An End-to-End Analysis of EMFI Attacks on Bit-sliced Post-Quantum Implementations

Richa Singh, Saad Islam, Berk Sunar, Patrick Schaumont

Motivation

Application of Bit-slicing for protection of Lattice-based Post-Quantum algorithms is still unexplored.

Bottom-up approach: Bit-slice Intra-Instruction Redundant design for Number Theoretic Transform (NTT).

- **Key building blocks:** NTT/Inverse NTT, Polynomial Multiplication.
- **Technique:** Bit-slicing, N-bit processor datapath is treated as N parallel single-bit datapaths.
- **Countermeasure:** Dual data-redundant for NTT.

Evaluation: Efficiency of countermeasure to detect EM Fault Injections.

- **Algorithm:** Dilithium, a digital signature finalist for NIST PQC competition.
- **Target Device:** 667 Mhz Arm Cortex-A9 processor integrated in a Xilinx Zynq SoC.

EMFI Parameters Search Process

Goal: Optimize ***spatial location***, ***timing*** and ***intensity*** of EM Pulses to maximize the probability of detected successful faults injections.

Faults Classification: No Effect, Crash, Faults Detected and Faults Not Detected.

Experiment 1:

- Right Upper Quadrant Scan
- Random between 0 ns and 10M ns
- Random between 80% and 100%

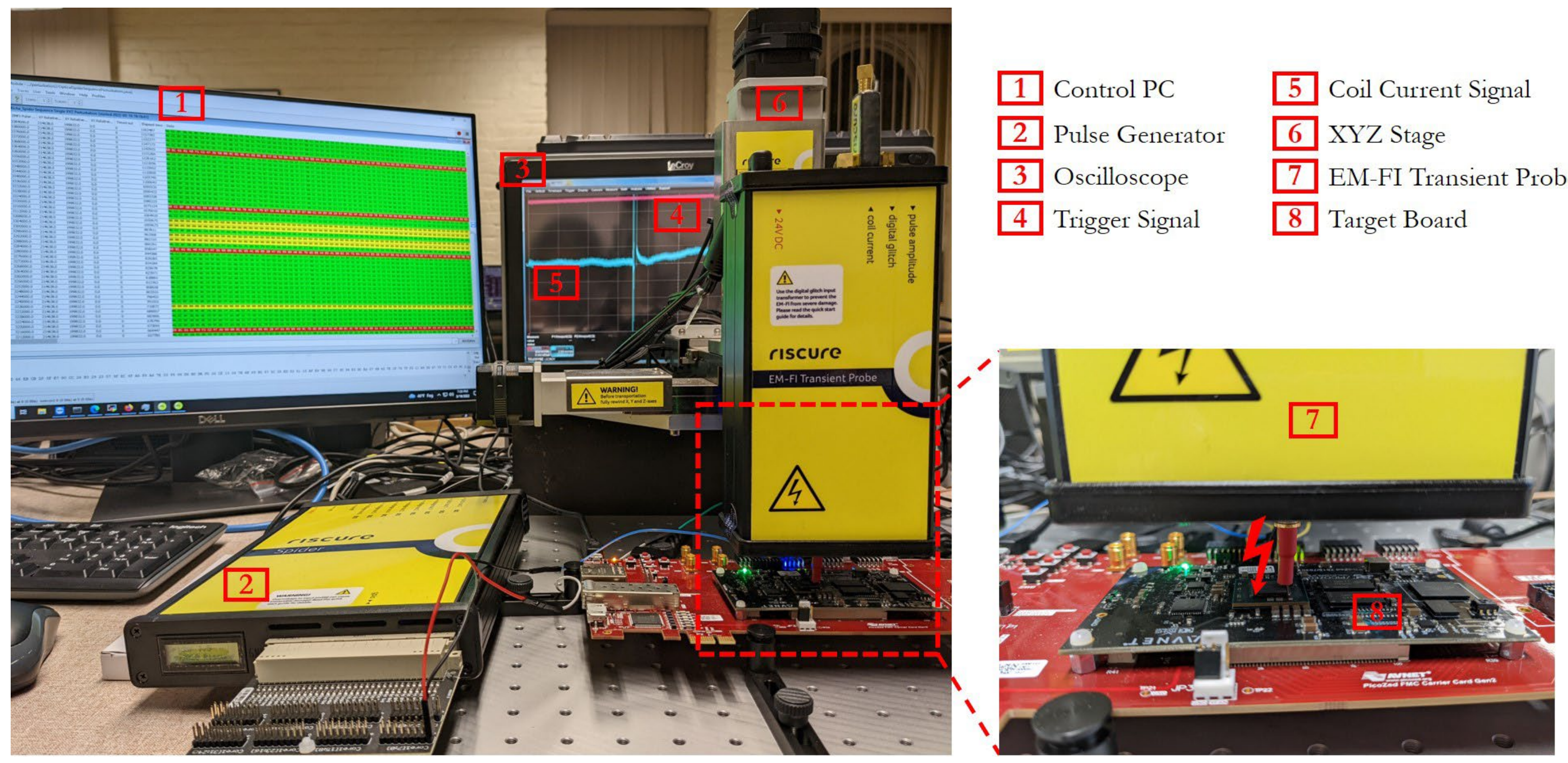
Experiment 2:

