

Modeling, Optimizing and Exploring Multi-Die FPGA Routing Architectures

Amirhossein Poolad^{*†}, Soheil Gholami Shahrouz^{*†}, Andrew Boutros[‡], and Vaughn Betz[†]

[†]University of Toronto, Canada [‡]University of Waterloo, Canada

{amir.poolad, s.shahrouz}@mail.utoronto.ca, andrew.boutros@uwaterloo.ca, vaughn@eecg.utoronto.ca

Abstract—Die stacking has enabled 2.5D FPGAs by integrating multiple active dice on a passive silicon interposer for improved yield and capacity, and has paved the way for 3D architectures that stack active dice directly atop one another. In these multi-die devices, the unique electrical and physical characteristics of the underlying die-stacking technology impose limitations on inter-die connection density and latency, necessitating a bespoke inter-die routing architecture. However, the absence of accurate and versatile modeling tools has left most questions about how to best design the inter-die routing architecture unanswered.

To address this gap, we enhance the open-source FPGA CAD tool VTR to flexibly specify a wide range of multi-die routing architectures. The placement and routing engines in VPR have also been augmented to improve optimization for both 2.5D and 3D FPGAs. We model a wide variety of inter-die connections in 2.5D/3D FPGAs that use a 7nm process node for active dice and a 45nm process node for the silicon interposer and several die-crossing technologies. Using this enhanced version of VTR, we conduct a detailed design space exploration of inter-die routing architecture in 2.5/3D FPGAs. We characterize the impact of die-crossing technology, the number of inter-die connections, and the detailed inter-die routing architecture on critical path delay (CPD) and area. We further investigate how varying the inter-die wire length in interposer-based FPGAs influences the trade-off between routability and CPD. Our results show that with suitable inter-die routing architectures and emerging die-stacking technologies, 2.5D and 3D FPGAs can increase capacity without significant routability or delay trade-offs. Specifically, 3D FPGAs achieve up to 14% wirelength reduction and 6% CPD improvement over 2D devices, and remain routable even with existing 10 μ m pitch technologies, while 2.5D FPGAs incur only a 2% wirelength and 4% CPD overhead at 32% inter-die connectivity.

Index Terms—2.5D FPGA, Silicon interposer, 3D FPGA, FPGA architecture, FPGA routing architecture

I. INTRODUCTION

Applications such as low-latency deep learning acceleration [1], pre-silicon ASIC emulation [2], [3], and data center networking offload [4], [5] have driven demand for FPGAs with ever-greater logic capacity. Historically, this demand was met through two approaches: scaling out to large multi-FPGA cluster systems [6], or relying on Moore’s law and process technology advances to deliver higher logic density on a single monolithic die [7]. Both approaches, however, face fundamental limitations: multi-FPGA systems suffer from high-latency, low-bandwidth inter-FPGA links that bottleneck application performance [8], while achieving good yield for large monolithic dice has become increasingly difficult with advanced process nodes, particularly in the early stages of a new node’s life cycle [9].

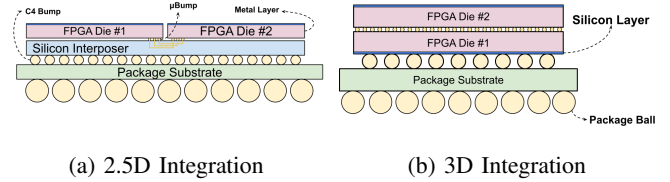


Fig. 1: Example of 2.5D and 3D integration in FPGAs.

Advanced packaging technologies address both limitations by integrating multiple smaller, higher-yield dice into a single package [10]. Using a passive silicon interposer to connect dice side by side (2.5D integration), or stacking active dice directly atop one another (3D integration), this approach delivers high logic density comparable to monolithic scaling while offering far greater inter-die bandwidth and lower latency than multi-FPGA cluster systems. FPGAs were among the first compute devices to adopt 2.5D integration [11]. Xilinx’s Virtex-7 [12] (28 nm) and Virtex UltraScale (20 nm) FPGAs, for example, use passive silicon interposers to integrate three to four FPGA dice; the largest interposer-based devices provide more than twice the logic elements of the largest monolithic FPGA at the same process node [13]. Beyond capacity scaling, 2.5D integration also enables heterogeneous devices combining FPGA fabric with memory or processing components in the same package [14], [15]. Several academic works have explored 3D FPGA architectures, demonstrating that stacking active dice vertically yields higher logic density and shorter critical paths [16]–[18]. Fig. 1 illustrates representative 2.5D and 3D FPGA integration schemes.

