Fraunhofer AISEC

HARRIS Workshop 2024

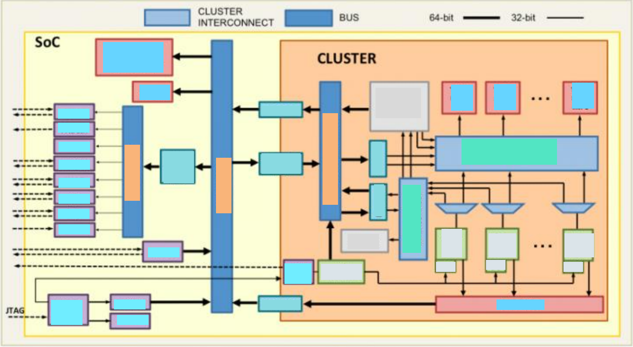
Exploring Netlist Reverse Engineering Benchmarks: Existing Approaches and Future Requirements

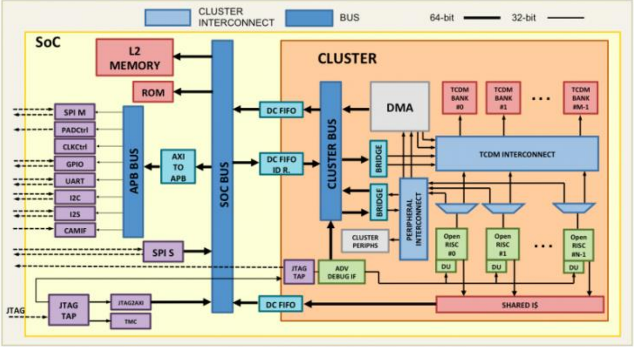
Netlist Analysis

The “final” step in hardware reverse engineering

Possible Goals:

- Recovery of high-level functionality
- Hardware Trojan detection
- Breaking (netlist) obfuscation

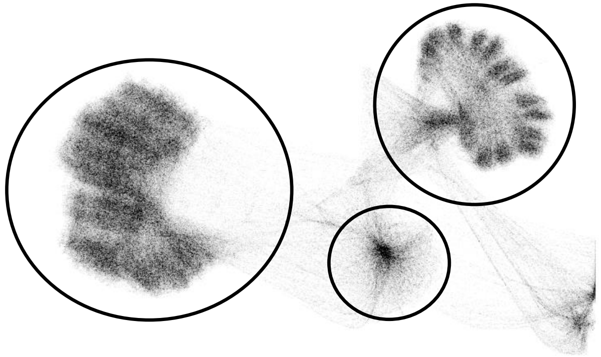


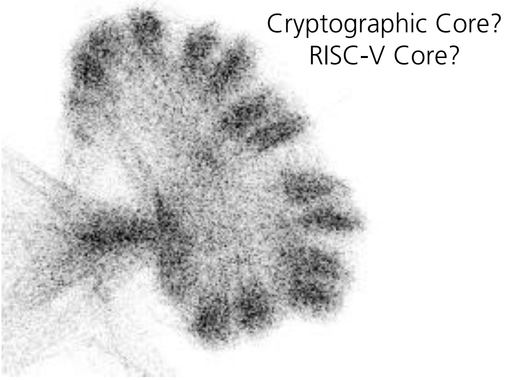


Functionality Recovery:

“Divide and Conquer approach”

