Entwurfsautomatisierung mit Layout-Generatoren für die frühzeitige Berücksichtigung von Parasitäten in A/MS-IC-Designs

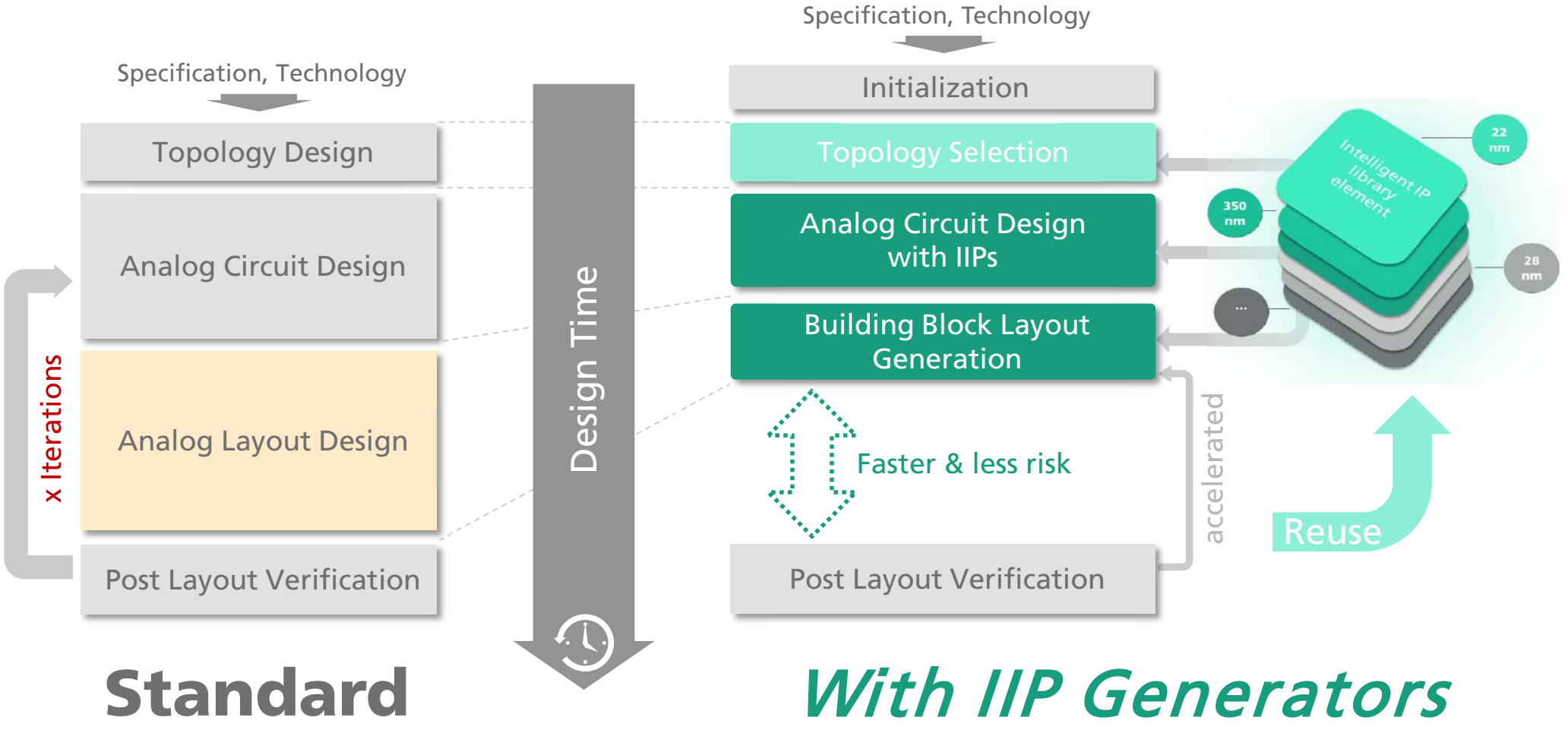
edaWorkshop 2025

Uwe Eichler, Torsten Reich, Benjamin Prautsch

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

Automated Analog IC Design with Intelligent IP

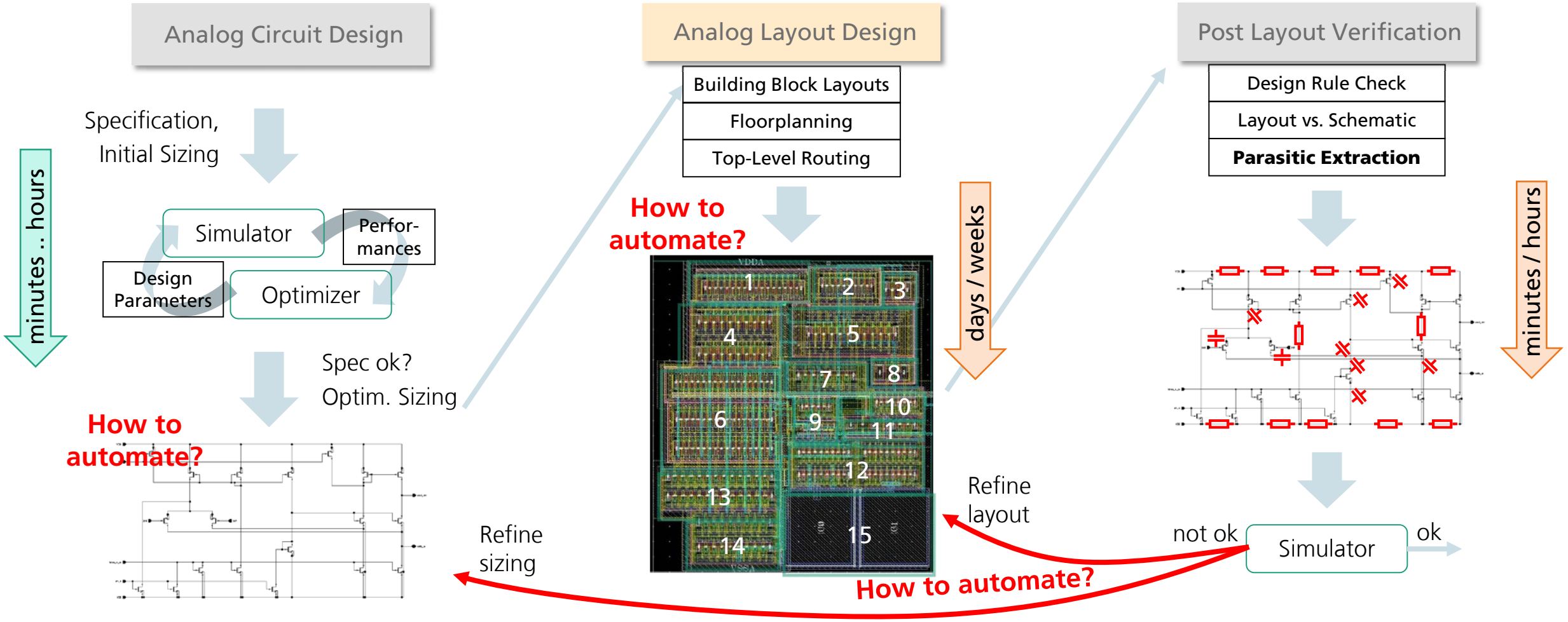
Intelligent IP (IIP) Technology: Generator-based and Reusable Analog IC Design



© Fraunhofer IIS
Restricted

Relevance and Challenge of Analog IC Design

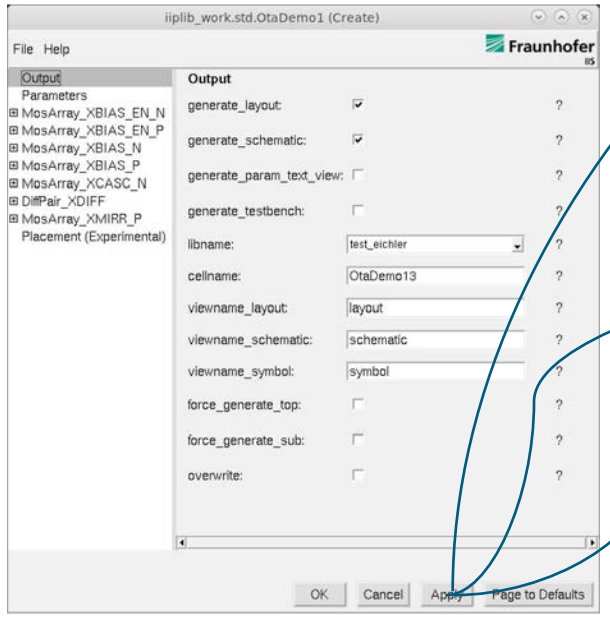
The Challenge of Analog Design – Few Layout Terms



© Fraunhofer IIS
Restricted

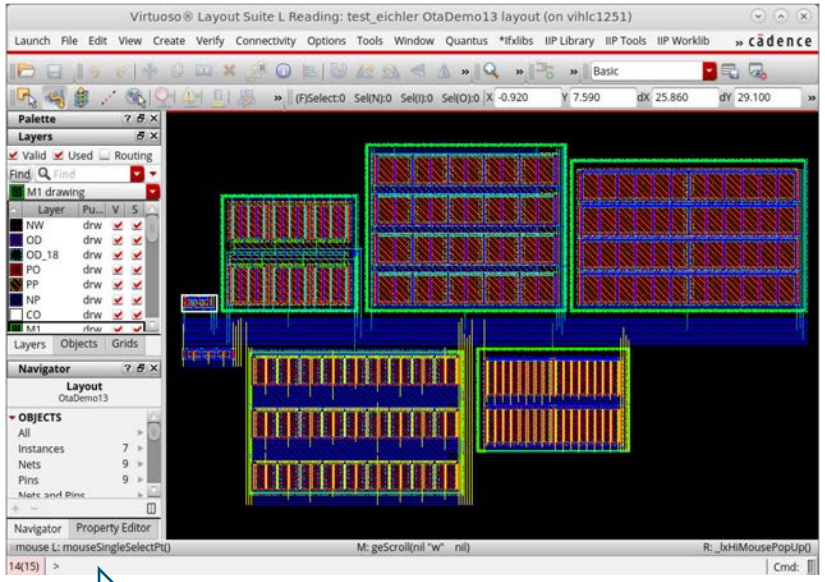
What is a Generator?

