Andreas Krinke
Institute for Electromechanical and Electronic Design

Generating DRC and LVS Runsets for KLayout

OpenPDK, OpenTooling and Open Source Design Workshop, IHP
Frankfurt (Oder), June 28, 2023

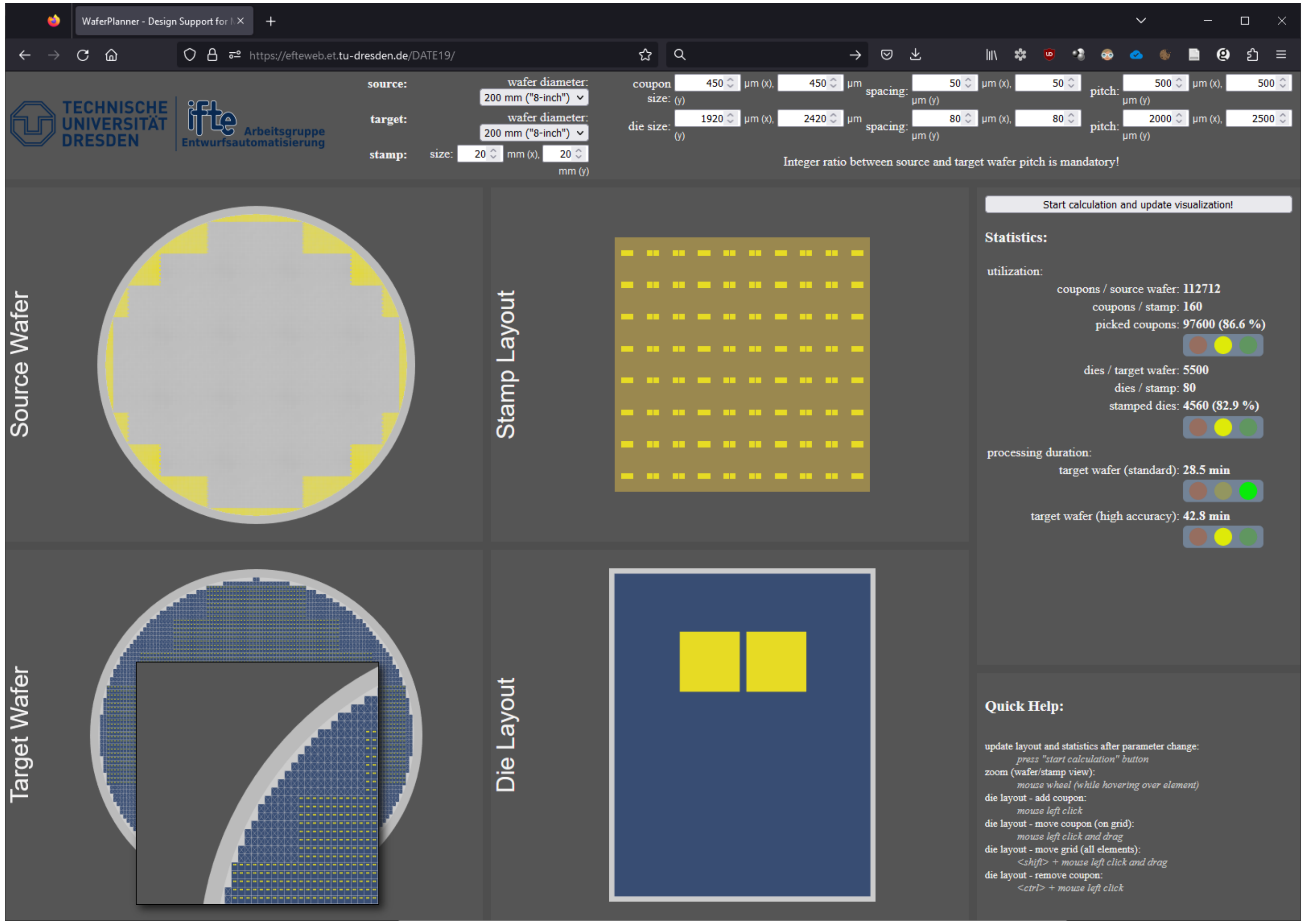
The Institute

- Director:
  Professor **Jens Lienig**
- **20** employees
- EDA group:
  8 scientific assistants
- Research & development of **algorithms** and **optimization methods**

