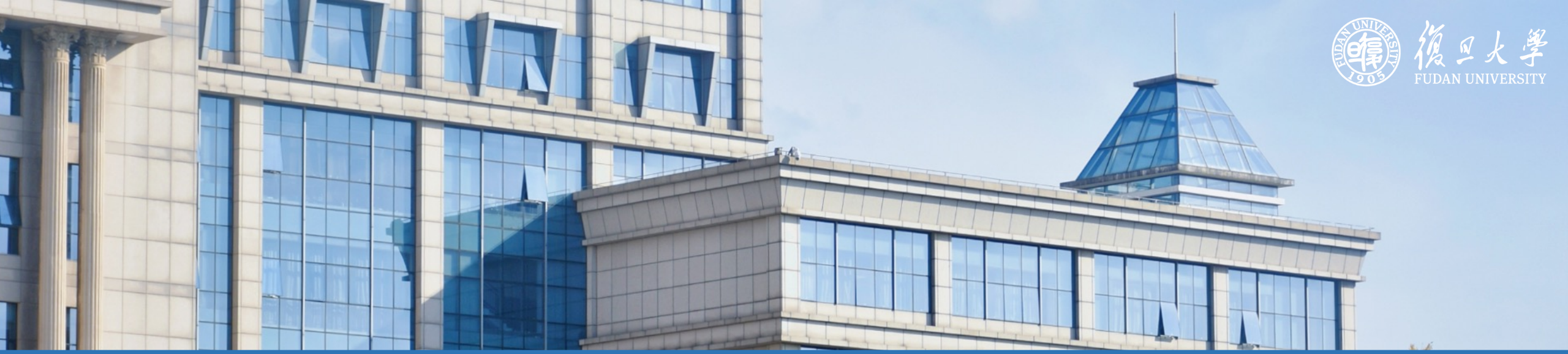


先进封装与集成芯片

Advanced Package and Integrated Chips

Lecture 10： Interconnect & Partition

Instructor: Chixiao Chen, Ph. D

Course Project

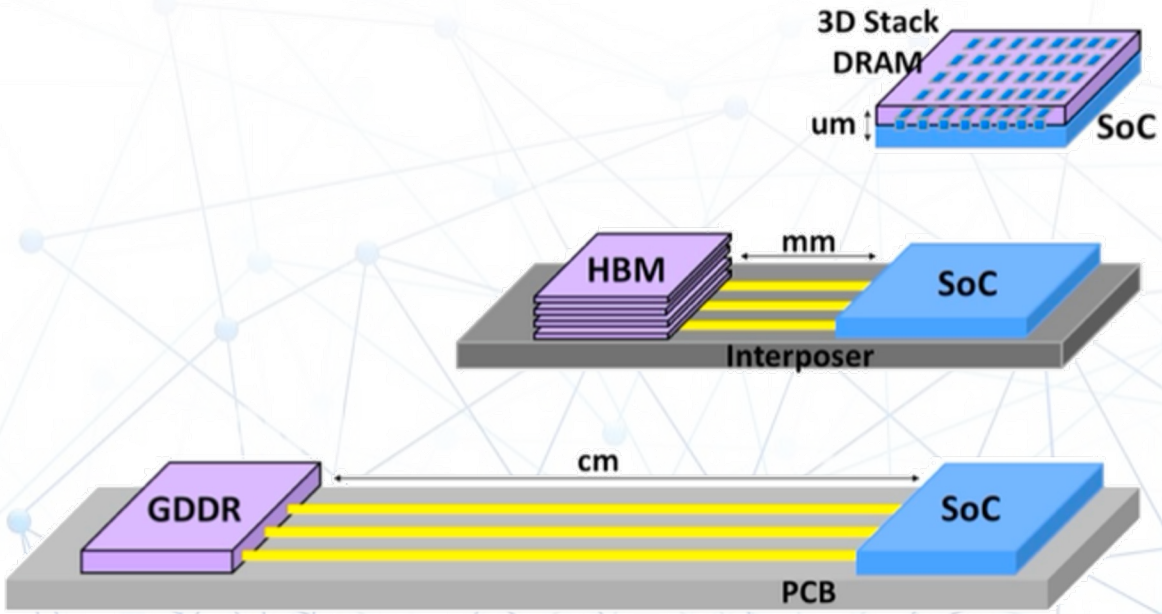
- Option I: Presentation (I guess most students would choose)
  - Pick up one paper from the pool (https://365.kdocs.cn/l/cpbY4B34cUnj first come first served), send a email or told me in wechat group, I will updated the paper selection in time.
  - If you want to pick a paper not listed, please send me an email to get granted.
  - Slides and Presentation: Prepare a 15min slides to introduce the work and 3-5 min Q&A. Send the slides to the homework mail after presentation. Presentation Day
- Option II: Project
  - Complete a component (AXI-streaming controller+PCS, PHY-TX/RX pair, Clock Generation,) design according to UCIe standard (not in a group)
  - Report and Presentation: Prepare a 5 min introduction and send the design report (<10 pages A4) to the homework mail, NO need for slides, show your draft report during presentation is OK.
- 1st Deadline: Tell me your choice by 5.6 on class / mail / wechat.

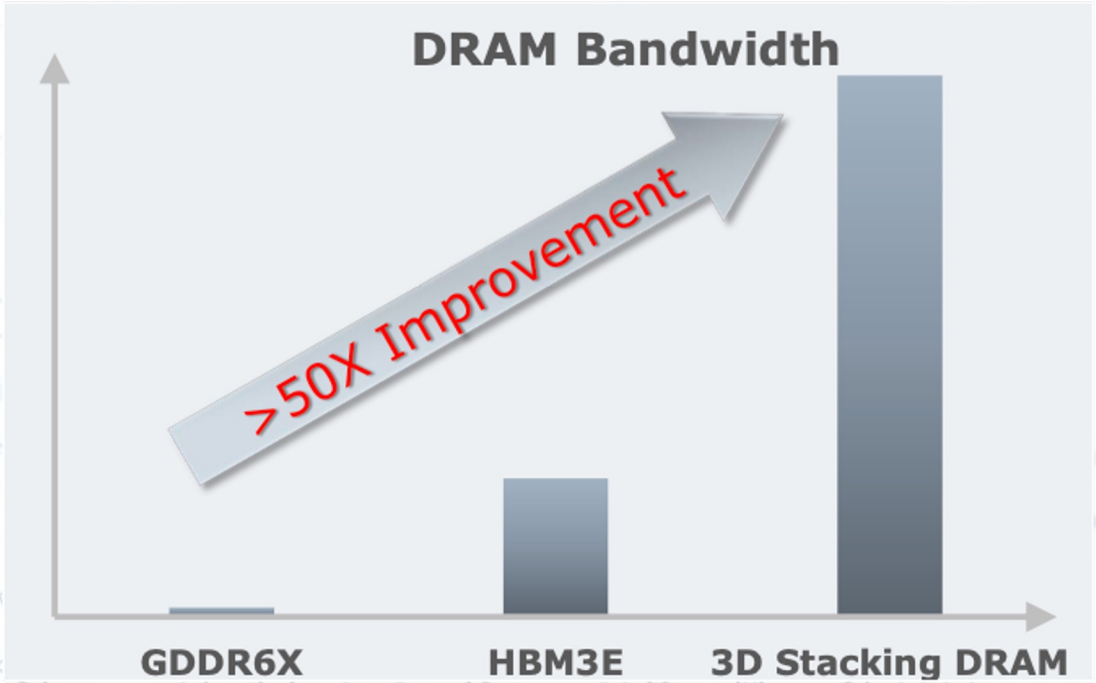
Overview

- From SoC Peripherals to Chiplet Interconnect
  - How advanced packaging affects System Performance
  - UCIe Brief Reivew
- Chiplet Partition Approach
  - 2D SoC → Chiplet Partition
  - 3D Soc Chiplet Partition

Memory Bandwidth Improvement

- C4 pitch: 130 um, CoWoS pitch: 40um, Hybrid Bonding Pitch: <10um
- Max Bandwidth density: HB can achieve 40-80x improvement than CoWoS
- Interconnect Energy Efficiency: 1/10 ~ 1/20 x less than CoWoS

