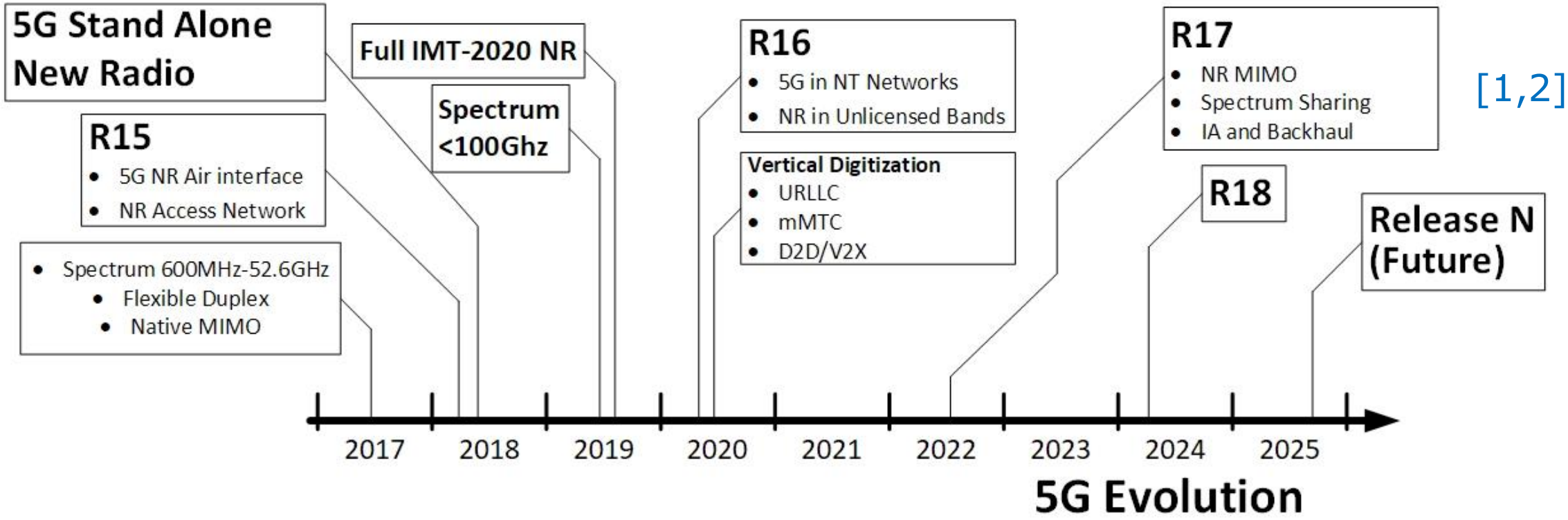
ISSCC 2024 Forum F1:
Efficient Chiplets and Die-to-Die Communications

1.5: Efficient Domain-Specific Compute with Chiplets

Prof. Dejan Marković
UCLA ECE Department
dejan@ucla.edu

Evolving Standards: *Flexibility & Efficiency*

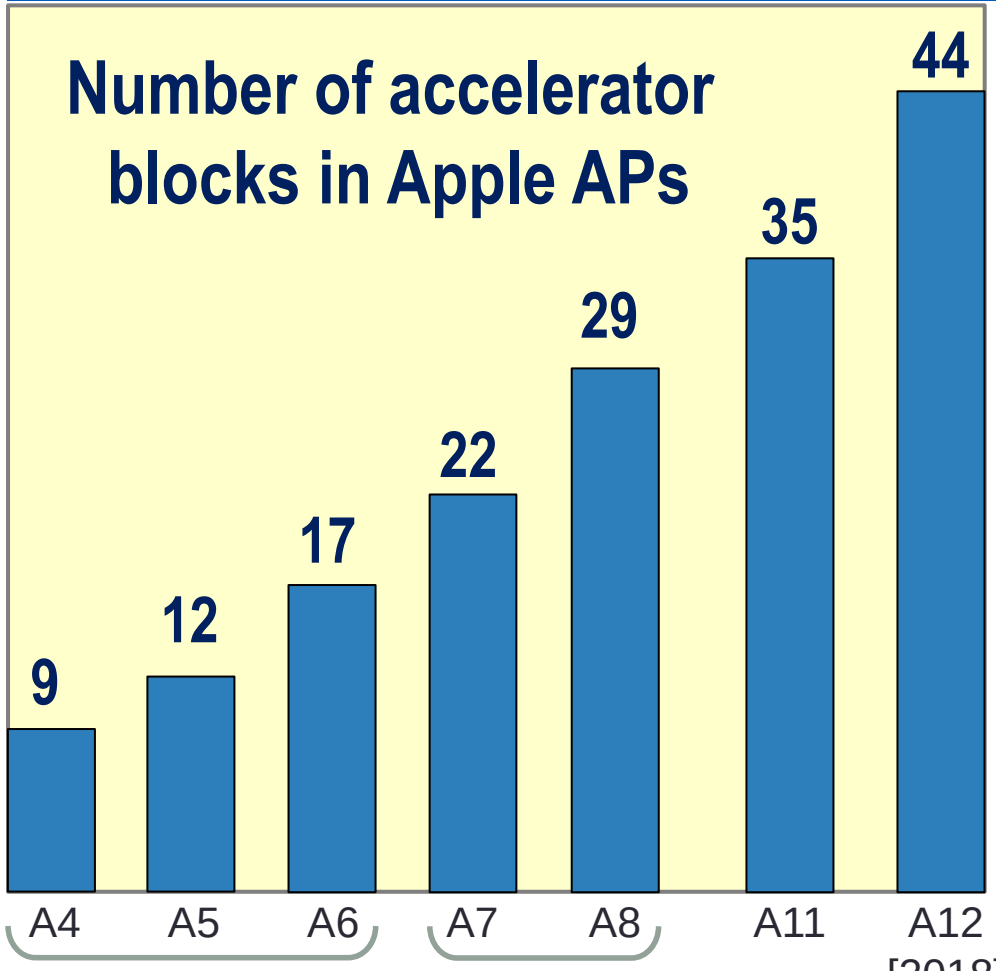
Objectives: lower development cost and shorter time-to-market

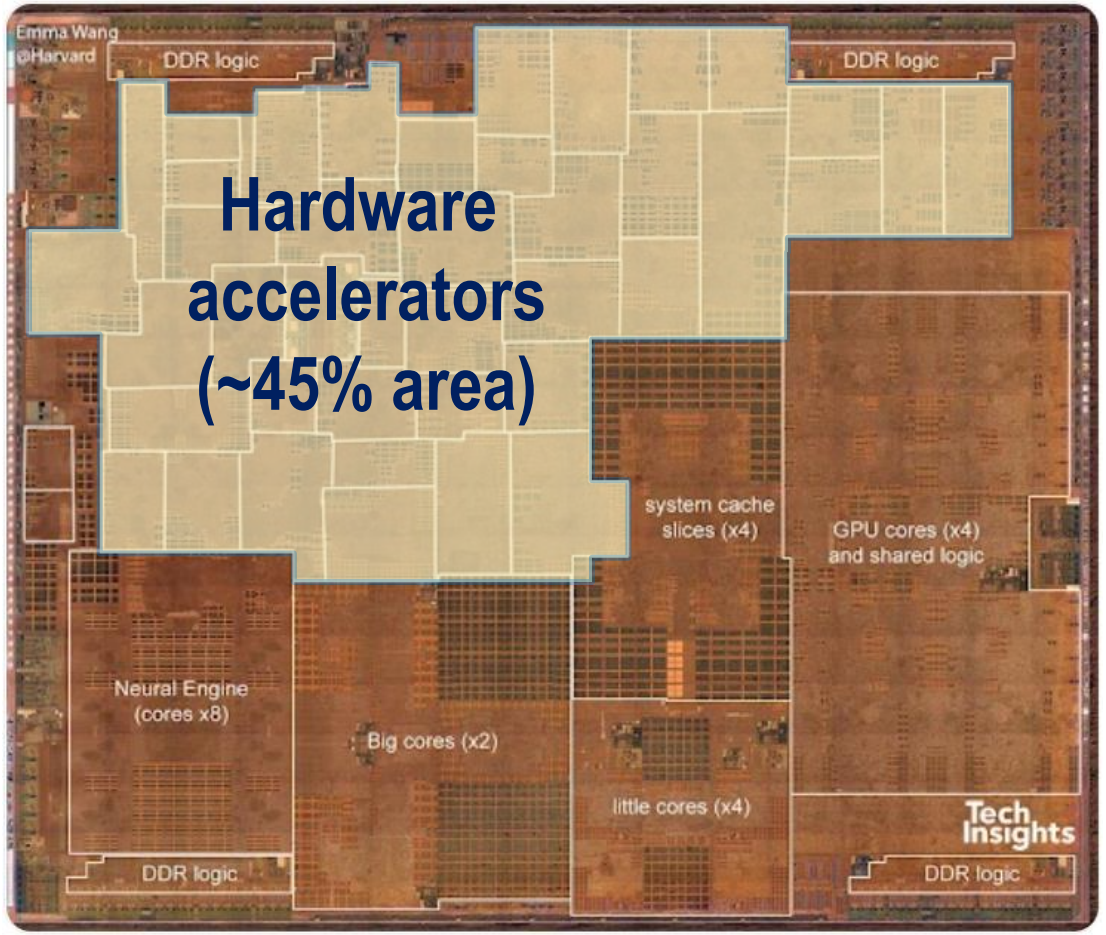


[1,2]

- SoC/ASIC revision / iteration is $$$ (~$100M in 16nm CMOS)
- Long design cycles (>1 yr)[3] with increasing design complexity

SoCs Today = CPU/GPU + Accelerators





Apple A12 die photo

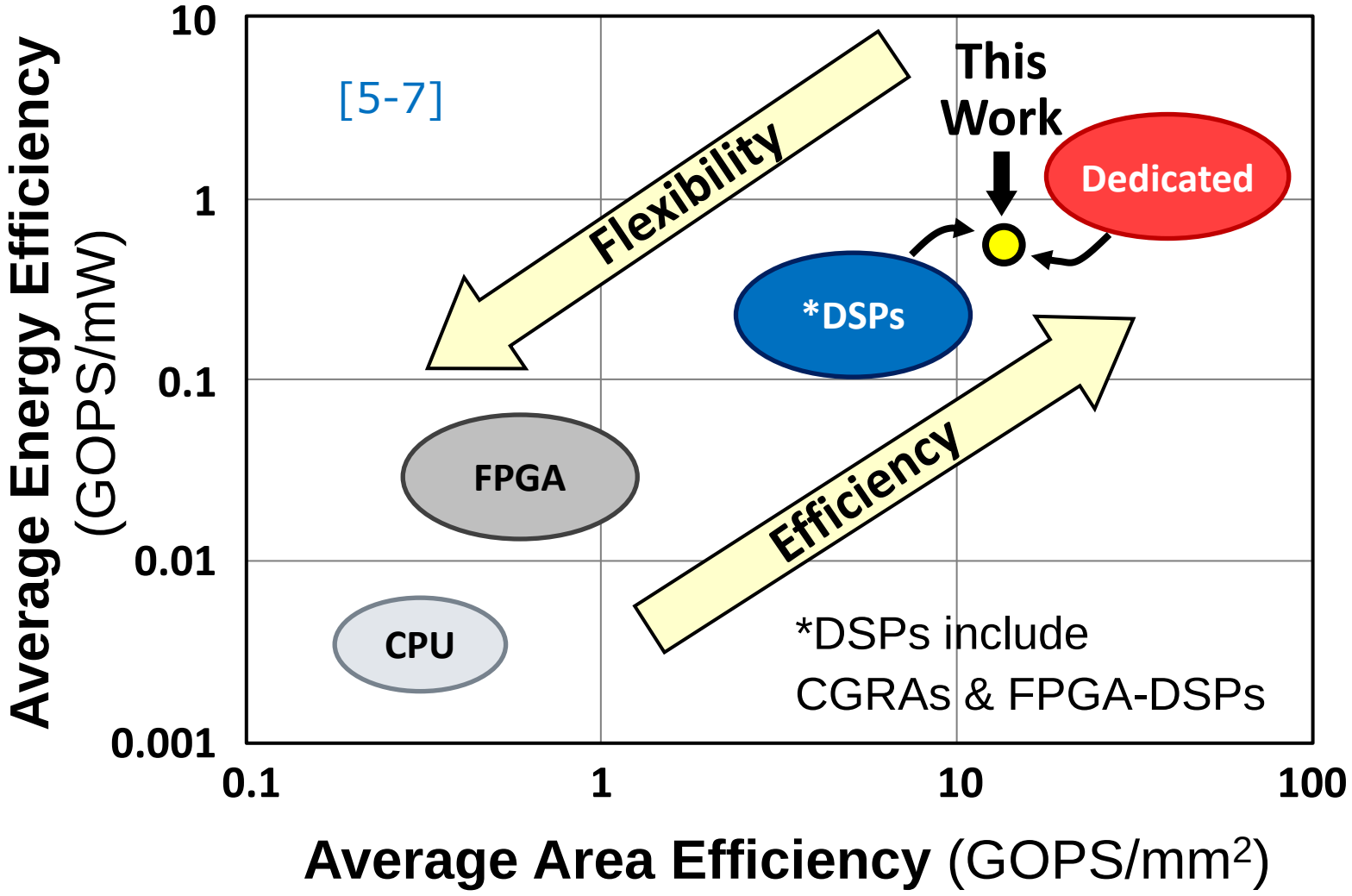
Optimize for Efficiency and Flexibility

Two ways to think about it

- Add flexibility to accelerators
- Narrow coverage of DSPs

The how

- Interconnect
- Switch-boxes
- Sw toolchain



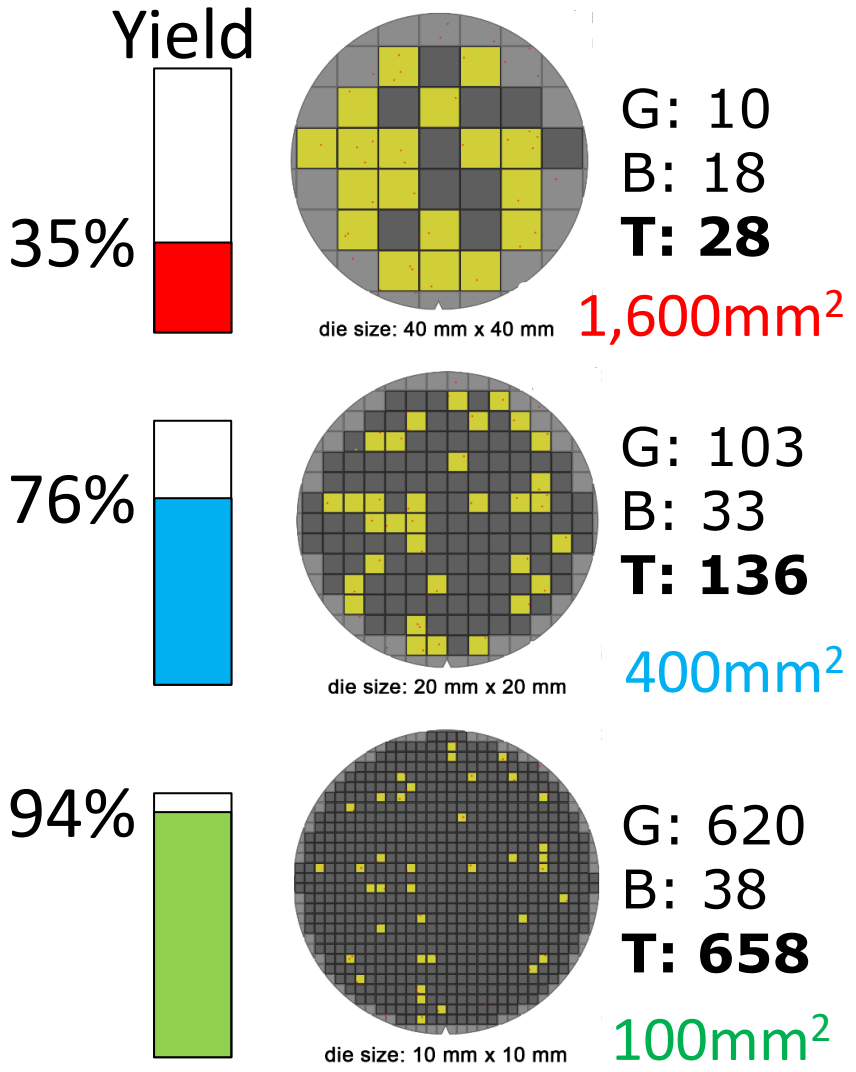
Efficient Multi-Chip Module (MCM) Scaling

- Large SoCs incur higher cost
  - Lower yield of larger chips
  - Delayed time-to-market

- Cost benefits of MCM scaling
  - Smaller chips give better yield

$$\boldsymbol{Yield} \propto \frac{\mathbf{1}}{(\mathbf{1} + \boldsymbol{D} \times \boldsymbol{A})^{\boldsymbol{k}}}$$ [8]

  - AMD 32-core chip (777mm^2): 1.0x Cost
  - 4 x 8-core chiplet (4x213mm^2): 0.6x[20]



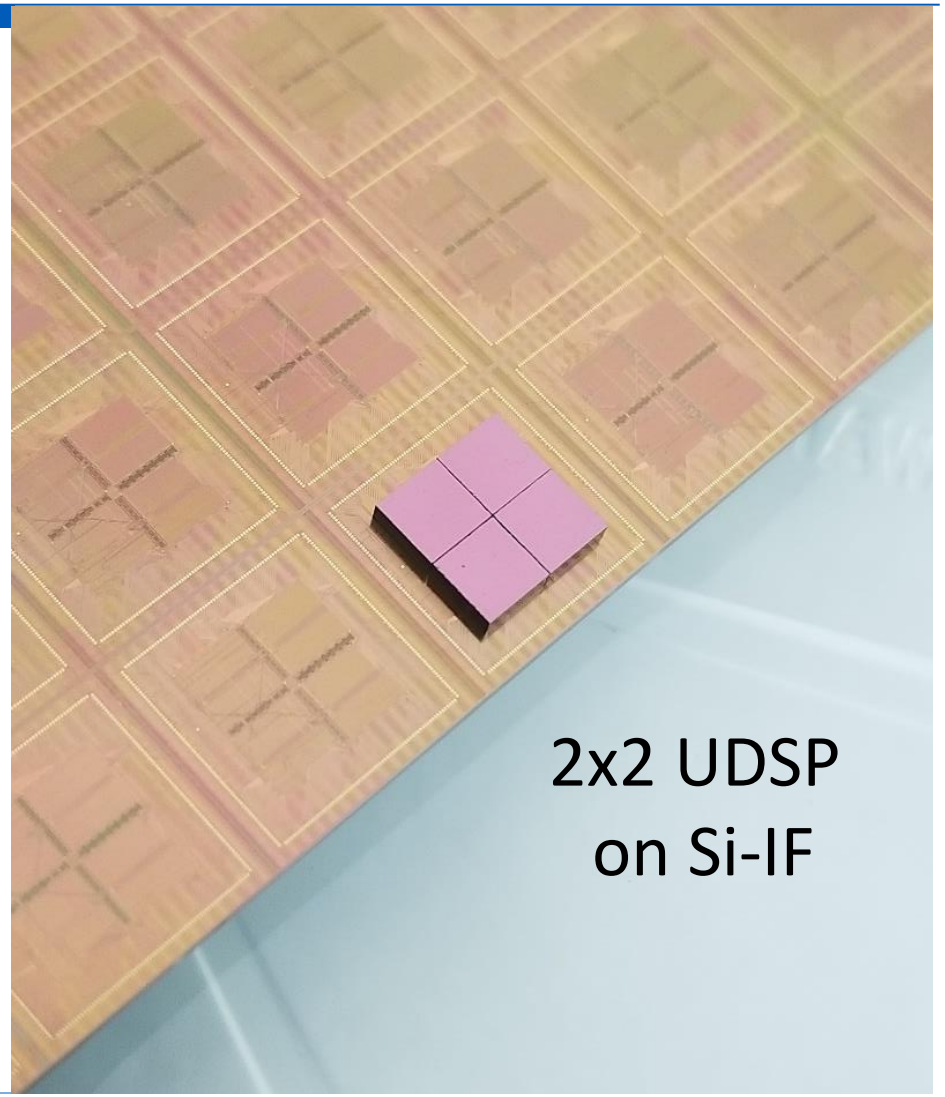
Challenges with MCM Design

Challenges:

- High bandwidth density
- Low link latency
- Low energy transfer
- Low I/O area

Chiplet size:

- Sweet spot: ~100mm^2
- UDSP prototype ($ limited): ~6mm^2



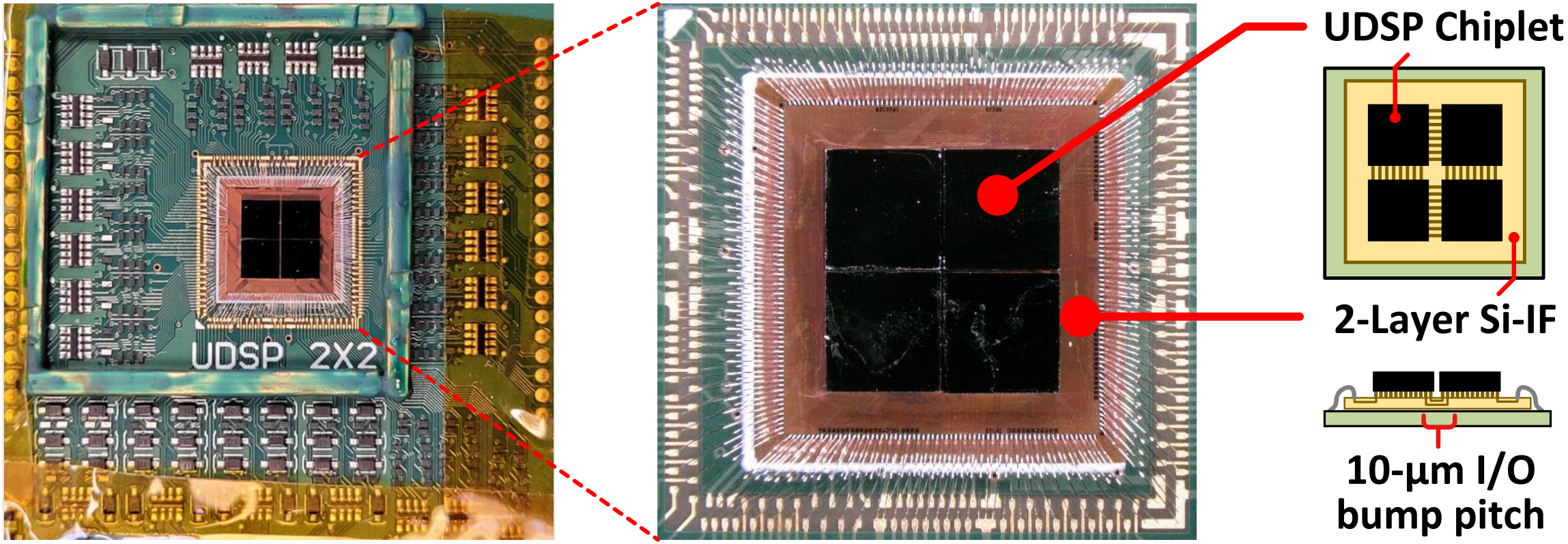
Research Aims

- **Domain-specific hardware acceleration**
  - ASIC-like energy efficiency and throughput
  - Just-enough flexibility for a domain
  - Key: flexible cores, efficient interconnect

- **Tile-able chiplets on Silicon Interconnect Fabric (Si-IF)**
  - Develop scalable interconnects
  - Near-range I/O and PHY for cutting-edge bandwidth/latency/energy
  - Low-area, portable timing correction circuits for Si-IF I/Os

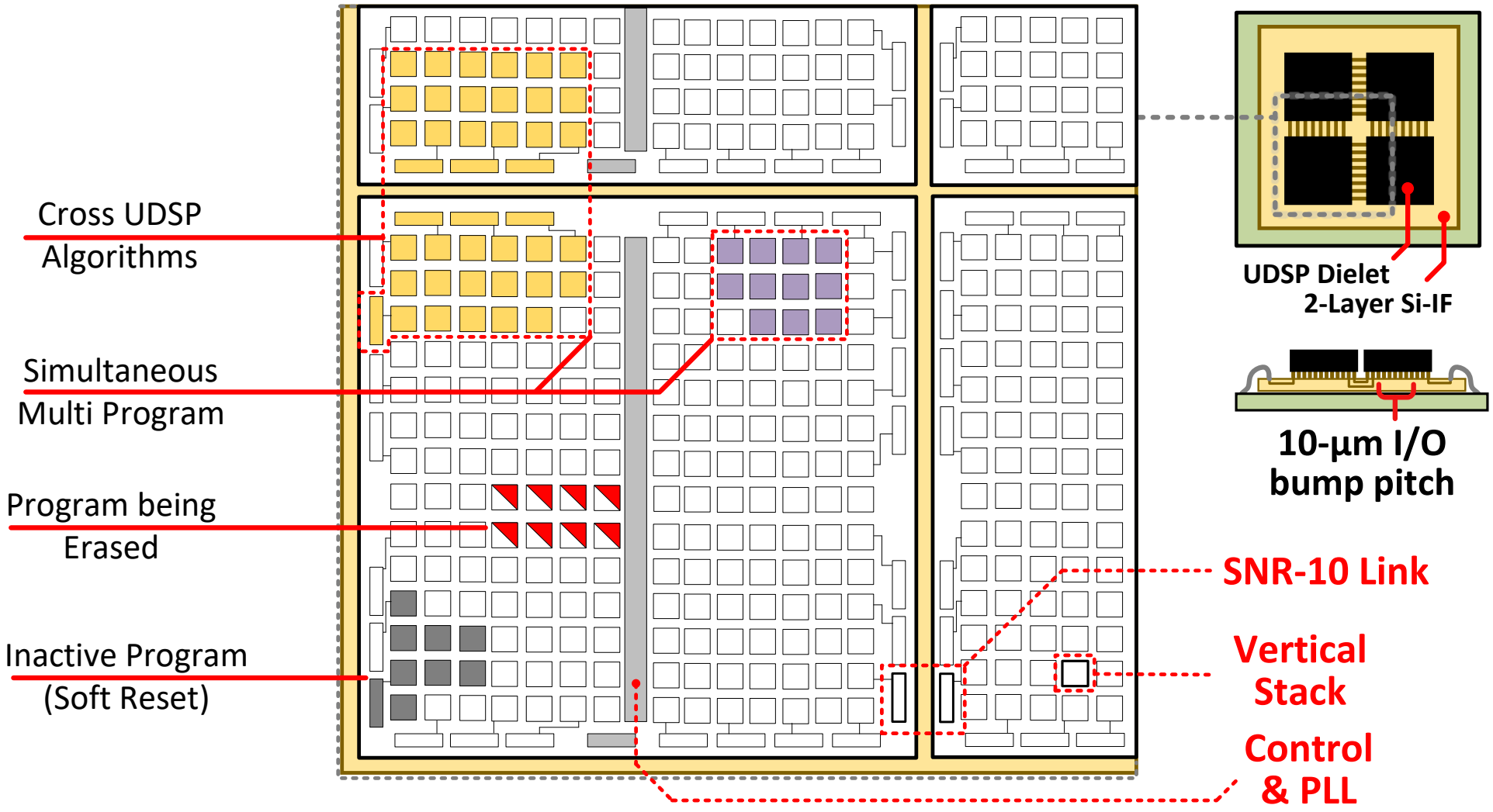
A 16nm 2x2 Chiplet with 10-µm Pitch-I/O

- Universal Digital Signal Processor (UDSP) Array



[9] U. Rathore, S. Nagi, S. Iyer, D. Markovic, ISSCC 2022.

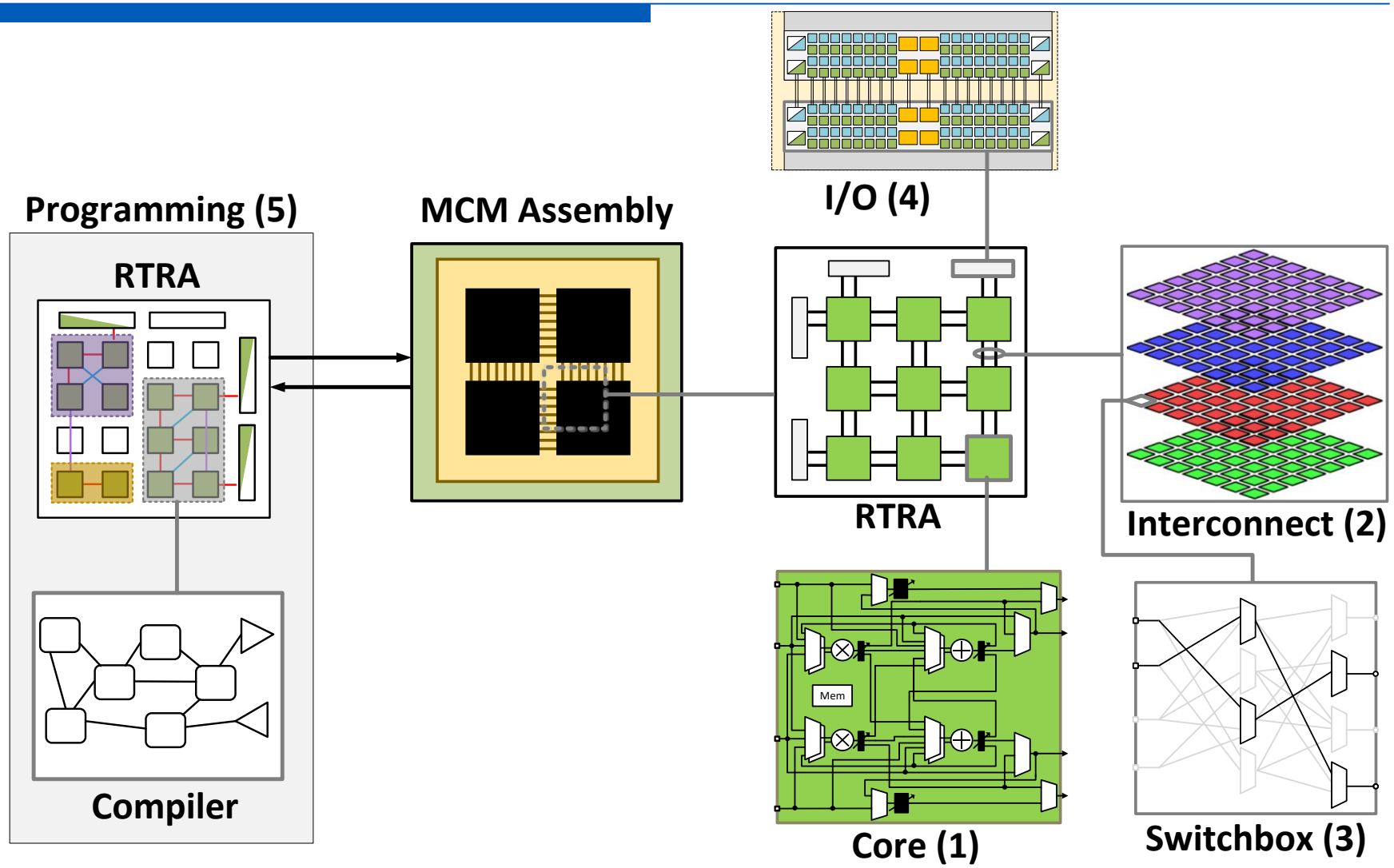
UDSP Multi-Chip, Multi-Program Tenancy



UDSP Overview

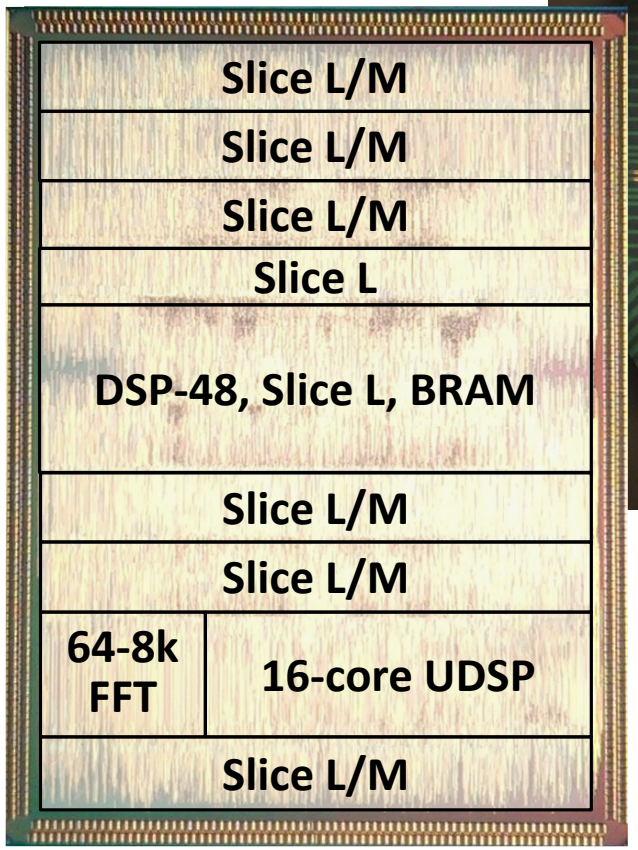
Co-design

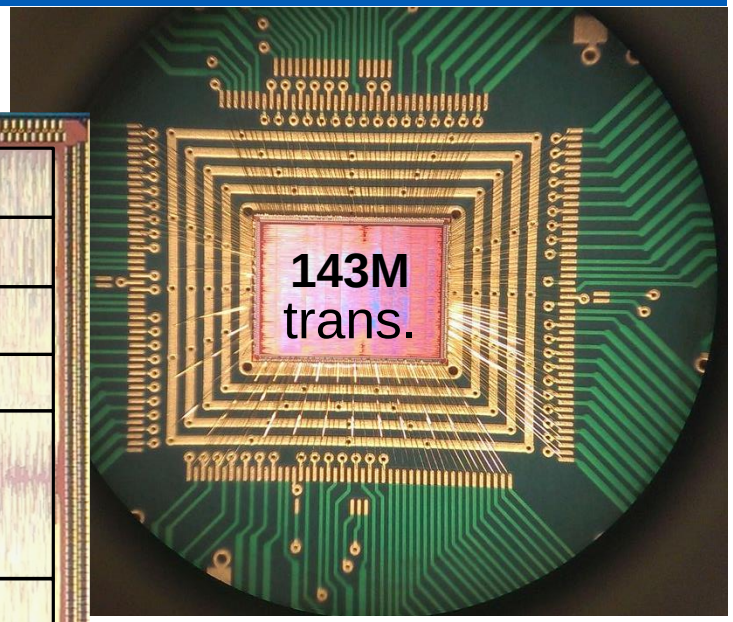
- Compute
- Interconnect
- I/O channel
- Compilers
- Package

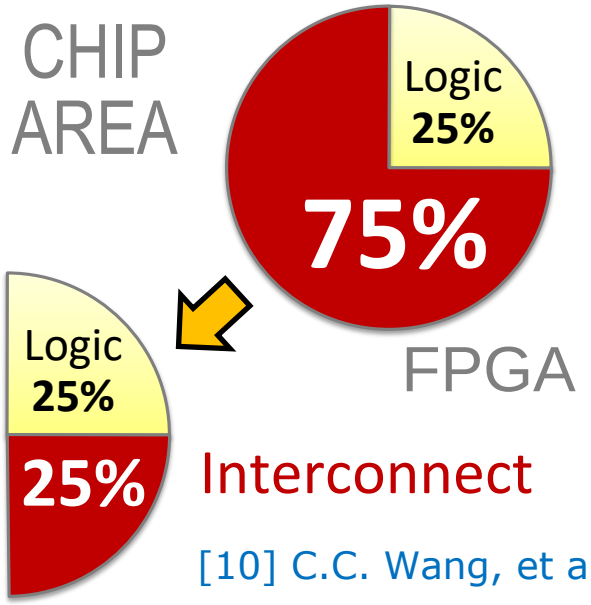


Evolution of UDSP Core

24.5mm^2 (40nm)

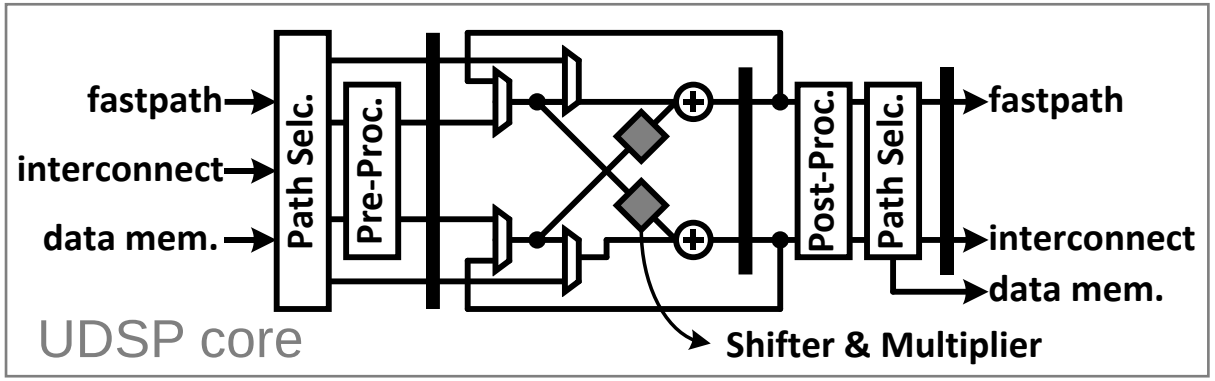




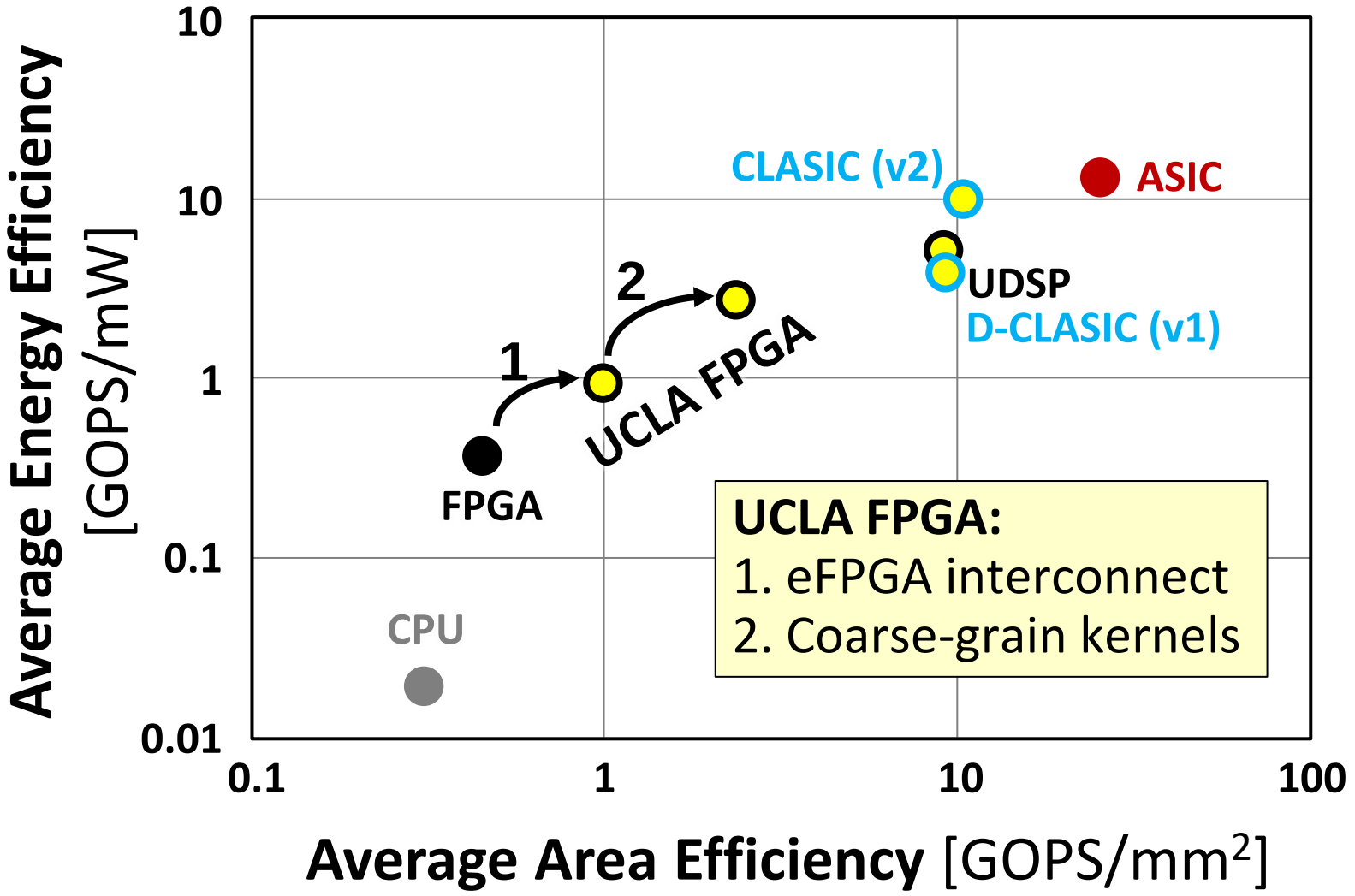


[10] C.C. Wang, et al., ISSCC 2014.

2014 Lewis Winner Award



Efficiency and Flexibility in Comm. DSP



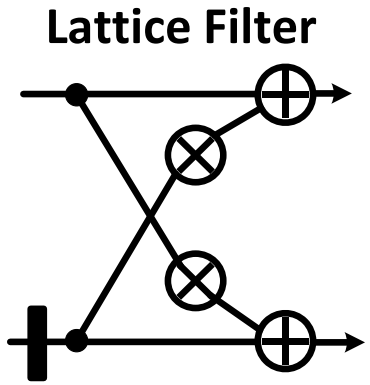
- UDSP based CLASIC designs
- Domain-specific for comm. DSP

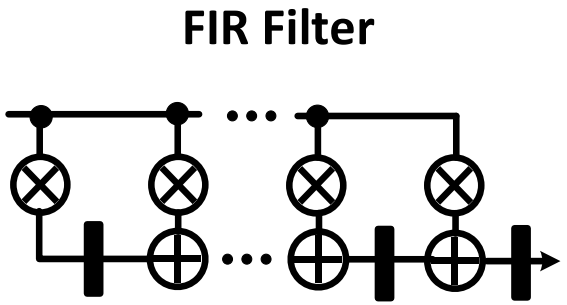
[10] C. Wang et al., ISSCC 2014.
[11] F.-L. Yuan et al., VLSI 2014.
[12] F.-L. Yuan et al., VLSI 2015.

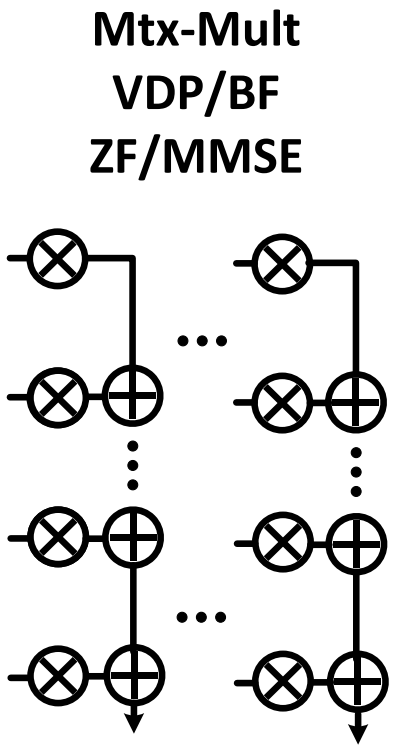
Algorithm Ontology: Example DSP Kernels

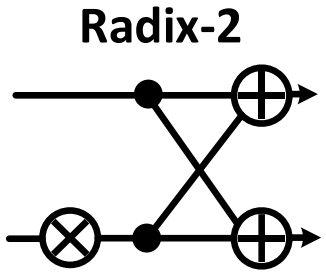
Example DSP kernels derived from common DSP algorithms

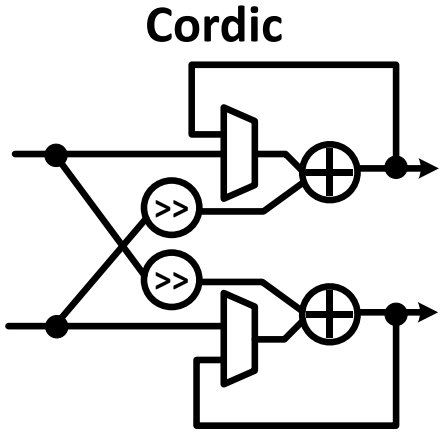
- Up/Down Conv.
- MIMO
- IFFF/FFT
- Neural Network
- Zero Forcing
- MMSE
- Vector-dot product
- MAC, FIR, Euclidian

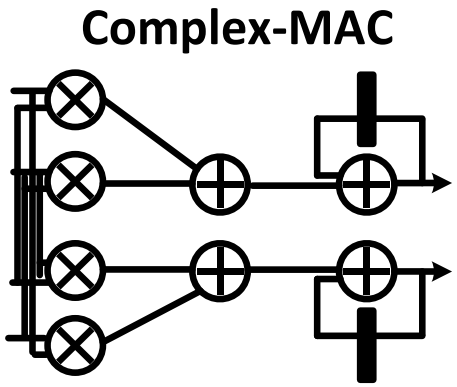


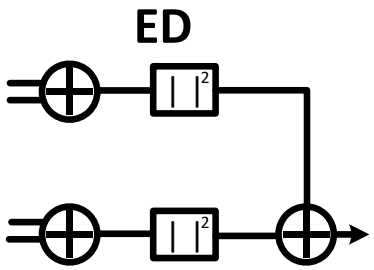




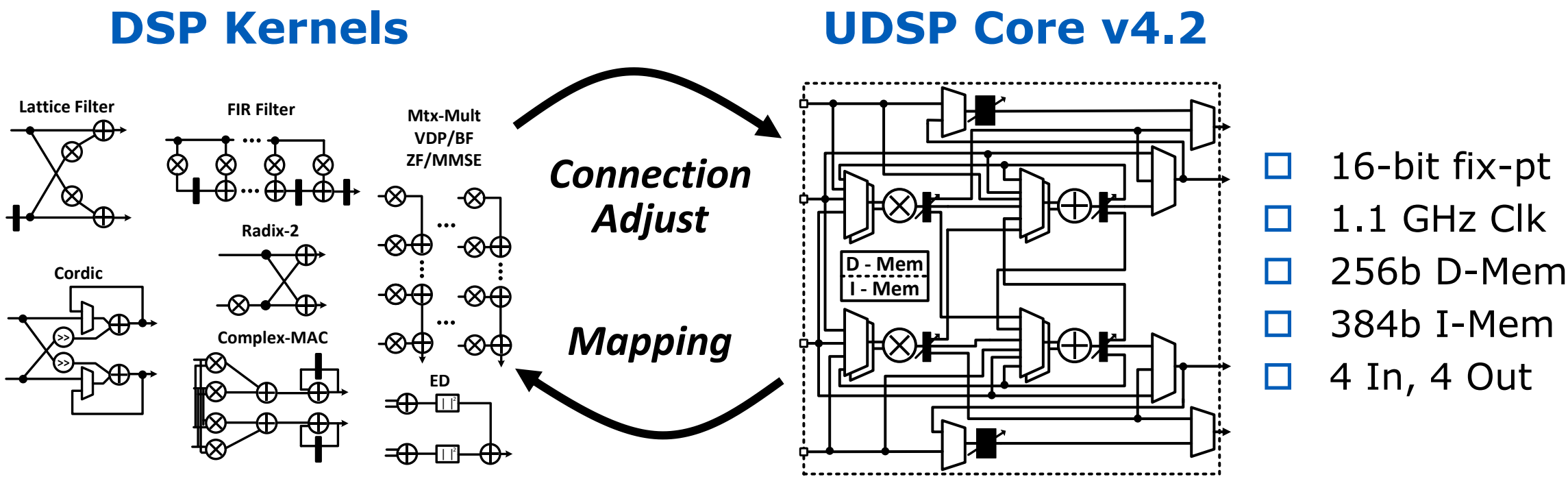








Iterative Process of Core Design

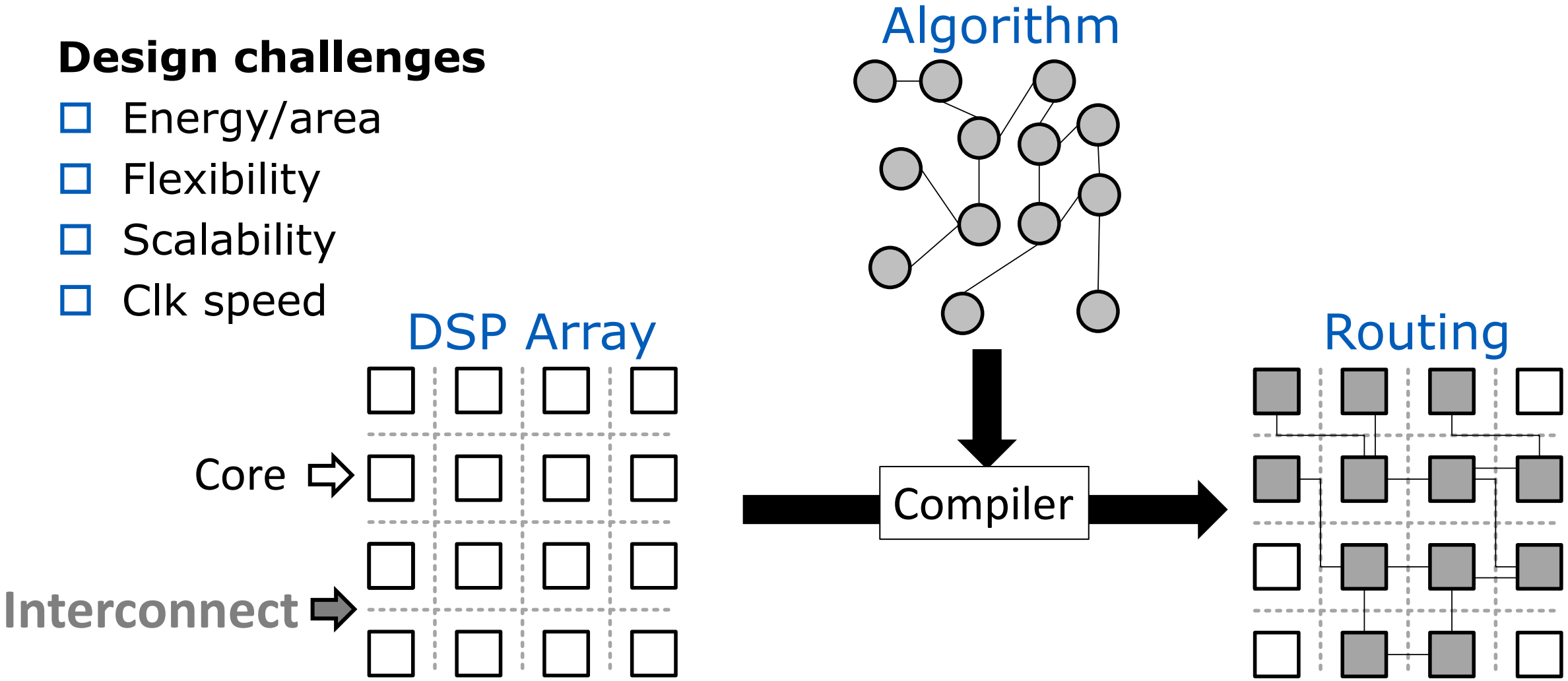


- Balancing *core granularity* and *core utilization* to maximize energy and area efficiency

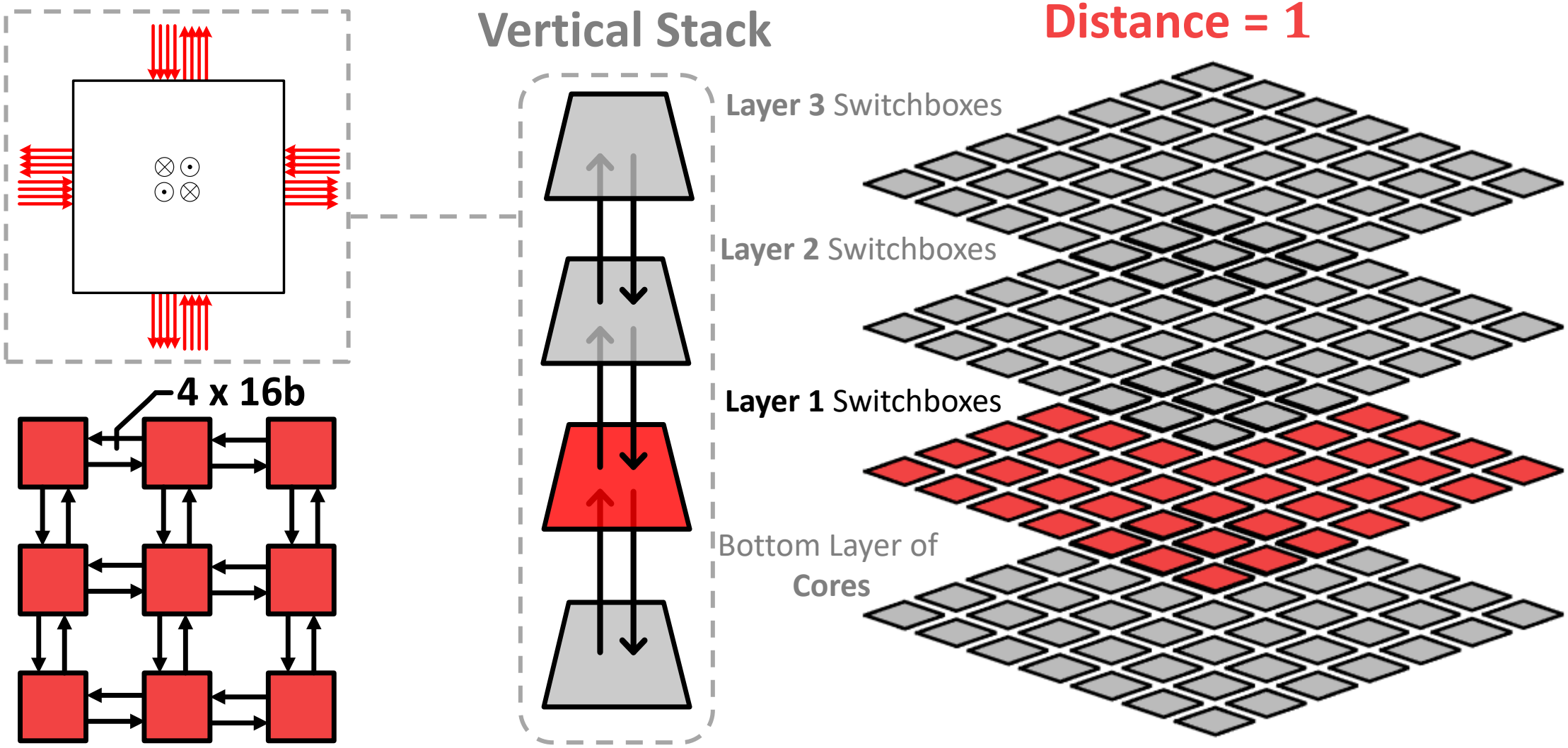
Interconnects: An Exercise in Co-Design

Design challenges

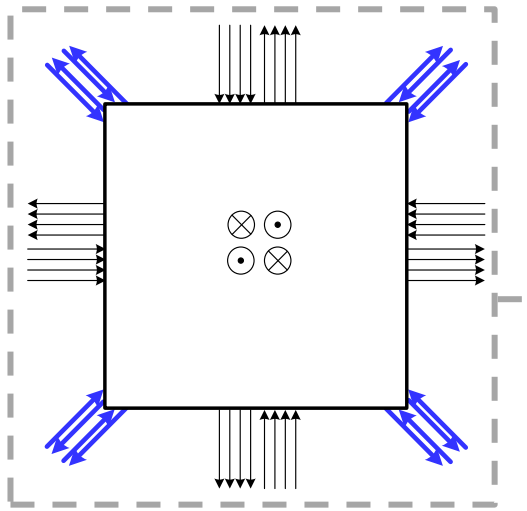
- Energy/area
- Flexibility
- Scalability
- Clk speed

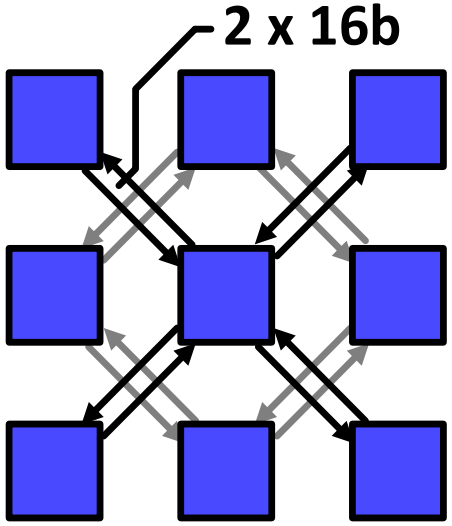


Layer-1 Interconnect (Distance = 1)

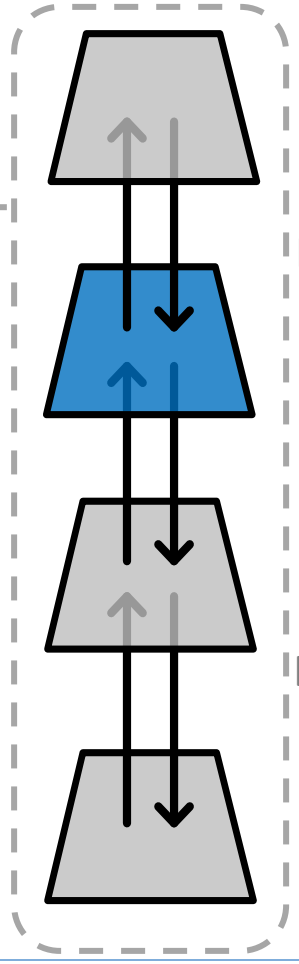


Layer-2 Interconnect (Distance = $\sqrt{2}$)

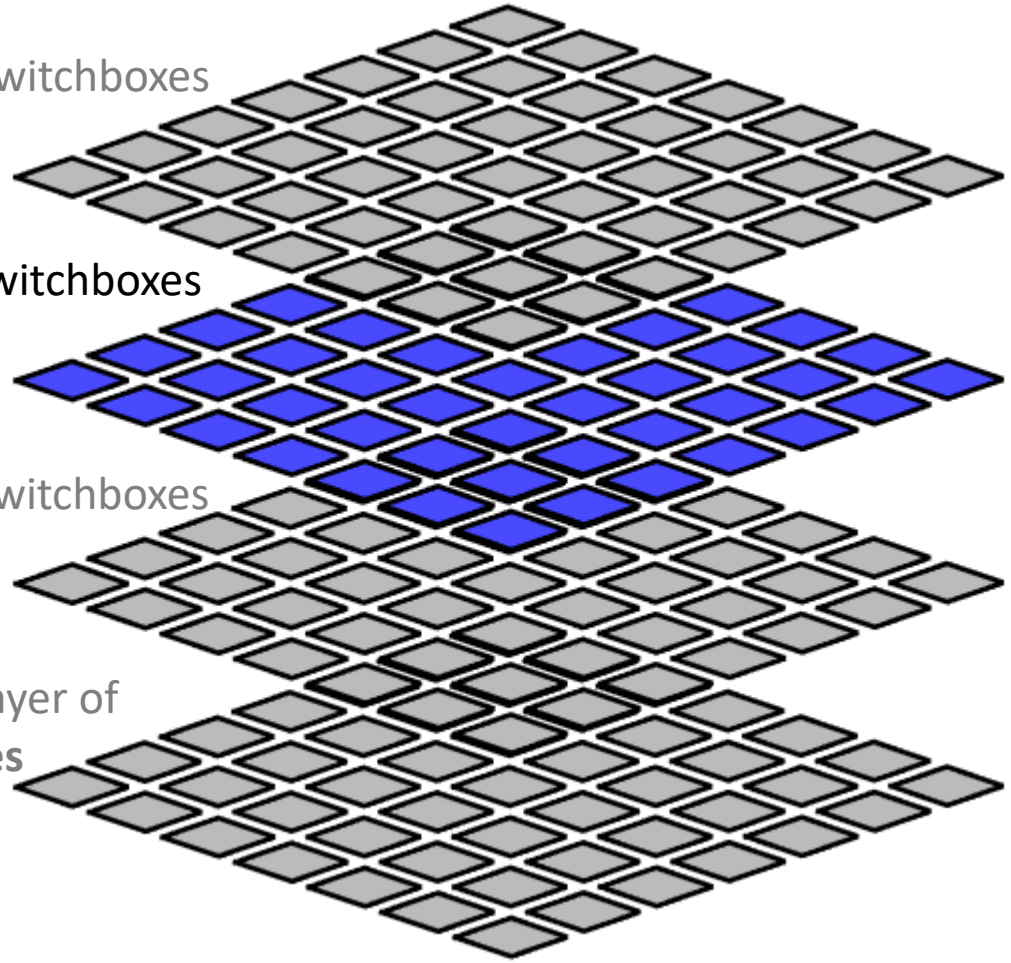




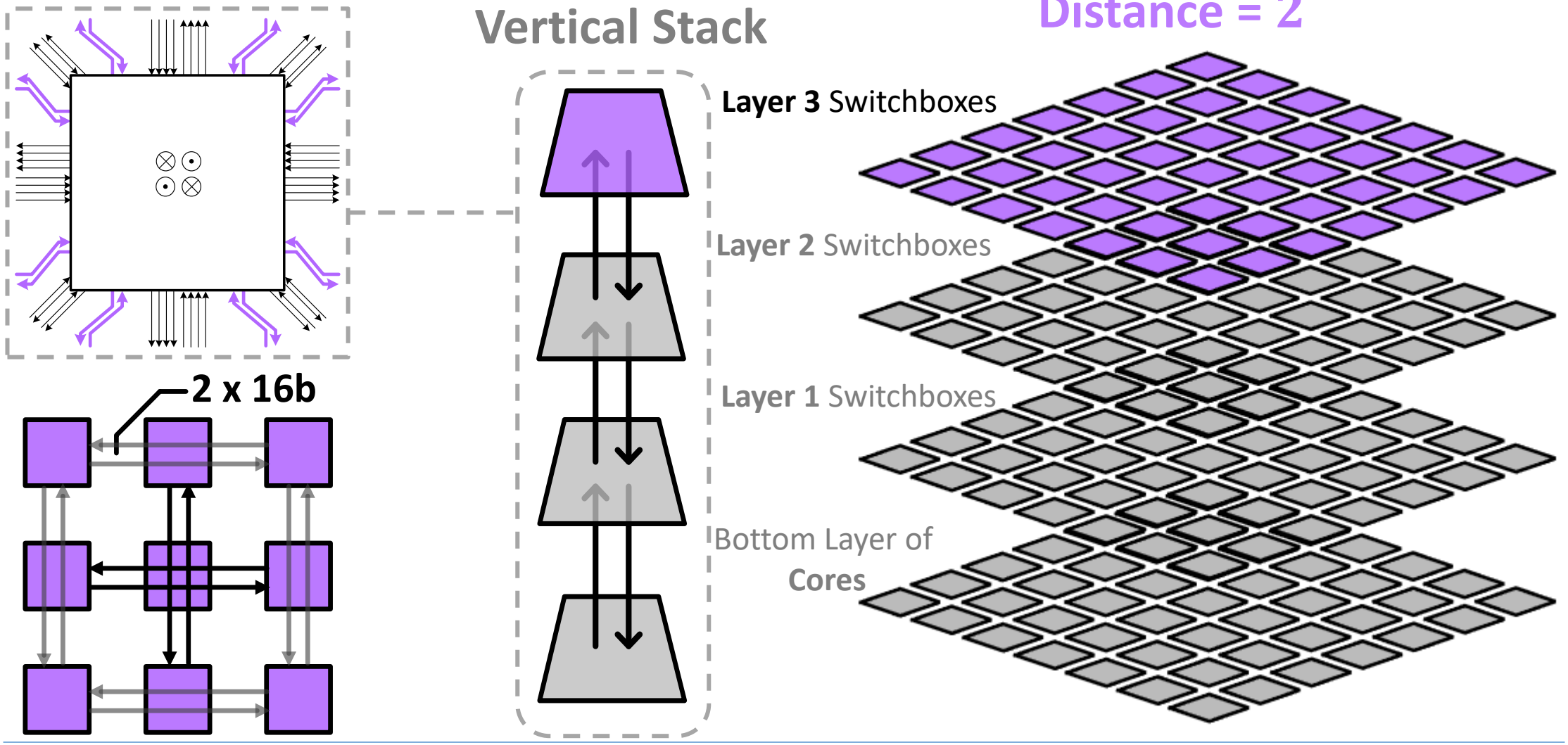
Vertical Stack



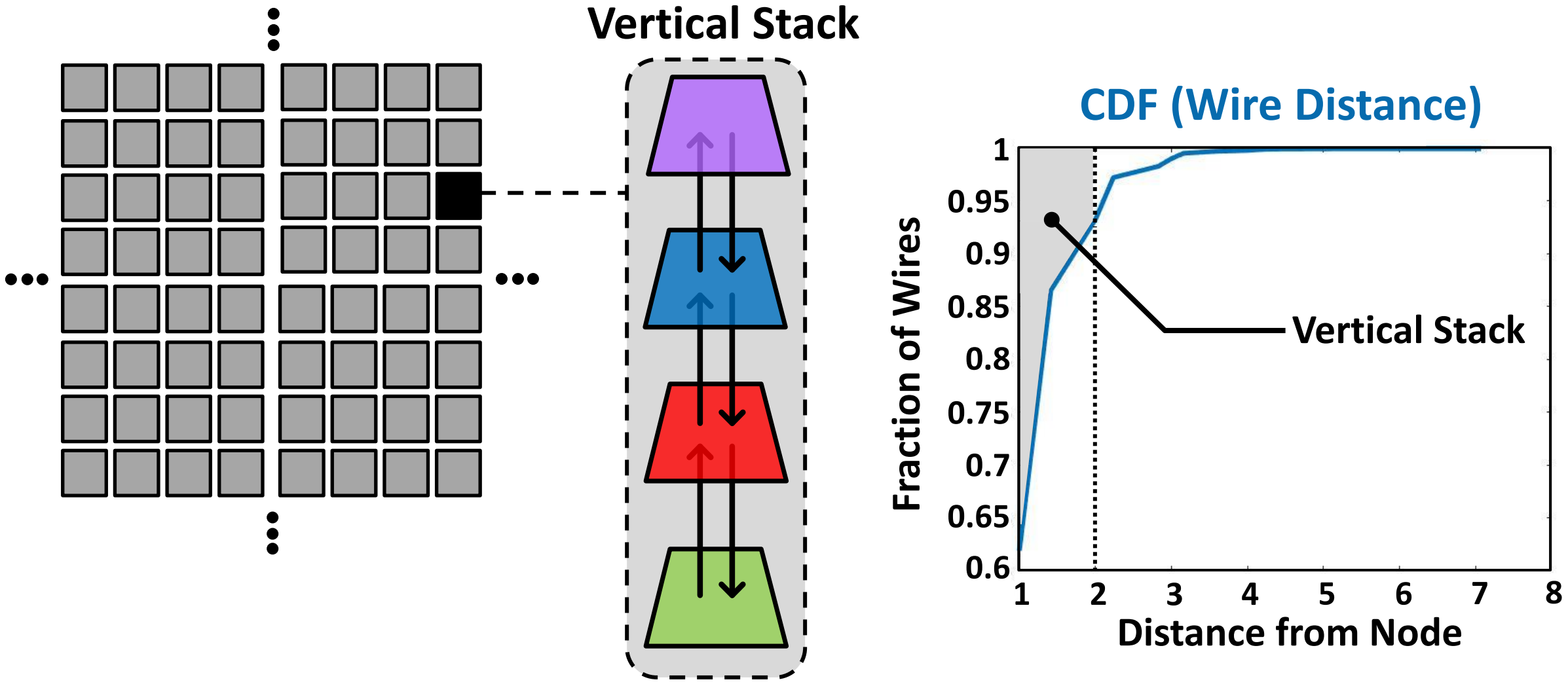
Distance = $\sqrt{2}$



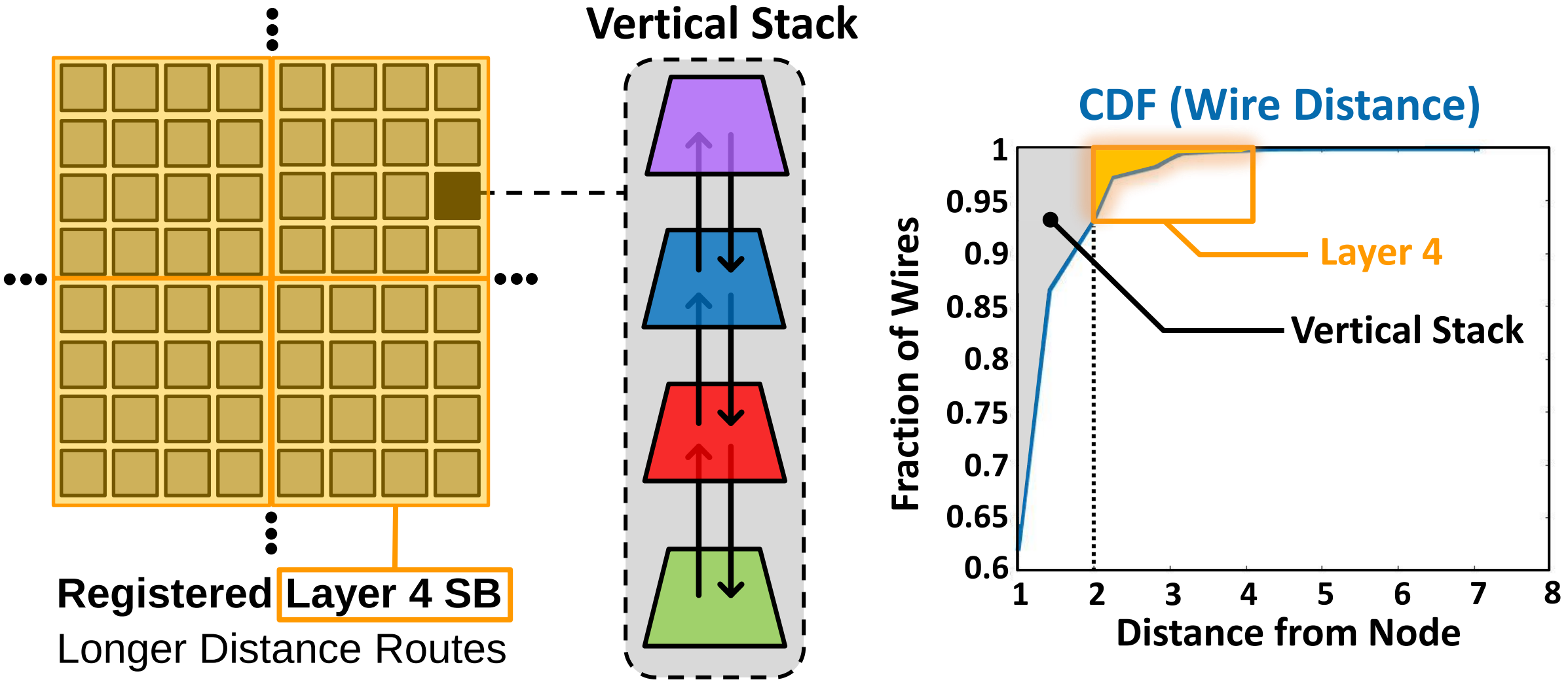
Layer-3 Interconnect (Distance = 2)



Layer-4 Interconnect

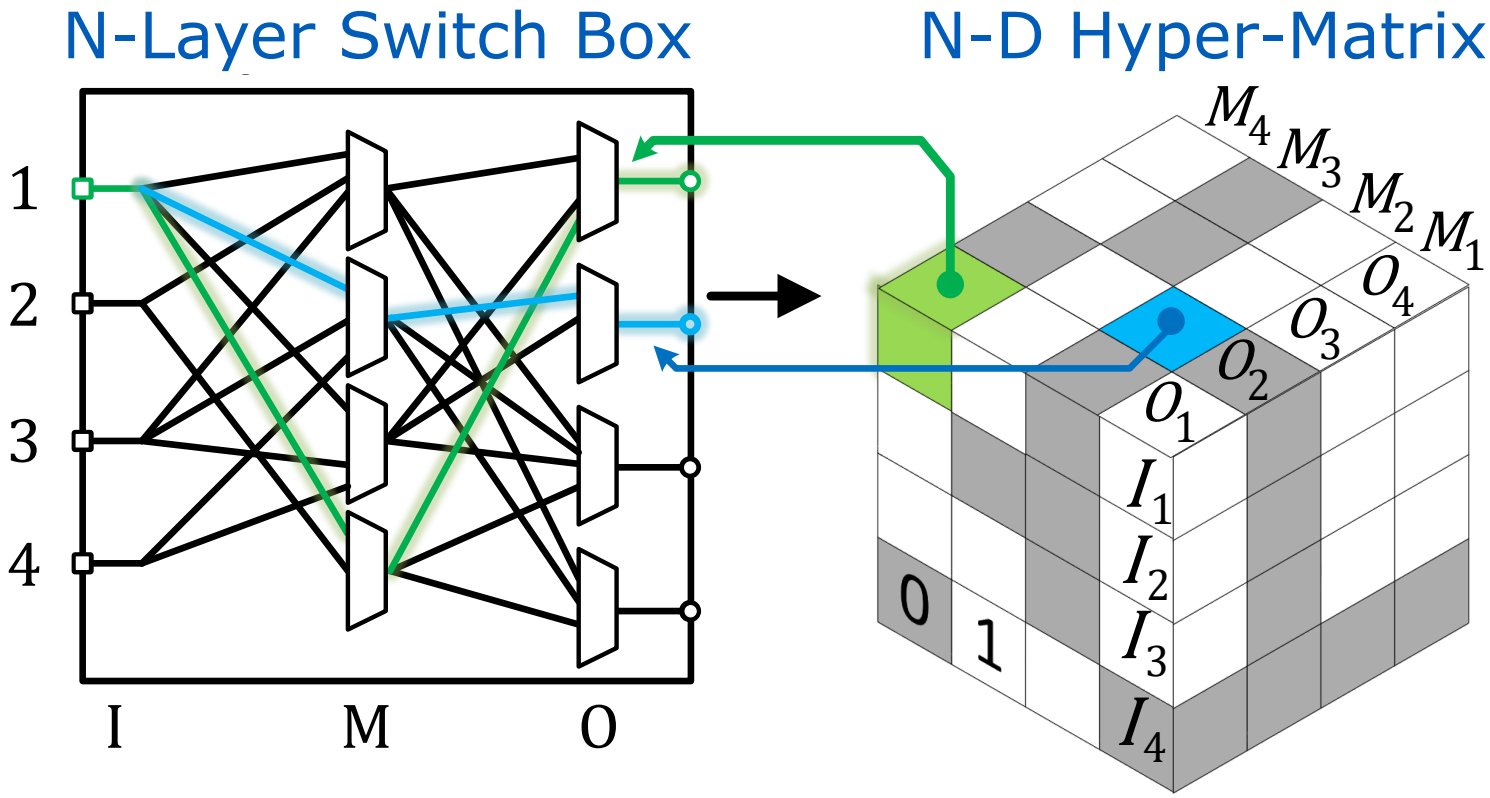


Layer-4 Interconnect



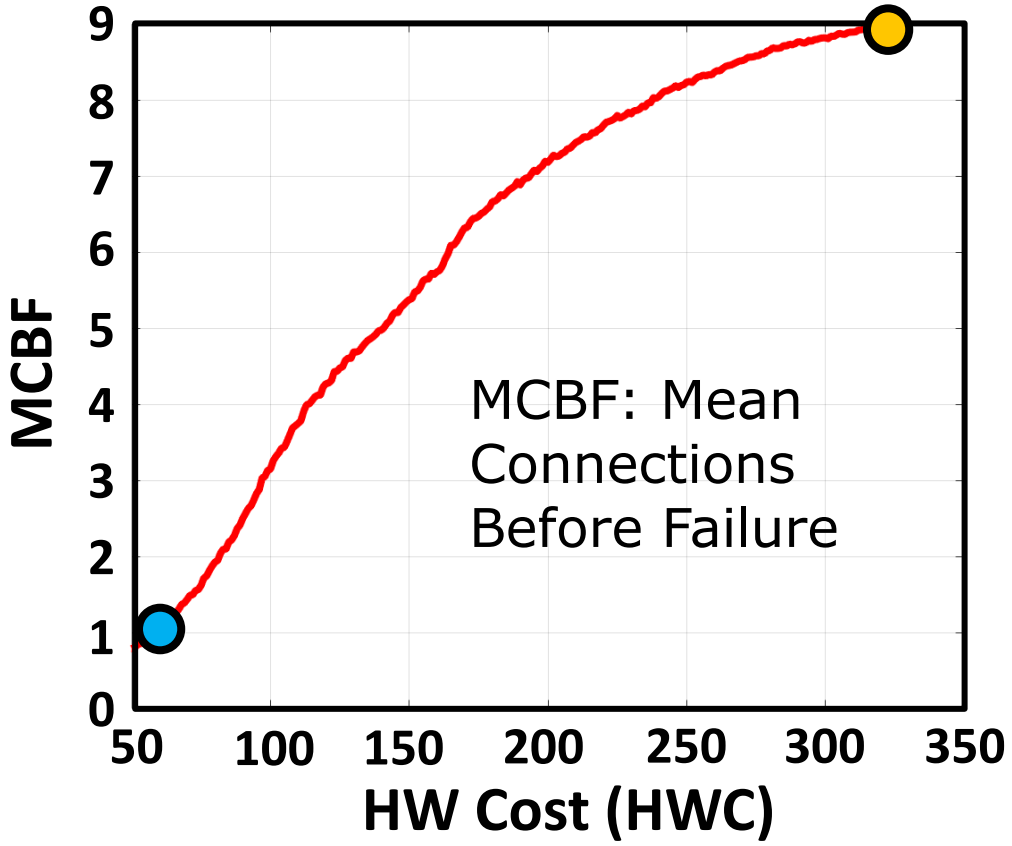
N-Layer Switch Box: Hyper-Matrix Model

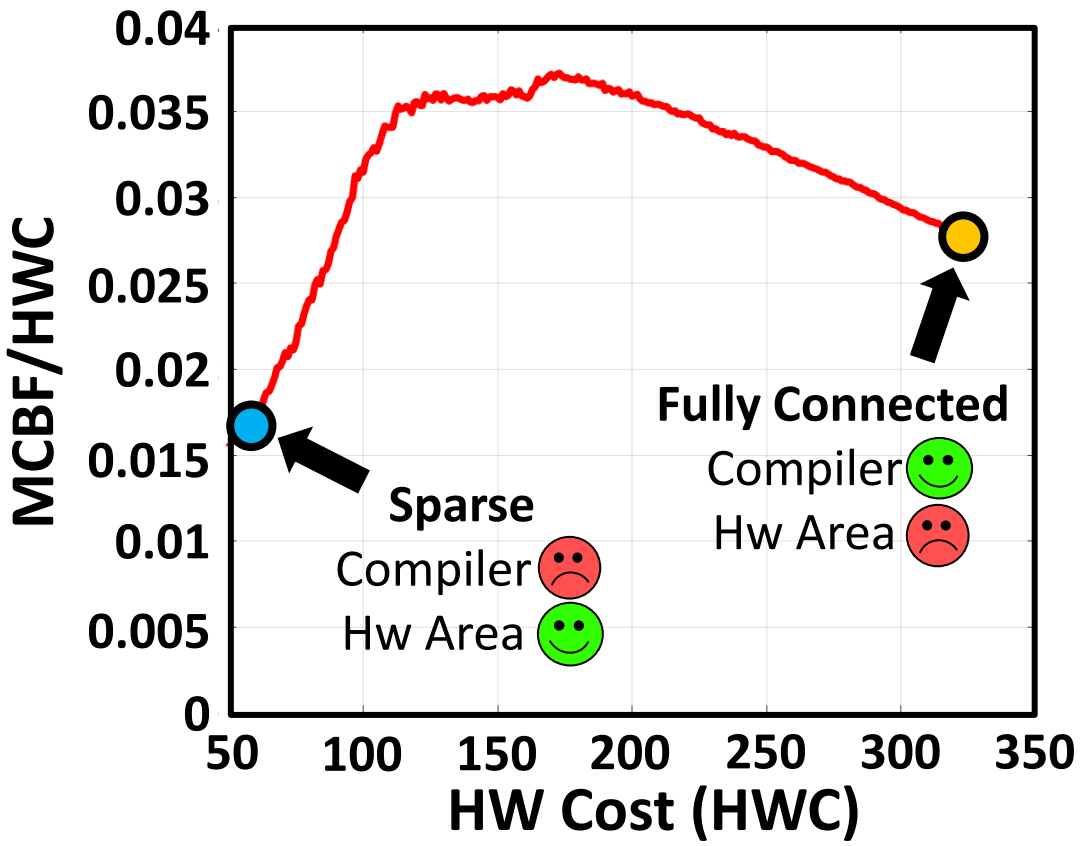
- Hyper-vector cross-correlation (HVCC) in each dimension (layer)
  - HVCC for a layer measures inter-dependencies of paths



DSE: Search Space Traversal

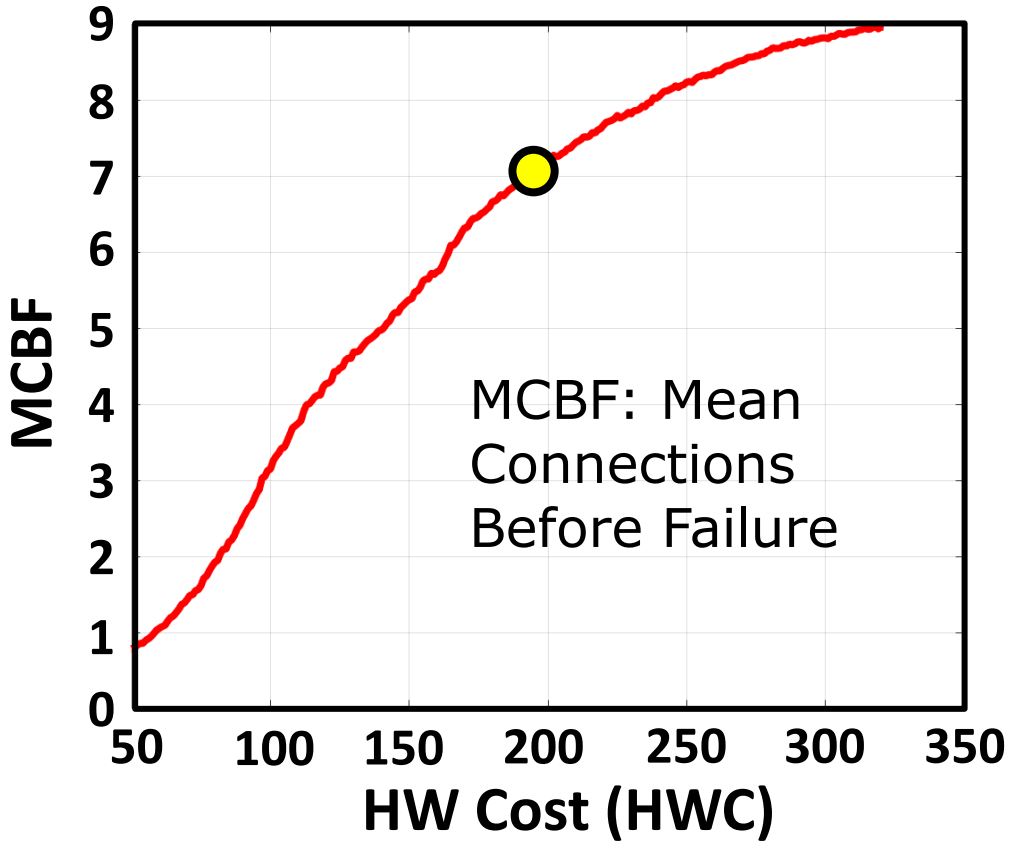
- MCBF & MCBF/HWC plotted against layer density for 3-layer SB

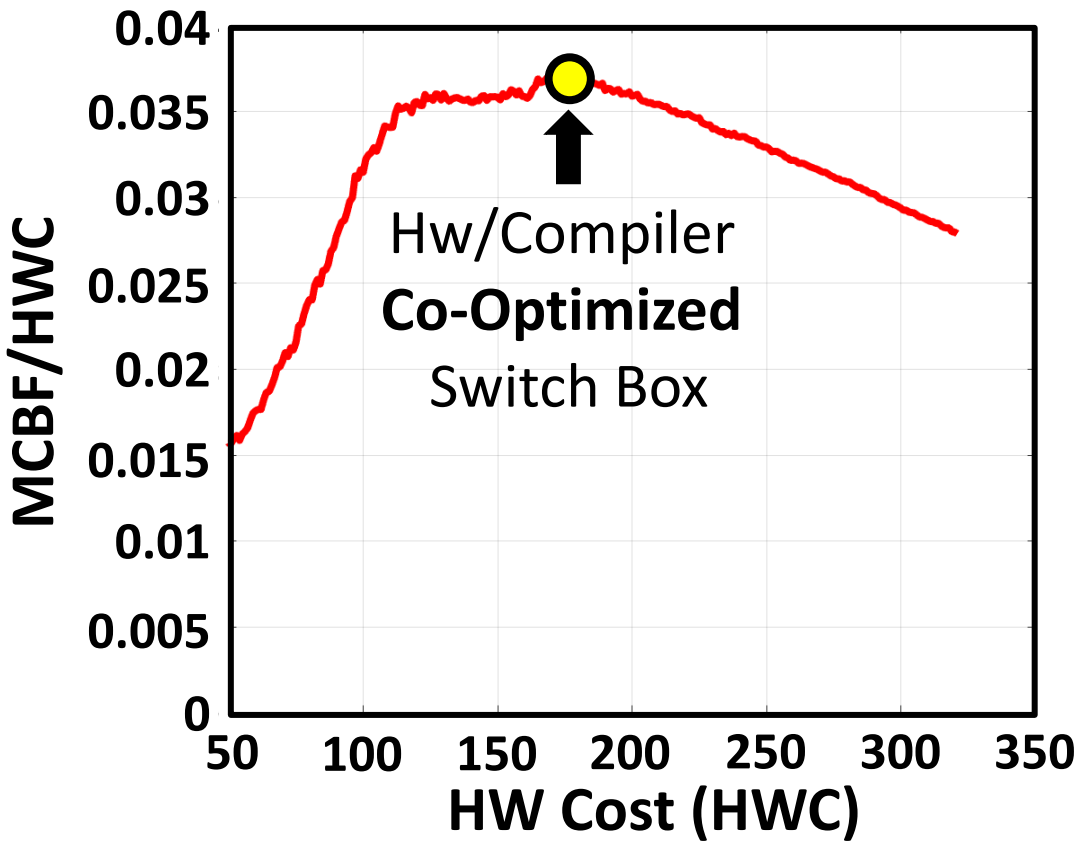




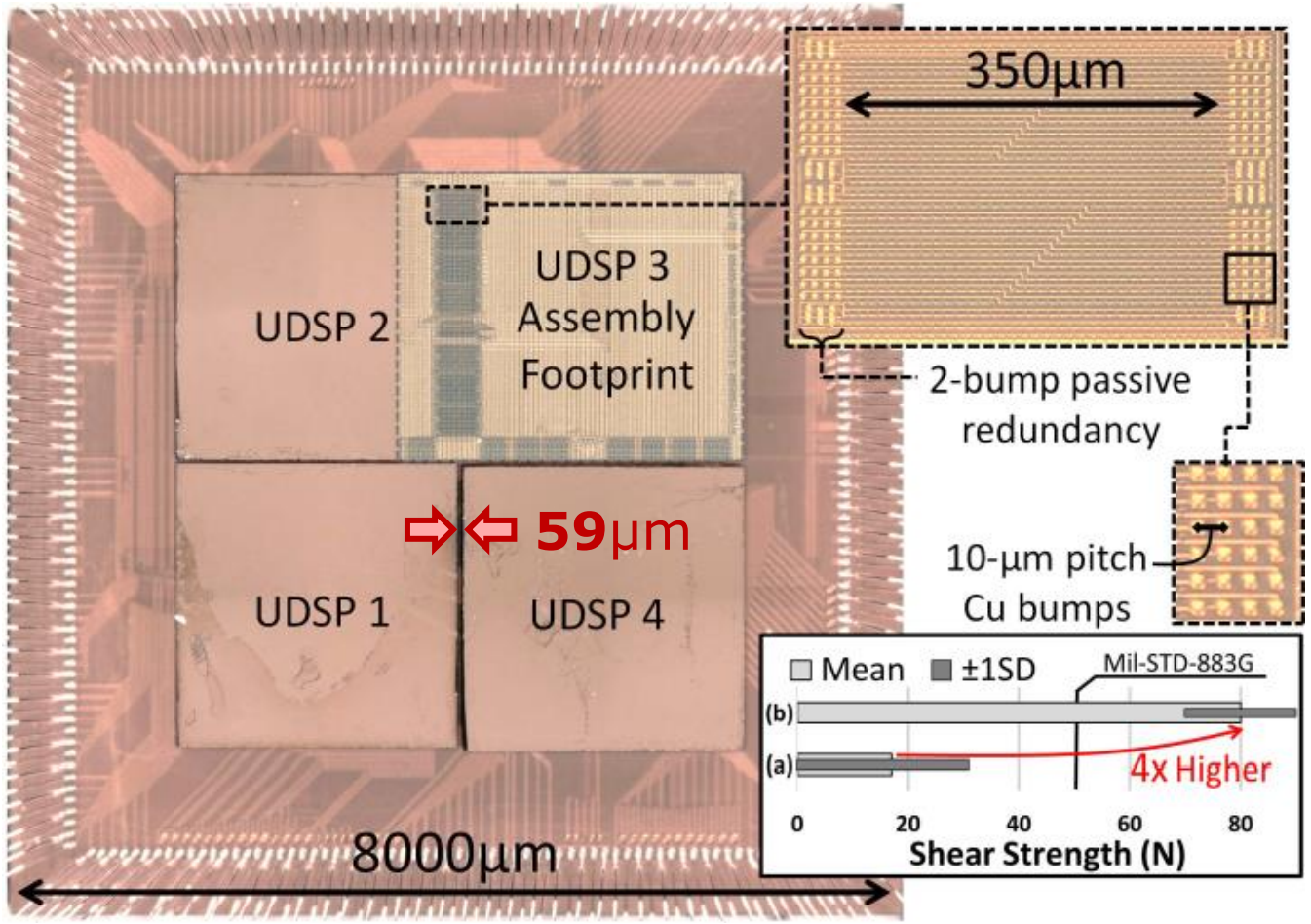
DSE: Maximizing Silicon Area Efficiency

- Sw/Hw balance is at the peak of MCFB/HWC





Si-IF Assembly Overview



- Four UDSP dielets
- 8mm x 8mm Si-IF

Si-IF interface

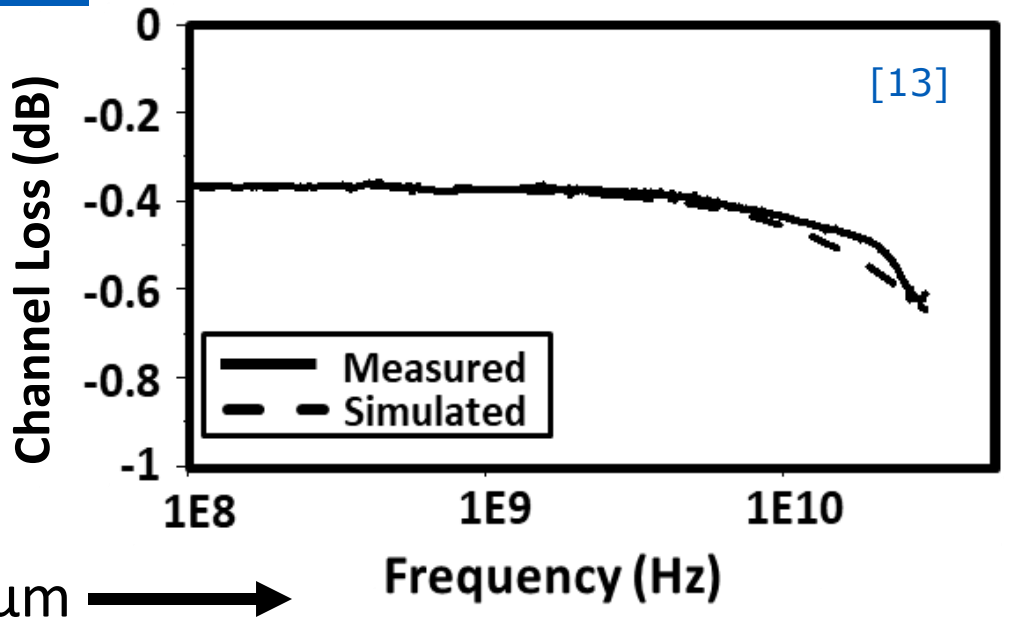
- 7,168 data pins
- 160 control + PLL pins
- 22,291 power/ground pins
- 2-layer routing

Assembly considerations

- Select known good dies
- Select known good Si-IF
- Die handling, cleaning, ESD
- Dielet alignment, bonding

Si-IF Characteristics

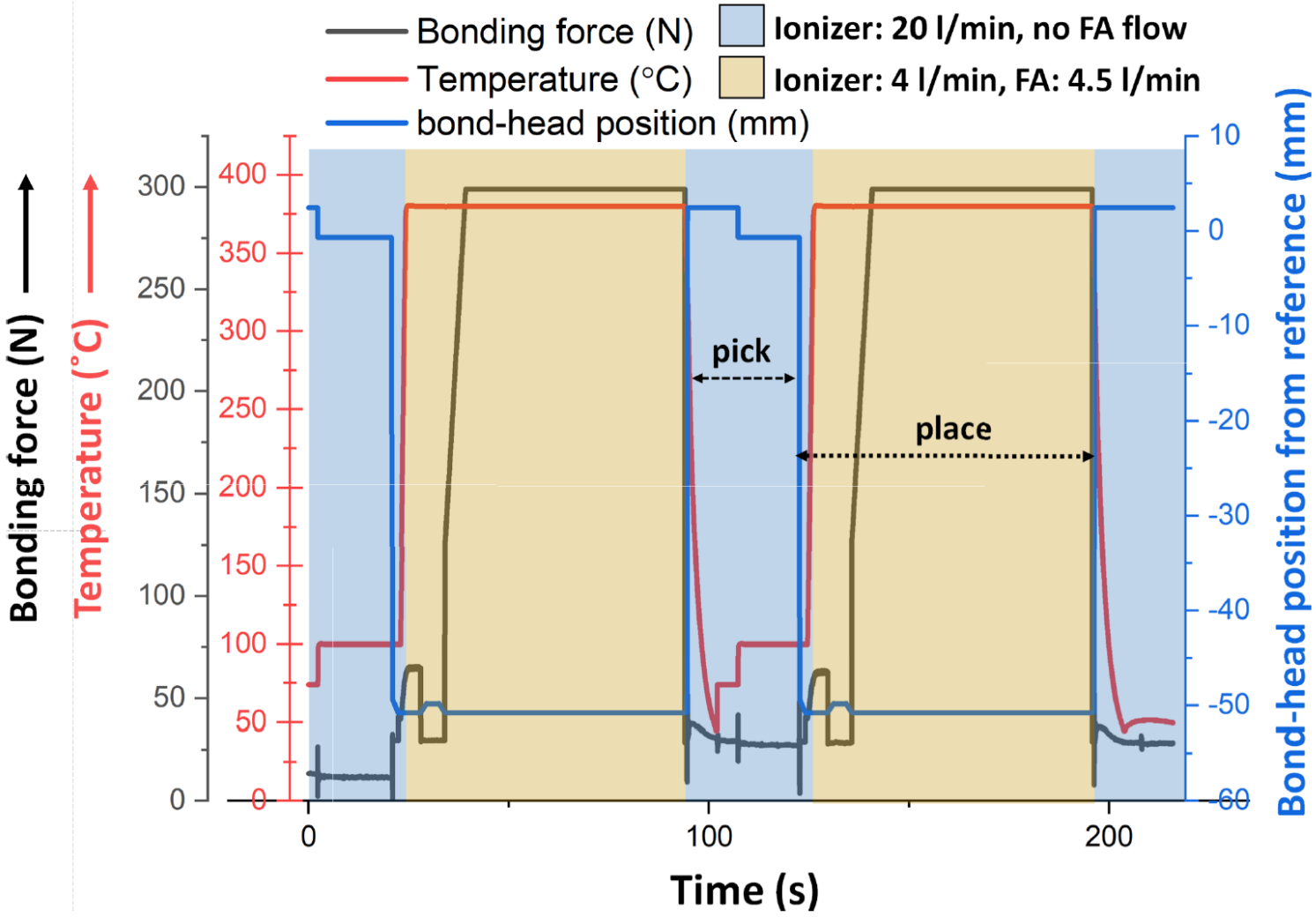
- Low loss at 10 GHz
- <10 ps RTT, negligible ISI



Si-IF Process and Assembly

- UDSP dielet powered on to verify Clk tree and shift-registers
  - Low-freq Clk applied using a probe station
- Dice defect-free Si-IF sites for assembly
  - Template-based wafer scan for repeated patterns
- Dielets assembled on Si-IF using direct Cu-Cu TCB
  - In-situ formic acid (FA) vapor treatment
- Ionizers on the bonding tool to ensure an ESD-safe assembly
  - Default **20 l/min** flow interferes with the FA vapor flow of **4.5 l/min**
  - Leads to inadequate cleaning of Cu pads, inferior bonding quality
    - Shear strength **<20N**, below MIL-SPEC 883G of **50N** for 2mm x 2mm dielets
  - Solution: the ionizer flow rate is optimized w.r.t. FA vapor time

Bonding Parameters of UDSP Dielets



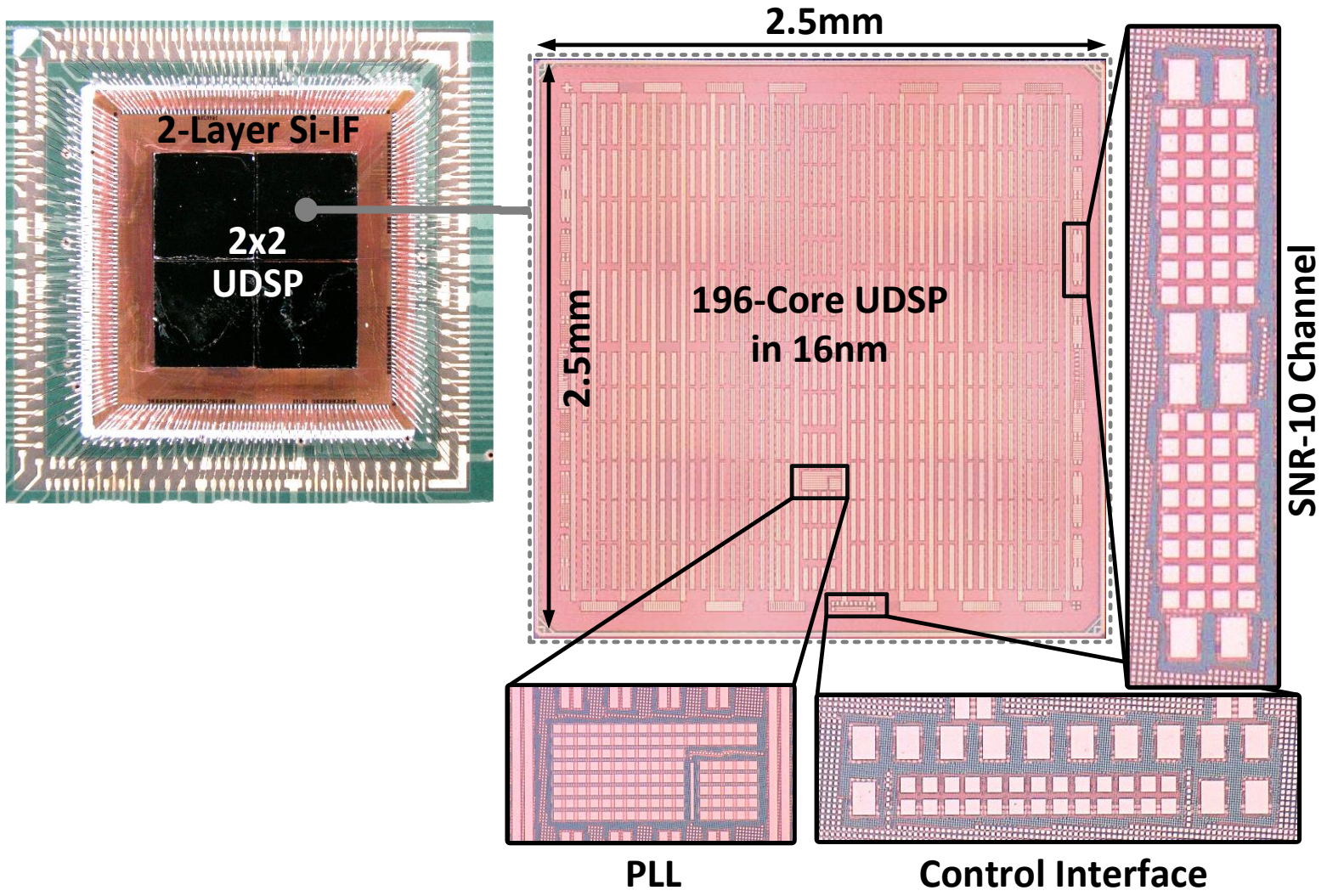
Bonding parameters

- Temperature
- Bonding force
- Bond-head position

Key insights

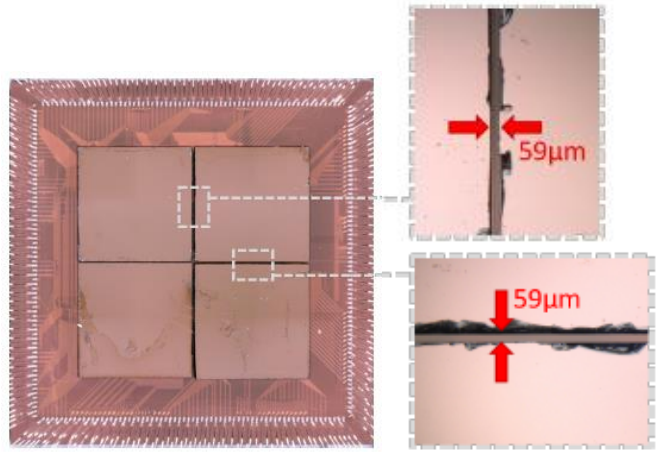
- FA flow of **4 l/min** during placement: bond strength **80N**
- No need to shut down ionizer (no ESD events on all 7,168 links tested)

SNR-10 I/O Channel



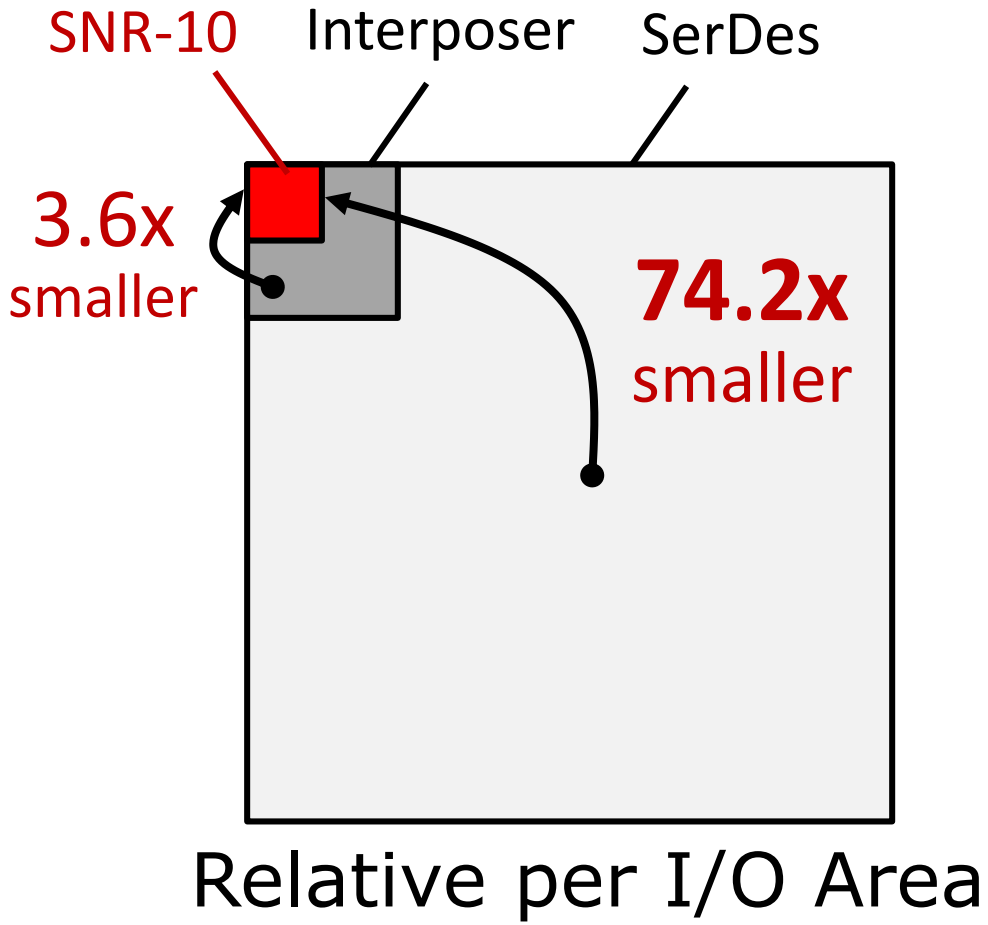
- I/O bump pitch: **10 µm***
- EMIB: 55 µm
- SerDes: 130 µm

* **9.8 µm** due to optical shrink



Challenges in Si-IF Link Design

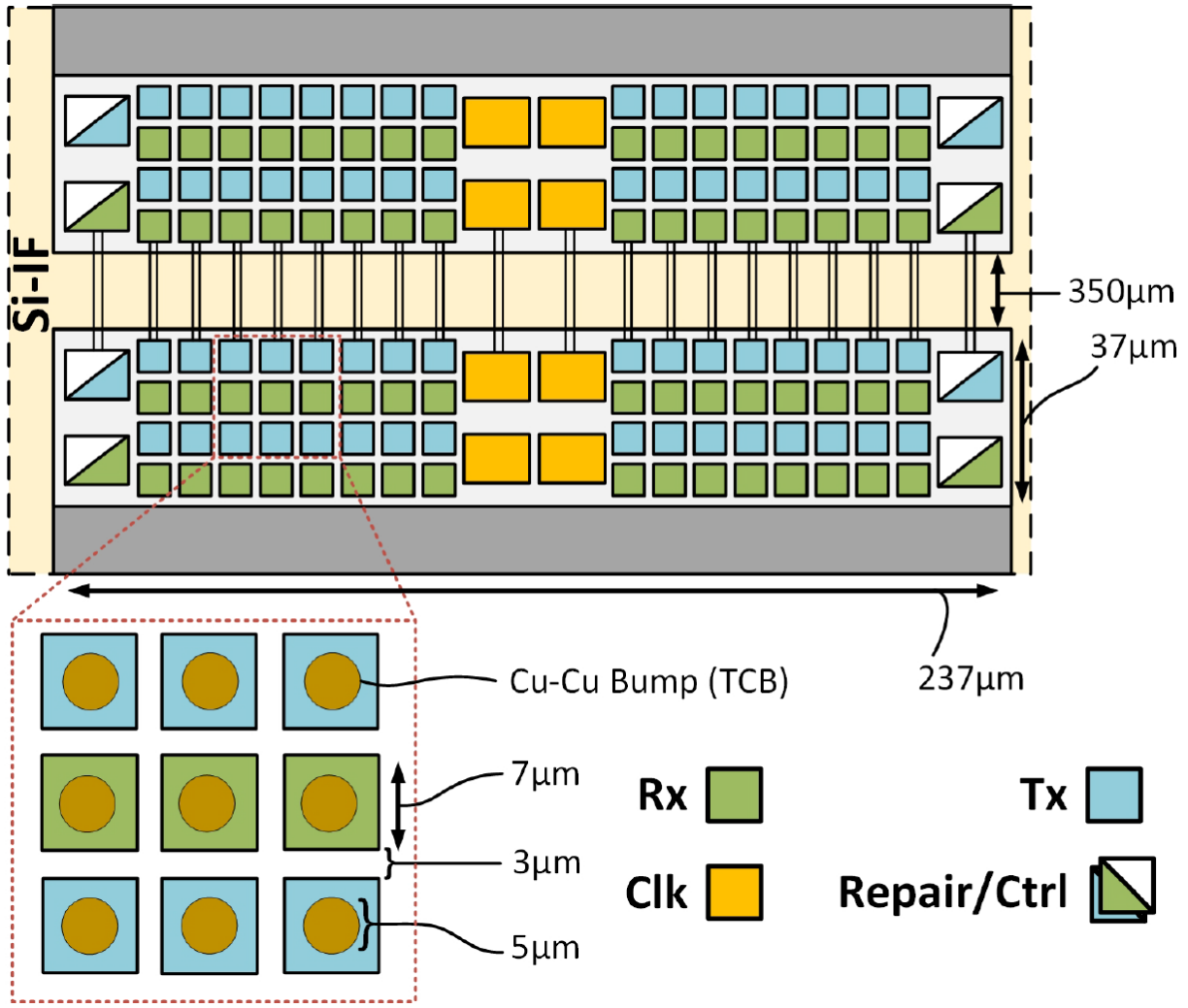
- Available area per I/O
  - SerDes[14] → 10,175 μm²
  - Interposer[15] → 500 μm²
  - Si-IF/SNR-10 → 137 μm²
- Required throughput
  - 1.1 Gbps/pin
- Logical redundancy
  - Mfg and integration defects



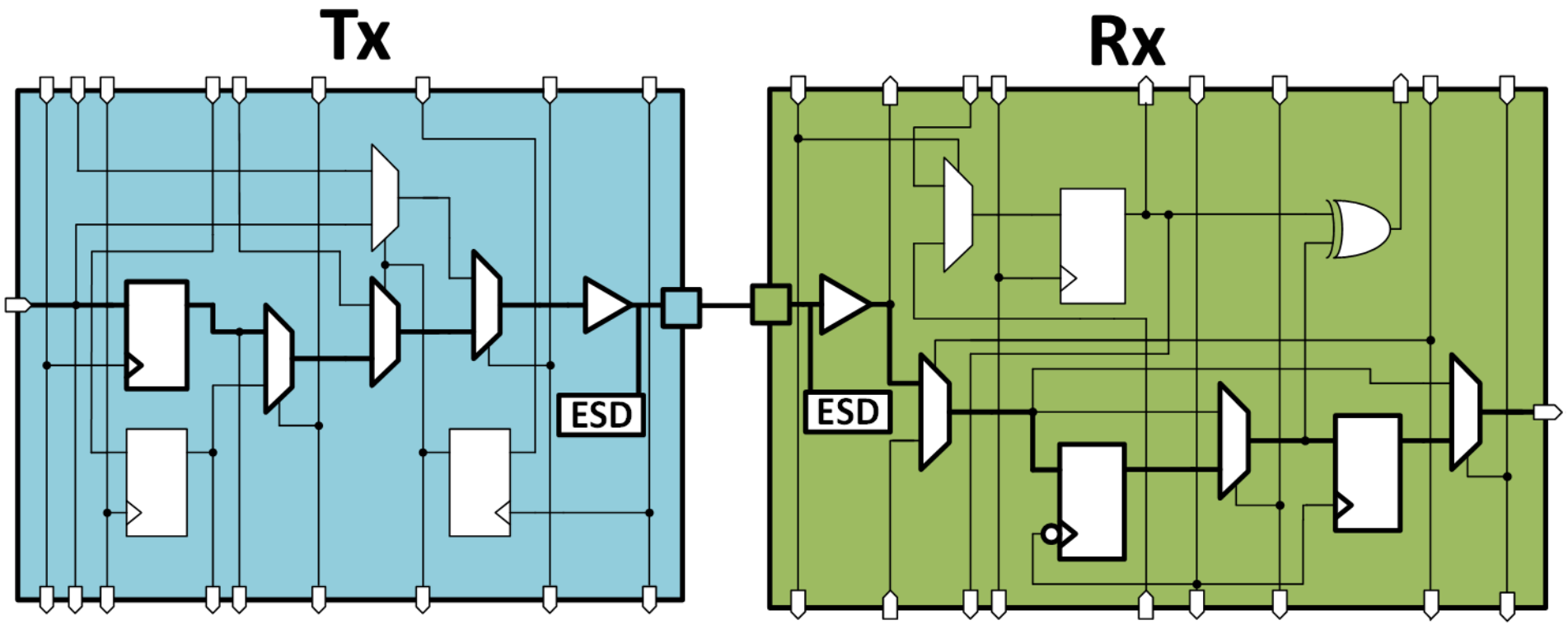
[14] M. Lin, T. Huang, JSSC, 2020.
[15] J. M. Wilson, ISSCC, 2018.

Streaming Near Range (SNR) Channel

- Uni-directional I/Os
- Clock-forwarded
- **64-bits / channel**
- Amortized Clk correction and transfer
- Redundancy
- Minimal ESD
- Minimal handshake
- Sync/Async modes
- 3 Clk cycles of latency
- FIFO for CDT

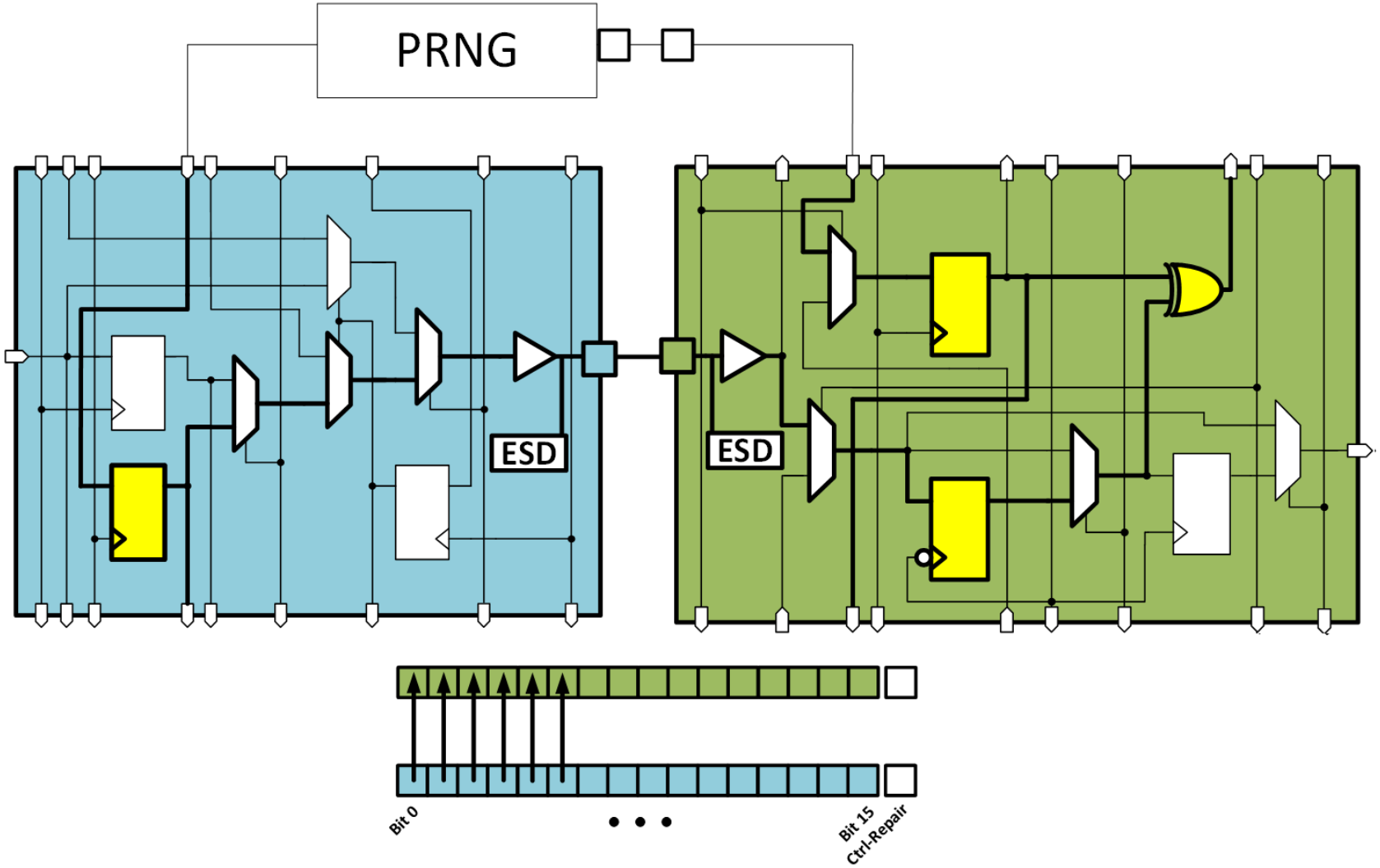


Tx/Rx Cells

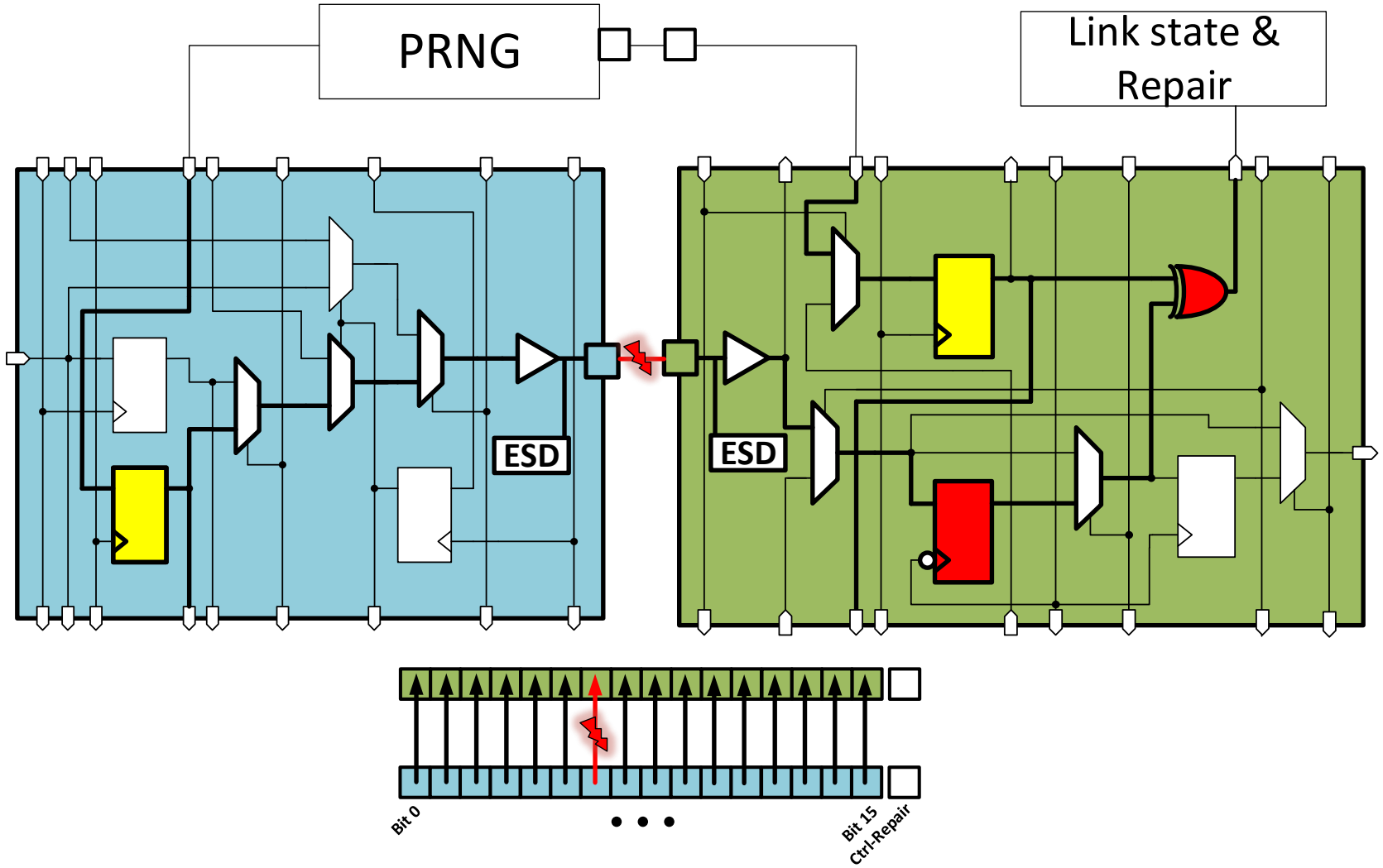


- Tx pads use minimal additional ESD (64x drive buffer diodes)
- Rx pads use relatively larger ESD protection diodes

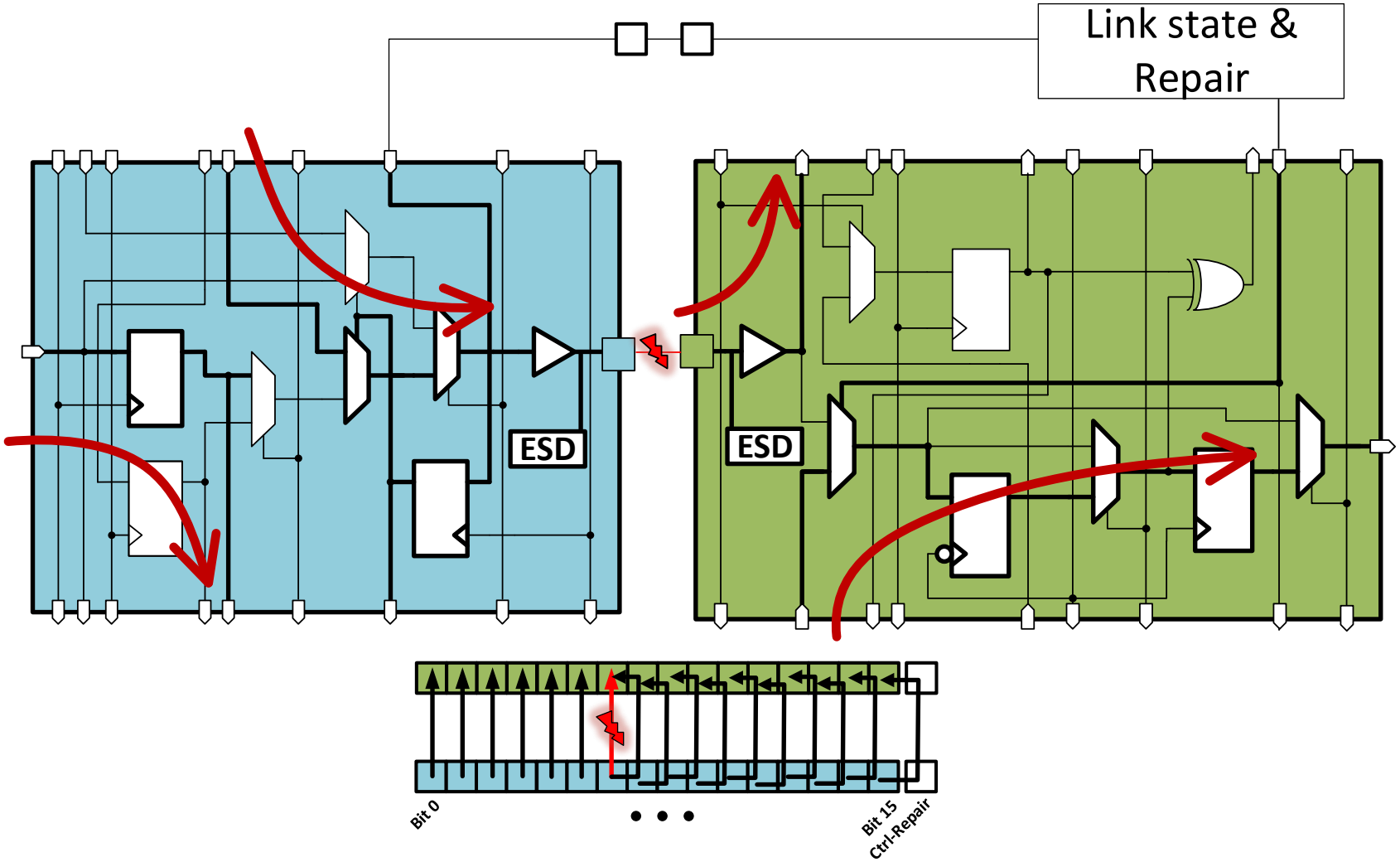
Redundancy and Repair: Check



Redundancy and Repair: Detect



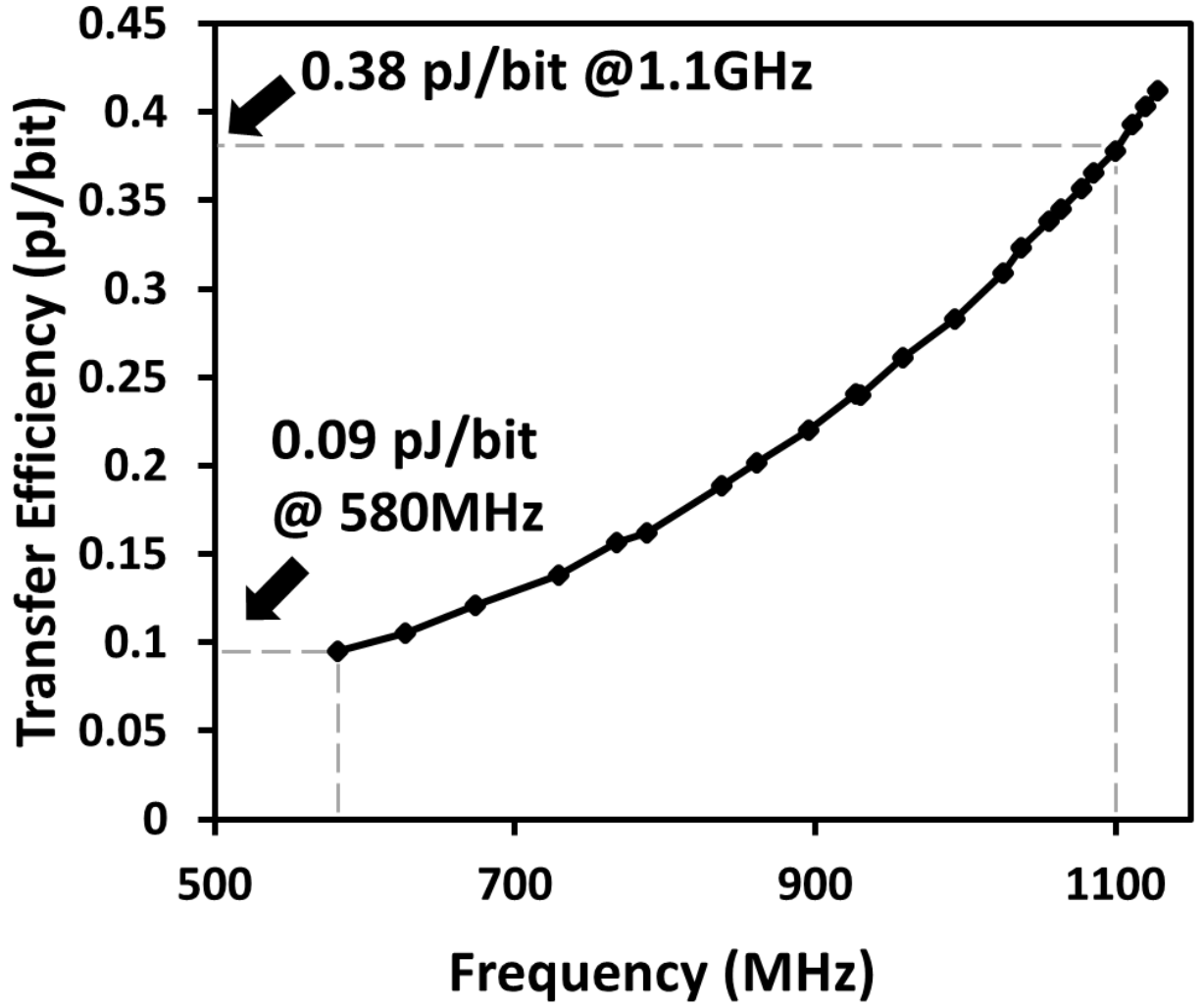
Redundancy and Repair: Repair



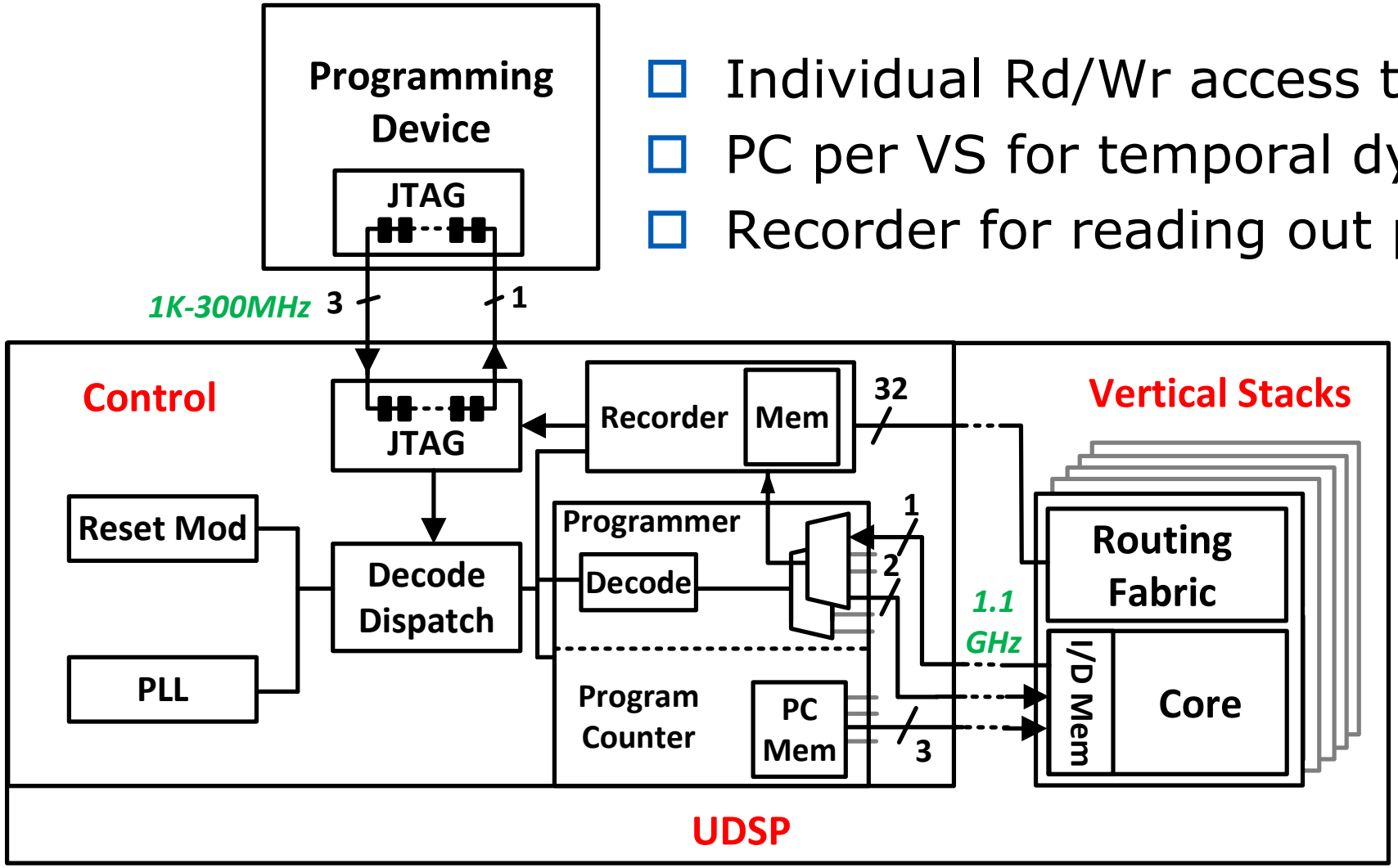
SNR-10 vs. State-of-the-Art

Reference	[This work]	CICC'21	JSSC'20	ISSCC'18
Techology	16nm	16nm	7nm	16nm
Package Substrate (# layers)	Si-IF (2)	EMIB (4)	CoWoS (15)	MCM/PCB
Bump Pitch (μm)	10	55	40	150
Reach (μm)	350	3,000	500	80,000
Data Rate (Gbps/pin)	1.1	2	8	25
Voltage (V)	0.8	0.9	0.3	0.95
Energy Effc. (pJ/bit)	0.38	0.83	0.56	1.17
I/O Area Density (μm²/bit)	137	203	500	10,175
Max Shoreline BW (Gbps/mm)	297	256	1600	292
Layer BW (Gbps/mm/layer)	149	64	107	25

SNR-10 Transfer Efficiency vs. Voltage

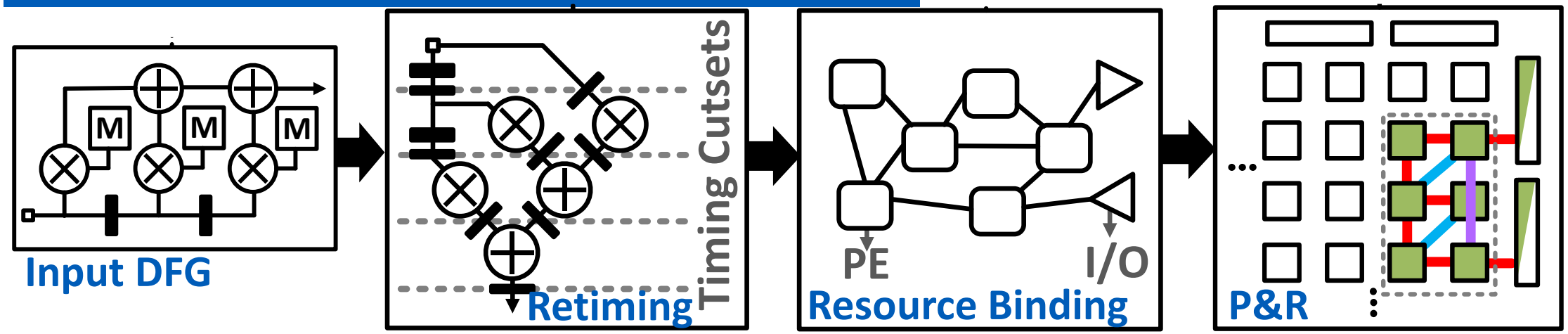


UDSP Control Logic

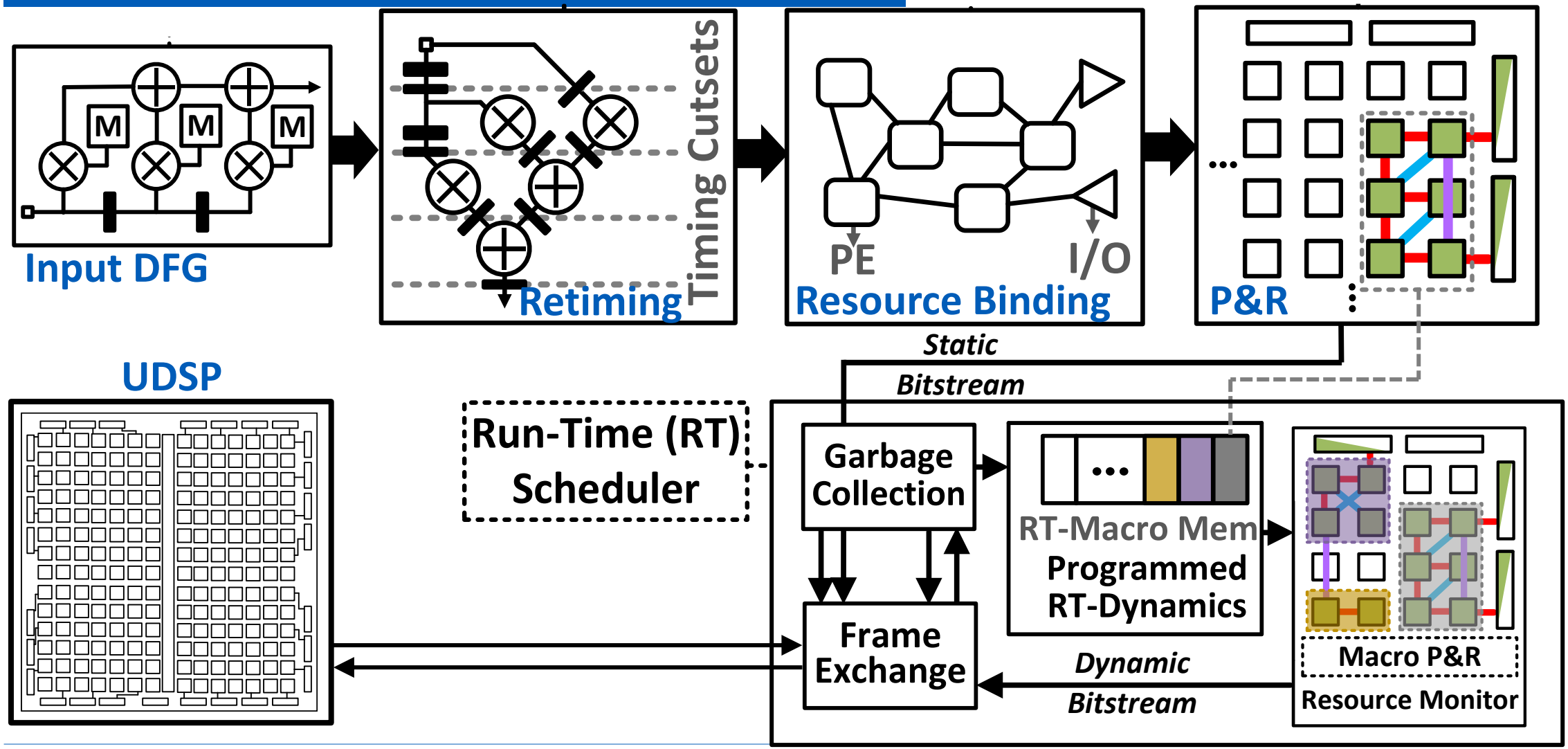


- Individual Rd/Wr access to VS I/D Mem
- PC per VS for temporal dynamics
- Recorder for reading out programs/data

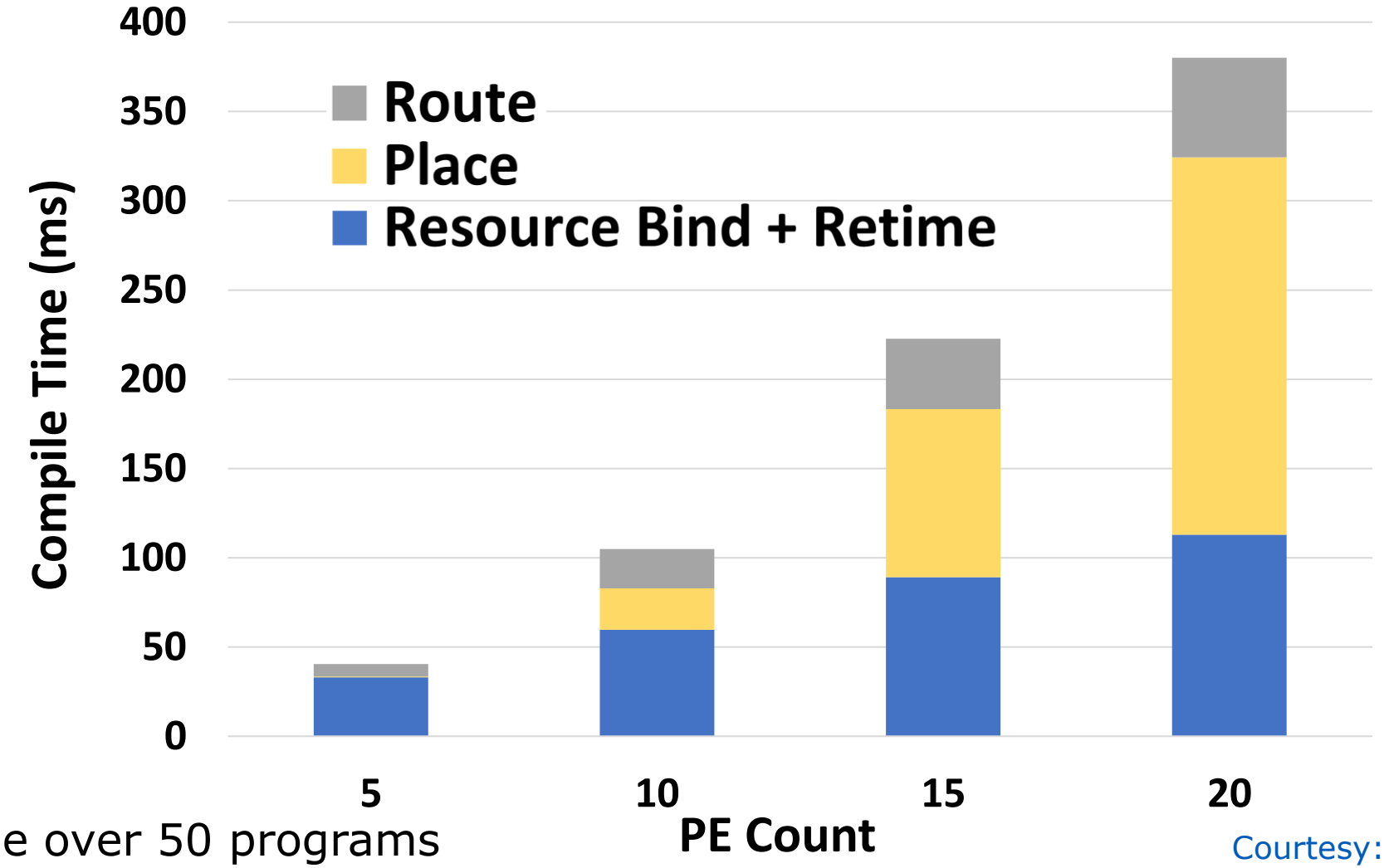
UDSP Compile Flow



UDSP Dynamic Reconfiguration



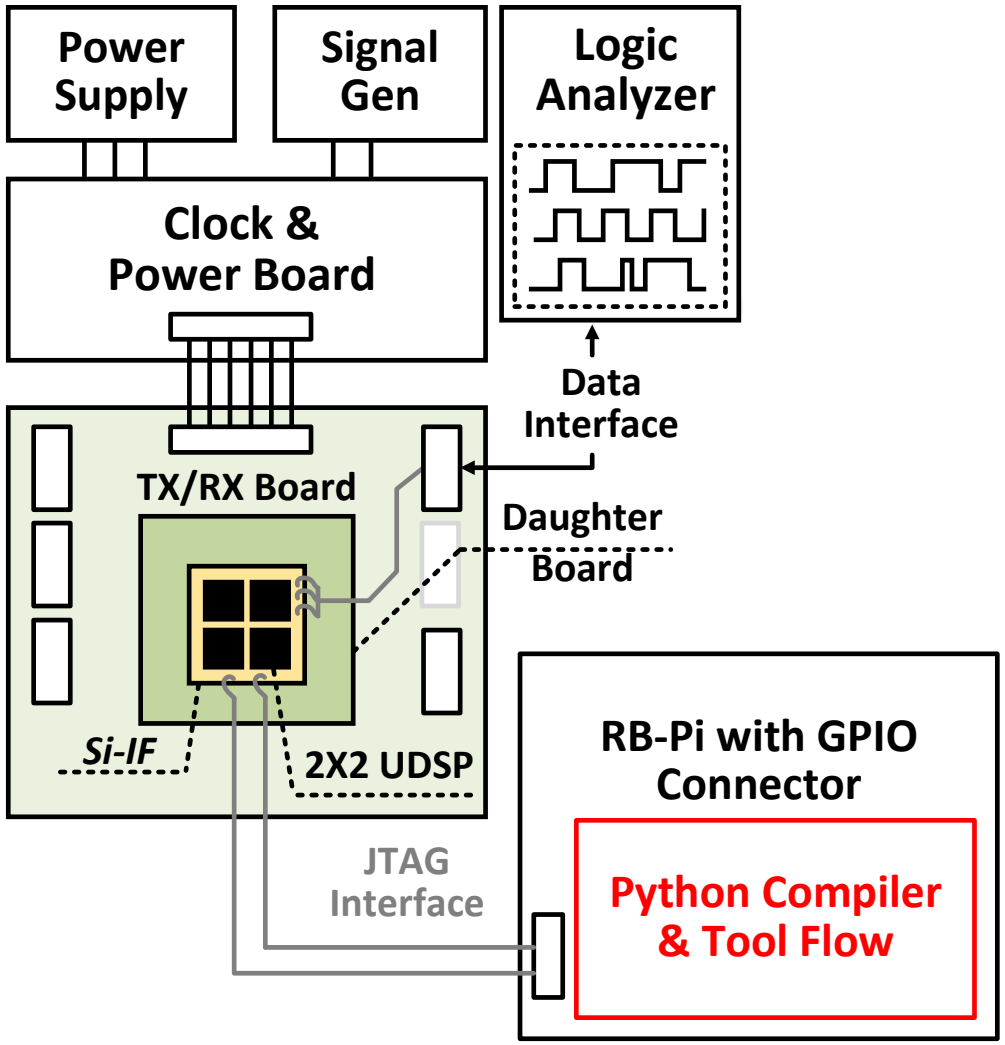
UDSP Compile Time Breakdown

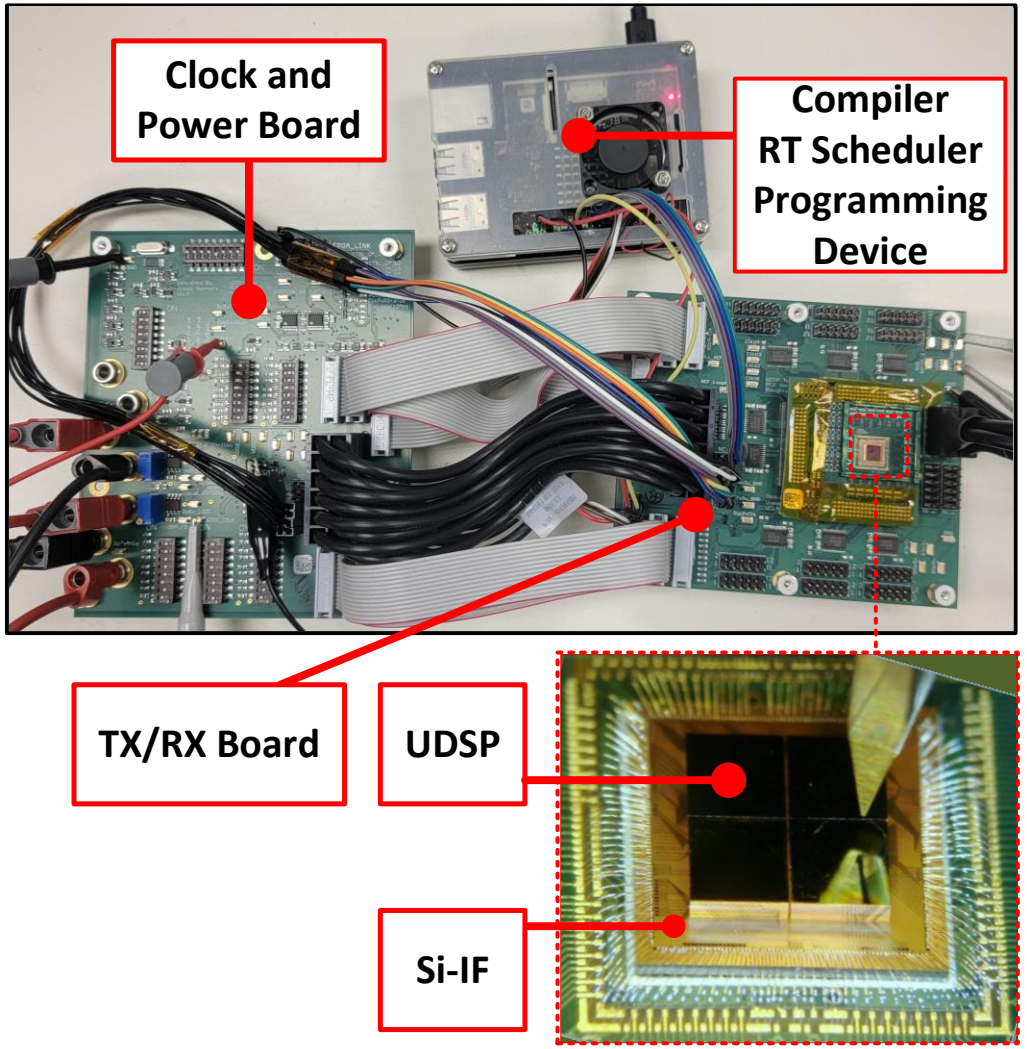


Average over 50 programs

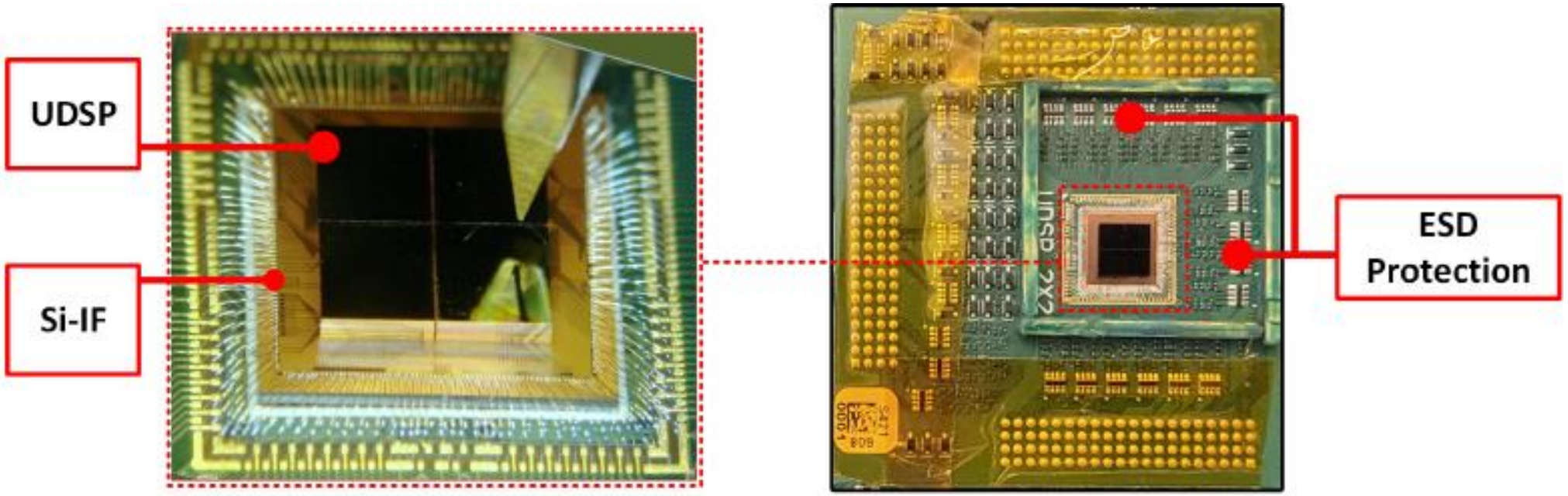
Courtesy: Tim Ling, UCLA

UDSP 2x2 Test Setup





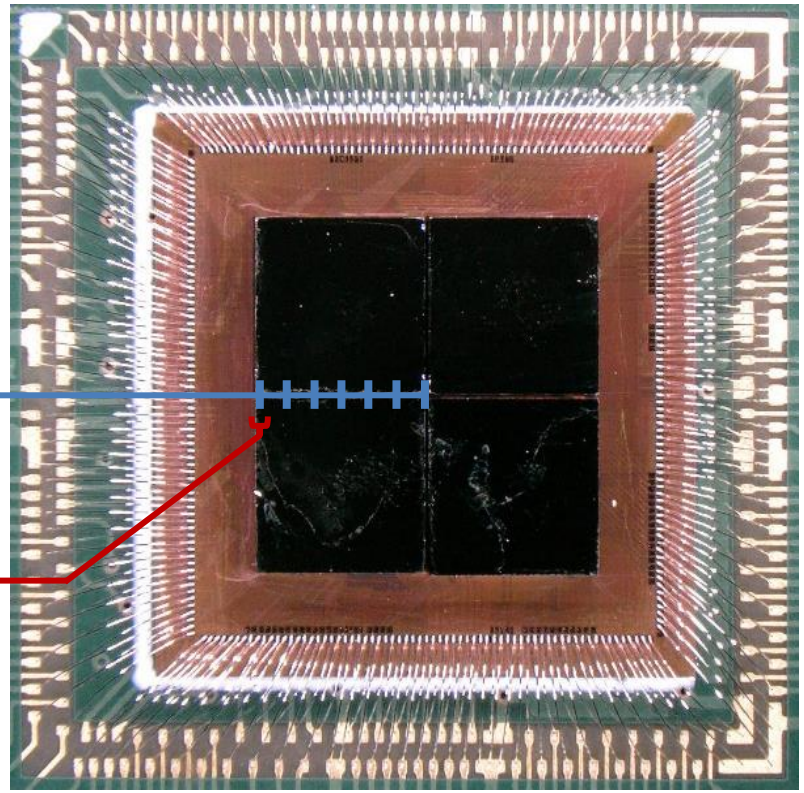
Si-IF Assembly on PCB



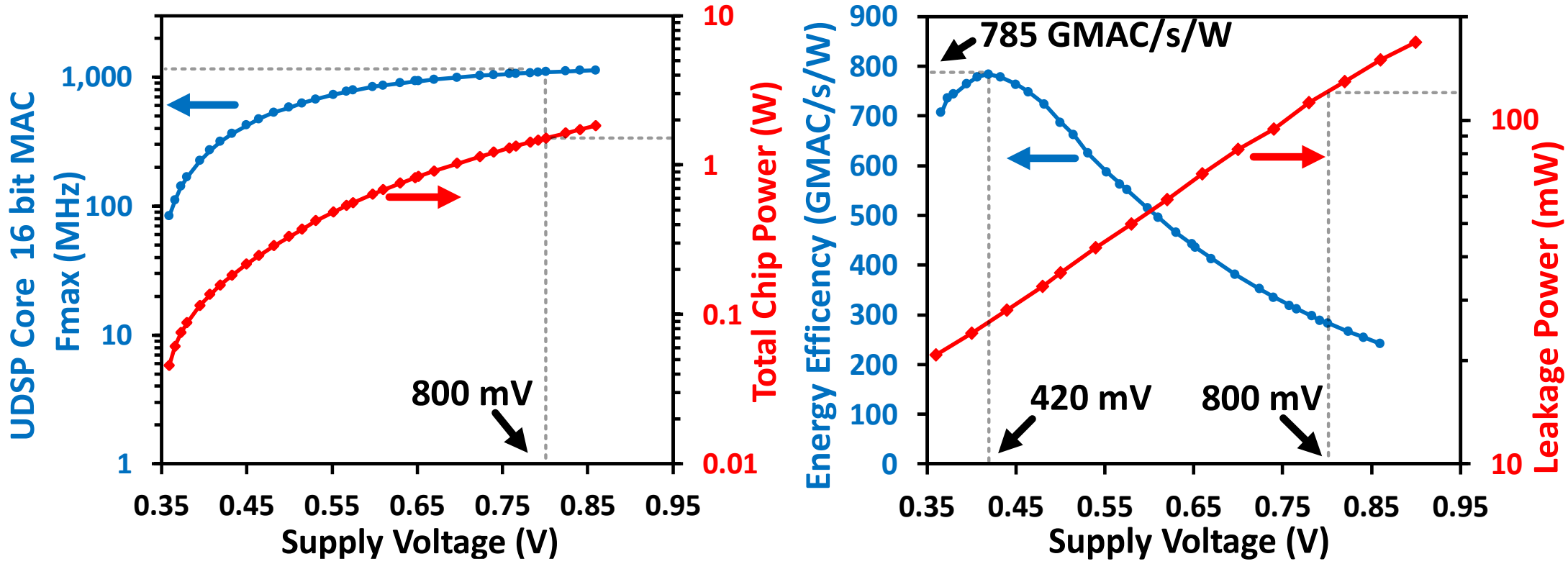
- No internal ESD events due to Si-IF assembly
  - Over 6,000 pads tested across two samples
- Wire-bonded Si-IF sample on a daughter board with ESD
  - No ESD events observed across 250 pins tested

SNR-10 Performance Summary

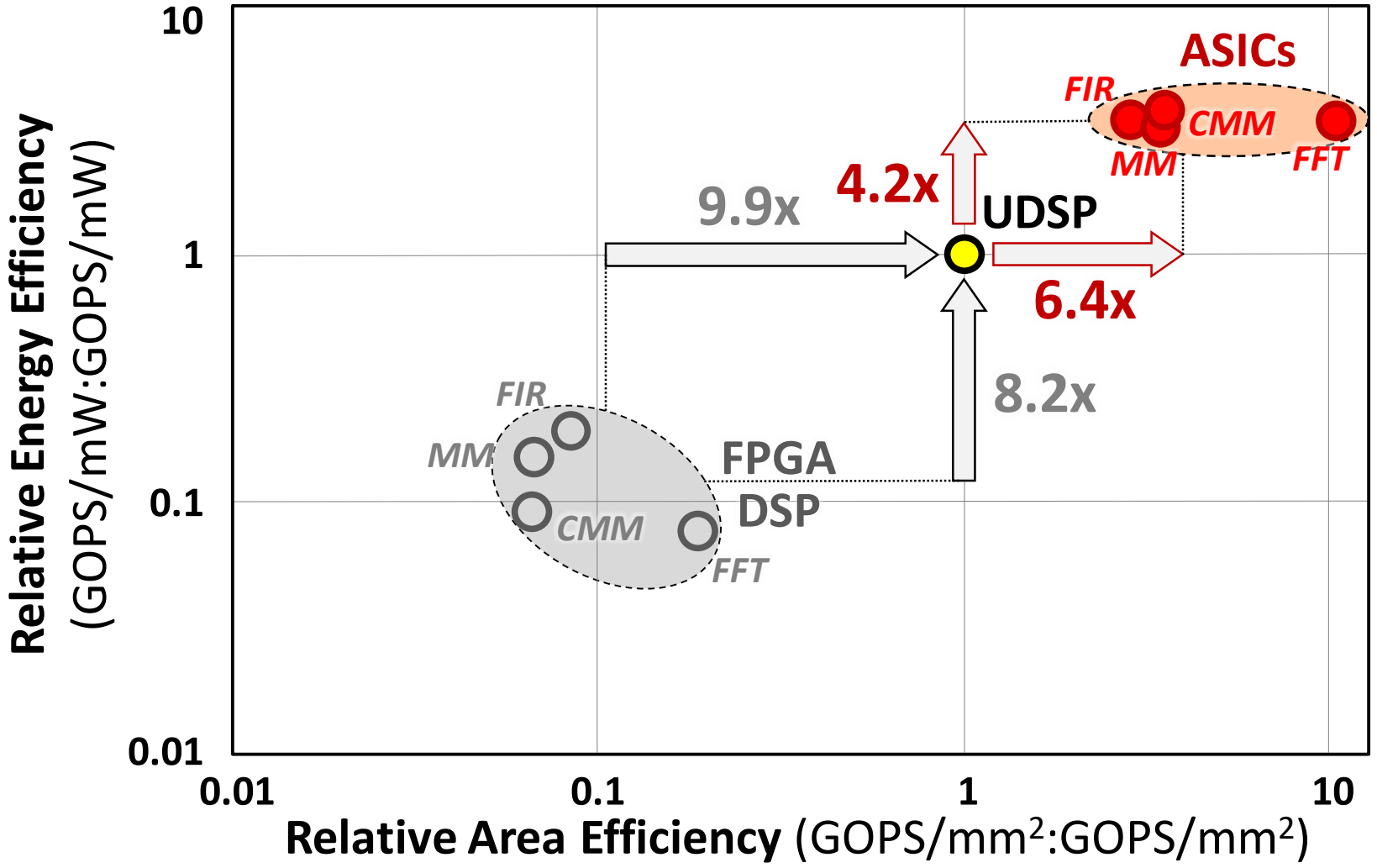
- No internal ESD events due to Si-IF assembly
- 7 64-b channels per side
  - 448 bits per side
- Fmax: 1.1 GHz
- Inter-die BW: 493 Gbps
- Max BW density: 297 Gbps/mm



UDSP Fmax, Power & Efficiency vs. V_{DD}



Bridging the Efficiency Gap



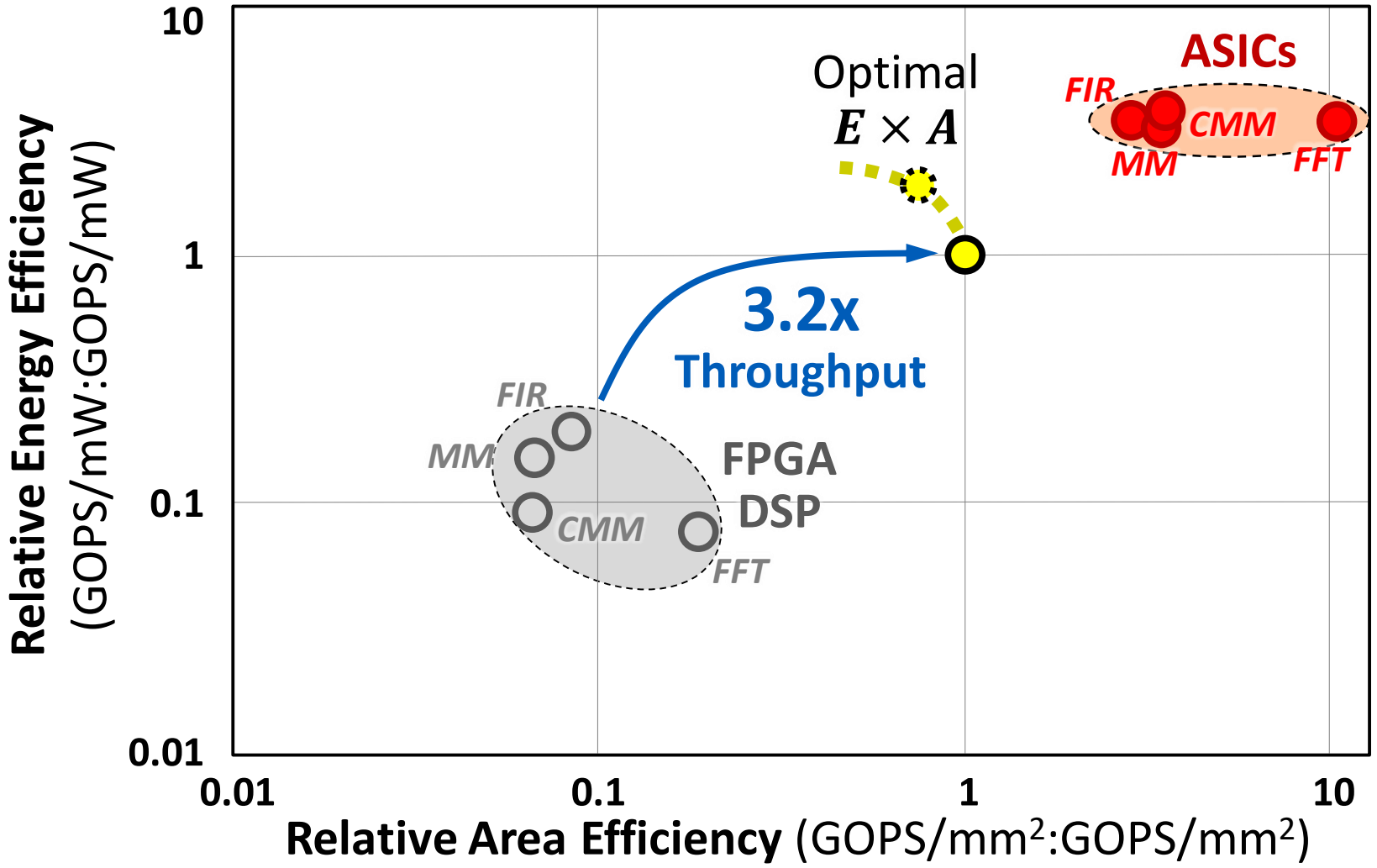
1.1 GHz UDSP

~10× better energy and area efficiency than FPGA

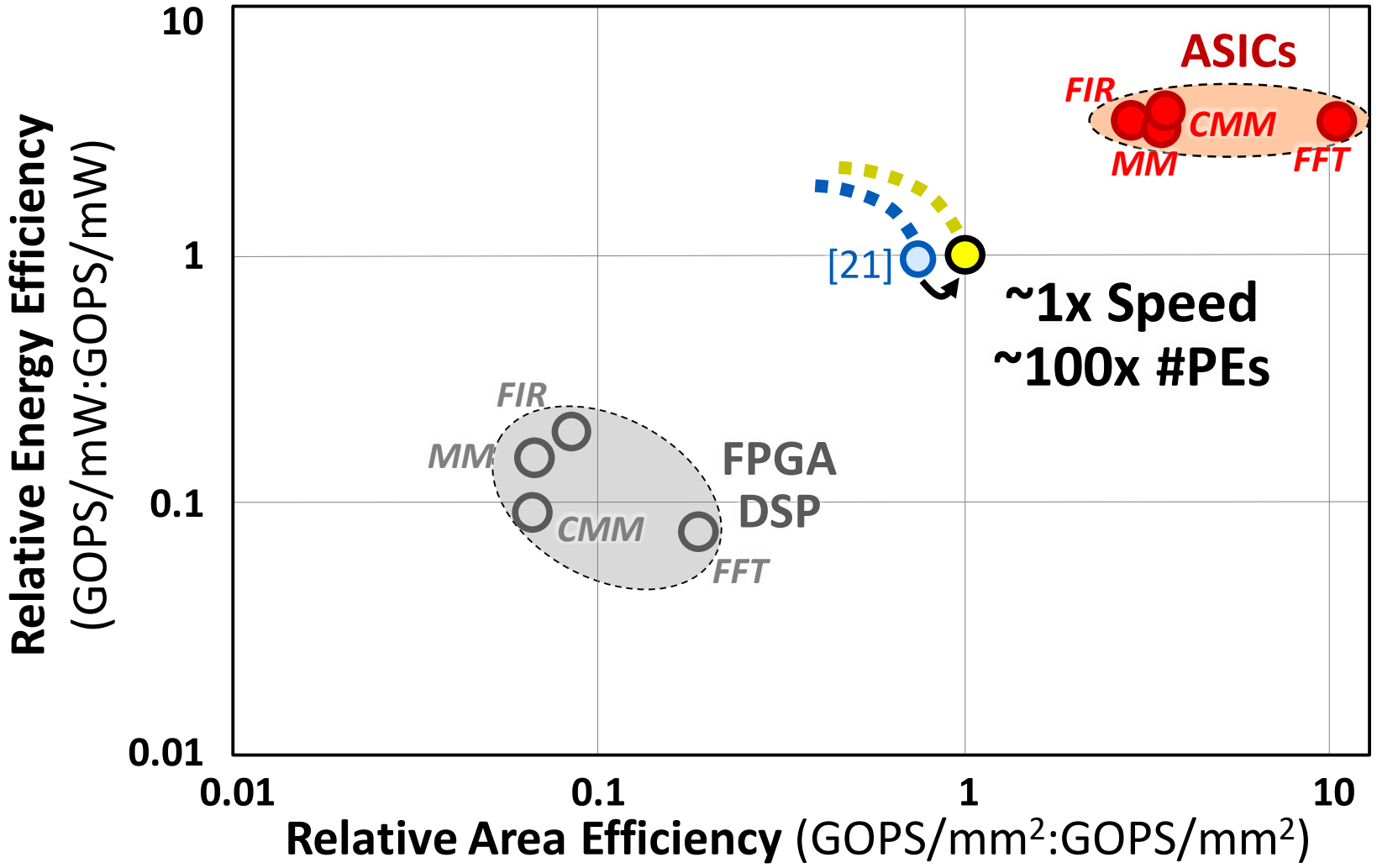
Within **~5×** of ASIC

x/y-axes relative to
~70 GMAC/s/mm²
~310 GMAC/s/W

Optimal E x A, Throughput Comparison



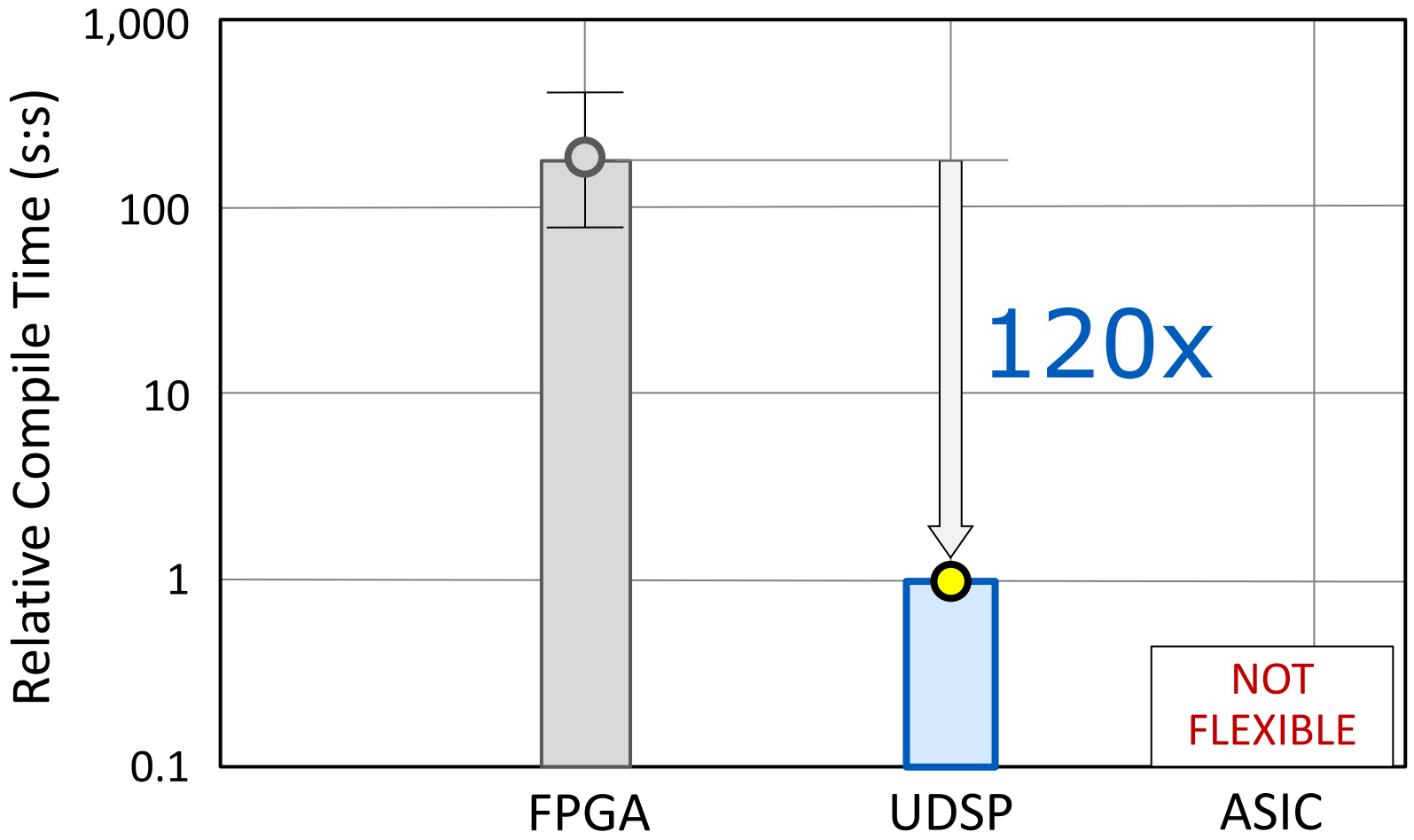
Prior Design Comparison



[16] F. Yuan, et al. JSSC 2015.

data normalized to TSMC 16nm

Average Compile Time Comparison



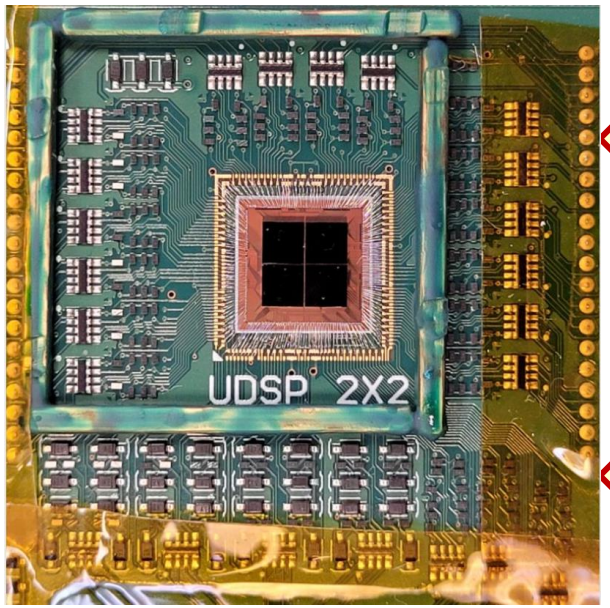
[17]

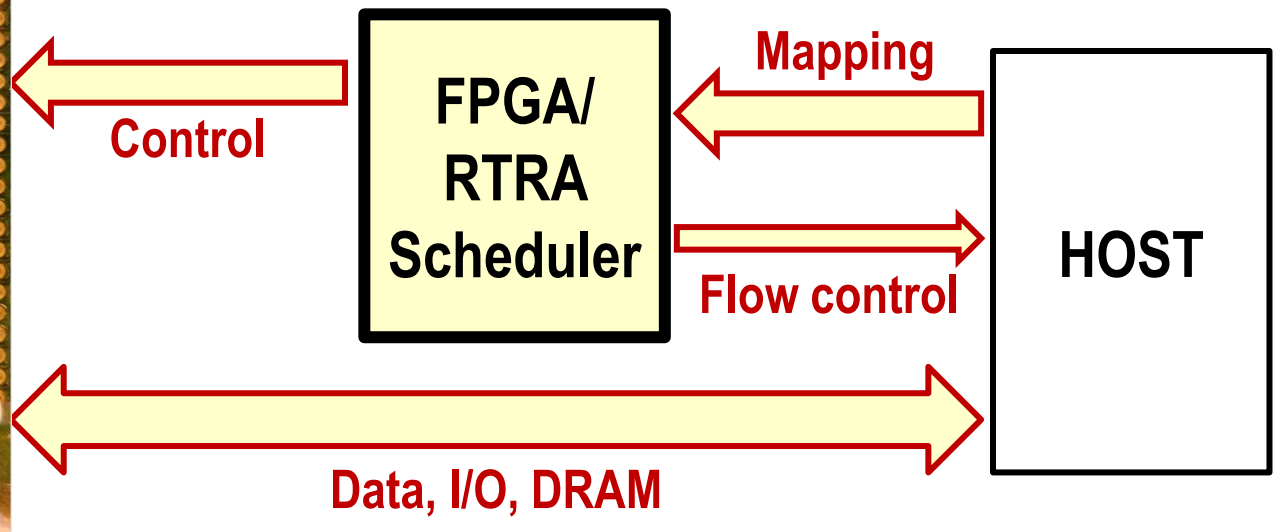
Runtime Reconfigurable Array (RTRA)

- What problem does it solve?
  - Data-driven dynamic relocation of compute resources
  - Fast response to environmental dynamics or uncertainty
- What does it do that others can’t?
  - Fast (sub-μs) reconfiguration of Hw resources
  - Multi-program tenancy with priority and multi-size compile
- How does RTRA do it?
  - Online scheduler for multi-program, multi-size, priority
  - Interconnect tailored to a computation domain
- Benefit vs. FPGA DSP?
  - 10x higher compute efficiency, 10x utilization, 3-4x throughput

Hw-in-the-Loop RTRA Feasibility Study

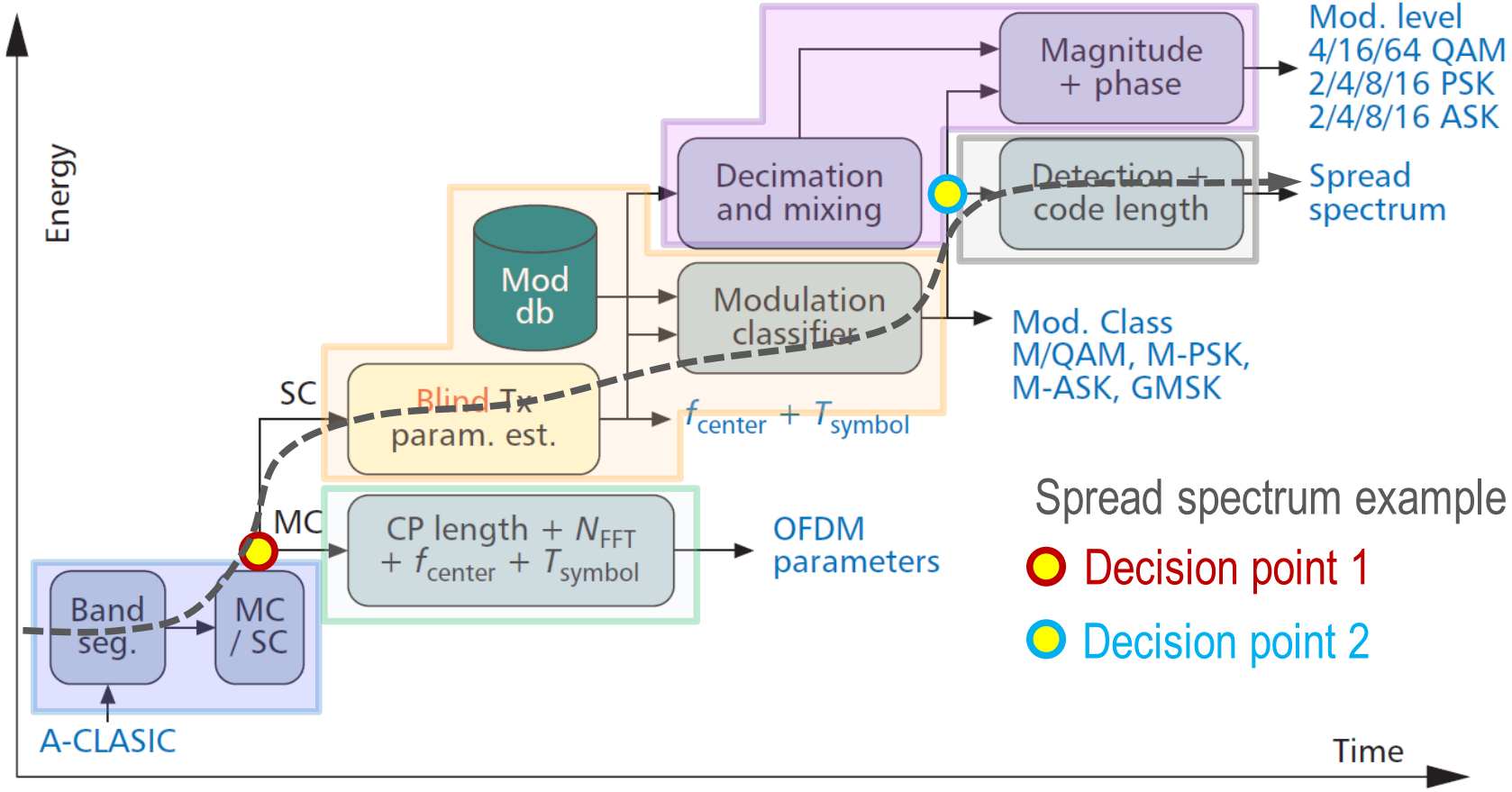
- Characterize utilization, time to repurpose, compute time (config + exe) under multi-program, multi-size, priority mappings
- Compare with statically configured UDSP and FPGA





Compare:
** RTRA (emulated)*
** UDSP*
** FPGA*

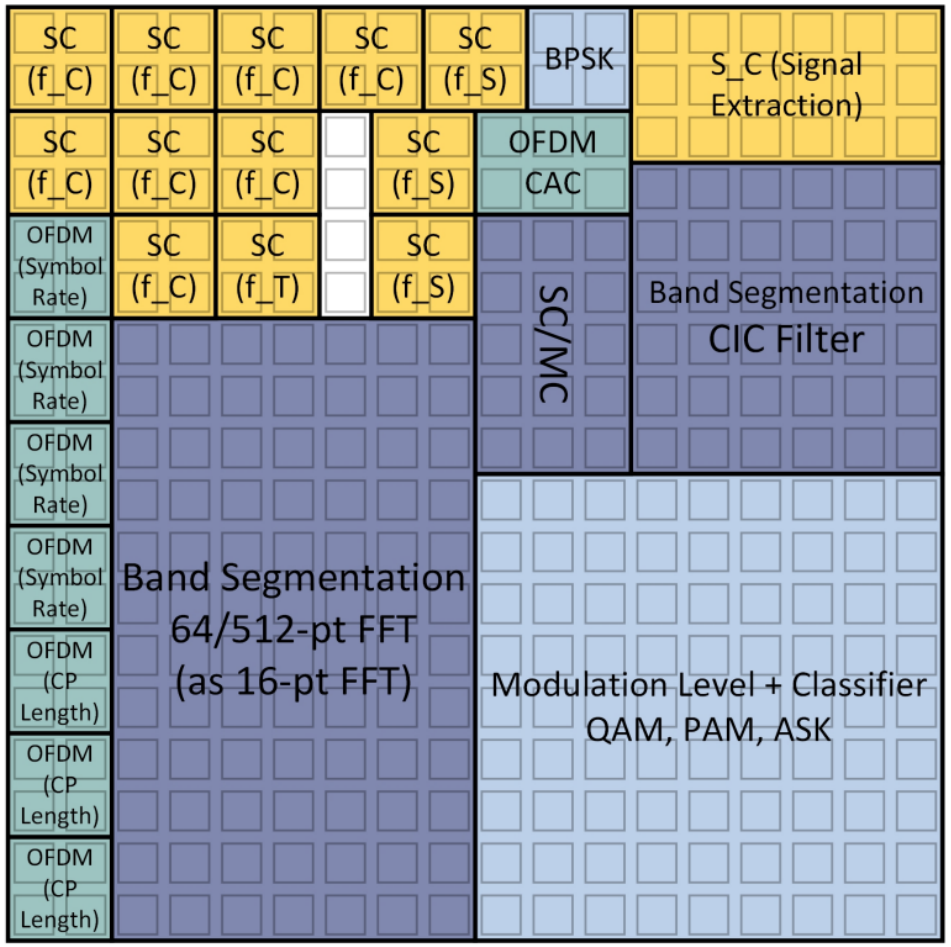
Blind Spectrum Sensing Application



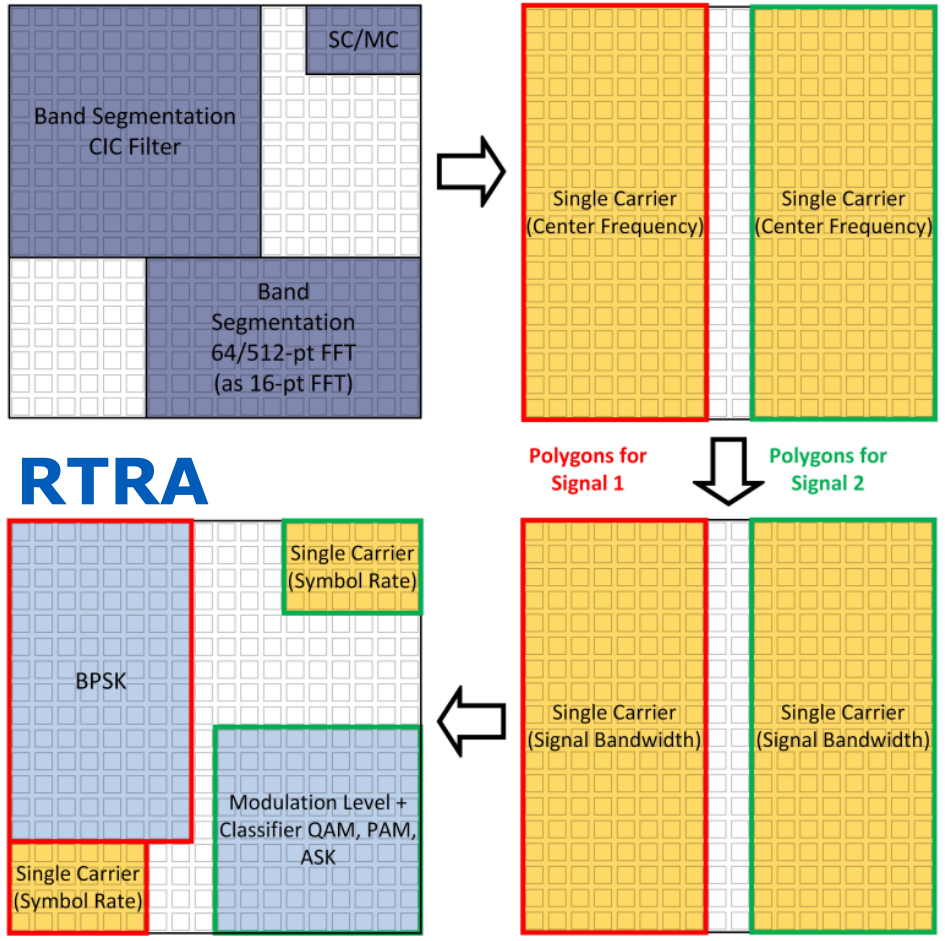
Data-driven attentive processing: detection, extraction, classification

Algorithms from: [18] R. Harjani, et al., IEEE Comm. Magazine, Oct. 2015.

RTRA Dynamically Schedules Hw Array

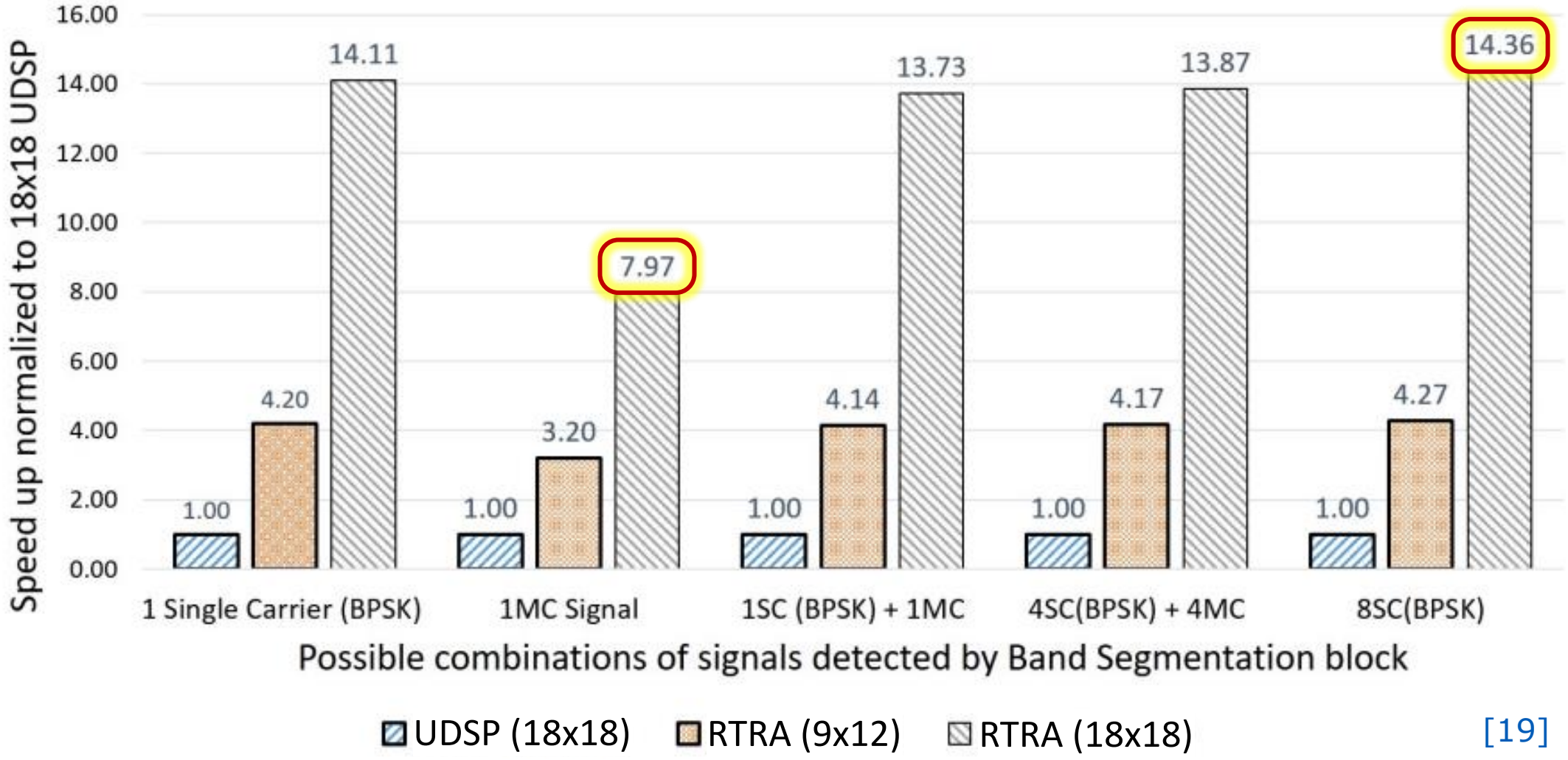


UDSP array size: 18x18



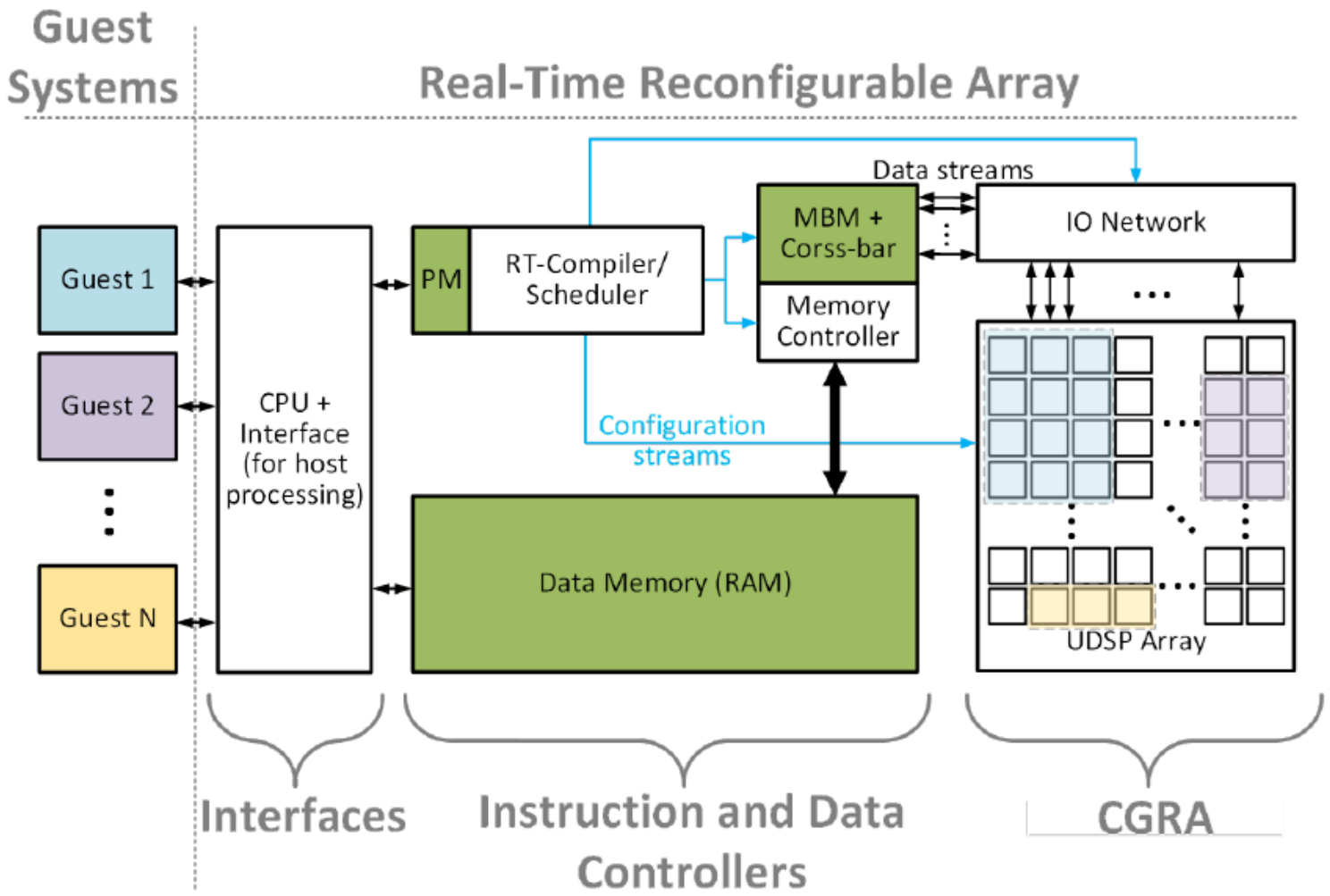
RTRA array size: 18x18

RTRA Maximizes Area Utilization



[19]

RTRA Architecture Building Blocks

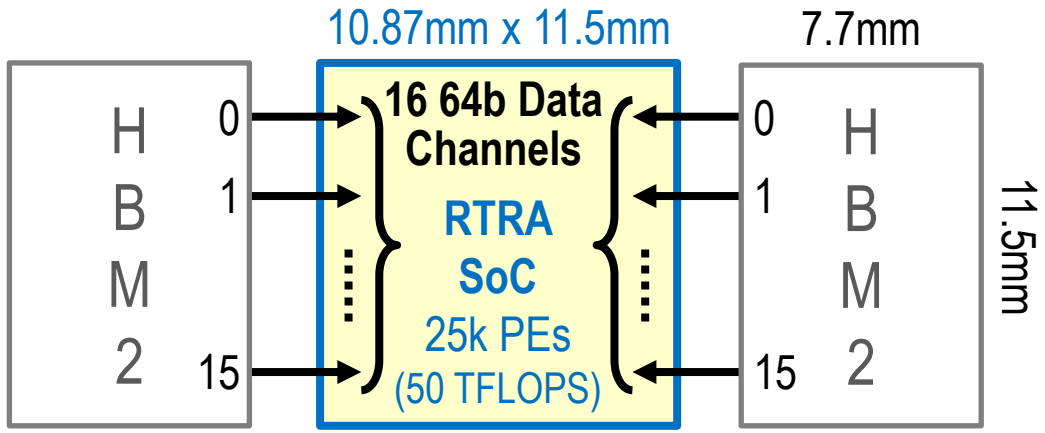


- **RT-Compiler/Sch:** Priority-based dynamic multi-program scheduling, multi-size compile
- **IO network:** equal latency to all Pes
- **Data RAM:** kernels with pre-mapped data
- **Multi-Bank Memory + x-bar:** sustains high data throughput

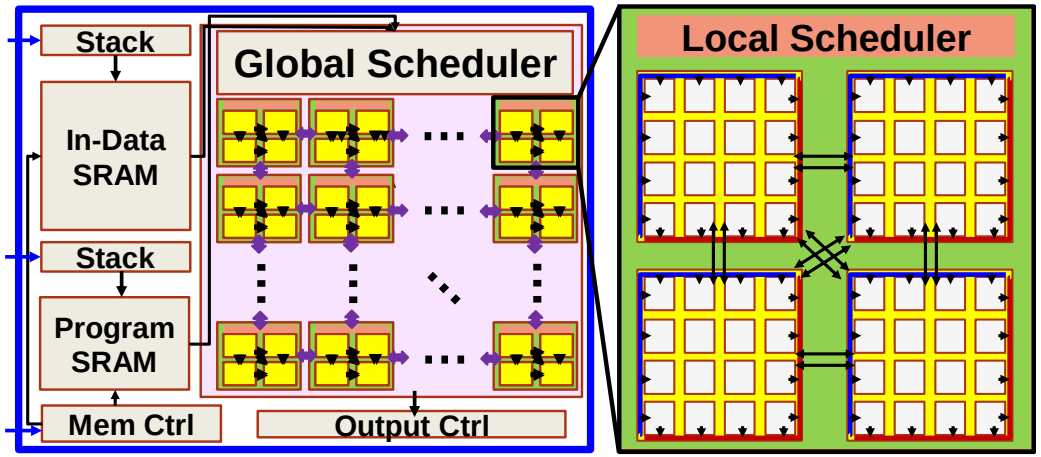
[17]

Scalable RTRA Compute Architecture

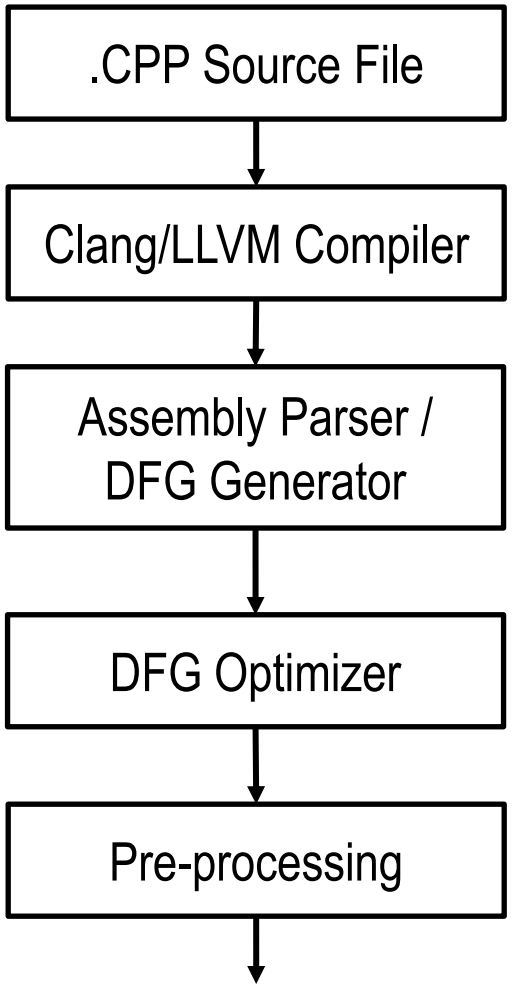
- Key architecture features
  - 16-bit floating-point
  - Modular for scalability
  - Multi-level scheduler
- ~125mm^2 baseline
  - CPU-sized, HBM2 interface
  - Tile-able chiplet assembly
- Proof-of-concept
  - Integrated Sw/Hw DSE
  - FPGA emulation
  - Adapt SoC to I/O & area



Peripheral I/O: **9.9x more native BW** than HBM2
(for chiplet assembly) doable in 6-layer 10-µm pitch



Software: Compiler Front-End



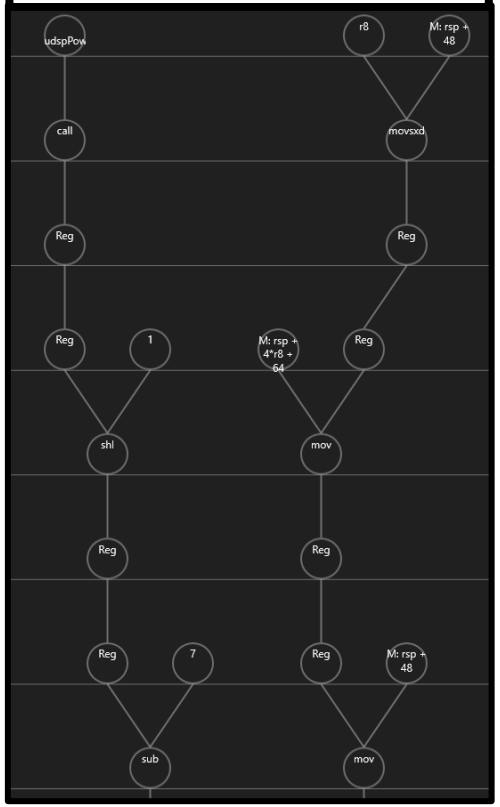
.CPP Source File

```
int main()
{
    int c[10];
    int a = 3;
    int b = 5;
    int i = 0;
    int d = 6;
    while (i <= 10)
    {
        b += 9 + d;
        a +=  4 * udspOperator(b, 3);
        c[i] = 3 * udspOperator( a , 3) - 7;
        i++;
    }
}
```

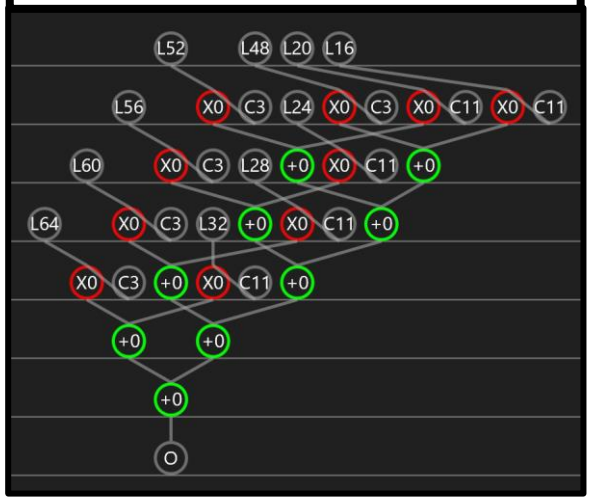
Clang/LLVM Compiler

```
39  .LBB0_1:                                # =>This Inner Loop Header: Depth=1
40      .cv_loc 0 1 20 0                    # LLVM_Test.cpp:20:0
41      cmp dword ptr [rsp + 48], 10
42      jg  .LBB0_3
43  # %bb.2:                                #   in Loop: Header=BB0_1 Depth=1
44  .Ltmp1:
45      .cv_loc 0 1 22 0                    # LLVM_Test.cpp:22:0
46      mov eax, dword ptr [rsp + 44]
47      add eax, 9
48      add eax, dword ptr [rsp + 52]
49      mov dword ptr [rsp + 52], eax
50      .cv_loc 0 1 23 0                    # LLVM_Test.cpp:23:0
51      mov ecx, dword ptr [rsp + 52]
52      mov edx, 3
53      call    "?udspOperator@@YAHHH@Z"
54      shl eax, 2
55      add eax, dword ptr [rsp + 56]
56      mov dword ptr [rsp + 56], eax
57      .cv_loc 0 1 24 0                    # LLVM_Test.cpp:24:0
58      mov ecx, dword ptr [rsp + 56]
59      mov edx, 3
60      call    "?udspOperator@@YAHHH@Z"
61      imul    eax, eax, 3
```

DFG Generator

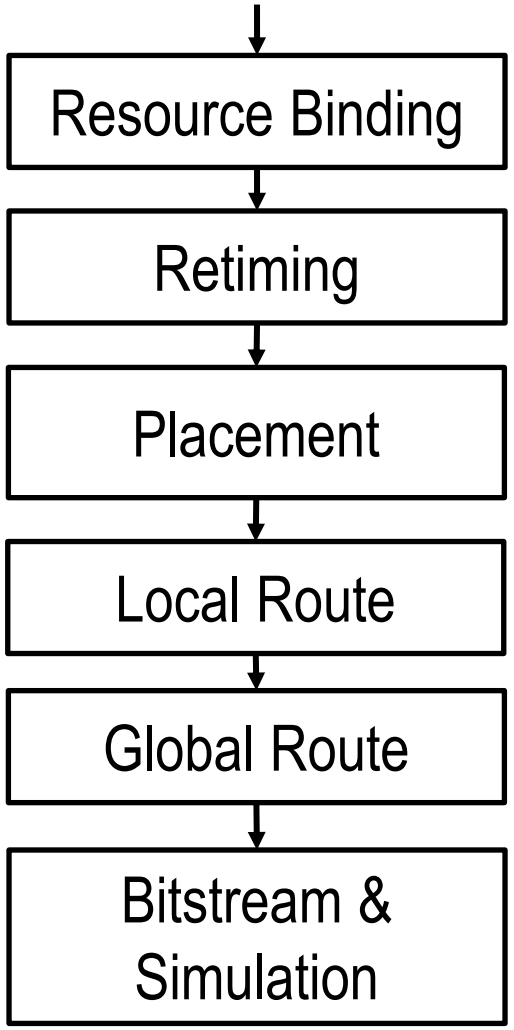


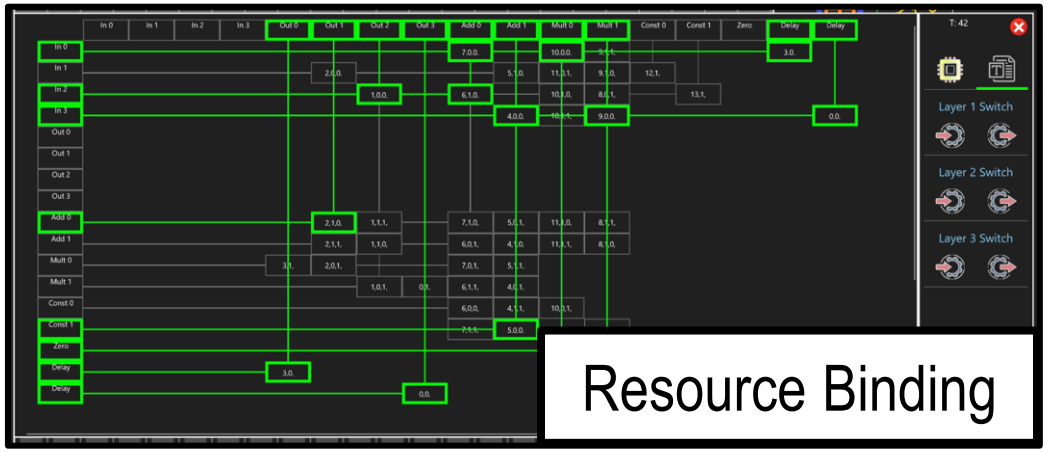
DFG Optimizer + Pre-processing

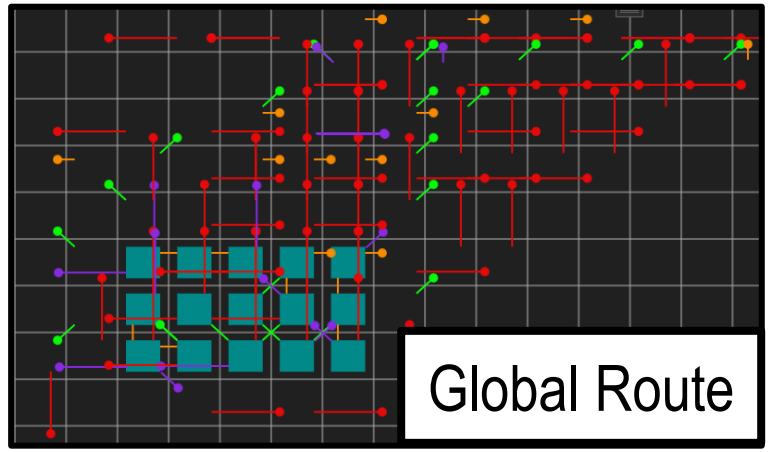


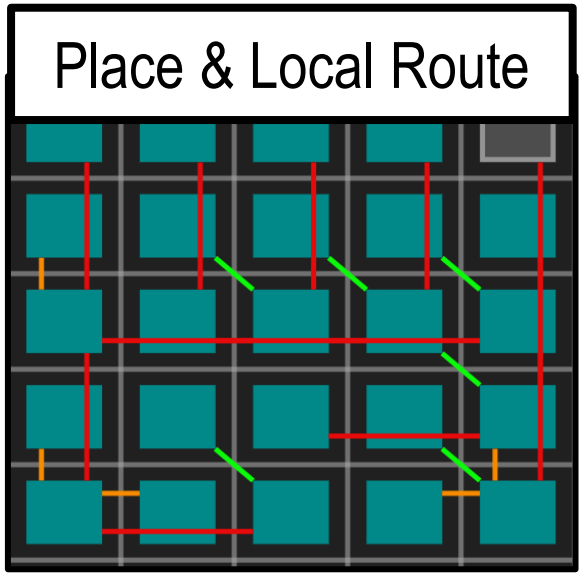
Python source: future work

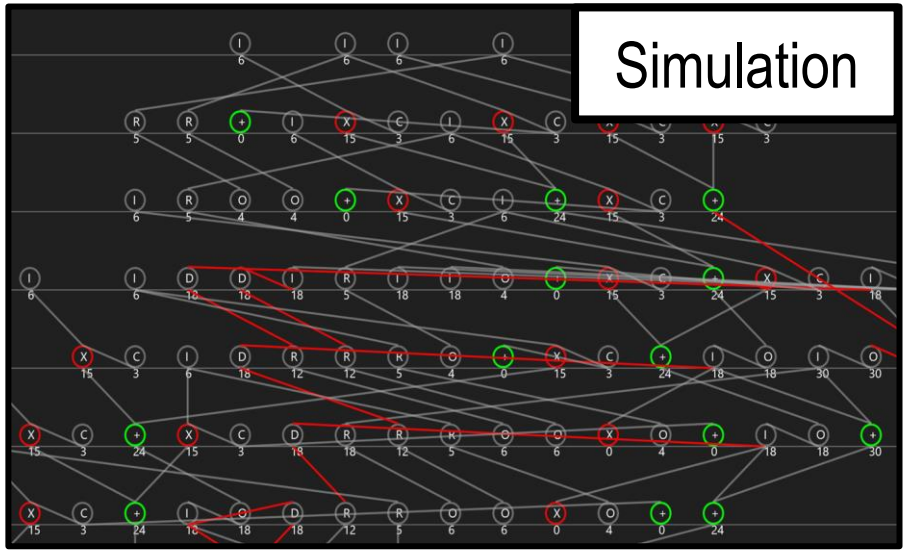
Software: Compiler Back-End





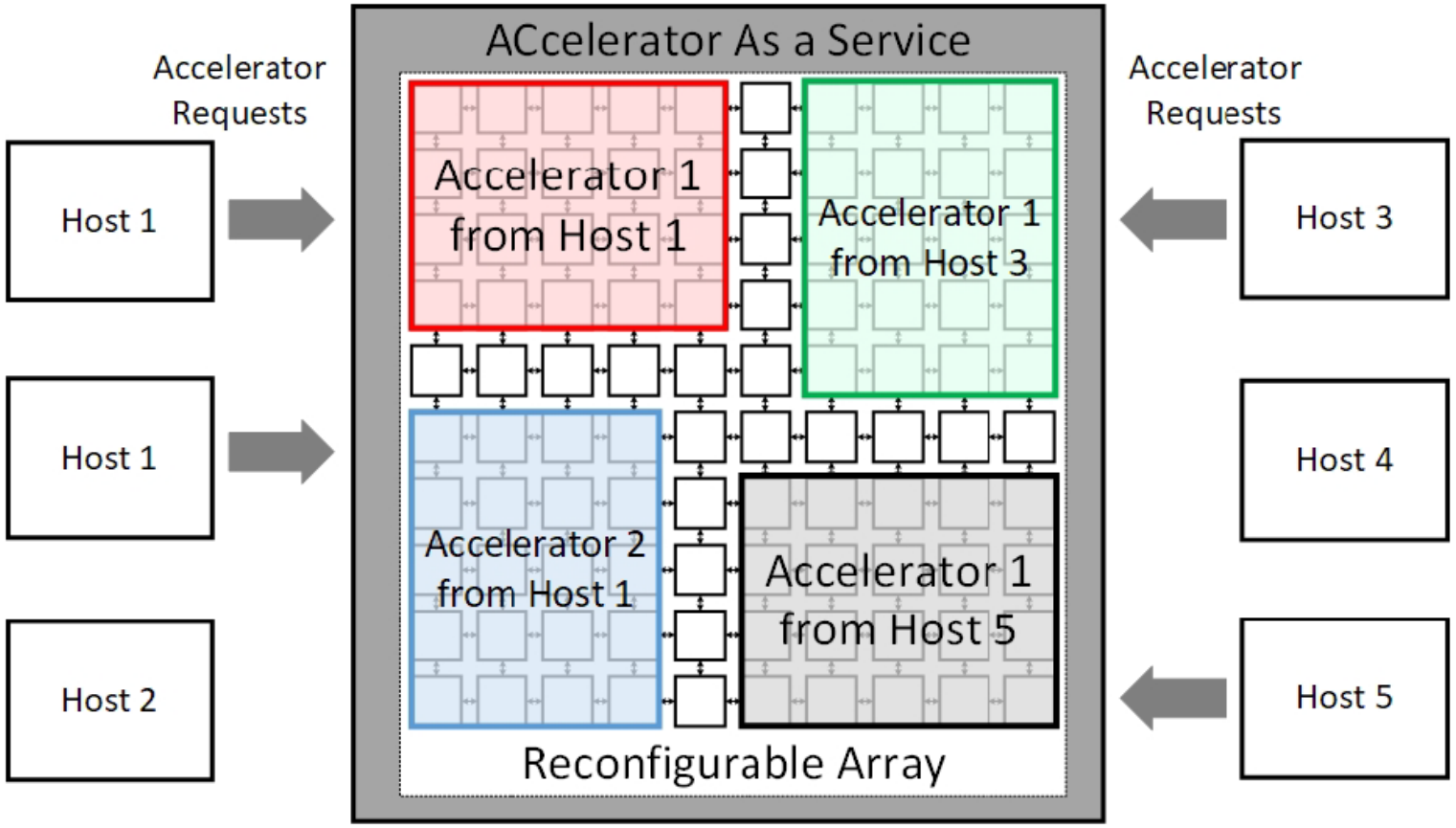






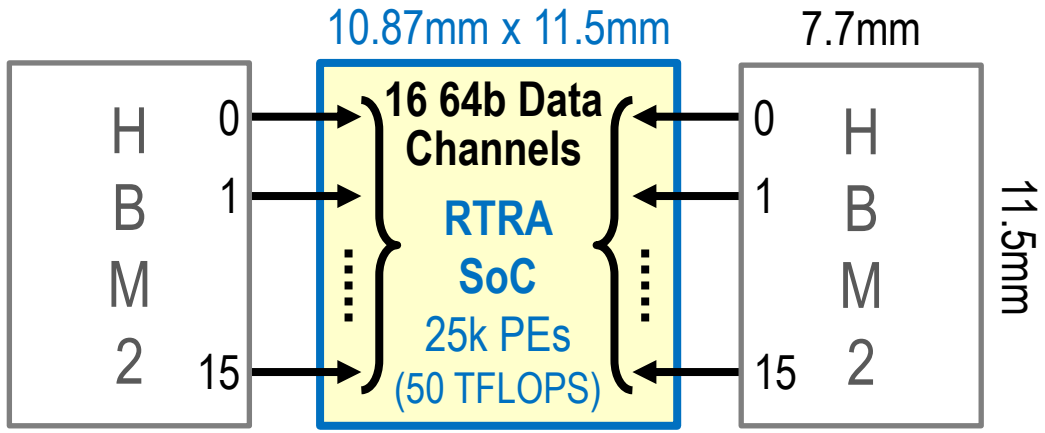
Future: Hardware Accelerator as a Service

- Dynamically compose and relocate accelerators on-the-fly and handle multiple accelerator requests simultaneously

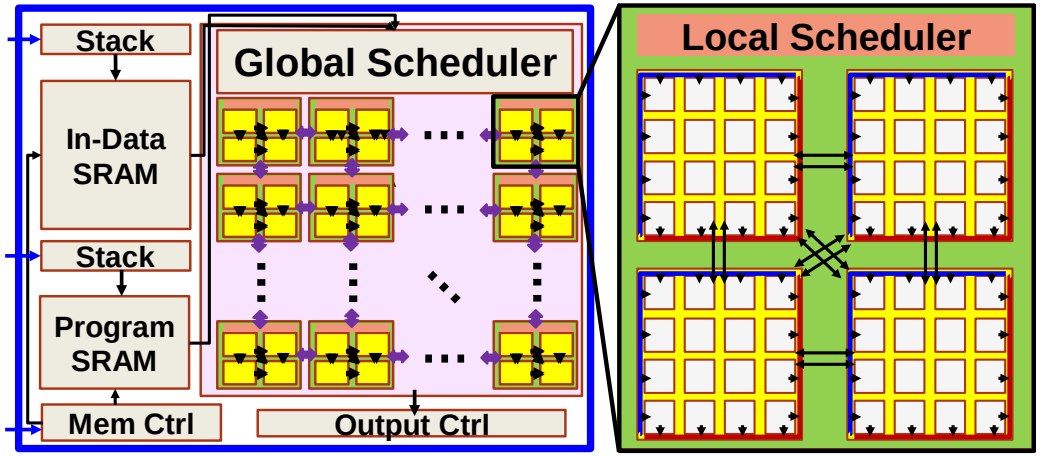


Advancements in Chiplet Assembly

- **HBM 2 interface**
  - 1024 bits
    - 8ch x 128b (16ch x 64b)
  - **89 b/mm**
- **RTRA interface**
  - 10,112 bits
    - 158 PEs per side
    - 64 bits per PE
  - **879 b/mm**
- A 9.9x gap
  - 149b/mm/layer
  - **6-layer 10-µm Si-IF I/O**



Peripheral I/O: **9.9x more native BW** than HBM2
(for chiplet assembly) doable in 6-layer 10-µm pitch



Acknowledgments

- Dr. Uneeb Rathore
- Dr. Sumeet Nagi
- Chenkai (Tim) Ling
- Prof. Subramanian Iyer
- Krutikesh Sahoo
- Dr. Sina Basir-Kazeruni
- Dr. Siva Chandra Jangam
- DARPA CHIPS, DRBE programs
- Accton, Inc.

References (1/2)

[1] Xilinx Zynq RFSoC DFE
[2] 3GPP, Online: https://www.3GPP.org
[3] J. Cong et al., IEEE Design and Test of Computers, 2011.
[4] Y.S. Shao, B. Reagen, G.Y. Wei, D. Brooks, IEEE Micro 2015, 35(3), pp. 58-70.
[5] C.C. Wang, UCLA PhD Thesis, 2013.
[6] F-L. Yuan, UCLA PhD Thesis, 2014.
[7] L. Leibo, ACM Comput. Survey 2020.
[8] Seeds, Bose-Einstein Yield Models, Online: https://www.eesemi.com/test-yield-models.htm
[9] U. Rathore, S. Nagi, S. Iyer, D. Markovic, ISSCC 2022, pp. 52-54.
[10] C.C. Wang, F.-L. Yuan, T.-H. Yu, and D. Marković, ISSCC 2014, pp. 460-461.
[11] F.-L. Yuan, T.-H. Yu, D. Markovic, VLSI 2014, pp. 65-66.
[12] F.L. Yuan, et al., VLSI 2015, pp. 150-151.
[13] S. Jangam, UCLA PhD Thesis, 2020.
[14] M. Lin, T. Huang, JSSC, 2020.
[15] J. M. Wilson, ISSCC, 2018.
[16] F.-L. Yuan, C.C. Wang, T.-H. Yu, D. Markovic, IEEE JSSC, Jan. 2015, 50(1), pp. 137-149.
[17] U. Rathore, UCLA PhD Thesis, 2022.
[18] R. Harjani, et al., IEEE Comm. Magazine, Oct. 2015, 53(10), pp. 173-181.
[19] S. Nagi, UCLA PhD Thesis, 2022.

References (2/2)

[20] AMD on Why Chiplets – And Why Now, Online: https://www.nextplatform.com/2021/06/09/amd-on-why-chiplets-and-why-now

[21] Tesla V100 Price, Online: https://www.microway.com/hpc-tech-tips/nvidia-tesla-v100-price-analysis