Coupling Extraction and Optimization for Heterogeneous 2.5D Chiplet-Package Co-Design

MD Arafat Kabir
University of Arkansas
makabir@uark.edu

Dusan Petranovic
Mentor Graphics
dusan_petranovic@mentor.com

Yarui Peng
University of Arkansas
yrpeng@uark.edu

Abstract

In recent years, 2.5D chiplet package designs have gained popularity in system integration of heterogeneous technologies. Currently, there exists no standard CAD flow that can design, analyze, and optimize a complete heterogeneous 2.5D system. The traditional die-by-die design approach does not consider any package layers during extraction and optimization, and an accurate chiplet-package extraction can not be applied to heterogeneous designs without fundamental changes in standard CAD tools. In this paper, we present our Holistic and In-Context chiplet-package co-design flows for high-performance high-density 2.5D systems using standard ASIC CAD tools with zero overhead on IO pipeline depth. Our flow encompasses 2.5D-aware partitioning, chiplet-package co-planning, in-context extraction, iterative optimization, and post-design analysis and verification of the entire 2.5D system. We design our package planner with a routing and pin-planning strategy to minimize package routing congestion and timing overhead. An ARM Cortex-M0-based microcontroller system is designed as the benchmark. The performance gap to the reference 2D design reduces by 62.5% when chip-package interactions are taken into account in the holistic flow. Our in-context extraction achieves only 0.71% and 0.79% error on ground and coupling capacitance on a homogeneous system. Further, we implement a heterogeneous 2.5D system to demonstrate our novel in-context design and optimization methodology.

Keywords

2.5D Design, Chiplet-Package Co-Optimization, Holistic, Heterogeneous, In-Context.

1 Introduction

In the post-Moore era, although transistor scaling and chip scaling are saturated, demands for increased functionality, performance, and bandwidth are still growing very fast. 2.5D integration technology is gaining popularity in increasing device density and performance at the system level. Moreover, it offers heterogenous integration and hardware security applications [13, 14]. To support this integration scheme, the industry is developing compact and high-performance Wafer-Level-Packaging (WLP) solutions. As depicted in Fig. 1(a), in system integration schemes using Printed Circuit Boards (PCB) the packages become sufficiently large compared to the chips. Moreover, the long interconnects between different parts of the system kill the system performance. The WLP integration solutions, as depicted in Fig. 1(b), (c) have chip-scale packages with a very fine pitch, and shorter interconnects, making them promising candidates for high-performance system design. In the last few years, the industry has developed WLP technologies like eWLB [1], SWIFT [5], and InFO [12]. All these technologies are bringing chips and packages closer and closer in their every iteration.

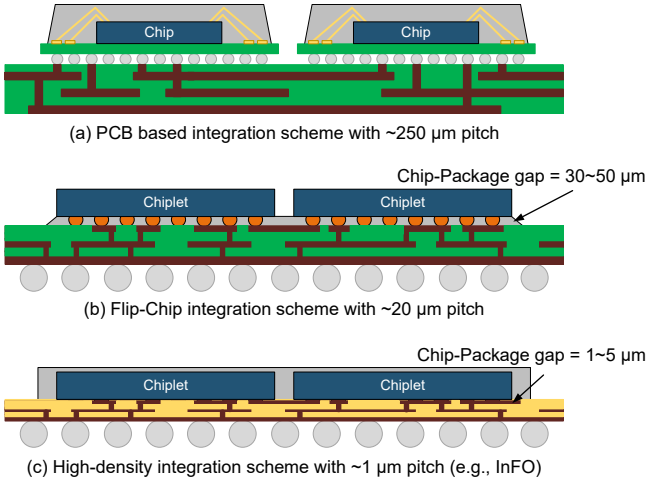


Figure 1: 2.5D integration schemes (a) PCB-based system, (b) flip-chip with an organic interposer, (c) high-density integration scheme such as wafer-level-packaging

In the current industry trend, all functional blocks of 2.5D systems are designed independently in their own design environments and then mounted on package redistribution layers (RDL) as a complete system [6, 7]. Fig. 2(a) illustrates this traditional flow. The analysis and optimization of chiplets and the package are also conducted separately, without consideration of the interactions between them. This traditional flow is sufficient when the gap between chip and package is large enough to make these interactions minimal. As shown in Fig. 1(b), this gap is around 30-50µm in the Flip-Chip WLP integration scheme. In such integration technologies, the traditional flow works fine. However, with the development of advanced WLP processes like InFO, this gap is decreasing rapidly. Starting from the order of 10µm, this gap was reduced to 1.5µm [11] within a few years. With this trend, the chip-package gap will soon reduce to the sub-micron level, making the interactions between chip and package more prominent. As a result, to ensure system reliability and signal integrity, chip-package interactions must be considered in timing and power analyses. In the current industry approach, it is

ICCAD '20, November 2–5, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8026-3/20/11…$15.00
https://doi.org/10.1145/3400302.3415718

