Efficient attack-surface exploration for electromagnetic fault injection

Daniele Carta – STMicroelectronics, Italy
Vittorio Zaccaria – Politecnico di Milano, Italy
Gabriele Quagliarella – Nozomi Networks, Switzerland
Maria Chiara Molteni – Security Pattern, Italy

COSADE 2023

D. A. E. Carta and G. Quagliarella completed this work while at Security Pattern.

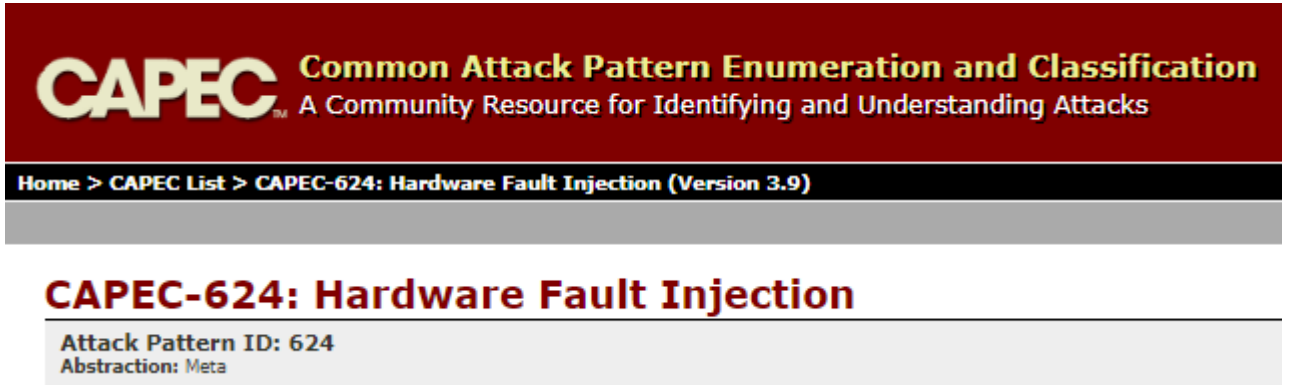
SECURITY PATTERN

POLITECNICO MILANO 1863

life.augmented

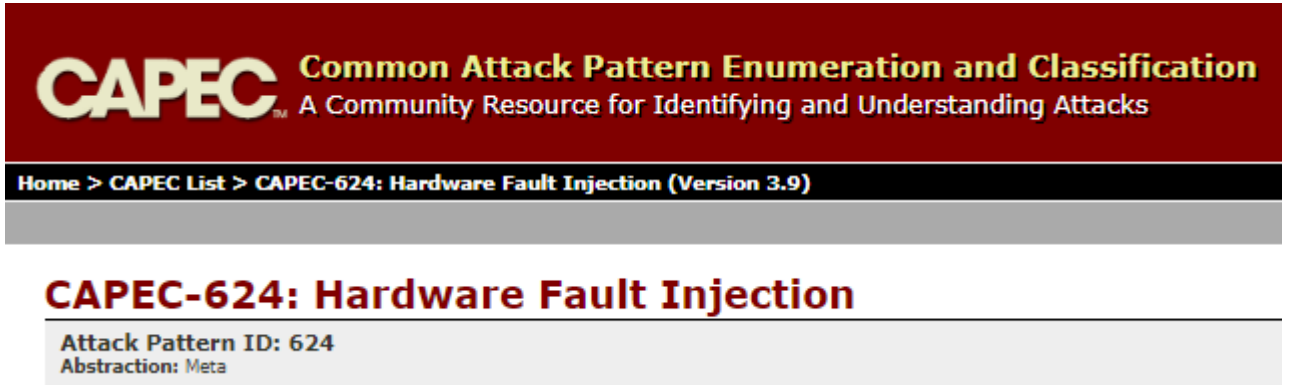
Fault injection

What it is



Fault injection

What it is

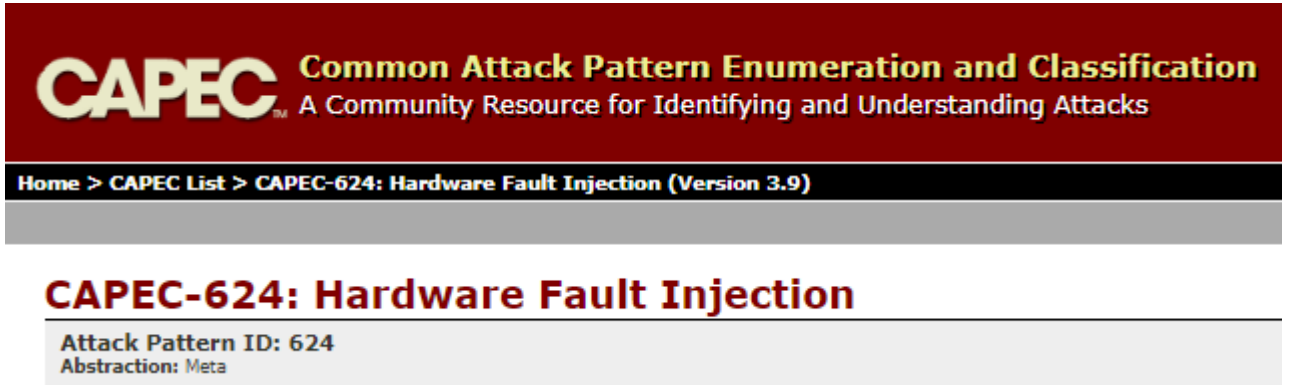


How it works

- Introducing "new" vulnerabilities
- Through nominal execution tampering
- And design assumption invalidation

Fault injection

What it is

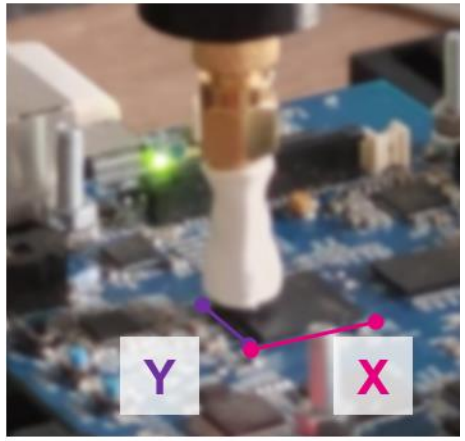


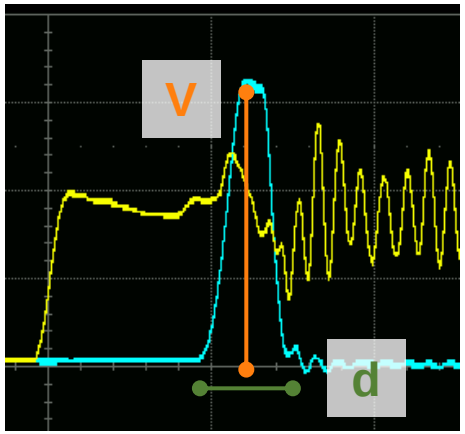
How it works

- Introducing "new" vulnerabilities
- Through nominal execution tampering
- And design assumption invalidation

Its main **cons**

- Requires physical access
- Difficult to execute
- Stochastic nature related to jitter and inaccuracy

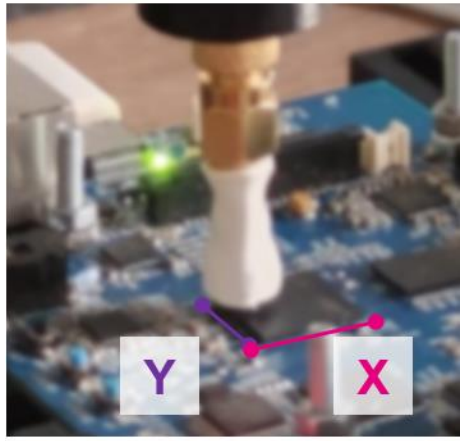


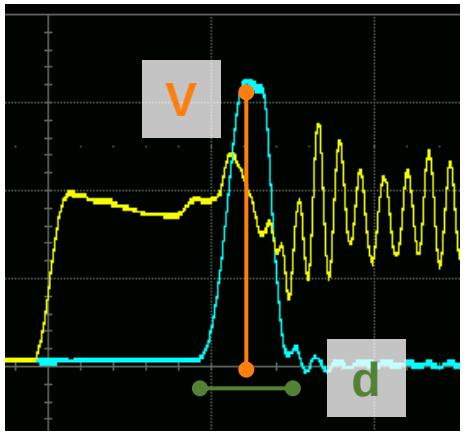


Electromagnetic fault injection

Injecting electromagnetic pulses tuning different parameters

- Location X, Y
- Intensity V
- Duration d
- Offset
- Angle
- Waveform
- Height
- Probe tip



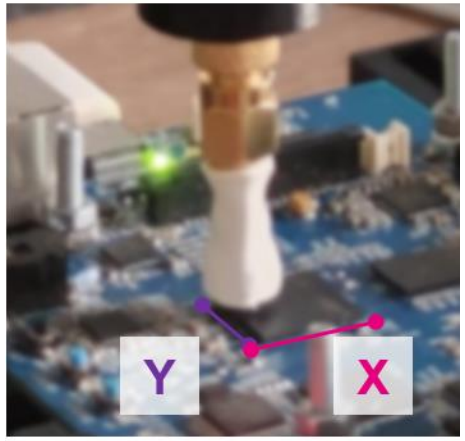


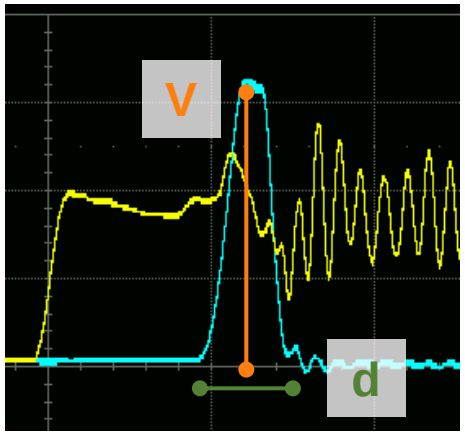
Electromagnetic fault injection

Injecting electromagnetic pulses tuning different parameters

- Location X, Y
- Intensity V
- Duration d
- Offset
- Angle
- Waveform
- Height
- Probe tip

"Good" configuration



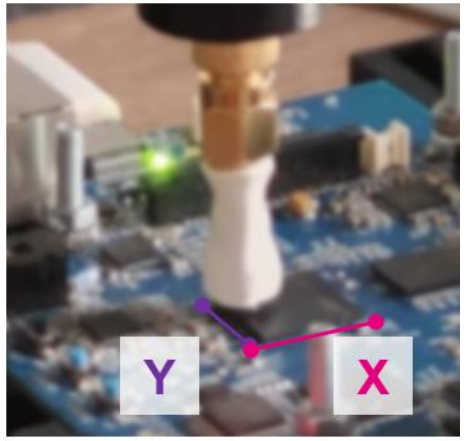


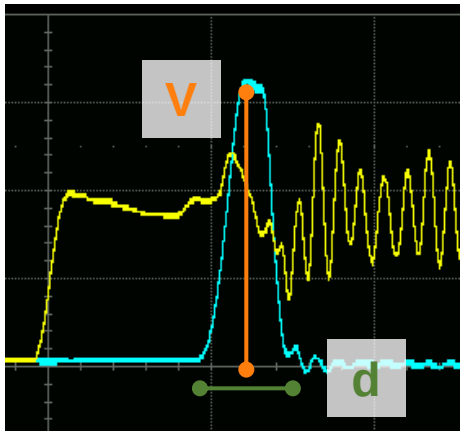
Electromagnetic fault injection

Injecting electromagnetic pulses tuning different parameters

- Location X, Y
- Intensity V
- Duration d
- Offset
- Angle
- Waveform
- Height
- Probe tip

"Good" configuration → **Attack success** ✓





Electromagnetic fault injection

Injecting electromagnetic pulses tuning different parameters

- Location X, Y
- Intensity V
- Duration d
- Offset
- Angle
- Waveform
- Height
- Probe tip

"Good" configuration ← **Attack success** ?

Motivation

"Determine your target's basic performance parameters… to provide some ballpark figures to **start finding effective faults**.

This is where **fault injection turns from science into a bit of art**. It now boils down to **tuning the fault injector parameters until they become effective**".

-Hardware Hacking Handbook, Jasper van Woudenberg and Colin o'Flynn

Motivation

Tune the parameters
Explore the search space

Naive search methods

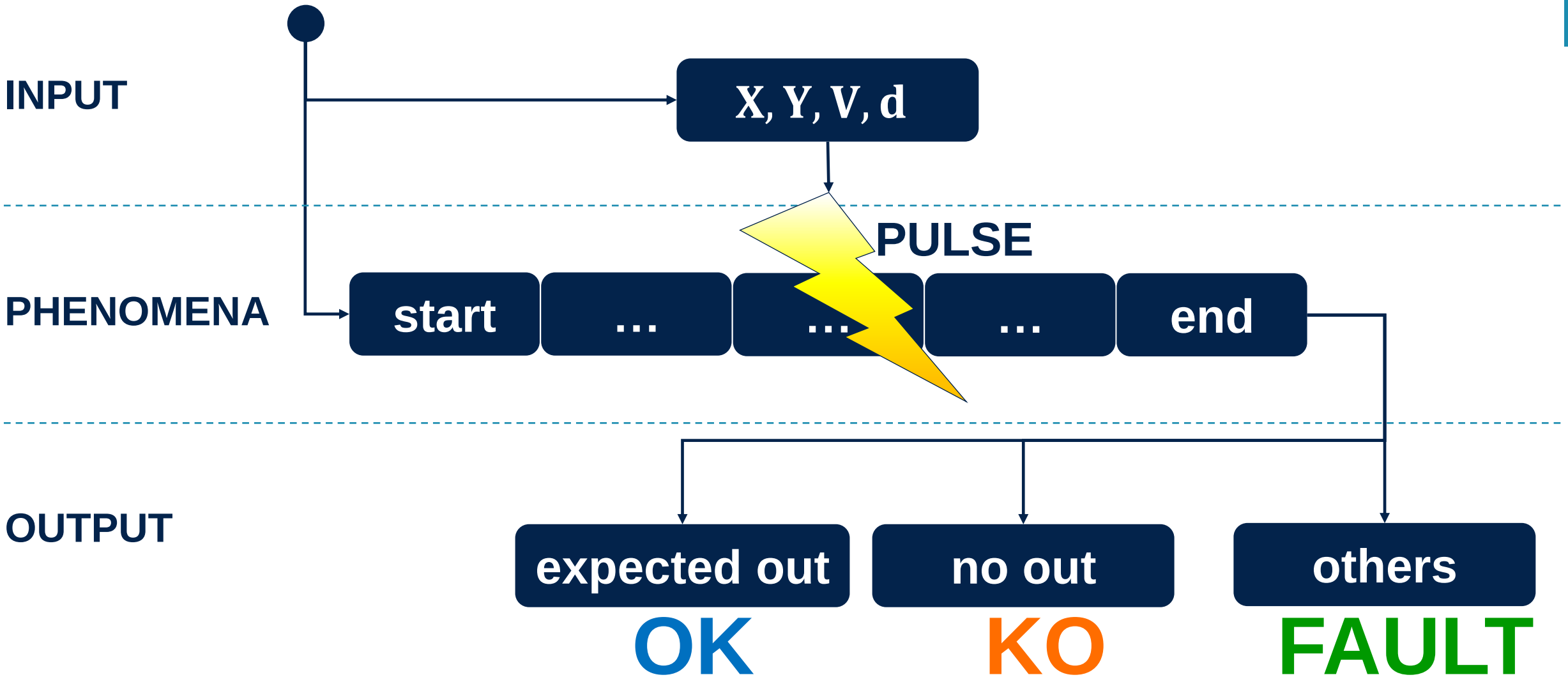
Exhaustive search	Random search
Unfeasible (case study: 57 years)	Sub-optimal (comparison)
	Moving the probe often introduces errors