- Fixed at (214638, 199832) probe x-y position
- Random between 0 ns and 5M ns
- Random between 80% and 90%

Experiment 3:

- Fixed at (214638, 199832) probe x-y position
- Sweep between 3M ns and 3.8M ns by steps of 4000 ns
- Sweep between 83% and 90% by steps of 1%

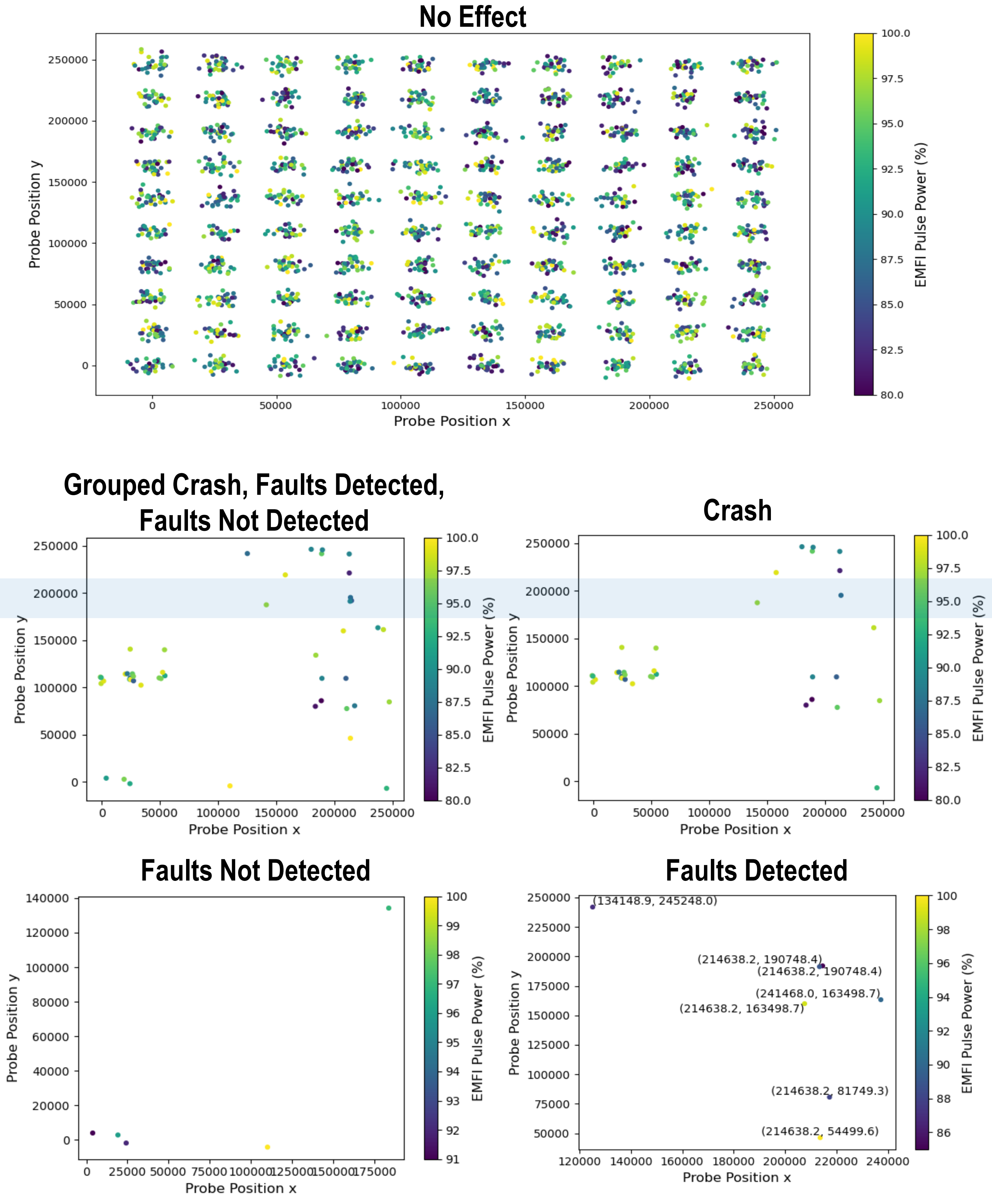
Our Experimental EMFI Setup



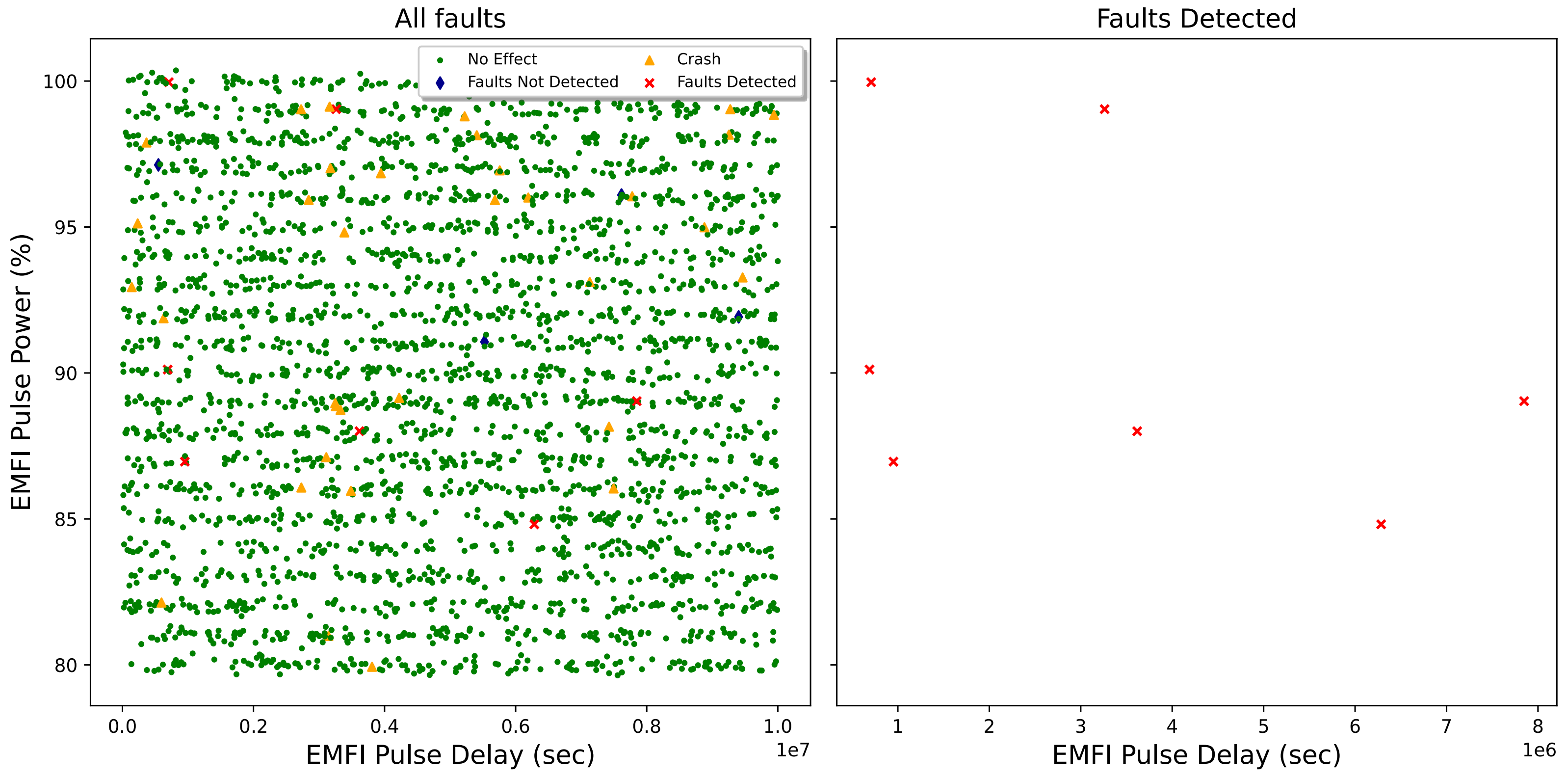
1 Control PC
2 Pulse Generator
3 Oscilloscope
4 Trigger Signal
5 Coil Current Signal
6 XYZ Stage
7 EM-FI Transient Probe
8 Target Board

