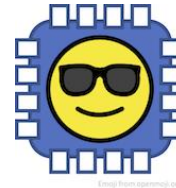


# Design and Generation of High-Performance Transceivers

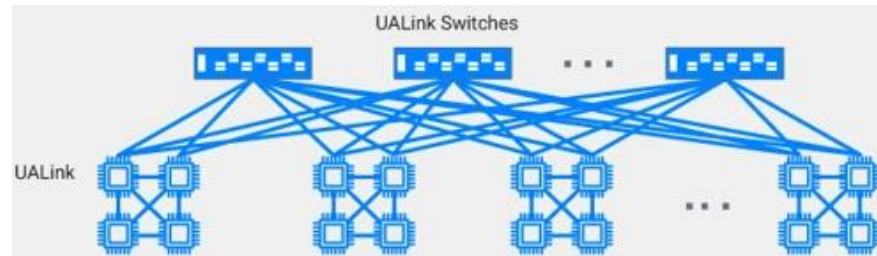
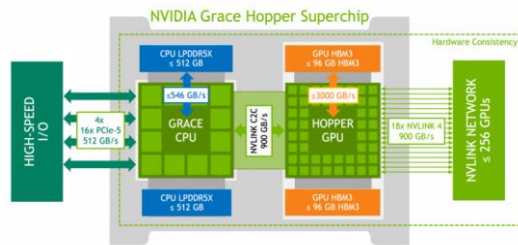
7. 4. 2025

Jaeduk Han

**Nifty Chips Laboratory @ HYU**  
[niftylab.github.io](https://niftylab.github.io)

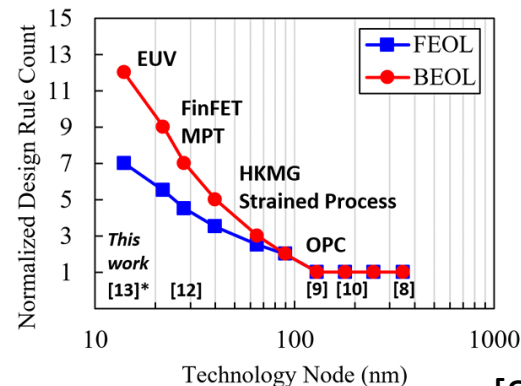
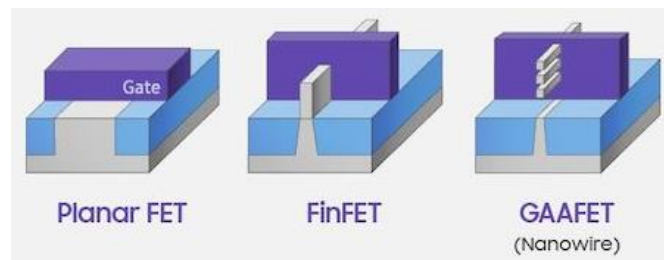


- **Bandwidth Scale-Up:** In the AI era, high-performance transceivers are essential to handle massive data, requiring over **100 Gb/s per channel**



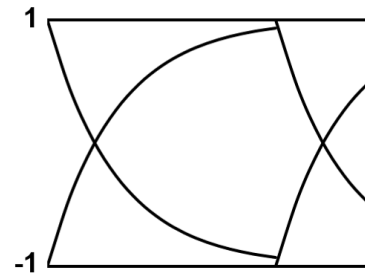
[NVIDIA, Synopsys]

- **Design Complexity:** Designing and fabricating these circuits in nanometer technologies require significant time and effort

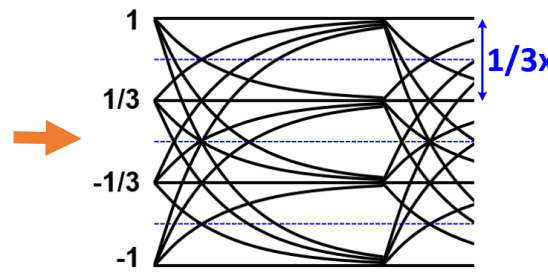


[Samsung, J. Han]

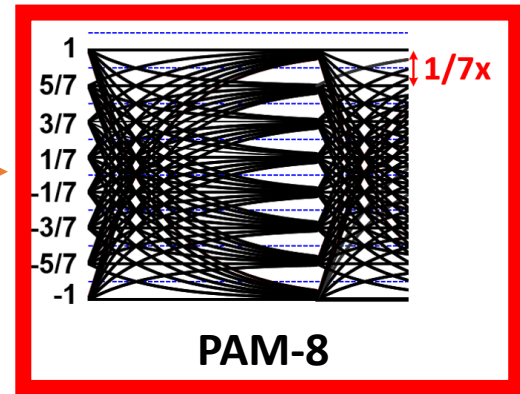
- Increase the number of transmitted bits per symbol