Motivation

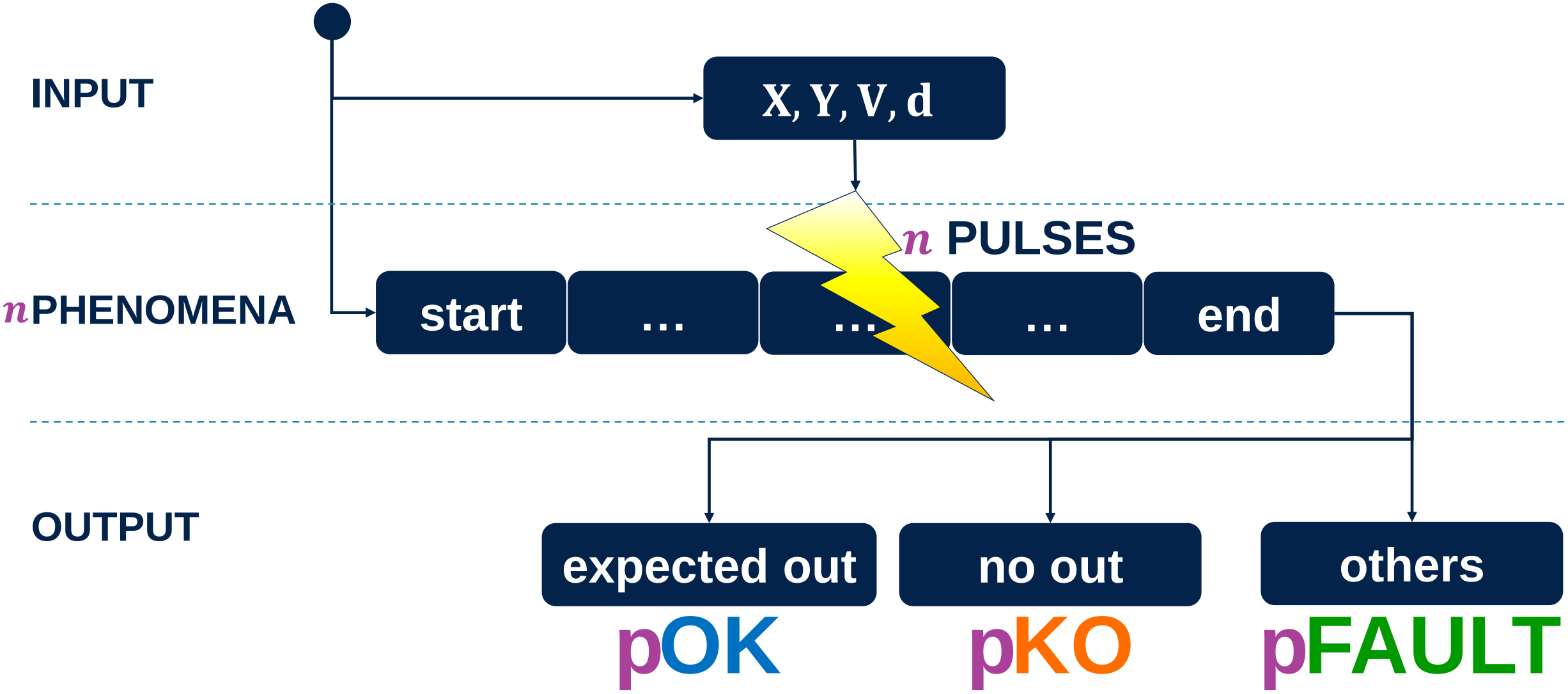
Previous research methods

State of the art	Limitations	Our contribution
“A methodology to localise EMFI areas on Microcontrollers.” – Madau	Validated at **fixed values for intensity and duration**	**4 variables** search (X,Y,V,d)
“Optimizing electromagnetic fault injection with genetic algorithms” – Maldini et al.	Fixed d, **maximizes faults occurrence**	Faults differentiation: **Rare faults might be useful**
“Electromagnetic fault injection as a new forensic approach for socs.” – Gaine et al.	Assumes sensitivity = susceptibility	No single best spot
Generic trial and error approaches	“Hope” that something happens	Use a method and be efficient

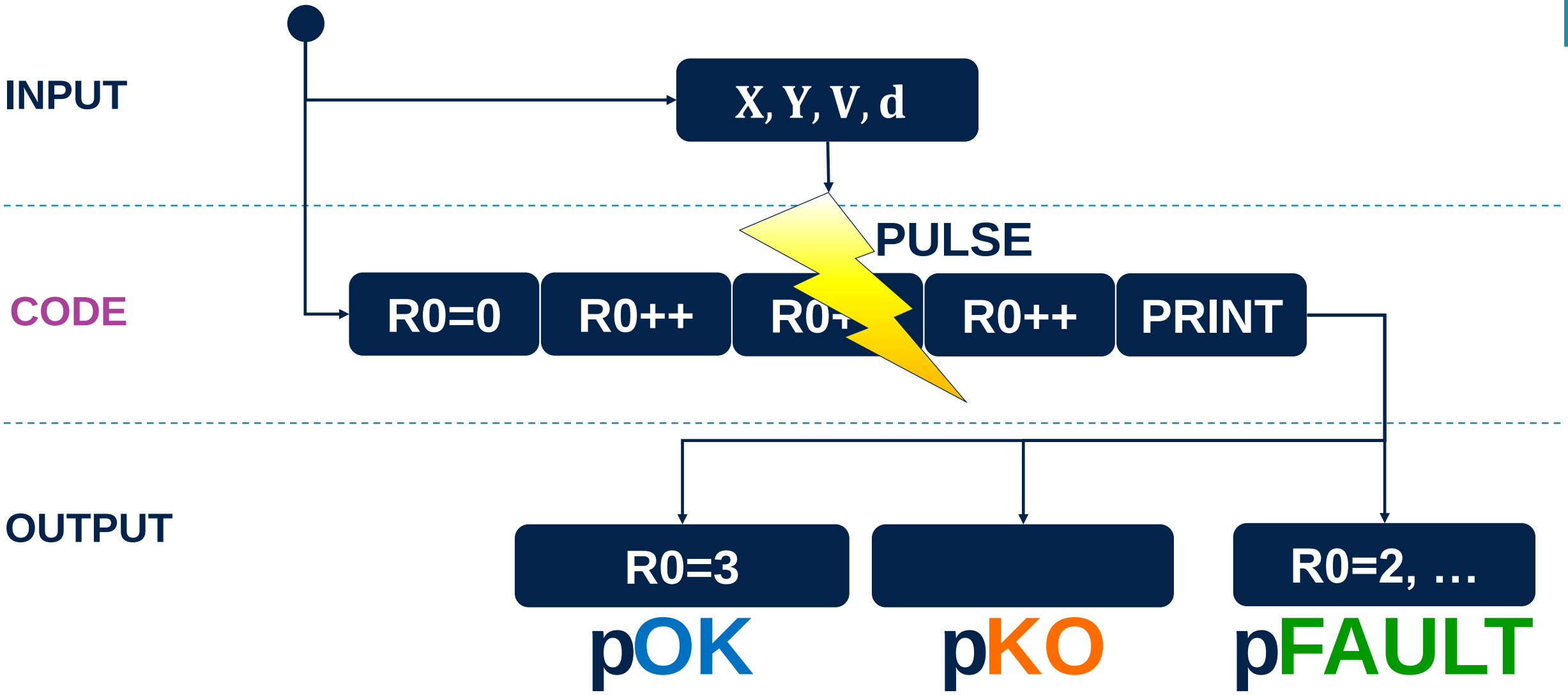
Evaluation model



Evaluation model



Evaluation model

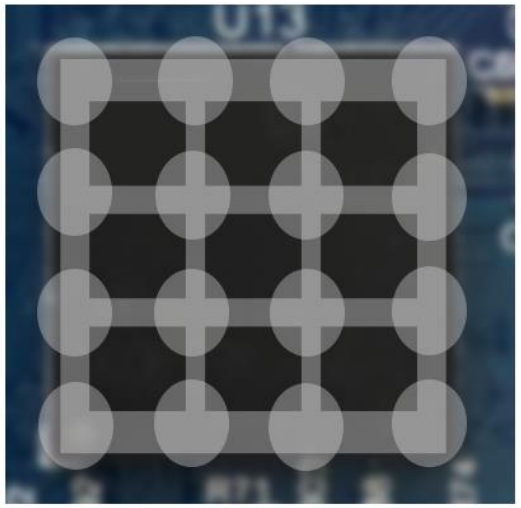


Surface Search – Tuning of X,Y

1-Define

Upon tools limitations

- X, Y coordinates grid
- Z_{min} probe's height
- V_{max} pulse's intensity
- d_{max} pulse's duration

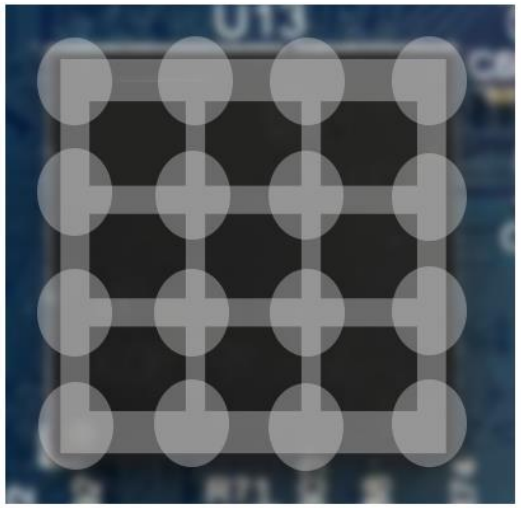


Surface Search – Tuning of X,Y

1-Define

Upon tools limitations

- X, Y coordinates grid
- $\mathrm{Z_{min}}$ probe's height
- $\mathrm{V_{max}}$ pulse's intensity
- $\mathrm{d_{max}}$ pulse's duration

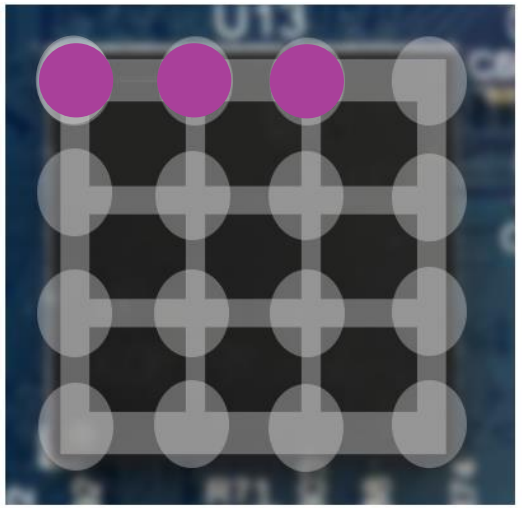


2-Evaluate

Fixed

$$\mathbf{EMFI}\left(\mathrm{X}, \mathrm{Y}, \bar{\mathrm{V}}_{\mathbf{max}}, \bar{\mathbf{d}}_{\mathbf{max}}\right) = \{\mathrm{P_{KO}}, \mathrm{P_{OK}}, \mathrm{P_{FAULT}}\}$$

On the **grids**' coordinates

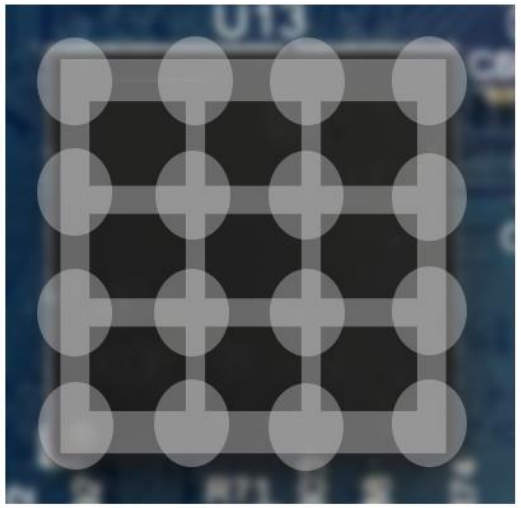


Surface Search – Tuning of X,Y

1-Define

Upon tools limitations

- X, Y coordinates grid
- Z_{min} probe's height
- V_{max} pulse's intensity
- d_{max} pulse's duration



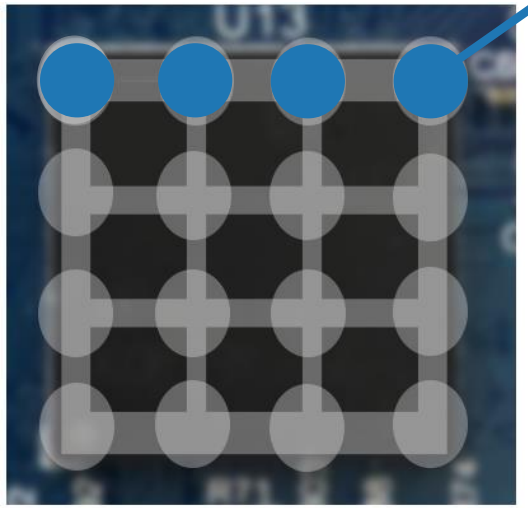
2-Evaluate

Fixed

$$\mathbf{EMFI}\left(X, Y, \bar{V}_{max}, \bar{d}_{max}\right) = \{P_{KO}, P_{OK}, P_{FAULT}\}$$