↓

EDA tools

WaferPlanner



https://efteweb.et.tu-dresden.de/DATE19/

source: wafer diameter: 200 mm ("8-inch")
target: wafer diameter: 200 mm ("8-inch")
stamp: size: 20 mm (x), 20 mm (y)

coupon size: 450 µm (x), 450 µm (y) spacing: 50 µm (x), 50 µm (y) pitch: 500 µm (x), 500 µm (y)
die size: 1920 µm (x), 2420 µm (y) spacing: 80 µm (x), 80 µm (y) pitch: 2000 µm (x), 2500 µm (y)

Integer ratio between source and target wafer pitch is mandatory!

Statistics:

utilization:
coupons / source wafer: **112712**
coupons / stamp: **160**
picked coupons: **97600 (86.6 %)**
dies / target wafer: **5500**
dies / stamp: **80**
stamped dies: **4560 (82.9 %)**

processing duration:
target wafer (standard): **28.5 min**
target wafer (high accuracy): **42.8 min**

Quick Help:

update layout and statistics after parameter change:
press "start calculation" button
zoom (wafer/stamp view):
mouse wheel (while hovering over element)
die layout - add coupon:
mouse left click
die layout - move coupon (on grid):
mouse left click and drag
die layout - move grid (all elements):
<shift> + mouse left click and drag
die layout - remove coupon:
<ctrl> + mouse left click

WaferPlanner - Design Support for
TECHNISCHE UNIVERSITÄT DRESDEN
ifte Arbeitsgruppe Entwurfsautomatisierung
Start calculation and update visualization!
Source Wafer
Stamp Layout
Target Wafer
Die Layout

Generating DRC and LVS Runsets for KLayout

Or: How we replicate ~60% of the XH018 DRC rules

Agenda

Motivation

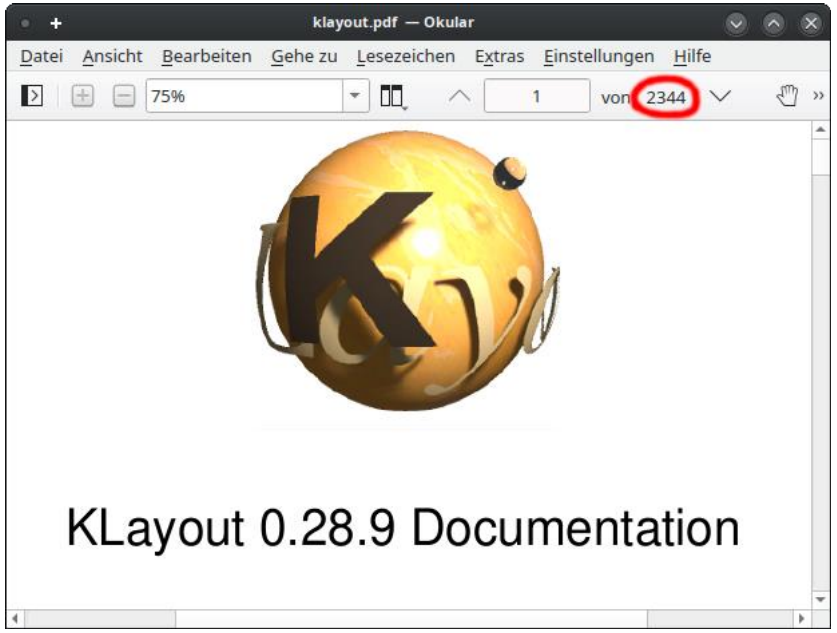
- Before production: Verification of mask layouts, e.g., **DRC**
- Design rules:
  - Part of a foundry's PDK
  - Available in proprietary formats (e.g. SVRF)
  - High licensing costs for required software tools
- Our goal: Lower barriers to entry for smaller companies by

Generating DRC and LVS runsets for KLayout

Why?

Motivation: Why KLayout?

