Architecture, Chip, and Package Co-design Flow for 2.5D IC Design Enabling Heterogeneous IP Reuse

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
School of ECE, Georgia Institute of Technology, Atlanta, GA
jinwookim@gateh.edu,limsk@ece.gatech.edu

ABSTRACT

A new trend in complex SoC design is chiplet-based IP reuse using 2.5D integration. In this paper we present a highly-integrated design flow that encompasses architecture, circuit, and package to build and simulate heterogeneous 2.5D designs. We chipletize each IP by adding logical protocol translators and physical interface modules. These chiplets are placed/routed on a silicon interposer next. Our package models are then used to calculate PPA and signal/power integrity of the overall system. Our design space exploration study using our tool flow shows that 2.5D integration incurs 2.1x PPA overhead compared with 2D SoC counterpart.

1 INTRODUCTION

Interposer-based 2.5D IC design allows block-level heterogeneous integration, where all functional circuit blocks are designed separately under different environments and integrated, rather than designed and fabricated monolithically into a single SoC. Figure 1 shows an interposer-based 2.5D IC design and its cross-section view. The 2.5D IC has an interposer on top of the package. The functional blocks, named *chiplets*, are mounted on the interposer.

Connections between chiplets are made through the interposer to achieve high speed and throughput. With this architecture, each intellectual property (IP) can be independently designed into a chiplet under its most suitable technology node and assembled into the SoC. This design approach enables SoC designers to simply choose appropriate off-the-shelf chiplets and heterogeneously integrate them into the target SoC, which drastically reduces design time and complexity by re-utilizing pre-designed chiplets as plug-and-play modules. In addition, system update is greatly simplified because it only needs to swap out chiplets that are necessary, instead of redesigning the entire SoC from scratch.

Before applying 2.5D technology to real designs, a thorough analysis of trade-offs between monolithic 2D SoC and interposer-based 2.5D design should be preceded. There are existing studies on 2.5D IC design focused on the design methodology or utility point of view such as analysis of design cost aspect[5] and bump assignment algorithm for 2.5D interposer design[4], however, there is no analysis of overheads in terms of actual power, performance and area of 2.5D design.

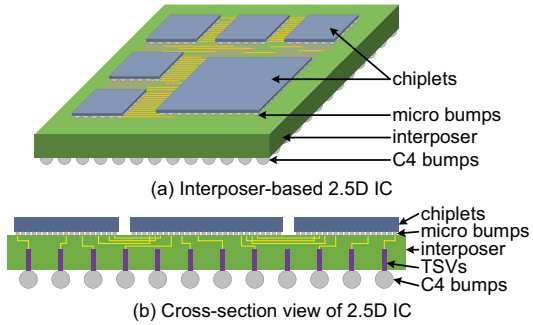

Figure 1: 2.5D chiplet integration with an interposer.

In this paper, we first present our new RISC-V-based 64-core architecture named ROCKET-64 for chiplet integration. Next, we present a vertically-integrated EDA flow for chiplet creation and integration, which covers the design phases of architecture, circuit and package. Next, we present a new logical protocol called Hybrid-Link to reduce overheads of 2.5D IC design. Moreover, we provide PPA data of 2.5D IC design and compare with its monolithic 2D counterpart for quantitative comparison of 2D and 2.5D designs. We chose a target design of Rocket-64 with Network-on-Chip (NoC) configuration to show stepwise explanation of the overall flow.

We claim the following contributions: (1) Our new 64-core RISC-V architecture is scalable and appropriate for chiplet integration. (2) We generate interposer-based 2.5D design including interposer routing and the layout of each chiplets by using commercial tools; (3) We propose a new logical protocol that is well fitted for 2.5D IC design; (4) We analyze PPA of 2.5D ICs using different interposer technologies to show overhead difference; (5) We analyze PPA of interposer-based 2.5D ICs and compare the result with monolithic 2D IC to investigate overheads of 2.5D design. To our best knowledge, this is the first work to fully quantify the design gap between 2D and 2.5D designs in terms of PPA using GDS layouts and sign-off simulations.