On the **grids**' coordinates

pOK = 100%

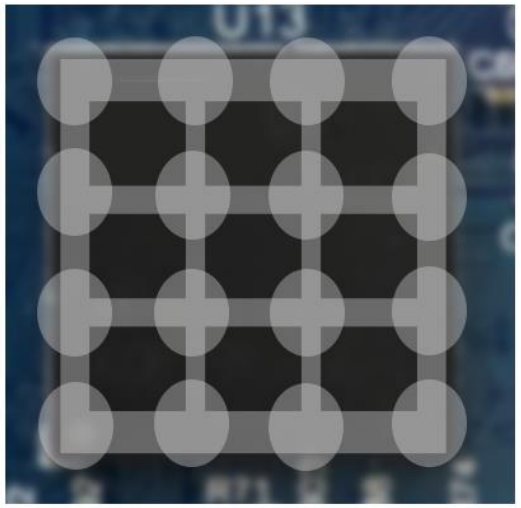


Surface Search – Tuning of X,Y

1-Define

Upon tools limitations

- X, Y coordinates grid
- Z_{min} probe's height
- V_{max} pulse's intensity
- d_{max} pulse's duration

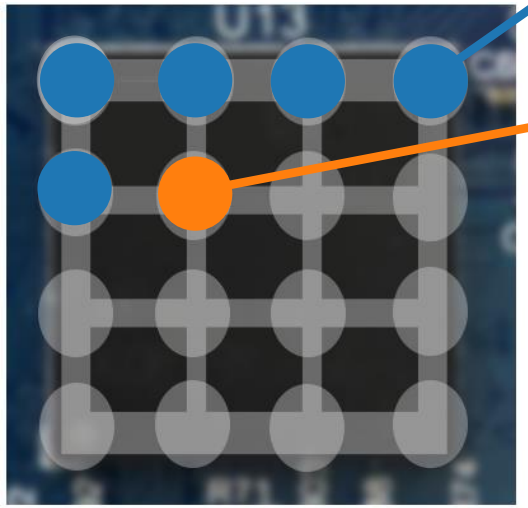


2-Evaluate

Fixed

$$\mathbf{EMFI}\left(X, Y, \bar{V}_{max}, \bar{d}_{max}\right) = \{P_{KO}, P_{OK}, P_{FAULT}\}$$

On the **grids**' coordinates



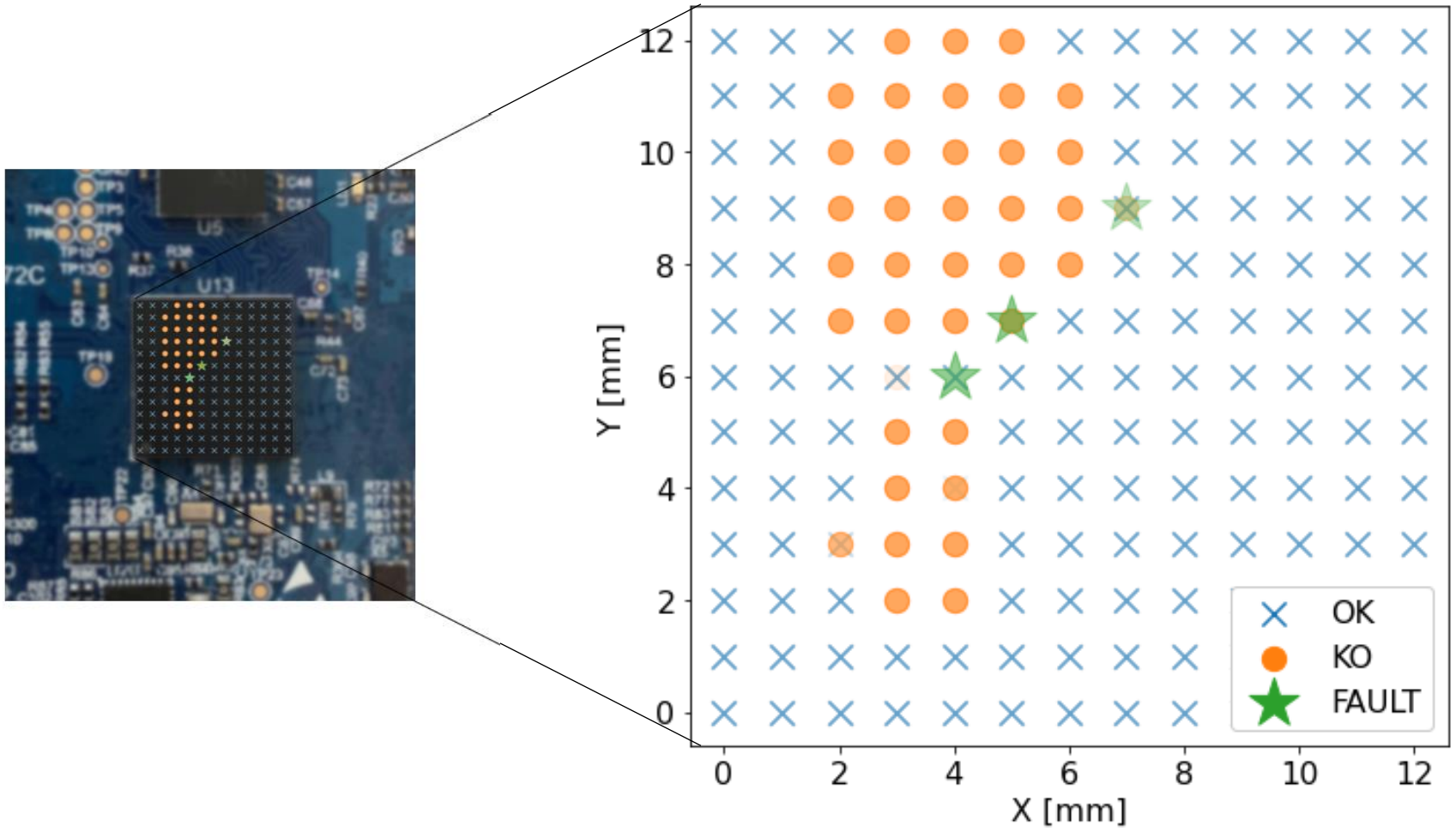
pOK = 100%

KO or **FAULT** ⟷ pOK < 100%

Susceptibility criterion

Can **FAULT**

Surface Search outcome example



Coordinate Search – Tuning of V,d

1-Define

Upon tools limitations

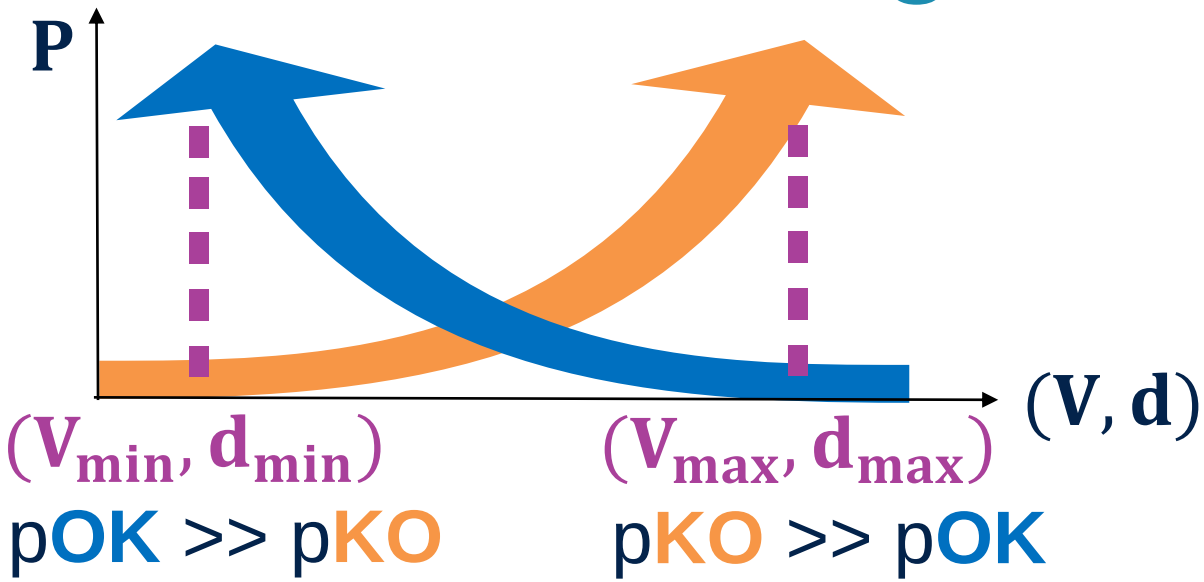
- $\mathbf{V_{min}}$ pulse's intensity
- $\mathbf{d_{min}}$ pulse's duration

Coordinate Search – Tuning of V,d

1-Define

Upon tools limitations

- $\mathbf{V_{min}}$ pulse's intensity
- $\mathbf{d_{min}}$ pulse's duration

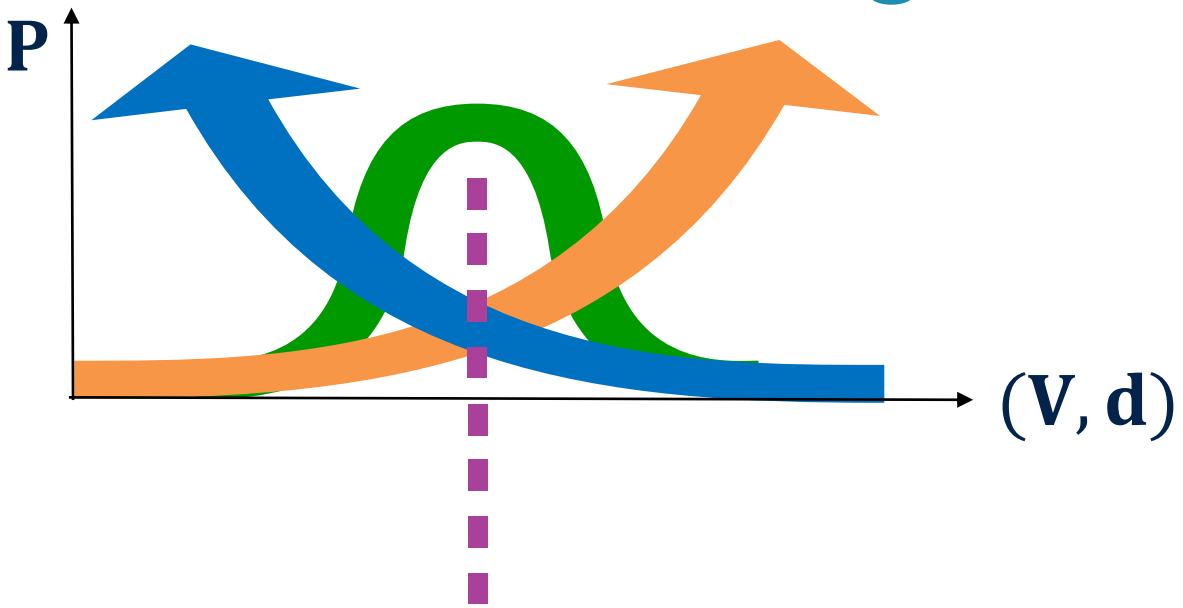


Coordinate Search – Tuning of V,d

1-Define

Upon tools limitations

- $\mathbf{V_{min}}$ pulse's intensity
- $\mathbf{d_{min}}$ pulse's duration



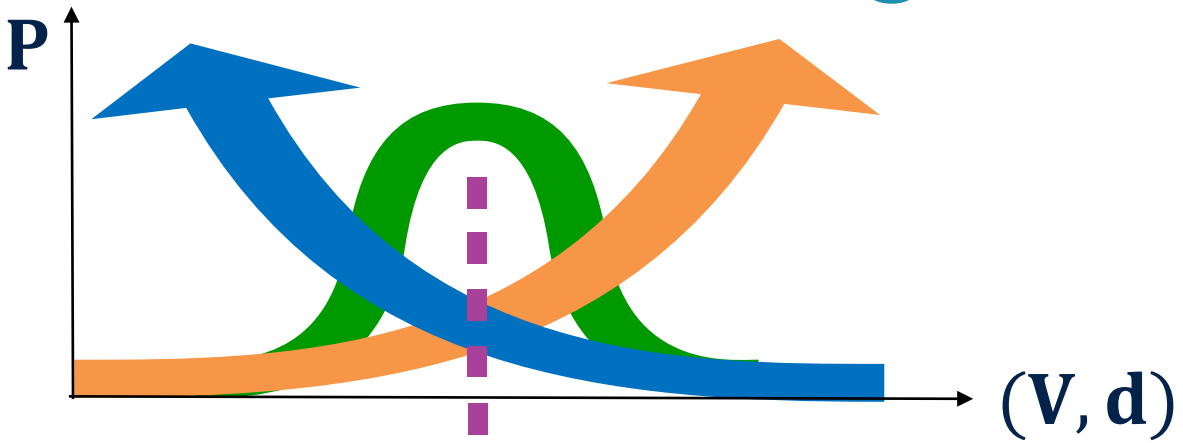
pKO ≈ pOK → Max pFAULT

Coordinate Search – Tuning of V,d

1-Define

Upon tools limitations

- $\mathbf{V_{min}}$ pulse's intensity
- $\mathbf{d_{min}}$ pulse's duration



pKO ≈ pOK → Max pFAULT

2-Search for the roots of the Equilibrium

$$E = P_{KO}(\overline{X}, \overline{Y}, V, d) - P_{OK}(\overline{X}, \overline{Y}, V, d) = 0$$