- See previous talk…
- KLayout DRC scripts are written in **Ruby**
  - Support for many typical DRC operations:
    antenna checks, density, connectivity, …
  - Extensible
- Support for parallelization
- Comprehensive documentation
- **Strong copyleft license (GPL)**

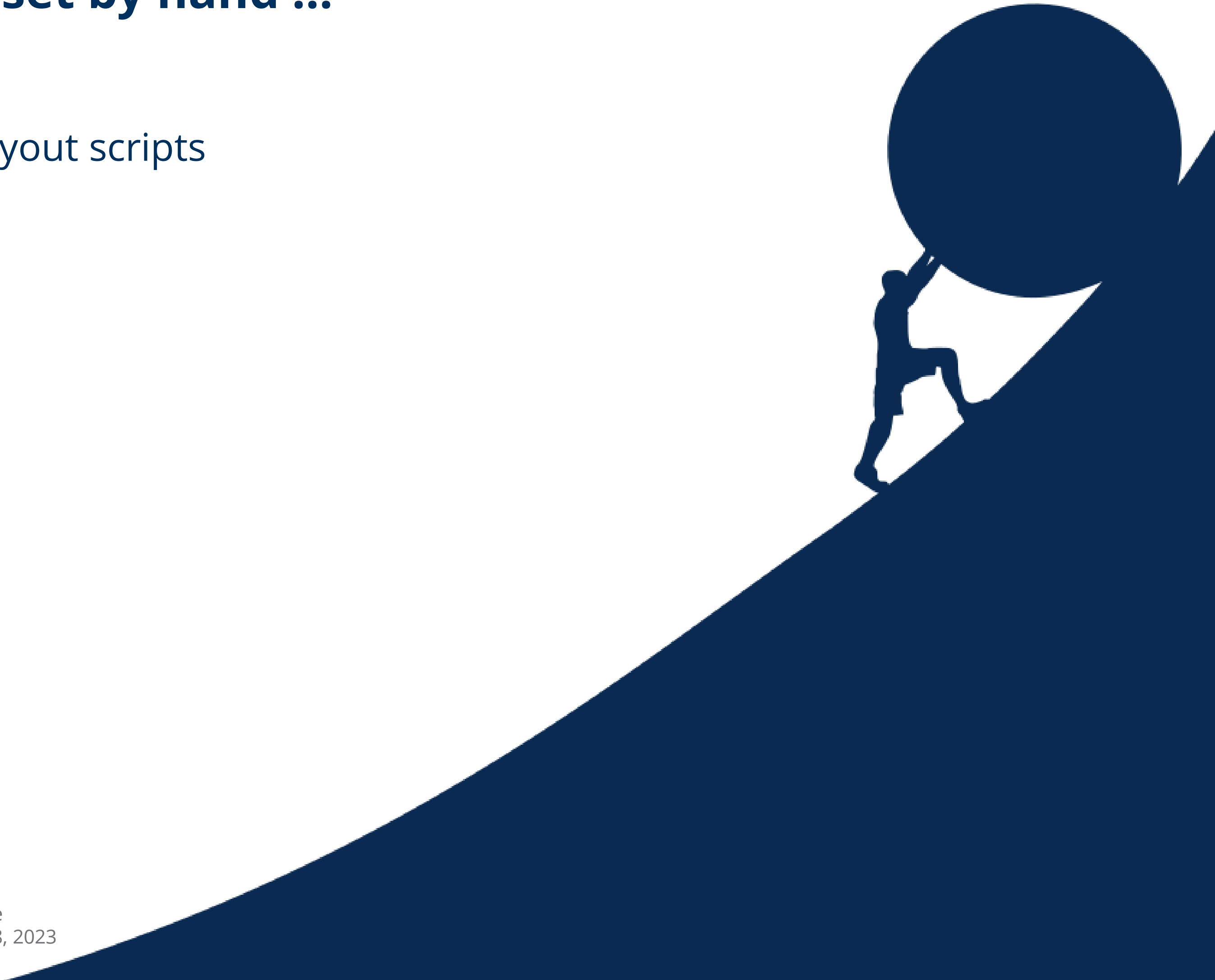

klayout.pdf — Okular
Datei Ansicht Bearbeiten Gehe zu Lesezeichen Extras Einstellungen Hilfe
75%
1
von 2344
KLayout 0.28.9 Documentation

