Into the Third Dimension: Architecture Exploration Tools for 3D Reconfigurable Acceleration Devices

Andrew Boutros*, Fatemehsadat Mahmoudi*, Amin Mohaghegh*, Stephen More*, Vaughn Betz
Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada
{andrew.boutros, sara.mahmoudi, amin.mohaghegh, stephen.more}@mail.utoronto.ca, vaughn@eecg.utoronto.ca

***Abstract*—Recent chip integration processes enable 3D stacking of multiple active dice in the same package, offering higher logic density, lower power consumption, and significant die-to-die bandwidth. Field-programmable gate arrays (FPGAs) can benefit from 3D chip integration either by stacking multiple homogeneous FPGA fabrics to increase logic capacity or by integrating with other heterogeneous application-specific integrated circuits (ASICs). This opens up a myriad of research questions and interrelated design choices. However, we lack the tools necessary to model these 3D reconfigurable devices and quantitatively explore their vast design space. In this work, we enhance existing FPGA architecture exploration tools and build new ones to address this gap, with a cross-stack focus on circuit-level fabric modeling, 3D integration considerations, system-level architecture, and computer-aided design (CAD) tools. We extend the RAD-Gen framework by integrating an upgraded version of the COFFE automatic transistor sizing tool that supports 7 nm FinFETs with a more accurate, metal-aware area model for newer process technologies. We also implement new tools in RAD-Gen for modeling the inter-die connections and power distribution networks of 3D architectures. In addition, we introduce a new version of the Versatile Place & Route (VPR) tool that can model 3D devices, with enhancements to its architecture description language and its placement and routing engines. Finally, we showcase the capabilities of our enhanced tools by modeling and evaluating both homogeneous and heterogeneous 3D reconfigurable devices.**

I. Introduction

Over the past 25 years, field-programmable gate arrays (FPGAs) have been continuously growing in capacity, enabling the implementation of larger and more complex systems. As an example, the graph in Fig. 1a shows the increase in the logic capacity of the largest Xilinx FPGAs over time. For the first five generations, the increase in logic density was mainly attributed to process technology scaling following Moore's law [1]. However, with more advanced process technologies and especially at early stages of their life cycles, it has become significantly harder to achieve good yield for large monolithic (i.e. single-die) devices. Therefore, FPGA vendors began creating larger devices by integrating multiple smaller (and higher-yield) FPGA dice in the same package using a passive interposer, which is commonly referred to as 2.5D integration [2]. This interposer is a silicon die with a conventional metal stack (but no active transistors on it and thus the name *passive*) that provides a large number of wire connections between two or more dice flipped on top of it, as illustrated in Fig 1b. Xilinx first started using this technology to integrate up to four FPGA dice in their Virtex-7 devices, providing a 4× increase in logic capacity compared to the previous generation. The same approach was used in

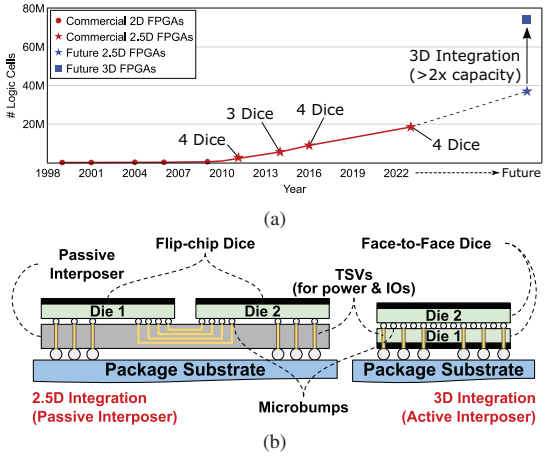


Fig. 1: (a) Growth in logic capacity of commercial FPGAs. The star markers represent devices using 2.5D multi-die integration technology to integrate multiple dice in a single package. Advanced 3D chip integration can further double the logic capacity of future devices by stacking multiple chips on top of each other. (b) Illustration of 2.5D (left) and 3D (right) multi-die integration.

the following generations (star markers in Fig. 1a) including the recently announced Versal Premium device which consists of 4 FPGA dice with a total of 18.5M logic elements [3]. Intel has used a similar technology [4] to integrate multiple FPGA dice or an FPGA die with multiple transceiver and/or high-bandwidth memory (HBM) chiplets in the same package, beginning with their Stratix 10 family [5].

While 2.5D integration significantly increases FPGA logic capacity and enables heterogeneous chiplets, it has two major limitations. Firstly, the number of wires crossing dice is limited and they have a considerably higher delay. For example, in the Xilinx Virtex-7 multi-die FPGA, less than one-fourth of the routing channel wires can cross between dice, with an additional delay of ∼1 ns (4× the delay of a long wire spanning 12 logic blocks on the same die) [6]. Secondly, the crossings between dice are limited to die edges. This creates a harder placement and routing problem as the netlist primitives attached to inter-die connections compete for the (limited) locations closer to the edge, potentially resulting in routing hot spots that need to be carefully managed by the CAD flow [7], [8].

More recent advances in chip integration technologies enable 3D stacking of multiple active dice on top of each other [9], as illustrated in Fig. 1b. This alleviates the limitations of 2.5D integration by providing a high density of vertical inter-die connections that are evenly distributed across the area of the integrated dice. These connections are also faster and consume less power as

*Authors contributed equally to this work.

2837-0449/23/$31.00 ©2023 IEEE
DOI 10.1109/ICFPT59805.2023.00027

signals do not have to traverse long wires on the passive interposer from one die to another. Such 3D chip integration technologies have been used in several commercial products. For example, AMD stacked a 64 MB SRAM chiplet on top of a Zen3 processor to triple its L3 cache capacity [10]. They also announced their next-generation MI300 accelerated processing unit that integrates 13 chiplets including CPU and GPU cores 3D-stacked on top of active dice that handle IOs and other functionalities [11]. Intel also introduced the industry's first logic-on-logic stack in their Lakefield architecture, which integrates both a DRAM memory die and a high-performance CPU/GPU compute die on top of a base die with low-power components such as chipset, IO, and power delivery circuitry [12]. Intel's roadmap for their 3D chip integration technology, Foveros [13], and their next-generation Meteor Lake CPU/GPU architecture that combines 2.5D and 3D integration were recently disclosed [14].

FPGAs could also benefit from advanced 3D chip integration to provide a $\sim 2\times$ increase in logic capacity by stacking two FPGA dice on top of each other (i.e. homogeneous integration). Alternatively, an FPGA fabric die could be stacked on top of a *base die* with large on-chip SRAM memories, coarse-grained ASIC accelerators, and high-performance networks-on-chip (NoCs) for system-level communication (i.e. heterogeneous integration) to implement novel reconfigurable acceleration devices (RADs) [15]. Moving *into the third dimension* opens up a vast design space with a myriad of research questions on both the architecture and circuit-level implementation of these devices, including: Is homogeneous integration feasible and what are its implications on place and route algorithms? What is the density and driver circuitry cost of inter-die connections? How should one architect the base die in the case of heterogeneous integration? However, existing tools cannot be used to model such 3D architectures or answer these questions, and this is the gap we address in this work.

Firstly, we extend the RAD-Gen framework [16] by integrating an enhanced version of COFFE, an automatic transistor sizing tool for FPGA circuitry [17]. We add support for using the ASAP 7 nm FinFET predictive process design kit (PDK) [18], and we include a more-accurate area model for FinFETs which accounts for metal-limited FPGA tile areas to generally improve COFFE's quality of results. Secondly, we introduce a new component for modeling 3D inter-die connections and power distribution networks (PDNs) to RAD-Gen. Using this new RAD-Gen component, we can determine the speed of inter-die connections and the area overhead of their drivers, as well as the density of C4 bumps and through-silicon via (TSV) holes necessary for delivering sufficient power from the package substrate to both dice. Thirdly, we implement new features throughout the Versatile Place and Route (VPR) FPGA CAD flow [19] to support 3D architectures. The new VPR-3D can model both homogeneous FPGA-on-FPGA stacks and heterogeneous 3D RADs with an FPGA fabric on top of an ASIC base die containing accelerator cores and NoCs. Finally, we demonstrate the combined capabilities of our tools via two case studies showcasing homogeneous and heterogeneous 3D integration. This paper's contributions include:

- Extending RAD-Gen with an enhanced version of COFFE that uses the ASAP7 PDK with a more accurate area model for minimum-width FinFETs and metal-limited FPGA tiles.
- Implementing new modeling tools for die-to-die connections and PDNs of 3D-integrated devices in RAD-Gen.
- Introducing VPR-3D which extends VPR's architecture description language and its placement and routing engines to enable modeling and evaluation of 3D architectures.
- Showcasing the capabilities of our architecture exploration tools through two example case studies on homogeneous and heterogeneous 3D reconfigurable architectures.