2 ARCHITECTURE AND DESIGN SETTING

2.1 Proposed 64-Core Architecture

We create a new 64-core architecture named ROCKET-64 based on RISC-V Rocketcore[2] as our benchmark design as shown in Figure

DAC '19, June 2–6, 2019, Las Vegas, NV, USA

ACM ISBN 978-1-4503-6725-7/19/06…$15.00
https://doi.org/10.1145/3316781.3317775

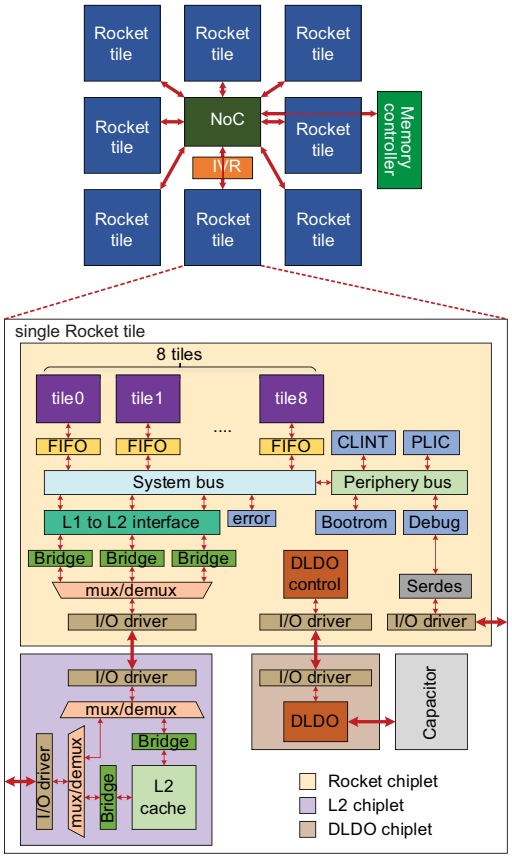

Figure 2: Our proposed 64-core architecture for chipletization and 2.5D integration.

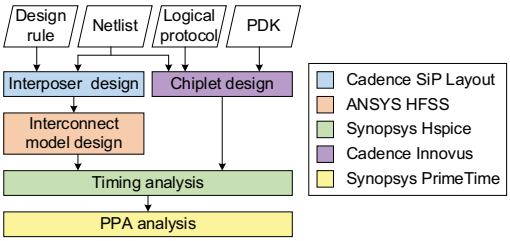

Figure 3: Our EDA flow using commercial tools.

2. ROCKET-64 consists of 8 Rocket tiles, a centralized network-on-chip(NoC) as an arbiter, a 4-channel memory controller to access external DRAMs and an integrated voltage regulator(IVR) as a power management module. Each Rocket tile consists of octa-core RocketCore, L2 cache and digital low-dropout(DLDO). Each module contains I/O drivers only for 2.5D interposer design. For monolithic 2D IC design, we map all modules without I/O drivers and power management modules such as IVR and DLDO on a single chip.

The centralized NoC consists of 12 routers interconnected in a 4x3 mesh topology. Links from each Rocket tile and memory controller are connected to the external ports of routers. Each router has five ports(N,E,S,W, and external) with four virtual channels at each port. The router implementation is based on a one-cycle pipeline design, which consumes one cycle in the router logic and additional one cycle for link traversal, used in OpenSMART[1]. We implement matrix arbiters that provides fairness for input virtual channel arbitration and switch allocation to prevent starving at any core.

2.2 Overall EDA Flow

Figure 3 shows the overall flow of our chiplet creation and integration. Our EDA flow takes interposer PDK, design netlist, logical protocol and chip PDK as initial input, generates the layouts of interposer and each chiplet, and performs timing and PPA analysis with existing commercial tools.