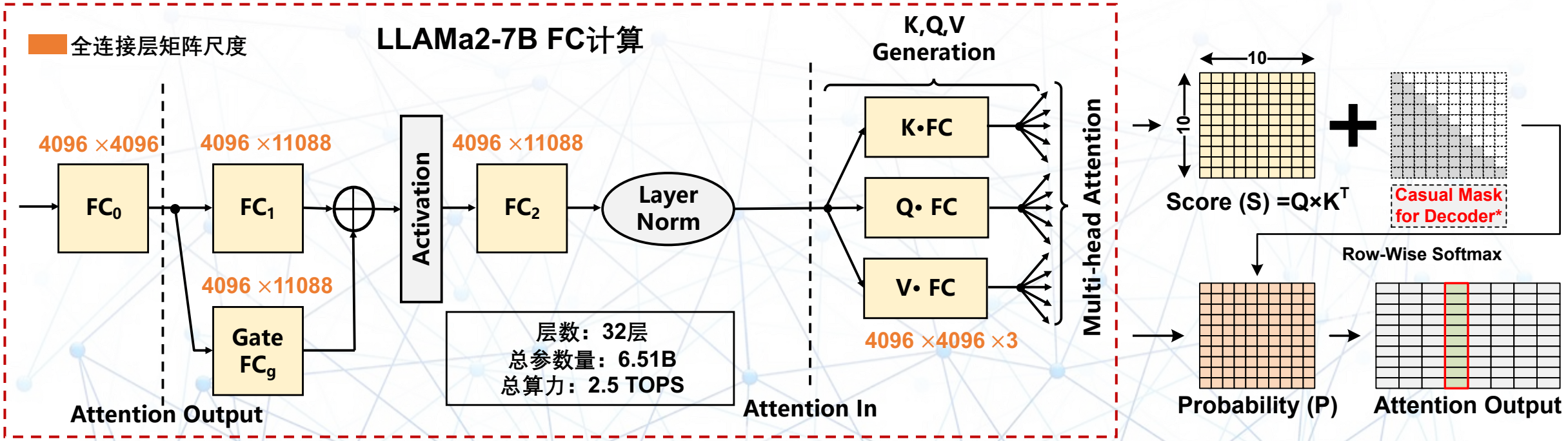




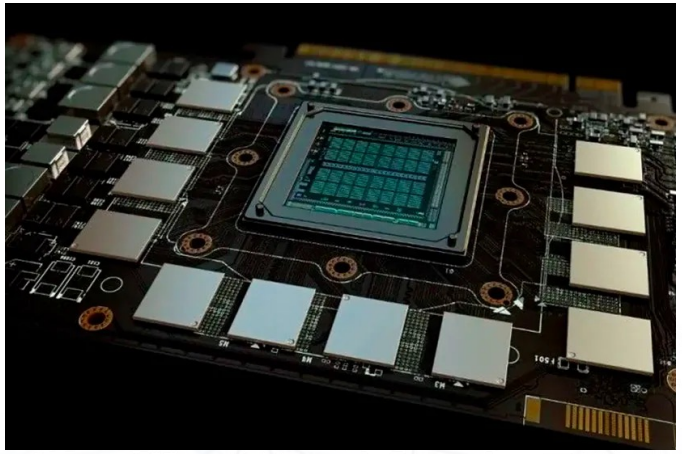
Source: SK Hynix, TSMC

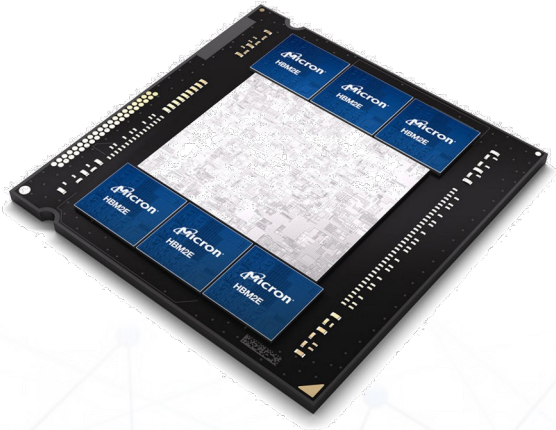
How advanced Packaging affects Interconnect?

- Given a 1466 TFLOPS @ FP8 AI processor with 64MB on-chip weight buffer storage (similar to Nvidia L40S), Please estimate the overall token rate when it is deployed to complete a 7-billion LLAMA-2 model inference (FP8). Assumed the overall model is buffered in (a) a 48GB GDDR6 external DRAM, (b) 80GB HBM2e .



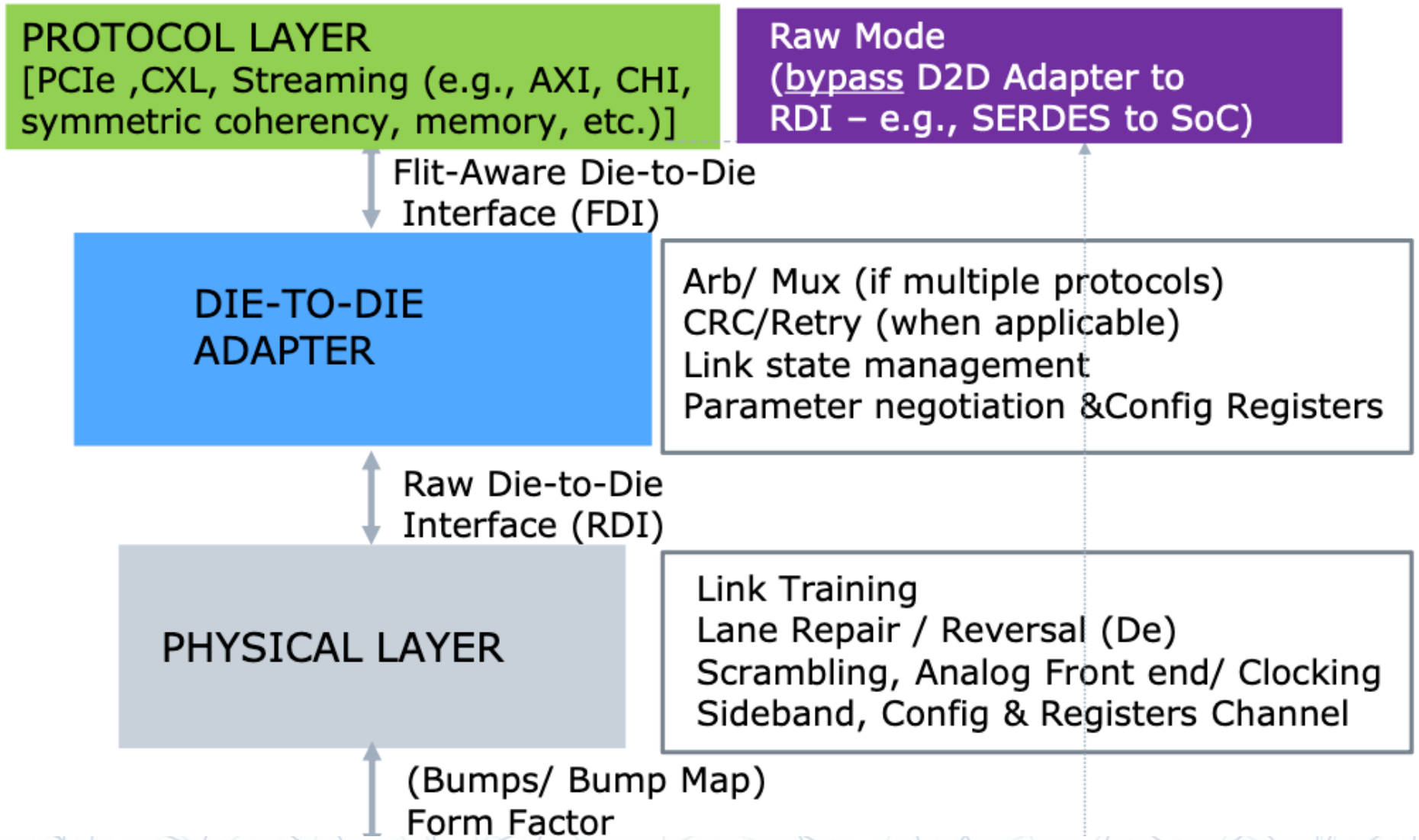
How advanced Packaging affects Interconnect?





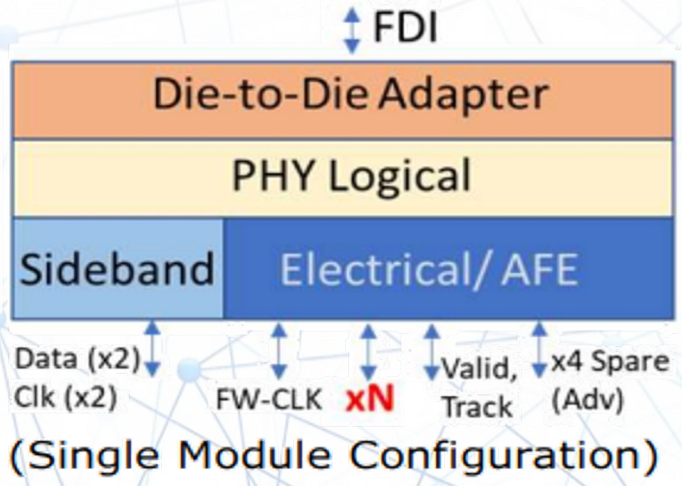
	GDDR 6 – 48GB	HBM 2E – 80GB
Per Pin Data Rate	16 Gbps	3.2Gbps
Data Pin Count Per Bank	32	1024
Per bank bandwidth	64 GBps	409.6 GBps
Bank Number	12	6
Overall Memory Bandwidth	864 GBps	~2040 GBps (with little loss)