Generator "OtaDemo1"

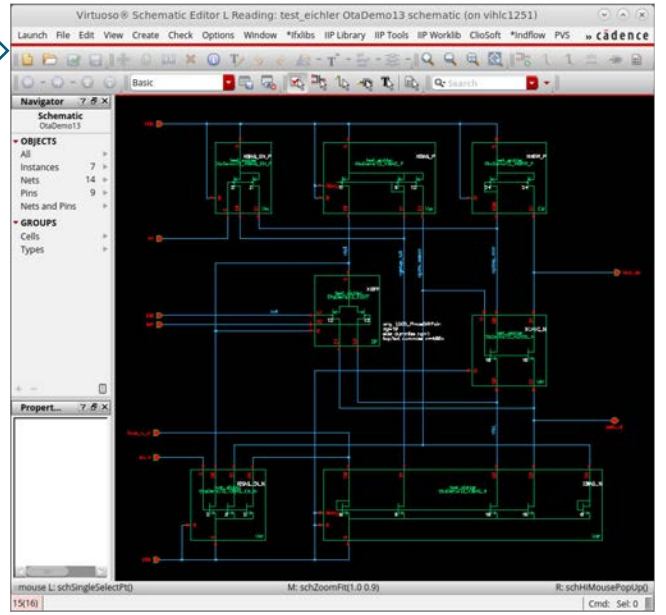


- Adjust parameters for e.g. topology, devices, sizing, placement, routing
- Generates cellviews in your design library (static, not as Pcell)

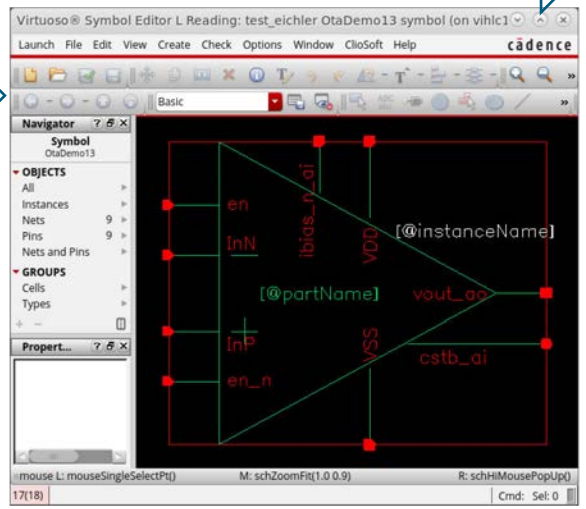
layout



schematic



symbol



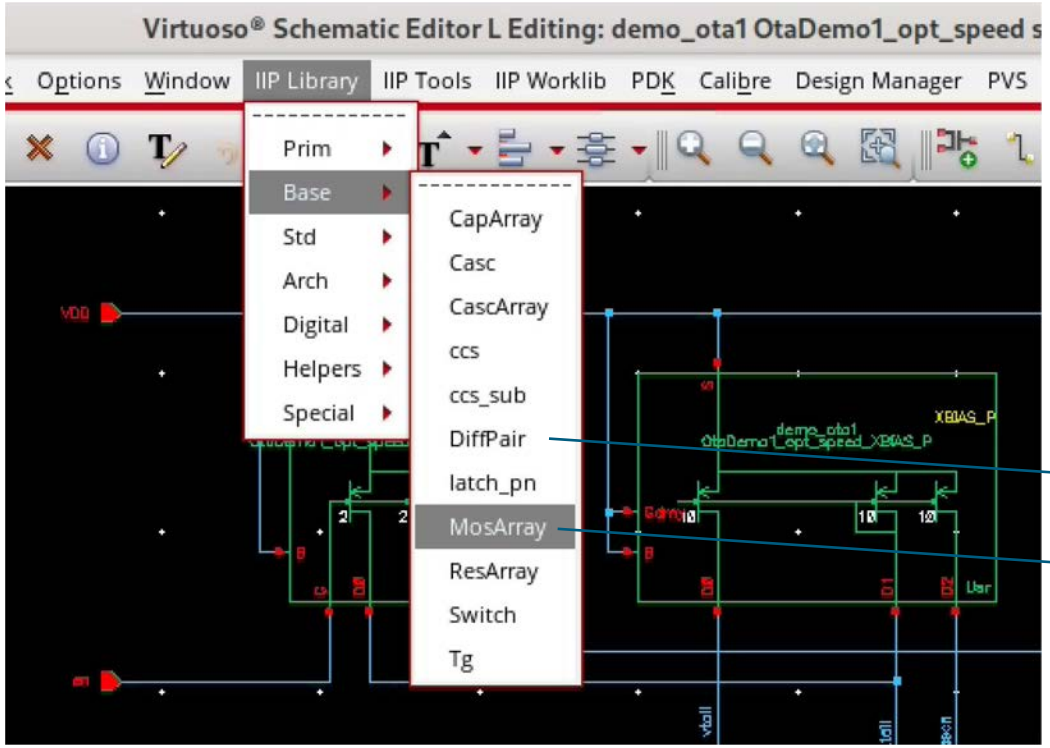
© Fraunhofer IIS

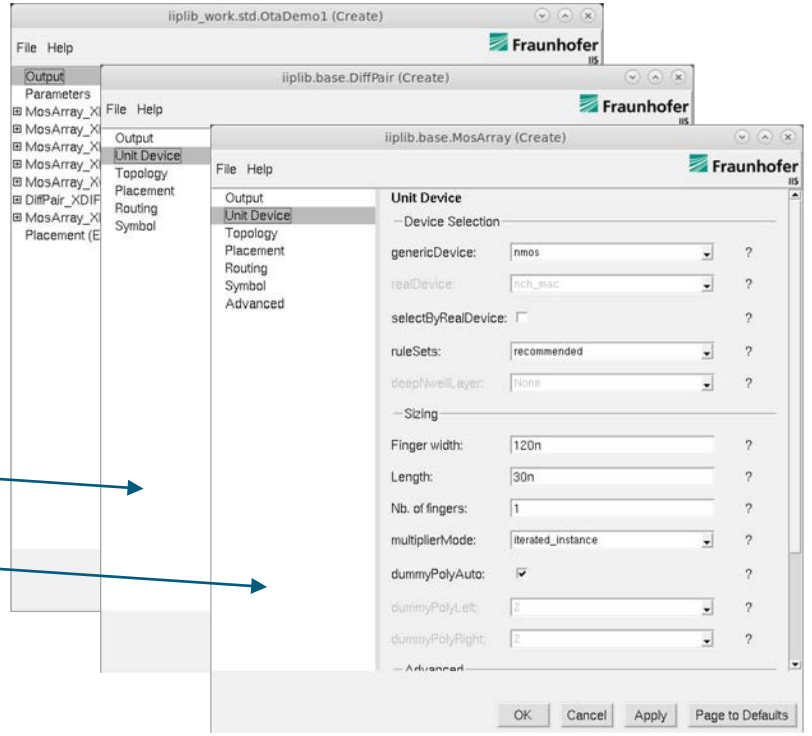
Restricted

Fraunhofer IIS

What is a Generator?