In interposer design step, we generate the layout of interposer including the footprint of each chiplet and the routing information between chiplets. We extract the wirelength distribution of interposer wires for timing analysis. The interposer channel with corresponding dimensions is characterized using a full-wave EM solver, Ansys HFSS. Next, S-Parameters defining the impedance and coupling profile are extracted. This is then converted to SPICE models using the broadband SPICE generator of Keysight ADS.

With selected I/O drivers, we generate the layouts of chiplets in chiplet design step. We used Cadence Innovus to perform place-and-route of chiplets with usual 2D design method. We analyze PPA of interposer-based 2.5D design in the final step using Synopsys PrimeTime. Full-chip timing and power analysis for individual chiplets is straightforward and done with Synopsis PrimeTime after their layouts are constructed. Once our inter-chiplet I/O drivers are built and chosen to handle the given interconnect length, we calculate their delay and power consumption using their SPICE models. We then add these values to chiplet delay and power data. Our interposer interconnects are pipelined due to the FFs used in the I/O divers, which simplifies timing calculation for the entire interposer design.

2.3 Interposer Design Rules

In past few years, as the design complexity of a single module increases, dense interposer design with fine pitch of RDLs and micro bump have been required in heterogeneous integration due to high I/Os and the increasing number of interconnections between chiplets. A representative example of satisfying these requirements is silicon interposer. Taiwan Semiconductor Manufacturing Company, Limited (TSMC) and Xilinx, Inc. have suggested Chip-on-Wafer-on-Substrate (CoWoS) technology[3] which provides minimum 0.8μm pitch RDLs and supports over 200K of micro bumps with 45μm micro bump pitch. They have demonstrated Virtex-7 2000T FPGA, which consists of four different 28nm FPGA dies and has more than 10,000 die-to-die connections, as the application of CoWoS.

The design rules for our interposer design in this paper are shown in Table 1 and Figure 4 based on TSMC CoWoS. We choose silicon interposer with 0.8μm fine pitch RDLs and 40μm-pitch micro bumps for our benchmark.

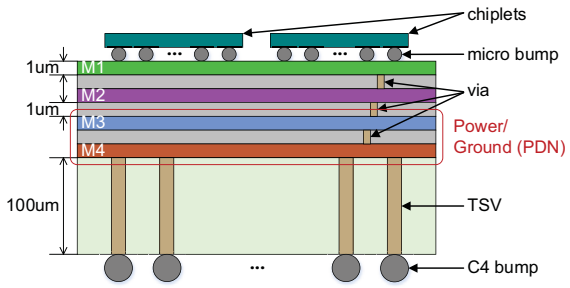

Figure 4: Vertical stack-up of our interposer-based 2.5D IC.

Table 1: Design rules for our silicon interposer (based on a commercial 65nm technology).

Metal layer#	4
Metal thickness	$1\mu m$
Dielectric thickness	$1\mu m$
Min. line width/spacing	$0.4\mu m/0.4\mu m$
Via size	$0.7\mu m$
Through Via size/depth	$10\mu m/100\mu m$
Die-to-die spacing	$100\mu m$
micro-bump pitch	$40\mu m$
C4 bump pitch	$180\mu m$
PDN width/spacing	$40\mu m/90\mu m$

3 CHIPLETIZATION RESULTS

For the interposer-based 2.5D IC design, we first divide a single SoC into multiple functional blocks. We use the natural IP boundaries - core, cache, NoC, and Memory controller to create a total of 27 chiplets. Before generating chiplets from these functional blocks, two design features must be strongly considered: an interface protocol and I/O drivers.