Chiplet D2D Standard -- UCIe

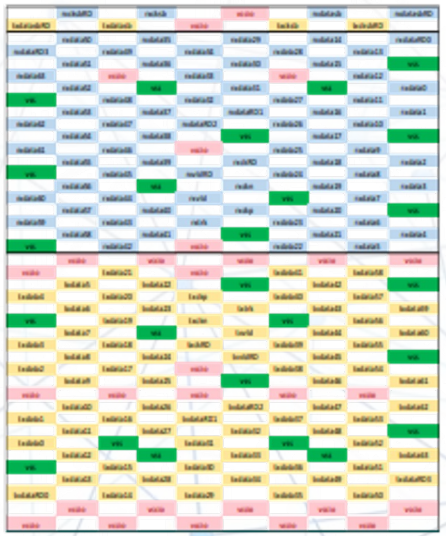


UCIe Physical Layer and Bump Map

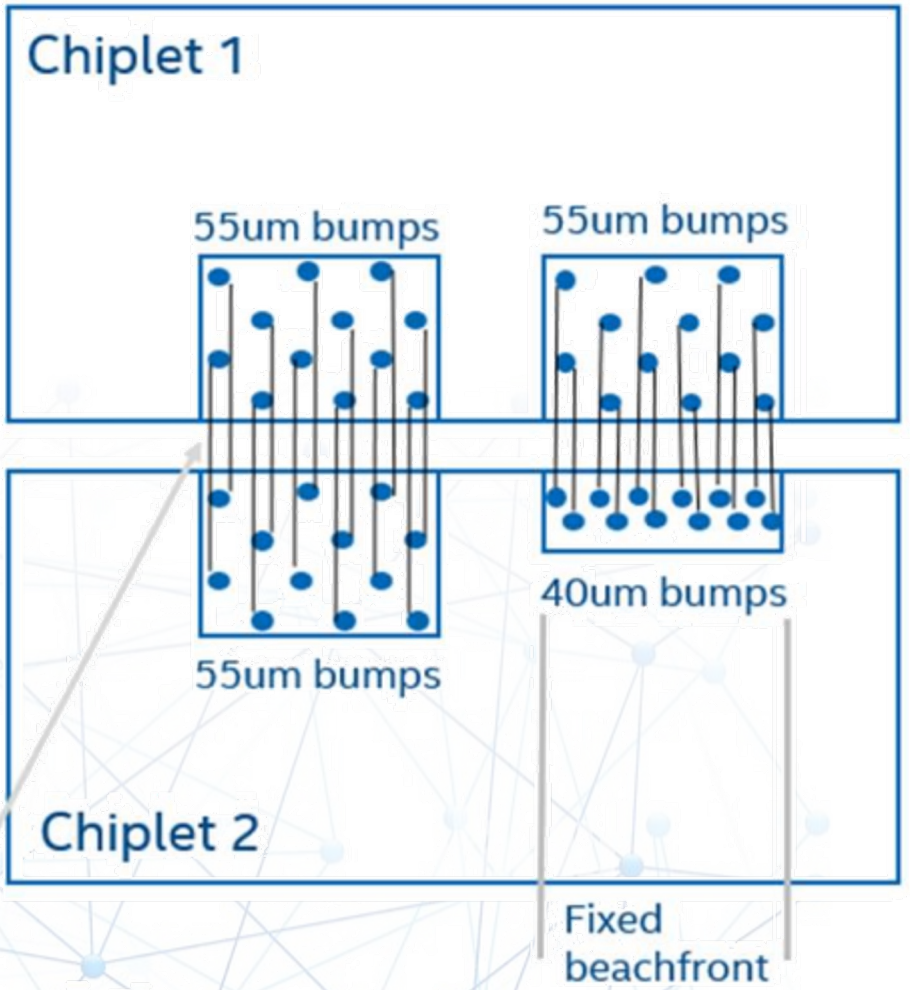
- Unit is One Module, support 1/2/4 module link
  - Standard (16 lanes) vs. Advanced (64 lances)
  - 1 valid, 1 track, 1 differential clk
  - Supported Frequency: 4/8/12/ 16/24/32 GHz
  - Side Band: 2 lanes/direction @800Mhz, data and clock



(Single Module Configuration)

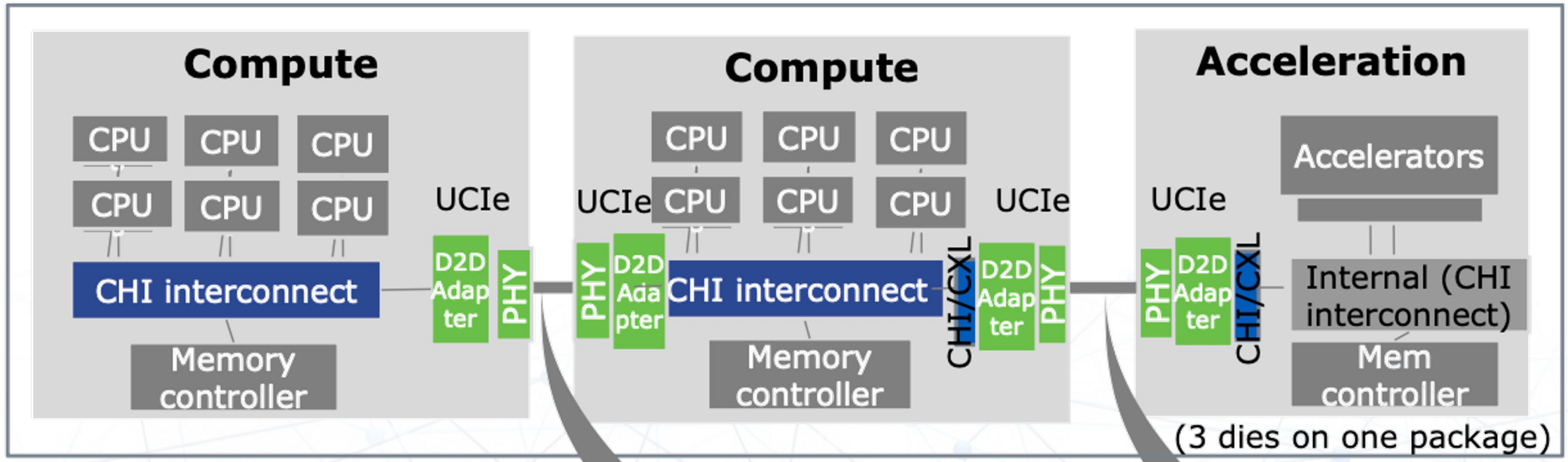

(UCIe-A Bump-out)

Fixed beachfront allows for Multi-generational compatibility As bump pitches decrease

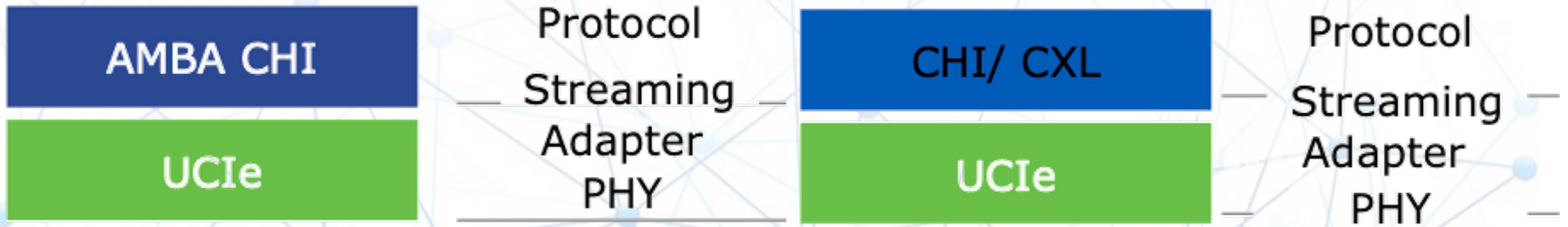


CoWos or EMIB or FoCoS or similar tight-pitch tech

UCIe-based Chiplet System

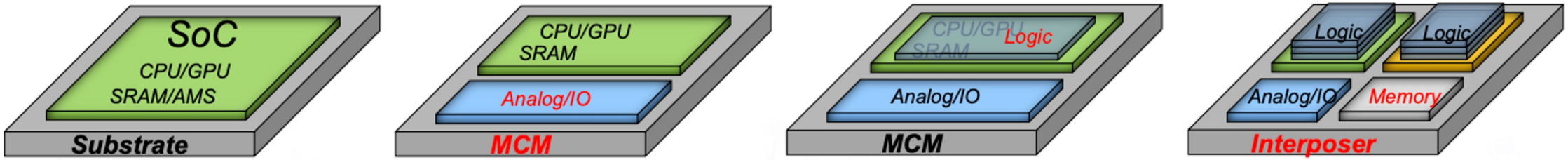


- Transporting same on-chip protocol allows seamless use of architecture specific features without protocol conversion
- Streaming interface with additional flit formats provide link robustness using UCIe defined data-link CRC & retry



Chiplet Partition: A story beyond SoC

- Historically process is optimized for SoC to serve broader audiences
  - With chiplet, process can be further optimized to achieve better PPA



