Cross-Boundary Inductive Timing Optimization for 2.5D Chiplet-Package Co-Design

MD Arafat Kabir
University of Arkansas
Fayetteville, AR
makabir@uark.edu

Dusan Petranovic
Mentor Graphics
Fremont, CA
dusan_petranovic@mentor.com

Yarui Peng
University of Arkansas
Fayetteville, AR
yrpeng@uark.edu

Abstract

With the popularity of 2.5D integration, an increasing number of chiplets are integrated into advanced system-in-package designs. In such systems, redistribution layer (RDL) wires become longer and denser, with a growing impact on system performance. However, RDL inductive impacts in timing analysis are ignored by the traditional CAD tools. This paper presents our chiplet-package co-optimization flow, which can capture the RDL inductance impact on system performance and automatically adjust the IO drivers to compensate for the inductance overhead. We develop our extraction and timing analysis tool that models RDL wire inductive timing impact on 2.5D system performance within +/-1% error. Our study shows 35% signal paths through RDL violate the timing requirement because of the inductive impact, and remain undetected through only RC-based STA.

CCS Concepts

• **Hardware → Physical design (EDA)**; **Multi-chip modules**; **Modeling and parameter extraction**; *Package-level interconnect*.

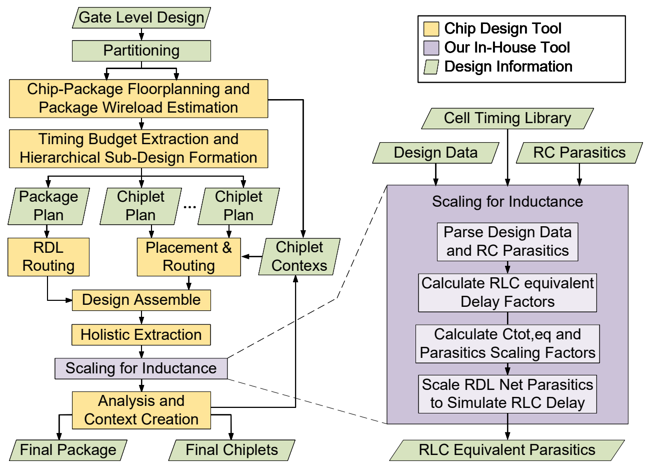
Keywords

2.5D, Chiplet-Package Co-Optimization, Holistic Design, Inductance Modeling, Timing Context

ACM Reference Format:
MD Arafat Kabir, Dusan Petranovic, and Yarui Peng. 2021. Cross-Boundary Inductive Timing Optimization for 2.5D Chiplet-Package Co-Design. In *Proceedings of the Great Lakes Symposium on VLSI 2021 (GLSVLSI '21), June 22–25, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3453688.3461505

1 Introduction

2.5D integration technology is gaining popularity in increasing device density and performance at the system level. It offers heterogeneous integration and energy-efficient high-bandwidth inter-die communication channels. The industry is putting a lot of effort in developing advanced packaging nodes. The dimensions of interconnects on redistribution layers (RDL) have already reached the sub-micrometer range [1]. Due to chipletization and the use of Known-Good-Dies (KGD), it is now feasible to design a very large 2.5D system containing several chiplets. In large systems containing tens of chiplets, long RDL wires are unavoidable. These long RDL wires are going to exhibit significant inductive behavior.


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

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

Figure 1: Holistic co-optimization flow with RDL inductance impact on timing

Traditionally, chip-scale interconnects are modeled using resistive and capacitive (RC) elements. Several previous studies have discussed the impact of inductive (L) elements of interconnects at the high-frequency range. It is observable that the signal oscillations in the RDL wires are significantly underestimated using only RC elements [2]. The study [3] also shows that properties of the driver also affect the oscillatory behavior of the voltage waveform. It is essential to take into account the inductive behaviors of the RDL wires to ensure system reliability and signal integrity of a high-performance 2.5D system with long interconnects.

Due to the complexity of chip routing, many inductance extraction methods proposed for package design are not able to handle dense and fine chip wires [4]. Recently, some studies are performed on novel design techniques for 2.5D system design and optimizations. The co-design methodology proposed in [5] tries to design and optimize a 2.5D system in a unified design environment. Though this extraction methodology is useful to optimize the system timing considering the impact of capacitive coupling between chiplets and the package, it cannot consider inductive effects of RDL wires on the timing. The design methodology proposed in [6] implements a plug-and-play design approach, however, inductance effects are not considered in the timing optimization steps.