In theory, no limits for what we can achieve

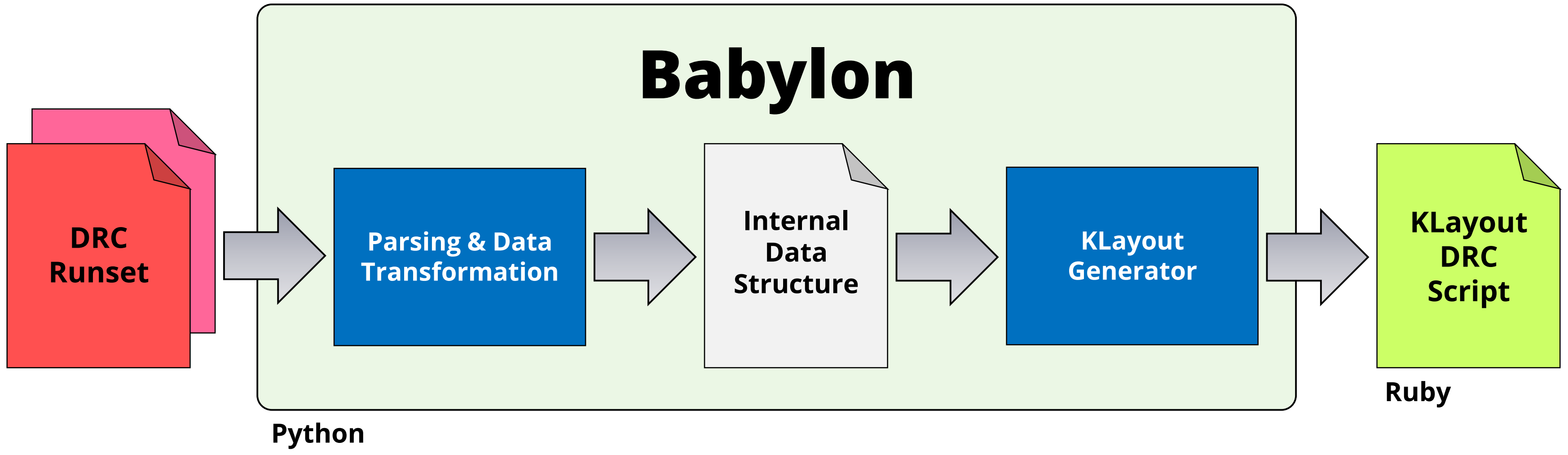
Motivation: Why generate?

If we would write the KLayout DRC runset by hand …

- Large variety of DRC commands → extensive KLayout scripts
- Possibly a lot of code for “simple” checks
- Mitigation:
  - Custom functions and methods
  - Modularization
- Still: Great effort for new technologies