1. Partition
2. Identify functionality (by comparison)





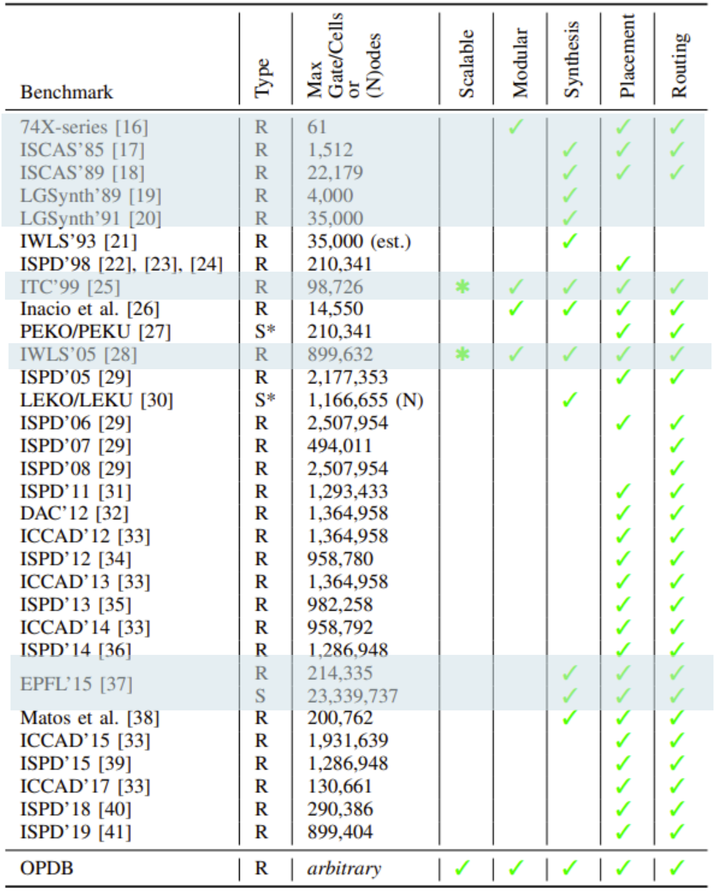
Circuit Benchmarks

Why do we need Benchmarks for netlist reverse engineering?

- comparable evaluation of methods
- real-world evaluation of methods
- training data for (supervised) machine learning methods

Circuit Benchmarks created for:

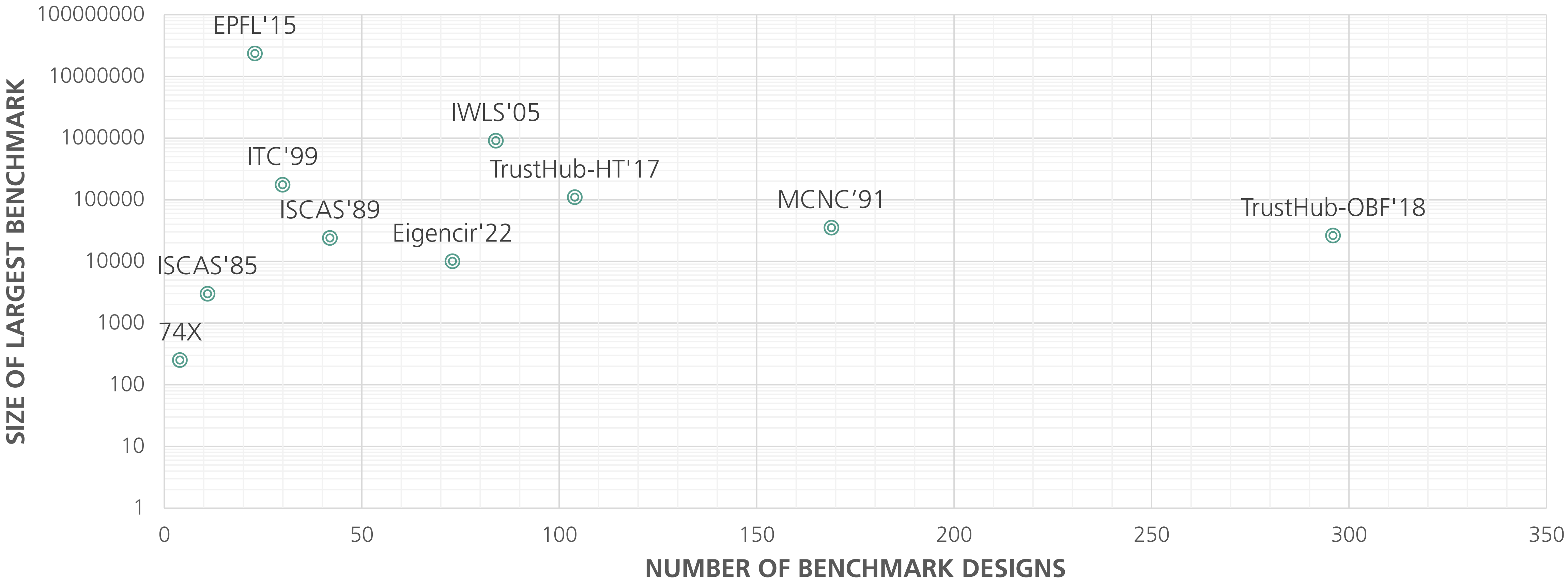
- **EDA Optimisation**
- Hardware Trojan detection
- Obfuscation / Deobfuscation