In this paper, we present our chiplet-package co-optimization flow for 2.5D systems, which takes into account the inductive effects

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GLSVLSI '21, June 22–25, 2021, Virtual Event, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8393-6/21/06…$15.00
https://doi.org/10.1145/3453688.3461505

of RDL wires on the system performance. Our flow can automatically co-optimize the IO drivers and receivers between chiplets taking into account the timing overhead introduced by the inductive behavior of the RDL wires. Through SPICE simulation, we develop our own RLC interconnect delay model to estimate the path delays through RDL interconnects. We develop our in-house tools that work hand-in-hand with the existing commercial ASIC CAD tools to incorporate the inductance timing overhead in design, analysis, and optimization steps.

Through the work presented in this paper, we claim the following contributions: (1) A new co-optimization flow for designing 2.5D systems taking into account the inductance effects of RDL wires on timing; (2) An RLC delay models for IO drivers based on RDL interconnect properties; (3) A new tool that works alongside the existing ASIC CAD tools and incorporates the inductance effects into the design flow; (4) Comparative study between implementations of a real micro-controller system designed with and without RDL wires inductance considerations.

To the best of our knowledge, there exists no other tool and design flow that implements chiplet-package co-optimization of 2.5D systems while taking into account the inductance overhead of RDL wires on the system performance.

2 RLC Delay Modeling

As discussed in the previous studies [2, 3, 7], at high frequencies, the inductive effects of interconnects become significant on timing. Before a system can be optimized for the inductance overhead, the RLC interconnect delay need to be modeled accurately. The study presented in [7] modeled the global interconnect delay of 0.25 µm CMOS technology taking into account the inductive effects. Though the properties of 2.5D RDL and 0.25 µm CMOS global wires are different, we utilize the delay modeling methodology to develop our own RLC delay model for the RDL wire drivers. We use Nangate45nm PDK as our standard cell library and develop our RDL wire driver RLC delay model for this technology.

2.1 Interconnect Delay Study using SPICE

Compared to the chiplet internal wires, the RDL wires are wider and so have smaller resistance and larger capacitance per unit length. In our experimental design, we use RDL wires with width/spacing of 10 µm/10 µm. For the purpose of comparative study through SPICE simulation, we model the RDL wires as having resistance 0.05 Ω/µm and capacitance 0.068 fF/µm. These values are taken from some previous studies, and the inductance is modeled using the partial inductance of the wire. We calculate the partial inductance using the following equation [8]:

$$L_{l,k} = \frac{\mu_0 l}{2\pi}\left[\ln(\sqrt{k^2+1}+k) - \sqrt{k^{-2}+1} + \frac{0.9054}{k} + 0.25\right] \quad (1)$$

For the partial self inductance, $k = l/r$, where l is the length of the RDL wire and r is the thickness of the wire. Referring to the TSMC InFO UHD technology [1], we use a thickness of 1 µm for the RDL wire. At 2 GHz, the skin depth of copper is approximately 1.45 µm, which is why we can ignore it in this simulation. Fig. 2 shows the circuit used to perform SPICE simulation. A pulse source with 2 GHz frequency and rise and fall times of 10 ps is applied to the gate of an inverter, which drives the gate of the RDL wire driver cell. This driver cell drives a receiver gate connected to the other end of the RDL wire. In a real design, the drivers and receivers are supposed to be within chiplets and connected to the RDL wires through IO pads. We are ignoring the IO pads for this simulation. In multiple runs of the simulation, the RDL interconnect length is changed, and the simulation is performed for the RLC model. For each RDL wirelength, simulation is also performed for the RC model, where the same resistance and capacitance values as in the RLC model are used, but the inductance is not included.

As highlighted by the delay arc in Fig. 2(a), the 50% delay from the gate of the driver cell to the gate of the receiver cell is measured for both RC and RLC interconnect model simulations. Fig. 2(b) shows the plotted simulation data for INV_X16 of the Nangate45nm cell library. As evident from the figure, as the inductance increases with the wirelength, the RC model of the interconnect cannot accurately estimate the propagation delay. Fig. 2(c) shows the relative error of the RC interconnect model w.r.t the RLC model delay measured through SPICE simulation. As observed from this figure, the RC model can underestimate the propagation delay by approximately 30%. This result is consistent with previous studies [3, 7], confirming it is essential to include the inductance impact on timing in the design, analysis, and optimization steps of a system connected through RDL wires.

