Holistic 2.5D Chiplet Design Flow: A 65nm Shared-Block Microcontroller Case Study

MD Arafat Kabir and Yarui Peng
Computer Science and Computer Engineering Department, University of Arkansas
makabir@uark.edu and yrpeng@uark.edu

***Abstract*—Traditionally, different components of a system are integrated through Printed Circuit Boards (PCB). The long traces on PCB have severe power loss and limit the bandwidth of the interconnects between the components. Advanced packaging offers high-bandwidth, low power, and high-performance inter-die communications with compact sizes and dense pin arrays. 2.5D integration further provides better thermal dissipation, lower cost, and higher yield compared to 3D stacking. Novel CAD tool flows dedicated to 2.5D chiplet designs are essential to enable flexible and efficient 2.5D system designs. In this paper, we present our design, optimization, and analysis methodologies and a design case study implementing an ARM Cortex-M0 microcontroller system using a holistic 2.5D tool flow. We use TSMC 65nm as our chiplet implementation technology with a modified metal stack referring to 2.5D Fan-Out Wafer-Level Packaging (FOWLP) solutions. We also discuss design techniques for chiplet reuse and the Drop-in design approach to develop low-power, low-cost, and high-performance flavors of a 2.5D system. We compare the 2.5D system with its 2D counterpart to validate the holistic design flow.**

***Keywords*—2.5D Design, Chip-Package Co-Design, Redistribution Layer Planning, Shared Block Tape-out, Drop-in**

I. Introduction

To support the ever-growing demand for increased functionality and performance, the sizes of modern chips such as GPU, FPGA, AI accelerators are reaching the reticle limit. Increased chip-size comes with high design complexity, longer wire-lengths, higher power consumption, and lower yield. As a result, the industry has developed the System-in-Package (SiP) design approach, where a complicated system is divided into smaller chiplets and then integrated as a whole system on the package. This modular design offers increased flexibility, reduced complexity, short chip wire-lengths, and heterogeneous integration. Traditionally, a Printed Circuit Board (PCB) is used as the system integration platform. Illustrated in Fig. 1(a), PCB design is simple, fast, and cheap. However, the interconnections through the PCB have long wirelength, high inductance and capacitance, limited bandwidth, and suffer from severe power and signal loss. As a result, the industry has developed 2.5D and 3D packaging for energy-efficient inter-chip communications. Fig. 1(b) and (c) illustrate a TSV-based 3D IC and a silicon-interposer-based 2.5D system, respectively. Previous studies [1, 2] have demonstrated orders of magnitude improvement on interconnect bandwidth and power efficiency in 2.5D and 3D systems compared to PCB-based systems. Along with these benefits, 2.5D and 3D system designs offer a compact package size, which makes them attractive candidates for portable devices. However, though 3D ICs have smaller form factors and higher bandwidth compared to 2.5D systems, it suffers from poor thermal dissipation and lower yield. The Wafer Level Packaging (WLP) process using Know-Good-Dies improves performance, power consumption, and cost of production of 2.5D systems. Moreover, a 2.5D system provides heterogeneous integration capability, where technology-specific optimization techniques can be applied to individual chiplet to further reduce the overall area and power consumption of the system. As a result, the 2.5D design approach is the most attractive candidate, especially for cost-sensitive low-power mobile systems.

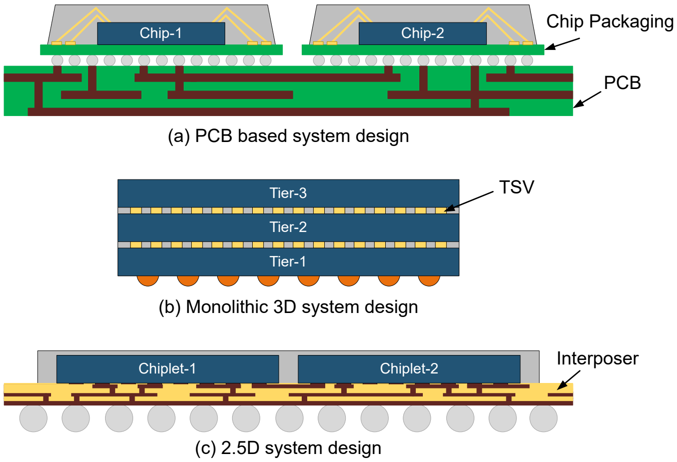
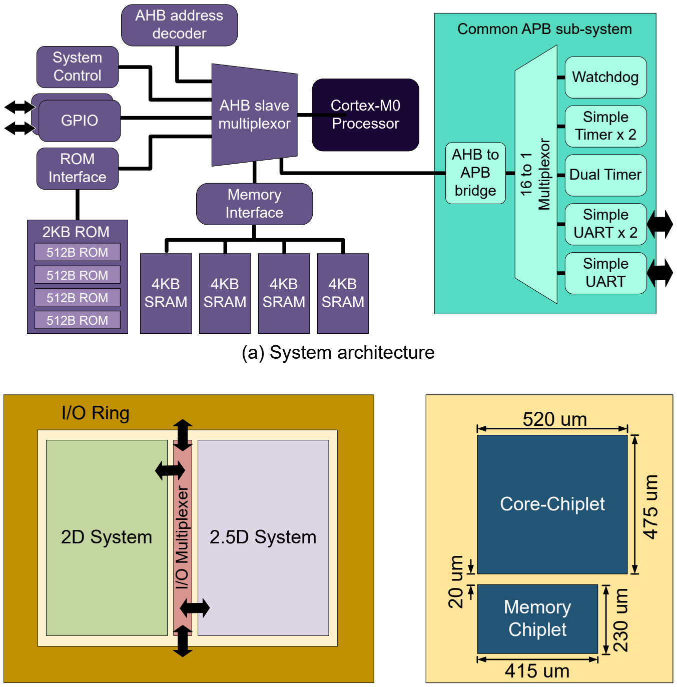