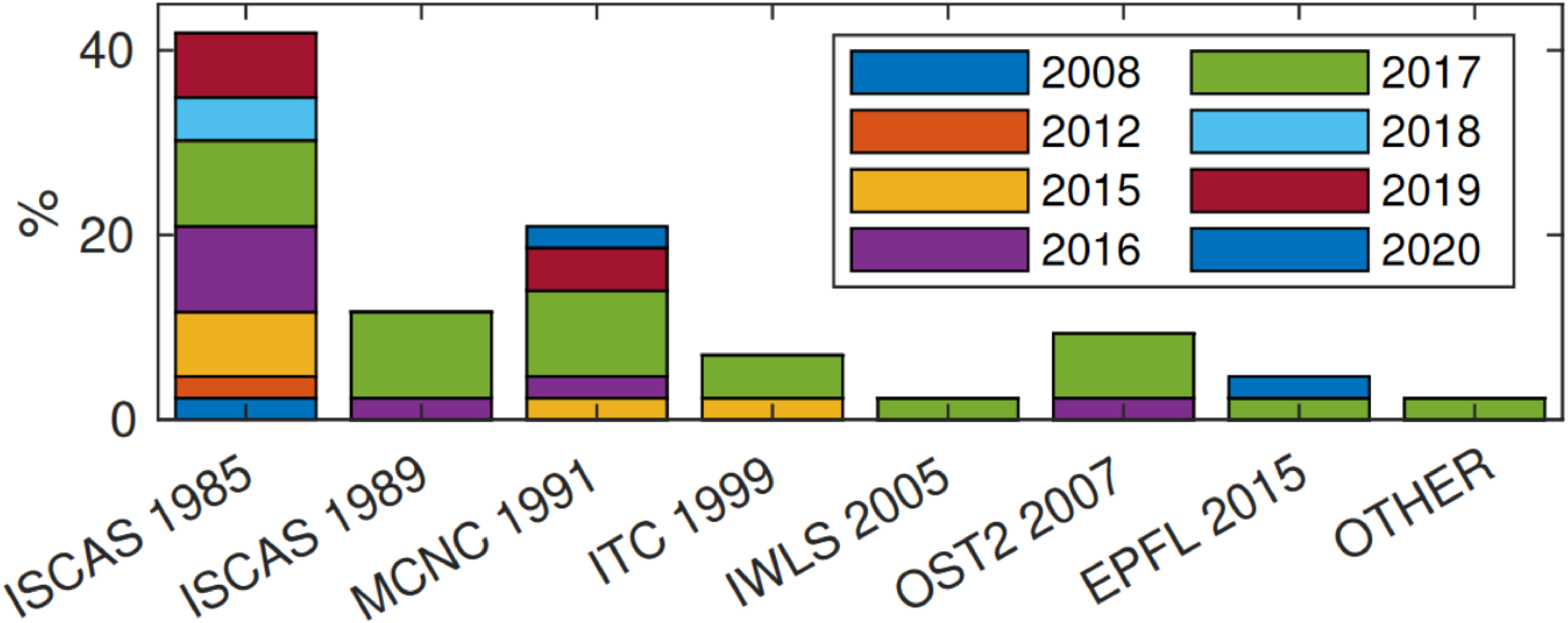
Benchmark	Type	Max Gate/Cells or (N)odes	Scalable	Modular	Synthesis	Placement	Routing
74X-series [16]	R	61		✓		✓	✓
ISCAS'85 [17]	R	1,512			✓	✓	✓
ISCAS'89 [18]	R	22,179			✓	✓	✓
LGSynth'89 [19]	R	4,000			✓		
LGSynth'91 [20]	R	35,000			✓		
IWLS'93 [21]	R	35,000 (est.)			✓		
ISPD'98 [22], [23], [24]	R	210,341				✓	
ITC'99 [25]	R	98,726	*	✓	✓	✓	✓
Inacio et al. [26]	R	14,550		✓	✓	✓	✓
PEKO/PEKU [27]	S*	210,341				✓	✓
IWLS'05 [28]	R	899,632	*	✓	✓	✓	✓
ISPD'05 [29]	R	2,177,353				✓	✓
LEKO/LEKU [30]	S*	1,166,655 (N)			✓		
ISPD'06 [29]	R	2,507,954				✓	✓
ISPD'07 [29]	R	494,011					✓
ISPD'08 [29]	R	2,507,954					✓
ISPD'11 [31]	R	1,293,433				✓	✓
DAC'12 [32]	R	1,364,958				✓	✓
ICCAD'12 [33]	R	1,364,958				✓	✓
ISPD'12 [34]	R	958,780				✓	✓
ICCAD'13 [33]	R	1,364,958				✓	✓
ISPD'13 [35]	R	982,258				✓	✓
ICCAD'14 [33]	R	958,792				✓	✓
ISPD'14 [36]	R	1,286,948				✓	✓
EPFL'15 [37]	R	214,335			✓	✓	✓
	S	23,339,737			✓	✓	✓
Matos et al. [38]	R	200,762			✓	✓	✓
ICCAD'15 [33]	R	1,931,639				✓	✓
ISPD'15 [39]	R	1,286,948				✓	✓
ICCAD'17 [33]	R	130,661				✓	✓
ISPD'18 [40]	R	290,386				✓	✓
ISPD'19 [41]	R	899,404				✓	✓
OPDB	R	*arbitrary*	✓	✓	✓	✓	✓