2.2 Our RLC Delay Model

To estimate the delay through RDL wires of a 2.5D system, we develop our own delay model using SPICE simulations. On RDLs between the chiplets, point-to-point connections are more common. For that reason, we are use a transmission line-based RLC delay model, as presented by Ismail and Friedman in [7]. Based on their work, using the transmission line model, the delay t_{pd} is a function of three variables, ζ, R_T, and C_T, as shown in (4).

$$R_T = \frac{R_{tr}}{R_t},\ C_T = \frac{C_L}{C_t},\ \zeta_{line} = \frac{R_t}{2}\sqrt{\frac{C_t}{Lt}} \quad (2)$$

$$\zeta = \zeta_{line}\frac{R_T + C_T + R_T C_T + 0.5}{\sqrt{1+C_T}} \quad (3)$$

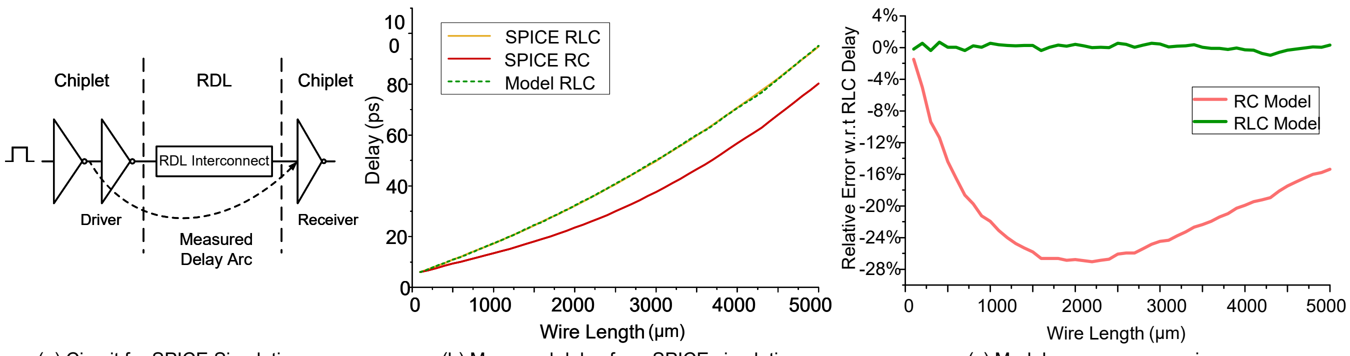
$$t_{pd} = \frac{t'_{pd}(\zeta, R_T, C_T)}{\omega_n} \quad (4)$$

They derived their final model for t_{pd} using only one variable, ζ, for the condition $0 < R_T, C_T < 1$. However, in our experimental setup, using Nangate45nm with InFO-like integration technology, this condition is not met. Additionally, to develop a more accurate delay model for each driver cell, we select two variables, namely ζ_{line} and C_T. The selection is based on the fact that, as seen from (2) to (4), t_{pd} is dependent on R_{tr}, R_t, C_L, C_t, L_t. For a given driver, R_{tr} is fixed and considering ζ_{line} and C_T includes the rest of the variables.

$$scalingFactor = k + a\zeta_{line}^3 + b\zeta_{line}^2 + c\zeta_{line} + d\zeta_{line}^2 C_T \quad (5)$$

$$RLC\ Delay = scalingFactor \times RC\ Delay \quad (6)$$

For a given driver, using ζ_{line} and C_T, we calculate a factor using (5). As shown in (6), this factor is multiplied with the RC delay to estimate the equivalent RLC delay for that driver. A multiplication factor is estimated because it is essential for modeling the RLC delay using RC parasitics as discussed in detail in the next section. Parameters of (5) are obtained by fitting the SPICE simulation data for each driver. Table 1 shows the fitted parameters for some cells



(a) Circuit for SPICE Simulation (b) Measured delay from SPICE simulation (c) Model accuracy comparison

Figure 2: SPICE simulation and validation of our models

of the Nangate45nm cell library. Fig. 2(b) shows the delay estimated by our fitted model for the same cell side-by-side with the simulated RC and RLC models. As seen from Fig. 2(c), where the RC model underestimates the propagation delay by 30% for long wires, our fitted model estimates the delay with an error of only +/-1%. Using this RLC model, in the next section, we develop our in-house tool, which can incorporate the RDL wire inductance impact on overall system performance in a holistic design flow, to iteratively optimize the chiplets to compensate for the overhead.