SoC → Chiplet → SoIC → Future

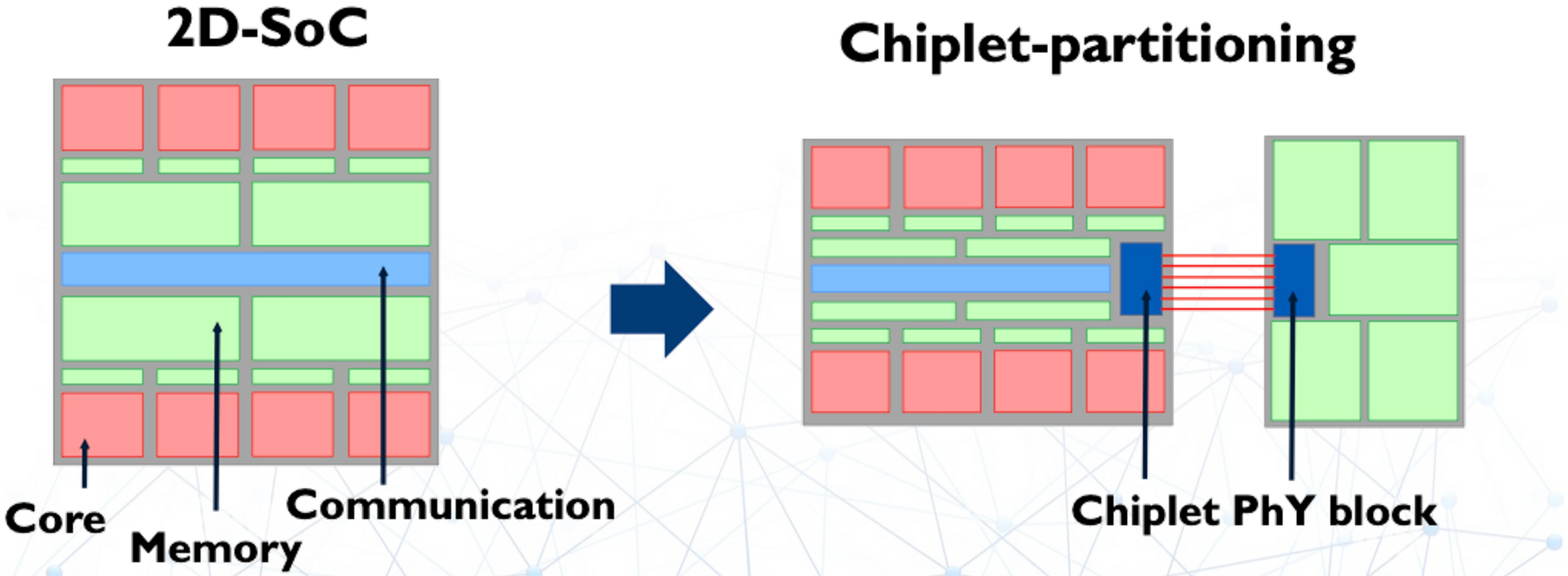
SoC
- Generations of success
- SRAM & analog/IO face scaling challenges

Chiplet
- Compute die on node N for highest performance
- Analog/IO on N-1 or N-2 to optimize cost
- MCM for low-cost interconnect

SoIC
- Logic SoIC to increase performance, CPU, GPU, or SRAM can stack

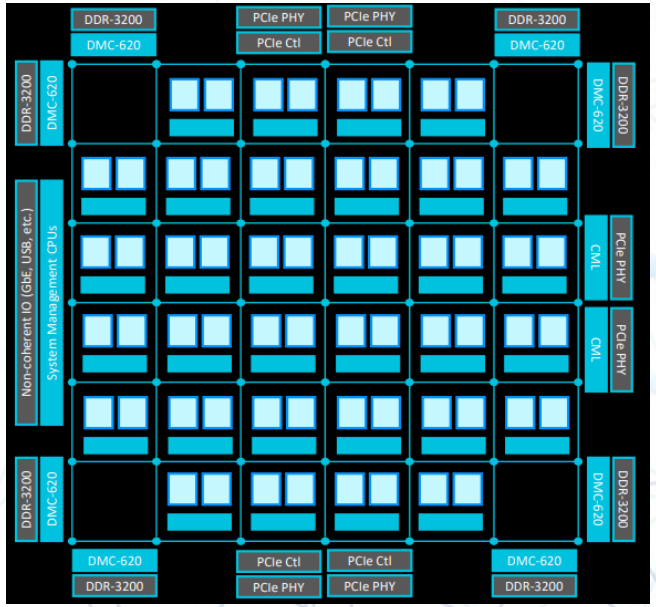
Future
- Optimizing logic with different technology nodes
- Logic stacking to increase performance
- On-board memory to improve memory bandwidth
- Interposer for higher connection bandwidth

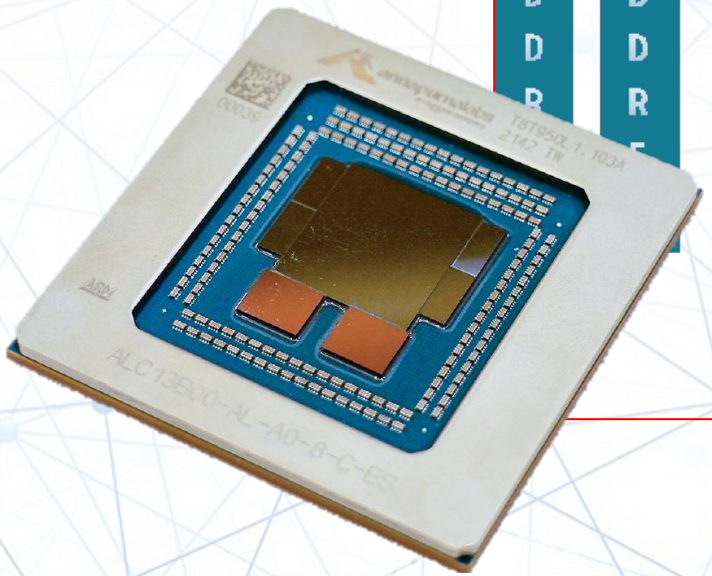
2D SoC → Chiplet Partition Approach

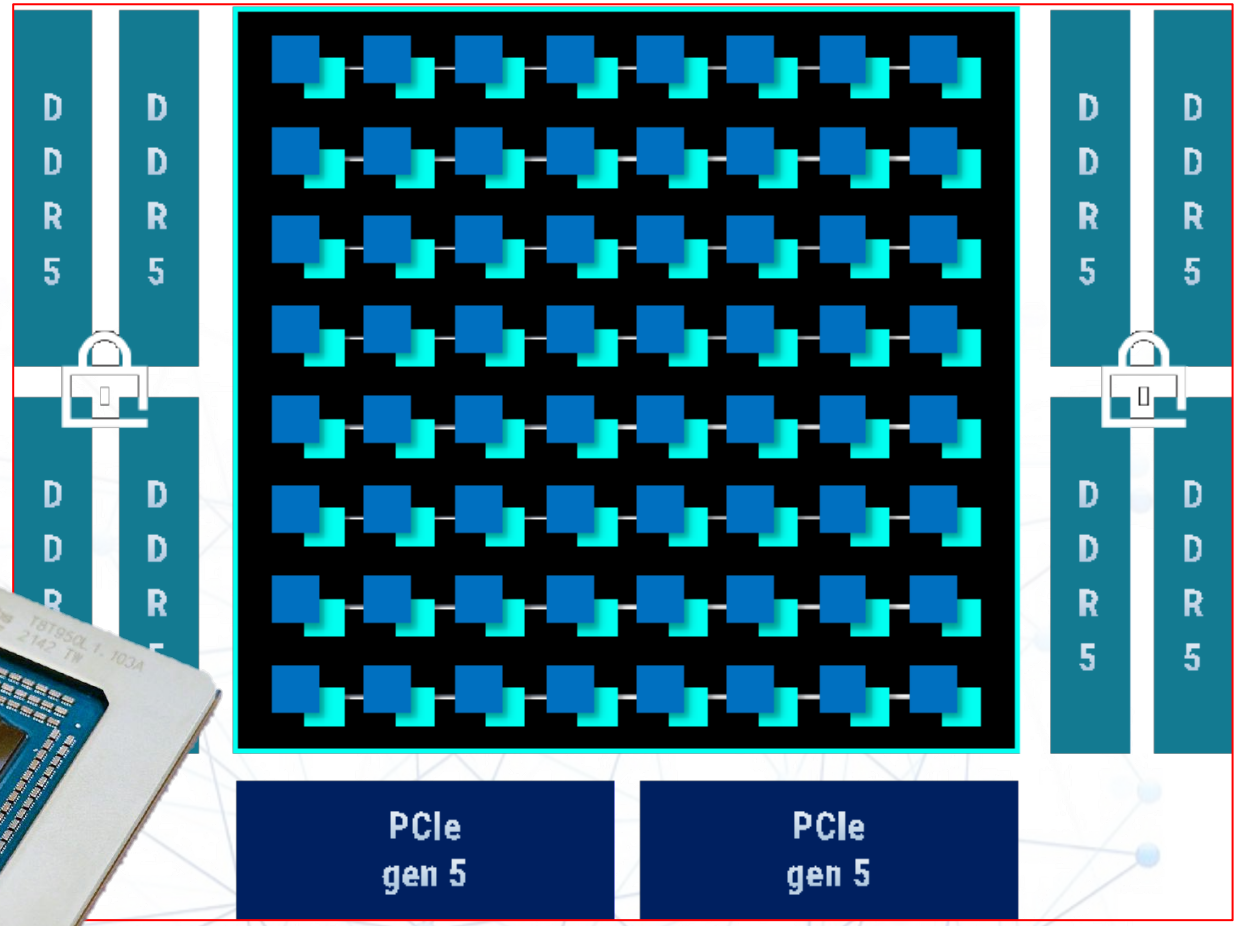


2D SoC → Chiplet Partition Example

- 1 Computing Die(N5) + 2 PCIe5 I/O Die + 4 DD5 I/O Die
- 64x-128x ARM Neoverse N1 CPU
- 8x8 Mesh up to 128MB of cache