Using a bidimensional bisection algorithm per each coordinate

Equilibrium search – Iterations example

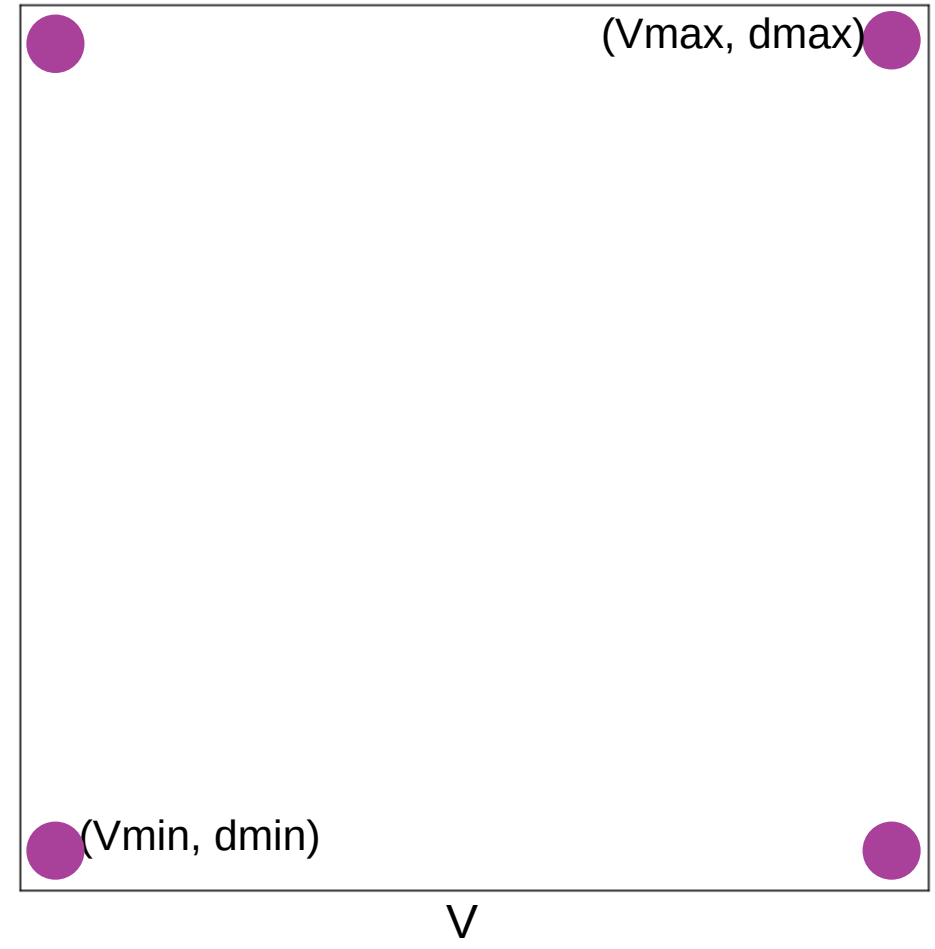
Iteration 0

0. Base rectangle

1. Evaluate configurations

2. Recognize bracketing rectangles

3. Refine them

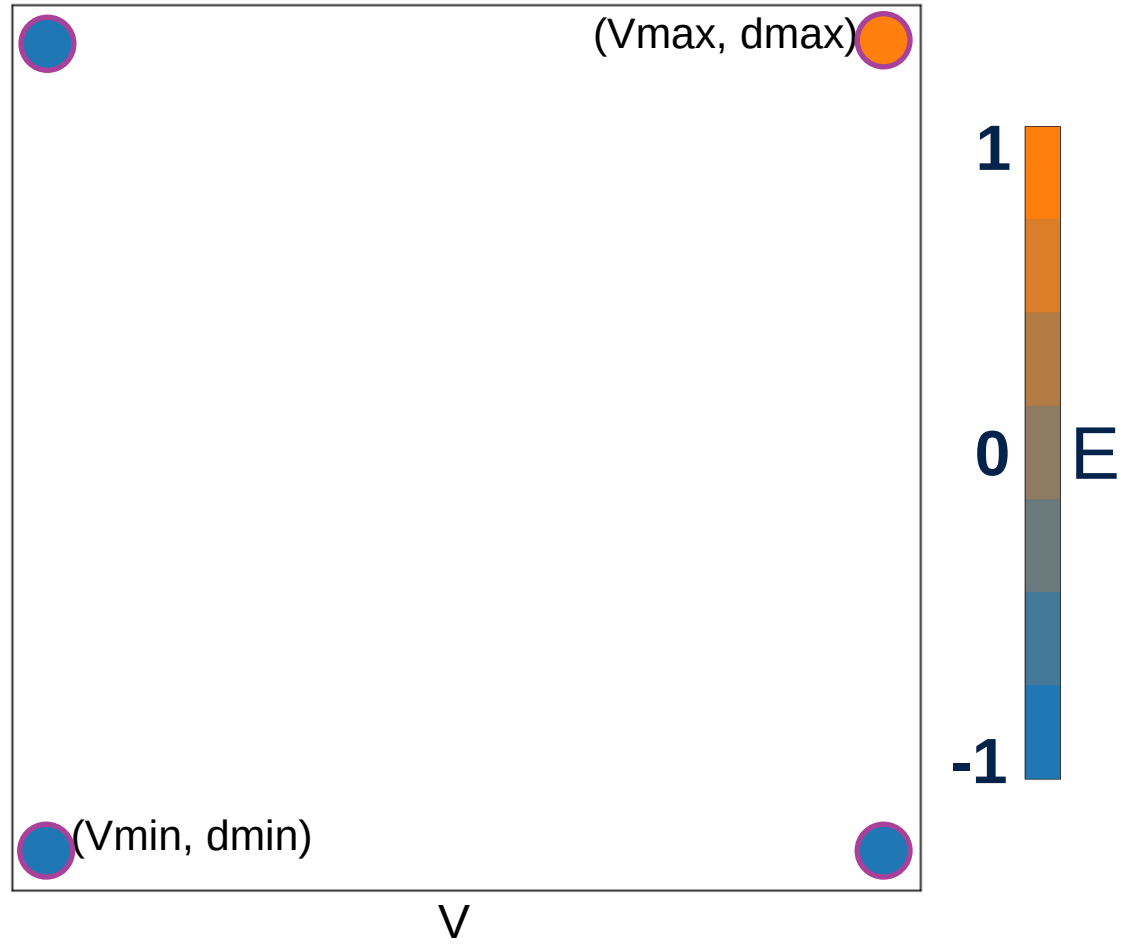


(Vmax,dmax),(Vmin,dmin) base vertexes

Equilibrium search – Iterations example

Iteration 0

0. Base rectangle
1. **Evaluate configurations**
2. Recognize bracketing rectangles
3. Refine them

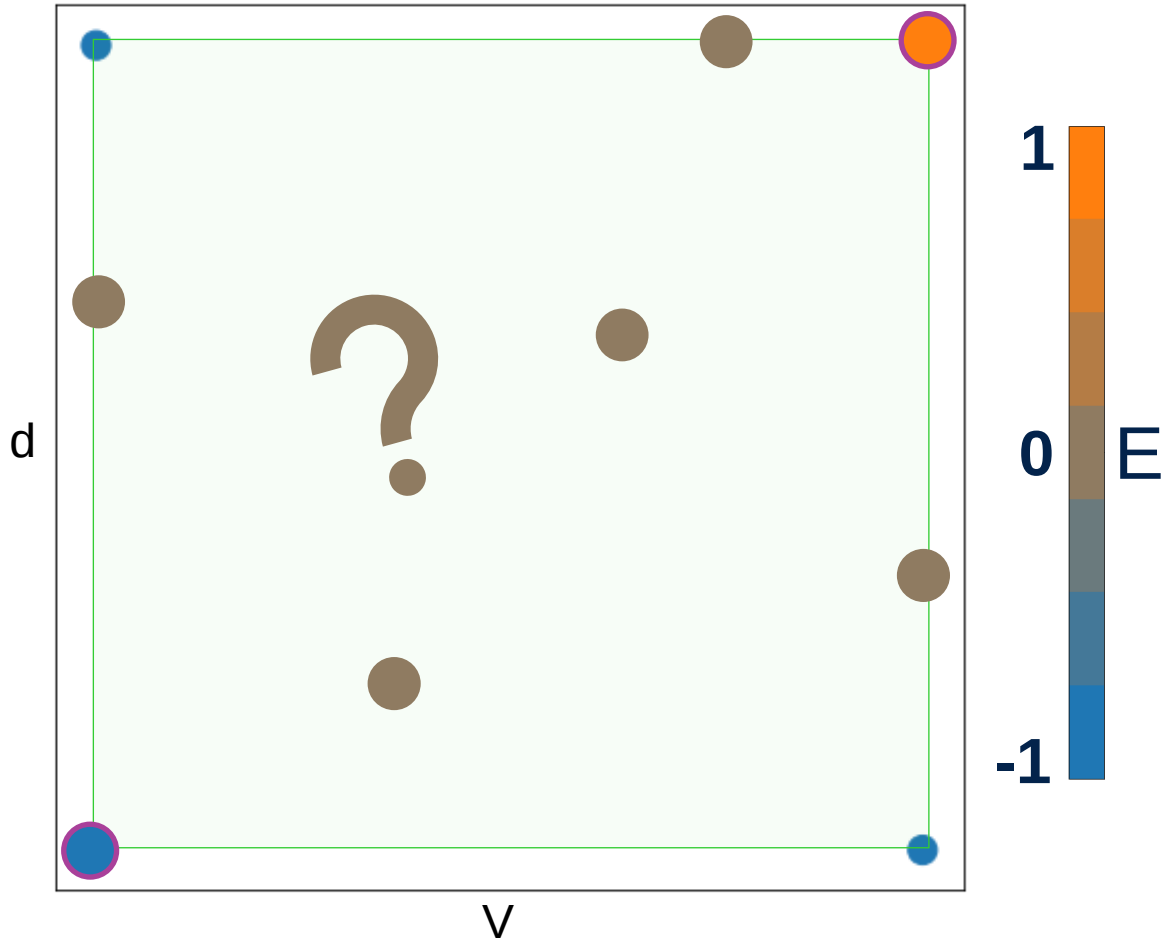


The roots of the function lie between the four vertexes

Equilibrium search – Iterations example

Iteration 0

0. Base rectangle
1. Evaluate configurations
2. Recognize bracketing rectangles
3. Refine them

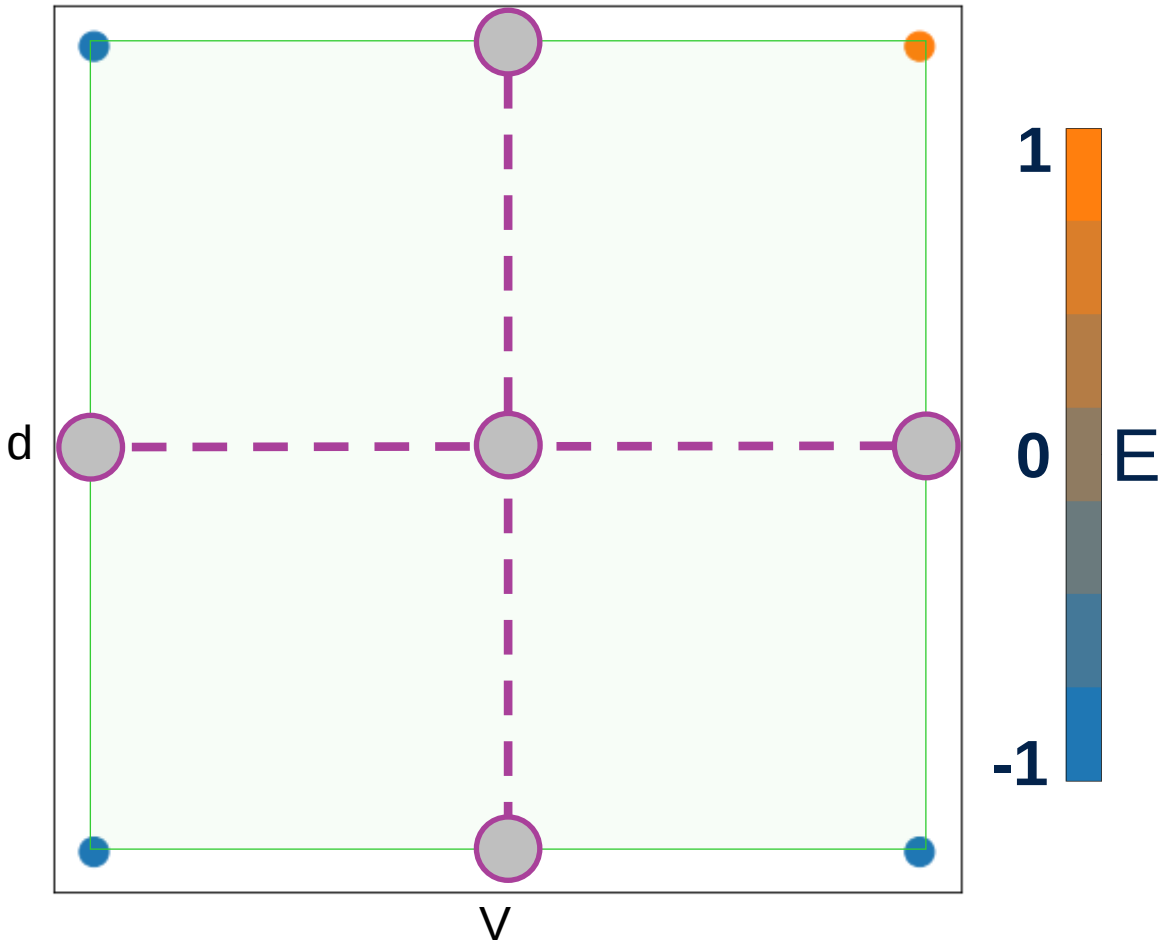


A bracketing rectangle has a vertex with E>0 and a vertex with E<0

Equilibrium search – Iterations example

Iteration 0

0. Base rectangle
1. Evaluate configurations
2. Recognize bracketing rectangles
3. **Refine them**

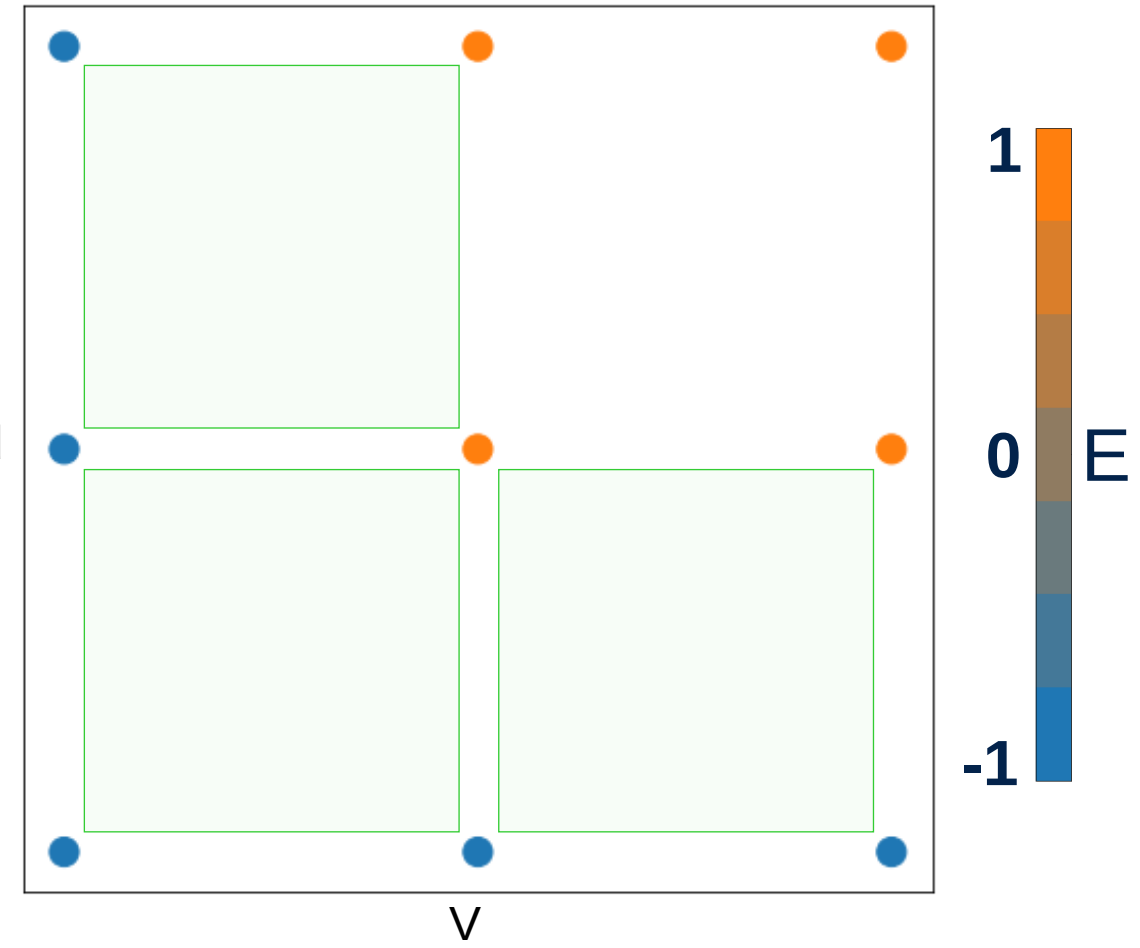


To refine a rectangle means to split it in 4 equal sub-rectangles