Results – Experiment 1

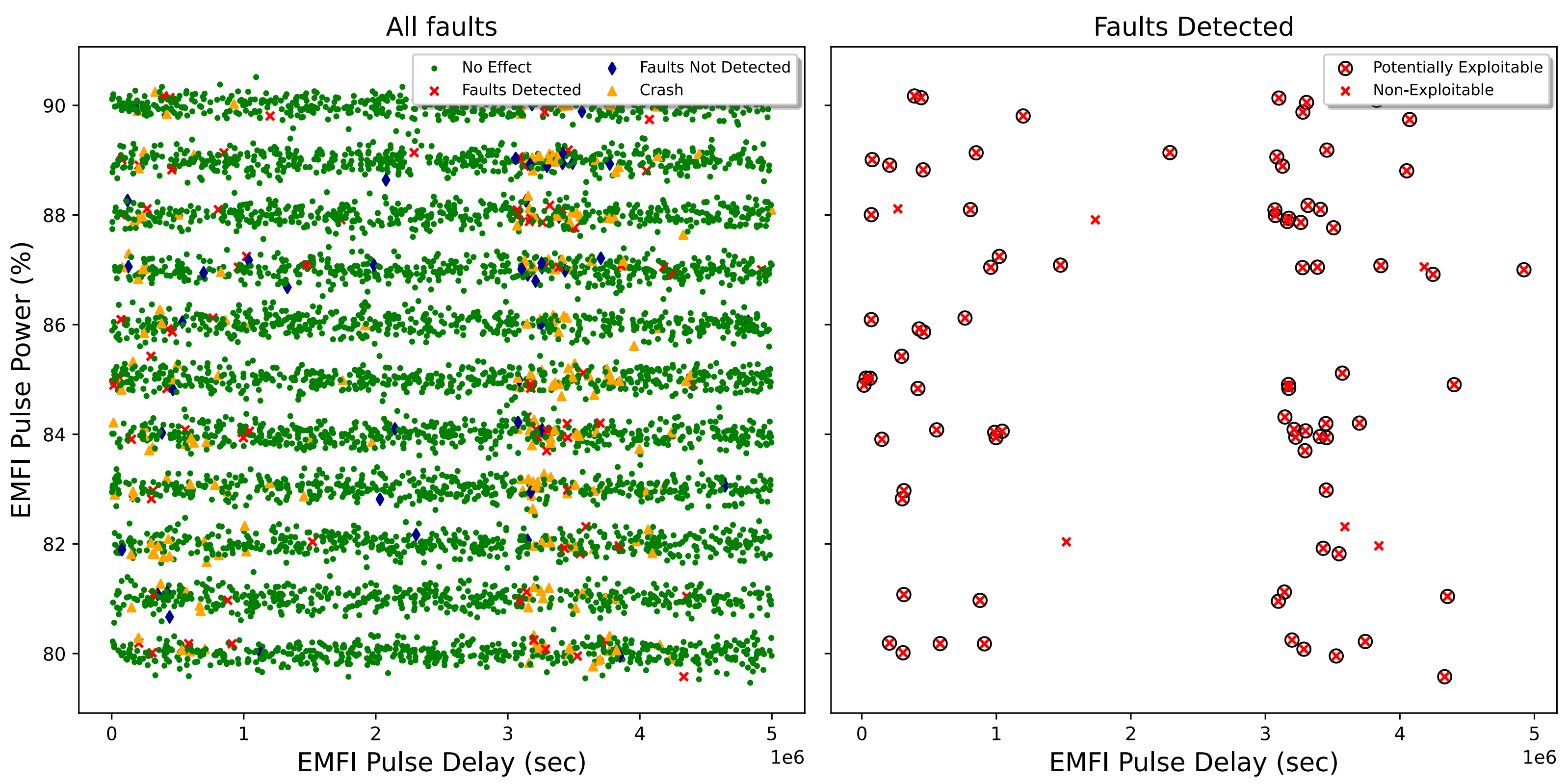
Probe positions over the chip leading to different fault categories