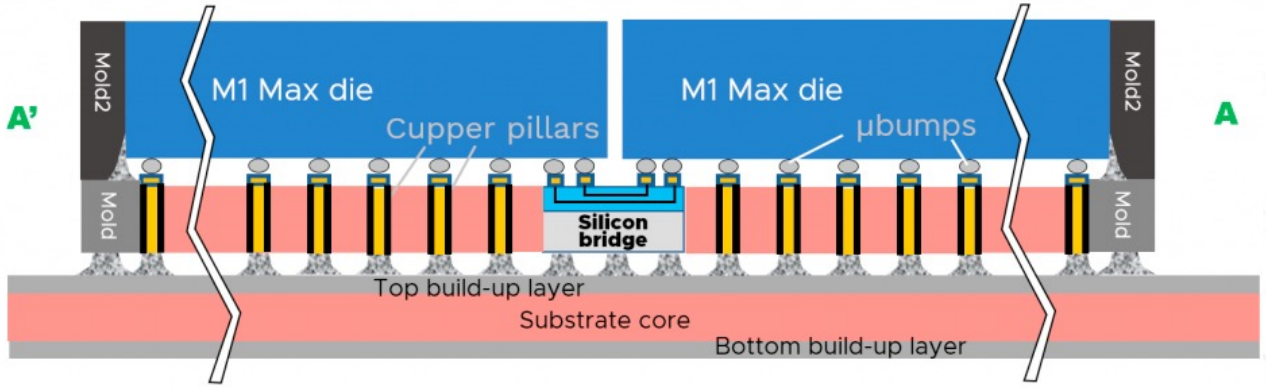


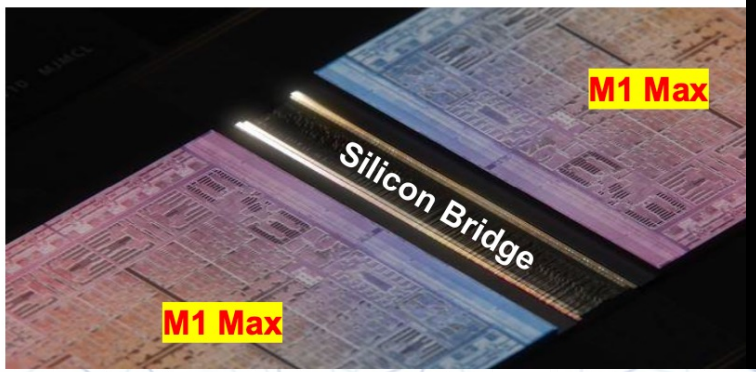


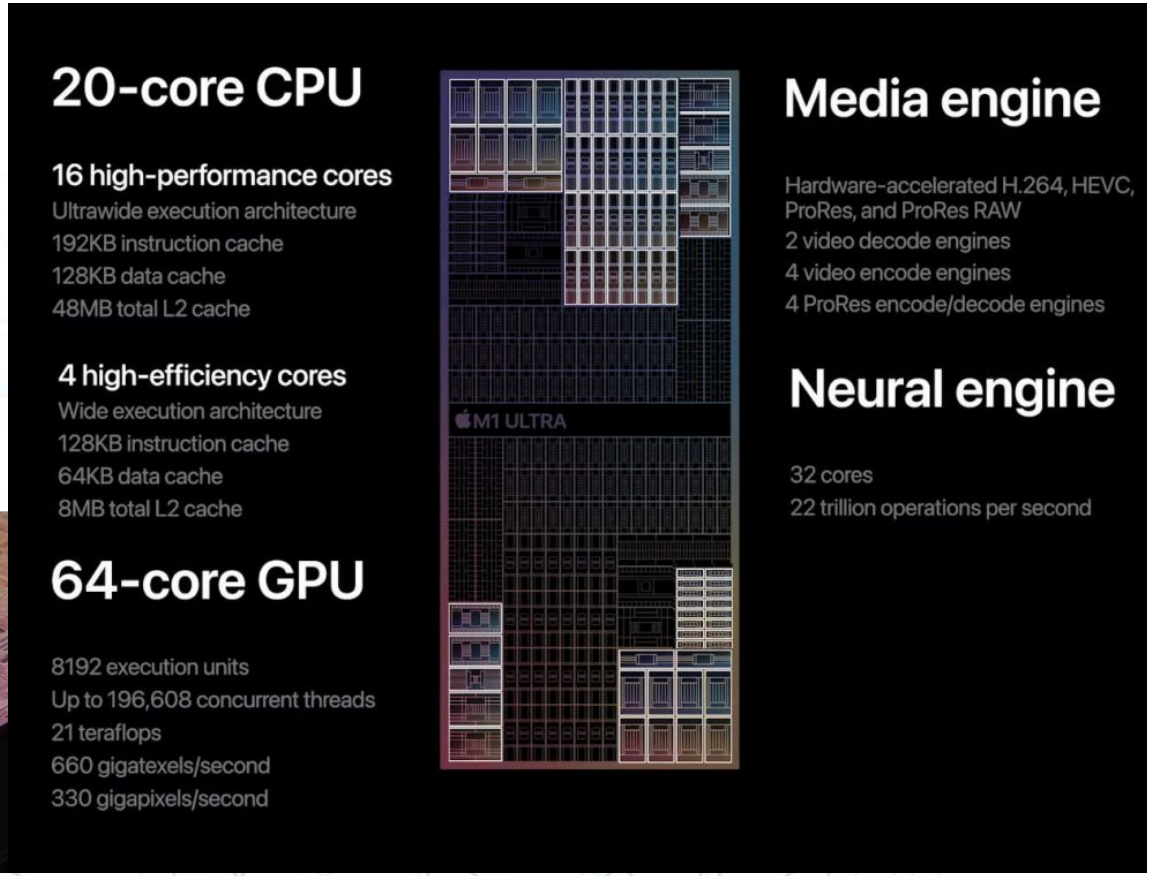


2D SoC → Chiplet Partition Example

- Apple's innovative packaging architecture that interconnects the die of two M1 Max chips to create a system on a chip (SoC) with unprecedented levels of performance and capabilities.

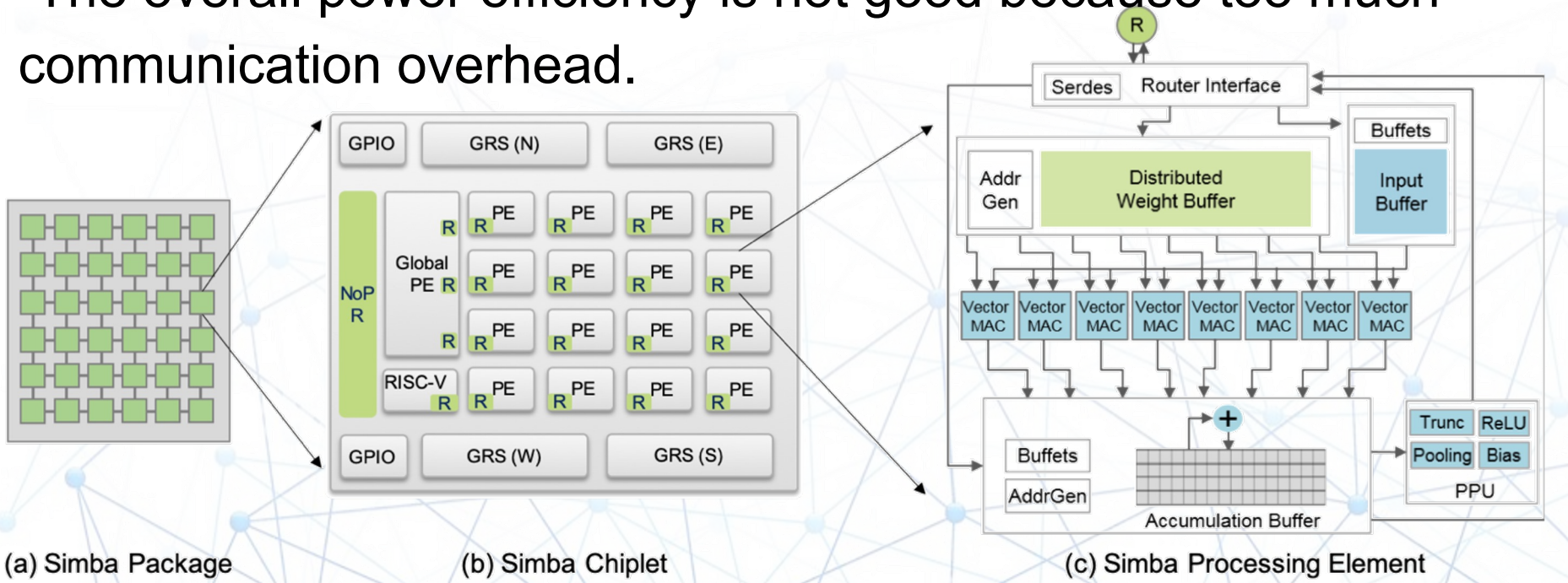




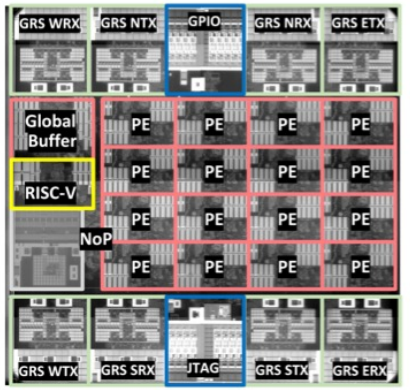


2D SoC → Chiplet Partition Example

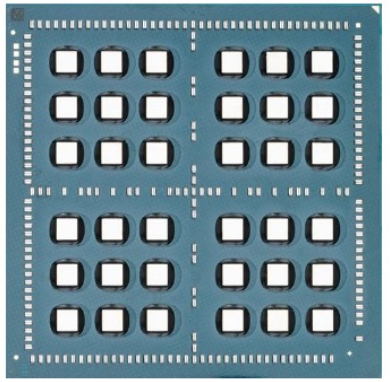
- Nvidia's Simba use 36 Chiplet to build a 128TOPS AI accelerator system. The network is implemented by a Network-on-Package with one router and 4x D2D transmitter.
- The overall power efficiency is not good because too much communication overhead.