Tziantzioulis, Georgios, Ting-Jung Chang, Jonathan Balkind, Jinzheng Tu, Fei Gao, and David Wentzlaff. "OPDB: A Scalable and Modular Design Benchmark." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, no. 6 (June 2022): 1878–87. https://doi.org/10.1109/TCAD.2021.3096794.

Details of commonly used Circuit Benchmarks

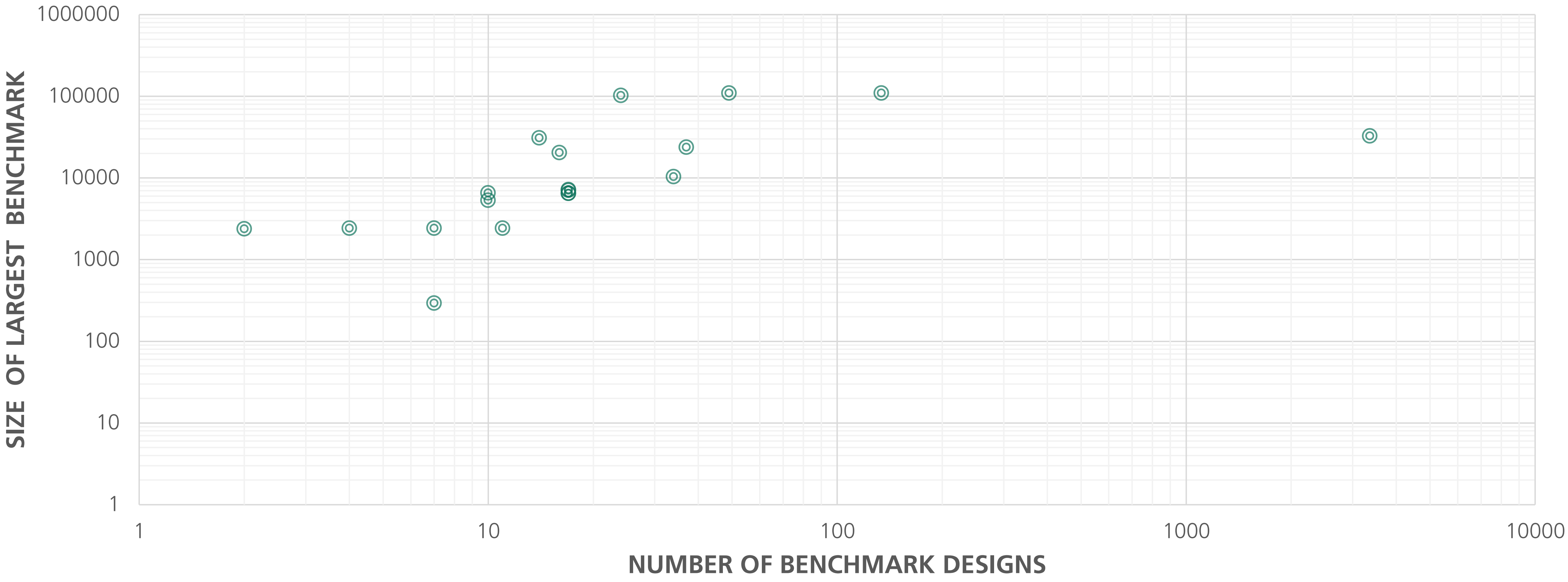