Inter-die connections introduce fundamental constraints on routing architecture design. In Virtex-7 interposer-based FPGAs, only 23% of vertical routing tracks cross between dice through the interposer [11], and each inter-die crossing incurs an additional delay of ~ 1 ns [12]. In 3D integration, full output pin connectivity across layers requires advanced hybrid-bonding technology, whereas μ bumps limit inter-layer connectivity to only a fraction of output pins [16]. These constraints make routing architecture a particularly critical design consideration in multi-die FPGAs, more so than in conventional 2D devices. Existing studies have explored various aspects of multi-die routing architecture, yet the multifaceted interplay of all relevant parameters has not been fully characterized, leaving room for a more comprehensive treatment. A versatile method to accurately describe and model multi-die routing architectures, capturing parameters such as inter-die connection density, fan-in and fan-out, and for 2.5D devices, the physical length and electrical characteristics of interposer wires,

^{*}Both authors contributed equally to this work.

is therefore needed for comprehensive design space exploration.

The main contributions of our work are:

- An early-evaluation methodology to assess the feasibility of 2.5D FPGAs across various die stacking technologies.
- Detailed electrical modeling of inter-die connections.
- A new method to accurately and flexibly describe inter-die routing architectures integrated into the VTR CAD tool’s architecture language and architecture generator to enable end-to-end evaluation of 2.5D and 3D FPGAs.
- Multi-die aware place and route flow enhancements to best exploit the capabilities of these new systems and ameliorate their weaknesses.
- A comprehensive study of 3D and 2.5D routing architectures in terms of delay, area and CAD runtime. Our results show that 3D devices with a better than 2D area-delay product can be built using existing die-stacking technologies that previous studies have ruled out as infeasible. We also show a 25% variation in area-delay product between our best and worst 2.5D architectures, further indicating the importance of inter-die routing architecture design.

II. BACKGROUND & PRIOR WORK

Semiconductor foundries currently use two main methods for 3D die stacking [10], [19]. The first approach uses μ -scale solder bumps (μ Bumps), in which bumps are deposited on each die and the dice are then soldered together. The second approach, hybrid bonding, eliminates solder entirely by directly fusing copper pads on opposing die surfaces. μ Bump-based interconnects are used in the TSMC CoWoS, TSMC SoIC-P, and some iterations of Intel Foveros, with reported pitches of $50\mu\text{m}$ [20], $45\mu\text{m}$ [21], $36\mu\text{m}$ [22], and $25\mu\text{m}$ [23]; the International Roadmap for Devices & Systems (IRDS) projects these pitches reaching as low as $10\mu\text{m}$ in the near future [24]. Hybrid bonding is employed in TSMC SoIC-X and Intel Foveros Direct 3D. SoIC-X supports pad pitches from $9\mu\text{m}$ down to $0.3\mu\text{m}$ [25], with $9\mu\text{m}$ and $6\mu\text{m}$ pitches already in production [26], while Intel reports sub- $10\mu\text{m}$ pitches for Foveros Direct 3D [27]. This finer pitch allows hybrid bonding to offer orders of magnitude greater inter-die connectivity than μ Bump-based processes. Although hybrid bonding is technically bump-less, throughout this work we use the term “bump” interchangeably for both technologies.

The choice of bump pitch has direct implications for 2.5D FPGA architects, as it governs the trade-off between die-crossing delay and inter-die connection density. The total number of available wires is bounded by the bump count. Extending interposer wires deeper into the die, away from the die edge, exposes more area for bump placement and thereby increases connection density, but comes at the cost of higher wire delay. To quantify this trade-off, Ravishankar et al. [28] show that for a $7\text{mm} \times 12\text{mm}$ die with a coarse $45\mu\text{m}$ μ Bump pitch, wires spanning $\frac{2}{3}$ of the die height, already considerably longer than a standard vertical wire, would amount to only 25% of the normal vertical routing wires. This illustrates how coarser bump pitches reduce inter-die connection density and increase die-crossing delay, compounding the design challenge. Raikar and Stroobandt [29] further investigated this trade-off by studying the optimal interposer wire length, finding that longer wires increase the available bump area and inter-die connectivity but at the cost of higher delay.