Equilibrium search – Iterations example

Iteration 1

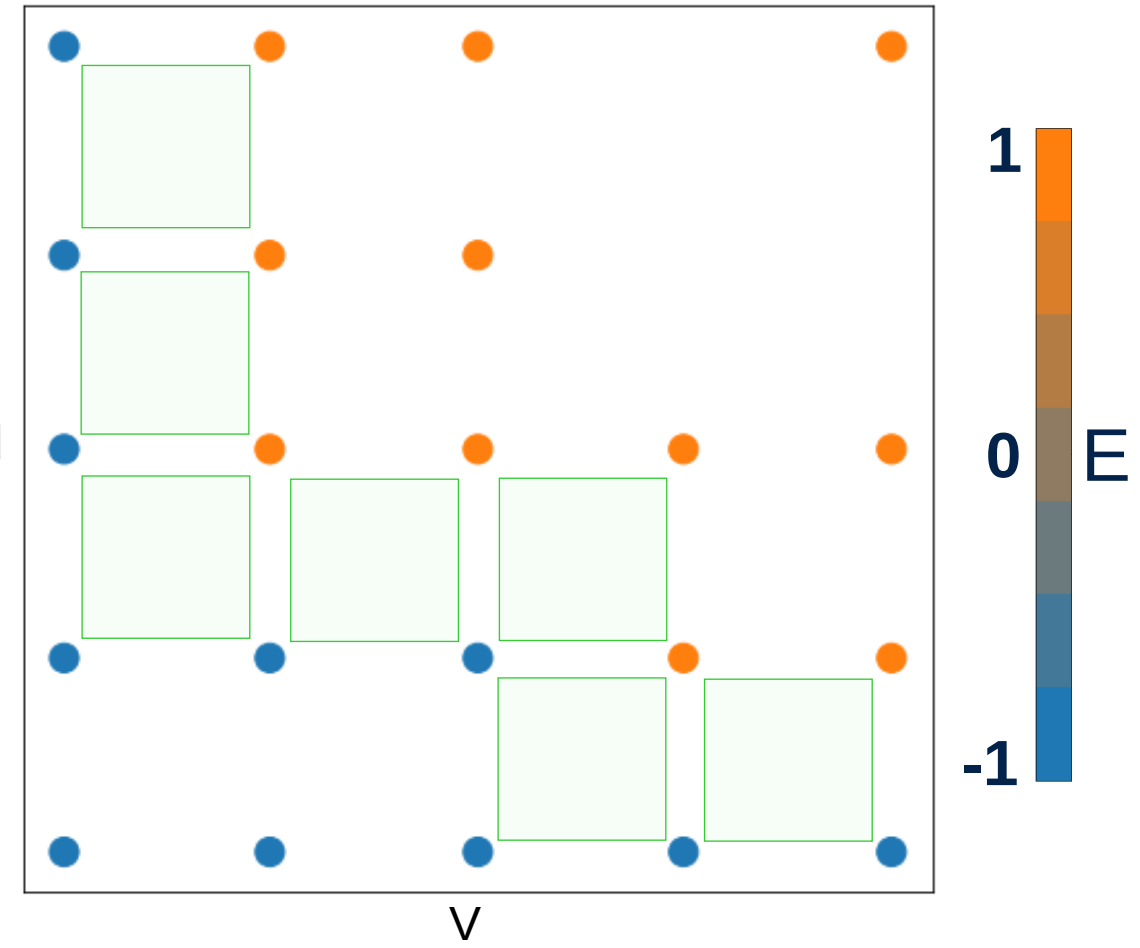
0. Base rectangle
1. **Evaluate configurations**
2. **Recognize bracketing rectangles**
3. **Refine them**



Equilibrium search – Iterations example

Iteration 2

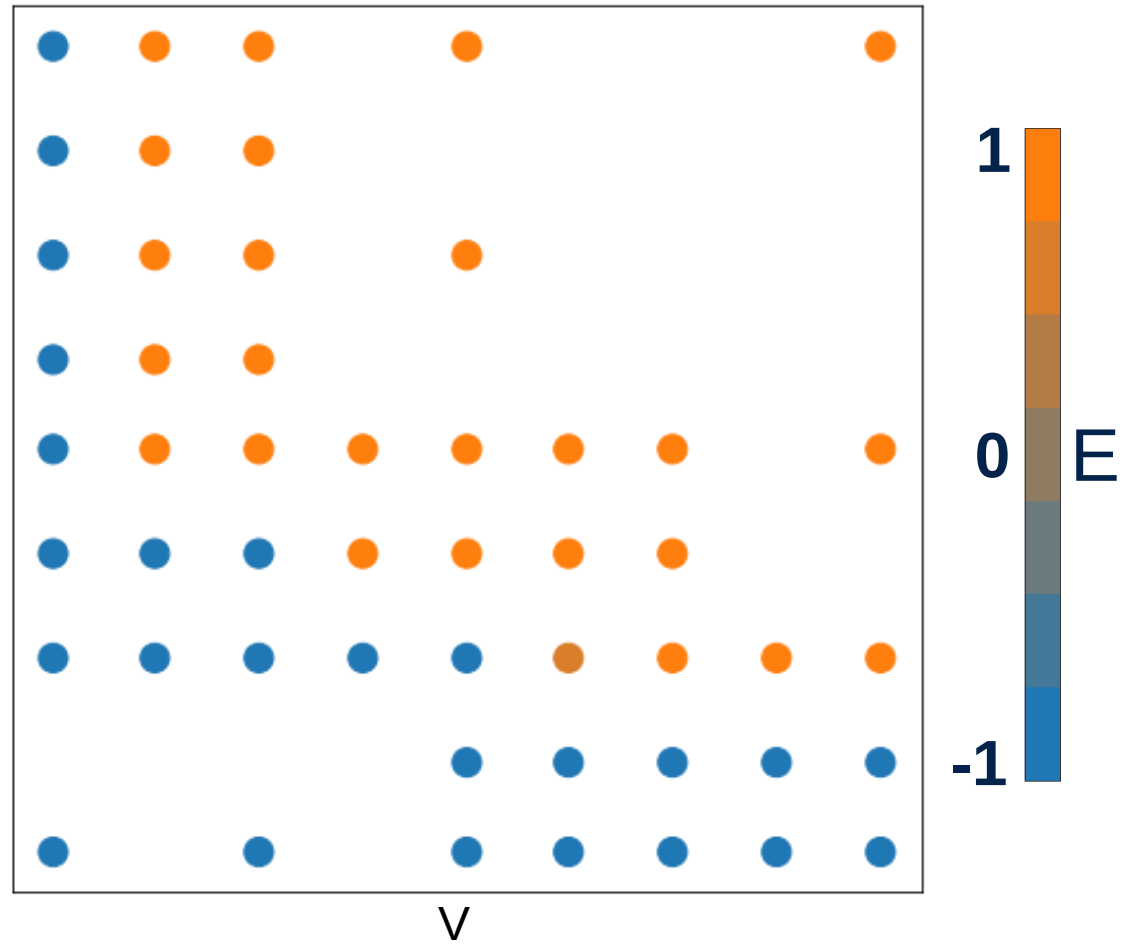
0. Base rectangle
1. **Evaluate configurations**
2. **Recognize bracketing rectangles**
3. **Refine them**



Equilibrium search – Iterations example

Iteration 3

0. Base rectangle
1. **Evaluate configurations**
2. **Recognize bracketing rectangles**
3. **Refine them**

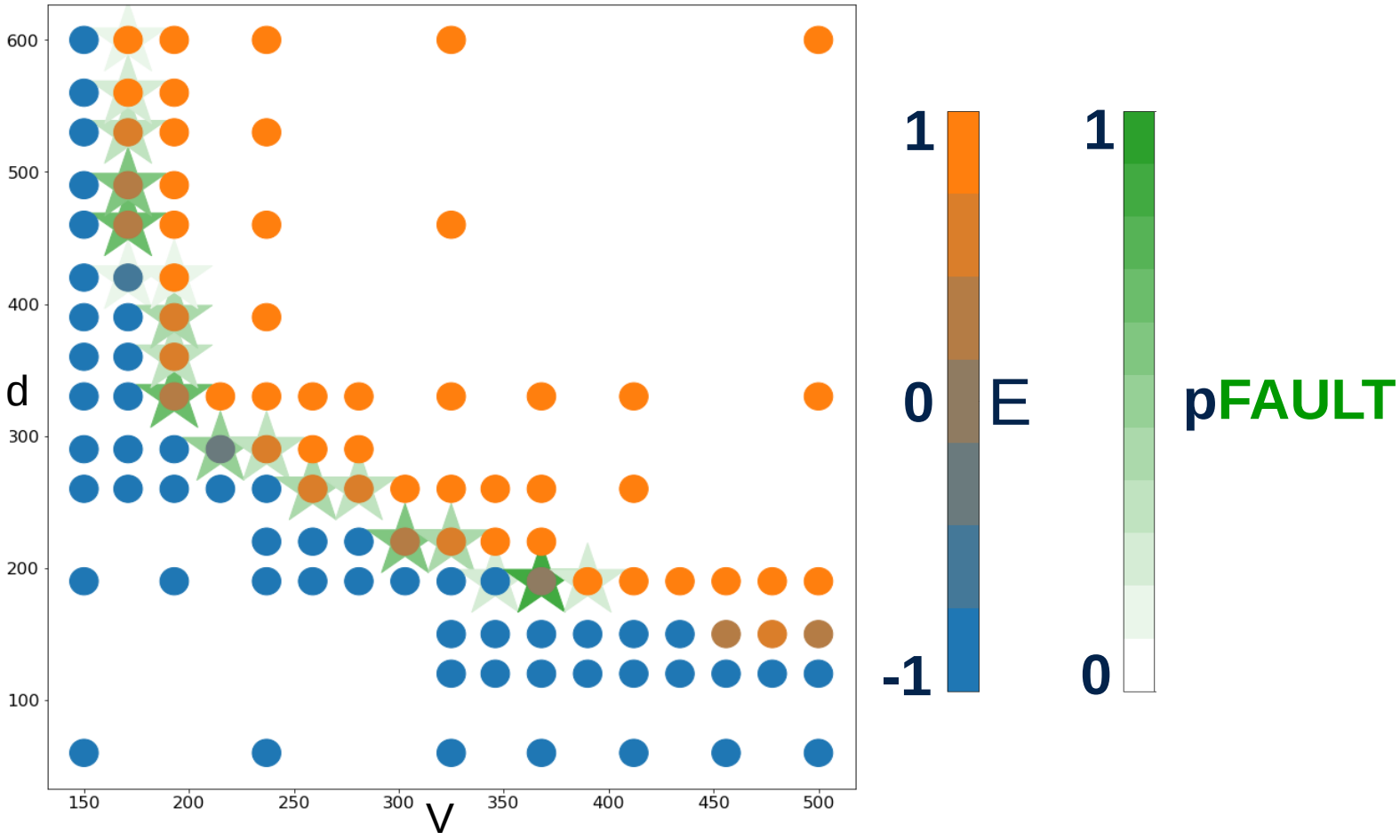


Iterations stop due to tools limitations or imposed threshold

Equilibrium search – Iterations example

Iteration 4

0. Base rectangle
1. **Evaluate configurations**
2. **Recognize bracketing rectangles**
3. **Refine them**

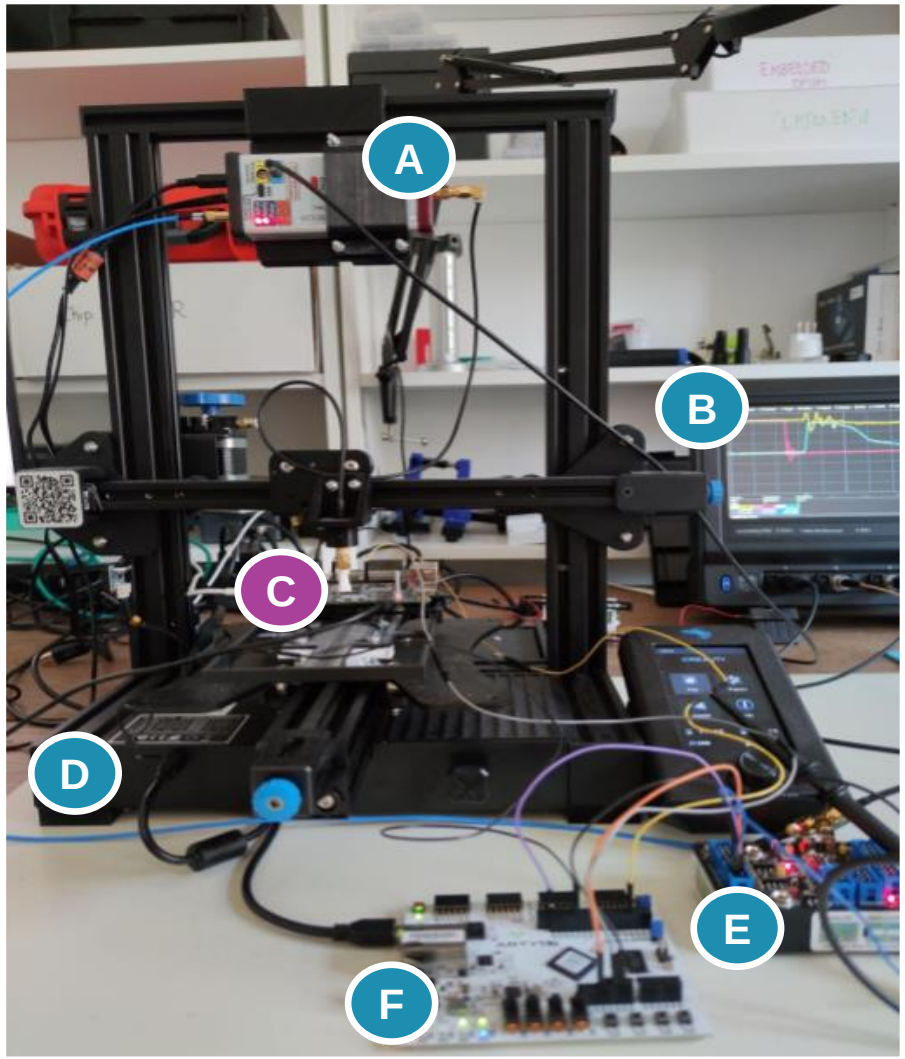


Faults seem located for the most part where $\mathrm{E} \cong 0$

Experimental Setup

C Target

- ARMv7 dual core, dual issue SoC
- Cortex A7 600 MHz
- Eight pipeline stages
- Data and instruction caches disabled
- No speculative execution