Benchmarks used for Obfuscation techniques and attacks from 2008 - 2020

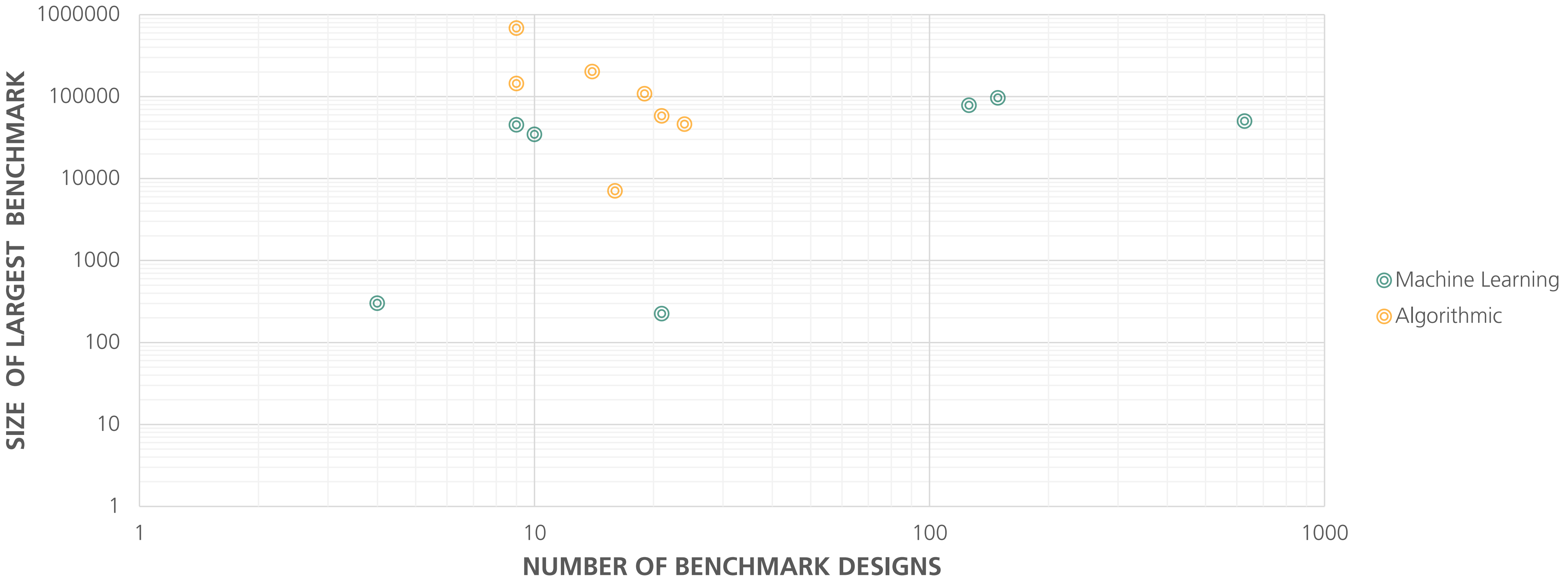


Amir, Sarah, and Domenic Forte. "EigenCircuit: Divergent Synthetic Benchmark Generation for Hardware Security Using PCA and Linear Programming." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022, 1–1. https://doi.org/10.1109/TCAD.2022.3166675.