Beyond the physical constraints imposed by bump pitch, the way interposer wires interface with the FPGA routing fabric also shapes

die-crossing performance, and this has evolved across Xilinx architectures. The 7-Series uses a *routing-based* approach, where interposer wires extend the normal routing fabric, with reported delays of around 1 ns [30]. UltraScale adopts a *tile-based* approach, replacing routing resources with specialized tiles featuring an optional register, improving timing at the cost of displacing CLBs. Versal takes a *hybrid* approach, adding an optional interposer connection to each CLB output pin, avoiding CLB displacement while still supporting registered die-crossing paths [14].

These commercial architectures have inspired a body of academic work aimed at modeling interposer-based FPGAs and developing interposer-aware CAD algorithms. Nasiri et al. [31] extend VPR to model routing-based 2.5D devices similar to the 7-Series architecture. They model 2.5D FPGAs as 2D devices with ‘interposer cuts’ that remove all connections across the cut boundary. Interposer wires are then added to the device to restore inter-die connectivity. Their findings show that an inter-die connectivity ratio of 20% relative to intra-die connectivity is sufficient for reliable routability without significant delay increase, particularly when interposer wires are bidirectional and have a fan-in and fan-out greater than 1. To mitigate the resulting routability issues and increased delay, they use a simulated annealing-based placer with a cut cost term that seeks to minimize the number of signals that cross between dice, and also evaluate an optional graph partitioning step to further minimize inter-die connections [31].

Expanding that work, Iyer et al. [32] built a bespoke timing-aware partitioning tool for 2.5D FPGAs. Ravishankar et al. [28] introduced a partitioning-based placer targeting AMD’s UltraScale architecture. Similarly, Raikar and Stroobandt introduced LiquidMD [33], extending the analytical placement flow Liquid [34] that incorporates a graph partitioning pre-processing step for 2.5D devices. Conversely, Di et al. [35] introduce LEAPS, a purely analytical placer that does not rely on a graph partitioner. While these placement algorithms help reduce die-crossing costs, inter-die latency continues to impact design speed, motivating CAD flows that incorporate interposer-awareness earlier in the design process. To this end, Guo et al. [36] introduce AutoBridge, a multi-die aware HLS framework that distributes logic across dice and inserts additional pipeline registers into inter-die paths.

The works above focus on 2.5D FPGAs, where dice lie side by side on a passive interposer. A complementary body of research investigates truly 3D FPGAs, where active dice are stacked directly on top of one another. Early work by Gayasen et al. [37] and Razavi et al. [38] examines how the switch block must be redesigned for 3D integration. Boutros et al. [16] propose a 3D FPGA architecture targeting a 7 nm technology node. In their design, inter-layer connectivity is realized by connecting the output pin of each logic block to a bump that drives intra-die wires on the adjacent layer. Youssef et al. [17] introduce LaZagna, an open-source framework that extends VTR and OpenFPGA [39] to model 3D FPGAs. LaZagna supports inter-layer connectivity through both connection blocks and switch blocks. For switch-block-based inter-layer connections, the authors propose a syntax similar to VPR’s existing custom switch block pattern specification. Their experiments show that 3D architectures with as few as 26 inter-layer connections per logic block remain routable. Cherukuri et al. [40] propose vFPGA, a 3D FPGA in which SRAM configuration bits are stored on a dedicated layer separate from the logic fabric. This decoupling allows the configuration layer to be implemented in an older technology node, which is advantageous because SRAM scaling has slowed considerably at recent process nodes.

$\alpha \backslash P$	45 μm	36 μm	25 μm	10 μm	5 μm	1 μm
0.1	101	75	30	10	1.25	0.05
0.2	202	150	60	20	2.5	0.10
0.3	302	225	90	30	3.75	0.15
0.4	403	300	120	40	5	0.20
0.5	503	375	150	50	6.25	0.25
0.6	604	450	180	60	7.5	0.30
0.7	705	525	210	70	8.75	0.35
0.8	805	600	240	80	10	0.40
0.9	906	675	270	90	11.25	0.45
1	1006	750	300	100	12.5	0.50

TABLE I: Interposer wire length ($\times H_{LB}$) required to achieve inter-die connectivity fraction α for different pitches. Cells are color-coded by RC delay, from lowest (green) to highest (red).