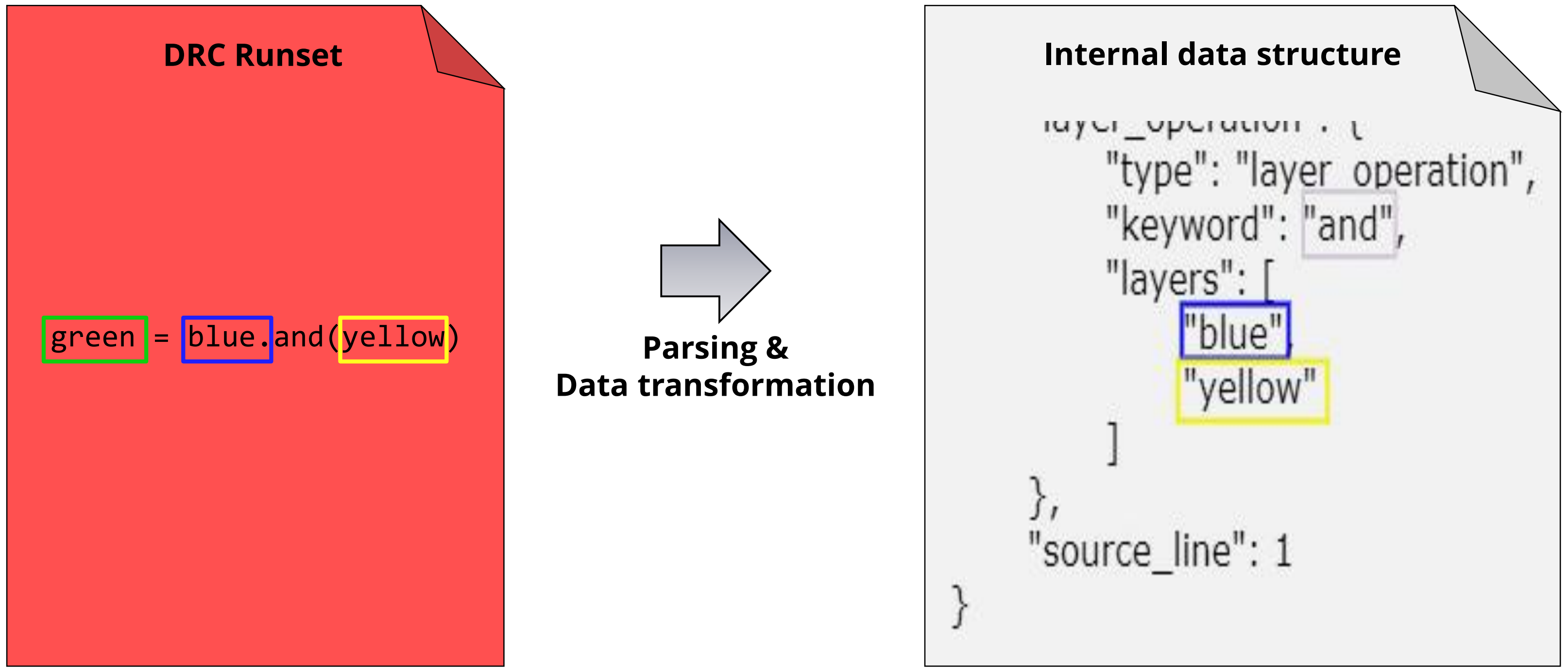


Our Approach



- Reference technology: X-FAB XH018
- Target format: KLayout DRC script (Ruby)

(Internal) Data Structure

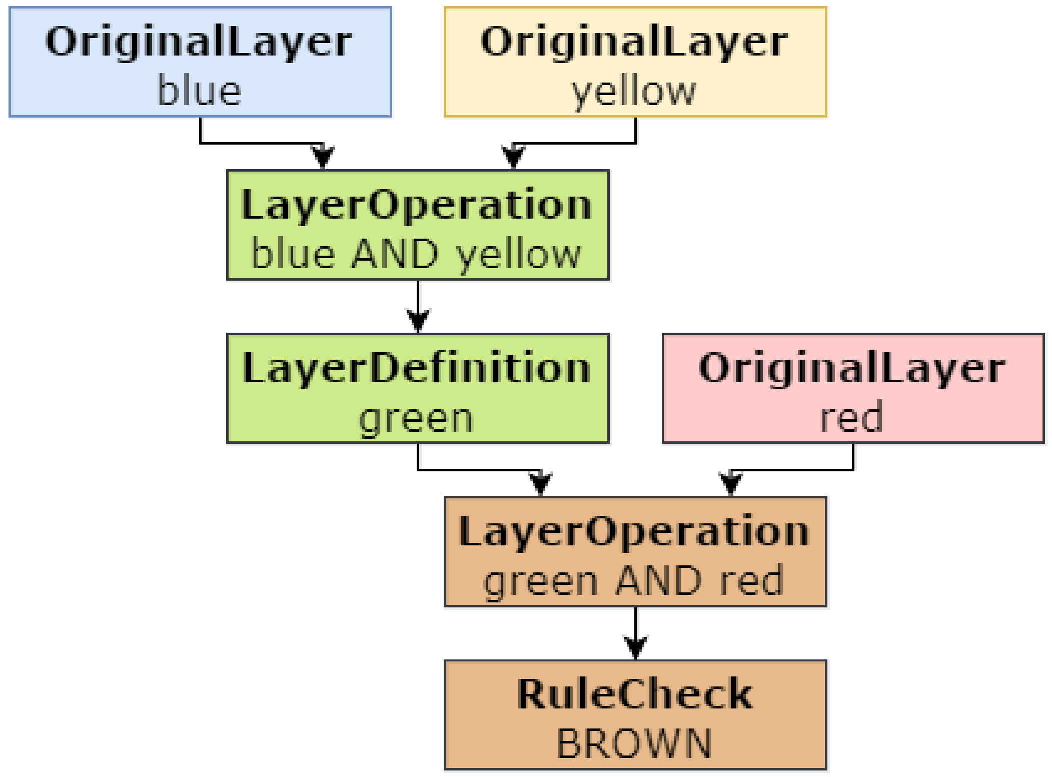


```
green = blue.and(yellow)
```

```
    "layer_operation": {
        "type": "layer_operation",
        "keyword": "and",
        "layers": [
            "blue",
            "yellow"
        ]
    },
    "source_line": 1
}
```