(a) Simba Package (b) Simba Chiplet (c) Simba Processing Element

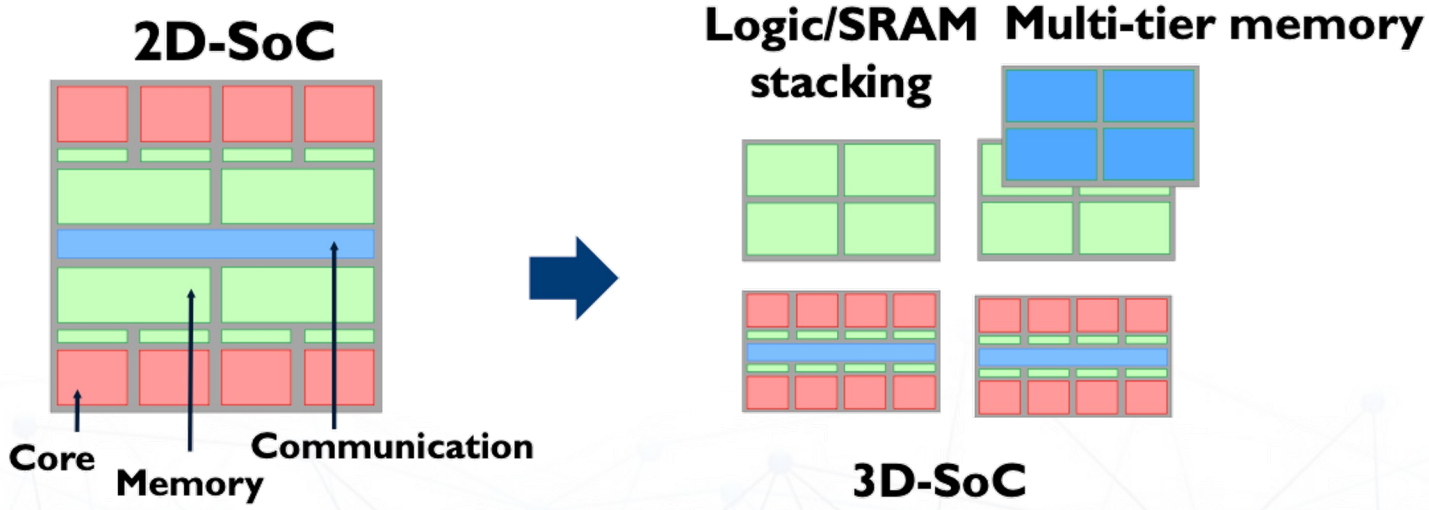


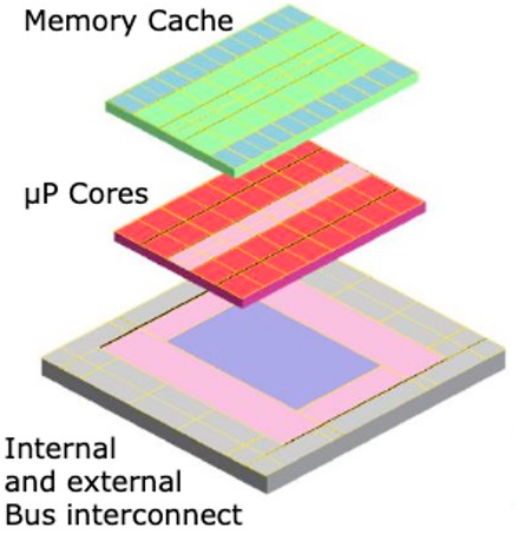
(a) Simba chiplet



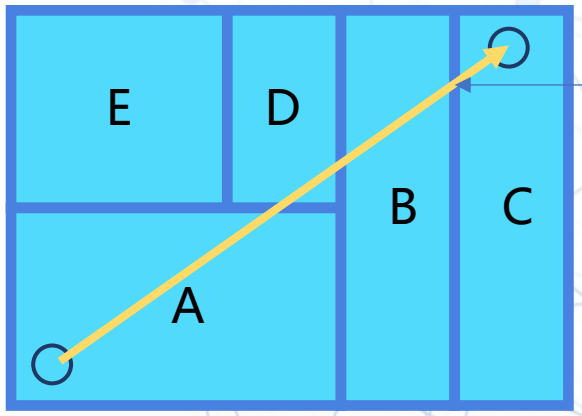
(b) Simba package

2D SoC → 3D SoC Partition Approach





平面 SoC

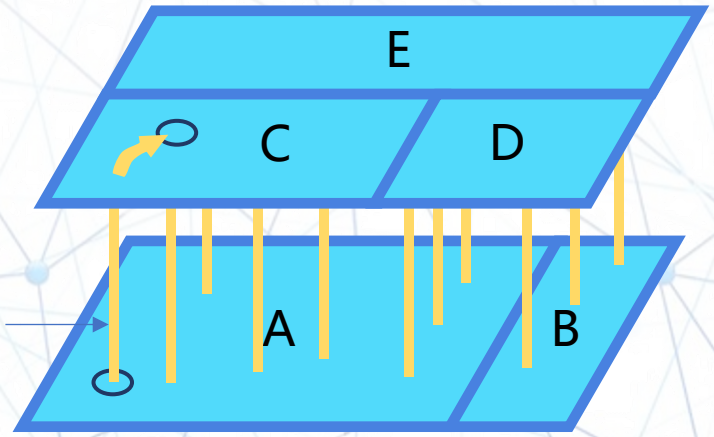


长距离
全局连线

vs

短距离
垂直连线

三维堆叠集成芯片



2D/3D 物理设计比较

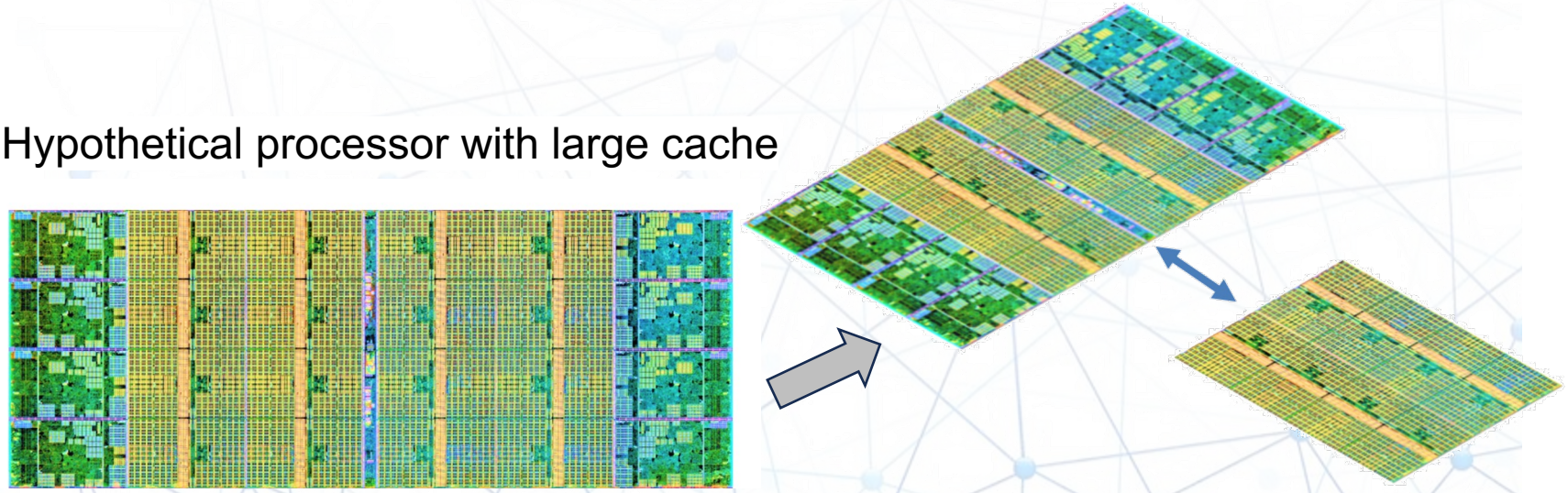
	2D 布局布线	3D 布局布线	收益
面积mm^2	0.55	0.34	37%
布局密度	54.9%	88.2%	
工作频率	3.7G	4.1G	13%