- specific for a type of circuits → also called an IIP (a kind of analog soft IP)

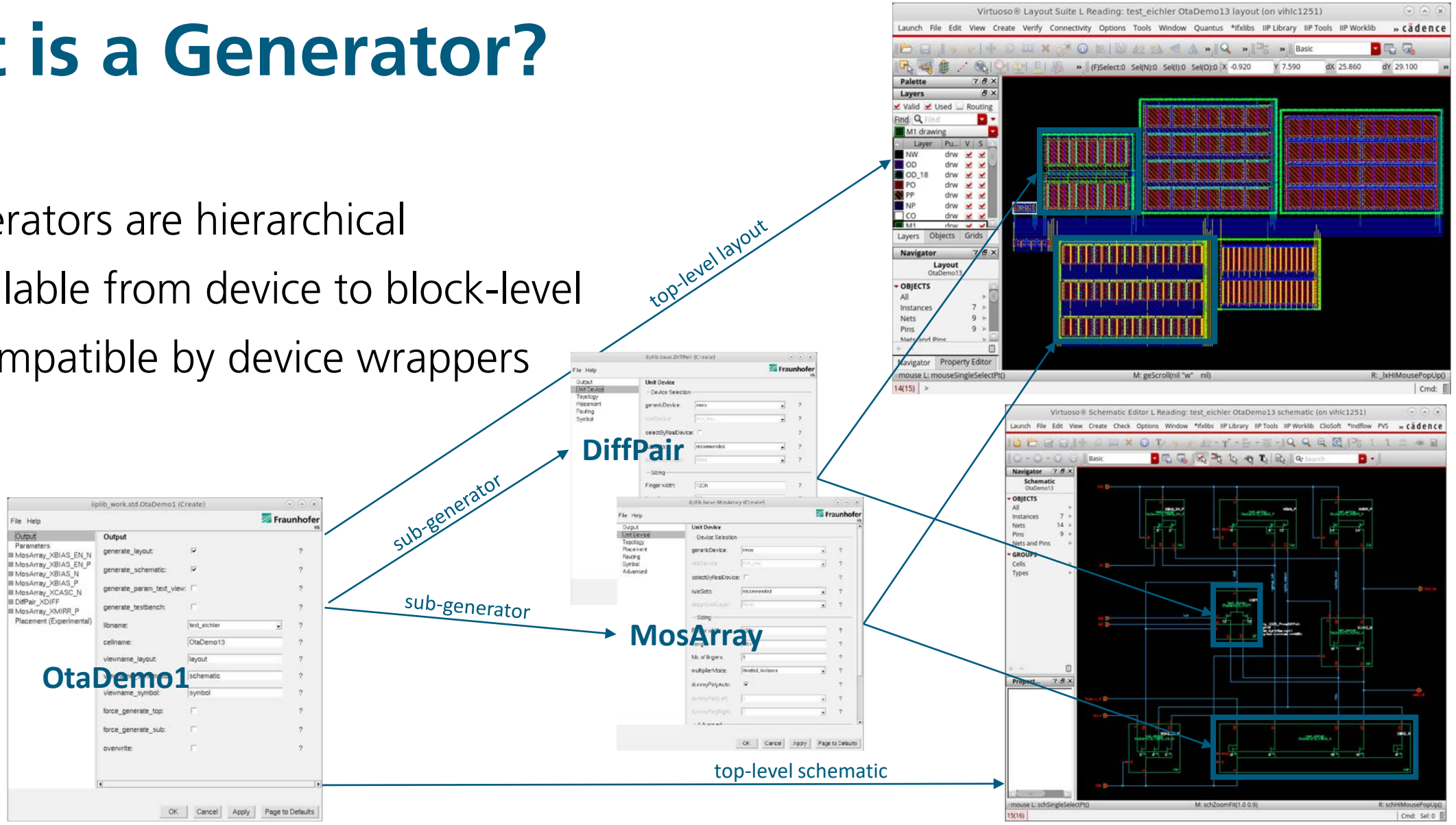




© Fraunhofer IIS
Restricted

What is a Generator?

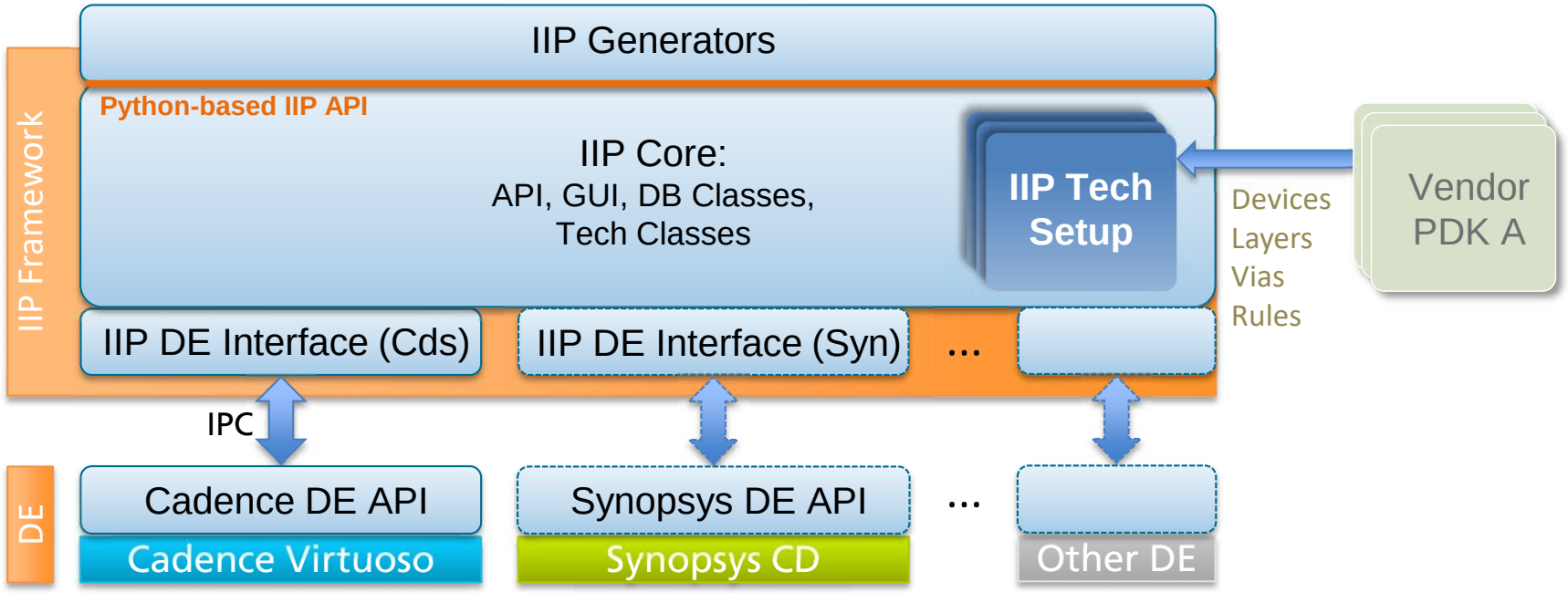
- IIP generators are hierarchical
- IIPs available from device to block-level
- PDK-compatible by device wrappers



© Fraunhofer IIS

What is a Generator?

- implemented using a generic circuit description language based on Python



© Fraunhofer IIS
Restricted

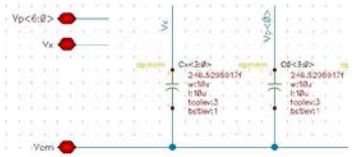
What is a Generator?

- Our usual generator code structure

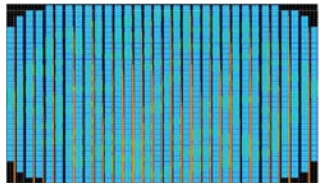
```
1. class Generator(iip.gen.HierBlock):
2.   def param_spec(self):
3.     self.params.add(…)
4.   def param_check(self):
5.     if self.params.l.v > self.params.w.v: …
6.   def symbol(self, cv):
7.     cv.create_rect(…)
8.   def schematic(self, cv):
9.     cv.create_instance(…)
10.  def layout(self, cv):
11.    cv.create_shape(…)
12.    self.template.draw(cv)
```

Parameters

Schematic



Layout



[B. Prautsch et al., "IIP Framework: A Tool for Reuse-Centric Analog Circuit Design," SMACD 2016, pp. 1-4, 2016.]

© Fraunhofer IIS

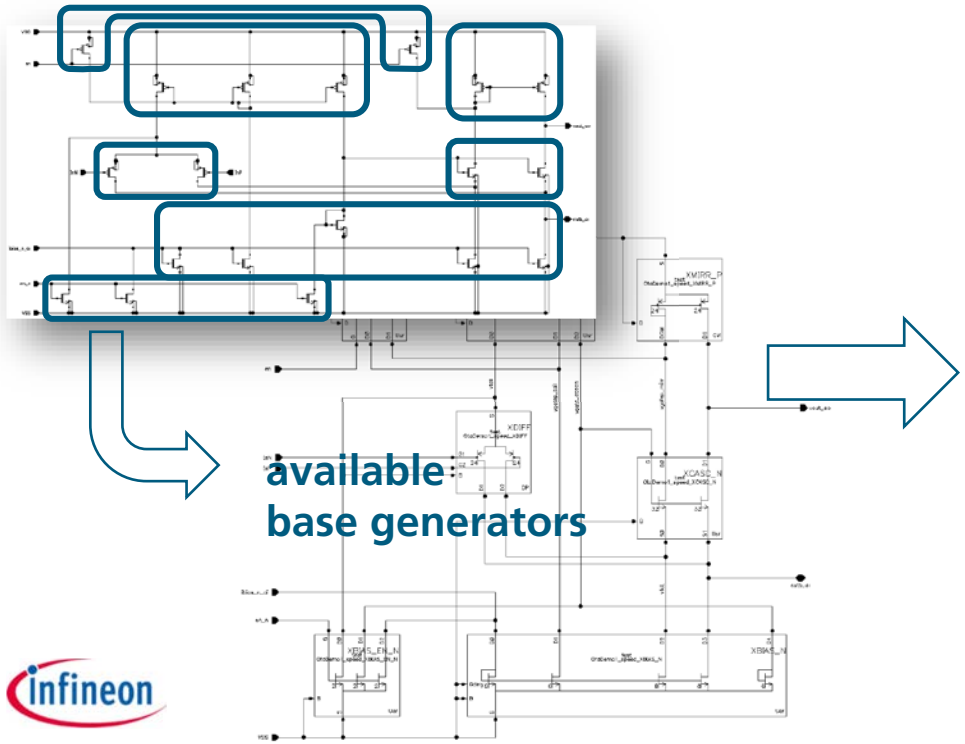
How to reuse designs by generator IPs?