JSON + JSON Schema

KLayout Generator: Internal Data Model



- Can an object be represented in KLayout?
- Are all required arguments available?
- Is an object part of a design rule?

KLayout Generator: Simple things are simple

Layer assignments

```
name     = input(number)
blue     = input(1)
```

Layer definitions

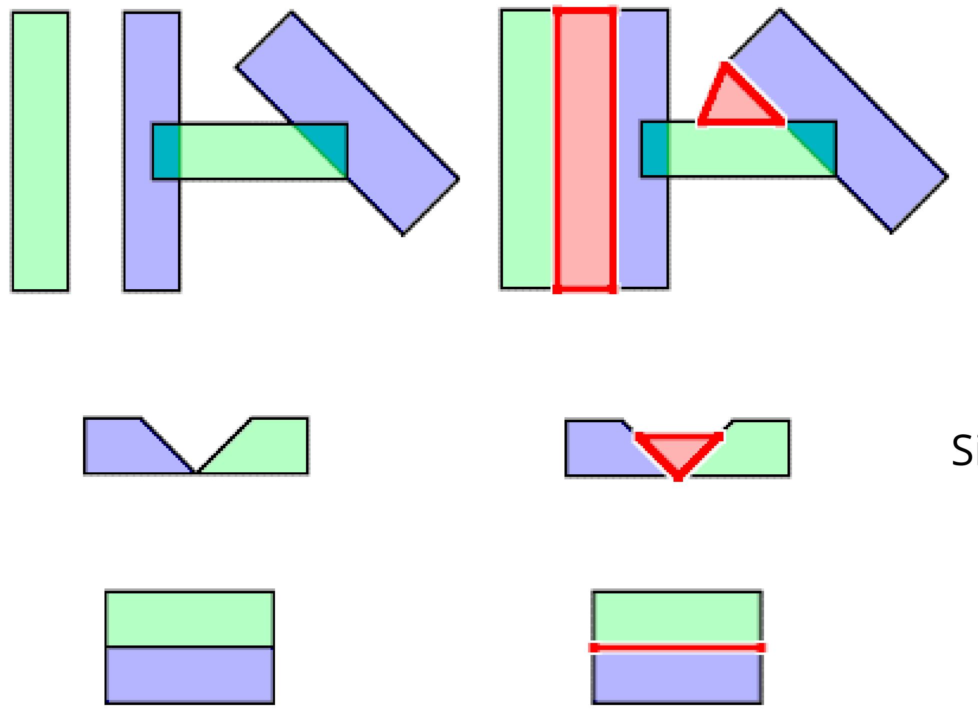
```
name     = layer operation
green    = blue.and(yellow)
```

Design rules

```
(layer operation).output(rule name, comment)
(green.and(red)).output(“BROWN”, “Is that chocolate?”)
```