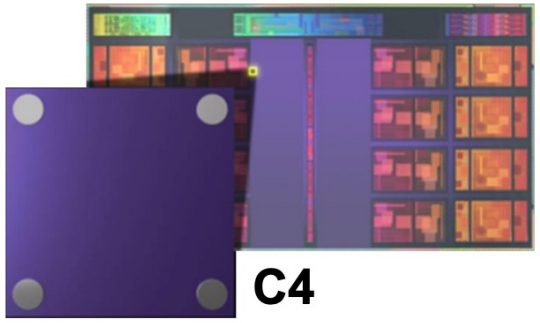
Source: Cadence 3D-IC

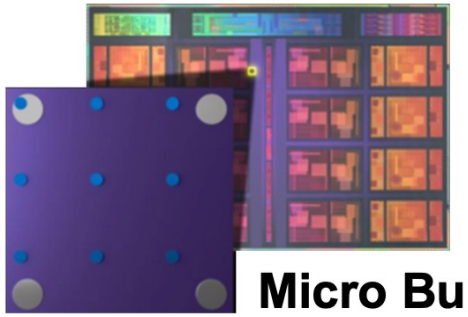
2D SoC → 3D SoC Partition Example

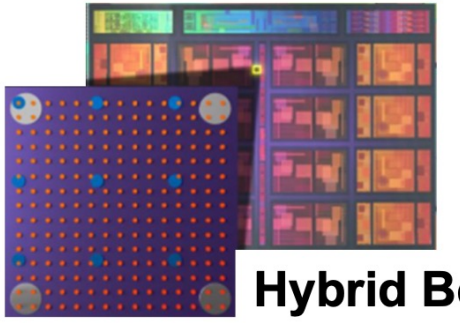
- Why 3D partition is more attracting?
  - Improves effective memory latency
  - Reduces long datapath and I/O's dynamic powers
  - Fits more transistors within a given package cavity size

Hypothetical processor with large cache





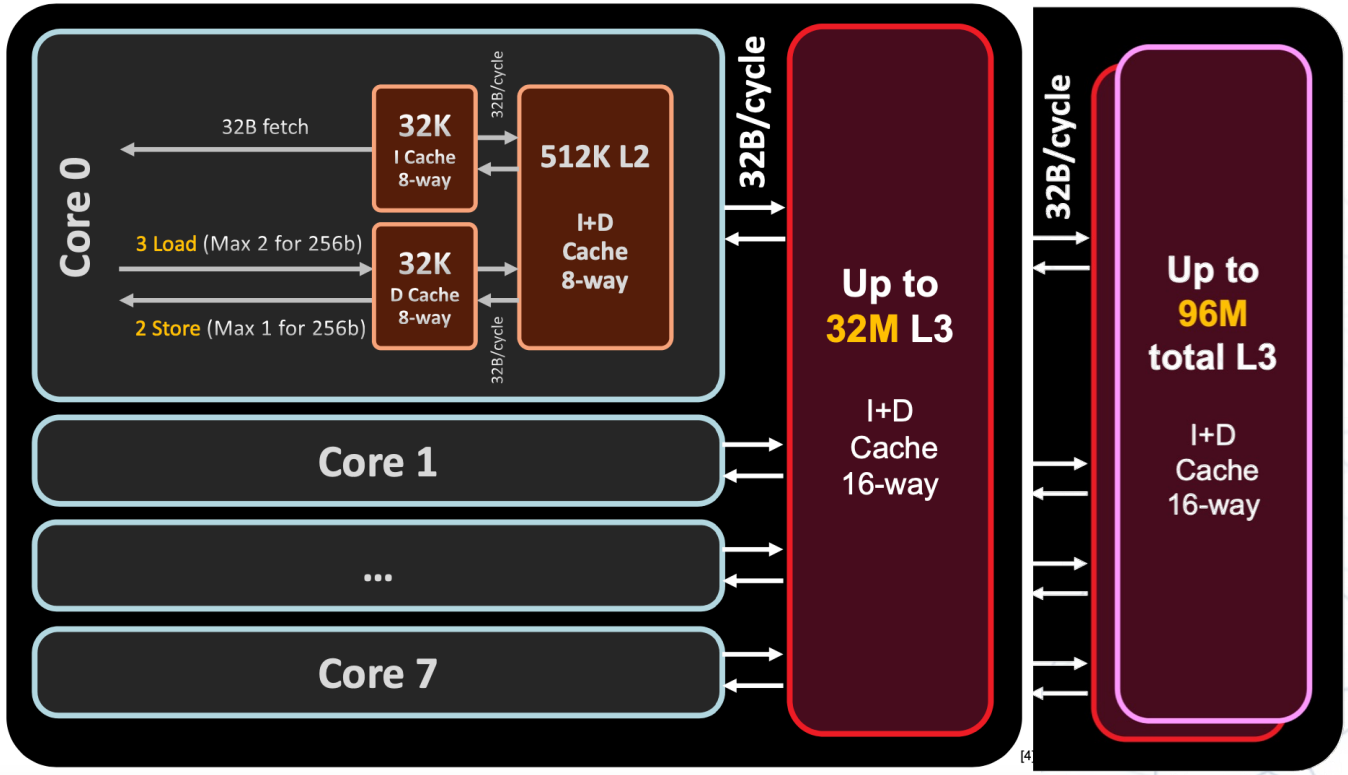


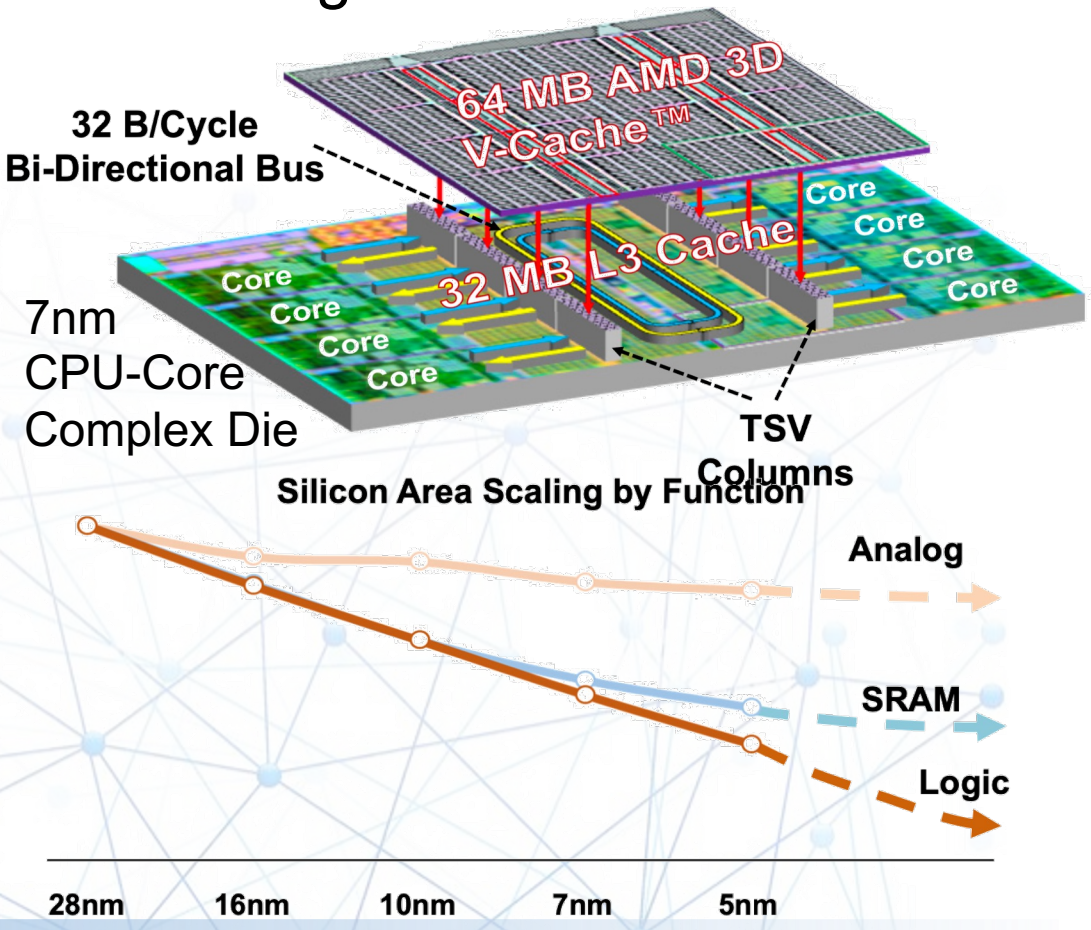


[1]

2D SoC → 3D SoC Partition Example

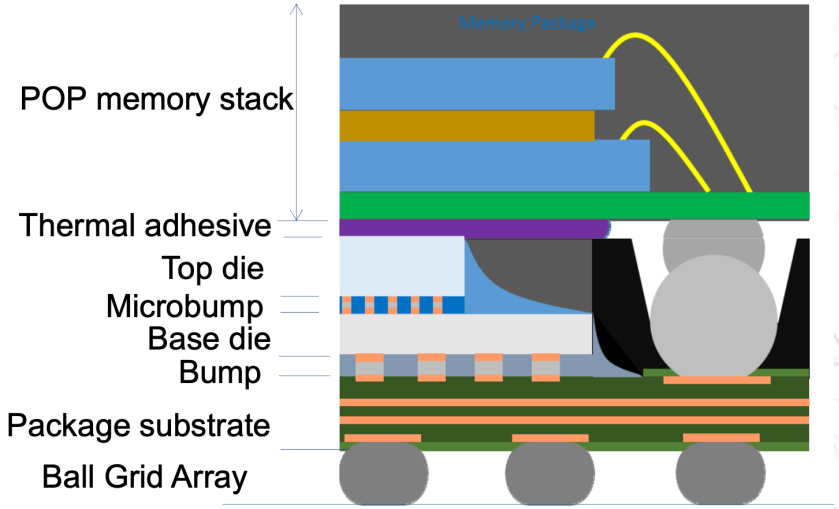
- AMD Zen 3： 8x x86-64 CPU core in TSMC 7nm with 32MB Cache
- 3D V-Cache: Extended 64MB L3 Cache via hybrid bonding