III. INTER-DIE CONNECTION PHYSICAL MODELING

A. Interposer Feasibility Study

Unlike 3D FPGAs, which stack dice to allow vertical connections across their entire surface, 2.5D FPGAs place dice side-by-side, restricting connections to the die boundaries. Because interposer wires have a finite reach (e.g., 2 mm), any bumps placed beyond that distance from the edge are useless for inter-die communication. Consequently, 2.5D inter-die connectivity depends on both bump pitch and interposer wire length, introducing an additional dimension in the design space. This motivates a first-order feasibility study to prune the 2.5D design space before undertaking detailed evaluation.

Consider a hypothetical FPGA with square LB tiles of height H_{LB} , routing channel width W , and bump pitch P . Each LB can accommodate at most $n_b = H_{LB}^2/P^2$ bumps in its footprint. To achieve an inter-die connectivity ratio α (i.e., $\alpha \cdot W$ connections cross the die boundary per LB column), the bumps from $\alpha \cdot W/n_b$ rows of LBs along the boundary must be used, requiring an interposer wire length of $L_{int} = \alpha \cdot W \cdot P^2/H_{LB}^2$. The maximum achievable inter-die connectivity ratio is therefore:

$$\alpha = \frac{L_{int} \cdot H_{LB}^2}{W \cdot P^2} \quad (1)$$

Equation 1 confirms the earlier qualitative argument: α increases with L_{int} and decreases with P , and both parameters must be considered when evaluating inter-die connectivity in interposer-based FPGA architectures.

In this work, we use the 7nm FPGA fabric introduced in [16], with $H_{LB} = 24.6 \mu\text{m}$, and a channel width of $W = 300$. Table I shows the interposer wire length (in units of H_{LB}) required to achieve various inter-die connectivity ratios α for different pitch sizes, as given by Eq. 1.

To estimate the intrinsic delay of interposer wires, we assume the interposer uses a 45nm technology node with inter-die wires implemented on the global metal layers. The electrical and physical characteristics of the global metal layer are extracted from [41]. Using a distributed RC delay model, Fig. 2 plots intrinsic delay against wire length. The highlighted regions indicate the wire length ranges and their associated delay ranges needed to achieve $\alpha \in [0.3, 0.4]$ for different pitch sizes. This range is chosen because early experiments indicated it is the minimum inter-die connectivity ratio that yields reliably routable FPGA devices.

For $P = 45\text{--}36 \mu\text{m}$, achieving $\alpha \in [0.3, 0.4]$ requires interposer wires spanning hundreds of LBs, with an estimated intrinsic wire

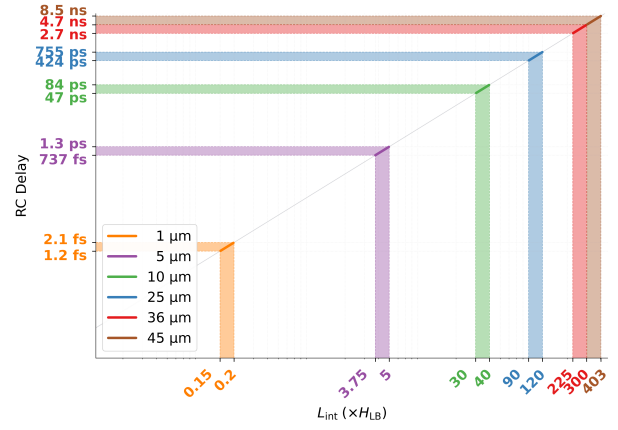


Fig. 2: Distributed RC delay of interposer wires vs. their length, assuming a 45nm process node. Highlighted regions indicate the wire length ranges and associated delay ranges required to achieve inter-die connectivity ratios of $\alpha \in [0.3, 0.4]$.

delay of 2.7–8.5 ns. A delay of this magnitude imposes a significant penalty on the critical path delay and makes high inter-die connectivity extremely costly. Furthermore, the required interposer wire lengths are extremely long, meaning that each wire crossing the die boundary must span hundreds of LBs. This can confuse placement and routing algorithms, as tiles physically adjacent across the inter-die boundary may not be reachable through short local routing wires. For $P = 25 \mu\text{m}$, the required wire lengths and associated delays are substantially reduced compared to the larger pitches, with intrinsic RC delays in the range of hundreds of picoseconds, but the needed interposer wire lengths remain very long. At $P = 10 \mu\text{m}$, a reasonable inter-die connectivity ratio can be achieved with interposer wires spanning only tens of LBs, comparable to intra-die long wires, making this pitch a practical candidate. With $P = 5 \mu\text{m}$ and $P = 1 \mu\text{m}$, $\alpha = 1$ can be reached with short interposer wires and negligible intrinsic delay, suggesting that near-full inter-die connectivity is physically feasible. However, very short interposer wires introduce a different concern: since inter-die routing resources are accessed via intra-die routing wires, concentrating a large number of inter-die connections near the die boundary can cause routing congestion in that region.