KLayout Generator: Complicated things are possible

Spacing Options



Spacing between polygons

Singularities

Abutting polygons

Ruby / KLayout

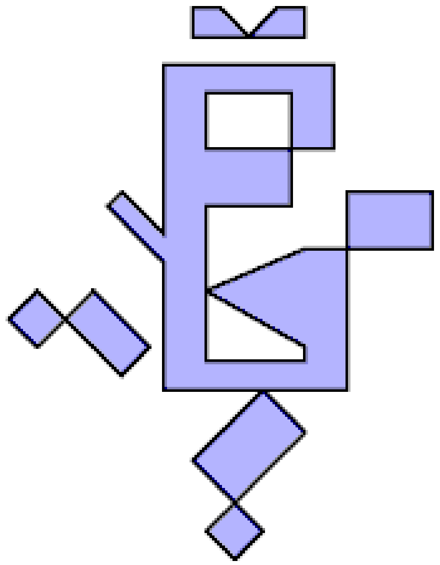
```
blue.std_external(green, 1.0)
```

```
DRCLayer
    def std_external(other, value)
        ...
        separation = ...
        ...
        singular = ...
        ...
        abut = ...
        return (separation+singular+abut)
    end
end
```

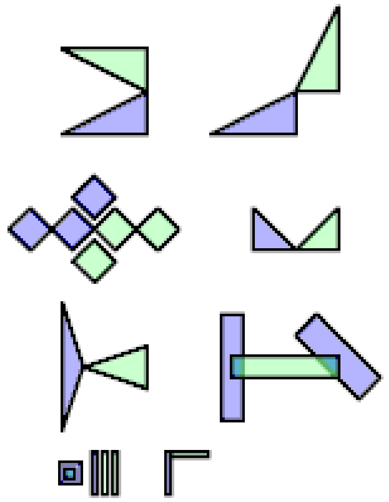
KLayout Generator: Verification Using Test Layouts

- Behavior of layer operations verified on test layouts
- Partial or full support of **33 layer operations** from the design manual

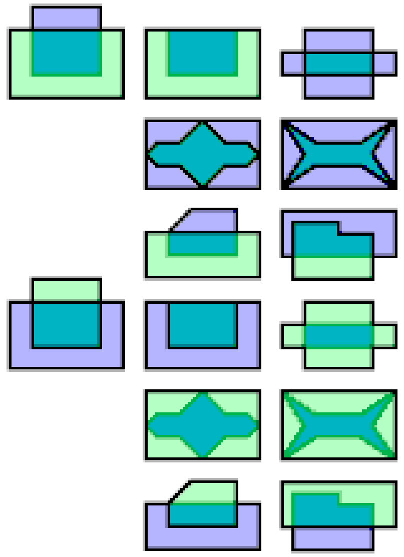
Width
(INTERNAL)



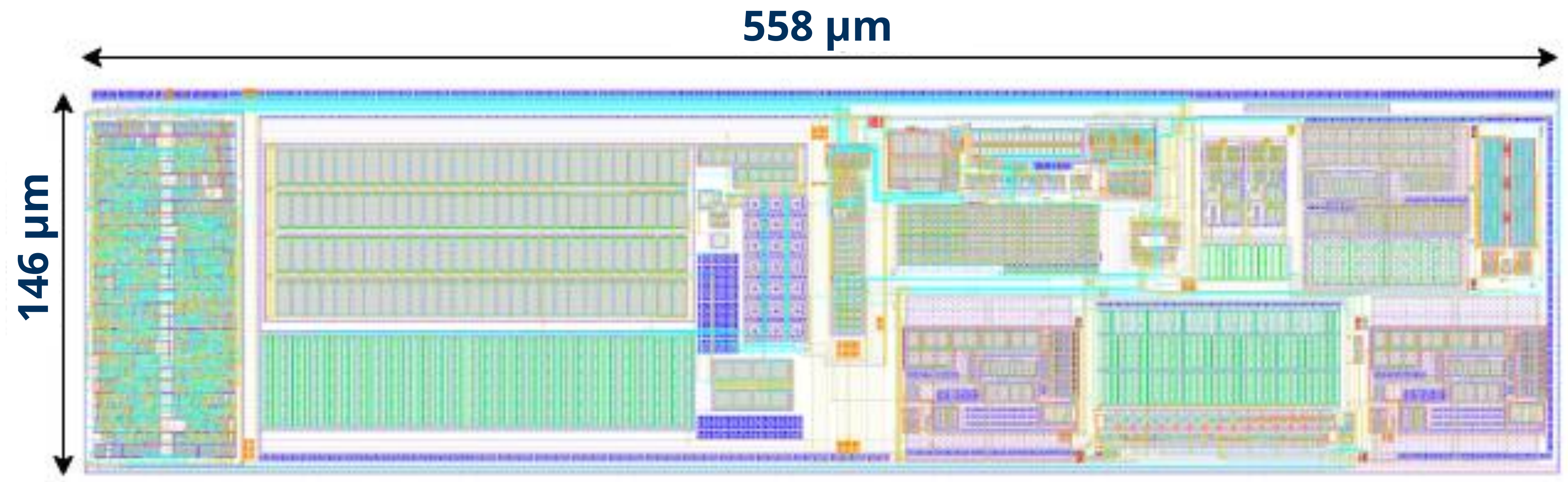
Spacing
(EXTERNAL)