- Use **available base generators**
  - already improves productivity for most common building blocks
  - but still manual effort for next hierarchies
- **Custom**, circuit-specific **generator** implementation
  - large implementation effort for flexible, area-efficient layout
- Generator using flexible **layout templates**
  - templates define relative positions of sub-block instances and interconnects
  - usually regular arrangements, reusable across hierarchies
  - faster **generator** implementation / **creation** as P&R code already part of API

© Fraunhofer IIS

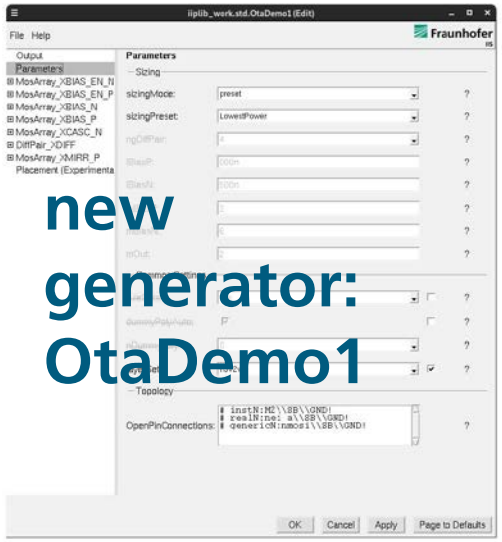
Automatic creation of generators

- Use IIP Creator to translate available schematic data into a new generator



IIP Creator

Code generation



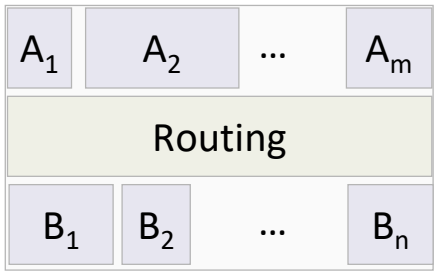
new generator: OtaDemo1

orig. OTA design by Infineon [F. Passerini et al., ESSCIRC/ESSDERC, 2021]

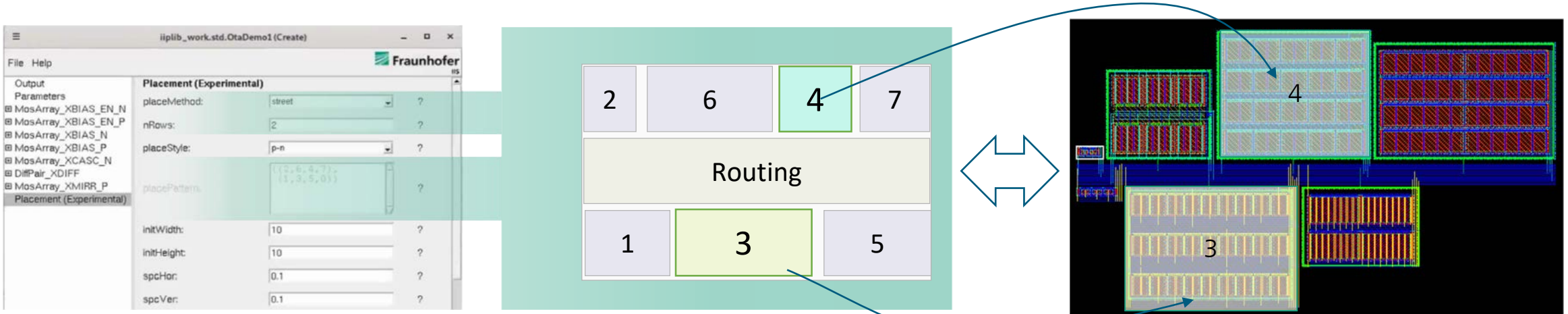
© Fraunhofer IIS

Template-based layout generation

- Template P&R algorithms were embedded into IIP
- Template properties accessible through user interface



Template „Street“



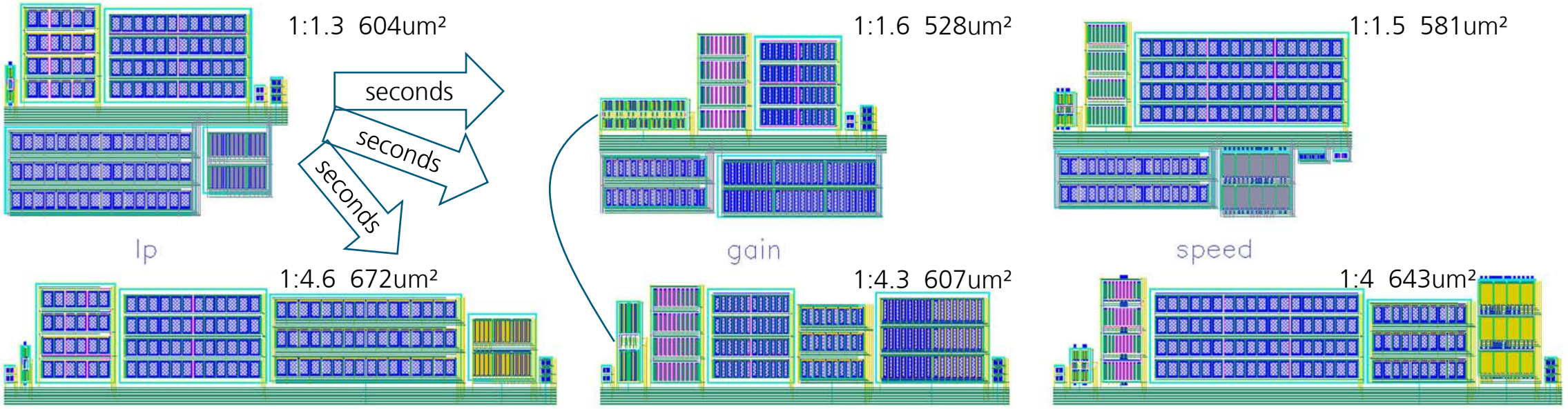
Generator GUI | Template | Generated layout

Template idea from: [R. Castro-López et al., "An Integrated Layout-Synthesis Approach for Analog ICs," IEEE TCAD, 2008]

© Fraunhofer IIS Restricted

Generated Layout Variants

- 3 sizing variants found by simulations
- 2 layout variants each based on street template



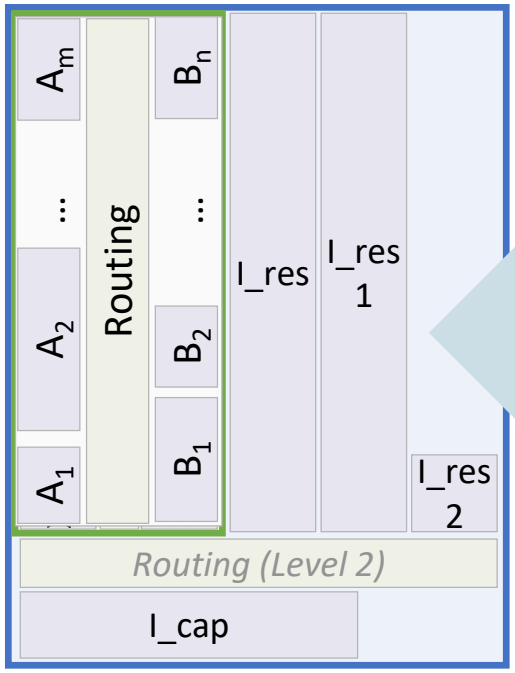
Low power | High gain | High speed

© Fraunhofer IIS Restricted

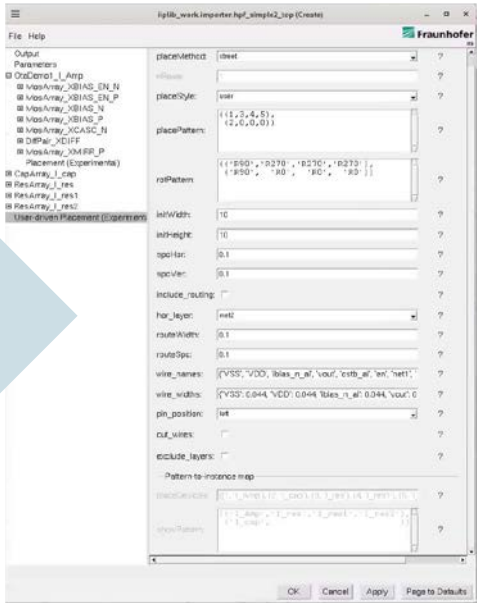
IIP Generators in Analog Design

More Flexible Reuse by using Templates – Hierarchical Approach

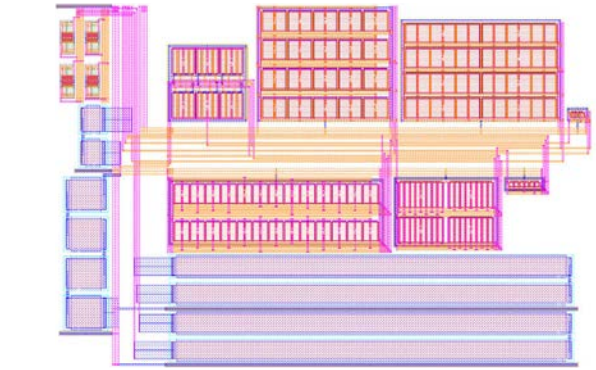
Hierarchical Templates: Variant Generation in a Matter of few Minutes with no Programming Required to Get Such (Layout) Generator