Fig. 1. System Design technologies: (a) PCB based system (b) TSV based monolithic 3D system (c) High density 2.5D integration scheme

In recent years, both industry and academia are investing great efforts in the development of 2.5D integration technology. Various integration schemes like Flip-Chip (FC), Package-on-Package (PoP), Ball-Grid-Array (BGA) have been explored using various substrate materials, including glass, ceramic, organic, and silicon. At this moment, there exist a few advanced high-density options like eWLB, SWIFT, and InFO [3]. To drive the interconnects through interposer layers novel high-speed and low power I/O circuits are developed with standard interface protocols [4]. Novel system design approaches like plug-and-play and Drop-in methods are also investigated for agile ASIC design. Dedicated algorithms and strategies to perform floorplanning, package routing, I/O redistribution [5, 6] of 2.5D systems are proposed. A recent published work [7] presented a holistic design methodology that can design, optimize, and analyze a complete 2.5D system using standard ASIC design tools. In this paper, we demonstrate the application of the holistic flow on a practical chip design technology (TSMC65nm) and present the analysis results. We also present a shared-block tape-out technique to design a chip that can be used for a comparative study between the 2.5D system designed in the holistic flow and a reference 2D system. This chip is fabricated on silicon to validate the flow.

Through the work presented in this paper, we claim the following contributions: (1) Demonstration of design, optimization, and analysis techniques of a 2.5D system in commercial chip and packaging technologies; (2) Comparative study between 2D and 2.5D systems

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.



(a) System architecture

(b) Integration plan for tape-out

(c) 2.5D Package floorplan

Fig. 2. (a) Architecture of the ARM Cortex-M0 micro-controller system (b) Integration floorplan for shared-block system design (c) Package floorplan generated by RDL planner tool

designed in a commercial technology; (3) Design innovations for shared-block tape-out for comparative study between designs; (4) Application of Drop-in design approach to develop low-power flavors of a 2.5D system. To our best knowledge, there exists no previous work that discusses holistic design, optimization, and analysis methodologies to implement an entire 2.5D system in a commercial technology with tape-out designs.

II. Design Settings and CAD Flow

A. System Architecture

The micro-controller system has an ARM Cortex-M0 processor core, 16KB of memory, bootloader ROM, and some common peripheral devices. The entire system organization is shown in Fig. 2(a). The AHB bus is connecting the processor core to an AHB address decoder, a system controller module, an APB sub-system, two GPIO modules, the ROM interface, and the memory interface. The APB sub-system is connected to the AHB bus through a multiplexer and an AHB-to-APB bridge. The UARTs of the APB sub-system share pins with the GPIO ports to reduce the system pin count. The bootloader ROM is 2KB in total and is divided into four 512B banks. The data memory system consists of four 4KB memory blocks.

B. Technology Settings for Tape-Out

We use TSMC 65nm process to implement a 2D system and the chiplets for 2.5D integration. For the 2.5D integration technology, we refer to TSMC InFO [3], which is one of the most advanced FOWLP technology available today. However, there is no PDK currently available for InFO technology that can be used to design and study 2.5D packages in academia. Therefore, we modified the top routing layers with updated design constraints in TSMC 65nm PDK to mimic the attributes of TSMC InFO package routing layers. Table I shows our settings for the PDK top routing layers, which are used as the package redistribution layers (RDL).