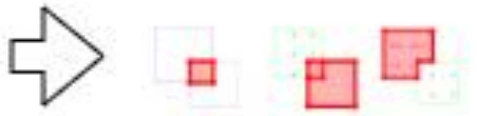
Overlap
(ENCLOSURE)



Verification on XH018 Example Layout



Only intended DRC errors



Final KLayout DRC Script

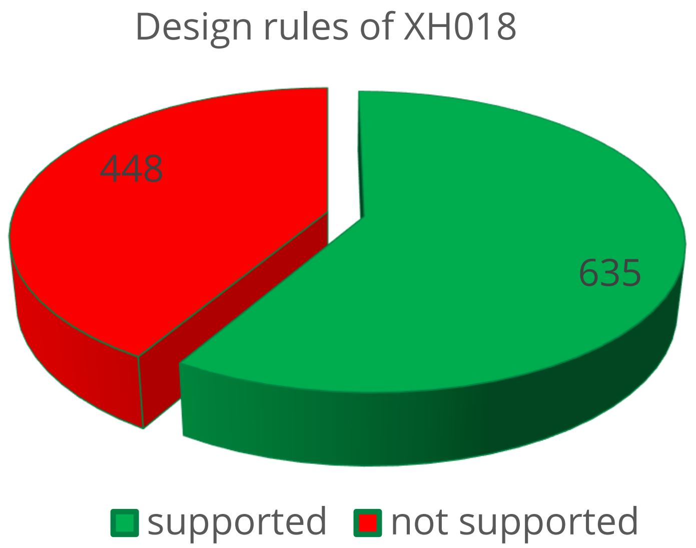
- Every element in the final KLayout DRC script …
  - … is supported in KLayout
  - … has access to all required data
  - … is part of a complete rule check

- Layer operations were verified on a test layout

Final DRC script is always executable

Current Status

Support for **<u>59%</u>** of the 1083 rule checks of XH018



Next Step: LVS

- **4 major steps:**
  - Device recognition
  - Device parameter calculation
  - Connectivity extraction
  - Netlist comparison

- **Support of new device types requires**
  - New `DeviceExtractor` class
  - New `DeviceParameterCompare` class

Next Step: LVS

- **4 major steps:**
  - Device recognition → `DeviceExtractor::setup`
  - Device parameter calculation → `DeviceExtractor::extract_devices`
  - Connectivity extraction → `connect`
  - Netlist comparison → `DeviceParameterCompare::less`

- **Support of new device types requires**
  - New `DeviceExtractor` class
  - New `DeviceParameterCompare` class

Conclusion

- Comprehensive DRC *and* LVS for commercial technologies using KLayout
- Including, e.g.
  - Marker Browser
  - Creation of result databases (RDBs)
- **Input:** Custom Rule Format (JSON) (→ Open Rule Format ?)
- **Output:** KLayout DRC script (Ruby)

- Our generator is <u>not</u> open source; output can be part of an OpenPDK

Outlook

EM-DRC

- Goal: Electromigration check based on new stress-based EM models
- Inputs: Currents, interconnect geometries
- Tool: KLayout

Assembly Rule Check

- Goals:
  - Formal description of packaging technologies
  - Automatic generation of DRC runsets
- Tool: KLayout

Outlook

EM-DRC

- Goal: Electromigration check based on new stress-based EM models
- Inputs: Currents, interconnect geometries
- Tool: KLayout

Assembly Rule Check

- Goals:
  - Formal description of packaging technologies
  - Automatic generation of DRC runsets
- Tool: KLayout

Thank you!