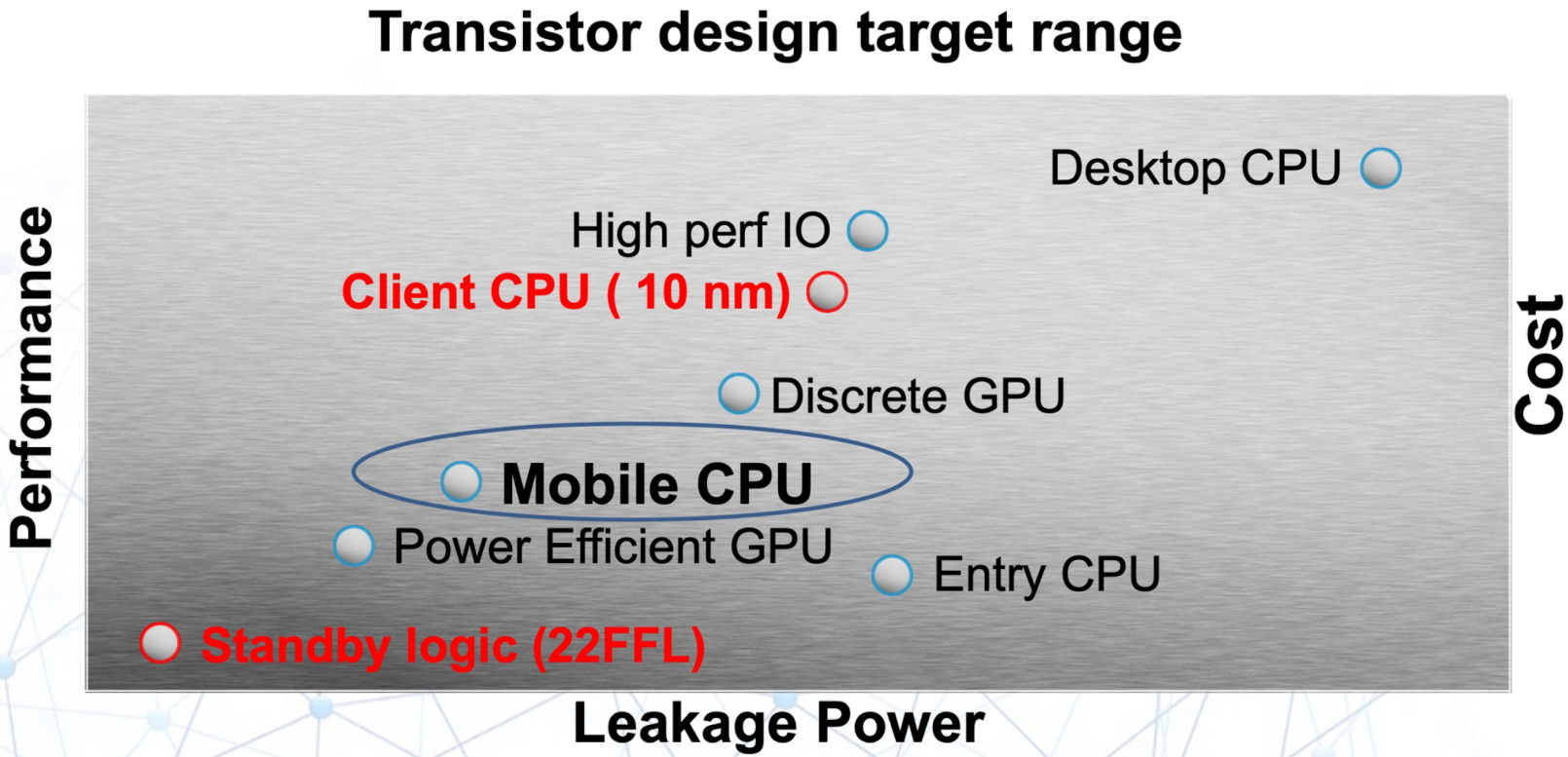




2D SoC → 3D SoC Partition Example

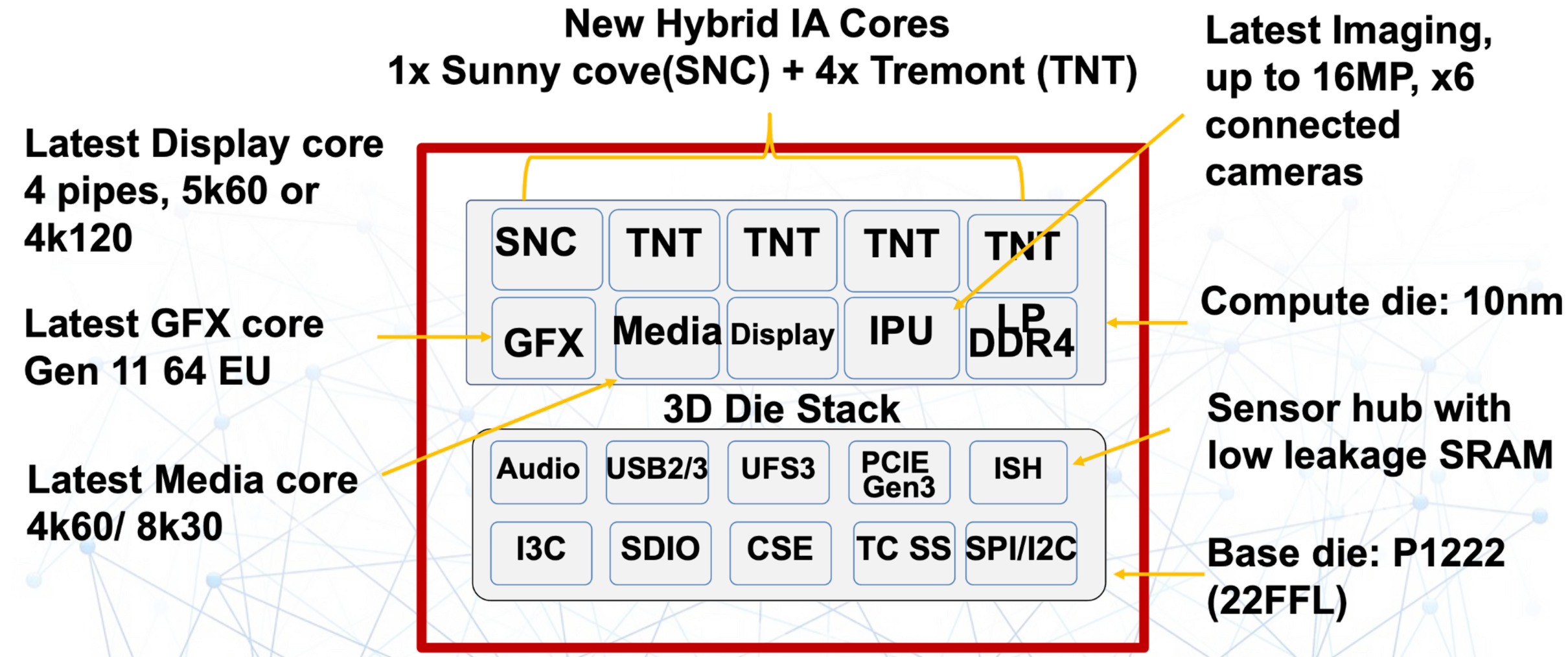
- For certain applications, no single node is optimal across all points!
- Leverage common IP in mature process: High Voltage, Passive, low scaling Analog
- Advanced technology for compute only.
- DRAM can be further stacked



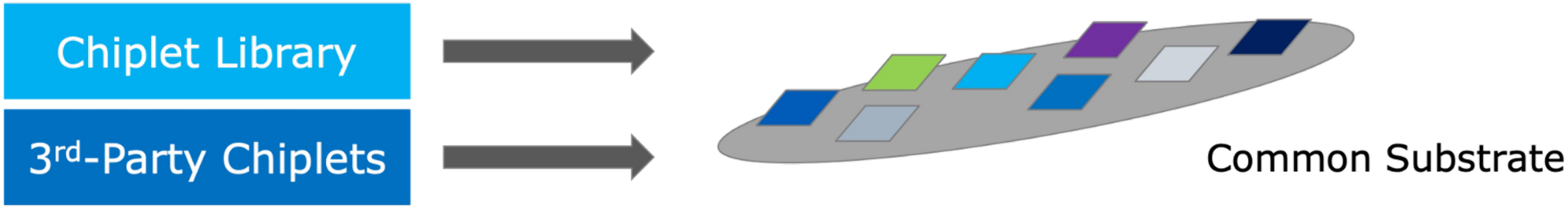


2D SoC → 3D SoC Partition Example

➢Since Lakefield, Intel use 3D stacking structure for mobile HP CPU products



Partition for Reuse and Flexibility



- Generality vs. Optimization
  - Interface widths and speeds, supported functionality/protocol(s), fixed pinouts
  - Memory options (e.g., not all systems need HBM)
  - Form factor/chiplet size (hard to support too many sizes, unnecessarily large silicon increases costs)
  - Power delivery: pinouts, supported voltage(s), voltage regulation, current draw, required decap
  - Thermal budgeting/allowance, cooling solutions
- Design for reuse in large-scale design
  - How to leverage wafer-scale systems across smaller scales?