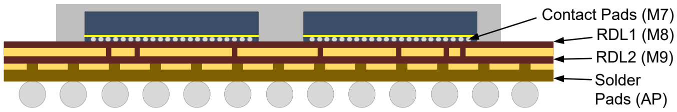


Fig. 3. Our modified 65nm package redistribution layer stack

TABLE I
Technology Parameters of Our Modified 65nm Layer Stack

Layer	Purpose	Width	Spacing
M1-M6	Chip Internal Routing	original	original
M7	Contact Pads	5 µm	5 µm
M8	RDL1	5 µm	5 µm
M9	RDL2	5 µm	5 µm
AP	Solder Pads	original	original

To reduce the I/O pad overhead and satisfy the minimum chip area requirement, we perform a shared-block tape-out where the separately designed 2D and 2.5D systems are taped-out in a single die with shared I/O pads. Fig. 2(b) illustrates our shared-block tape-out plan. The two microcontrollers have their own independent I/O sub-systems. We design an I/O multiplexing module that receives the I/O signals from both systems and bridge any one of them with the external world. The two systems also share the Power Distribution Network (PDN) of the die. This shared-die shared-I/O design technique can be used to design a chip containing multiple small sub-designs for comparative study among them. This will reduce the tape-out cost and also make the measurement results independent of process variations.

We implement the aforementioned system using the standard cells and memory compilers from ARM for the TSMC 65nm technology. For the physical design, we use M1 to M6 to perform routing of the 2D chip and the internal routing of the 2.5D chiplets. A holistic design flow requires a unified PDK that can handle both chiplet and package in the same design environment. As depicted in Fig. 3, M7 of the original technology corresponds to the contact pads of the chiplets. M8 is modified to mimic the first package routing layer (RDL1) that connects to the chiplet contact pads. M9 layer corresponds to the second package routing layer (RDL2). The solder pads will be placed on another layer, which is next to RDL2 and corresponds to the AP layer of the original technology.

C. Overall CAD Flow

We follow a holistic tool flow, which is designed to extract the maximum performance out of a 2.5D system. Our entire design flow is illustrated in Fig. 4. We start with the Register Transfer Level (RTL) netlist of the system and perform synthesis in a standard commercial synthesis tool. The synthesis process is the same as in the traditional 2D flow. To break down the system into chiplets, we partition the synthesized netlist using a suitable partitioning scheme. In a holistic flow, the partitioner needs to take into account the impact of RDL layers while exploring the solutions. Next, we perform the top-level planning of the system. At this step, following a holistic co-design strategy described in previous work [7] we prepare the package floorplan, chiplet pin configurations, RDL routing, and initial floorplans of the chiplets. Before package routing, we configured chiplets pin locations to minimize the impact of long package wires on system performance. The package routing, chiplet footprints, and package floorplan are determined and exported for later use in the package design tool. Based on the pin-configuration, we prepare some initial floorplans of the chiplets that is used in their physical design steps.

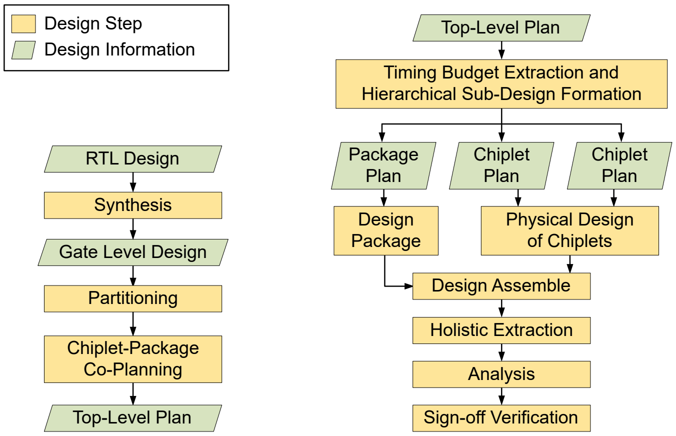


Fig. 4. Overall CAD flow for 2.5D system design

After preparing the top-level plan, we load the partitioned netlist in a chip design environment that supports hierarchical design flow. The design environment is set up with the modified PDK that supports both chiplet and package routing layers. We implement the package floorplan, chiplet pin-configurations, and the initial chiplet floorplans as determined in the top-level planning step. Next, we perform trial routing for timing budget extraction of the chiplets and the package wires. After this step, the chiplets and the package plans are separated into hierarchical sub-designs that are implemented independently in their own design environments. The package design is implemented according to the top-level plan with the package wires routed. The physical design of chiplets is performed following the traditional 2D chip design flow. After Design Rule Check (DRC), we assemble the chiplets and the package with the unified PDK. We perform another holistic DRC to check for violations in the assembled design. Next, we perform holistic parasitic extraction of the design. With the entire design assembled in the same environment, this extraction process can capture the interactions between the chiplet and package routing wires. Then we perform analysis and verification of the entire system with the extracted parasitics.

III. Design Methodology

In the design process, we start with the RTL netlist of the system mentioned in Section II-A. We use Synopsys Design Compiler to synthesize the RTL netlist for TSMC 65nm technology. We set up the design constraints and run through the standard synthesis flow to generate the gate-level netlist at the target technology. The next step is to partition this gate-level netlist into chiplets for 2.5D system design.