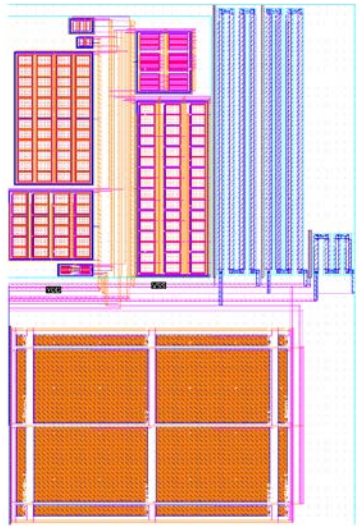


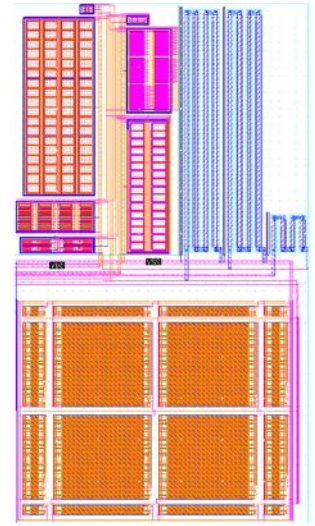
User-driven Placement



Variants generated in ~2 min







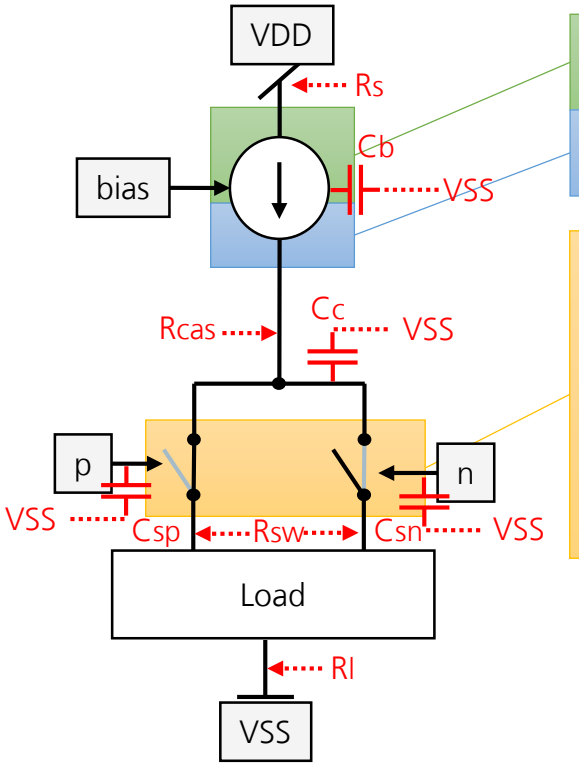
[B. Prautsch, U. Eichler, and U. Hatnik, "**Generating the Generator: A User-Driven and Template-Based Approach towards Analog Layout Automation**" Electronics 12, no. 4: 1047, 2023, https://doi.org/10.3390/electronics12041047]

© Fraunhofer IIS

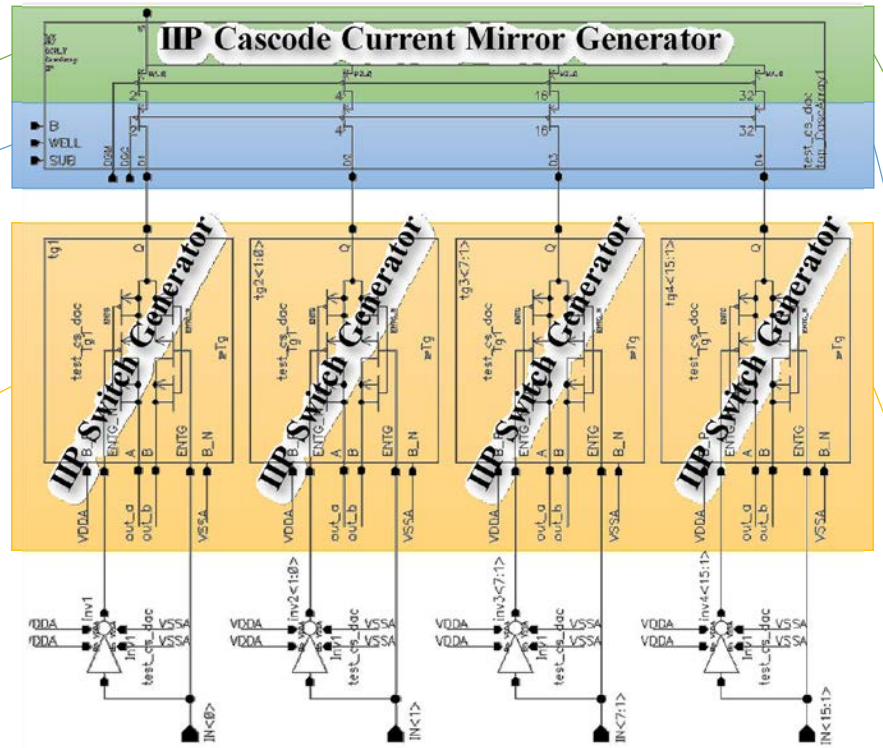
IIP Generators for HoLoDEC partners

Generator Concept for a Current-Steering DAC (cooperation with IMST)

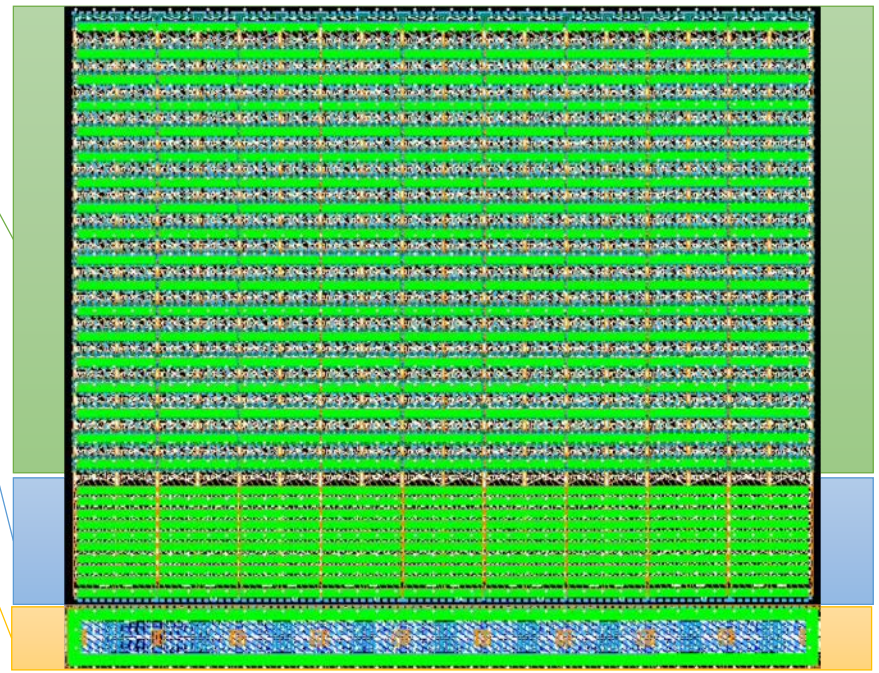
CS-DAC block diagram with critical parasitics



CS-DAC schematic using IIP blocks



CS-DAC layout prepared for generator implementation



[R. Wittmann, B. Prautsch et al., "**Design Flow Concept for Power-minimized Wireless RF Edge-Computing Sensor Transceivers**" submitted for IEEE conference SMACD 2025]

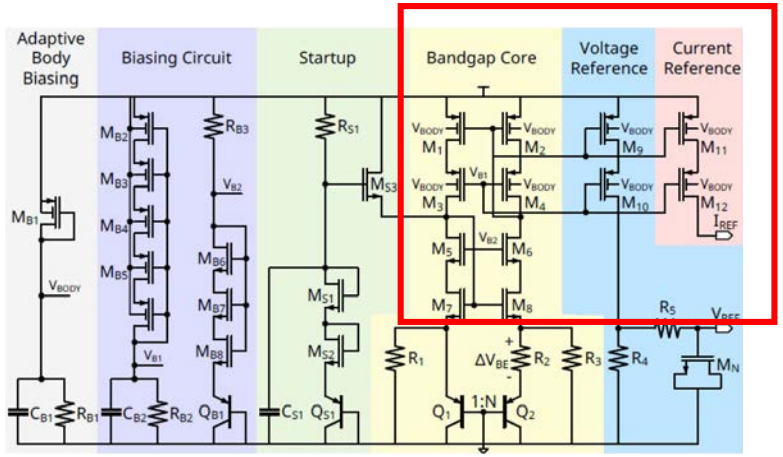
© Fraunhofer IIS

IIP Generators for HoLoDEC partners

Low Voltage Bandgap reference in GF 22 FDX (cooperation with TU Braunschweig)