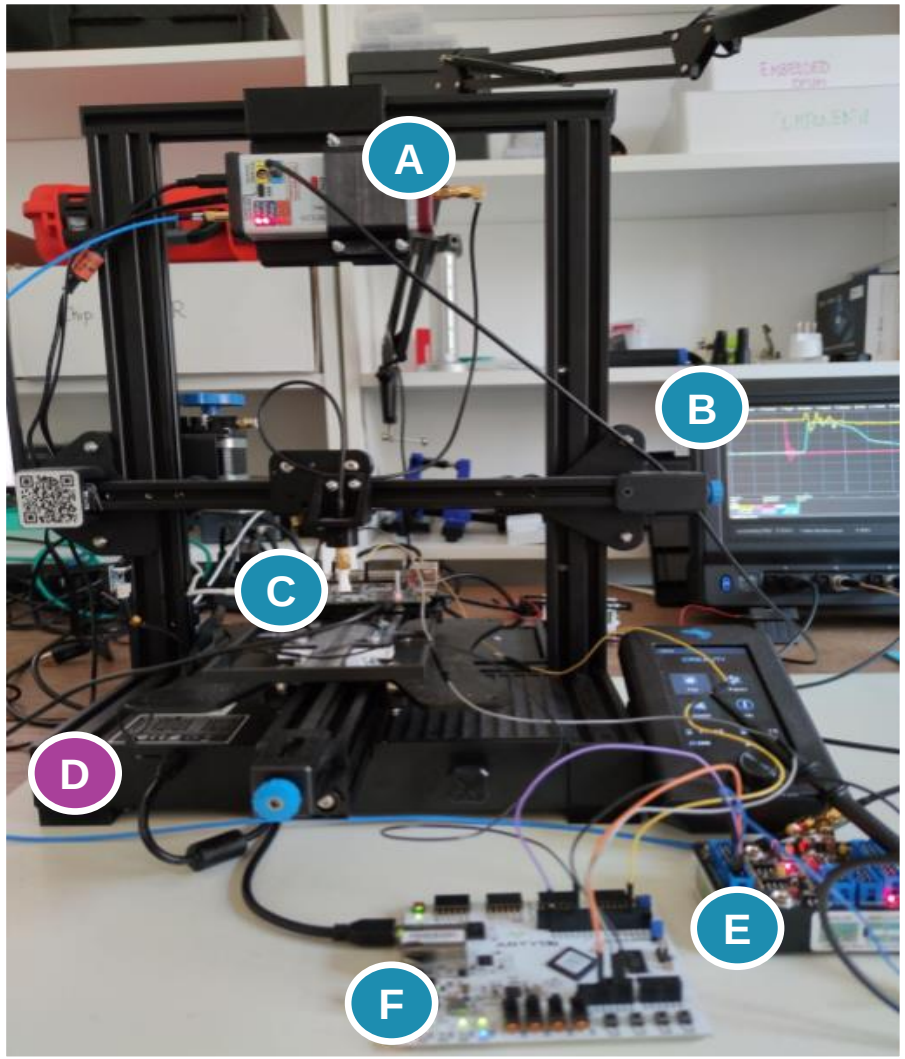
Experimental Setup

C Target

- ARMv7 dual core, dual issue SoC
- Cortex A7 600 MHz
- Eight pipeline stages
- Data and instruction caches disabled
- No speculative execution

D Location controller

- 3D printer
- X, Y steps = 0.1 mm



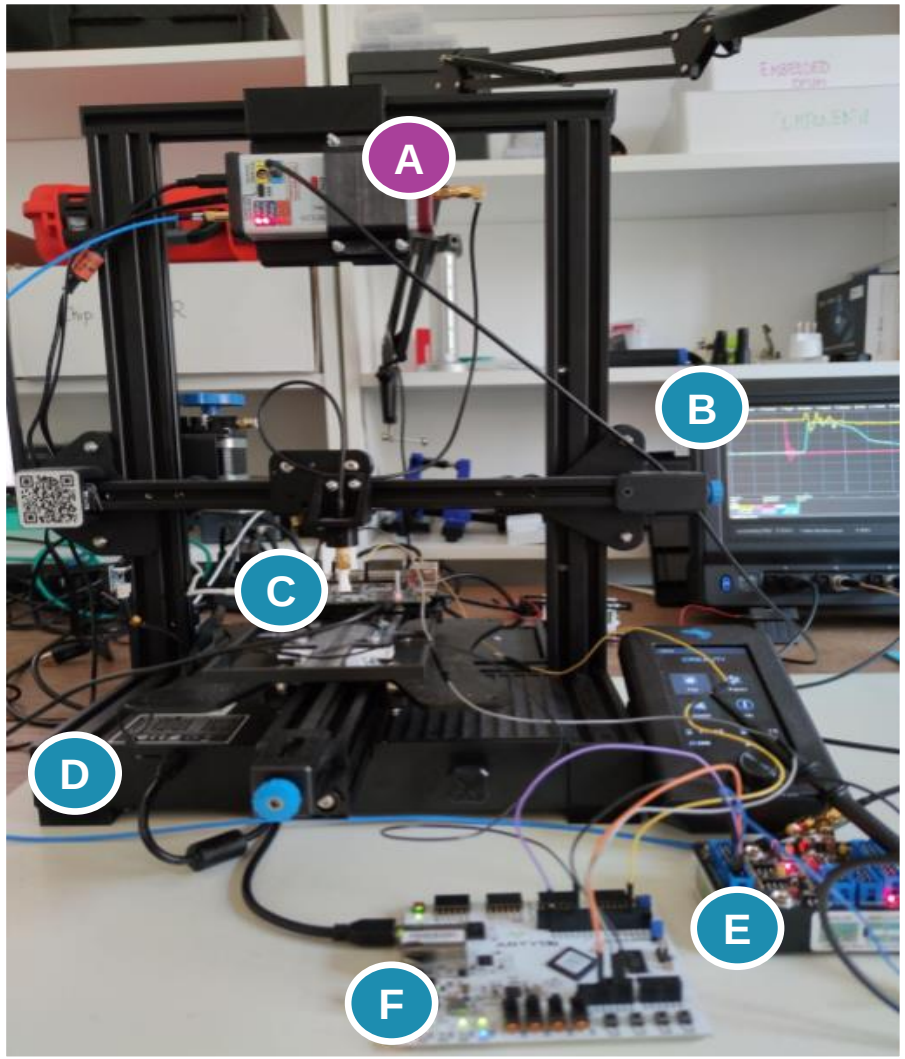
Experimental Setup

C Target

- ARMv7 dual core, dual issue SoC
- Cortex A7 600 MHz
- Eight pipeline stages
- Data and instruction caches disabled
- No speculative execution

D Location controller

- 3D printer
- X, Y steps = 0.1 mm



A Fault injector

- NewAE ChipShouter
- V_{min} = 150 V
- V_{max} = 500 V
- V steps = 1 V
- d_{min}= 60 ns
- d_{max} = 600 ns

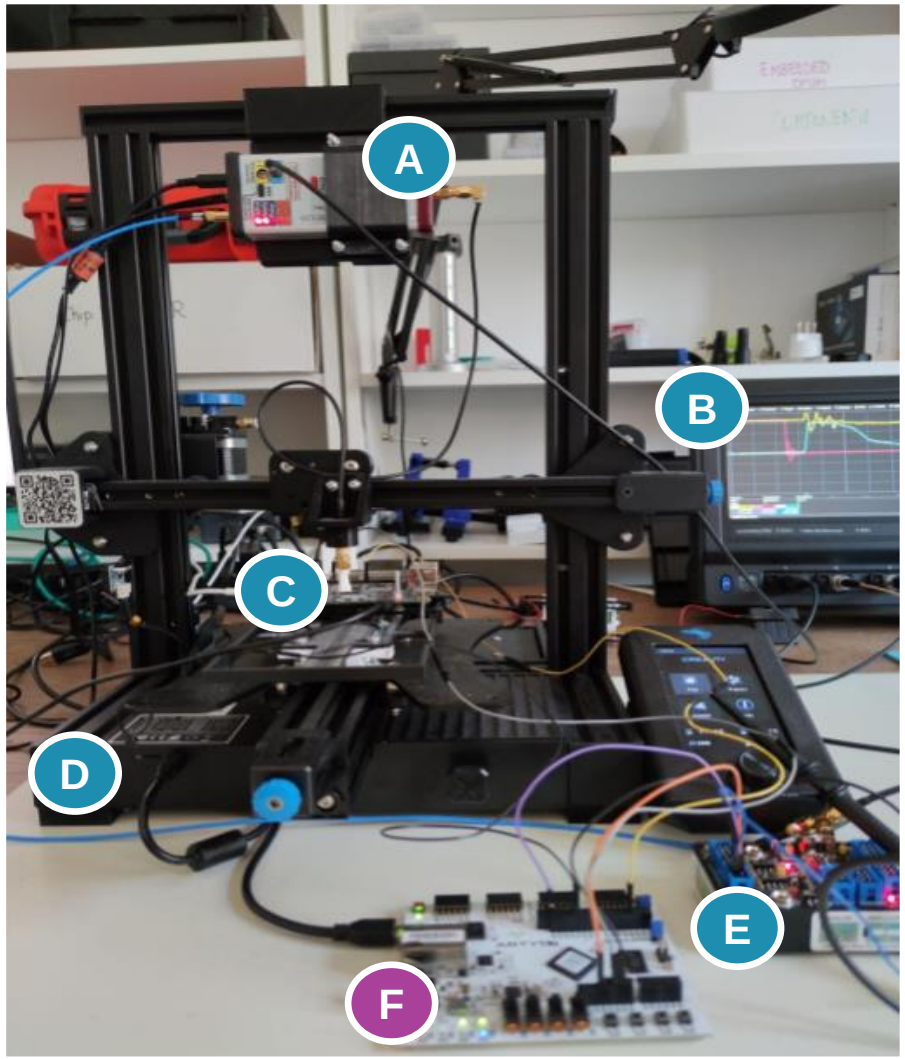
Experimental Setup

C Target

- ARMv7 dual core, dual issue SoC
- Cortex A7 600 MHz
- Eight pipeline stages
- Data and instruction caches disabled
- No speculative execution

D Location controller

- 3D printer
- X, Y steps = 0.1 mm



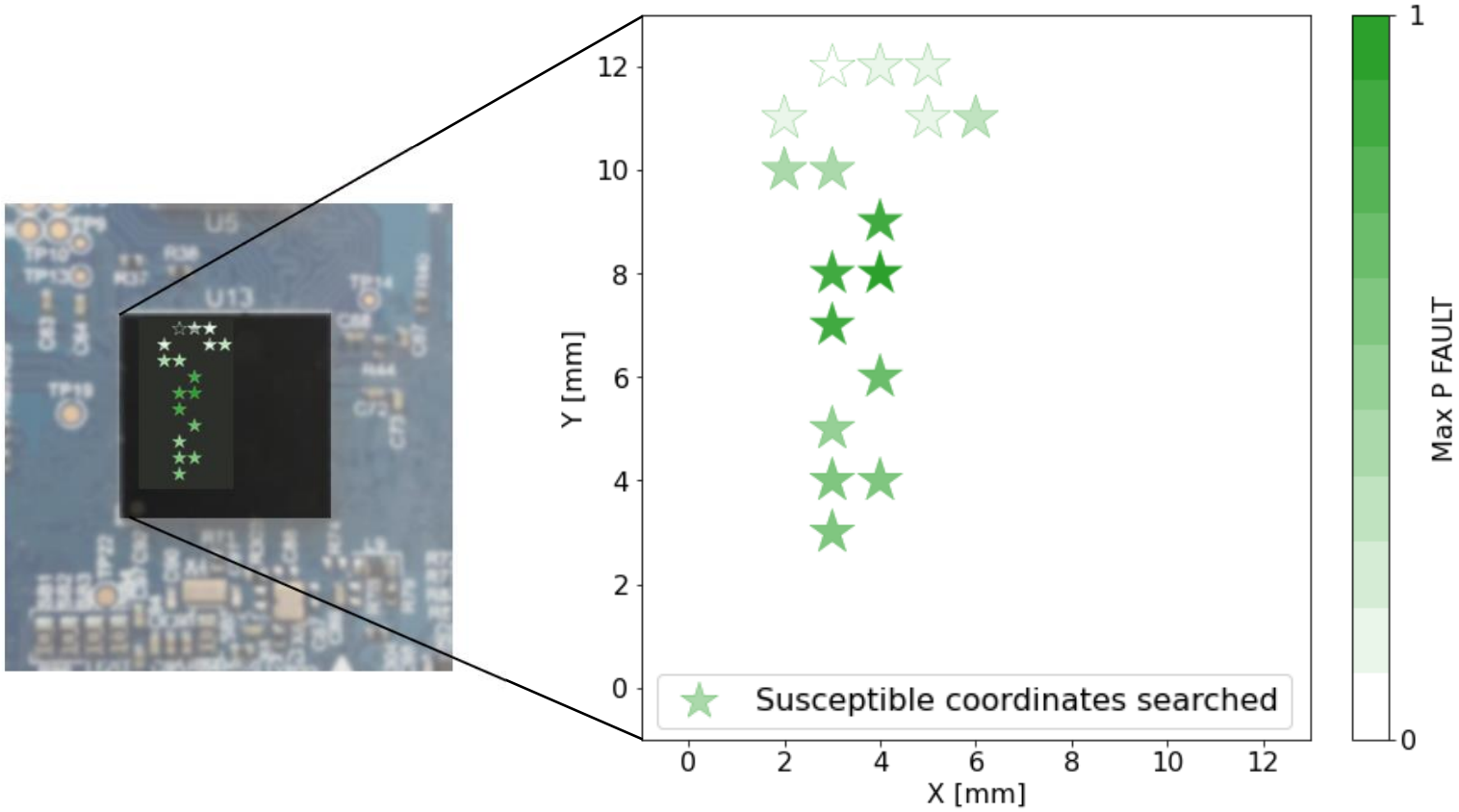
A Fault injector

- NewAE ChipShouter
- V_{min} = 150 V
- V_{max} = 500 V
- V steps = 1 V
- d_{min}= 60 ns
- d_{max} = 600 ns