A. Partitioning

The first step of a 2.5D design flow is to partition the system into chiplets. Even in the 2D chip design flow, the entire design is partitioned into several sub-design modules for parallel implementation. However, there are significant differences in the partitioning of a 2.5D system into chiplets and a 2D system into sub-design modules. The modules of a 2D system are interconnected through on-chip wires, which are similar to the wires used in their internal routing. However, 2.5D chiplets are interconnected using package wires, which are very different from the on-chip routing wires. The package wires have large parasitic capacitances, significant inductances, different dielectric parameters, different coupling, etc. Moreover, optimization techniques like buffer/repeater insertion are not possible in the package wires. As a result, while partitioning a system into chiplets for 2.5D implementation, the partitioner needs to consider the impacts of package wires on system performance.

TABLE II
2.5D Chiplet Partitioning Results

Parameter	Core-Chiplet	Memory Chiplet
Frequency	100 MHz	100 MHz
Power	1.844 mW	0.539 mW
Logic Cell #	20,206	0
Macros	6	2
Area (μm^2)	179,655	72,826
Area Balance	71.16%	28.84%
Pin Count	141	101

To be implemented as a 2.5D system, we partition the microcontroller system into two chiplets. We explore several partitioning algorithms/schemes to study the impact of package wires in the partition stage. We study area-balanced partition using hMetis [8] and FLARE [9] algorithms, logic-vs-memory partitions, and Architecture-Aware partitions. In the Architecture-Aware partition, where we utilize our knowledge of the system architecture to create the partitions. Among all these schemes, though we achieve the best performance in the logic-vs-memory scheme, it produces pin counts of the chiplets that cannot be accommodated within reasonable chiplet areas. In the rest of the design, we use the architecture-aware partition results because it has reasonable chiplet pin counts and enables the Drop-in design approach. In this approach, a smaller system can be extended simply by adding some additional chiplets to the system. In Section III-G we discuss how this design approach can be utilized in 2.5D integration technology to design low-power flavors of a system without changing the design flow.

In the implemented partition scheme, we name the chiplet with all the logic cells as Core-Chiplet and the other chiplet with only two memory macros as Memory Chiplet. The memory macros in Core-Chiplet are addressed by the lower 8KB of the memory address range, and the memory macros in Memory Chiplet are addressed by the upper 8KB of the memory address range. As a result, Core-Chiplet can operate independent of Memory Chiplet with reduced memory. Table II shows the parameters of these chiplets, which are used in our 2.5D system design.

B. Chiplet and Package Co-Planning

The overall system performance of a 2.5D system is highly dependent on package-wire routing. Unplanned pin configurations of the chiplets can cause package routing issues like congestion, detour, long wires, uneven bus delays, or even unroutable pins. Package routing can be simplified a lot if the pins of the chiplets are arranged keeping their relative position on the package into consideration. At this step, we determine the pin configurations of the chiplets, package floorplan, and routing together in a way to minimize all the aforementioned package routing issues. To implement this Chiplet and Package Co-Planning, we follow the strategy mentioned in previous work [7] on holistic design flow. To automate this co-planning process, we write an RDL planning program that implements this strategy.

At first, we determine the dimensions, pin size, and pin pitch of both chiplets. As mentioned in Table I, the width and spacing of RDL1 and RDL2 wires are both 5µm. For this design, we use a pin pitch of 30µm, which allows three routing tracks in between any two consecutive pins. The width and height of the Core-Chiplet are determined to be 520µm and 475µm, respectively. The width and height of the Memory Chiplet are determined to be 415µm and 230µm, respectively. Next, we load the partitioned netlist, technology settings, and the chiplet dimensions and pin information in the RDL