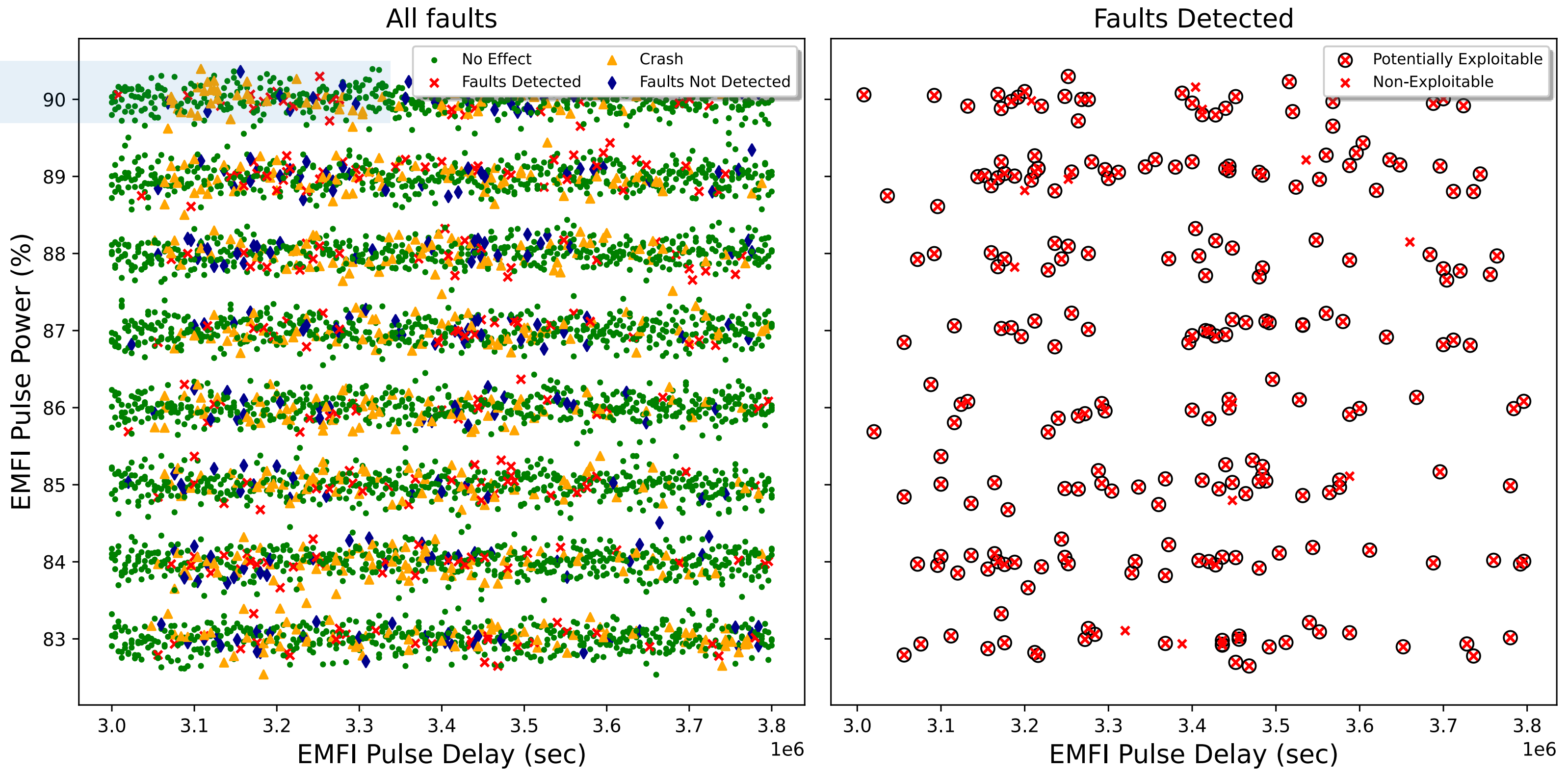
Relation between EMFI Pulse Delay and EMFI Pulse Power



Results – Experiment 2



Results – Experiment 3



Percentage Occurrence of different fault categories

Classification	Amount	Percentage (%)	Potentially Exploitable	Non-Exploitable
No Effect	3762	77.9851	0	3762
Crash	583	12.0854	0	583
Faults Not Detected	230	4.76783	146	84
Faults Detected	249	5.16169	236	13

Conclusion

- Data faults are fully detected by our countermeasure design.
- Estimated Potentially Exploitable Data Faults = 62%, remaining are control and memory faults.

NSF Award 1931639: SaTC: Core: Finding and Mitigating Side-Channel Leakage in Embedded Architectures