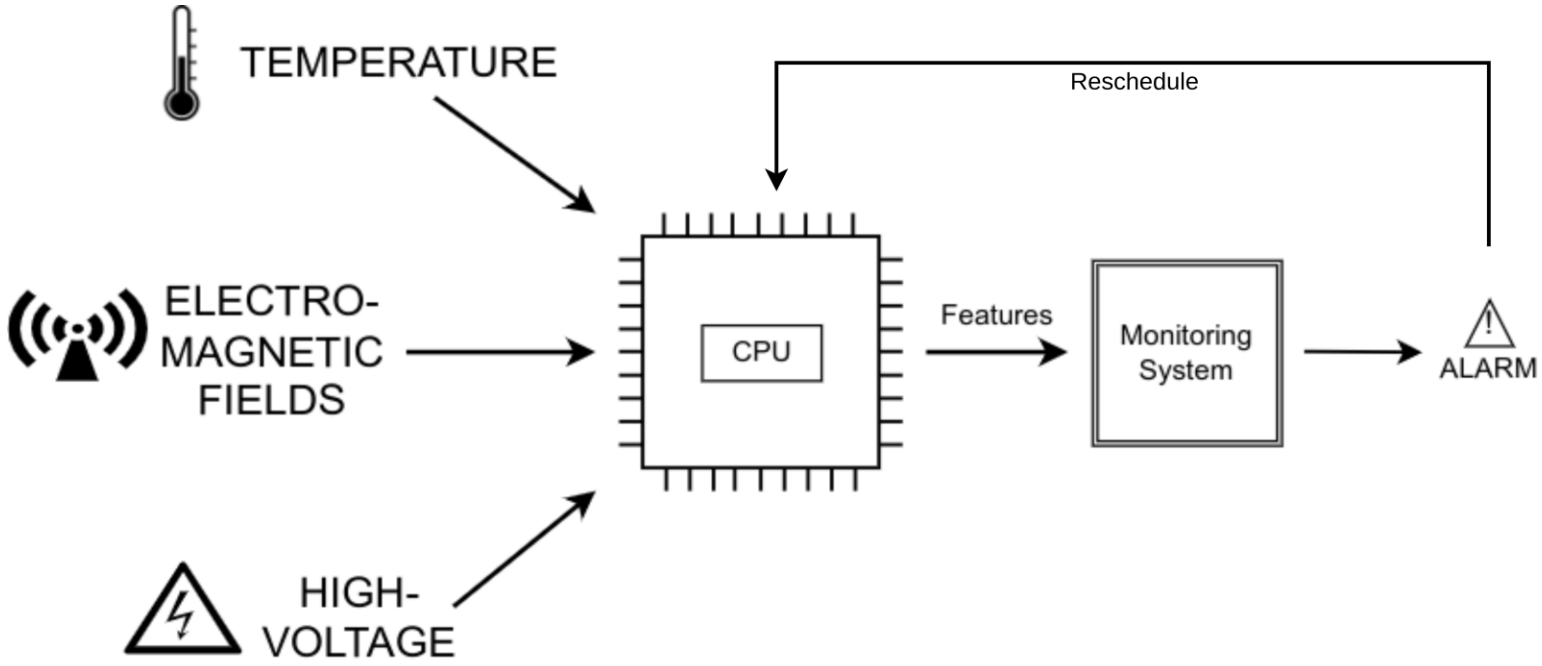




FAULT INJECTION CAMPAIGN RESULTS

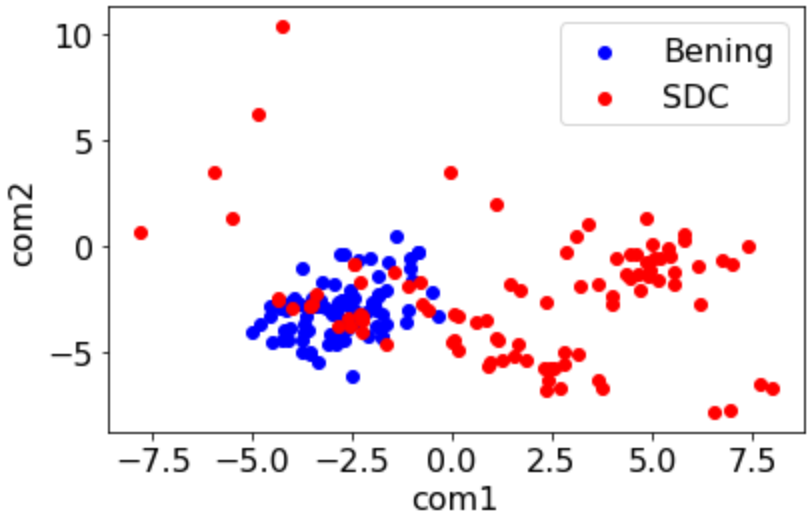


FUTURE WORK: COMPLETE FRAMEWORK

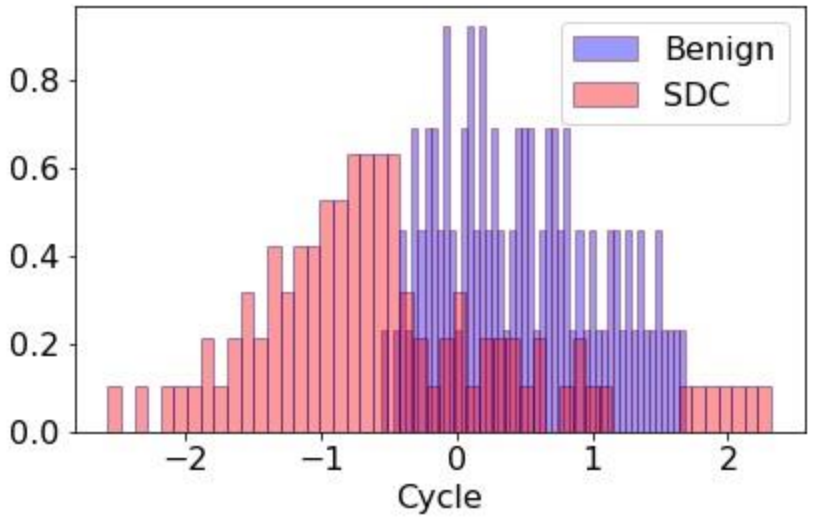


- Final framework with monitoring system able to detect SDC and reschedule the task.

SCATTER AND HIST PLOTS



- Scatter plot of the PCA-2



- Histogram plot of the two categories Benign and SDC

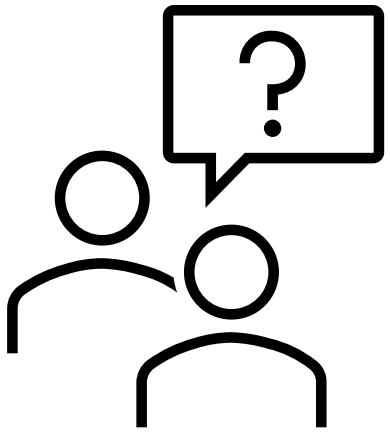
CONCLUSION

The paper introduces a novel fault injection (FI) environment tailored for SACRES, enabling real-time bit-flip injection in embedded hardware via CPU debug units, includes a profiling tool utilizing PMU for analyzing architectural events.

URL: (https://github.com/smilies-polito/marvin)

Scan For Code

THANKS FOR YOUR ATTENTIONS



Scan For Code