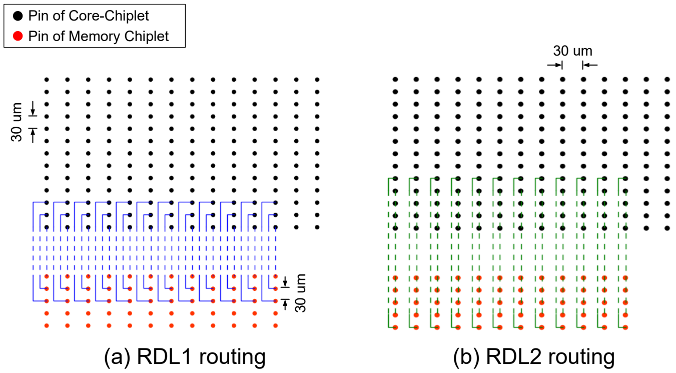


Fig. 5. 2.5D package routing generated by the RDL planner tool

planning tool. Based on the algorithm presented in the previous work [7], the RDL planning tool performs track assignment to the chiplet pins. After track assignement, it determines the relative position of the chiplets on the package. Afterward, it performs routing and signal assignment of the chiplet pins. Fig. 2(c) shows the floorplan of the package and Fig. 5 shows the package routing generated by the RDL planner tool. The black pin array on the top of Fig. 5(a) and (b) represents the Core-Chiplet pins, and the red pin array at the bottom represents the Memory Chiplet pins. As described in the track assignment algorithm of [7], the pins are routed on RDL1 first and then on RDL2. As mentioned earlier, the pin pitch allows three routing tracks in between two consecutive pins. Therefore, three rows of RDL1 pins are connected between the chiplets, leaving only two pin rows of Memory Chiplet to be routed on RDL2.

C. Hierarchical Sub-Design Formation

We set up the design environment with partitioned netlists and the modified PDK that can accommodate both chiplet and package routing layers. The chiplet partitions appear as 2D modules in the design environment. The purpose of this step is to implement the top-level plan generated at the chiplet-package co-planning step. We resize the modules and arrange them according to the floorplan generated by the RDL planning tool. Next, we specify the pin configurations of the chiplets according to the top-level plan. We also prepare the initial floorplans of the chiplets. Next, we perform a global cell placement and trial routing to estimate the timing budget of each part of the design. Before running the trial routing tool, we specify routing blockages around the chiplet partitions so that the router uses only the RDL layers to connect the chiplet pins. Finally, we extract the timing budgets of the chiplet partitions and split the whole design into hierarchical sub-designs. After this step, we have sub-designs for both chiplets and a top-level design, which corresponds to the 2.5D package.

D. Physical Design of Chiplets

After hierarchical sub-design formation, the chiplets can be implemented independently in parallel. We import a chiplet sub-design with the top-level design constraints and its initial floorplan. Then, the chiplet is implemented as a 2D chip using the traditional chip design techniques. At first, we adjust the floorplan of the chiplet to make room for the PDN. We define power and ground (PG) rings on M5-M6 layers along the chiplet boundary and block rings around the memory macros on M3-M4 layers. The power routing tool connects the internal PG mesh of the macros and the PG rails of the standard cell-rows to the block rings. In the Memory Chiplet, we use PG stripes over the SRAM macros to ensure sufficient power delivery. After standard cell placement, we perform routing on M1-M6 layers for intra-chiplet wires. Finally, we perform post-routing optimizations to fix some minor timing violations and reduce power consumption. Though in the original netlist of the memory chiplet, it only has the two SRAM blocks, in all the optimization steps, Innovus inserts some buffers and inverters to meet the timing constraints. Fig. 6 show the finished designs of both chiplets.