3.0.1 Interface Protocol. The study of interface protocols for systems with modular IP blocks is important for easy system design, integration, and verification. On-chip IPs today use a rich set of protocols; examples include AXI used by ARM-based IPs, TileLink used by RISC-V based IPs, Avalon used by Intel/Altera, and so on. Unfortunately, these cannot be ported directly to chiplets as they have hundreds of I/O signals to support address, data, and commands for multiple individual channels. Wires are relatively cheap on-chip since the area of an IP block is dominated by logic, not I/O, since the minimum wire pitch in modern technology nodes is $0.09\mu m$. For a chiplet, however, C4 bumps to connect to the interposer are much wider such as $180\mu m$, and can potentially completely dominate the area of a chiplet, as we quantify later in Section 5.1. Moreover, chiplet-to-chiplet interconnections are generated through the interposer layer which has larger dimensions and longer wire length compared to monolithic 2D design, so additional I/O drivers are necessary for each input and output to drive the signals without any loss.

In this work, we propose a new protocol called Hybrid-Link. Hybrid-Link is designed keeping three goals in mind - (i) standard protocol applicable across different chiplets, (ii) 2.5D ICs should have low number of external I/Os, (iii) different chiplets have different communication requirements. A sample flit[1] representation of common commands is shown in Figure 5. Hybrid-Link uses a default flit width of 40 bits - though this can be further reduced, at the cost of serialization. The protocol can operate in two modes - lightweight and extended. The lightweight mode is for simple point-to-point connections. In this mode, the protocol provides a few bits for command, while the rest of the bits are used by address and data. As shown in Figure 5, Lightweight mode requires only one flit for read requests and responses, and two-flits for write requests. In the extended mode, more complex transactions can be supported.

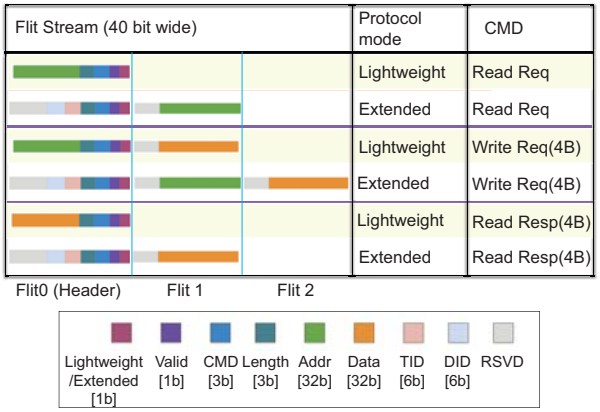

Figure 5: Flit representation of Hybrid-Link

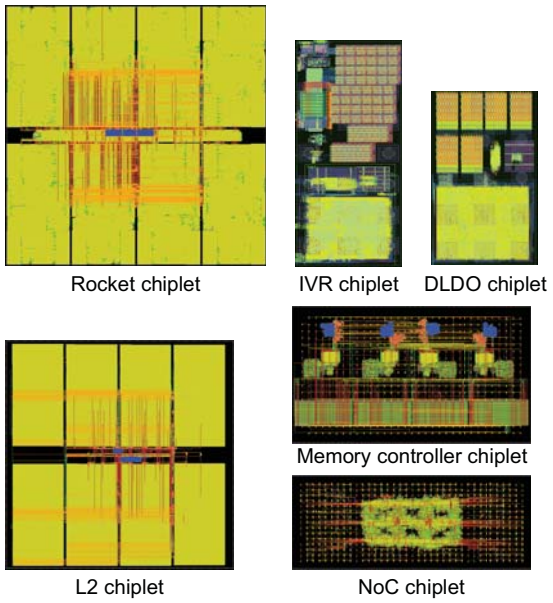

Figure 6: Commercial 28nm and 130nm physical layouts of the chiplets in our ROCKET-64 architecture (not drawn in scale). The blue part shows protocol translator/bridge logic.