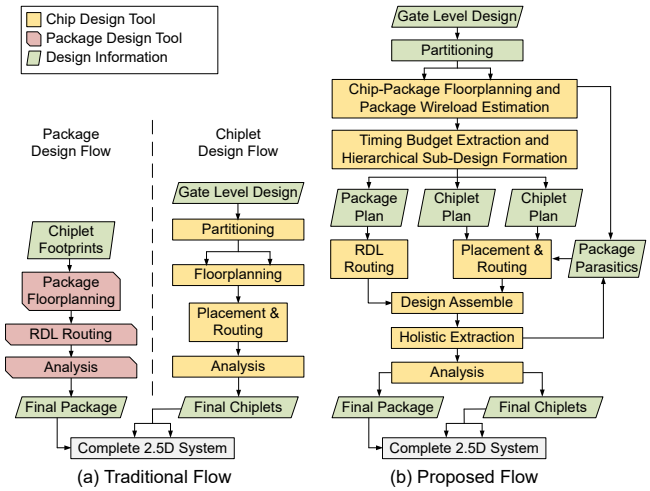


Figure 2: The traditional Die-by-Die design flow of a 2.5D system versus our proposed chiplet-package co-design flow with standard CAD tools

not possible to accurately capture the interactions among the tightly-connected components of a 2.5D system. Standard design flow is in demand to fill in the missing pieces and address the new challenges of 2.5D system design.

In the die-by-die design approach, it is possible to achieve the shortest design time using off-the-shelf chiplets to implement a 2.5D system. In this flow, chiplets and the package never actually interact with each other until after they are manufactured as a complete system. As seen in Fig. 2(a), all steps of design and optimization are performed independently in their own environments. However, to extract the best performance out of the system and to achieve high system reliability optimizations and analysis in each design step need to be performed in a holistic [10] way rather than considering individual parts independently without taking into account rest of the system.

Although the holistic design method is powerful and adaptive to any technologies, one fundamental issue is that it cannot be applied to heterogeneous systems using standard ASIC tools. The heterogeneity consists of multiple chiplets that are implemented in different technologies. For example, AMD designed a processor core chiplet in 7nm with an IO chiplet in 14/12nm technology. For a small design house, one challenge to implementing such designs is the limitation of the CAD tools. In addition, some designs are intrinsically very small in area and power budget, such as IoT devices. Having a large/high-performance IO system will create too much overhead for such system to be implemented with multiple chiplets. One solution is to allow highly-customized IO interface to be used between chiplets which can be simplified into a few standard cells. However, as these cells are not designed for driving long RDL wires with many technology variations, parasitics and STA analysis must be performed very carefully to avoid potential violations. To overcome these challenges, it is essential to design a CAD method for low-cost IO systems that reduces timing and power overheads but still captures all couplings between the chiplet and package to ensure all design constraints are met.

In this paper, we propose a chiplet-package co-optimization flow that incorporates the features required to achieve the design goals in a high-density 2.5D integration technology. Fig. 2(b) shows the overall steps of our flow. In this flow, we design 2.5D packages together with chiplets in the same design environment of the existing commercial chip design tools. This enables the exchange of necessary design information between the chiplets and the package during design and optimization steps. We also propose a novel in-context method to design heterogeneous systems using standard ASIC CAD tools. Our flows use all standard libraries to design custom pin drivers, achieve zero overhead on pipeline depth, and minimize the timing and power overhead.

Through the work presented in this paper, we claim the following contributions: (1) A unified tool flow that, for the first time, designs and optimizes chiplets and the package of high-density 2.5D systems together taking into account the mutual interactions between them; (2) A new holistic parasitic extraction and STA analysis flow for homogeneous 2.5D systems with chiplets and the package considered together; (3) A new in-context parasitic extraction and STA analysis flow for heterogeneous 2.5D systems with chiplets-package interactions captured; (4) A comparative study between two 2.5D designs to validate our Drop-in design approach and demonstrate chiplet-package interaction impacts on two 2.5D systems Performance, Power, and Area (PPA).

To our best knowledge, there exists no other tool flow that implements holistic planning and optimizations of high-density 2.5D systems, including placement and routing of chiplets and the packages together using commercial chip design tools.

2 CAD Flow and Reference Design

Our overall flow is demonstrated in Fig. 2(b). This flow is based on a previous work [4] that proposed a basic holistic design methodology for 2.5D systems. The first step of the flow is partitioning the synthesized netlist into chiplets. The partitioning tool takes into account the impacts of RDL and possible solution cases enabled by 2.5D integration while exploring the partition solutions. After partitioning, we perform the planning of chiplets and the package together in the same design environment. The PDK is modified to include the package layers along with the chiplet routing layers. Next, we generate an initial package routing and estimate the wire-loads at the chiplet pins. Then, we split the overall design into individual chiplet and package sub-designs for parallel implementation. After the co-planning and RDL routing, the chiplets and package can all be implemented independently in their own design environments, with constraints propagated from the top level.

Individual chiplets are implemented following the traditional 2D flow, using the top-level constraints and estimated package wire-loads. After the first iteration of placement and routing, the entire system is assembled for extraction. With this assembly, the extraction tool can capture the interactions among the routing layers across chiplets and the package. Using the extracted parasitics, we create timing contexts for the chiplets and perform the second iteration of chiplet design using these contexts. This ensures holistic optimization of the system and improves system performance if possible. Additional iterations can be carried out if there is scope of more improvement. Finally, we assemble all the finished designs and perform extraction for analysis and sign-off verifications.

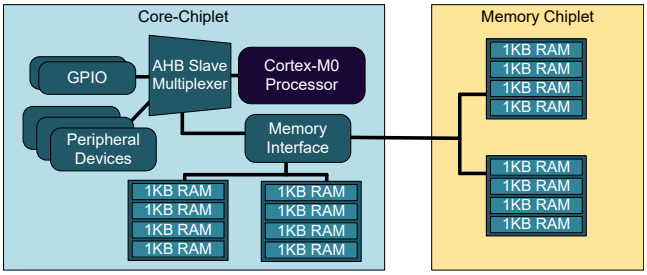


Figure 3: System architecture and chiplet partitions of the Cortex-M0-based reference design

For our case study, we design a micro-controller system based on ARM Cortex-M0 core using our flow with Nangate45nm PDK. The system consists of 16KB of memory and some peripheral devices commonly found in a micro-controller system, like UART, timers, etc. The processor core is connected to an AHB bus, which acts as the bridge to the rest of the system. The 16KB memory system has four 4KB banks. We further subdivide the 4KB banks into four 1KB memory blocks. With such a granular design of the memory system, we have more options while performing partition and floorplan. The system architecture and chiplet partitions of the micro-controller system are shown in Fig. 3. The core-chiplet contains all the logic blocks and 8KB memory while the mem-chiplet contains only the rest 8KB of the memory. We use OpenRAM [2] memory compiler to compile the 1KB memory module with a one-byte word size.

Currently, standard chip design CAD tools do not support package routing layers. To perform holistic planning and verification of a 2.5D system, we need to load the chiplet and package designs in the same design environment. For this reason, we modify the Nangate45nm technology to support both chiplets and the package designs together in a chip design environment. We use M1-M7 for chiplet internal routing and adjust the top three layers, M8-M10, to mimic TSMC 2.5D InFO package routing layers. Table 1 and Fig. 4 together describe our settings for the package layers.

3 Chiplet-Package Co-Planning and Modeling

In the traditional flow, floorplans of package and chiplets are prepared independently, without considering the interactions among them. If the package routing is not planned carefully, though each chiplet might achieve very high performance, the entire system will perform poorly because of the timing bottlenecks through package wires. At this step of our flow, we aim to plan the chiplet pin configuration and package floorplan in a holistic way to minimize the package-routing-related issues.

3.1 Package Floorplanning and RDL Routing

In 2.5D systems, the RDL wires act as the timing bottlenecks, so performance and signal integrity considerations play the main role in RDL routing. Existing works [3, 8, 9] try to solve the routability between chiplet pins in the system. However, there are much fewer RDL nets compared to intra-chip connections. As a result, routability and minimization of total wire-length are not the primary concerns. We develop an RDL planning tool that implements our strategy of chiplet-package floorplanning. It takes in chiplet netlists, technology, and timing information to generate package floorplan, RDL routing, and package wire-load estimations.

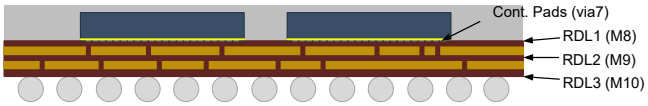


Figure 4: Package redistribution layer stack of our modified Nangate45 PDK

Table 1: Parameters (in µm) of our modified Nangate45 PDK routing layers

	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

In our strategy, we focus on developing a compact RDL routing plan with short and uniform wire-lengths to minimize routing issues like congestion, detours, and unequal bus wire delays between chiplets. We consider two chiplets at a time at the track assignment step. At first, we assign tracks to the pins of each chiplet separately. Then, a package floorplan is determined that connects all the tracks between the chiplets. Two pins of different chiplets assigned to the same track are considered connected through the package wires. Fig. 5 shows the routing generated following our strategy. With the connectivity defined, signal assignment of the chiplet pins are determined using a greedy strategy based on net-slacks reported by the synthesis tool.

3.2 Package Wireload Estimation

After RDL routing, we calculate a rough estimation of package wire-loads at the chiplet pins for the first iteration of chiplet implementation. For this estimation, package wire-load is calculated as a linear function of the wire-length. This estimation is used to inform the chiplet implementation tool about the loads at the output of the driving cells of the pins. Being aware of the output load, during the optimization steps, it can make necessary adjustments like buffer insertion, cell resizing at the output nets. After the first iteration of chiplet implementation, we extract the parasitics from the assembled design and use it to complete the second iteration of chiplet implementation. Subsequent iterations can be performed until design timing results converge. However, if the first estimate is good enough, the second iteration should be sufficient to meet a practical performance goal.

4 Chiplet-Package Co-Design and Optimization

The physical design of individual chiplets and the package can be performed using any commercial chip design environment that supports hierarchical design flow. With the modified version of Nangate45nm technology, we load the entire system, chiplets and the package together, into the design environment. The chiplets are defined as modules in the partitioned netlist. We perform the placement and pin assignment of each module (chiplet) and package routing using scripts generated by our RDL planner tool. Then, we extract the timing budgets of chiplets and the package. After this step, the modules are separated as hierarchical sub-designs, and the package design is saved as the top-level design.

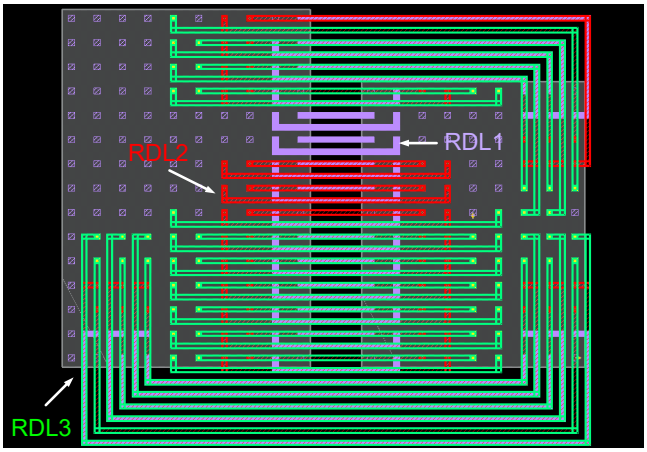


Figure 5: Inter-chiplet routing generated in our strategy. Some RDL2 and RDL3 routes are hidden to show other layers.

4.1 Chiplet Implementation

During implementation, each chiplet is treated as a single 2D chip and designed using traditional chip design techniques. We modify the chiplet design constraint files to include the wire-loads at the chiplet pins estimated by our RDL planner tool. We already have the pin placement and the initial floorplan prepared at the top level. However, this floorplan can be adjusted if necessary as long as pin configurations are unchanged. After the floorplan is finalized, the power distribution network is designed to ensure uniform power delivery to all the parts of the chiplet. Then, we use standard tools to perform the standard cell placement, power routing, clock network design, routing, and timing optimization. Lastly, filler cells and metal fills are added to finish the chiplet design. Fig. 6(a) shows the core-chiplet, which contains all the logic blocks and 8KB memory. Fig. 6(b) shows the extended memory chiplet, which contains the other 8KB memory.

4.2 Package Implementation

With the timing budget and RDL plans generated by the RDL planner tool, package designing can be finished in parallel alongside the chiplet implementations. However, more accurate and reliable optimizations can be performed on the package design if we use the chiplets interface timing models extracted after their implementations are complete. Because of the differences in the package and chip routing techniques, chip routing tools do not produce good routing for the package routing layers. Based on the strategy explained in Section 3, our RDL planner generates routing scripts for package routing. We utilize these scripts to perform the routing between the chiplets. After inter-chiplet routing is finished, we manually route the rest of the I/O pins of the core-chiplet to the package I/O pads. Fig. 7(b),(c) show the package routing of the two versions of the example 2.5D system designed using the Drop-in method. The package wires connecting the two chiplets in Fig. 7(c) resemble the routing generated by the RDL planner tool.

4.3 Holistic Extraction

With implemented chiplets and the package layouts, they are ready to be assembled in the integrated design environment again. We load the modified version of Nangate45nm PDK in the chip design tool and assemble the chiplets on the top-level design (package design) using the partitions assemble commands of the tool. Though this is a common step in the traditional 2D chip design flow, in 2.5D systems, it is the step to harness some important benefits of this integration technology. Various interesting design techniques like plug-and-play [6], Drop-in design can be adopted at this step.

Using the Drop-in approach, we design two 2.5D systems with the chiplets. Fig. 7(b) and (c) resemble these two systems. As both chiplets and the package are together in the same environment, it is possible to accurately capture the interactions among them. This creates a scope to perform some incremental optimizations of each part of the system to further improve the overall system performance and reliability. Fig. 7(d) is a zoomed-in view of the assembled system that clearly shows wires from the chiplet and package altogether. The wide horizontal wire marked RDL3 is a package wire connecting pins of the two chiplets. The vertical wire marked M6 is a part of the power ring within the core-chiplet. The horizontal wires marked M1 are core-chiplet wires that connect the power and ground rails of the standard cell rows to the power/ground ring on M6.

4.4 Iterative Optimizations

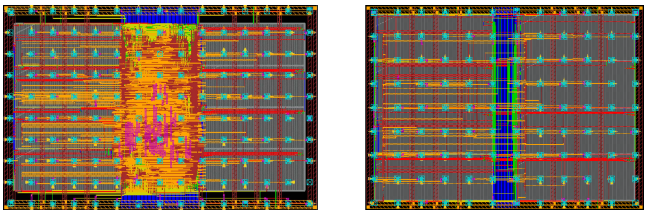
After design assembly and extraction, we run analysis to verify system performance. In this first iteration, as the package wire-loads are just rough estimates, we almost always expect some room for improvement. After the first iteration of design assembly, the extraction tool can provide accurate parasitic information, and the analysis tool can generate a tighter timing budget. In the following iterations, instead of using the estimated wire-loads and timing budgets, we use the timing contexts extracted after design assemble of previous iteration. With the new implementation of the chiplets, we can perform another round of design-assemble and analysis. There can be multiple iterations of this process, each time with more accurate parasitics and timing budget until it is no longer possible to improve system performance or the target performance is met. However, with a good estimation generated by our RDL planner, only a second iteration would be enough to verify and close the discrepancies. To justify this argument, in the next section, we present three design cases with no estimate, a rough estimate, and a near-accurate estimation of parasitics and timing budget.

5 Holistic Design for Homogeneous Systems

5.1 Design Variations

To study the impact of our flow on 2.5D system design, we prepare several design cases as presented here. In the results section, we perform a comparative study among these design cases. Table 3 shows the design parameters of these designs.

5.1.1 Case-1: Reference 2D Design Fig. 7(a) shows the finished 2D chip design. For this 2D implementation, we use the synthesized netlist prepared before the partition stage. The die area is a square with a side length of 550µm. Though the standard cells occupy approximately 10% of the area, this floorplan allows them to spread out in all directions to some extent. We perform the design steps like standard cell placement, clock tree synthesis, and time design, routing, and post-routing optimizations using the tools integrated with the chip design environment.



(a) Core System Chiplet (b) Extended Memory Chiplet

Figure 6: Layouts of the chiplets for 2.5D integration

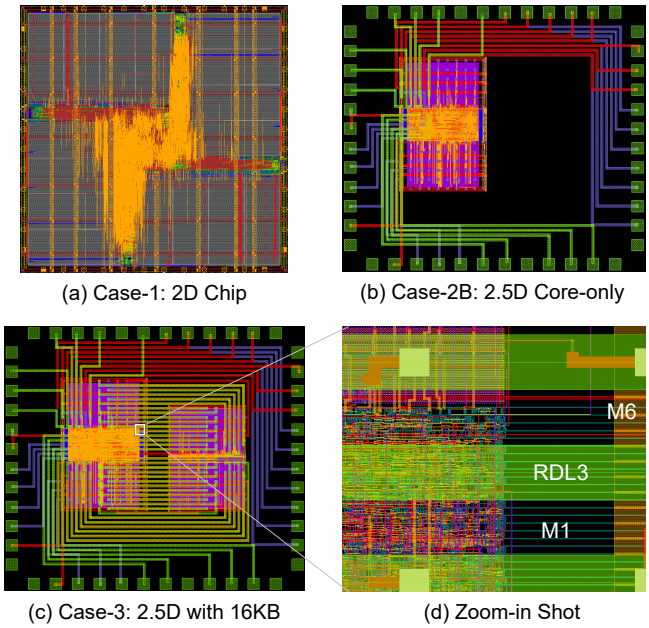
5.1.2 Case-2: 2.5D Designs The first 2.5D design (Case-2A) is a Context-Free Single-Pass 2.5D design. In this design, the RDL planner tool does not estimate the impact of package wires. Fig. 6 shows the finished chiplets prepared in the Case-2A design. 2.5D integration enables several flexible design approaches like plug-and-play, Drop-in method. Based on Case-2A design, we implement two systems (Case-2B) using the Drop-in design approach at the design assemble stage. In one design, we have just the core-chiplet in the package, which is a fully-functional system with only 8KB of memory. Fig. 7(b) shows this core-chiplet only system. In the other design, we include both chiplets in the package where the complete system has 16KB memory. Fig. 7(c) resembles this extended memory system. Table 4 presents a comparative study of these two Drop-in systems.

5.1.3 Case-3: Context-Aware Optimized 2.5D Designs In this design, we try to include chip-package interactions in the design and optimization steps as much as possible. Unlike the traditional flow, we perform iterative improvement to achieve the maximum achievable performance out of the 2.5D system. We use the RDL planner tool to generate top-level floorplans, package routing, greedy signal assignment, and estimated wire-loads. After hierarchical sub-design formation, we use the estimated wire-loads to implement the chiplets in the first iteration. After the second iteration, we performed more iterations. As there was no additional improvement in system performance, we take the second iteration output as the final design for this design case.

5.2 Holistic Extraction and Analysis Results

Traditional industry-standard flow uses FEM tools to perform package extraction with S-parameters to determine package power and signal integrity. Unlike the traditional flow, the entire 2.5D system is in the same design environment after design assembly. Using this holistic extraction result, our flow can achieve more accurate and reliable analysis results. The extraction results of Design Case-3 final iteration are presented in Table 2. For readability, we lumped the coupling capacitances among layers M1-M5.

As observed from the extraction result, there exists sufficient coupling between RDL1 of package and M6 of chiplets. Though the top routing layer of the chiplets is M7, as seen in Table 3, the total wire-length on M7 is very small compared to that on M6. That is why the RDL1 coupling capacitance value with M6 is greater than that with M7. Such detailed interaction between chiplet and package can only be captured through a holistic extraction process as presented in our flow. This chip-package coupling, along with the delay introduced by the package wires, greatly affects the system performance. Shown in Table 3, our analysis flow reveals this performance degradation. After all the possible traditional chip-level optimizations of



(a) Case-1: 2D Chip (b) Case-2B: 2.5D Core-only

(c) Case-3: 2.5D with 16KB (d) Zoom-in Shot

Figure 7: Design layouts of (a) monolithic 2D reference design, (b) assembled Core-only system, (c) assembled 2.5D system with extended memory, and (d) zoom-in shot.

Table 2: Coupling and ground capacitances (in fF) between routing layers of design Case-3 final iteration

	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

Design Case-2, the best system performance we achieve is 245MHz. This is significantly slower than the performance of the reference 2D system, which is 333MHz. In Design Case-3, we apply our own strategies to minimize this degradation.

The estimated package wire-loads used in the first iteration of chiplet design are calculated purely as a linear function of package wire-length and do not consider the coupling. Because of this simplified wire-load model, on average, the wire-loads are underestimated by the RDL planner tool. However, even with this crude estimation, we can achieve a good performance improvement over the Case-2 Design. As observed from Table 3, after the first iteration of Case-3 Design, which uses this estimated wire-load, we achieve a system frequency of 280MHz, which is approximately 40% reduction in the performance gap between the reference 2D design and the Case-2 Design. This result reveals the significance of the holistic consideration of chiplet-package interactions, even at the early design steps of 2.5D systems.

Table 3: Comparison of die/chiplet analysis results of design cases with both chiplets

Design Case	Case-1	Case-2		Case-3 first iteration		Case-3 2nd/final iteration	
Chip Design	2D Chip	Core Chiplet	Mem. Chiplet	Core Chiplet	Mem. Chiplet	Core Chiplet	Mem. Chiplet
Logic Gates#	17595	17783	132	17915	148	18214	45
Buffer/Inverter#	3700	2740	132	2865	148	2955	45
Die Size (μm×μm)	550×550	390×590	350×470	390×590	350×470	390×590	350×470
Total Chip Wirelength (mm)	412.9	350.9	40.14	361.2	45.07	366.3	41.99
M6 Wirelength (mm)	79.94	30.81	5.986	31.86	8.201	31.42	8.445
M7 Wirelength (mm)	0	1.783	0.598	1.875	0.589	2.02	0.624
Max Frequency (MHz)	333	245		280		300	
Performance Gap	0%	100%		60.23%		37.50%	
Chip Power (mW)	10.6	7.751	0.194	9.043	0.216	9.840	0.162

Table 4: Comparison between Drop-in systems

System Design	Core Only	Full System	Optimized
Total Memory	8KB	16KB	16KB
Chip-Package Cap	120.7864fF	217.4089fF	232fF
Max Frequency	300MHz	245MHz	300MHz
System Power	9.578mW	8.26mW	10.0mW
Pakacage wirelength	35.41mm	94.027mm	94.027mm
Package Size	1.3mm x 1.15mm		

5.3 Iterative Optimization Results

In the second iteration of Case-3 Design, we use the extracted parasitics from the assembled design of the first iteration. This iteration adjusts the chiplet designs to match the actual capacitive loads on driving cells. The adjustments in the second iteration can be observed from the buffer counts of the chiplets in Table 3. Compared to the first iteration design, the total buffer count of the core-chiplet is increased. However, the total buffer count of memory-chiplet is significantly decreased in the second iteration. This is because, though the buffers driving the previously under-estimated wire-loads are up-sized by two to four times, the smaller buffers driving the over-estimated wire-loads are completely removed. All these adjustments are performed by the traditional chip design tools without applying any special settings. Moreover, the system performance further improved to 300MHz, which is very close to our reference 2D design case.

Comparisons between these designs clearly show that it is essential to consider the chip-package interactions holistically to obtain the best performance out of a 2.5D system. Our flow aims at achieving this holistic co-design and optimization goal starting at the early planning stage of the design. Shown in Table 3, with minimal chip-package interaction consideration, the performance gap between the reference 2D system and Case-2 2.5D system is 88MHz. Even with an early estimation in the first iteration, we close this performance gap in the first iteration of Case-3 Design by around 40%. Finally, with the iterative approach, we close this gap by 62.5% in the second iteration.

5.4 Comparative Study of Drop-in Designs

Table 4 shows the comparison between two 2.5D systems designed using the Drop-in approach. The Chip-Package coupling capacitance is larger for the memory extended full system because of more package wires, as seen in Fig. 7(b),(c). The critical timing path for the extended system is between the core and memory chiplets. As a result, in the absence of the extra memory chiplet, the system can operate at a higher system frequency. As Table 4 shows, the Core-Only system can run at 300MHz while the full system without optimizations runs at 245MHz. The Core-Only system can be a low-cost, high-performance solution for the applications where 8KB memory is sufficient. On the other hand, the memory extended system is suitable for memory-intensive applications. Between these two implementations of the system, the only change needed is in the package level design, which is much cheaper and easier than making changes in the chip level designs. This approach offers application engineers the opportunity to make tradeoffs between cost, performance, and memory while selecting his implementation system. Designers can utilize our holistic flow to take full advantage of this design approach and implement several flavors of a 2.5D system per application needs.

Table 5: Comparison of Holistic (Holi) vs In-Context (In-C) ground (GCAP) and coupling (CCAP) capacitance extraction results (in fF) of Case-3 final homogeneous design.

Metal Layer	M1-M5	M6	M7	R1	R2	R3
In-C GCAP	21119	2053	273	1103	306	696
Holi GCAP	21119	2054	272	1040	247	636
In-C GCAP Err	0.00%	-0.01%	0.09%	6.03%	24.0%	9.46%
In-C CCAP	9171	1265	153	1563	2489	1765
Holi CCAP	9172	1263	156	1544	2421	1721
In-C CCAP Err	-0.01%	0.17%	-2.10%	1.20%	2.81%	2.56%

6 In-Context Design for Heterogeneous Systems

Since the holistic flow requires to assemble the chiplets into a unified design environment, it cannot be applied to heterogeneous systems where the device stack are different. At the present, no standard CAD flows support including two different technology files into a single physical design tools. Therefore, we present our in-context design method which allows an arbitrary number of chiplets in different technologies integrated with chiplet-package coupling considered altogether. It is completely compatible with all standard ASIC tools for design, extraction, and analysis.

6.1 In-Context Design and Validation Results

The first step is to create in-context designs as another level of design hierarchy. The context of a chiplet should include the area covering the whole chiplet and necessary neighboring regions. This ensures all chiplet-package interactions are considered during the

Table 6: In-Context heterogeneous design results with 7M3R core chiplet in Nangate45 and 6M3R Mem chiplet in gscl45.

Design iteration	LPD (ns)	Max Frequency	
with RDL wireload	3.55	281 MHz	
In-Context 1st iteration	3.35	298 MHz	
In-Context 2nd/final	3.35	298 MHz	
Final Design	Wire	Cell	Total
In-Context Power (mW)	4.29	6.22	10.51

extraction. Note that each in-context chiplet can be implemented with different technology files. Therefore, heterogeneous systems are partitioned into several sub-designs, where each one is an extended 2D design.

Once all bare chiplets are converted into the in-context chiplets, a top-level design is generated to connect the individual in-context chiplets into a merged system. However, as the top-level design does not need details within each in-context chiplets, only RDL routing layers are included in the design. This hides the device layer to the top-level, thus entire heterogeneous systems can be assembled. Standard extraction tools can then be used to perform extraction on each in-context design and the top-level, and the SPEF files are stitched with hierarchical annotations.

We implement this flow using our RDL planner. It partitions the floorplan and RDL routing into each chiplets and create the corresponding design hierarchy with Verilog netlists. Then, the RDL wires are imported during the chiplet implementation to form the in-context chiplets. The extracted SPEF files are then merged using an STA tool. To validate our in-context extraction, the best strategy is compare it against the holistic method. Since the holistic method can only be applied to homogeneous systems, we performed the in-context design extraction on the homogeneous system which is designed using the holistic design method. Two in-context chiplets are created all in our modified Nangate45 with seven metal and three RDL layers (7M3R).

The extraction results are compared in Table 5. As results show, our in-context extracted ground (GCAP) and coupling (CCAP) capacitance are highly close to the holistic extraction on all chiplet layers, even though the chiplets only see a partial design of the package. The total ground capacitance between holistic (25550 fF) and in-context (25367 fF) extraction is only 0.71% while coupling capacitance difference between holistic (16278 fF) and in-context (16406 fF) extraction is only 0.79%. However, the package layer capacitance are slightly overestimated, especially on the ground capacitance. This is mostly because of the RDL partition impact. With RDL wires separated into multiple designs, the fringe capacitance is computed on the cutting edge of the package wires. However, as these cutting faces do not exist in the holistic design, the capacitance are extracted correctly. This needs a small modification to existing extraction tools, or they can be subtracted from the results for in-context design in the future.

6.2 In-Context Heterogeneous Design Results

Since our in-context design strategy targets heterogeneous systems, to demonstrate its capability, we design a new Mem chiplets in a different PDK. However, since the OpenRAM compiler only supports a limited selection of technologies, we implement the same

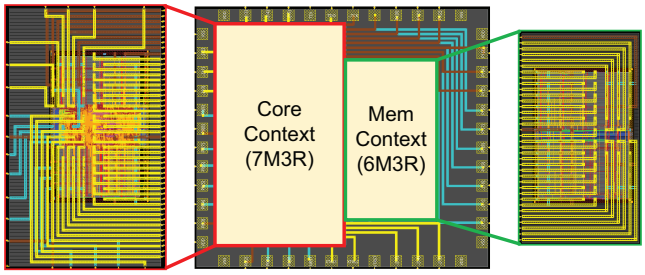


Figure 8: Layouts of the in-context chiplets for heterogeneous integration

Mem chiplet using gscl45 cell library which is bundled with the FreePDK45. Instead of using the same 7M homogeneous technology, the Mem chiplet only uses six metal layers (6M3R). The Core chiplet remains the same with the 7M Nangate45 PDK, forming a heterogeneous system with the Mem chiplets. Fig. 8 shows the layout of our heterogeneous systems with two in-context chiplets.

We performed the in-context extraction on each chiplets and followed the timing optimization flow similar to the holistic design flow. As observed from Table 6, in-context extraction has successfully enabled the iterative timing optimization on the heterogeneous designs. The designs with RDL wire-load estimation and final iteration of Table 6 correspond to the Case-3 first and final iterations of Table 3, respectively. Comparing these designs, we can see that in-context designs achieve the same performance as the holistic designs in the corresponding iterations. The minor changes in performance are caused by the small errors in the in-context extraction results as previously discussed. However, the results still prove that our in-context flow, which can support heterogeneous techonologies, is as effective as the holistic approach.

7 Conclusions

In this paper, we present our holistic and in-context flow to design and optimize chiplets and the package of a 2.5D system in commercial chip design environments. The system performance and reliability are highly affected by the chip-package interactions in high-density 2.5D integration schemes. Unlike traditional die-by-die design flow, both of our flow takes into account these interactions in design, optimization, and analysis steps of the system. In our experimental designs, we found significant coupling between chiplet and package routing wires. Careful considerations of these interactions in planning and design stages in a holistic way can significantly improve the system performance. With proper planning, holistic extraction, analysis, and iterative design optimization, we reduced the performance gap between the 2D chip and 2.5D system by 62.5%. Our in-context design flow matches with the holistic flow closely but further enables the heterogeneous design. Moreover, our flow supports the Drop-in design approach enabled by 2.5D integration technology which offers design flexibility.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1755981. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Brunnbauer, T. Meyer, G. Ofner, K. Mueller, and R. Hagen. 2008. Embedded Wafer Level Ball Grid Array (eWLB). In *International Electronics Manufacturing Technology Conference*. 1–6. https://doi.org/10.1109/IEMT.2008.5507866

[2] Matthew R. Guthaus, James E. Stine, Samira Ataei, Brian Chen, Bin Wu, and Mehedi Sarwar. 2016. OpenRAM: An Open-source Memory Compiler. In *International Conference on Computer-Aided Design*. 93:1–93:6. https://doi.org/10.1145/2966986.2980098

[3] Jia-Wei Fang and Yao-Wen Chang. 2008. Area-I/O flip-chip routing for chip-package co-design. In *International Conference on Computer-Aided Design*. 518–522. https://doi.org/10.1109/ICCAD.2008.4681624

[4] MD Arafat Kabir and Yarui Peng. 2020. Chiplet-Package Co-Design For 2.5D Systems Using Standard ASIC CAD Tools. In *Asia and South Pacific Design Automation Conference*. 351–356. https://doi.org/10.1109/ASP-DAC47756.2020.9045734

[5] W. Ki, W. Lee, I. MoK, I. Lee, W. Do, M. Kolbehdari, A. Copia, S. Jayaraman, C. Zwenger, and K. Lee. 2018. Chip Stackable, Ultra-thin, High-Flexibility 3D FOWLP (3D SWIFT® Technology) for Hetero-Integrated Advanced 3D WL-SiP. In *IEEE Electronic Components and Technology Conference*. 580–586. https://doi.org/10.1109/ECTC.2018.00092

[6] Jinwoo Kim, Gauthaman Murali, Heechun Park, Eric Qin, Hyoukjun Kwon, Venkata Chaitanya, Krishna Chekuri, Nihar Dasari, Arvind Singh, Minah Lee, Hakki Mert Torun, Kallol Roy, Madhavan Swaminathan, Saibal Mukhopadhyay, Tushar Krishna, and Sung Kyu Lim. 2019. Architecture, Chip, and Package Co-design Flow for 2.5D IC Design Enabling Heterogeneous IP Reuse. In *Design Automation Conference*. 178:1–178:6. https://doi.org/10.1145/3316781.3317775

[7] W. Liu, Min-Sheng Chang, and T. Wang. 2014. Floorplanning and signal assignment for silicon interposer-based 3D ICs. In *Design Automation Conference*. 1–6. https://doi.org/10.1145/2593069.2593142

[8] J. Minz and S. K. Lim. 2006. Block-level 3-D Global Routing With an Application to 3-D Packaging. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25, 10 (Oct 2006), 2248–2257. https://doi.org/10.1109/TCAD.2005.860952

[9] J. R. Minz and Sung Kyu Lim. 2004. A global router for system-on-package targeting layer and crosstalk minimization. In *Electrical Performance of Electronic Packaging*. 99–102. https://doi.org/10.1109/EPEP.2004.1407557

[10] Y. Peng, T. Song, D. Petranovic, and S. K. Lim. 2017. Parasitic Extraction for Heterogeneous Face-to-Face Bonded 3-D ICs. *IEEE Transactions on Components and Packaging and Manufacturing Technology* 7, 6 (June 2017), 912–924. https://doi.org/10.1109/TCPMT.2017.2677963

[11] H. Pu, H. J. Kuo, C. S. Liu, and D. C. H. Yu. 2018. A Novel Submicron Polymer Re-Distribution Layer Technology for Advanced InFO Packaging. In *IEEE Electronic Components and Technology Conference*. 45–51. https://doi.org/10.1109/ECTC.2018.00015

[12] C. Tseng, C. Liu, C. Wu, and D. Yu. 2016. InFO (Wafer Level Integrated Fan-Out) Technology. In *IEEE Electronic Components and Technology Conference*. 1–6. https://doi.org/10.1109/ECTC.2016.65

[13] Y. Xie, C. Bao, Y. Liu, and A. Srivastava. 2016. 2.5D/3D Integration Technologies for Circuit Obfuscation. In *International Workshop on Microprocessor and SOC Test and Verification*. 39–44. https://doi.org/10.1109/MTV.2016.17

[14] Y. Xie, C. Bao, and A. Srivastava. 2017. Security-Aware 2.5D Integrated Circuit Design Flow Against Hardware IP Piracy. *Computer* 50, 5 (May 2017), 62–71. https://doi.org/10.1109/MC.2017.121