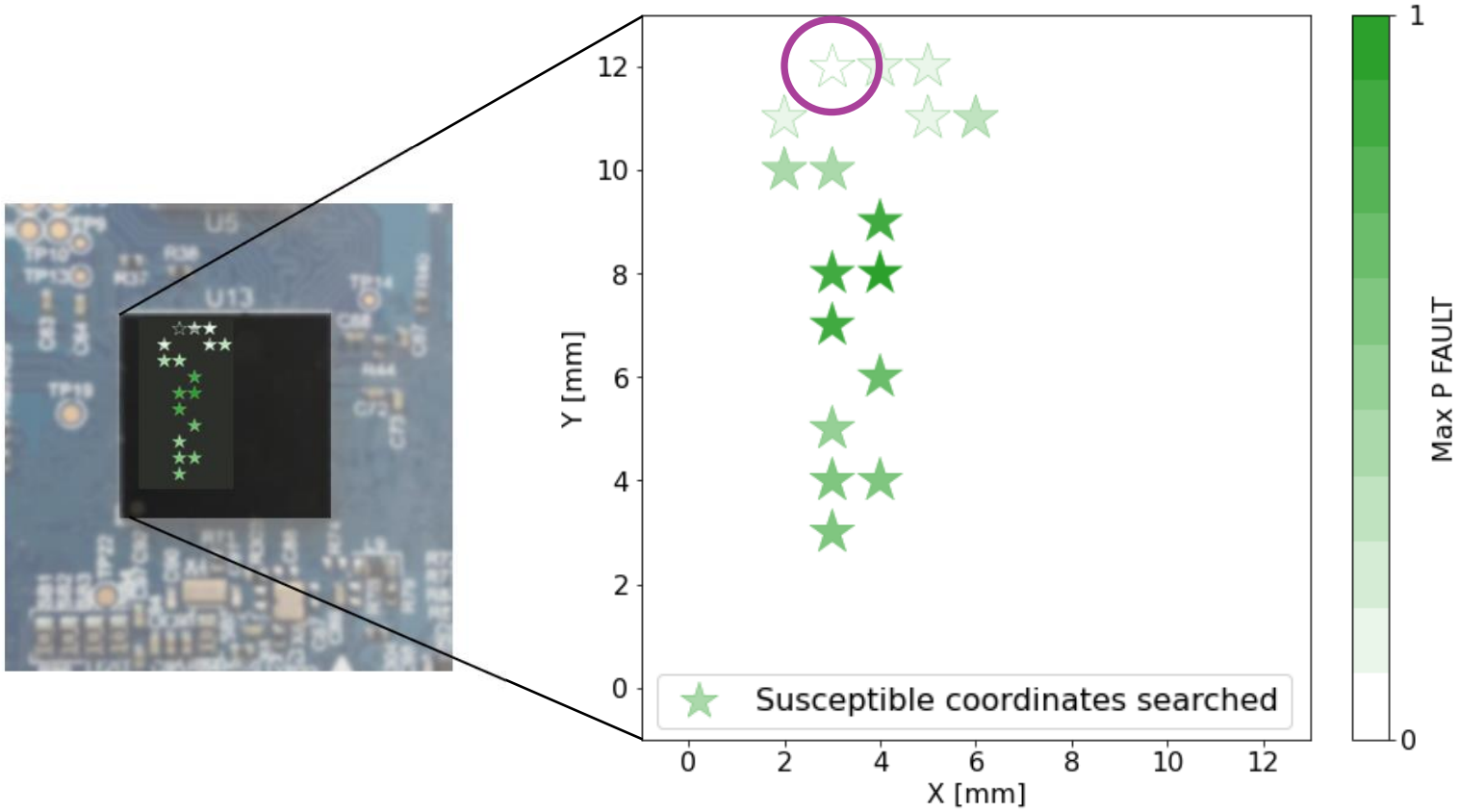
F Fault injector controller

- Artix-7 35T Arty 100 MHz
- d steps = 10 ns

Methodology outcome



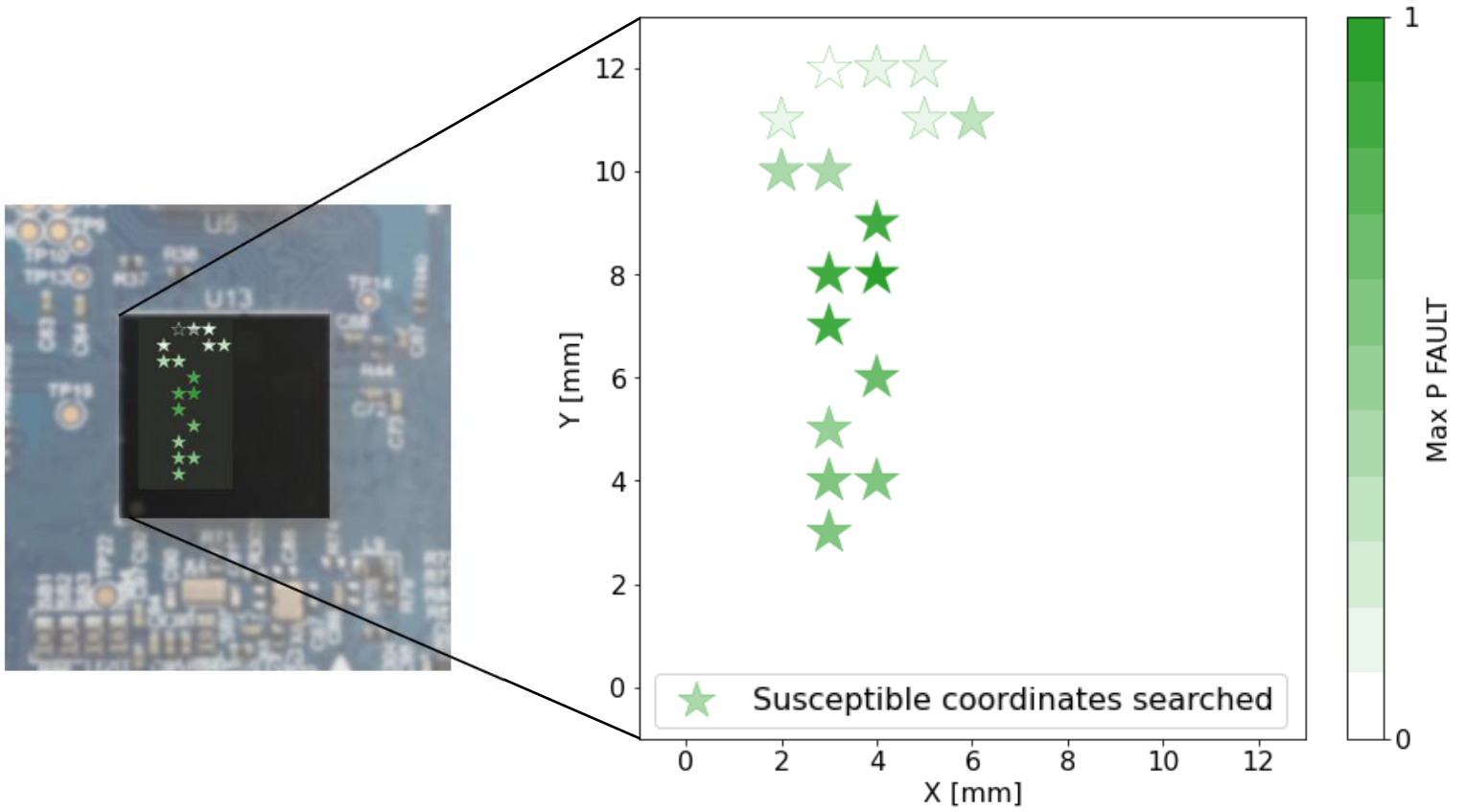
Susceptibility criterion validation

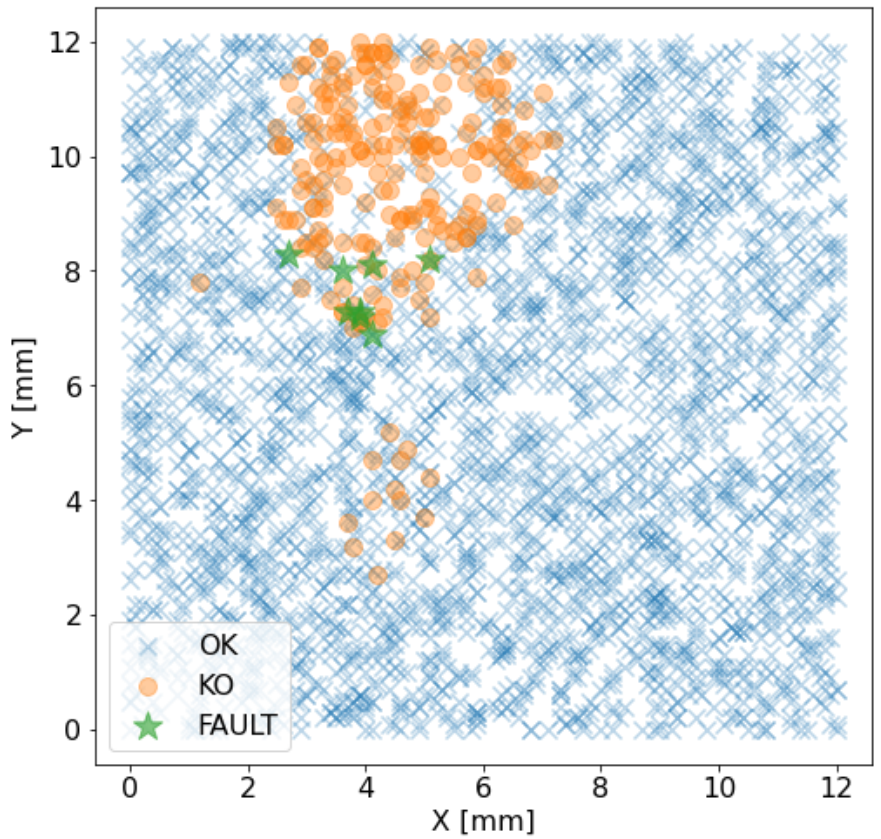


False Positive Ratio

5.4%

Surface search validation



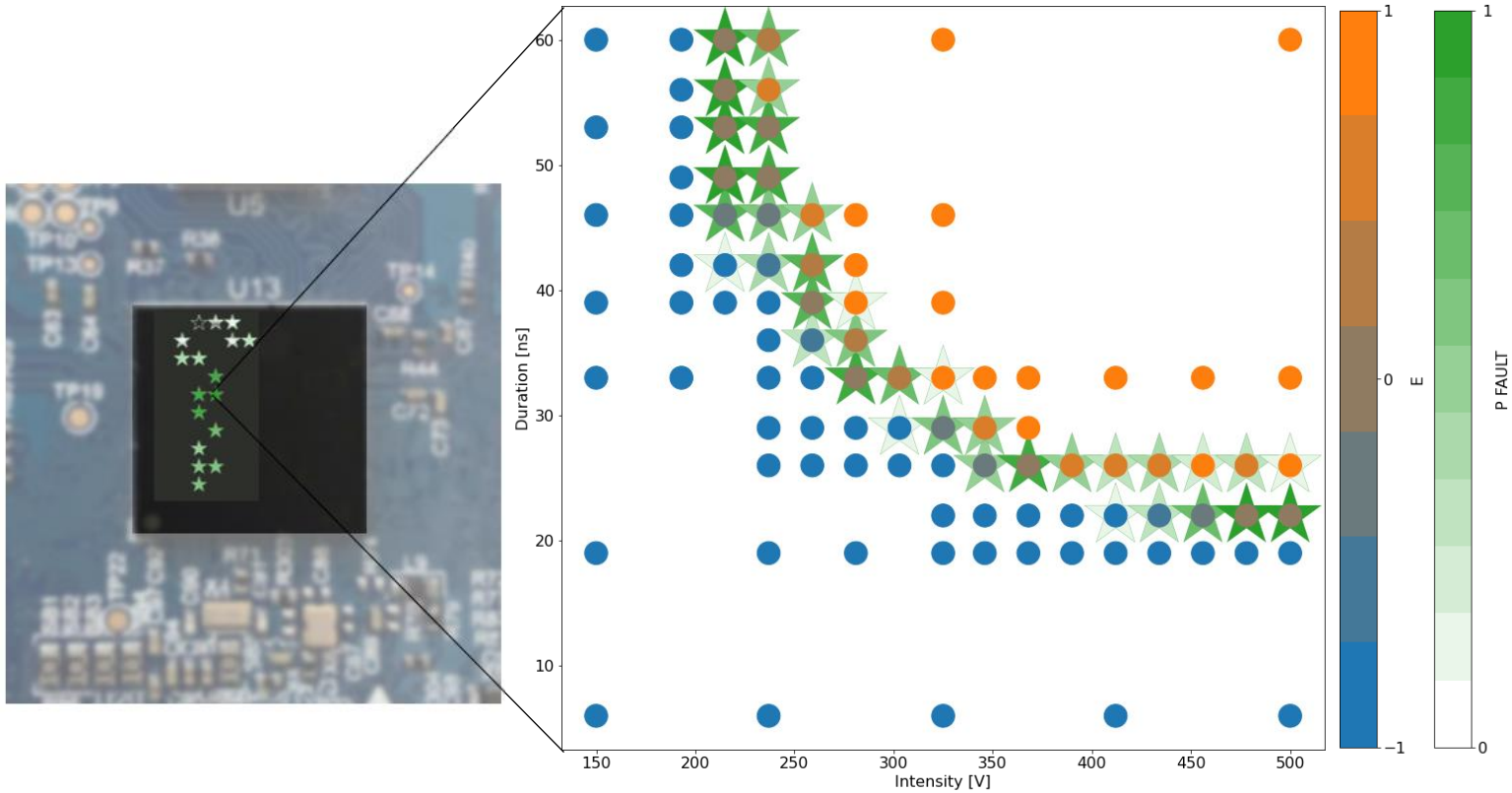


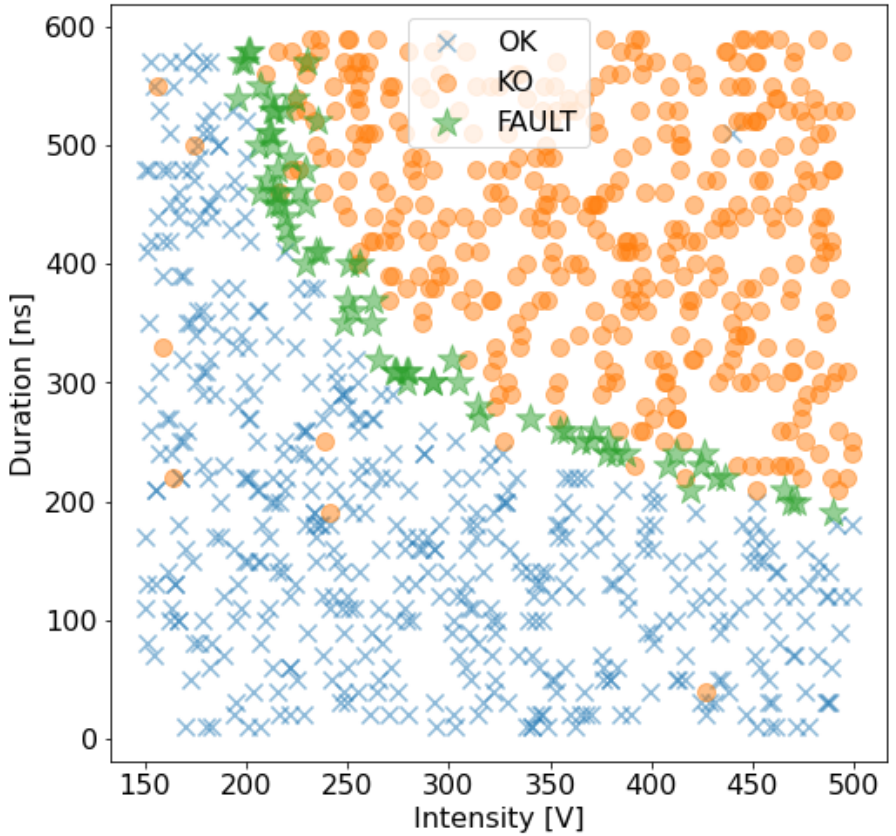
False Positive Ratio
5.4%

Coverage
100%

Surface Reduction
21.4%

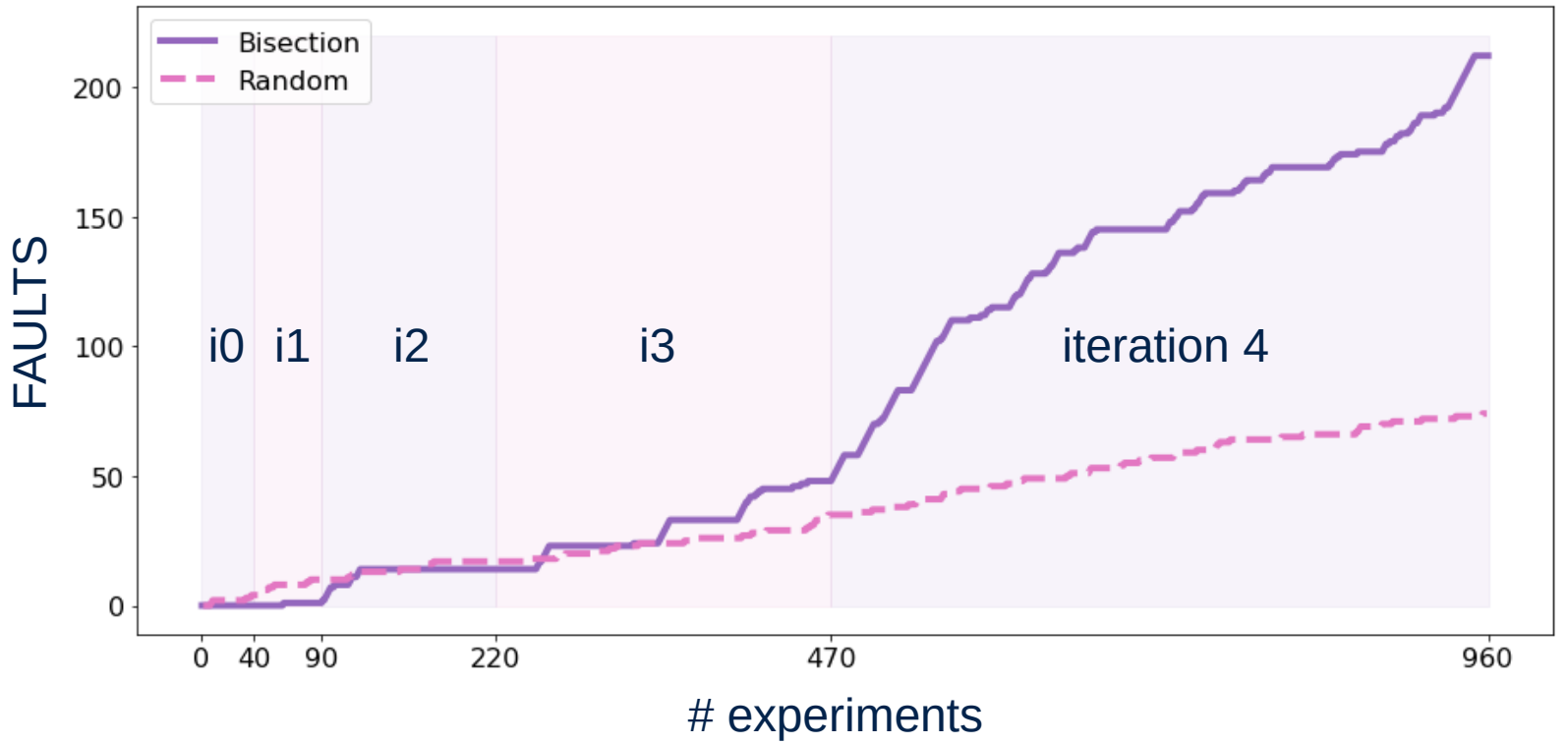
Coordinate search validation





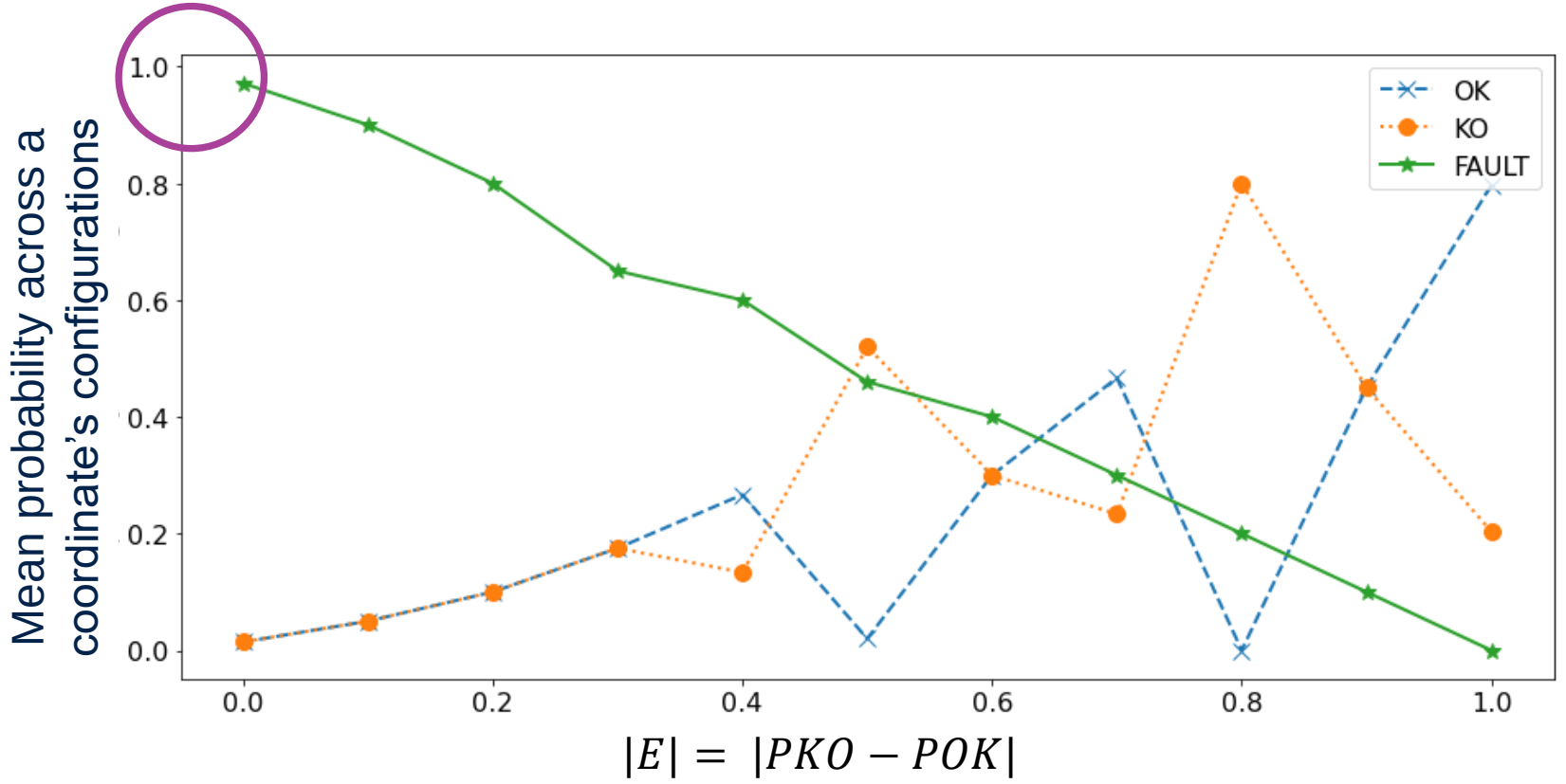
Bisection vs random visual coverage

Fault occurrence



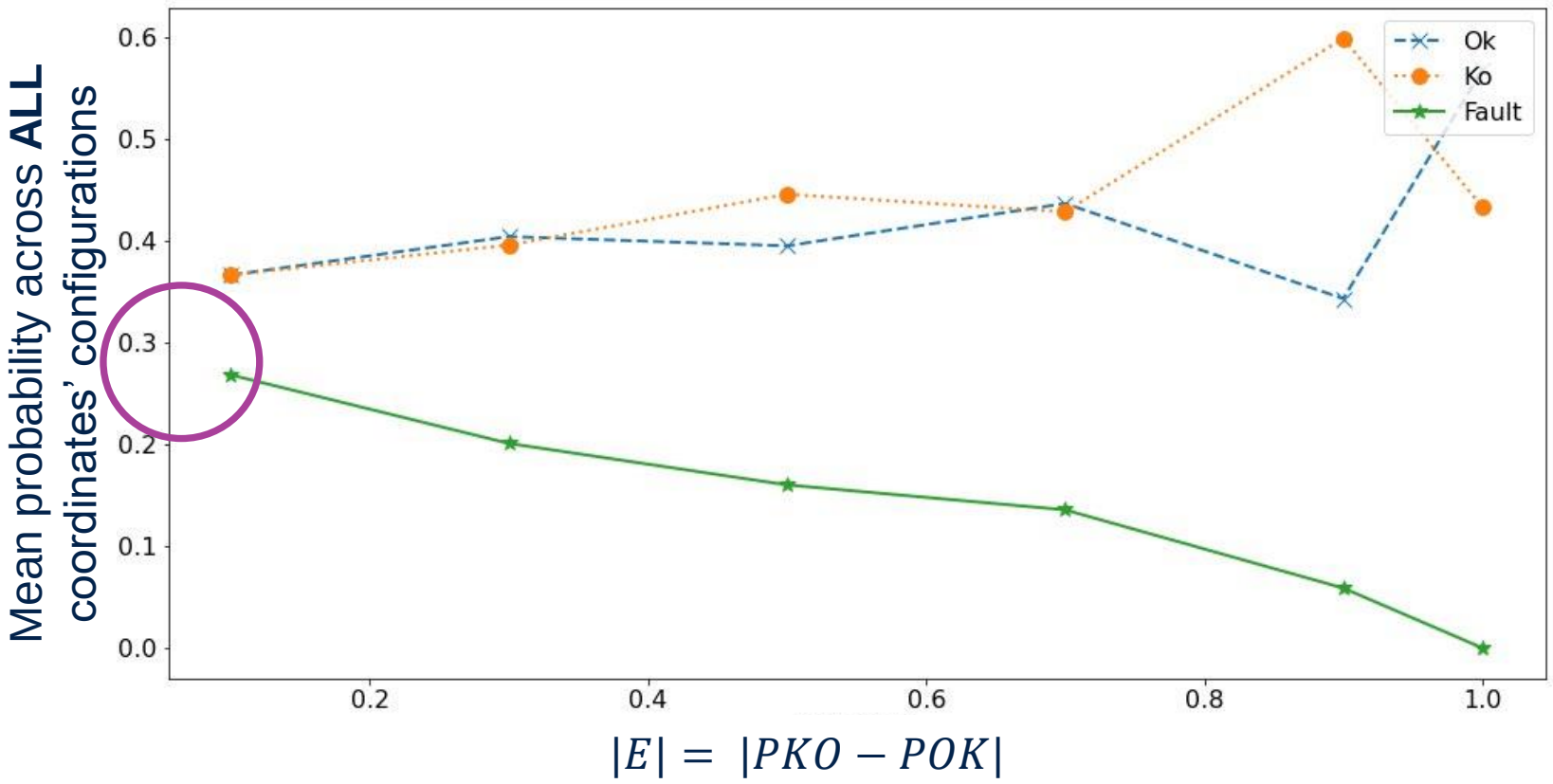
The faults' amount difference builds up with the search algorithm convergence

pFAULT related with pKO and pOK



On this coordinate, the closer you get to the roots, the higher the **pFAULT** is

pFAULT related with pKO and pOK



On **average**, the closer you get to the roots, the higher the **pFAULT** is

Achievements

- Methodological approach
  - Versatile (target architecture agnostic)
  - Efficient (random comparison)
  - Repeatable (low-cost equipment and open-source tooling)

- Tests performed on a state-of-the-art target
- Search for a rare FAULT behavior using a more common OK and KO
- Introduced successfully the duration variable proving its effectiveness
- Validated assumptions for susceptibility and search criteria

Future work

- Test on different targets
- Optimize the algorithm under distribution assumptions
- Tune
  - Number of experiments to make an evaluation
  - Threshold
- Develop the outcome of the methodology and the achieved FAULT differentiation

Thank you!
Any Questions?

daniele.carta@st.com