NRZ

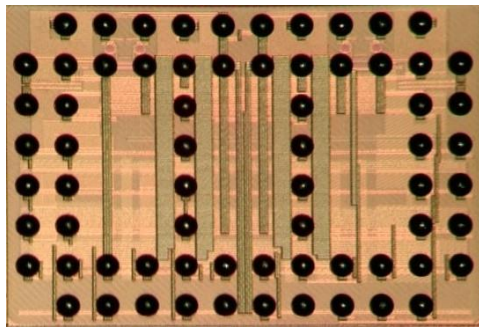


PAM-4

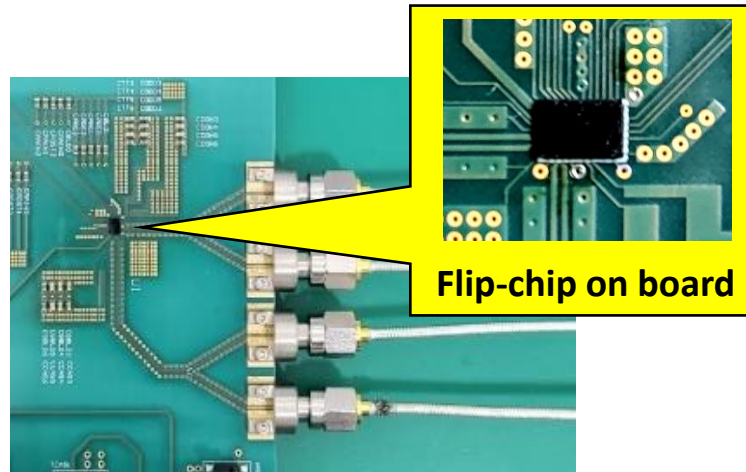


PAM-8

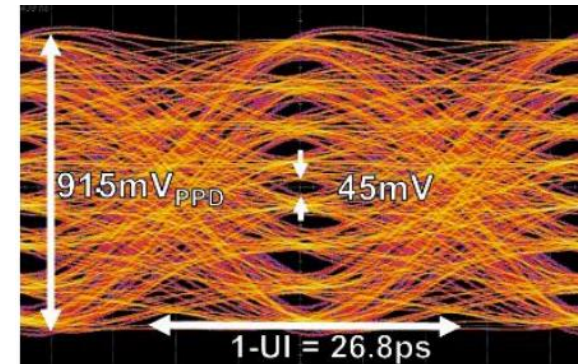
PAM Signal Evolutions



Chip Micrograph

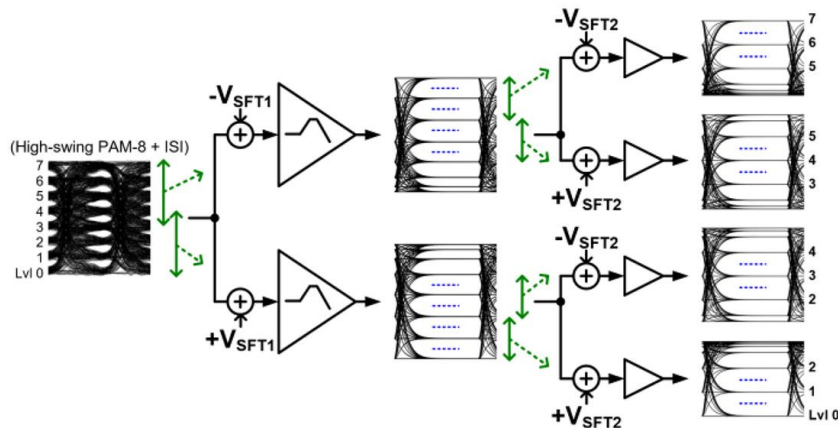


Measurement Setup

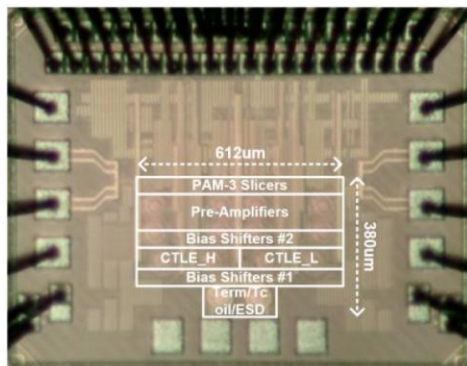


112Gb/s PAM8 Eye Diagram

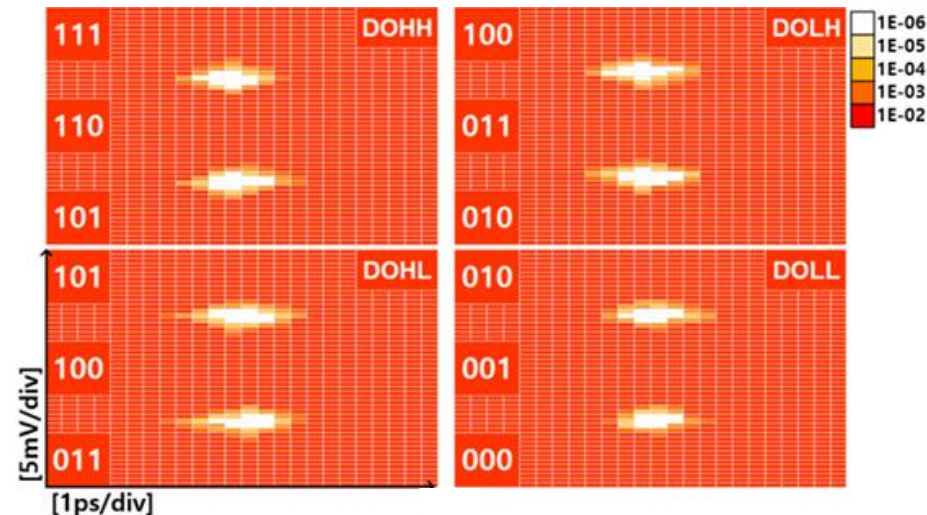
- A multi-path receive chain ensures high-fidelity recovery of PAM-8 signals, demonstrating **the highest PAM-8 mixed-signal data rate** ever reported



Receiver Architecture



Chip Micrograph



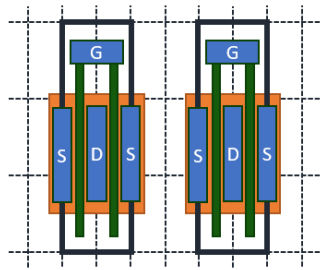
Measured Eye Diagrams at 103Gb/s



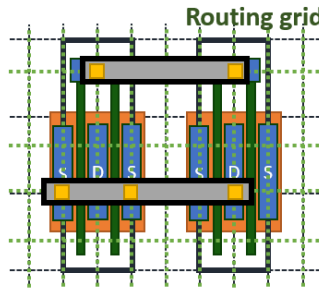
- Our framework, **LAYGO**, automates layout generation using templates and grids