The existence of the Xilinx Virtex-7 family, a 2.5D FPGA that uses a bump pitch of $45 \mu\text{m}$, may appear to contradict our analysis. This discrepancy is explained by several differences between the 7nm device we analyze and the 28nm Virtex-7. Most importantly, the Virtex-7 LBs are much larger than those of our 7 nm fabric (we would expect their H_{LB} to be approximately $4\times$ as large) and Equation 1 shows that α has a quadratic dependence on H_{LB} . Second, Virtex-7 uses only approximately 50 die-crossing wires per routing channel rather than the [90, 120] implied by $\alpha \in [0.3, 0.4]$ with $W = 300$; applying our model at 50 connections yields an estimated RC delay of approximately 1.5 ns, which is in the range of the 1ns interposer delay of Virtex-7 [31]. Finally, our estimates assume minimum-pitch interposer wires; widening the wires would reduce resistance and improve delay.

B. Detailed Interconnect Modeling

The wire delays reported in Fig. 2 are intrinsic RC delays of the interposer wire alone and do not account for drivers, multiplexers,

or ESD protection circuitry. For a rigorous exploration of inter-die routing architectures, we need accurate delays based on realistic circuit models. To this end, we perform SPICE simulations of 2.5D and 3D interconnect models to obtain accurate delay and area estimates. The 3D model is derived from [16], while the 2.5D model is an extension we introduce in this work. Fig. 3 shows both circuit models. Each model consists of an input multiplexer, a two-stage driver, a series of distributed RC loads, and ESD protection circuitry [42]. In the 3D model, the RC loads represent the signal path from ① the source die's silicon layer and through its via stack, ② a worst-case distance of half the perimeter of an SRAM macro on top-level metal to reach the inter-die connection, ③ the inter-die connection, ④ the destination die's top-level metal, and ⑤ the destination die's via stack. The 2.5D model extends the 3D model by inserting an interposer wire between the two bumps: after passing through ① the source die's via stack and ② its top-level metal to reach ③ the first bump, the signal traverses ④ the interposer wire, ⑤ the second bump, ⑥ the destination die's top-level metal, and finally ⑦ the destination die's via stack. Interposer wire resistance and capacitance are taken from the calculations in Section III-A, while the active-die metal and via stack parameters are based on a 7 nm node process [43], [44], and the inter-die connection and bump parasitics are adopted from [45].

For each combination of die-crossing technology, input multiplexer size, and interposer wire length (for 2.5D models), we sweep over a range of driver stage ratios and gather mux transistor sizes in HSPICE and select the configuration that minimizes the area $\times$ delay² product. Delay and area are similarly simulated for 3D inter-die connections, though these results are omitted here for brevity. Fig. 4 plots the simulated total delay versus interposer wire length for the 2.5D case. Compared to the intrinsic RC delay of Section III-A, the full circuit delay is substantially higher due to multiplexer, driver, and ESD overhead. Higher fan-in/fan-out increases delay through higher mux and output capacitance. The 10 μm pitch shows slightly higher delay than 5 μm because of the switch from μBumps to the bump-less hybrid bonding technology and the subsequent reduction in resistance and capacitance.

IV. ARCHITECTURAL MODELING AND CAD FLOW FOR MULTI-DIE FPGAS

A. Scatter-Gather Syntax

Having characterized the delay and area of inter-die connections, these parameters must now be incorporated into the CAD flow. To flexibly specify inter-die routing architectures in VTR, we introduce the *scatter-gather* (SG) syntax as an extension to the VTR architecture description XML format [46]. The SG abstraction models an inter-die connection, illustrated in Fig. 5: a *gather* phase aggregates F_g local routing wires from a source switch block through a multiplexer, and a *scatter* phase fans out the multiplexed signal to F_{sc} wires at a destination switch block, which may reside on a different die or a distant location on the same die.