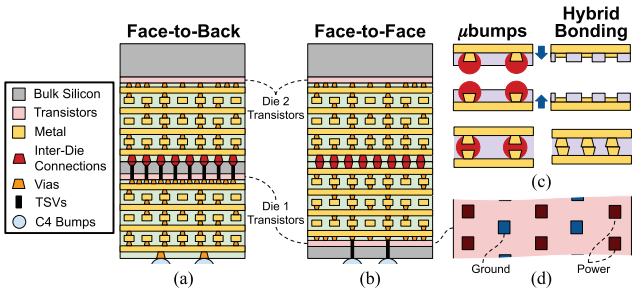


Fig. 2: (a) Face-to-back stacking, (b) Face-to-face stacking, (c) Different approaches for connecting dice, and (d) Top view of bottom die in F2F stacking with PDN Swiss cheese TSV holes.

II. Background & Related Work

3D stacking of multiple active dice offers higher transistor density, a smaller form factor, lower power consumption, and the ability to integrate chips using different processes. This section presents a brief background on the key concepts and terminology of 3D chip integration relevant to this paper, as well as a review of 3D reconfigurable architectures and CAD.

Stacking Options: As illustrated in Fig. 2a and 2b, there are two main options for the 3D integration of active dice: face-to-back (F2B) and face-to-face (F2F) stacking [9]. In this terminology, the *face* of a die is the metal layers, while the *back* is its silicon substrate. In F2B stacking, the bottom die is typically a normal flip-chip with the top metal layer connected to the package substrate via large solder bumps known as *C4 bumps* [20]. The bulk silicon of this die is thinned down and has TSVs going through it to provide die-to-die connections, power and package IO access to the top die as shown in Fig. 2a. This option is used by AMD's 3D-stacked L3 cache [10] and is a repeatable structure that allows the stacking of more than two dice. However, it requires a large number of TSVs in the bottom die for inter-die connections, costing area. On the other hand, F2F integration is used in Intel's Foveros 3D stacking [13]. In this option, the bottom die is no longer a flip-chip; it is facing upwards and has TSVs drilled in it to provide connections to the C4 bumps for IOs and power. Then, the second die is flipped on top so that the highest metal layers of both dice can be directly connected as in Fig. 2b. In this paper, we focus mainly on F2F stacking of two active dice. However, our tools and infrastructure are parameterizable and can be easily extended to model more than two stacked dice using either of the F2B or F2F options.

Die-to-Die Bonding: For F2F stacking, the inter-die connections between the metal stacks of the two dice (dark red in Fig. 2b) can be realized using the two approaches illustrated in Fig. 2c. The first approach uses μm-scale solder bumps (μbumps). The top metal layers of the two dice have short metal pillars protruding from them at the

inter-die connection locations. These pillars are plated with tin-silver (SnAg) solder and then the two dice are bonded by melting the solder using thermal compression [13]. These μbumps can be fabricated with pitches ranging from 50-10 μm, providing 400-10,000 inter-die connections/mm^2. For a higher density of inter-die connections, a more advanced process known as hybrid bonding is used [21]. In this process, small copper pads are placed on the top metal layers of both dice and separated by insulating oxide. When the two F2F dice are pressed together, the oxide is chemically activated and bonds at room temperature. Then, the copper pads are annealed to expand and form direct connections between the two metal layers, as illustrated in Fig. 2c. This provides a higher density of connections since there is no longer a need for the (relatively) larger μbumps. However, it requires a higher precision fabrication process to ensure the proper alignment of the copper pads on both dice. Intel's latest Foveros Direct [14] technology can realize hybrid bonding connections with a pitch size of $<$10μm (i.e. more than 10,000 inter-die connections/mm^2), and several research efforts have demonstrated the feasibility of scaling down to 1μm-pitch bonds [22], [23]. From a modeling perspective, there is no difference between a μbump and a directly-bonded connection except for their density, resistance (R) and capacitance (C) values. Therefore, our tools can model both bonding approaches given their pitches and electrical properties.

Power Distribution Networks: A chip's PDN is responsible for distributing power and ground voltages to all the active transistors on the chip. An ideal (but infeasible in practice) PDN would deliver the exact Vdd and ground voltages to the transistors. However, since resistive solder bumps, via stacks, and metal wires are used to distribute current from the power source to transistors, a voltage drop (i.e. *IR drop*) is incurred and the voltages supplied to the transistors are less than Vdd and higher than ground, resulting in slower switching speeds. Therefore, PDN design aims to minimize IR drop, as it directly affects the speed of the chip, without consuming excessive metal area. For F2F 3D-stacked dice, enough C4 bumps are needed to supply power for both dice. As shown in Fig. 2b, the power C4 bumps are connected to the metal stack of the bottom die via TSVs, creating unusable regions of silicon due to the PDN's TSV holes (analogous to *Swiss cheese holes*) as illustrated in Fig. 2d. For our PDN modeling in this work, we mainly focus on quantifying the unusable portion of the bottom die area (FPGA resources or silicon footprint in homogeneous and heterogeneous integration) due to the PDN holes required for a chip of a given size and power consumption.

3D FPGAs: The idea of a 3D FPGA was presented by Alexander et al. in 1995 as a conceptual generalization of a conventional 2D FPGA in which each switch box has 6 instead of 4 neighbors [24]. Although it was referred to as a 3D device, the envisioned physical implementation did not stack FPGA dice on top of each other, but rather integrated them using a passive interposer where the (conceptually) vertical connections are implemented as inter-die interposer wires. Several works investigated stacking multiple FPGAs using monolithic 3D integration; a process that sequentially constructs multiple transistor layers on a single substrate [25]. The Rothko 3D FPGA [26] proposed stacking multiple sea-of-gates architectures on top of each other, with the output of each block (3-input lookup table and D-latch) connected to same-die routing as well as the blocks below and above. This work was then extended to integrate an FPGA fabric, a routing fabric, and a memory layer for storing multiple bitstreams to dynamically reconfigure the other two layers [27]. Lin et al. [28] also investigated the performance benefits of stacking the FPGA's routing switches and configuration SRAMs in separate layers on top of an FPGA's logic block grid. The work by Ababei et al. studied the potential benefits of stacking FPGA dice by implementing placement and routing algorithms for 3D FPGAs [29], [30]. Their flow first partitioned a circuit netlist and independently placed each partition on its corresponding layer. Then, nets are routed with penalties for bends from 2D to vertical connections to minimize the use of vertical connections if same-layer routes between two netlist primitives are available. Other works also investigated the design of 3D switch boxes for 3D FPGAs and their implications on FPGA routing algorithms [31], [32].

Heterogeneous Integration: Gadfort et al. investigated the 3D integration of DRAM, an accelerator FPGA with floating-point cores, and a control FPGA [33]. They demonstrated the power efficiency benefits of such heterogeneous integration on simple fast Fourier transform and sorting applications, but did not consider any physical implementation details for 3D integration. More recent work from Intel also evaluated the integration of FPGAs with deep learning acceleration ASIC chiplets using passive interposers for enhanced performance and energy efficiency [34], [35].

In comparison to all prior efforts, this work is the first to present a cross-stack (circuits, architecture, and CAD) study of homogeneous and heterogeneous 3D reconfigurable devices, with careful consideration of the physical implementation details of modern 3D stacking processes. Additionally, we add support for modeling these 3D architectures in the widely used, open-source VPR CAD flow and evaluate the tool's quality of results (QoR) using large benchmarks representative of today's FPGA use cases from the Koios suite [36].