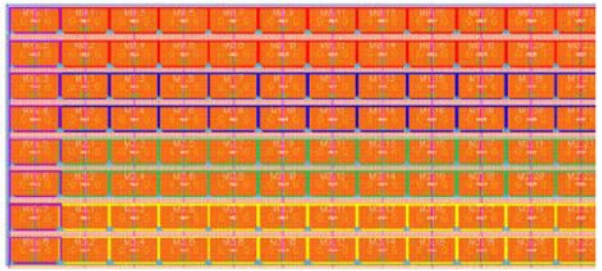
Bandgap Schematic

by IIP

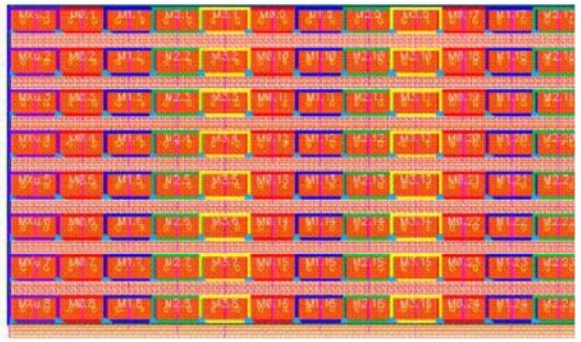


Investigated layout variants for $M_{1,2,9,11}$

a)



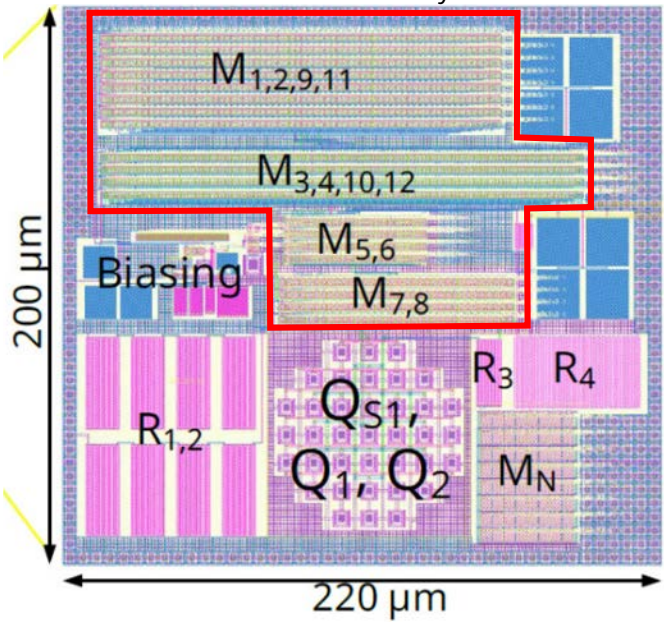
b)



▪ ▪ ▪

Top-level layout of the bandgap

by IIP



[A. Dossanov, Z. Guo, U. Eichler, B. Prautsch, V. Issakov, "**Low Voltage Bandgap Reference using Intelligent Layout Generators in 22 nm FDSOI CMOS**" accepted for IEEE conference SMACD 2025]

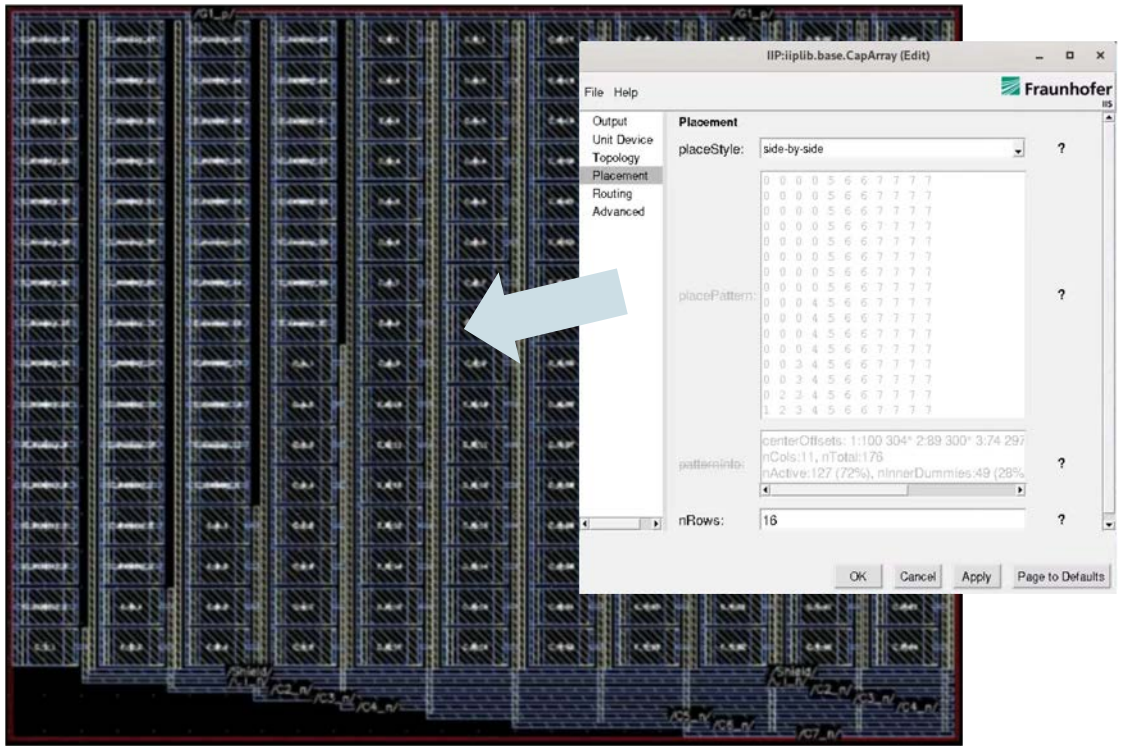
© Fraunhofer IIS

IIP Generators for HoLoDEC partners

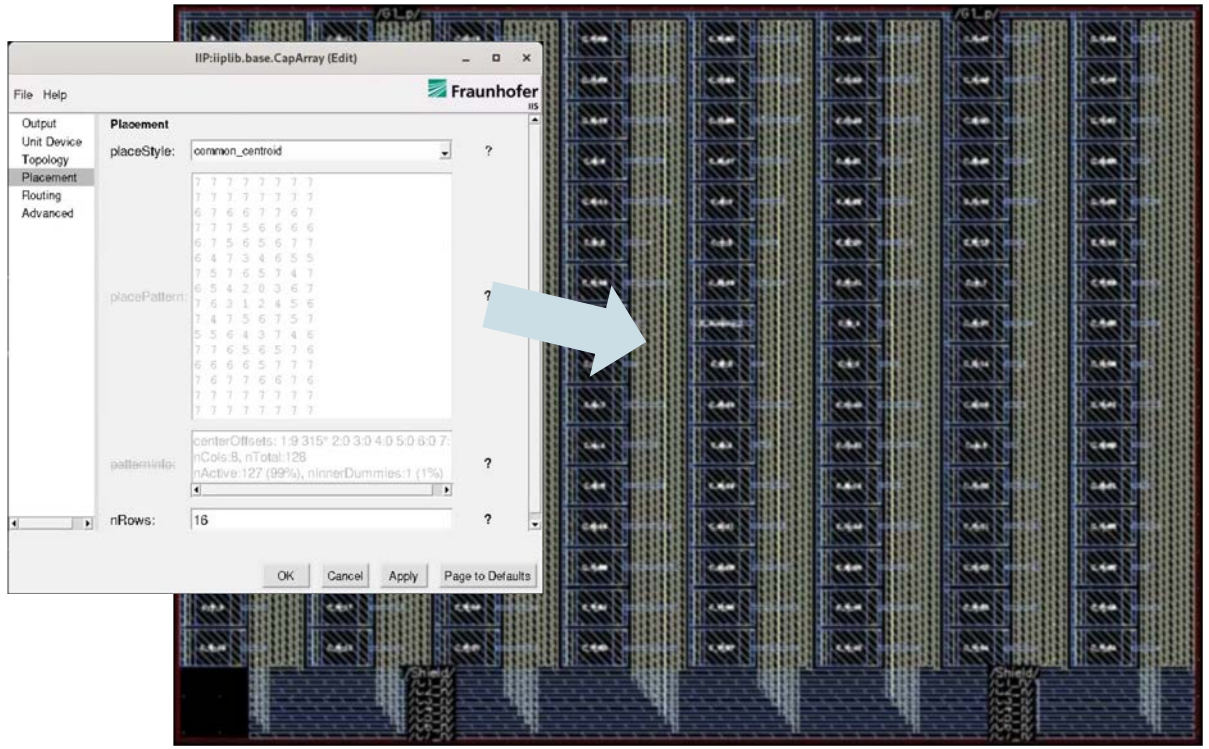
MosArray and CapArray IIP blocks for OpAmp and SAR-ADC (cooperation with IMMS)