Table 1: Model parameters for selected Nangate45nm cells

Parameter	INV			BUF		
	X1	X4	X16	X1	X4	X16
a	-0.023	0.132	3.312	0.004	-0.036	1.931
b	0.047	-0.242	-5.783	0.007	0.000	-3.788
c	-0.013	0.156	2.804	-0.006	0.048	2.085
d	-0.008	-0.136	-1.009	-0.007	-0.076	-0.561
k	1.003	1.001	0.961	1.004	1.008	0.938

3 Holistic Co-Optimization Flow

In the current industry trend, 2.5D systems are designed in a die-by-die approach. However, in advanced packaging technologies like InFO UHD, the gap between chiplets and the package is so small that the interactions between them significantly affect the overall system. To ensure maximum performance and reliability, a holistic analysis of the system is essential to capture the interactions among all the components.

2.5D integration technology offers a myriad of features and design opportunities. Though large design houses can benefit themselves from these features, designing a multi-chiplet system is a challenge for small design houses due to limitations of their resources and existing CAD tools. Some designs, like Internet of Things (IoT) devices, are intrinsically small in area and power budget. Having a high-performance IO system adds too much overhead to such a system when implemented as a multi-chiplet system utilizing 2.5D integration technology. This can be solved by implementing a highly-customized IO interface between chiplets simplified into a few standard cells [9]. However, as these cells are not designed for driving long RDL wires, parasitics and static timing analysis (STA) must be performed very carefully to avoid design failures. To overcome these challenges, a CAD flow is essential that can automatically optimize these low-cost IO systems making holistic considerations of the entire 2.5D system. In this section, we present our holistic iterative co-design flow, which uses all standard libraries to design custom pin drivers taking into account the inductance overhead on overall system performance.

3.1 Overall Flow

Fig. 1 demonstrates the steps of our holistic iterative co-optimization flow. The gate-level netlist is synthesized from the RTL description of the entire 2.5D system. This gate-level netlist is then partitioned using a partition tool, which takes into account the impact of package wires on the timing while exploring the partition solutions. After the system is partitioned into chiplets, we perform the co-planning of chiplets and the package together in a unified design environment. A unified PDK is designed using similar settings as presented in Table 1 of [5] to set up the unified environment. As a result, design information can be exchanged between chiplets and the package while performing timing budget extraction and hierarchical sub-design formation of the chiplets and the package. Using an RDL planning tool, the package RC loads are calculated based on the wirelength of the RDL nets. These estimated loads are appended with the hierarchical contexts, created by the top-level design tool, for each chiplets. After this step, chiplets and the package sub-designs can be implemented independently in parallel using the top-level constraints.

The traditional 2D chip design flow is followed to implement the chiplets. However, as the top-level constraints are always in effect during the design and optimization steps, the chiplet design tool takes into account the holistic considerations imposed in the top-level planning step. After finishing the chiplet designs, they are individually checked for design rule violations (DRC). If all the chiplets are violation-free, they are again assembled together in the “Design Assemble" step in the unified design environment for holistic extraction.

Though previous works [5] have shown that the holistic extraction process can capture the impact of the package RC parasitics, no effort has been made to consider the inductive impact of the RDL wires. In the “Scaling for Inductance" step of our flow, we utilize our in-house tool to modify the holistically-extracted parasitic netlist to include the inductance impact on the timing paths going through RDL wires. The inner workings of this tool are discussed in detail in Section 3.2. The tool adjusts the capacitances on the RDL nets in the parasitic netlist such that the STA and context creation tool calculates the RLC equivalent delay on the timing arcs going between chiplets through RDL wires. This adjusted parasitic netlist is used to perform timing analysis and create timing contexts for all chiplets using standard STA tools. As a result, the timing context created for each chiplet contains the timing overhead caused not only by the RC elements, but also the inductive element of the RDL wires.

These timing contexts of the chiplets are used to re-implement the chiplets in their own design environments, but with more accurate estimates of the timing constraints. As a result, during timing optimization steps, the chip design tool makes adjustments within chiplets to compensate for the timing overhead caused by the RDL interconnects. This iterative optimization can be performed until the design goal is met or no more improvement can be achieved. Finally, the finished chiplets and the package designs can be analyzed and checked for the sign-off verifications.