Templates



On-grid placement

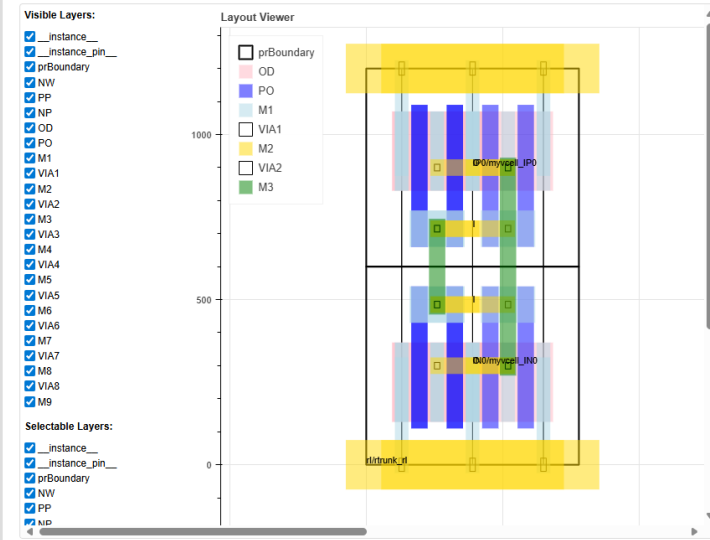


On-grid routing

```
Source Code      laygo2_tech.yaml

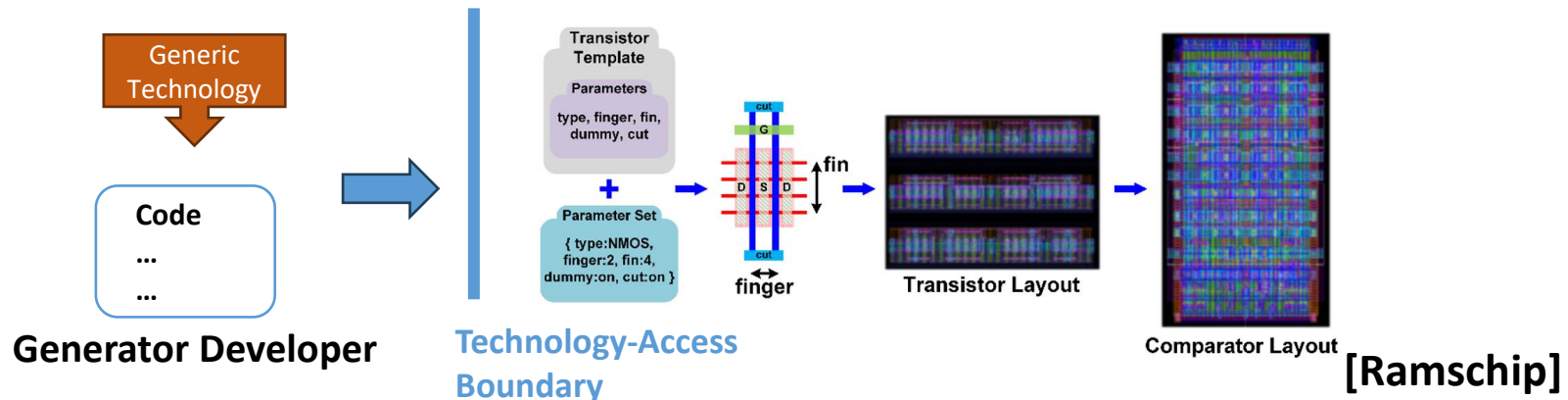
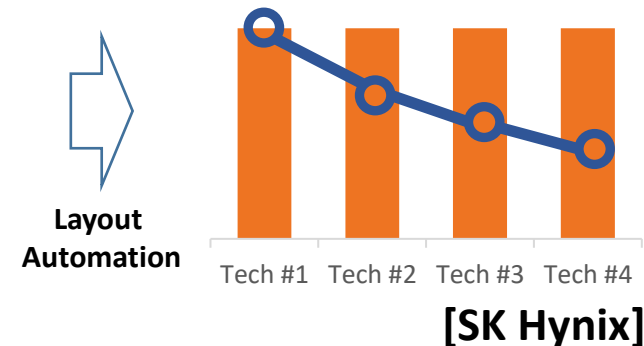
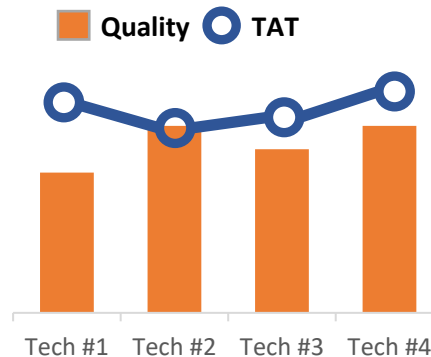
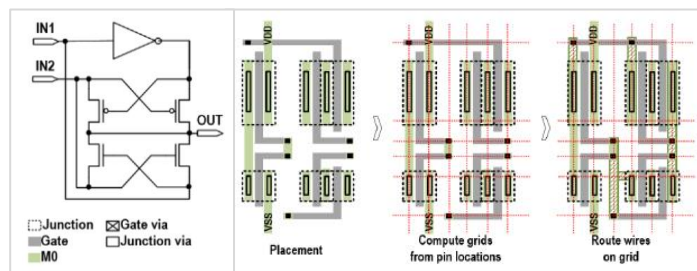
1 # The following imports are done automatically. No further imports are
2 # allowed.
3 #import laygo2
4 #import laygo2.tech as tech
5 #import numpy as np
6 # Design Parameters
7 libname = 'logic_generated'
8 cellname = 'test'
9
10 # Templates and Grids
11 templates, grids = tech.load_templates_and_grids()
12 pg, r23, r34 = grids['placement_basic'], grids['routing_23_cmos'],
13   grids['routing_34_cmos']
14
15 # Create a design and generate instances
16 dsn = laygo2.Design(name=cellname, libname=libname, pgrid=pg,
17   rgrid=r23)
18
19 # Place instances
20 dsn.place(inst=[(INO)], [(IPO)])
21
22 # Create and place wires
23 rc = laygo2.RoutingMeshTemplate(grid=dsn.rgrid)
24 rc.add_trunk(name='r0', index=[INO.p['D'].right[0], None], netname='r0')
25 rc.add_trunk(name='r1', index=[INO.p['G'].left[0], None], netname='r1')
26 rc.add_node(dsn.instances) # Add all instances to the routing mesh as
27   nodes
28 rc.add_node(dsn.virtual_instances) # Add all virtual instances to the
29   routing mesh as nodes
30 rc.generate_routes()
```

Run Code



A temporary website is open, please try! - <http://laygo.hanyang.ac.kr>

- Applied to sub-7nm FinFET, DRAM, and CIS, over **5x productivity improvement** is **proven**
- We're now integrating AI and LLMs—**stay tuned!**



A temporary website is open, please try! - <http://laygo.hanyang.ac.kr>