[1]A flit is the number of bits of data transfer over the physical link

Table 2: Chiplet list in our benchmark design

Chiplet	I/O bump#			Signal bump#			Footprint ($\mu m \times \mu m$)	Bump array	Technology node
	Total	Signal	P/G	Internal	External	Common			
Rocket	169	65	104	53	10	2	1,600 × 1,600	13 × 13	commercial 28nm
L2	210	92	118	90	-	2	1,460 × 1,460	14 × 15	commercial 28nm
NoC	663	655	108	660	-	3	1,560 × 680	39 × 17	commercial 28nm
Memory controller	700	588	112	185	400	3	1,400 × 800	35 × 20	commercial 28nm
IVR	252	12	240	-	9	3	480 × 1,200	12 × 21	commercial 130nm
DLDO	204	12	192	7	-	5	480 × 800	12 × 17	commercial 130nm
Passive L	-	-	-	-	-	-	1,600 × 3,400	-	Embedded L
Passive C	-	-	-	-	-	-	2,000 × 3,600	-	SMD type

The extended mode provides fields for destination and transaction identifiers (DID and TID) to support AXI transactions. The extended mode also supports multiple Virtual Channels to allow better buffer utilization and provide deadlock freedom. Additional communication features may be added to the RSVD field. There is one protocol bit in the header flit that determines whether the packet will be read in lightweight or extended mode. A Finite-State Machine will determine how to parse the following flits fields based on protocol bit. Both protocol modes allow variable packet lengths and common commands. ROCKET-64 uses the extended mode for the Rocket, L2, NoC chiplets and memory controller chiplets, and the lightweight mode for the DLDO chiplets.

3.0.2 Bridges and I/O Drivers. To translate common interface protocols such as AXI4 and TileLink to Hybrid-Link, we implemented FIFO queues and bridge FSMs. The FIFO queues are used to store common flit fields across the two prototypes, and the FSMs are used to remap the field representation to Hybrid-Link and vice versa. The FSMs are also responsible for flit arbitration and ready signals handling. The bridge consumes negligible area compared to the size of the rocket chiplet.

3.0.3 Chiplet Layouts. We perform chiplet place-and-route using Cadence Innovus as the physical design tool with selected protocol translator and I/O driver. We first run the pin placement based on the micro bump assignment. As the chiplet is mounted on an interposer with micro bumps, each I/O pin is placed on the position of its micro bump. With well-defined pin placement, the tool places I/O drivers on the proper positions to meet the timing design constraint. The chiplet list of our benchmark design and their GDS layouts with $1GHz$ target frequency are shown in Table 2 and Figure 6.

4 INTERPOSER-BASED 2.5D IC DESIGN

4.1 Interposer Layout Results

The process of designing the interposer consists of bump assignment according to the floorplan and placement of chiplet dies and interposer routing. Since each chiplet is connected to the interposer through the bumps, the bump assignment is an important factor in determining the length of the signal interconnection. We chose a regular bump assignment which is placing signal bumps in the center of die and power bumps at periphery. With bump assignments, we generate die data, which contains bump coordinate and type,

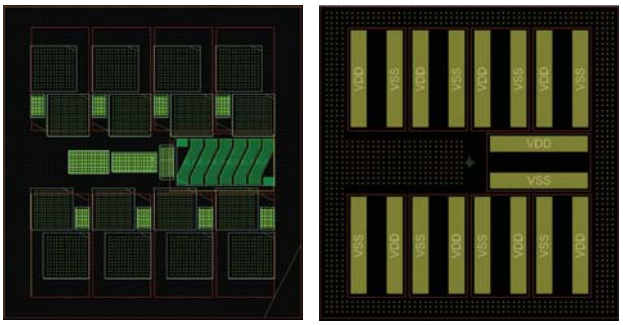

Figure 7: Floorplan of our silicon interposer: top and bottom side

from verilog netlists as an input for floorplanning and interposer routing.

GUI-based floorplanning and interposer routing have been done by using Cadence SiP Layout. We first set up technology file including metal stack and via structures which provides physical and electrical information. By importing die data into the tool, we place all the dies of chiplets on the interposer for the routing step. In our benchmark design, we placed passive capacitors at the bottom of the interposer to reduce entire footprint as shown in Figure 7. Automatic Router provided by Cadence SiP Layout, which performs Manhattan routing same as on-chip routing, is used for over 1,000 interconnections in interposer layer.

While in the routing step, the data skew problem should be considered as an important factor. Unlike monolithic 2D ICs, the wire length of the signal between chiplets in 2.5D system can reach several millimeters in case of non-neighboring connections. Due to the distance differences between bump pairs in the single bus, each signal can arrive at its destination with different timing. Especially in the case of non-neighboring connection where source and sink chiplets are placed far apart, this problem should be highly critical in interposer routing. To avoid it, we added a design constraint, named Match Group (MG).

The new design constraint creates a new design rule that causes wire lengths or propagation delays of signals to be in the specified target distribution for signals belonging to the same group. Compared to when MG is applied to one of our benchmark design buses

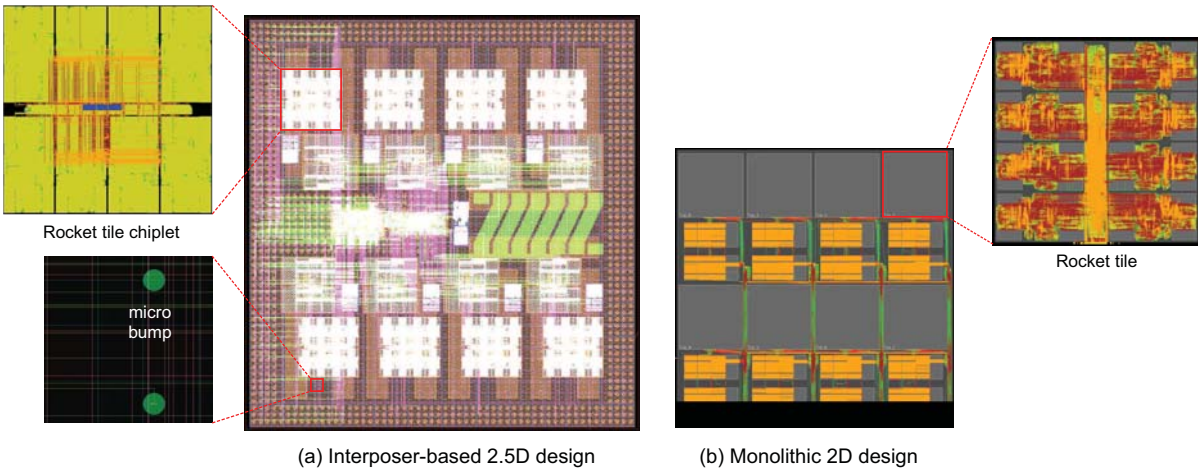

Figure 8: Interposer-based 2.5D design vs. Monolithic 2D (GDS layouts)

Table 3: 2.5D Interposer design results (see Figure8(a)).

Routed net #	1,441
Metal layers used	4
Min wirelength	$780\mu m$
Ave wirelength	$3{,}781.9\mu m$
Max wirelength	$7{,}020\mu m$
Via usage	5,968
PDN DC resistance	20.1mΩ
Area	$111.65mm^2$

and when MG is not, the wire length variation is reduced from $1400\mu m$ to $200\mu m$. We assign each bus in our design as each MG with a design constraint of $200\mu m$, so that the length of the signals in one bus is within $200\mu m$ deviation.

Our silicon interposer design results are shown in Table 3 and Figure 8. 1,441 nets are routed on the silicon interposer layer and 4 metal layers are used in order to demonstrate the 2.5D design of our benchmark including power delivery network(PDN).

4.2 Interposer Timing and Power Analysis

We considered digital inverter with full-swing signal as I/O drivers. A strong output driver is required to drive long interposer wires. Moreover, due to their large dimensions, interposer wires have significant inductance leading to signal reflections from both driver and receiver ends. To eliminate reflections, impedance of the final driver stage is matched to the characteristics impedance of the package wire. To reduce overheads of the I/Os, I/O driver runs at full-swing of the supply voltage. The final driver size is chosen to be x128, resulting in an output impedance of 47.4Ω.

For timing analysis, chiplet-to-chiplet communication delay and skew between all the wires in data bus as well as with the clock is measured from end-to-end. We performed the timing analysis for our design by generating a transmission line model for the interposer interconnect channel using Ansys HFSS tool. The interconnect lengths in our design varies from 500 um to 7500 um. We performed a delay analysis of all the interconnect channels in the design by incorporating the corresponding RLGC model into an HSPICE circuit. We obtained the worst case propagation delay to be 152.3ps. As our design is targeted to run at a frequency of $1GHz$, these longest propagation delays are well within the limits to meet the setup and hold times of the receiver.

In power analysis, we obtain each power of the chiplet core and the I/O drivers to estimate the total power of interposer system. Each routed net in interposer layer which is connected between two I/O drivers has a different wire length. However, this difference is not reflected in logic synthesis tool, so the power estimation in our EDA flow reflecting the wire length correctly is as follows:

$$P_{2.5D} = P_{CORE} + P_{I/O} \quad (1)$$

where, $P_{2.5D}$ is total power of 2.5D design, P_{CORE} is the power of chiplet core, and $P_{I/O}$ is the power of I/O drivers. For $P_{I/O}$, we run HSPICE simulation of a testbench with self-generated SPICE models. The power estimation of each chiplet core is done by Synopsys PrimeTime.

4.3 Interposer Signal and Power Integrity

We performed the signal integrity analysis and generated the eye diagrams by converting the RLGC matrices of the transmission line model into corresponding S-parameters and feeding them into Keysight ADS. Our routing involves the use of complex interconnect structures, as they help in reducing the cross talk compared to the simple structures. We focus on a complex interconnect channel for crosstalk analysis. The characteristics of the eye diagrams are as follows: eye width is $0.985ns$, and eye height is $0.430V$. These results are obtained based on simulations done at a data rate of $1Gbps$, I/O driver impedance of 50Ω and receiver chiplet pad parasitics of $2pF$ capacitance.

The power integrity of our design is ensured with the use of a global IVR chiplet and 8 local DLDO voltage regulators chiplets and distributing power through a mesh type PDN. Our IVR has a dynamic voltage scaling speed of $69mV/\mu s$ and has an efficiency of 89.7%. In addition to the carefully designed PDN to maintain power integrity across the interposer, each chiplet has a minimum of 100

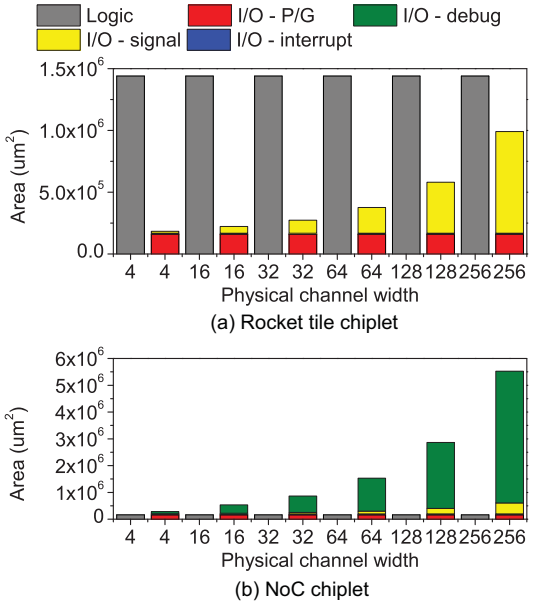

Figure 9: Relationship between the size of chiplet vs. I/O counts.

power bumps placed on the chiplet periphery to ensure power integrity across the chiplet.

5 DESIGN SPACE EXPLORATION RESULTS

5.1 Interface Protocol Comparison

The relationship between chiplet area and I/O count is shown in Figure 9 with examples of chiplets in our benchmark design. In the case of rocket chiplet, the logic area overshadows the physical channel overhead. This means that the I/Os are not contributing to additional area. However, in the case of NoC chiplet, there is huge C4 bump area cost even with very narrow physical channel width. This is because NoC contain numerous Hybrid-Link IO ports along with a much smaller logic overhead than rocket chiplet. A narrow interface protocol like Hybrid-Lite for 2.5D ICs is necessary to keep the chiplet area reasonable, and not let I/O bump area dominate. Moreover, Hybrid-Link's 40b interface can help design smaller chiplets without incurring an area penalty due to I/O.

5.2 Monolithic 2D vs. Interposer-based 2.5D

In monolithic 2D design, we perform hierarchical design so that it has the same structure as interposer based 2.5D design except power management IPs. We used TSMC CLN28HPC as the technology node and Cadence Innovus as the physical design tool. The layout and PPA result of monolithic 2D design with the target frequency as $1GHz$ is shown in Figure 8 and Table 4. The total power is $8.948W$ and the area of design including 8 RocketCores is $53.14mm^2$.

In 2.5D design, the total power consumption has increased by 0.8% compared to 2D design due to I/O drivers and translator which are added for chiplet-to-chiplet communication. However, the overall power gap is not too significant: $8.948W$ vs. $9.023W$. A main reason is that in 2D, the number of channels in the NoC module is higher than 2.5D, which causes the NoC module in 2D to consume more power than the NoC chiplet in our 2.5D design. This is because we do not use a package-based protocol in 2D, making it necessary to increase the number of channels to handle additional traffic. In terms of area, interposer-based 2.5D design has increased by 2.5x compared to monolithic 2D design. The main reason for the increase is the addition of power management modules including passive L and C since logic synthesis and P&R flow optimizes the logical area of chiplet.

Table 4: The design comparison between monolithic 2D and interposer-based 2.5D design

	2D	2.5D Design
Frequency	1.0GHz	1.0GHz
Min wirelength	$0.3\mu m$	$780\mu m$
Avg wirelength	$222.4\mu m$	$3{,}781.9\mu m$
Max wirelength	$1435.1\mu m$	$7{,}020\mu m$
Cell #	7,887,365	7,979,736
Total power	$8.948W$	$9.023W$
Logic power	$8.948W$	$8.703W$
I/O power	-	$0.320W$
Area	$53.14mm^2$	$111.65mm^2$
Footprint	$7.29mm \times 7.29mm$	$10.30mm \times 10.84mm$

Since the interconnections between chiplets are implemented via interposer layer, the average length of connection in 2.5D design is increased by 17x compared to 2D design as shown in Table 4, indicating that 2.5D design has longer connections than 2D monolithic design.

6 CONCLUSION

In this paper, we presented our vertically-integrated EDA flow, which covers and fully automates the whole design phases of architecture, circuit and package. We verified our EDA flow by detailed descriptions of each step using a target design of ROCKET-64 with NoC configuration. We performed PPA comparison between 2.5D IC and its monolithic 2D counterpart. This work, for the first time, provided a full set of quantified comparison results of the 2.5D and 2D designs, which enables the SoC designer to have an objective criteria of evaluating interposer-based design.

ACKNOWLEDGMENTS

This research is funded by the DARPA CHIPS project under Award N00014-17-1-2950.

REFERENCES

[1] opensmart: Single-cycle multi-hop noc generator in bsv and chisel.
[2] K. Asanović et al. The Rocket Chip Generator. Technical Report UCB/EECS-2016-17, EECS Department, University of California, Berkeley, Apr 2016.
[3] R. Chaware, K. Nagarajan, and S. Ramalingam. Assembly and reliability challenges in 3D integration of 28nm FPGA die on a large high density 65nm passive interposer. In *2012 IEEE 62nd Electronic Components and Technology Conference*, pages 279–283, May 2012.
[4] W. Liu, M.-S. Chang, and T. Wang. Floorplanning and signal assignment for silicon interposer-based 3D ICs. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2014.
[5] D. Stow, I. Akgun, R. Barnes, P. Gu, and Y. Xie. Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5D/3D integration. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–6, Nov 2016.