3.2 Parasitics Scaling for Inductance

Though holistic extraction using existing industry-standard tools can accurately capture the capacitive coupling between chiplets and the package, it cannot capture the inductive impact of package wires. As discussed in Section 2, RDL wire inductance can severely affect the overall system performance. Existing commercial STA tools do not support inductance during timing analysis. Although standard parasitics description formats like SPEF support inductance, industry standard tools like Synopsys PrimeTime simply ignore those elements. That is why we design our own tool that can utilize an RLC delay model and adjust the parasitic information for existing STA tools in a way such that the timing contexts created for iterative optimization will take into account the timing overhead caused by the RDL inductance.

Fig. 1(b) shows the workflow of our parasitic adjustment tool. It reads the design data and parasitics netlist extracted through the holistic extraction process. The design data is parsed for RDL nets, their wirelength, driver, and receiver cells within the chiplet. This information is necessary to estimate the RLC equivalent delay using the model discussed in the previous section. The holistically-extracted parasitics contain the RC netlist of the entire system, both chiplets, and package. This netlist is parsed for the RC netlist of the RDL wires. The tool also reads in the timing calculations performed by the analysis and context creation tool using the RC parasitics. All of this information is used to calculate the scaling factor, using equation (5) for driver cells and their RDL interconnects.

The delay calculation result extracted from the STA tool is used to calculate the total RC delay, from the driver input pin in one chiplet to the receiver input pin in another chiplet, by adding the driver cell-timing arc and the RDL wire net-timing arc. Using the parasitics and design information parsed in the previous step, the RDL interconnect parameters, ζ_{line} and C_T, are calculated. Then, using these interconnect parameters in equation (5), along with the fitted parameters for the driver, *scalingFactor* is computed. The RC delay, calculated using the STA tool report, is multiplied by this *scalingFactor* to estimate the RLC delay of the timing arc as shown in Fig 2(a). This estimated RLC delay is used to adjust the RC parasitics netlist and to allow the context creation tool to calculate RLC delay instead of RC delay for the paths running between the chiplets through RDL.

Standard STA tools calculate a path delay as a summation of timing arcs: gate delay from an input pin to its output pin is defined as the cell-arc, and the net delay from an output pin to the input pin is defined as the net-arc. The cell-arc is calculated using a look-up table (LUT) described in the cell timing library, and the net-arc is calculated using the Elmore-delay model on the RC tree of a net. The cell-arc depends on the input transition time (t_r) and the total capacitance (C_{tot}) at the output pin. The way Elmore-delay works on a RC tree, keeping the resistances of the RC tree constant, if a common factor scales all the capacitances in the tree, the total net-delay will be scaled by the same factor.

$$\begin{aligned} RLC &= cell\ delay + net\ delay \\ &= LUT(C_{tot,eq}, t_r) + scalePar \times (RC\ net\ delay) \end{aligned} \tag{7}$$

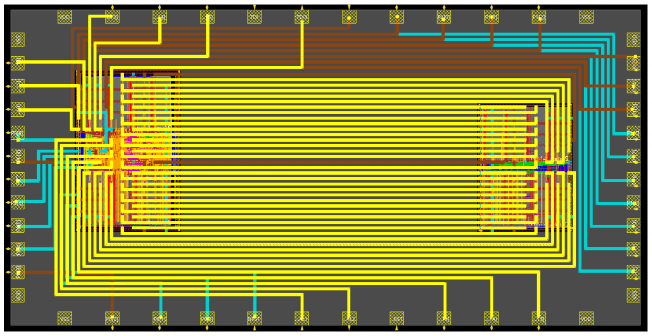
Where,

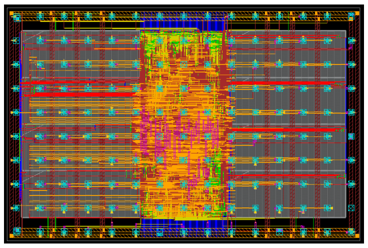
C_{tot}	: Total Capacitance in the Parasitic RC network,
t_r	: Input transition time of the driver cell,
$C_{tot,eq}$	: Total equivalent capacitance in the parasitic network required to simulate the estimated RLC delay, and
LUT	: Cell timing library look-up table
$scalePar$	: $C_{tot,eq}/C_{tot}$

The delay estimated by our model for a specific RDL interconnect, using equations (5)-(6), is the summation of the cell-arc of the driver gate and the net-arc of the RDL wire. Based on the aforementioned relationships, we develop the equation (7), which can be used to look up the total equivalent capacitance ($C_{tot,eq}$) from the cell timing library. That capacitance, when used in the RC parasitics netlist, will force the STA tool to compute the RLC delays for the timing arc from the driver input to the receiver input pins of the RDL interconnects. As a result, the timing contexts for all chiplets created by the tool will take into consideration the timing overhead caused by the RDL wire inductance, along with the RC elements.