III. RAD-Gen for Modeling 3D Architectures

RAD-Gen is part of the architecture exploration and evaluation flow for novel RADs that combine an FPGA fabric, application-specific accelerator cores, and high-performance NoCs for system-level communication [16]. It is used to evaluate the feasibility and implementation cost of a candidate RAD architecture by obtaining ASIC power, performance and area (PPA) results for system infrastructure components (e.g. NoC routers) and user-specified accelerator cores. In this work, we expand the scope of RAD-Gen to also evaluate 3D reconfigurable devices.

To model such devices, we need to model FPGA fabrics in recent process technologies that match the advanced 3D integration processes we are targeting. Therefore, we enhance the COFFE tool [17] to use the 7nm ASAP PDK [18] and integrate it into the RAD-Gen framework. We also implement new tools in RAD-Gen to model 3D physical design considerations such as inter-die connections and PDNs. With these enhancements to the RAD-Gen framework, it can model: (1) full-custom FPGA logic blocks (LBs) and programmable routing circuitry, (2) fabric-embedded standard-cell (i.e. *hard*) blocks and their full-custom interfaces to the programmable routing, (3) hard NoCs and coarse-grained accelerator cores in RADs, and (4) inter-die connections and PDNs in 3D devices.

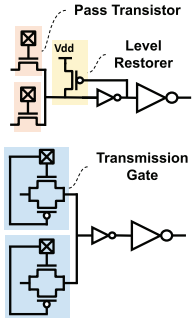
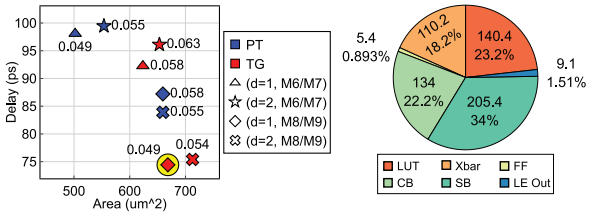


Fig. 3: Pass transistor (top) and transmission gate (bottom) switches.

TABLE I: List of fabric architecture parameters modeled in COFFE.

Parameter	Value
LUTs per LB (N)	10
Frac. LUT size (K)	6
Frac. LUT independent inputs	1
Adders per LUT	2
Channel width (W)	320
Wire Length (L)	4/16
LB inputs (I)	40
LUT outputs to routing (O_r)	2
Feedback LUT outputs (O_{fb})	2
Switch box flex. (F_s)	3
Input connection flex. (F_{ci})	0.2
Output connection flex. (F_{co})	0.025
Input crossbar flex. (F_{cl})	0.5
Nominal voltage (V_{dd})	0.7
Gate boost voltage (V_{boost})	0.1



(a) (b)

Fig. 4: (a) COFFE area-delay results of L4 pass transistor (PT) and transmission gate (TG) logic tiles for general routing on different metal layers and different delay optimization weights d. The annotations are the area-delay products. (b) Area (μm^2) of the logic tile for an architecture with 280×L4 and 40×L16 wires per channel.

A. COFFE Enhancements

COFFE is an automated transistor sizing tool for FPGAs. It generates netlists for all the full-custom FPGA circuitry in logic blocks, routing, block memories (BRAMs), and the interfaces of embedded ASIC blocks (i.e. *hard blocks*) to the programmable routing. Then, it runs HSPICE simulations to iteratively optimize the transistor sizes of these circuits given a cost function of $area^a \times delay^d$. COFFE calculates the area of subcircuits using a model that estimates the area of a transistor of a given drive strength (i.e. diffusion width) compared to a minimum-width transistor, which is the smallest contactable transistor in a specific process plus the spacing area to its neighboring transistors. COFFE also enables the evaluation of new FPGA hard blocks by running an ASIC implementation flow to obtain their PPA results.

*1) **More Accurate & Metal-Aware Area Model**:* COFFE's area model was originally formulated for planar (i.e. bulk) transistor technologies, which was shown to be inaccurate for FinFETs in newer process technologies [37]. Therefore, to obtain more accurate results when using the FinFET-based ASAP7 PDK, we modified COFFE to use the area model from [37] which was adjusted for FinFETs and verified against actual layouts of FPGA pass transistor multiplexers. Additionally, the COFFE area model only accounted for the active area of transistors and did not consider the metal wires needed for routing as highlighted in [38]. As we move to newer processes, it is desired to improve the delay of general routing wires by moving them to higher metal layers that are less resistive and have wider pitches. This could lead to metal-limited FPGA tile areas and result in different area-delay trade-offs. For example, if an FPGA tile area is limited by metal, it could be more favorable to upsize transistors for better delay results to make the best use of the non-active tile area. To consider this aspect, we added two new inputs to COFFE: the metal pitch and number of metal layers used for general routing. The tool then uses these inputs to calculate the minimum tile dimensions needed for a routing channel of width W in the horizontal and vertical dimensions, and flags cases when the tile area is metal-limited. In such cases, the user can re-run transistor sizing optimization with a higher weight on delay optimization (d) if desired.

*2) **Modular ASIC Flow for Hard Blocks**:* Originally, COFFE wrote custom scripts to run the ASIC flow for hard block cores using Synopsys Design Compiler for synthesis, Cadence Innovous for place and route (PnR), and Synopsys PrimeTime for static timing analysis (STA). If a user did not have access to one of these tools they had to define their own custom scripts for an alternative flow, which required ASIC tool expertise due to the large number of settings and configurations. In the case of ASAP7, the documented and tested standard cell ASIC flow uses Cadence Genus for synthesis [39]. To solve such issues, we upgraded COFFE to use the HAMMER tool [40] from the UC Berkeley Chipyard framework [41]. This tool provides a unified YAML-based configuration syntax to enable ASIC design flow reusability across different tools/vendors and process technologies. It also comes with plug-ins for commonly used Cadence, Synopsys and Siemens tools for synthesis, PnR, and STA, as well as a technology plugin for ASAP7.

*3) **Modeling a 7nm FPGA Fabric Architecture**:* The area of an FPGA fabric is dominated by lookup tables (LUTs) and programmable routing multiplexers (MUXes), which are built using transistor switches. These switches can be implemented as pass transistors (PTs) or transmission gates (TGs), as shown in Fig. 3. An NMOS PT switch is a single transistor, but its output saturates at $V_g - V_{th}$, where V_g and V_{th} are the gate and process threshold voltage. This requires a PMOS level restorer to pull-up the output voltage of the PT to V_{dd} and/or applying $V_g > V_{dd}$ (i.e. gate boosting) to mitigate this issue. On the other hand, a TG can pass the full voltage to the output, but it is larger; it consists of an NMOS and a PMOS with their gates driven by the bitline and inverse bitline of an SRAM cell, respectively. Chiasson and Betz [42] demonstrated that in 22nm process technology, a pass transistor implementation with 0.2 V gate boosting still had a 2% better area-delay product than a transmission gate one. However, PT performance and reliability issues have increased in recent processes because V_{th} is scaling poorly and the amount of gate boost that can be safely applied is declining. Therefore, we revisit the study in [42] for the 7nm process technology we are targeting.

Table I shows the fabric architecture parameters that we use in COFFE. We obtain the R and C values per unit length of wire using the scripts from [43] by plugging in the corresponding pitch, barrier thickness, copper's relative permittivity, and the maximum aspect ratio of the wire height and width of the ASAP7 metal layers. We first model an architecture with only length 4 wires (i.e. L4) and we experiment with general routing wires using one M6/M7 or M8/M9 layer per direction with 64 nm or 80 nm pitches, respectively and with both pass transistor and transmission gate implementations. We also experiment with both $area \times delay$ and $area \times delay^2$ optimization cost functions (i.e. $d = \{1, 2\}$) and find that using $d = 1$ yields a similar delay and better area-delay product than $d = 2$. Fig. 4a shows that

TABLE II: Homogeneous integration feasibility. Cells highlighted in green (yellow) are design points with enough inter-die connections for full (partial) tile output connectivity to the tile above/below. Cells highlighted in red are infeasible design points.