Benchmarks used for Hardware Trojan methods from 2022 - 2024



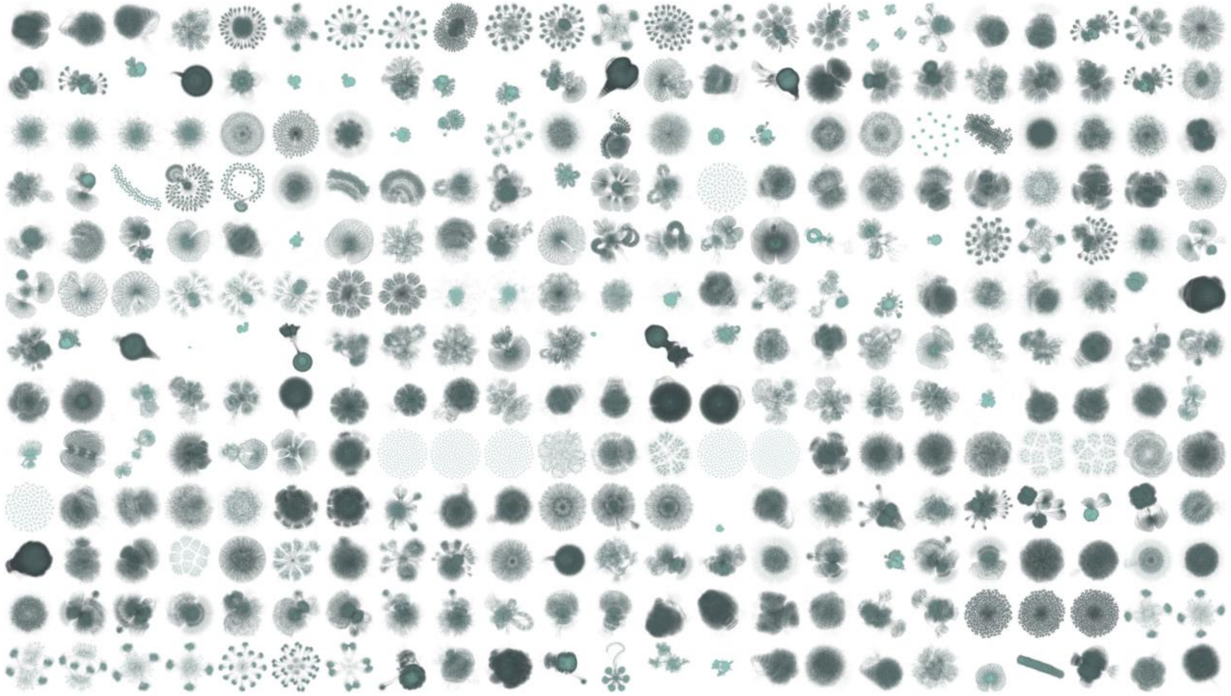
Benchmarks used for Netlist Reverse Engineering methods from 2018 - 2023



Requirements for Netlist Benchmarks

- Comparable evaluation of methods, issues are:
  - Synthesis tool, optimisations (NDAs)
  - Technology libraries (NDAs)
  - Random methods
  - Random errors

- Real-world evaluation of methods, requires Benchmarks with:
  - Large size (1,000,000+ gates)
  - Different functionalities
  - Meaningful objective
  - Netlist and partitioning errors (1%?)
  - Module data (for partitioning)
  - Functionally correct



- Training data for machine learning methods, requires Benchmark suites with:
  - Many designs (10,000+, NDAs)
  - Large structural and functional variation
  - Labelled data (function, obfuscation, hardware Trojan)

Overview of proposed Netlist Reverse Engineering Benchmarks

- Comparable evaluation of methods, issues are:

Synthesis tool, optimisations (NDAs)	Open-Source EDA (QFLOW)
Technology libraries (NDAs)	Open-Source Technology
Random methods	Concrete implementation of obfuscation methods
Random errors	Concrete implementation of errors

- Real-world evaluation of methods, requires Benchmarks with:

Large size (1,000,000+ gates)	Open-Source Hardware with wide range of functionalities
Different functionalities	
Meaningful objective	
Netlist and partitioning errors (1%?)	Error Insertion Tool
Module data (for partitioning)	Hierarchy Data
Functionally correct	

- Training data for machine learning methods, requires Benchmark suites with:

Many designs (10,000+, NDAs)	Open-Source Hardware
Large structural and functional variation	
Labelled data (function, obfuscation, hardware Trojan)	Function / state / obfuscation labels

Netlist Formats:

- Verilog netlist (with tech data)
- Bench
- Adjlist
- Graph output

2100+ modules (from 120+ projects)

4 mio gates in largest design

Max 6 hierarchy levels

Next Steps

1. **Publication of Benchmarks**

2. **Addition of (open-source) layout data**

    - Distance based analysis
    - Realistic defect implementations

3. **Explicit Implementations of Hardware Trojan Insertion**

    - Automatic hardware Trojan insertion tools already exist

4. **Support for open-source VHDL Synthesis (solved: use correct ghdl-yosys plugin)**

    - Solved: use ghdl-yosys plugin

Further thoughts

What else is required?

1. **Real World Benchmarks**

2. **SEM Benchmarks**

    - First efforts exist
    - Difficult due to NDAs
    - Data augmentation and artificial image generation (including defect insertion)

3. **Hardware Trojan Benchmarks for side-channel based detection**

    - Commonly based on simulated data
    - First tests on real chip show further evaluation required

4. …. ?

Contact

Johanna Baehr
Hardware Security Department
Tel. +49 12 3456-1006
Johanna.baehr@aisec.fraunhofer.de

Fraunhofer Institute for Applied and Integrated Security AISEC
Lichtenbergstraße 11
85748 Garching near Munich
www.aisec.fraunhofer.de