CapArray example variants:

binary weighted, side-by-side, proportional routing



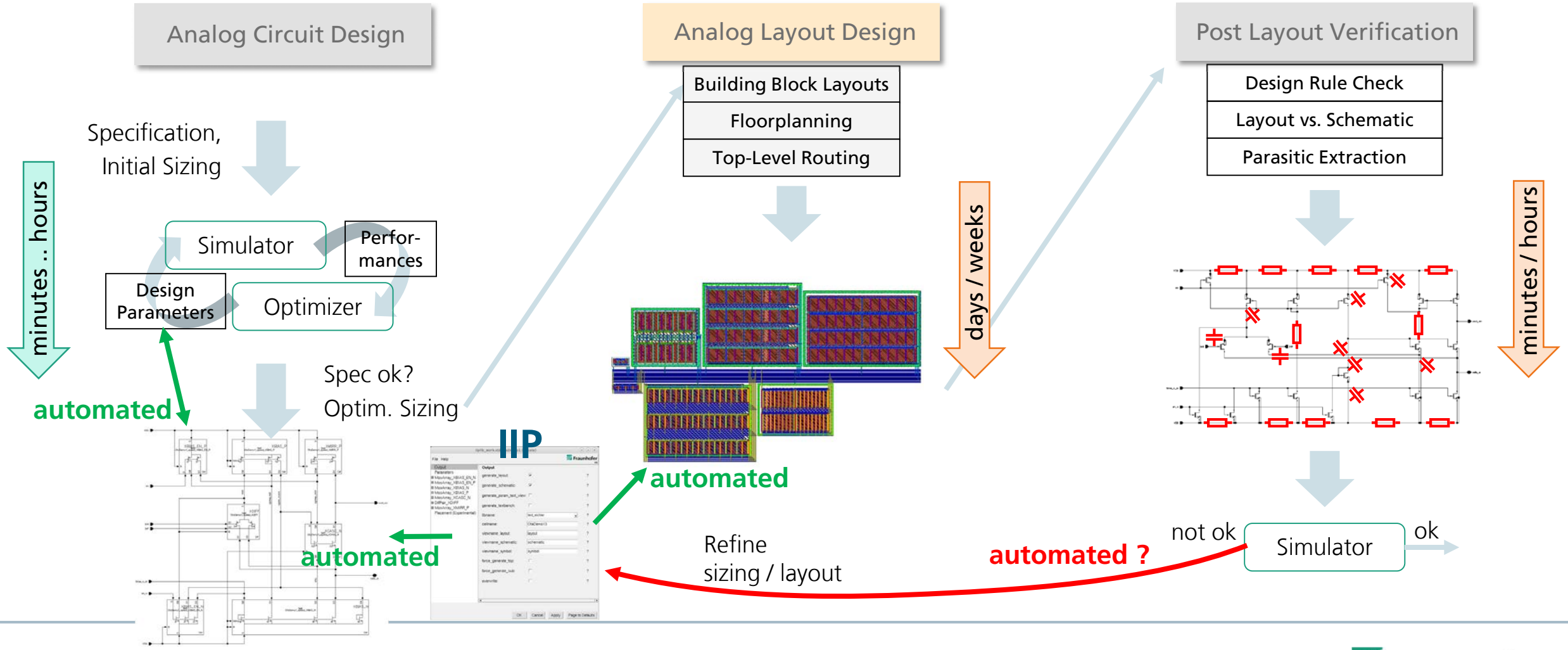
common-centroid, filled routing, shielding



© Fraunhofer IIS
Restricted

Relevance and Challenge of Analog IC Design

The Challenge of Analog Design – Few Layout Terms

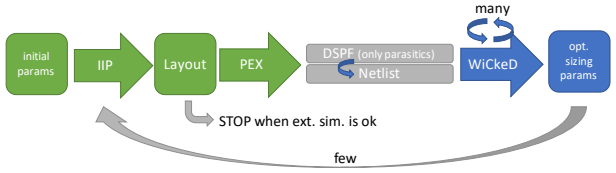


© Fraunhofer IIS
Restricted

How to consider parasitic data with IIP?

- 1) Show parasitic influences to the user
  - Display early parasitic estimates in IIP

- 2) Parasitic-aware simulation/optimization
  - Iteratively include extracted parasitics in original netlist

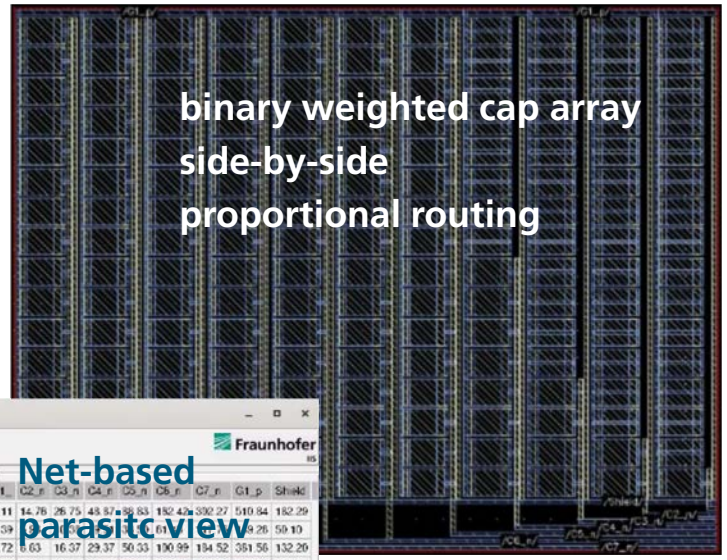


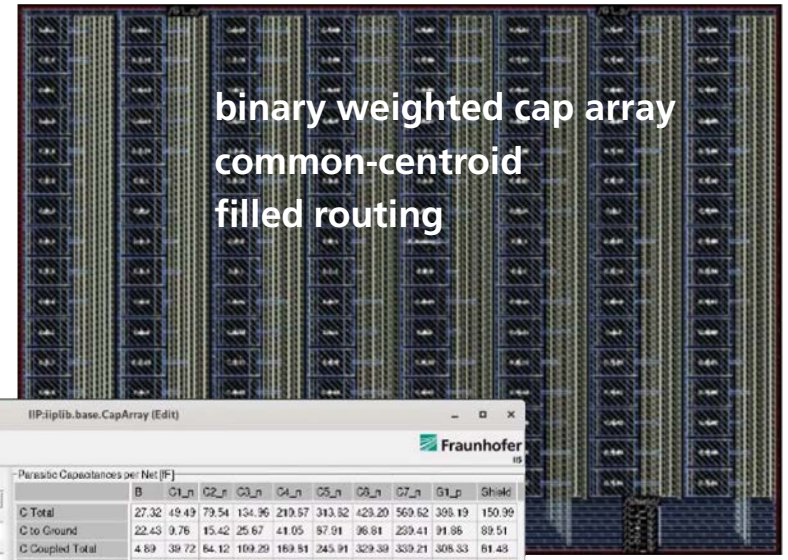
- 3) Parasitic- and layout-aware optimization