	μbumps					H-bonds	
Pitch (μm)	**55**	**40**	**36**	**25**	**10**	**5**	**1**
μbumps/bonds per LB ($604\mu m^2$, 20 Outputs)	0.1	0.2	0.2	0.5	3	12	302
μbumps/bonds per DSP ($2150\mu m^2$, 72 Outputs)	0.3	0.6	0.8	1	10	43	1075
μbumps/bonds per BRAM ($2418\mu m^2$, 40 Outputs)	0.4	0.7	0.8	2	12	48	1208

TG designs (red markers) are faster than PT designs (blue markers). PT designs have lower area with M6/M7 general routing, but are similar in area to TG when M8/M9 general routing is used as the tile becomes metal-limited. PT circuitry is also less reliable and harder to design as it cannot provide full output swings and requires carefully-sized level restorers, which are very sensitive to on-die variation in newer process technologies. Moving from M6/M7 to the faster M8/M9 metal layers significantly reduces the representative critical path delay by 17% on average. Therefore, it is a more favorable design choice and corresponds to the RC values of the metal layers used for general routing in [43]. Overall we find a $d = 1$ TG design that uses M8/M9 metal for L4 routing wires is the best design point (highlighted in yellow in Fig. 4a).

A major drawback in COFFE is that it can only model architectures that have a single length of routing wires. Therefore, to model an architecture that uses a mix of short and long wires, we model an L16 architecture in COFFE and combine the resulting switch block MUXing delay and area with the L4 results to model an architecture with 280×L4 and 40×L16 wires per channel similar to that used in the Stratix-IV VPR architecture capture [44]. Fig. 4b shows the area breakdown of the LB tile of the L4/L16 architecture. The LUTs and FFs constitute <25% of the tile area, while the rest of the area is dedicated to routing interfaces and switches.

In addition to logic tiles, we also use COFFE to obtain implementation results for digital signal processing (DSP) tiles. We use the baseline DSP from [45], which implements all the operations of a Stratix 10 DSP except for floating-point support. Our enhanced COFFE (1) invokes the HAMMER flow to synthesize, place, and route the DSP core using the ASAP7 standard cells and (2) implements the DSP interfaces to the programmable routing using its full custom flow. The DSP standard cell core runs at more than 1 GHz and occupies an area of 1175 μm^2. The DSP's full-custom interface and tile's programmable routing circuitry (local crossbar, drivers for dedicated routing between DSPs, switch/connection blocks) sized by COFFE take an additional 975.6 μm^2. Therefore, the DSP tile area is 3.6× larger than the LB tile area. This is in reasonable agreement to the 3:1 DSP-to-LB area ratio in Stratix V [46], particularly considering that the fracturable LUT we model in COFFE is simpler (and hence smaller) than that of Stratix V. COFFE does not model 7nm-optimized BRAM circuits, such as sense amplifiers and write drivers, which we plan to add in future work. As an alternative, we conservatively estimate 20Kb BRAM delays using values from a capture of the 14nm Stratix 10 architecture in VPR [47] and estimate BRAM area as 4× the area of an LB based on the Stratix V values [46].

*4) **Homogeneous Integration Feasibility**:* For 3D architectures stacking two FPGA dice, we assume that the output signals of each FPGA tile can connect to the inputs of the switch block of the tile above or below. This means that homogeneous integration is feasible only if the number of inter-die connections (μbumps or direct bonds) that can fit within a tile area is 2× the number of tile outputs, assuming that the inter-die connections are unidirectional links (i.e. their drivers are not tri-state buffered). Table II lists the area of each FPGA tile from COFFE and its number of outputs in comparison to the number of inter-die connections that can fit within this area per direction for different inter-die connection pitches. Using 1μm-pitch hybrid bonding can provide significantly more inter-die connections than needed to connect all tile outputs to the tile above/below. Homogeneous architectures could still be feasible using 5μm-pitch hybrid bonding but with a restricted number of tile outputs crossing dice (12 out of 20 for LBs and 43 out of 72 for DSPs). However, homogeneous integration is infeasible using larger pitch μbumps, as they provide a very limited number of inter-die connections.

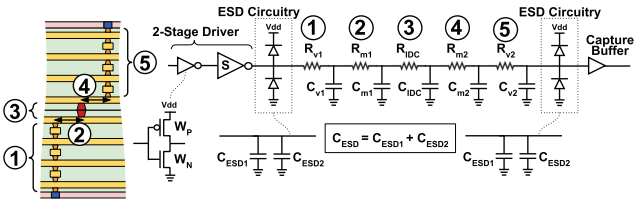


Fig. 5: Die-to-die connection modeled as a series of RC loads with electrostatic discharge (ESD) protection circuits on both ends.

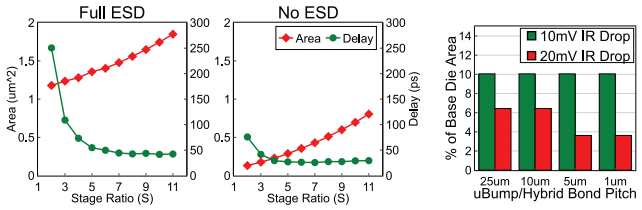
B. Modeling 3D Considerations

We develop new tools in RAD-Gen to model different physical implementation considerations for 3D die stacking, with key questions being: (1) How fast are the vertical connections? (2) What is the area overhead for their drivers? (3) What percentage of the base die area is unusable due to TSV holes for power delivery?

*1) **Modeling of 3D Signal Interfaces**:* As illustrated in Fig. 5, the path from a driver output on one die to a buffer input on the other die consists of: ① a via stack to the top metal layer of one die, ② some wire length on the top metal layer to reach an inter-die connection (μbump or direct bond), ③ the inter-die connection itself, ④ some wire length on the top metal layer of the other die, and ⑤ another via stack to reach transistors. This path can be modeled as a series of RC loads for the via stack (R_v, C_v), top metal layer distance (R_m, C_m), and the inter-die connection (R_{IDC}, C_{IDC}). In addition, the inter-die signal interfaces can be exposed to electrostatic discharge (ESD) events during the handling and bonding of dice in the 3D integration process, which needs to be controlled by ESD protection circuitry [48]. As shown in Fig. 5, these ESD protection circuits typically consist of a pair of diodes (sometimes with a series resistor), introducing an additional area and delay overhead for each inter-die signal interface. However, the inter-die signal interfaces in 3D-stacked dice may require less ESD protection compared to external IO pads as they do not necessarily lie on any of the main ESD current paths [49]. The continuous scaling of die-to-die connection densities is also pushing towards more thorough evaluation of the ESD protection levels needed for inter-die signal interfaces to reduce their area and delay

TABLE III: RAD-Gen inputs for inter-die signal delay & PDN modeling.

General Parameters									
Metal Stack [18] [43]									
Metal	M_a		M_b		M_c		M_d		M_e
# Layers	3		2		2		2		3
Pitch(nm)	36		48		64		80		648
R(Ω/μm)	131.2		58.5		27.1		15.3		0.14
C(fF/μm)	0.23		0.23		0.23		0.23		0.23
Via Stack [18] [43]									
Via	V_a	V_{ab}	V_b	V_{bc}	V_c	V_{cd}	V_d	V_{de}	V_e
Pitch(nm)	36	$\rightarrow$	48	$\rightarrow$	64	$\rightarrow$	80	$\rightarrow$	648
R(Ω)	13.1	9.2	7.3	5.2	4.2	3.2	2.8	0.3	0.1
Total Via Stack R	58.6 Ω								
Inter-die Connections [22]									
Technology	μbump				H-bond				
Pitch(μm)	25		10		5		1		
R($m\Omega$)	40		99		17		97		
C(fF)	34		3		0.1		0.07		
Inter-die Signal Delay Modeling									
ESD Circuitry Capacitance [49]	{0, 20} fF								
ESD Circuitry Area [51]	1.048 μm^2								
Top Metal Distance [52]	34 μm ($\frac{1}{2}WH$ 20Kb SRAM)								
PDN Modeling									
C4 Dim. [22]	80×80μm			C4 Pitch [22]	100 μm				
C4 R [22]	13 mΩ			TSV Pitch [22]	10 μm				
TSV R [22]	47 mΩ			Power	45W/die				
Metal Layers Used	2			% of Metal	50%				
Target IR Drop	10/20mV								



(a) (b)

Fig. 6: (a) Area overhead and delay of inter-die 3D signals with ESD (left) and without ESD (right) for 1μm pitch hybrid bonds. (b) Unusable area of the base die due to TSV holes for homogeneous integration targeting 10mV and 20mV IR drop with 50% top metal layer utilization and 45W power consumption per die.

overhead [50]. For delay measurements, these ESD diodes can be modeled as two parallel capacitances (C_{ESD1}, C_{ESD2} in Fig. 5).

RAD-Gen takes as inputs all the relevant RC values and the traversed top-layer metal distance, and uses HSPICE to size the two-stage inter-die signal driver given a range of sizing ratios between driver stages (S). Then, it calculates the area of the driver for each design point using the enhanced COFFE area model. For the ESD diodes, Karp et al. [51] describe the design of an area-efficient ESD solution from the 2.5D Xilinx Ultrascale architecture, with an area of 0.224μm^2 per NMOS-based ESD diode in 20nm process technology. We conservatively use this result to calculate the total cost of the two ESD circuits (see Fig. 5) with two NMOS-based and two PMOS-based diodes as 1.048μm^2. Table III lists the input parameters to RAD-Gen for modeling inter-die signal delays for different μbump and hybrid bond pitches, and the value(s) that we use for each parameter. Fig. 6a shows the die-to-die signal delay and its driver's area overhead with and without ESD circuitry for the 1μm hybrid bonds that we use for homogeneous integration (see Table II). ESD protection constitutes 56-89% of the driver area overhead, and increases the signal delay due to its capacitive load compared to the case with no ESD circuitry. Increasing the stage ratio (S)

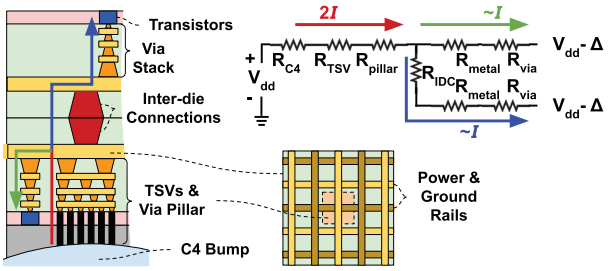


Fig. 7: PDN modeling in RAD-Gen.

results in a larger and stronger second stage inverter, with $S = 4$ achieving a good area-delay trade-off. For this $S = 4$ design choice with full ESD protection, the inter-die signal delay is 73 ps and the driver area overhead is 1.28μm^2 per signal. For a LB with 20 output signals, the inter-die signal drivers to provide full output connectivity to the die below/above represents a modest 4.2% tile area overhead. Results for different μbump/bond pitches show similar trends but were omitted for brevity. We present the 1μm pitch results as an example to highlight the utility of our tools in guiding the architectural decisions of inter-die connectivity and its area-delay tradeoffs.

*2) **Modeling of 3D PDN**:* As explained in Section II, delivering power to both the bottom and top F2F-stacked dice requires drilling TSVs in the base die substrate to connect the package C4 power bumps to the bottom die metal stack. Fig. 7 details how RAD-Gen models the PDN to quantify the unusable area of the base die due to the TSV holes. It assumes that a grid of TSVs of given dimensions is centered on top of a package C4 bump. The area on top of the TSV grid is filled with as many parallel vias as possible depending on the via pitch of each metal layer to reduce the overall resistance of the *via pillar*. The pillar is connected to the highest metal layers of the bottom die (directly) and top die (via μbumps or hybrid bonds) for power and ground distribution to the vicinity of the C4 bump using alternating rails that consume a user-specified percentage of these metal layers. Then, via stacks are used to deliver power/ground down to the transistors of both dice.

RAD-Gen starts by assuming a single power C4 bump in the center of the base die. Then, it recursively splits the bump region into quadrants and places power C4 bumps in their centers to gradually shrink the region powered by each bump, and thus reduce the distribution IR drop (Δ) to an acceptable user-specified threshold. Finally, it replicates the whole grid to form ground C4 bumps with an offset in the horizontal and vertical dimensions to create an alternating checkerboard pattern (see Fig. 2). The amount of IR drop can be determined using the PDN circuit model shown in Fig. 7, with resistances for the C4 bump (R_{C4}), TSV grid (R_{TSV}), via pillar (R_{pillar}), distribution metal (R_{metal}), inter-die connection (R_{IDC}), and via stacks down to transistors (R_{via}). As the region powered by a C4 bump is smaller, R_{metal} is reduced resulting in smaller IR drop at the cost of more power C4 bumps and thus more TSV holes in the base die. RAD-Gen determines the current drawn by each die (I) assuming a uniform current density and a user-specified power consumption estimate. For current distribution throughout a 2D region, RAD-Gen assumes that the current decays linearly as we move from the center of the region (via pillar) towards the edges. Therefore, it divides this distance into equal slices corresponding to the pitch of the via stacks, and performs a discrete integral to estimate the total IR

```
<!--FPGA on top of NoC die w/o prog. routing-->
<layout name="3d_rad" height="4" width="4">
  <layer die="0" has_prog_routing="false">
    <fill type="noc_die0">
  </layer>
  <layer die="1">
    <fill type="fabric_die1">
  </layer>
</layout>
<!--NoC router on layer 0 connects to layer 1-->
<tile>
  <sub_tile name="noc">
    <pinlocations pattern="custom">
      <loc layer_offset="1">noc.tdata[128:0]</loc>
    </pinlocations>
  </subtile>
</tile>
```

Listing 1: Snippet from a VPR architecture file specifying a heterogeneous 3D RAD with a fabric die stacked on a NoC die.

drop of the region.

Table III also lists RAD-Gen's input parameters for PDN modeling and the value(s) that we use for each parameter. We estimate a 45W power consumption for each die using the Intel Quartus power estimator for a medium-sized Agilex device with 70% resource utilization and a signal toggle rate of 12.5% as a proxy to the FPGA architectures we model. For the homogeneous integration case, we limit the TSV grid on top of the C4 bump to the area of 2×2 LBs (60μm×40μm from our COFFE results in Section III-A). This is the largest integer multiple of LB area that can fit on top of a C4 bump (80μm×80μm) to minimize the number of FPGA tiles removed by base die TSVs, rather than just minimizing the number of TSV holes. On the other hand, for the heterogeneous integration case, we fit as many TSVs as possible on a C4 bump to reduce holes resulting in the least disruption to the ASIC base die.

Fig. 6b shows the portion of unusable area of the base die of a homogeneous device for 10mV and 20mV maximum allowable IR drop, when using 50% of the top two metal layers for power and ground distribution (one for horizontal and one for vertical rails). The results show that the unusable base die area is 10.1% for a 10mV IR drop, and ranges from 3.6% for the smaller hybrid bonds to 6.4% for the larger μbumps in case an IR drop of 20mV is tolerable. An analysis of the PDN of an L2 cache 3D-stacked on top of a 16nm Arm CPU core reported IR drops in the range of 24-43mV for different μbump and hybrid bond pitches [22]. Thus, we use a 20mV IR drop target for our PDN modeling for the rest of the paper.

IV. 3D Versatile Place & Route

VPR is an open-source CAD flow used for FPGA architecture and CAD research [19]. It takes as inputs an FPGA architecture described in `xml` format and a design `blif` netlist [53]. Then, it performs packing, placement and routing of the given design on the target FPGA architecture, and produces implementation results such as the total routing wirelength (WL) and critical path delay (CPD). In this work, we modify the different stages of VPR to support multiple 3D-stacked FPGAs and push back our upgrades to the main open-source VPR repository. While this paper focuses on two-die F2F stacking, the modifications to VPR also support RADs with any number of stacked dice.

A. Architecture Description Language Extensions

We extend VPR's architecture description language by introducing a new `<layer>` tag that allows users to flexibly describe multiple stacked dice and their resource compositions/layouts (i.e. homogeneous or heterogeneous). Listing 1 shows the definition of a heterogeneous RAD with an FPGA fabric (`fabric_die1`) stacked on top of a NoC base die (`noc_die0`). The attribute `has_prog_routing` allows users to specify whether a particular die has a programmable routing fabric or not, which is set to `false` for this base die as it uses a NoC for all connectivity. To describe the connectivity between dice, we also extend the `<loc>` tag used to describe the position of the input/output pins of each block to include a new `<layer_offset>` tag. For a given input/output pin, this offset is modulo-added to the block's own layer ID to identify the layer on which this pin logically exists. In addition, the extended architecture description language allows users to specify different switch types with specific delays and electric properties (obtained from RAD-Gen) for the inter-die connections. These upgrades maintain backward compatibility; existing 2D architecture files do not need to include these new tags as the default remains a single die with all pin `layer_offset` set to 0.

B. 3D Placement & Routing

VPR uses a simulated-annealing-based placer guided by reinforcement learning (RL) [54] to dynamically choose more effective placement move types during different phases of the anneal. To add support for 3D-stacked architectures in VPR, we first modify its placement representation to a 3D grid. We also extend a subset of the existing placement moves to enable not only xy location changes of netlist primitives, but also 3D layer changes. The RL agent automatically learns to use these moves and tunes their selection probabilities accordingly. The programmable routing fabric of an architecture is modeled as a routing resource (RR) graph, in which input/output pins and wire segments are nodes and the connections between them are edges. Therefore, the routing problem can be formulated as finding non-overlapping trees between sources and sinks within this graph, while optimizing certain implementation metrics such as WL and CPD. To perform routing in 3D architectures, we significantly extend the RR graph generation code in order to meet the new 3D specifications, and add a new `layer_num` value to each RR graph node. Both the placement and routing engines require fast estimates of the amount and delay of the routing resources needed to connect to locations; we update both these *lookahead* data structures to consider not only the $(\Delta x, \Delta y)$ distance spanned but also any die layer crossing. While these changes are extensive, they maintain WL and CPD quality and increase VPR's runtime by <3% on 2D devices.

V. Case Studies

Fig. 8 illustrates sectors of the two example 3D architectures we model: a homogeneous FPGA-on-FPGA and a heterogeneous FPGA-on-ASIC devices. A thorough architecture exploration of such devices is beyond the scope and capacity of this paper, so these case studies are presented to showcase the modeling capabilities and QoR of our tools.

A. Homogeneous Integration

The homogeneous architecture stacks two identical FPGA dice on top of each other as shown on the left of Fig. 8. Firstly, we use RAD-Gen to model the FPGA fabric circuitry via COFFE and obtain area and delay results of different FPGA blocks and routing components, as presented in Section III-A.

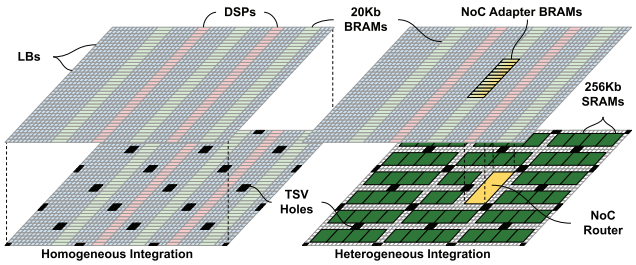


Fig. 8: Example sectors of the homogeneous and heterogeneous 3D-stacked architectures we can model and evaluate using our tools.

The areas of different FPGA tiles obtained from RAD-Gen showed that homogeneous integration is only feasible using advanced hybrid bonding 3D integration using 5μm and 1μm pitch bonds with partial and full connectivity of block output pins to the switch box on the die above/below respectively (see Table II). Then, we model the inter-die signal driver area overhead and delay as explained in Section III-B1. The design point with the best area-delay tradeoff has an inter-die signal delay of **73ps** and **76ps** for the 1μm (full output connectivity) and 5μm (60% output connectivity) hybrid bonding technologies, respectively. The inter-die signal drivers with full ESD protection circuitry come at the cost of a **4.2%** increase in programmable fabric area. After that, we model the PDN to quantify the unusable area of the base die due to TSV holes as detailed in Section III-B2. We design a PDN targeting a 20mV IR drop, which requires drilling TSV holes into 5.6% of the base die LBs (**3.6%** of the base die area) for both the 1μm and 5μm hybrid bonding cases.

Finally, we use the area/delay results and fabric layout obtained from RAD-Gen to write a 3D VPR architecture file as explained in Section IV. We introduce empty TSV hole blocks in the architecture file at the grid locations and spacings determined by RAD-Gen as shown in Fig. 8. To evaluate the QoR of VPR-3D, we first create a family of five 2D fixed-layout fabric architectures of varying sizes and synthesize, place and route each of the Koios benchmarks [36] on the smallest device with sufficient resources to collect the baseline results. Then, we implement each circuit on a 3D homogeneous architecture where each die has roughly half the resources of the corresponding 2D architecture for both the full (1μm h-bonds) and 60% (5μm h-bonds) output connectivity cases. For the case with limited output connectivity, we constrain the packer to use no more than 60% (12) of the 20 LB outputs. This constraint guarantees it is always possible to route all necessary signals between dice and increases the number of utilized LBs by only 4% on average. For a fair comparison, we also apply the same output-pin-limited packing option to the 2D baseline architecture so it can be compared to the 3D architecture with partial connectivity. We also modify the router for the 3D architecture with partial connectivity such that it routes the connections of a net that must cross dice before the within-die connections of that net. This ensures that a die-crossing branch always has higher priority to use the limited number of outputs connected to the other die; routing these more restricted connections first creates a partial routing tree from which same-die connections can branch off. This helps resolve output pin congestion and improves routing quality for this architecture. Fig. 9 compares the routed WL and CPD of the 2D baselines (with default and output-pin-limited packing) to the two 3D architecture variations (with full and partial 3D connectivity). The results show that the 3D-stacked architecture with full output connectivity can improve the CPD and WL by 3% and 4% respectively. The 3D architecture with partial output connectivity also achieves similar gains for CPD and WL (2% and 5% respectively) compared to the 2D baseline architecture with output-pin-limited packing. Fig. 10 highlights that netlist primitives are placed fairly evenly on both dice, indicating that the placer is leveraging the 3D structure to shorten connections. The 3D CAD flow is also efficient; the total VPR run time is 3% and 30% lower on average for the full and partial output connectivity architectures compared to the 2D architecture, respectively.

B. Heterogeneous Integration

For the heterogeneous integration scenario, we consider a RAD architecture that stacks an FPGA fabric on top of an ASIC base die including a high-performance packet-switched NoC for system-level communication, as well as many large SRAM banks that offer larger on-chip memory capacity at a higher access latency through the NoC compared to in-fabric BRAMs. Moving an embedded NoC to the lower die will not only save active area on the fabric die, but also reduce demand for scarce upper-metal routing wires on the fabric die. Unlike the homogeneous case, this ASIC base die does not have programmable routing switch blocks that output pins on the top die can connect to. Instead, we decide to use some top-die BRAMs as access points to the NoC routers on the base die. The interface between the FPGA fabric logic and the NoC routers requires clock domain crossing and width adaptation circuitry, which are typically implemented using asynchronous FIFOs with different read and write widths. These FIFOs can be implemented using a group of BRAMs on the top die that directly connect to the NoC router below (as illustrated in Fig. 8) using the μbumps or h-bonds available within the BRAM tile areas. This provides the additional advantage of performing the width adaptation on the top die, reducing the number of die-crossing signals by $4\times$. We model a heterogeneous architecture with 8×8 fabric sectors on the top die and an 8×8 mesh NoC (i.e. one NoC router per sector) on the base die. The sectors are sized to have a 36×41 grid of FPGA tiles with 3/2 columns of BRAMs/DSPs each, which is in the same size range of *clock sectors* in commercial architectures. The tile areas of all the BRAMs in a sector can provide enough μbumps/h-bonds to connect to the NoC input interface (145b per direction in this example) when using μbumps of 25μm pitch or smaller.

We follow the same steps as in homogeneous integration to model the FPGA fabric on the top die, the inter-die signal delay and driver overheads, and the 3D PDN. Then, we capture this heterogeneous architecture in VPR, and use the multi-layer perceptron (MLP) NoC-attached streaming accelerator benchmark from [55] to evaluate the QoR. We compare this 3D heterogeneous RAD to a 2D one with a comparable NoC integrated in the same die [55]. The 3D RAD reduces the CPD by 23%, programmable routing WL by 1%, NoC average latency by 3% and NoC aggregate bandwidth by 3%. The significant reduction in CPD is mainly because the relatively large NoC router blocks (each of which occupies an 8×8 grid of FPGA tiles in the 2D case) are moved to the base die, allowing the rest of the circuit primitives to be placed closer to each other.

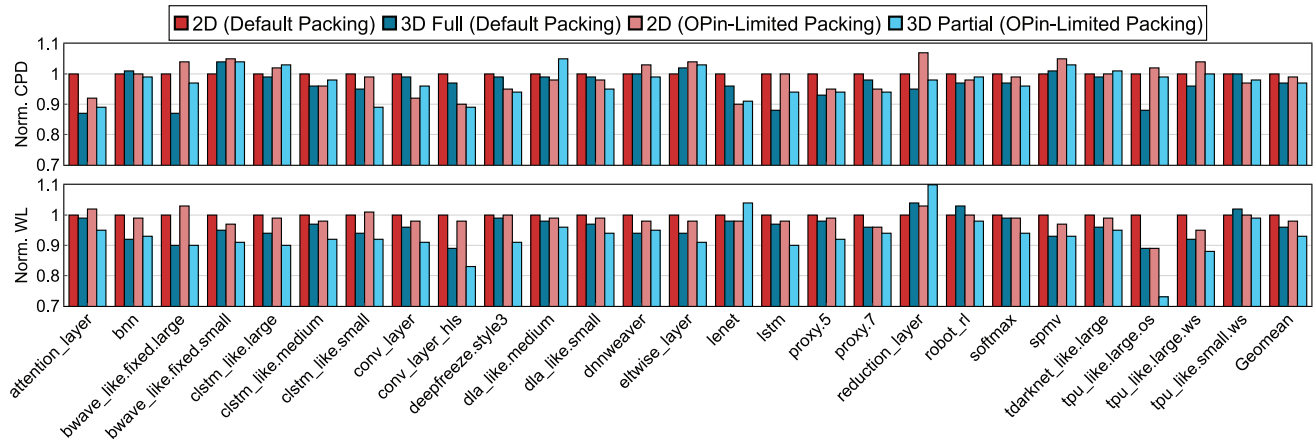


Fig. 9: CPD (top) and WL (bottom) of the Koios benchmarks on baseline 2D architectures with default and output-pin-limited packing vs. 3D-stacked homogeneous architectures with full and partial block output pin connectivity. The output-pin-limited packing option is essential for 3D architectures with partial connectivity between dice, but results in a 4% increase in the LB count.

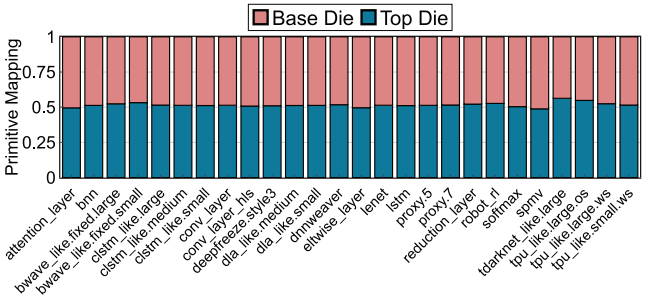


Fig. 10: Allocation of netlist primitives of the Koios benchmarks to the top/base die of the 3D homogeneous architecture with full output connectivity and default packing.

We also use the HAMMER standard cell flow in RAD-Gen to evaluate the ASIC implementation cost of the NoC router and the SRAM capacity that can fit in a base die sector area budget. We use the NoC router from [56] with a virtual channel buffer size of 8 words and a flit width of 195b (145b from the fabric plus a control header), and combine four 64b×1024 ASAP7 SRAM macros to implement 256Kb SRAM banks to be placed in the base die sector. Given the layout and spacings of TSV holes obtained from RAD-Gen, we can lay out 68 SRAM banks per sector as illustrated in Fig. 8, which consumes 56% of the base die area. For our 8×8 sector device, this offers 136 MB of on-chip SRAMs that are accessible from anywhere on the FPGA fabric via the NoC. Besides SRAM, the 64 NoC routers on the base die including their inter-die signal drivers consume 3.2% of its area while the TSV holes take another 3.6%. This leaves ∼37% of empty base die area that can be used to harden any additional application-specific accelerators. Using RAD-Gen, we estimate that this empty area can fit forty-six 32-lane `fp16` dot product engines, providing an additional 188 tera operations per second (TOPS) of peak base die performance.

VI. Conclusion

Recent chip integration technology advances open the door for new 3D-stacked reconfigurable architectures with higher logic capacity, smaller form factor, and higher bandwidth die-to-die connectivity. In this work, we developed the tools necessary to quantitatively explore this new architecture space. First, we extended the RAD-Gen framework to model all the physical aspects of 3D reconfigurable systems. RAD-Gen builds on and enhances COFFE to automatically optimize and model the full custom portion of programmable fabrics and routing in advanced process technologies, using a FinFET-based and metal-aware area model to improve accuracy. RAD-Gen also models the physical aspects of networks-on-chip and standard cell hard blocks that can either attach via the programmable routing or directly to an NoC. In addition, RAD-Gen models the key costs and constraints of 3D integration, including the delay and area of inter-die connections and the area consumed by the power delivery network TSV holes in the base die. Second, we add the ability to model and target a wide variety of 3D RADs to the VTR flow; this support is not only open-source but also integrated into the VTR master branch for future research to build on.

We showcase the features of these new tools with two case studies: a homogeneous FPGA-on-FPGA device and a heterogeneous FPGA-on-ASIC device. We demonstrate that homogeneous 3D FPGAs are feasible with partial and full block output inter-die connectivity at hybrid bond pitches of 5μm and 1μm, respectively. Both these design points perform well, not only increasing logic capacity but also improving CPD by 2-3% and WL by 4-5%. The FPGA-on-ASIC device demonstrates that an FPGA can move the system-level NoC to another die; by leveraging BRAMs to perform the rate/width conversion between the programmable fabric and the NoC we minimized 3D signal count so that even a fairly relaxed bump pitch of 25 μm is sufficient. The ASIC die has sufficient space to also accomodate 136 MB of SRAM and 188 TOPS of `fp16` dot-product engines, highlighting the potential of heterogeneous RADs for compute-intensive applications. The design space of heterogeneous 3D-RADs is very large; with RAD-Gen and VPR-3D we now have the tools to explore it.

Acknowledgments

The authors would like to thank Phil Knag and Gregory Chen from Intel's Circuits Research Lab for the insightful discussions on 3D chip integration technologies and the Intel/VMware Crossroads 3D-FPGA Academic Research Center for funding support.

REFERENCES

[1] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[2] A. Boutros and V. Betz, "FPGA architecture: Principles and progression," *IEEE Circuits and Systems Magazine*, vol. 21, no. 2, pp. 4–29, 2021.

[3] AMD, Inc., "Versal Premium VP1902 Adaptive SoC Product Brief," 2023.

[4] R. Mahajan *et al.*, "Embedded Multi-Die Interconnect Bridge (EMIB): A High Density, High Bandwidth Packaging Interconnect," in *IEEE Electronic Components and Technology Conference (ECTC)*, 2016.

[5] Intel Corp., "Intel Stratix 10 GX/SX Device Overview," 2022.

[6] E. Nasiri *et al.*, "Multiple Dice Working as One: CAD Flows and Routing Architectures for Silicon Interposer FPGAs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 5, pp. 1821–1834, 2015.

[7] C. Ravishankar *et al.*, "Placement Strategies for 2.5 D FPGA Fabric Architectures," in *International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2018.

[8] M. Khatti *et al.*, "PASTA: Programming and Automation Support for Scalable Task-Parallel HLS Programs on Modern Multi-Die FPGAs," in *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2023.

[9] G. H. Loh *et al.*, "Processor Design in 3D Die-Stacking Technologies," *IEEE Micro*, vol. 27, no. 3, pp. 31–48, 2007.

[10] J. Wuu *et al.*, "3D V-Cache: the Implementation of a Hybrid-Bonded 64MB Stacked Cache for a 7nm x86-64 CPU," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2022.

[11] L. Su, "AMD Keynote," in *Consumer Electronics Show (CES)*, 2023. [Online]. Available: https://tinyurl.com/amdceskeynote

[12] S. Khushu and W. Gomes, "Lakefield: Hybrid Cores in 3D Package," in *Hot Chips Symposium (HC)*, 2019.

[13] D. Ingerly *et al.*, "Foveros: 3D Integration and the Use of Face-to-Face Chip Stacking for Logic Devices," in *IEEE International Electron Devices Meeting (IEDM)*, 2019.

[14] W. Gomes *et al.*, "Meteor Lake and Arrow Lake Intel Next-Gen 3D Client Architecture Platform with Foveros," in *Hot Chips Symposium (HC)*, 2022.

[15] A. Boutros *et al.*, "Architecture and Application Co-Design for Beyond-FPGA Reconfigurable Acceleration Devices," *IEEE Access*, vol. 10, pp. 95 067–95 082, 2022.

[16] ——, "A Whole New World: How to Architect BeyondFPGA Reconfigurable Acceleration Devices?" in *International Conference on Field Programmable Logic and Applications (FPL)*, 2023.

[17] S. Yazdanshenas and V. Betz, "COFFE 2: Automatic Modelling and Optimization of Complex and Heterogeneous FPGA Architectures," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 12, no. 1, pp. 1–27, 2019.

[18] L. T. Clark *et al.*, "ASAP7: A 7-nm FinFET Predictive Process Design Kit," *Microelectronics Journal*, vol. 53, pp. 105–115, 2016.

[19] K. E. Murray *et al.*, "VTR 8: High-Performance CAD and Customizable FPGA Architecture Modelling," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 13, no. 2, pp. 1–55, 2020.

[20] K. DeHaven and J. Dietz, "Controlled Collapse Chip Connection (C4): An Enabling Technology," in *IEEE Electronic Components and Technology Conference (ECTC)*, 1994.

[21] H.-W. Hu and K.-N. Chen, "Development of Low Temperature CuCu Bonding and Hybrid Bonding for Three-Dimensional Integrated Circuits," *Microelectronics Reliability*, vol. 127, p. 114412, 2021.

[22] L. Zhu *et al.*, "Power Delivery Solutions and PPA Impacts in Micro-Bump and Hybrid-Bonding 3D ICs," *Transactions on Components, Packaging and Manufacturing Technology (CPMT)*, vol. 12, no. 12, pp. 1969–1982, 2022.

[23] A. Jouve *et al.*, "1μm Pitch Direct Hybrid Bonding with¡ 300nm Wafer-to-Wafer Overlay Accuracy," in *IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, 2017.

[24] M. J. Alexander *et al.*, "Three-Dimensional Field-Programmable Gate Arrays," in *IEEE International Application Specific Integrated Circuits Conference*, 1995.

[25] K. Dhananjay *et al.*, "Monolithic 3D Integrated Circuits: Recent Trends and Future Prospects," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 3, pp. 837–843, 2021.

[26] M. Leeser *et al.*, "Rothko: A Three-Dimensional FPGA," *IEEE Design & Test of Computers*, vol. 15, no. 1, pp. 16–23, 1998.

[27] S. Chiricescu *et al.*, "Design and Analysis of a Dynamically Reconfigurable Three-Dimensional FPGA," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 1, pp. 186–196, 2001.

[28] M. Lin *et al.*, "Performance Benefits of Monolithically Stacked 3D-FPGA," in *International Symposium on Field Programmable Gate Arrays (FPGA)*, 2006.

[29] C. Ababei *et al.*, "Three-Dimensional Place and Route for FPGAs," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2005.

[30] ——, "Placement and Routing in 3D Integrated Circuits," *IEEE Design & Test of Computers*, vol. 22, no. 6, pp. 520–531, 2005.

[31] G.-M. Wu *et al.*, "Universal Switch Blocks for Three-Dimensional FPGA Design," *IEE Proceedings-Circuits, Devices and Systems*, vol. 151, no. 1, pp. 49–57, 2004.

[32] A. Gayasen *et al.*, "Designing a 3-D FPGA: Switch Box Architecture and Thermal Issues," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 7, pp. 882–893, 2008.

[33] P. Gadfort *et al.*, "A Power Efficient Reconfigurable System-in-Stack: 3D Integration of Accelerators, FPGAs, and DRAM," in *IEEE International System-on-Chip Conference (SOCC)*, 2014.

[34] E. Nurvitadhi *et al.*, "In-package Domain-Specific ASICs for Intel Stratix 10 FPGAs: A Case Study of Accelerating Deep Learning using TensorTile ASIC," in *IEEE International Conference on Field Programmable Logic and Applications (FPL)*, 2018.

[35] ——, "Why Compete When You Can Work Together: FPGA-ASIC Integration for Persistent RNNs," in *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2019.

[36] A. Arora *et al.*, "Koios 2.0: Open-Source Deep Learning Benchmarks for FPGA Architecture and CAD Research," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[37] S. R. Sani *et al.*, "Measuring the Accuracy of Layout Area Estimation Models of Tile-Based FPGAs in FinFET Technology," in *IEEE International Conference on Field-Programmable Logic and Applications (FPL)*, 2020.

[38] S. Rostami-Sani *et al.*, "Evaluating the Impact of using Multiple Metal Layers on the Layout Area of Switch Blocks for Tile-Based FPGAs in FinFET 7nm," in *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2022.

[39] X. Xu *et al.*, "Standard Cell Library Design and Optimization Methodology for ASAP7 PDK," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017.

[40] H. Liew *et al.*, "HAMMER: A Modular and Reusable Physical Design Flow Tool," in *ACM/IEEE Design Automation Conference (DAC)*, 2022.

[41] A. Amid *et al.*, "Chipyard: Integrated Design, Simulation, and Implementation Framework for Custom SoCs," *IEEE Micro*, vol. 40, no. 4, pp. 10–21, 2020.

[42] C. Chiasson and V. Betz, "Should FPGAs Abandon the Pass-Gate?" in *IEEE International Conference on Field programmable Logic and Applications (FPL)*, 2013.

[43] S. Nikolić *et al.*, "Global is the New Local: FPGA Architecture at 5nm and Beyond," in *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2021.

[44] K. E. Murray *et al.*, "Titan: Enabling Large and Complex Benchmarks in Academic CAD," in *IEEE International Conference on Field programmable Logic and Applications (FPL)*, 2013.

[45] A. Boutros *et al.*, "Embracing Diversity: Enhanced DSP Blocks for Low-Precision Deep Learning on FPGAs," in *IEEE International Conference on Field Programmable Logic and Applications (FPL)*, 2018.

[46] R. Rashid *et al.*, "Comparing Performance, Productivity and Scalability of the TILT Overlay Processor to OpenCL HLS," in *IEEE International Conference on Field-Programmable Technology (FPT)*, 2014.

[47] K. T. Khoozani *et al.*, "Titan 2.0: Enabling Open-Source CAD Evaluation with a Modern Architecture Capture," in *IEEE International Conference on Field Programmable Logic and Applications (FPL)*, 2023.

[48] Global Semiconductor Alliance, "Electrostatic Discharge (ESD) in 3D-IC Packages," 2015.

[49] E. Rosenbaum *et al.*, "ESD Protection Networks for 3D Integrated Circuits," in *IEEE International 3D Systems Integration Conference (3DIC)*, 2012.

[50] Industry Council on ESD Target Levels, "White Paper 2: A Case for Lowering Component Level CDM ESD Specifications and Requirements (Rev 3.0)," 2021.
[51] J. Karp *et al.*, "Interposer FPGA with Self-Protecting ESD Design for Inter-Die Interfaces and its CDM Specification," in *IEEE International Reliability Physics Symposium (IRPS)*, 2016.
[52] V. Vashishtha *et al.*, "Robust 7-nm SRAM design on a predictive PDK," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.
[53] University of California at Berkeley, "Berkeley Logic Interchange Format (BLIF)," 1992.
[54] M. A. Elgammal *et al.*, "RLPlace: Using Reinforcement Learning and Smart Perturbations to Optimize FPGA Placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2021.
[55] S. Srinivasan *et al.*, "Placement Optimization for NoC-Enhanced FPGAs," in *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2023.
[56] D. U. Becker, *Efficient Microarchitecture for Network-on-Chip Routers*. Stanford University, 2012.