To adjust the RC parasitic netlist, a scaling factor, *scalePar*, is calculated for each RDL net. The factor *scalePar* is defined as the ratio of the RLC equivalent total capacitance ($C_{tot,eq}$) and the total capacitance (C_{tot}) on the RC tree of the extracted parasitics netlist. This *scalePar* factor is used to multiply all the capacitances attached to the RDL net RC tree in the parasitic netlist. This scaled parasitic netlist is exported to be used by the STA tool for timing analysis and chiplet timing-context creation. As only the RDL net capacitances are scaled, all the delays calculated by the STA tool using this scaled parasitics netlist, from RDL wire driver input to the receiver input, are the equivalent RLC delay, although the chiplet internal nets are still calculated using the RC delay models. Due to the iterative nature of our design flow, as shown in Fig. 1, the chiplet design tools then adjust the RDL wire drivers and receiver, as well as the internal gates of the chiplets to compensate for the timing overhead caused by the RDL interconnect. Thus, our flow achieves the co-design and optimization goals of the overall 2.5D system.

Due to the modular design of our tool, the RLC interconnect delay model is separate from the rest of the parts of the tool. A more accurate and physics-based model can replace this delay model to create a more accurate and general tool. Combined with existing ASIC CAD tools, they implement holistic co-optimization flows with accurate inductance considerations on timing.


(a) Assembled 2.5D system with chiplets and the package together


(b) Core-Chiplet

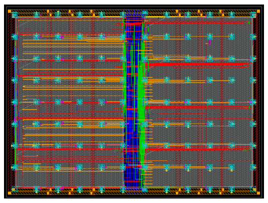

(c) Memory Chiplet

Figure 3: Physical design layouts of chiplets and the package

4 Experimental Study

4.1 Design Setup

To study our flow on a real design, we select an ARM Cortex-M0 based microcontroller to implement as a two-chiplet 2.5D system using InFO-like integration technology. The Nangate45nm PDK is modified to create a unified PDK for planning and designing the package and chiplets together using similar settings as in Table 1 of [5]. The PDK contains seven metal layers for chiplet internal routing and three RDLs for package routing. The system consists of 16KB of memory and several peripheral devices. The 16KB memory system is divided into four banks of 4KB, where each bank consists of four 1KB SRAM macro compiled using OpenRAM [10] memory compiler. The area being dominated by the memory macros, such granular design of the memory system offers flexibility in partition and floorplanning steps.

Fig. 3(a) shows the package floorplan and RDL routing of the experimental system. This system, being a small one, can be designed using very short package wires. We place the chiplets 1000 µm apart on the package, as it is typical for the chiplets to be connected with RDL wires in 1 cm range. As our chiplets are small, we use only the minimum spacing. We choose such a small system because it is easy to control the design parameters for the experimental setup. However, in a more practical 2.5D system with tens of chiplets, RDL wires extended over multiple millimeters would be very common. There are 100 signals running between the two chiplets in this design, which have wirelength varying in the range 1000–2500 µm. In our SPICE simulation, we covered this wirelength range and fitted the driver RLC delay parameters based on the simulation results.

Following the holistic iterative co-optimization flow discussed in the previous section, the entire system is implemented at a target system frequency of 300 MHz. Fig. 3(b)-(c) shows the finished physical design of the two chiplets. Two different designs are prepared for comparative study. In one design, the exact flow of Fig. 1 is followed to perform iterative optimization through holistic extraction and parasitics netlist scaling for inductance consideration. In the other design, all the steps of this flow are followed except for the "Scaling for Inductance" step. The original parasitics extracted through holistic extraction is directly used for timing analysis, context creation, and iterative optimizations. This way, we can pinpoint the exact optimizations performed by the chip design tool accounting for the inductance impact on the overall system performance. Both of the designs required two iterations to reach their best performance.

4.2 Analysis and Results

In this work, we name the system with RDL inductance considerations using scaled parasitics as the RLC-Design, and the system without RDL inductance considerations as the RC-Design. Fig. 4 shows the histogram of the timing analysis result of the paths going through the RDL wires in the RC-Design. The paths with total delay varying within 0.05 ns are binned together. The red bars and green bars show the analysis result obtained using the holistically-extracted parasitics and RLC equivalent scaled parasitics, respectively. As seen from the figure, without consideration of the inductive overhead on the timing path, the STA tool reports zero violating paths at the target frequency of 300 MHz. However, when STA is performed on the same design taking into account the inductive overhead, approximately 35% of the paths through the package violate the timing requirement. The worst violating paths miss the timing requirement by 0.15 ns. In a high-performance GHz design, that means a violation by 20–30% of the clock period. Without careful considerations of these timing overhead caused by the inductive behavior of the RDL wires, the system will fail to run at its nominal speed, even though the sign-off verification report says the system met the timing requirement.

After every iteration of the chiplet physical design, as shown in Fig 1, finished chiplets are assembled with the package for holistic extraction, scaling for inductance, and context creation. Using the chiplet timing context created after the previous iteration means the chiplet physical design tool obtains a more accurate view of the overall system in every subsequent iteration. As a result, it can fine-tune the chiplet designs to suit the system requirements. In our experimental designs, we observe a significant difference in the cell sizes of timing paths going through RDL wires. Fig. 5 shows the distribution of cell size of the drivers and receivers of the signals going through RDL wires. The red and green bars show the cell counts of the final implementations of the RC-Design and the RLC-Design, respectively. As observed from the driver size distribution, the chiplet physical design tool has inserted larger drivers in the RLC-Design to compensate for the delay overhead caused by the inductive effect of RDL wires. Unaware of this delay overhead, the RC-Design uses drivers as small as X3, thus failing to meet the timing requirement. As the only difference between the two implementations is the consideration of RDL wire inductance, this shift in the distribution of the driver size clearly highlights that the chiplet design tool successfully optimizes the delay impact of the inductance of RDL wires.

Significant changes are also observed on the receiver side of RDL wires. The size distribution of the receiver cells in Fig. 5 shows that many of the larger receiver cells in the RC-Design are replaced by smaller cells in the RLC-Design. For example, although there

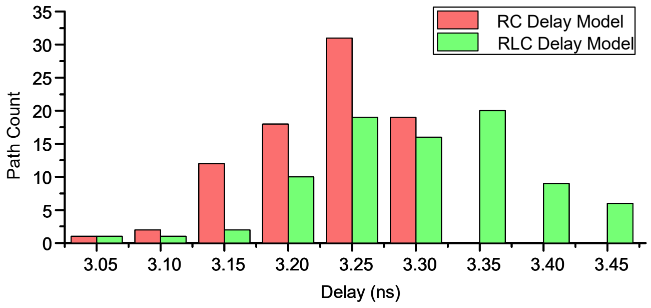


Figure 4: Timing path count per 0.05 ns delay bin through RDL

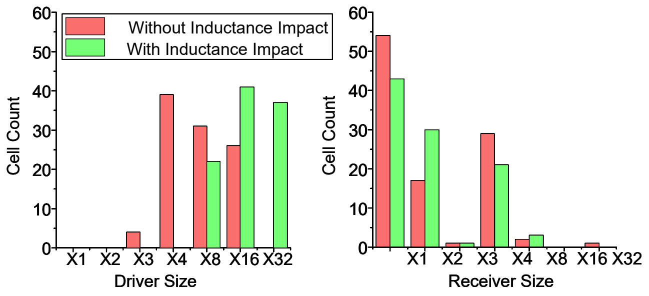


Figure 5: Package inductance impact on cell size distribution

are many X4 and some X32 receivers in the RC-Design, they are replaced by smaller X2 cells in the RLC-Design. This change is performed by the chiplet design tool to reduce the capacitive load on the chiplet pin, which reduces the overall delay on the affected paths.

When using even a large cell in the driving chiplet is not enough to meet the timing, the chiplet design tool takes drastic measures on the receiver side, which is briefly highlighted in Table 2. Though, in general, the receivers are downsized to reduce the load at the input pins, it can be noticed in Fig. 5 that the X1 receiver count decreased and the X2 receiver count increased in the RLC-design. These changes are due to the adjustments performed as in Path-1 and Path-2 of Table 2. In Path-1, four buffer cells of different sizes in the RC-Design are replaced by only one buffer of size X2. In Path-2 of RC-Design, different logic cells of size X1 are directly connected to the chiplet input pin putting a large capacitive load on it. In the optimization steps of the RLC-Design, a single X1 buffer is placed as the receiver that subsequently drives the logic cells within the receiver chiplet. That is, a group of smaller X1 receivers connected to a single input pin is replaced by a single X1 or X2 receiver, thus decreasing overall X1 receiver count or increasing X2 receiver count. In some cases, as in Path-3, where inserting a buffer as the receiver does not help in timing improvement, it simply reduced the logic gate cell size. Note that both chiplets are implemented independently in parallel in their own design environment. This cross-boundary co-optimization between chiplets, to compensate for the inductive delay overhead, is achieved by using chiplet timing contexts created using the RLC equivalent parasitics generated by our in-house tool.

5 Conclusions and Future Work

This paper presents a holistic chiplet-package co-optimization flow and our in-house extraction and timing analysis tools. Our method takes into account the delay overhead caused by the inductance and RC elements of the RDL wires. We developed our RLC delay model and characterized them for the Nangate45nm library driver cells with less than 1% error. Using this delay model, we prepare an experimental design of an ARM Cortex-M0 based microcontroller system and implement it with and without the inductance consideration for a comparative study on a real system. Our study shows that, when using RC model without inductance consideration of RDL wires, approximately 35% of the signal paths through RDLs violate the timing requirement but remain undetected. Through the use of RLC equivalent parasitics and iterative optimization, our flow can automatically co-optimize the drivers and receivers, in different chiplets, connected to the same RDL wire keeping the chiplet physical design process parallel and independent. Our parasitics adjustment tool enables the inductance delay overhead consideration in the existing STA tools and can be further extended to accommodate all RCLM elements in future work.

Table 2: Changes in receivers between RC and RLC Designs

Design	Path-1	Path-2	Path-3
RC	BUF_X4 BUF_X1 BUF_X8 BUF_X2	AOI21_X1 NAND4_X1 XOR2_X1 BUF_X1	AOI22_X4
RLC	BUF_X2	BUF_X1	AOI22_X2

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] H. Pu, H. J. Kuo, C. S. Liu, and D. C. H. Yu, "A Novel Submicron Polymer Re-Distribution Layer Technology for Advanced InFO Packaging," in *IEEE Electronic Components and Technology Conference*, May 2018, pp. 45–51.
[2] Y. Peng, D. Petranovic, and S. K. Lim, "Chip/Package Co-Analysis and Inductance Extraction for Fan-Out Wafer-Level-Packaging," in *IEEE Conference on Electrical Performance of Electronic Packaging and Systems*, Oct. 2017, pp. 1–3.
[3] A. Shebaita, D. Petranovic, and Y. Ismail, "Including Inductance in Static Timing Analysis," in *International Conference on Computer-Aided Design*, Nov. 2007, pp. 686–691.
[4] A. C. Yucel, I. P. Georgakis, A. G. Polimeridis *et al.*, "VoxHenry: FFT-Accelerated Inductance Extraction for Voxelized Geometries," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 4, pp. 1723–1735, 2018.
[5] M. A. Kabir, D. Petranovic, and Y. Peng, "Extraction and Optimization for Heterogeneous 2.5D Chiplet-Package Co-Design," in *International Conference on Computer-Aided Design*, Nov. 2020, pp. 1–8.
[6] J. Kim, G. Murali, H. Park *et al.*, "Architecture, Chip, and Package Codesign Flow for Interposer-Based 2.5-D Chiplet Integration Enabling Heterogeneous IP Reuse," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 28, no. 11, pp. 2424–2437, 2020.
[7] Y. I. Ismail and E. G. Friedman, "Effects of Inductance on the Propagation Delay and Repeater Insertion in VLSI Circuits," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 8, no. 2, pp. 195–206, 2000.
[8] H. A. Aebischer and B. Aebischer, "Improved Formulae for the Inductance of Straight Wires," *Advanced Electromagnetics*, vol. 3, no. 1, pp. 31–43, 2014.
[9] M. Lee, A. Singh, H. M. Torun *et al.*, "Automated I/O Library Generation for Interposer-Based System-in-Package Integration of Multiple Heterogeneous Dies," *IEEE Transactions on Components and Packaging and Manufacturing Technology*, vol. 10, no. 1, pp. 111–122, 2020.
[10] M. R. Guthaus, J. E. Stine, S. Ataei *et al.*, "OpenRAM: An Open-source Memory Compiler," in *International Conference on Computer-Aided Design*, Nov 2016, pp. 93:1–93:6.