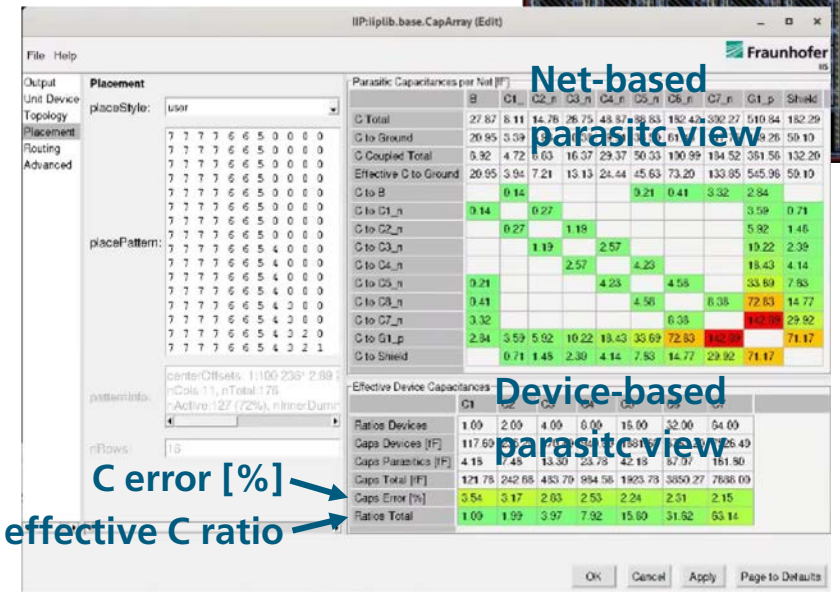
© Fraunhofer IIS
Restricted

1) Display early parasitic estimates in IIP

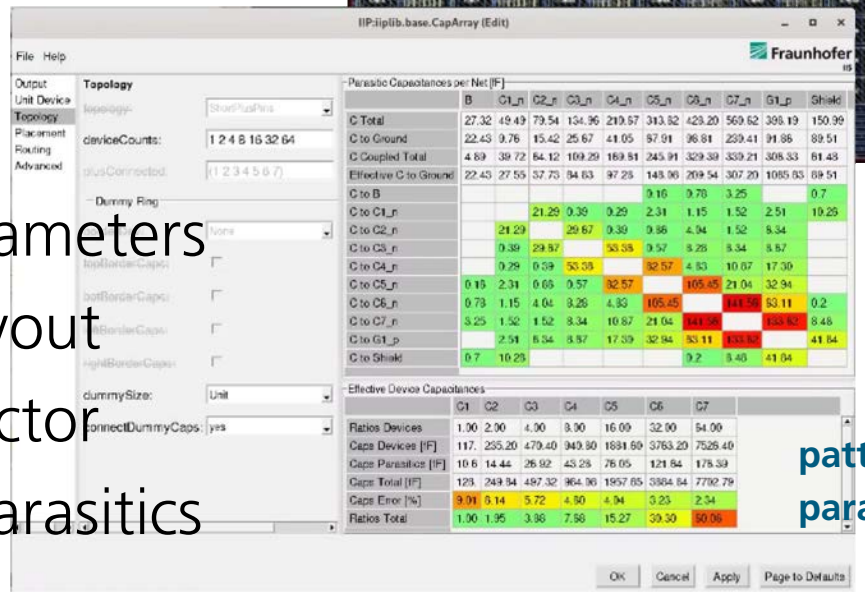
- A) based on external extractor







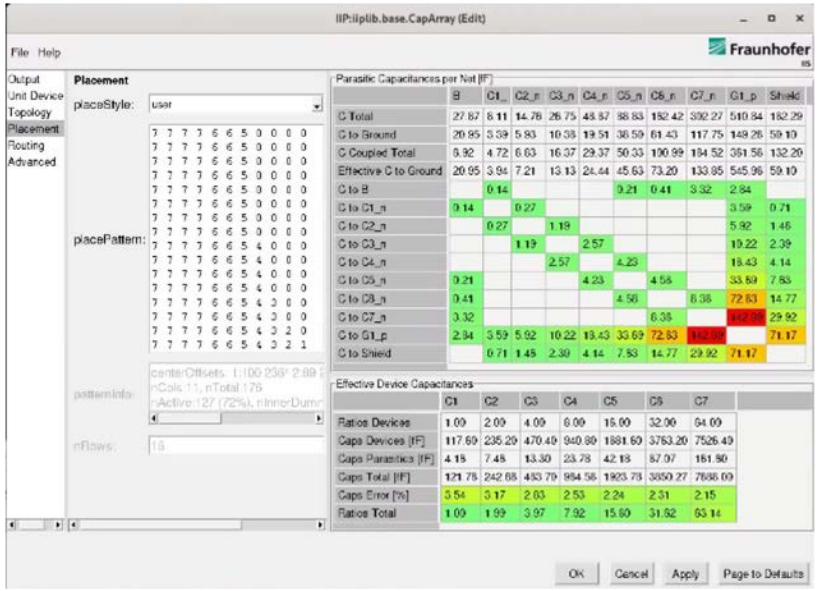
- user sets parameters
- IIP creates layout
- IIP calls extractor
- IIP displays parasitics

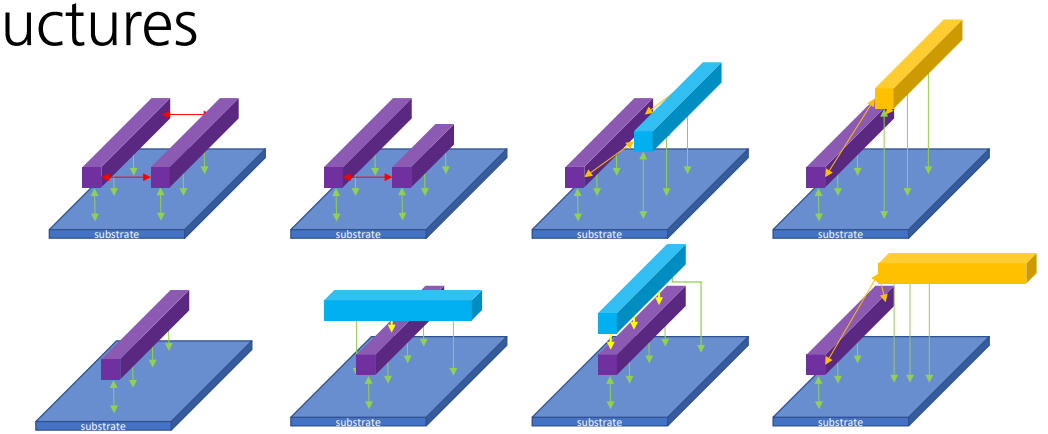


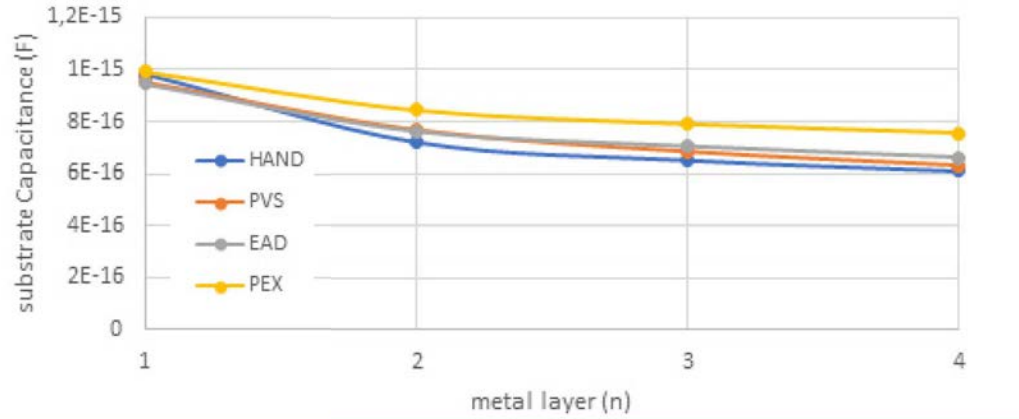
© Fraunhofer IIS Restricted

1) Display early parasitic estimates in IIP

- B) based on internal estimation for basic structures
  - Does not require layout creation → fast
  - (in progress)



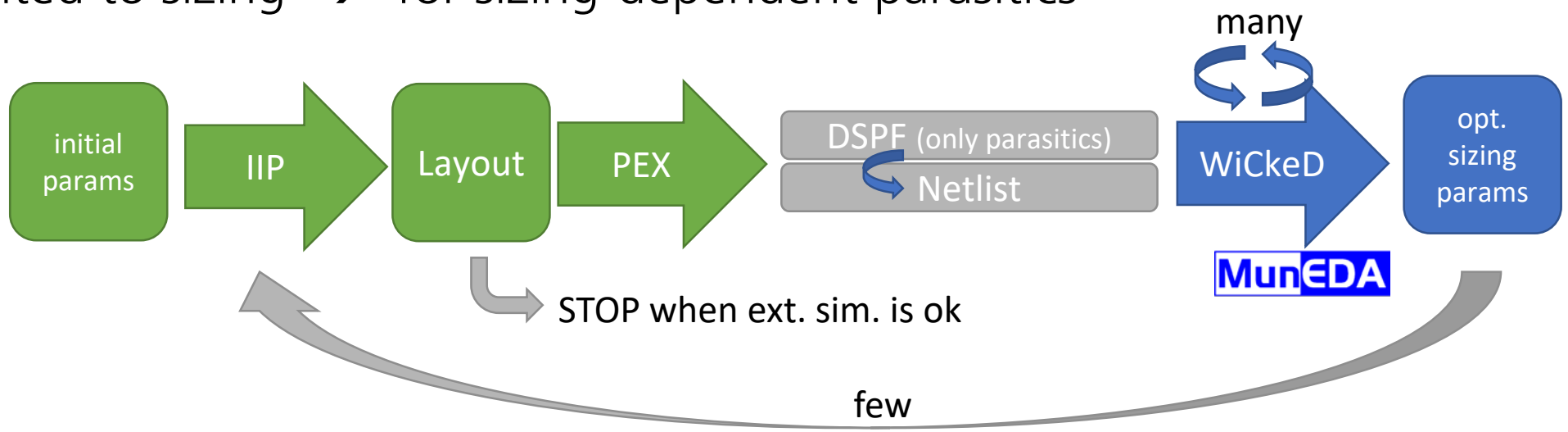




© Fraunhofer IIS Restricted

2) Parasitic-aware simulation/optimization

- Iteratively include extracted parasitics in original netlist
  - static parasitic data backannotated into netlist before each optimization run
  - iterate outer loop until parameters are settled
  - limited to sizing → for sizing-dependent parasitics



© Fraunhofer IIS
Restricted

3) Parasitic and Layout-aware optimization

Remaining slides available on demand – please ask.

© Fraunhofer IIS
Restricted

Summary

- Intelligent IP generators enable a customized automation of design and migration of analog IC building blocks
- Automatic creation of template-based generators may close the gap between available basic IIP building blocks and designer-driven design reuse and synthesis for more complex circuits
- Early layout availability enables early parasitic analysis, layout-aware sizing and even layout optimization

© Fraunhofer IIS
Restricted

Outlook

- Investigate scalability and validate the parasitic and layout optimization approach
- Combine IIP layout generation with flow-based generators such as EDP
- Open the Intelligent IP generator API for advanced users

- Keep and extend cooperation with our current and future project partners

© Fraunhofer IIS

Kontakt

Dr.-Ing. Benjamin Prautsch
Gruppenleiter Mixed-Signal Automation
Phone +49 351 45691-280
Mobil: +49 170-5917 514
benjamin.prautsch@iis.fraunhofer.de

Fraunhofer-Institut für Integrierte Schaltungen IIS
Institutsteil Entwicklung Adaptiver Systeme EAS
Münchner Straße 16
01187 Dresden
www.eas.iis.fraunhofer.de

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung