*2021 International Symposium on VLSI Design, Automation and Test*

*Invited Talk*

# Holistic and In-Context Design Flow for 2.5D Chiplet-Package Interaction Co-Optimization

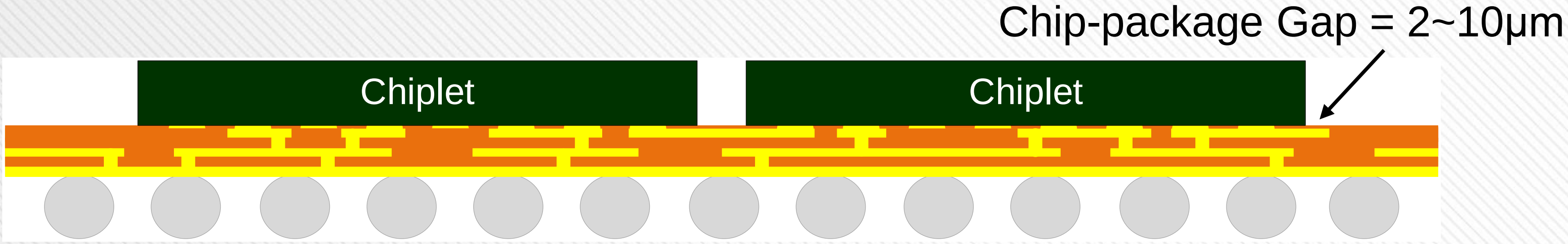
MD Arafat Kabir[1], Weishiun Hung[2], Tsung-Yi Ho[2] , and Yarui Peng[1]

[1]CSCE Department, University of Arkansas

[2]CS Department, National Tsing Hua University

# Overview

- ❑ **Package becomes increasingly critical in post-Moore's Law era**
  - High-density 2D, 2.5D IC, 3D Mem, 3D Sensor, 3D IC, Monolithic 3D IC
  - Heterogeneous integration capabilities (AMD EPYC2, Intel Lakefield)
- ❑ **Interactions between the package and chiplets are growing:**
  - Pin density requires advanced-yet-low-cost integration
  - Package layers are getting closer and more similar to chip BEOL
- ❑ **Cross-boundary Chip-Package Co-design CAD tools are missing:**
  - No existing standard flow that designs 2.5D systems considering chiplets and package interactions during optimization and analysis



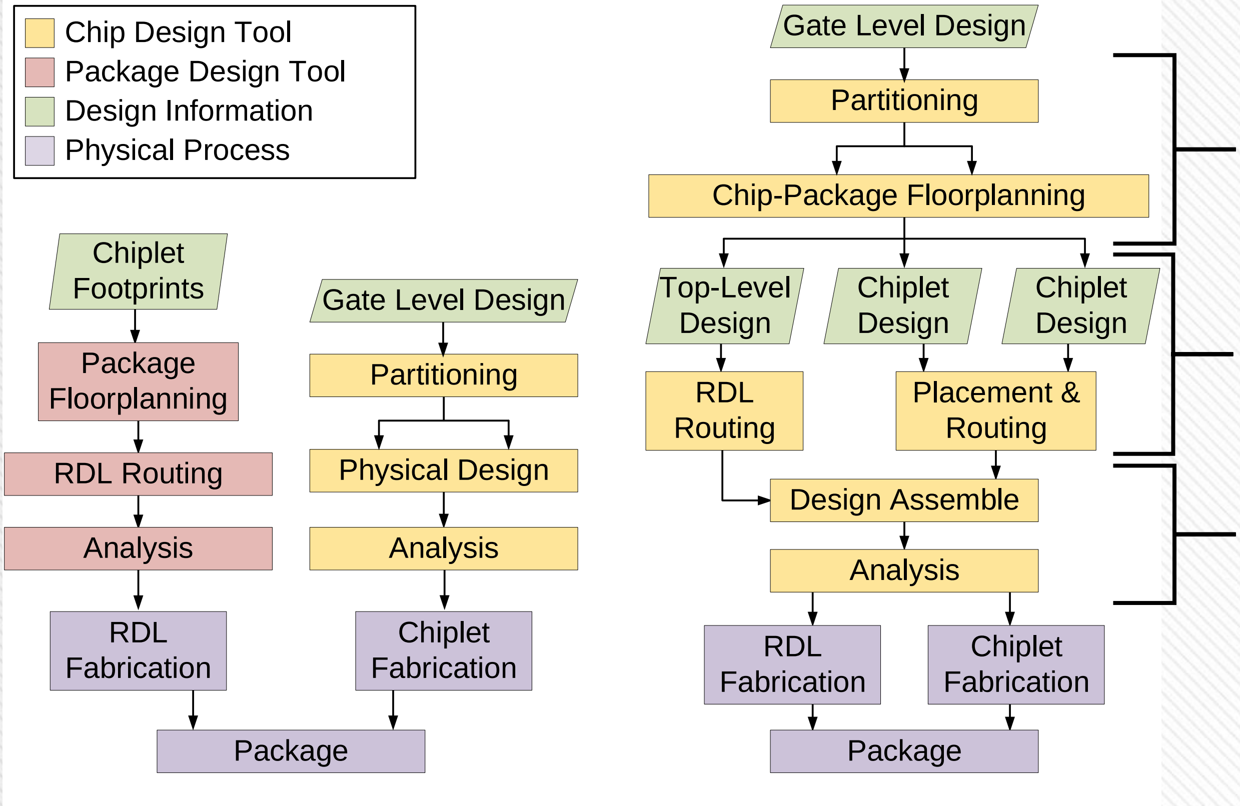
High-density integration scheme with ~5 μm pitch (e.g., InFO)

# Our Target

- ❑ **Traditional die-by-die design flow can achieve the shortest possible 2.5D system design time using off-the-shelf chiplets.**
  - Cannot ensure the maximum performance and highest reliability
  - Pin-dominate nature requires both chip and package characteristics
- ❑ **This research aims to develop the key models and CAD tools to:**
  - Combine chip and package into a single design environment
  - Enable integrating heterogeneous components with advanced multi-die packaging techniques (InFO, CoWoS, EMIB, etc).
  - Provide an open-source design platform for agile 2.5D chiplet designs

# Traditional vs. Our Holistic Flow

❑ **We incorporate the necessary interactions between package and chiplets during design, optimization and analysis steps [1]**



🟨 Chip Design Tool
🟥 Package Design Tool
🟩 Design Information
🟪 Physical Process

Chiplet Footprints → Package Floorplanning → RDL Routing → Analysis → RDL Fabrication → Package

Gate Level Design → Partitioning → Physical Design → Analysis → Chiplet Fabrication → Package

(a) Traditional Flow

Gate Level Design → Partitioning → Chip-Package Floorplanning → Top-Level Design, Chiplet Design, Chiplet Design

Top-Level Design → RDL Routing → Design Assemble

Chiplet Design → Placement & Routing → Design Assemble

Design Assemble → Analysis → RDL Fabrication, Chiplet Fabrication → Package

(b) Proposed Flow

Holistic top-level planning of the entire system

Maintaining parallelism in implementation of individual component

Capturing interactions among all the components of the system in optimization and analysis

[1] Md. Arafat Kabir, and Yarui Peng, **"Chiplet-Package Co-Design For 2.5D Systems Using Standard ASIC CAD Tools"**, in *Proc. Asia and South Pacific Design Automation Conference*, pp. 351-356, Jan 2020.

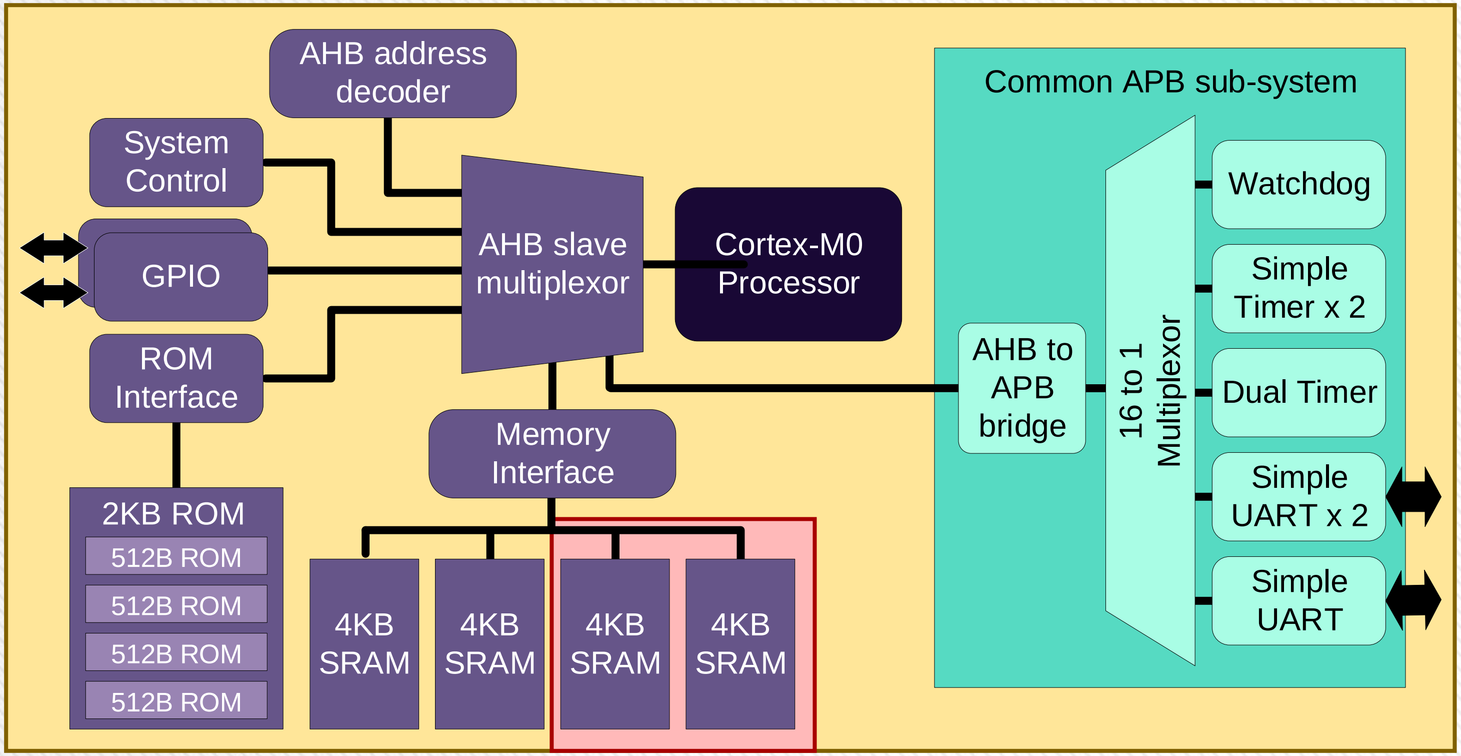
# MCU Architecture and Partitions

## ❑ System architecture of proof-of-concept design

- Microcontroller system based on ARM Cortex-M0 core
- 16KB RAM with some common peripheral devices

🟨 Core Chiplet Components

🟥 Memory Chiplet components

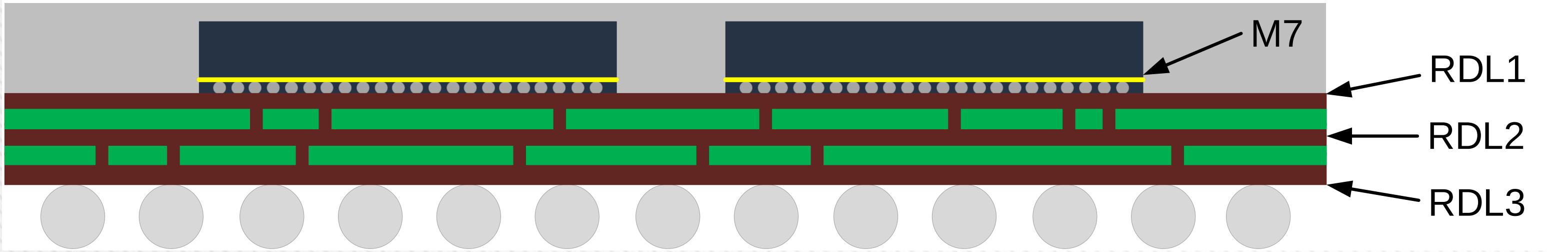


System architecture

# Technology Settings (45nm)

- ❑ **We use Nangate45nm as our baseline PDK**
  - M1-M7 used for chiplet routing
- ❑ **We modify the top three layers to include 2.5D package RDLs**
  - Dimensions are similar to the TSMC 2.5D InFO technology

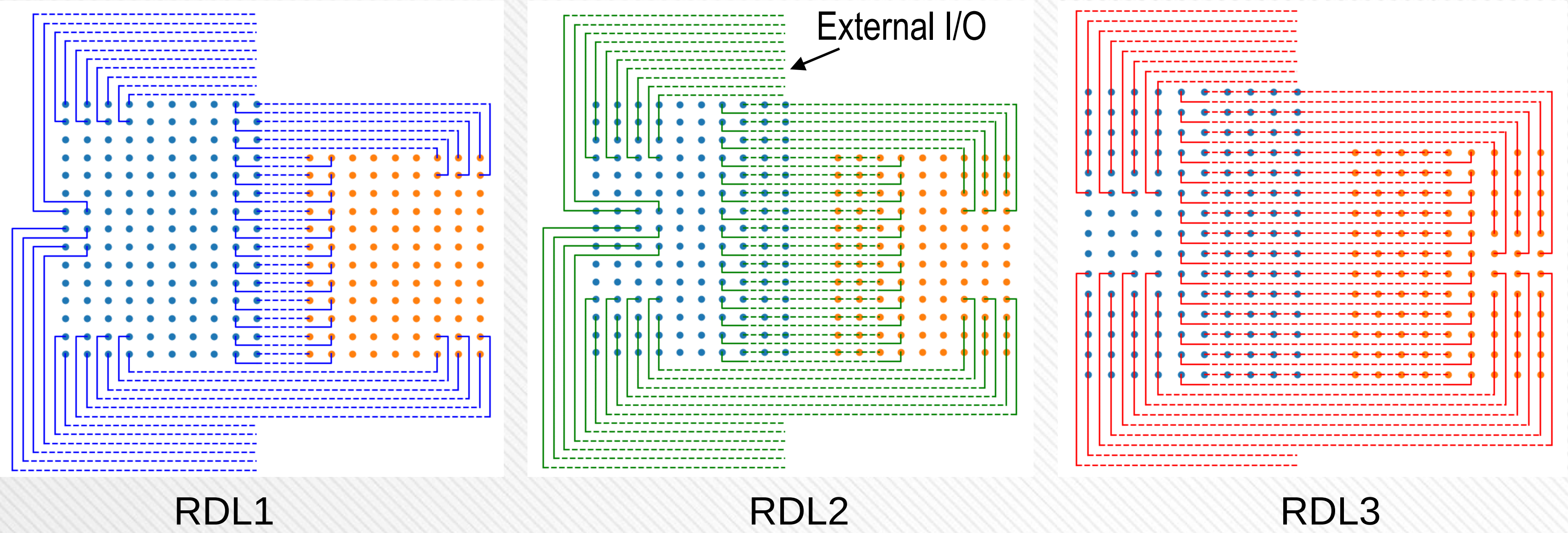
| | M6 | via6 | M7 | via7 | RDL1 | viar1 | RDL2 | viar2 | RDL3 |
|---|---|---|---|---|---|---|---|---|---|
| **Height** | 2.28 | 3.08 | 3.9 | 7.5 | 12.5 | 17.5 | 22.5 | 27.5 | 32.5 |
| **Thickness** | 0.8 | 0.82 | 3.6 | 5 | 5 | 5 | 5 | 5 | 5 |
| **Width** | 0.4 | 0.4 | 2 | 5 | 10 | 10 | 10 | 10 | 10 |
| **Spacing** | 0.4 | 0.44 | 2 | 10 | 10 | 20 | 10 | 20 | 10 |



# RDL Routing Strategy

❑ **To minimize long wires and detours on RDLs, we are using following strategies.**

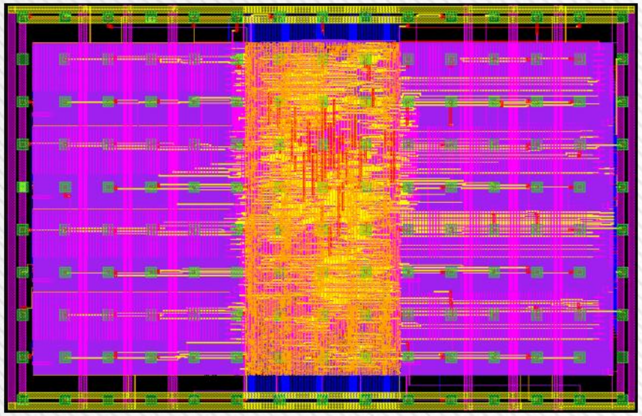
- We don’t assign signals to chiplet pins before routing.
- We route the pins first, and then assign signals based on the routing. This way we have more control and can achieve a very regular routing.
- Use as many straight wires as possible to connect the chiplet pins.



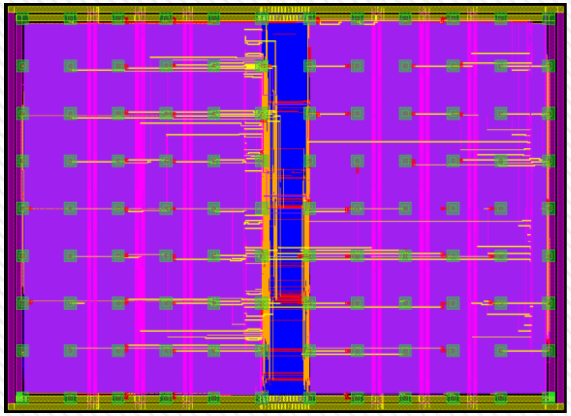
Routing Generated by our program

# Placement & Routing

- **After top level planning, chiplets and package are implemented independently with constraints propagated from top-level**
  - Top level design is hierarchically split like 2D partitioning.
  - Chiplet floorplan may change as required, only the pin arrangement needs to be the same as fixed by top level planning.
  - Chiplet implementation is the similar as the conventional 2D chip that includes power planning, placement, time design, routing and post routing optimizations.



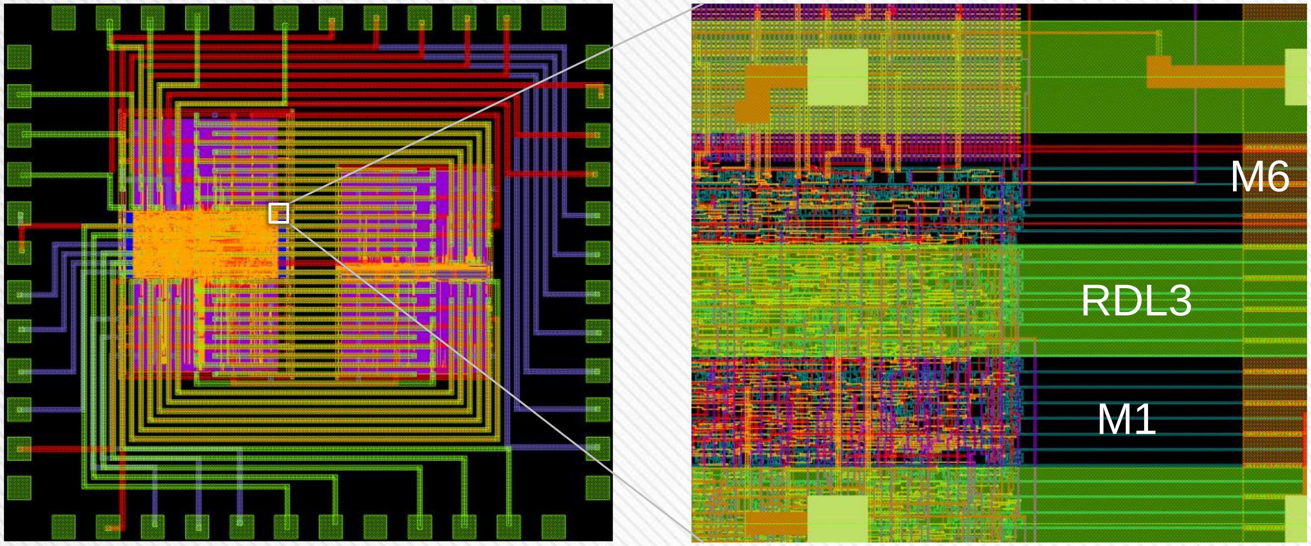
(a) Core System Chiplet



(b) Extended Memory Chiplet

# Assembly & Holistic Extraction

- **Finished package and chiplet designs are assembled for holistic extraction.**
  - As the design environment has everything together, incremental optimizations can be performed to improve overall system performance.
  - The analysis and optimization tools have all the information needed to account for the impacts of RDLs on chiplet design.



Assembled System

Zoom-in View

# Holistic Extraction Result

## ❑ Chiplet-Package coupling capacitance

- The columns for RDL1, RDL2, and RDL3 show the coupling capacitance between package layers and chiplet layers (in fF).
- M7 and RDL1 are extracted with considerations from the other side
  - Package-to-M7 is low because of a smaller number of wires on M7.
  - However, package-to-M6 coupling is captured in the parasitic extraction

**Coupling Capacitance**

| | M1-M5 | M6 | M7 | RDL1 | RDL2 | RDL3 |
|---|---|---|---|---|---|---|
| **M1-M5** | 6120 | 442.2 | 28.65 | 52.95 | 8.102 | 5.862 |
| **M6** | 442.2 | 596.6 | 78.03 | 122.8 | 12.98 | 10.53 |
| **M7** | 28.65 | 78.03 | 30.63 | 15.02 | 1.509 | 2.256 |
| **RDL1** | 52.95 | 122.8 | 15.02 | 299.3 | 1016 | 39.06 |
| **RDL2** | 8.102 | 12.98 | 1.509 | 1015 | 298.3 | 1085 |
| **RDL3** | 5.862 | 10.53 | 2.256 | 39.06 | 1084 | 578.4 |

**Ground Capacitance**

| Metal Layer | M1-M5 | M6 | M7 | RDL1 | RDL2 | RDL3 |
|---|---|---|---|---|---|---|
| **Capacitance** | 21119 | 2054 | 272 | 1040 | 247 | 636 |

# Iterative Optimization Results

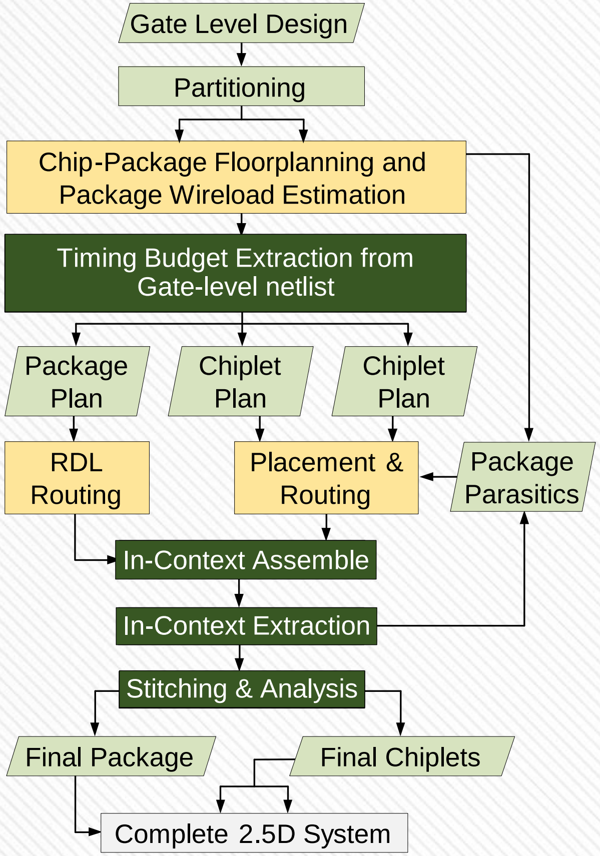
- ❑ **Chiplet-package interaction is used to improve the system performance through iterative optimizations**
  - Chiplet design tool automatically optimizes the inter-chiplet IO buffers to compensate for package overhead by 62.5%

| Design Case | Chiplet Design | Logic Gates# | Buf/ Inv# | Die Size ($um^2$) | M6 WL (mm) | M7 WL (mm) | Power (mW) | Freq. (MHz) | Freq. Overhead |
|---|---|---|---|---|---|---|---|---|---|
| **2D** | | 17595 | 3700 | 550x550 | 79.94 | 0 | 10.6 | 333 | **0%** |
| **2.5D base** | Core | 17783 | 2740 | 390x590 | 30.81 | 1.783 | 7.751 | 245 | **100%** |
| | Mem | 132 | 132 | 350x470 | 5.986 | 0.598 | 0.194 | | |
| **2.5D initial** | Core | 17915 | 2865 | 390x590 | 31.86 | 1.875 | 9.043 | 280 | **60.23%** |
| | Mem | 148 | 148 | 350x470 | 8.201 | 0.589 | 0.216 | | |
| **2.5D final** | Core | 18214 | 2955 | 390x590 | 31.42 | 2.02 | 9.840 | 300 | **37.50%** |
| | Mem | 45 | 45 | 350x470 | 8.445 | 0.624 | 0.162 | | |

# In-Context for Heterogeneous

- ❑ **Existing EDA tools cannot handle multiple heterogeneous technologies together in a common design scope**

  - Holistic timing budget and parasitic extraction not possible for heterogeneous technologies.
  - We break down the package into sub-regions around chiplets (package contexts) and create an extended partition for each chiplet.
  - We perform in-context extraction and then stitch all SPEFs in the analysis tool for analysis and timing context creation.

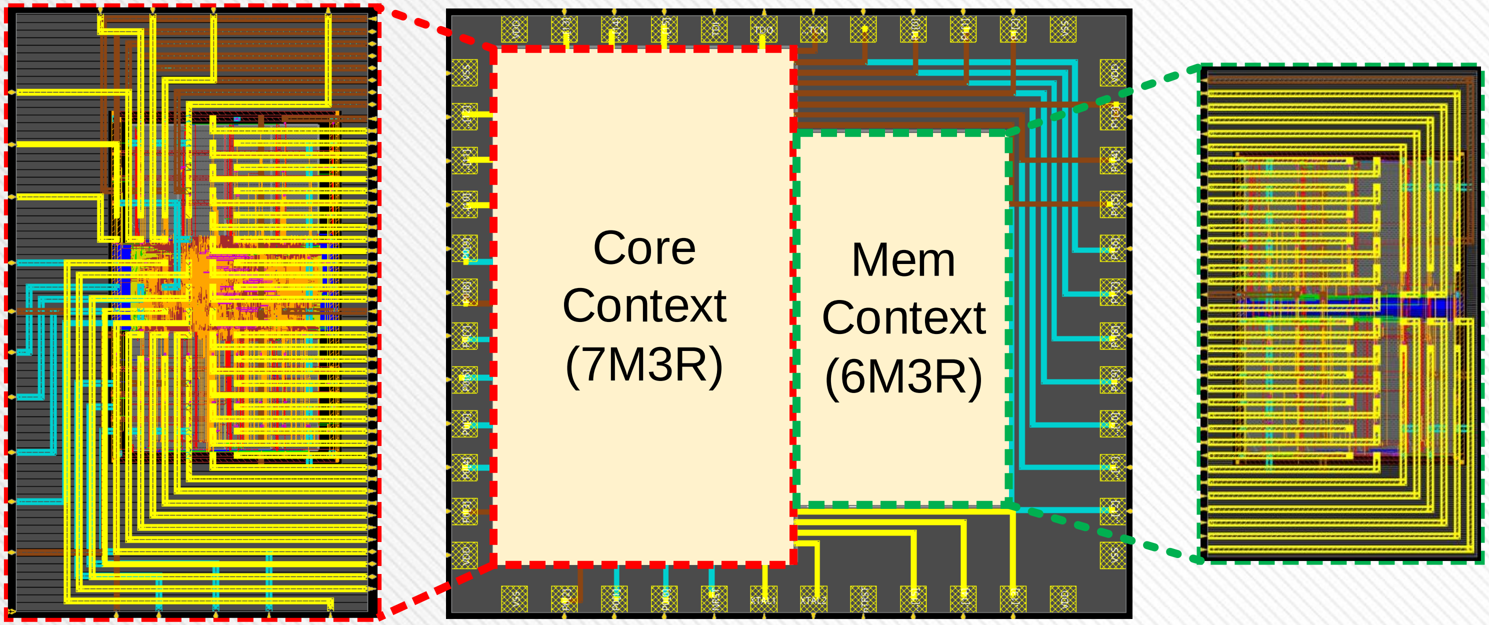


Our In-Context Flow [2]

[2] Md. Arafat Kabir, Dusan Petranovic, and Yarui Peng, **"Extraction and Optimization for Heterogeneous 2.5D Chiplet-Package Co-Design"**, in *Proc. International Conference on Computer-Aided Design*, 2020 Nov.

# A Heterogeneous Design

- □ **We prepared a 45nm proof-of-concept design using different metal stacks and cell libraries for different chiplets**
  - ● Package is routed using three RDLs (3R)
  - ● Core-chiplet uses seven chip-routing layers (7M) and Nangate library
  - ● Memory-chiplet uses six chip-routing layers (6M) and GSCL library



# In-Context Extraction Comparison

- ❑ **We performed in-context extraction on the homogeneous design for comparative study**
  - The total GCAP error is only 0.71% and total CCAP error is only 0.79%
  - InC package GCAP is overestimated due to fringe cap. on cutting edges
  - Die-by-die extraction (DbD) overestimates GCAP and underestimates CCAP on all layers which may cause signal integrity issues

| Metal Layer | M1-M5 | M6 | M7 | R1 | R2 | R3 |
|---|---|---|---|---|---|---|
| **Holi GCAP** | 21119 | 2054 | 272 | 1040 | 247 | 636 |
| **DbD GCAP** | 21139 | 2090 | 278 | 1539 | 362 | 658 |
| **InC GCAP** | 21119 | 2053 | 273 | 1103 | 306 | 696 |
| **DbD GCAP Err** | 0.10% | 1.78% | 2.09% | 47.97% | 46.77% | 3.45% |
| **InC GCAP Err** | 0.00% | -0.01% | 0.09% | 6.03% | 24.0% | 9.46% |
| **Holi CCAP** | 9172 | 1263 | 156 | 1544 | 2421 | 1721 |
| **DbD CCAP** | 9125 | 1213 | 141 | 1378 | 2287 | 1699 |
| **InC CCAP** | 9171 | 1265 | 153 | 1563 | 2489 | 1765 |
| **DbD CCAP Err** | -0.52% | -3.95% | -9.94% | -10.75% | -5.55% | -1.30% |
| **InC CCAP Err** | -0.01% | 0.17% | -2.10% | 1.20% | 2.81% | 2.56% |

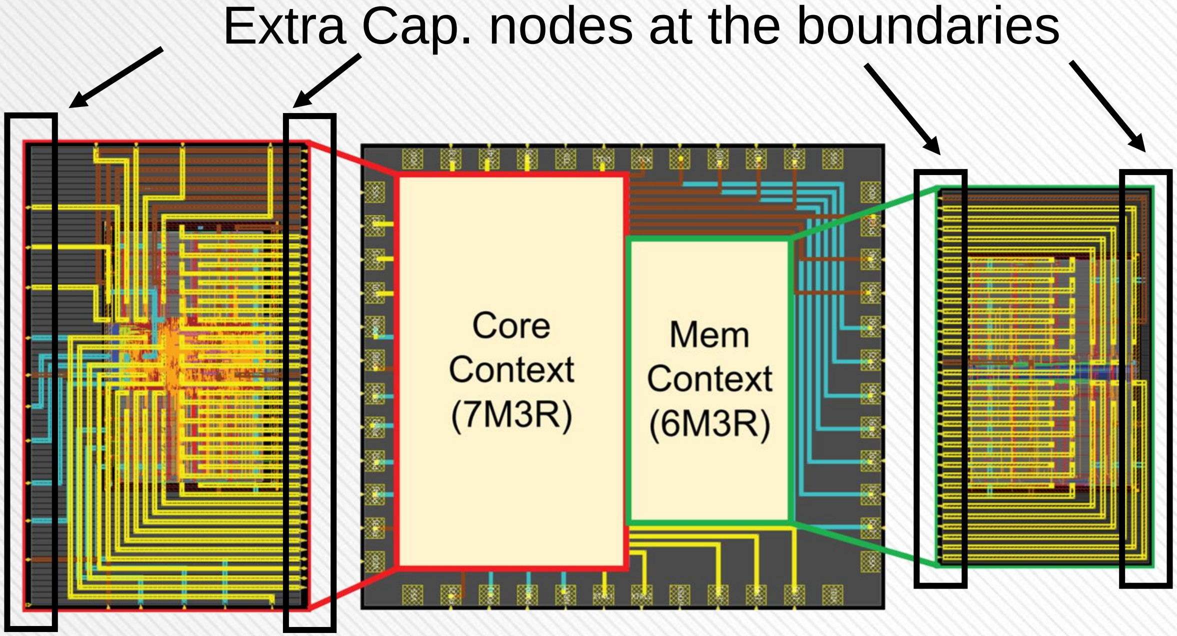
# Fringe Cap Overestimation

❑ **Our first in-context implementation has some inaccuracy**

- **Overestimated ground capacitance on RDL wires**

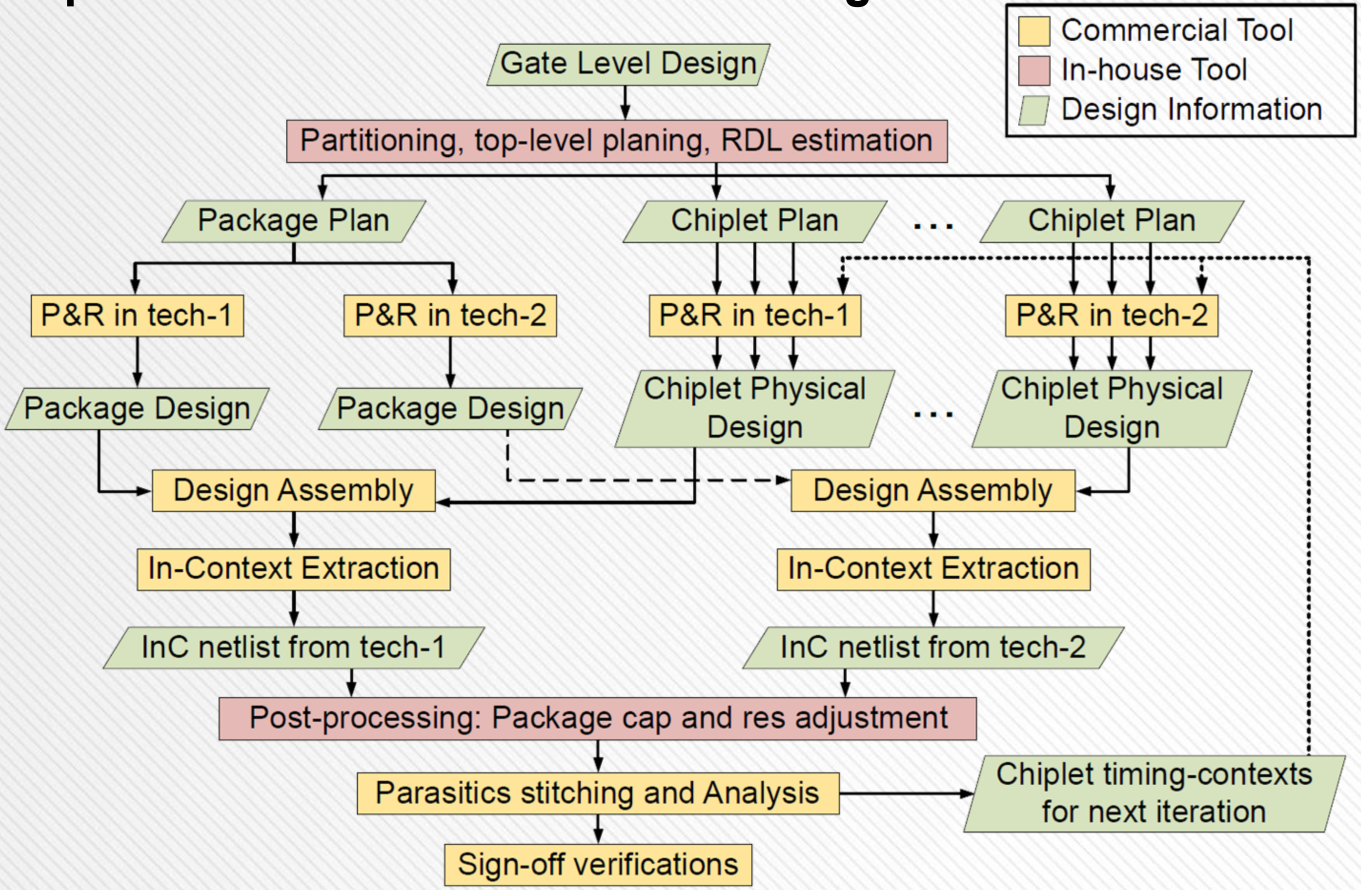
| Metal Layer | M1-M5 | M6 | M7 | R1 | R2 | R3 |
|---|---|---|---|---|---|---|
| **InC GCAP Err** | 0.00% | -0.01% | 0.09% | 6.03% | 24.0% | 9.46% |
| **InC CCAP Err** | -0.01% | 0.17% | -2.10% | 1.20% | 2.81% | 2.56% |

- **This is due to the fringe capacitance at the hierarchical boundary**



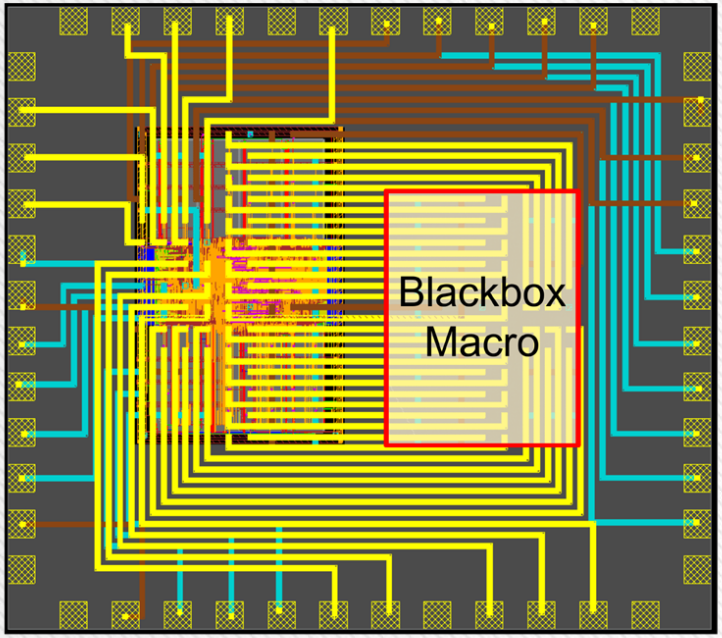
# New In-context Flow

- **Avoid cutting in the new flow**
  - **Assemble chiplet separately, each with a full package design**
  - **Post-process to correct double-counting with incremental SPEF**

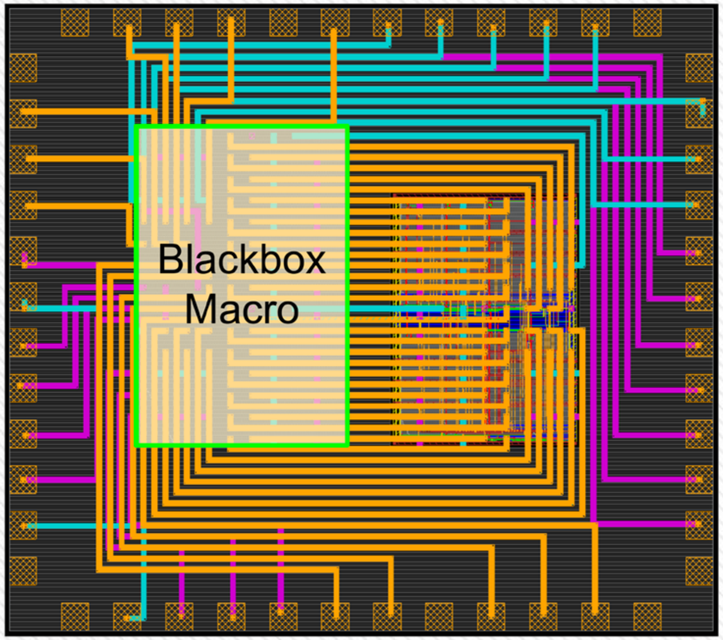


# Post-Processing Methodology

- **In-context parasitics are adjusted based on top-level package**
  - **The entire package is included in all netlists (double-counted)**
  - **The overestimation on package nets are exactly equal to the top-level package (all black-box chiplets) parasitics.**
    - Can be used to remove double counting



(a) Assembled Core-Context (7M3R)



(b) Assembled Mem-Context (6M3R)

# Post-Processing Methodology

❑ **Layer-wise capacitances in an in-context netlist are reduced by a fraction of the layer-wise capacitances of the top-level netlist.**

- User specifies, what percent (userFact) of the top-level parasitics need to be reduced from the package nets.
- All cap. nodes (gnd and coup.) of a package net is multiplied using the corresponding factor (layerFact$_x$).
- The resistance values of the double-counted nets are doubled. But the equivalent resistance due to parallel connection remains correct

$$layerFact_x = \frac{CapRDL_x - userFact \times TCapRDL_x}{CapRDL_x} \quad (1)$$

$$newNodeCap = nodeCap \times layerFact_x \quad (2)$$

# New In-Context Results

❑ Per-layer error is less than 1% (expected)

| Metal Layer | M1-M5 | M6 | M7 | R1 | R2 | R3 |
|---|---|---|---|---|---|---|
| In-C GCAP Err [2] | 0.00% | -0.01% | 0.09% | 6.03% | 24.0% | 9.46% |
| In-C GCAP Err this work | 0.00% | 0.00% | 0.01% | 0.24% | 0.6% | 0.00% |
| In-C CCAP Err [2] | 0.01% | 0.17% | -2.10% | 1.20% | 2.81% | 2.56% |
| In-C CCAP Err this work | 0.00% | 0.04% | 0.64% | 0.03% | -0.01% | 0.00% |

❑ Per-net error is also less than 1% (validates the flow)

| Parameter | Max. Error | Min. Error | Avg. Error |
|---|---|---|---|
| Path delay | 3.30% | 0.00% | 0.61% |
| Design constraint | 1.80% | 0.30% | 0.62% |
| Load Capacitance | 1.70% | 0.00% | 0.29% |

[2] Md. Arafat Kabir, Dusan Petranovic, and Yarui Peng, “Extraction and Optimization for Heterogeneous 2.5D Chiplet-Package Co-Design”, in *Proc. International Conference on Computer-Aided Design*, 2020 Nov.

# In-Context Iterative Optimization

❑ **Using in-context flow, we performed design and iterative optimizations of the chiplets**

- The heterogeneous in-context design achieved the similar optimization results as homogeneous holistic design

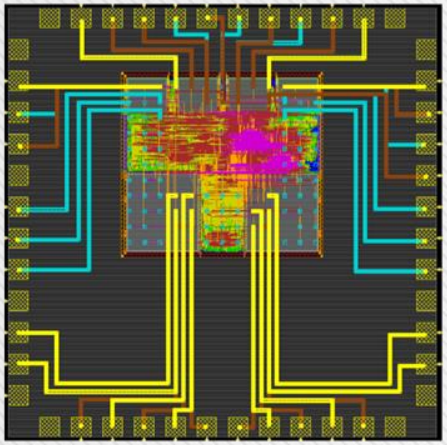
| Design Iteration | Homogen Holi | Homogen In-C | Heterogen In-C |
|---|---|---|---|
| **With RDL wireload** | 288 | 288 | 287 |
| **In-Context 1st iteration** | 293 | 294 | 294 |
| **In-Context 2nd/final** | **300** | **300** | **300** |

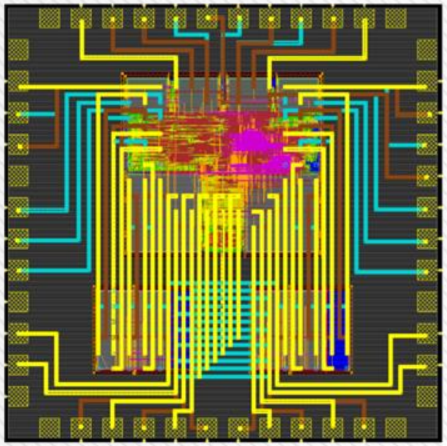
| Power Group | Homogen Holi | Homogen In-C | Heterogen In-C |
|---|---|---|---|
| **Wire** | 4.34 | 4.30 | 4.24 |
| **Cell** | 6.35 | 6.37 | 6.22 |
| **Total** | **10.69** | **10.67** | **10.46** |

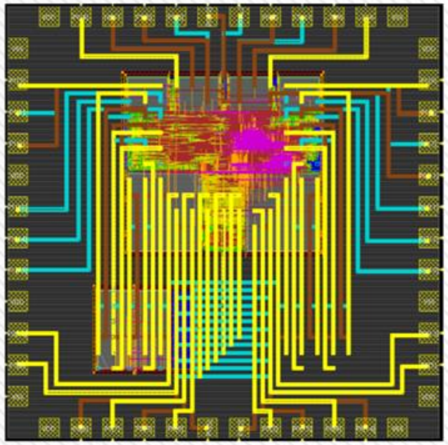
UNIVERSITY OF ARKANSAS

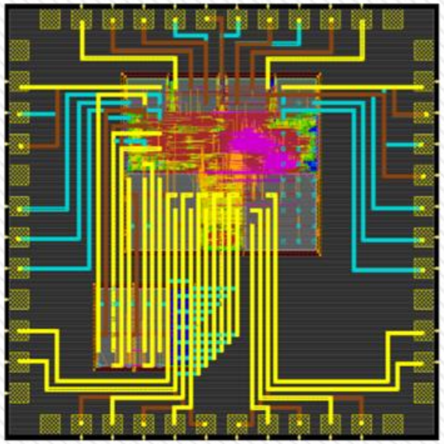
# Agile Custom 2.5D Designs

- ❑ **Our flow also offers design flexibility and agile customization**
  - System (a): Core-only system without any memory chiplet.
  - System (b-c): 2.5D systems with various chiplet/package configurations