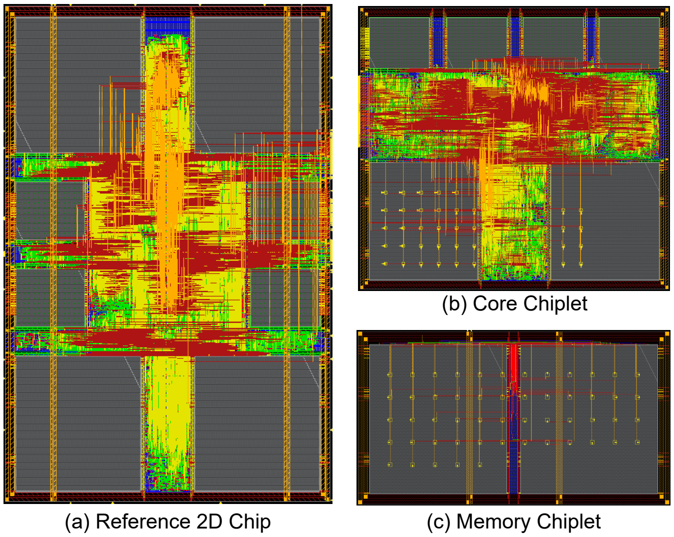


Fig. 6. Layouts of the reference 2D chip and the chiplets for 2.5D integration

E. 2.5D Package Design

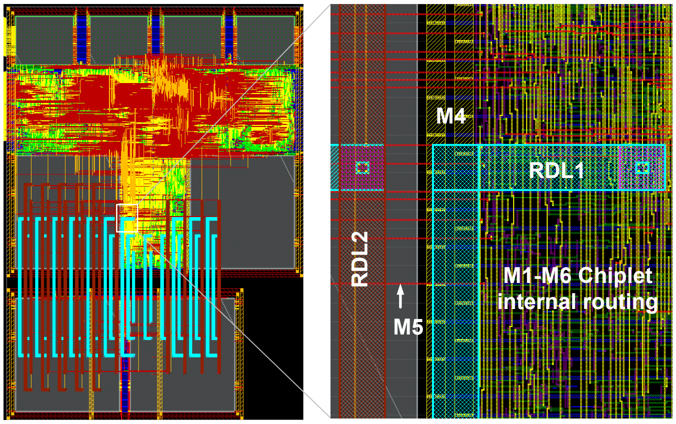
The top-level sub-design prepared in the hierarchical sub-design formation step corresponds to the 2.5D package plan. When loaded in the design environment, the chiplets appear as 2D macros. The floorplan is already fixed at the top-level planning step. Because of the differences in the chip and package routing techniques, chip routing tools cannot generate good routing for package layers. The RDL planner tool generates a routing script that can be used to perform package routing according to the top-level plan. We use that script to route the chiplet pins on RDL layers and then modify some of the routes as necessary. When the chiplet designs are complete, we can extract their interface timing models, which can be used to verify and further optimize the package design.

F. Design Assemble and Holistic Extraction

When chiplet designs are complete, we perform sign-off verifications on each individual chiplet. The package design is also verified separately to remove any potential DRC violations. We again set up the design environment with the unified PDK for the design assemble step. At this step, we assemble the DRC clean chiplets and package designs for holistic extraction and analysis. We assemble the designs in the unified design environment and then export the necessary files for analysis. In Table III, we present the holistic extraction results. As observed from the table, the interactions among the routing layers across the chiplets and the package have been captured. A traditional design and extraction method can only calculate the coupling presented in the second and the fourth quadrants of the table. Only a holistic extraction method can produce the results in the first and the third quadrants.

G. Low Power Drop-in System

In 2.5D integration, it is possible to design a system in a way so that even if one or more chiplets are not included in the package rest



(a) Assembled design (b) Zoomed-in view

Fig. 7. Finished 2.5D Design: (a) Innovus view after Design Assemble (b) Zoomed-in view showing package and chiplet routing

of the system is still functional with fewer capabilities. This reduced system can be extended to its full potential simply by dropping the previously missing chiplets in the package. We call this technique the “Drop-in” design approach. Using this approach, it is possible to design different flavors of the same 2.5D system at the package level without any additional design effort. As discussed in Section III-A, chiplets are partitioned in a way so that the Core-Chiplet can operate independent of the Memory Chiplet. For this design, after chiplet fabrication and testing, we can simply avoid a memory chiplet in the package, which will give us the same micro-controller system with 8KB of memory. In our design case, this system achieved 125 MHz maximum operating frequency and a lower power at its design frequency. As the memory chiplet is not included in the package, the overall system is cheaper. This system can be a low-cost and low-power/high-performance solution for the applications where 8KB memory is sufficient. For memory-intensive applications, the complete system with 16KB of memory is almost readily available. We just need to include both chiplets in the package. This is how the Drop-in approach can be utilized to design different flavors of a large 2.5D system.

IV. Reference 2D and Die-Level Designs

A. Reference 2D Design

To design the 2D system, we start with the netlist generated by the synthesis tool before partitioning. Shown in Fig. 6(a), The width and height of the entire floorplan are 475µm and 725µm, respectively. We place the the ROM macros in the middle region of the floorplan area and place the SRAM macros at the corners keeping offset for PG rings. We design the PDN with a PG ring around the core area on M5-M6 layers and PG block rings on M3-M4 layers. We insert PG stripes on M6 running over the macros to ensure sufficient power supply to them. After standard cell placement, we perform the timing design steps that include clock tree synthesis and timing optimizations. Finally, we route the design using six metal layers and perform post-routing optimizations. Fig. 6(a) illustrates the finished design of the 2D system.

B. Combined Die-Level Design

After finishing the 2D and 2.5D system designs, we perform DRC to ensure both of the designs are violation free. We extract their interface timing models, and layout abstracts exposing the PG rings and stripes to be used in the Die-level design. We define PG rings in between the I/O ring and the core area of the die and then connect the PG pin pads with the rings. Both systems have PG rings around their cores on M5-M6 layers. We use PG stripes on M7 to connect the PG rings and stripes of the systems with the PG ring of the Die-level design. Fig. 9(a) illustrates the Die-level design. Then we perform cell placement, CTS, and routing at Die-level, which routes the 2D/2.5D system pins to the I/O multiplexing module. And the I/O multiplexing module is routed to the I/O pads of the die. After finishing the Die-level design, we export the GDS file combining with the GDS of the system designs and perform sign-off verifications.

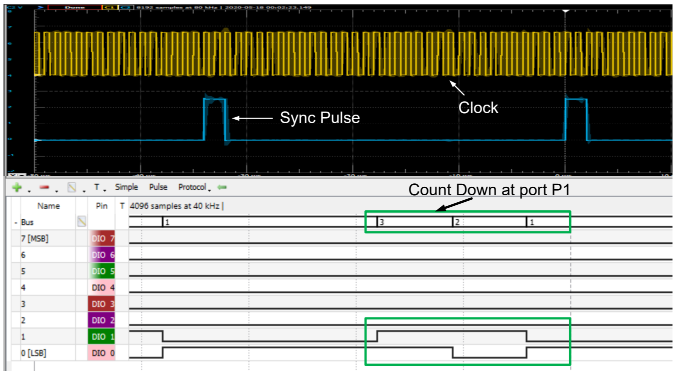


Fig. 8. Chip testing waveforms from logic analyzer

TABLE III
Holistic Capacitance (in fF) Extraction Results

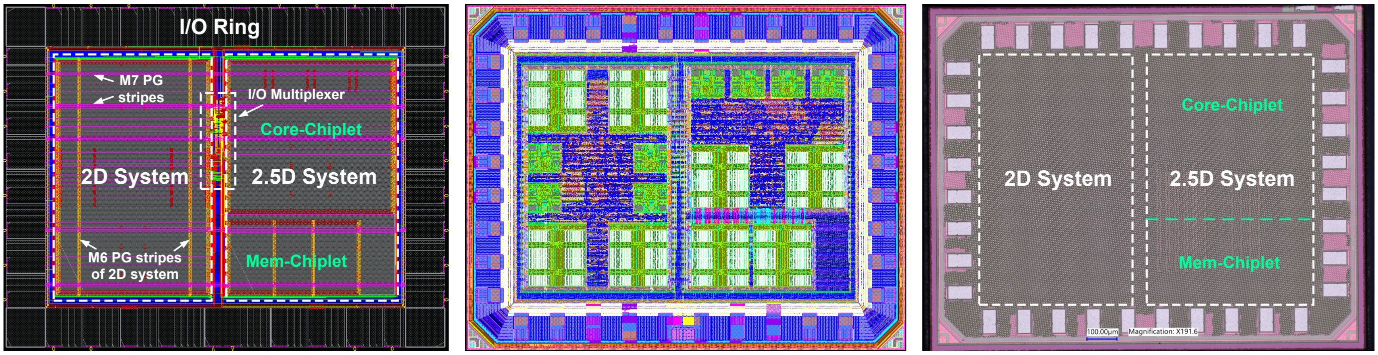
	M1-M3	M4	M5	M6	Cont. Pad	RDL1	RDL2
M1-M3	7505.6	2494.7	1389.3	38.0	0.3	13.3	0.8
M4	2494.7	2445.3	648.8	150.7	1.5	12.8	0.4
M5	1389.3	648.8	2756.7	90.0	1.3	40.8	4.9
M6	38.0	150.7	90.0	190.6	8.6	31.1	6.8
Cont. Pad	0.3	1.5	1.3	8.6	0.0	0.6	0.1
RDL1	13.3	12.8	40.8	31.1	0.6	10.8	146.2
RDL2	0.8	0.4	4.9	6.8	0.1	146.2	33.8
Ground Capacitance							
Metal Layer	M1-M3	M4	M5	M6	Cont. Pad	RDL1	RDL2
Capacitance	20784.7	6828.6	4993.4	1477.2	94.8	132.4	95.3

V. Analysis Results

A. Holistic Extraction Results

Table III presents the holistic extraction results obtained after assembling the chiplets and package designs at the top-level. For readability, we merged the coupling capacitances among layers M1-M3 in the table. As seen from the table, the holistic extraction method effectively captures the interactions between the chiplet and package wires. Using traditional extraction flows, one can get the results in the second quadrant (among intra-chiplet layers) and fourth quadrant (among package routing layer). Even though, our design is a small system there exists sufficient coupling between RDL1 and top chiplet layers like M4-M6. In a large system with a lot of package wires and denser chiplet routing on the top routing layers, these couplings will be severe and if ignored may cause signal integrity issues leading to total system failure.

If we observe, we can see that the coupling between RDL1 and M5 is greater than that between RDL1 and M6. The usual expectation is that the coupling should be greater between M6 and RDL1 as these two are adjacent routing layers. However, in this design case, the routing on M6 is significantly less compared to M5, which is why M5 has more coupling with RDL1 compared to M6. This kind of detailed extraction data can be utilized to effectively optimize the



(a) Die-level design in Innovus

(b) GDS for tapeout

(c) Microscopic die-shot

Fig. 9. Final design for tape-out and the fabricated die: (a) Die-level design, (b) Combined GDS for tape-out, (c) Microscopic image of the taped-out die.

TABLE IV
ANALYSIS RESULTS OF 2D CHIP AND 2.5D SYSTEM

Chip Design	2D Chip	Core Chiplet	Mem Chiplet
Standard Cells #	20,061	20,096	27
Total Wirelength (mm)	544.70	478.57	12.96
Die Size (µm×µm)	475×725	520×475	415×230
System Frequency	125 MHz	100 MHz	
Chip Power	7.0 mW	5.12 mW	0.718 mW

routing and improve system reliability. Coupling numbers on Cont. Pad layer is negligible as there is no routing on this layer. There exists sufficient coupling between the RDL1 and RDL2 because many of the wire traces on these layers exactly overlap with each other.

B. Timing and Power Analysis Results

Table IV presents the timing and power analysis results. The standard cell count of the Core-Chiplet and the 2D chip are comparable. As mentioned in Section III-D, the optimization steps of chiplet design insert some buffers/inverters which is why the Mem-Chiplet has those 27 standard cells apart from the SRAM macros. The total wire-length at the chiplet level is shorter in the 2.5D system compared to the 2D chip. This result is consistent with a previous study [10] which reveals the reduction of total chip wire-length in 2.5D design. The overall performance of the 2.5D system is worse than the 2D system because of the package wire overhead. The maximum system frequency we could achieve is 125MHz for the 2D system and 100MHz for the 2.5D system; the performance gap being 20% w.r.t the 2D system. This result is also consistent with the previous study [7], where the 2D system achieved an operating frequency of 333MHz while the 2.5D system could only achieve 245MHz, a 26% performance gap w.r.t the 2D system. The power numbers in the table correspond to the maximum system frequency. The lower power of the 2.5D chiplets is because of the reduced system frequency.

C. Chip Testing and Validation

The fabricated chip is tested and validated using test vectors generated by a logic analyzer. Fig. 8 shows one of the testing waveforms. In this test, the micro-controller reads a top value from a GPIO port and performs count-down on another port. After each countdown is finished, it sends a synchronization pulse to the logic analyzer. The fig. 8 shows the clock signal, synchronization pulse, and the count-down on a digital bus connected to the GPIO port.

VI. CONCLUSION

In this paper, we present the entire design methodology of a 2.5D system in a commercial chip design technology, starting with its RTL netlist to the sign-off verification of the final GDS. We follow a holistic design, optimization, and analysis flow to implement an ARM Cortex-M0 processor-based micro-controller system in TSMC 65nm PDK to be integrated using TSMC InFO technology. The design techniques presented for shared-block tape-out and application of Drop-in design approach can be used for low-cost, low-power, and high performance applications. This design case study validates the effectiveness of the holistic design and analysis flow for 2.5D system designs in real-world technologies. From our extraction results, we can conclude that the holistic extraction process effectively captures the interactions between different components of a 2.5D system across chiplet and package layers. Our timing and power analysis results reveal that the holistic analysis approach takes into account the impacts of package overhead on system performance, which is essential for reliable system design.

REFERENCES

[1] M. A. Karim, P. D. Franzon, and A. Kumar, “Power comparison of 2D, 3D and 2.5D interconnect solutions and power optimization of interposer interconnects,” in *IEEE Electronic Components and Technology Conference*, May 2013, pp. 860–866.

[2] J. U. Knickerbocker, P. S. Andry, E. Colgan *et al.*, “2.5D and 3D technology challenges and test vehicle demonstrations,” in *IEEE Electronic Components and Technology Conference*, May 2012, pp. 1068–1076.

[3] C. Tseng, C. Liu, C. Wu, and D. Yu, “InFO (Wafer Level Integrated Fan-Out) Technology,” in *IEEE Electronic Components and Technology Conference*, May 2016, pp. 1–6.

[4] J. Kim, G. Murali, H. Park *et al.*, “Architecture, Chip, and Package Co-design Flow for 2.5D IC Design Enabling Heterogeneous IP Reuse,” in *Design Automation Conference*, 2019, pp. 178:1–178:6.

[5] W. Liu, Min-Sheng Chang, and T. Wang, “Floorplanning and signal assignment for silicon interposer-based 3D ICs,” in *Design Automation Conference*, June 2014, pp. 1–6.

[6] Jia-Wei Fang and Yao-Wen Chang, “Area-I/O flip-chip routing for chip-package co-design,” in *International Conference on Computer-Aided Design*, Nov 2008, pp. 518–522.

[7] M. A. Kabir and Y. Peng, “Chiplet-Package Co-Design For 2.5D Systems Using Standard ASIC CAD Tools,” in *Asia and South Pacific Design Automation Conference*, Jan. 2020.

[8] G. Karypis and V. Kumar, “Multilevel k-way Hypergraph Partitioning,” *VLSI Design*, vol. 11, no. 3, pp. 285–300, 2000.

[9] Jason, J. Cong, S. K. Lim, and C. Wu, “Performance Driven Multi-level and Multiway Partitioning with Retiming,” in *Design Automation Conference*, 2000, pp. 274–279.

[10] Y. Deng and W. P. Maly, “Interconnect Characteristics of 2.5-D System Integration Scheme,” in *International Symposium on Physical Design*, 2001, p. 171–175.