(a) System with 8KB Memory


(b) System with 16KB Memory


(c) Drop-In 12KB Design

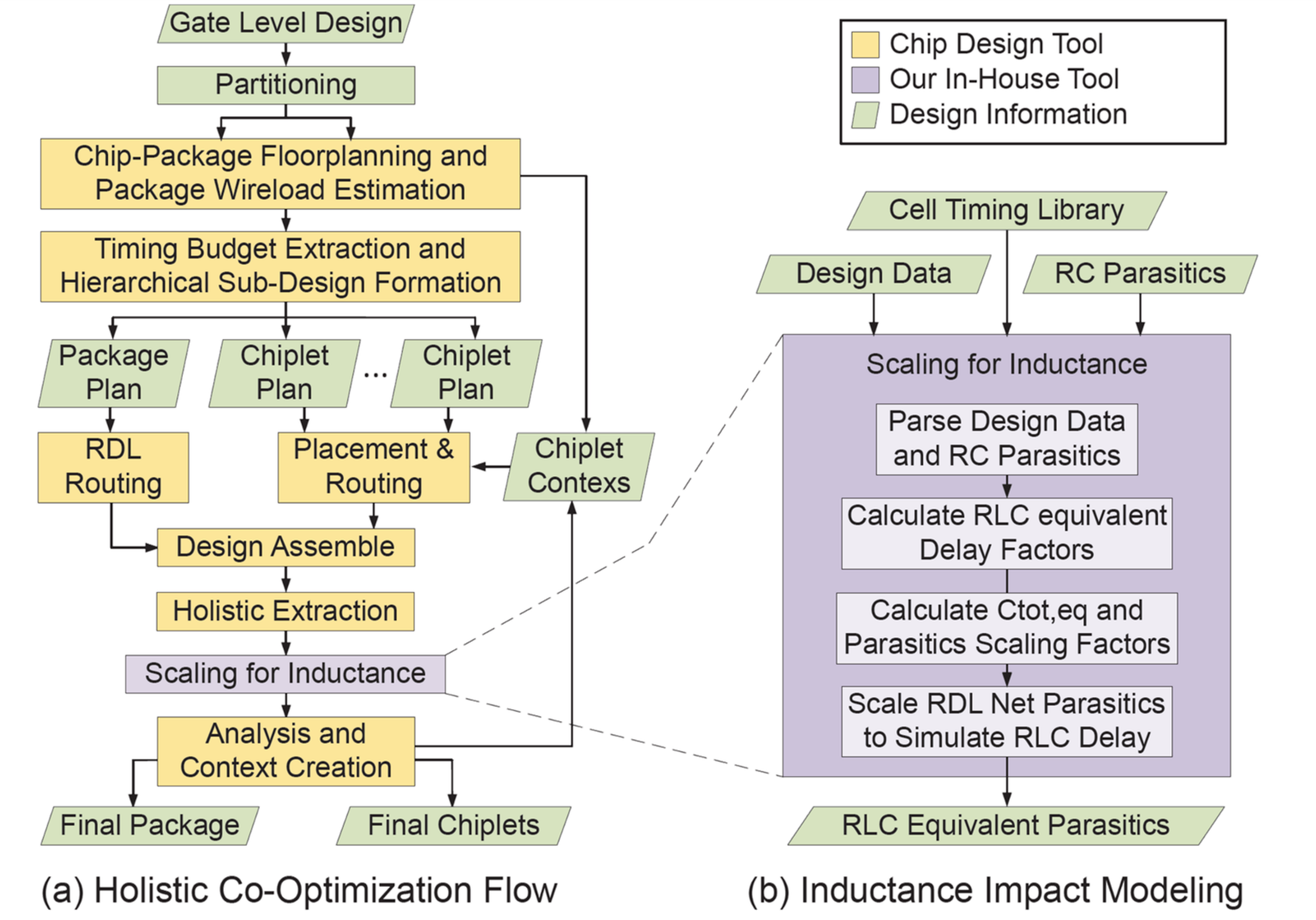

(d) Optimized 12KB Design

| Design | LPD | Frequency | Power | RDL WL |
|---|---|---|---|---|
| **(a)** | 2.50 ns | 400 MHz | 18.1 mW | 20.9 mm |
| **(b)** | 2.62 ns | 380 MHz | 19.7 mW | 46.8 mm |
| **(c)** | 2.56 ns | 390 MHz | 18.8 mW | 46.8 mm |
| **(d)** | 2.52 ns | 396 MHz | 18.8 mW | 35.5 mm |

[3] Md. Arafat Kabir, and Yarui Peng, “Holistic Chiplet-Package Co-Optimization for Agile Custom 2.5D Design”, (accepted) IEEE Transactions on Components, Packaging, and Manufacturing Technology, 2021.

# Inductance Consideration

- ❑ **Convert inductance to effective RC for CAD flow compatibility [4]**



Gate Level Design
Partitioning
Chip-Package Floorplanning and Package Wireload Estimation
Timing Budget Extraction and Hierarchical Sub-Design Formation
Package Plan
Chiplet Plan
...
Chiplet Plan
RDL Routing
Placement & Routing
Chiplet Contexs
Design Assemble
Holistic Extraction
Scaling for Inductance
Analysis and Context Creation
Final Package
Final Chiplets
Chip Design Tool
Our In-House Tool
Design Information
Cell Timing Library
Design Data
RC Parasitics
Scaling for Inductance
Parse Design Data and RC Parasitics
Calculate RLC equivalent Delay Factors
Calculate Ctot,eq and Parasitics Scaling Factors
Scale RDL Net Parasitics to Simulate RLC Delay
RLC Equivalent Parasitics

(a) Holistic Co-Optimization Flow (b) Inductance Impact Modeling

[4] Md. Arafat Kabir, Dusan Petranovic, and Yarui Peng, “Cross-Boundary Inductive Timing Optimization for 2.5D Chiplet-Package Co-Design”, (accepted) in Proc. ACM Great Lakes Symposium on VLSI, 2021.

# Conclusion and Future Work

## ❑ Conclusions

- Chiplet-Package interactions need to be considered in 2.5D systems
- Our flow effectively captures interactions between package and chiplet designs for holistic planning and optimization.
- Our flows can handle both homogeneous and heterogeneous designs making use of standard ASIC CAD tools with highly accurate extraction

## ❑ Future Work

- Cross-boundary RCLM extraction and study of their impacts
- Study of timing, signal and power integrity with full RCLM models
- Custom IO placement and RDL routing of 2.5D systems
- Cross-boundary optimization with active packages

# Thank You

**Any question?**

For more information, please visit E3DA Lab website:

https://e3da.csce.uark.edu