An SG pattern is specified via the `<sg_pattern>` tag, which supports two directionality modes: `unidir` (the inter-die wire is driven by a single non-tristatable driver from one end) and `bidir` (the inter-die wire can be driven by a tri-statable driver at either end). The `<gather>` tag contains one or more `<wireconn>` elements, each selecting a wire type from a source switch block, with `num_conns` setting the fan-in F_g and `side` restricting which switch-block sides contribute. The `<sg_link>` tag specifies the multiplexer type, the inter-die wire segment, and the spatial

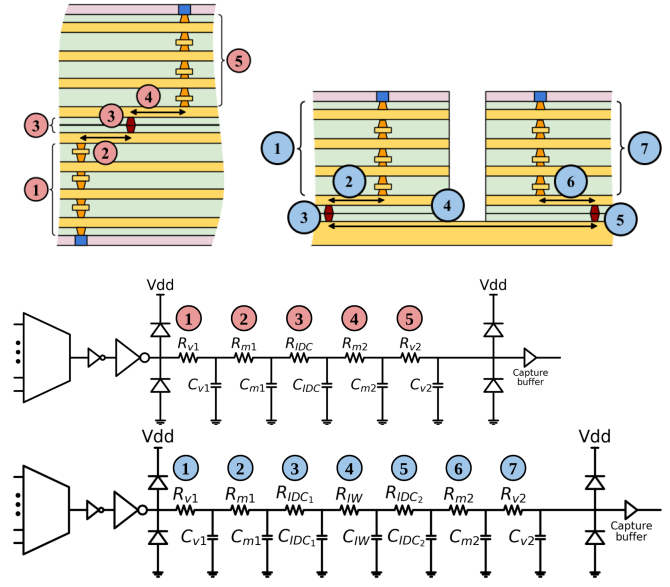


Fig. 3: 2.5D (blue) and 3D (red) inter-die connection circuit models used for detailed interconnect analysis.

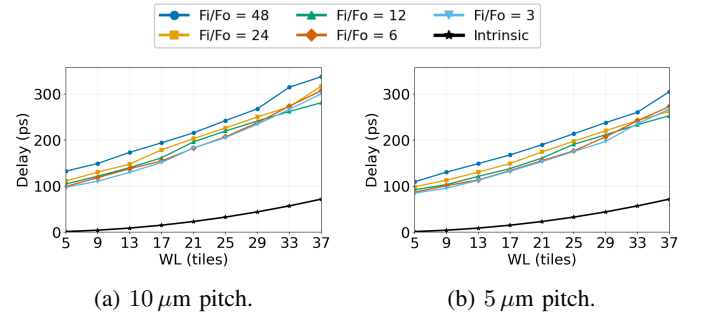


Fig. 4: Total interconnect delay vs. wirelength for varying fan-in/fan-out, with intrinsic Elmore delay reference.

displacement to the destination switch block via spatial offsets. The `<scatter>` tag mirrors the *gather*, distributing the signal to F_{sc} wires at the destination switch block.

Fig. 5 shows an example SG pattern for a 2.5D interposer-based architecture, where the `y_offset` on each `<sg_link>` encodes the interposer wire displacement and length L_{int} . For brevity, only the upward link (`y_offset`=+3) is shown; a downward link can be instantiated identically. We additionally introduce the `<interposer_cut>` tag, which specifies the inter-die boundary location across which there is a discontinuity in the routing fabric and instantiates SG patterns in a user-defined area around it to model inter-die connectivity. Fig. 7 illustrates the resulting routing architecture. The same syntax naturally extends to 3D stacked architectures by using `z_offset` instead of `y_offset` or `x_offset`. Together, the parameters N_{sg} (connections per logic block), F_g/F_{sc} (fan-in/fan-out), and L_{int} (inter-die wire length) fully parameterize inter-die routing architecture design for both 2.5D and 3D FPGAs.


```

<sg_pattern name="2.5D_conn" type="unidir">
  <!-- Gather fan-in=3 from bottom, left, right -->
  <gather>
    <wireconn num_conns="3" from_type="L4"
      from_switchpoint="0" side="blr"/>
  </gather>
  <!-- Scatter fan-out=4 to all sides -->
  <scatter>
    <wireconn num_conns="4" to_type="L4"
      to_switchpoint="0" side="rltb"/>
  </scatter>
  <sg_link_list>
    <!-- Cross interposer cut (y_offset=+3, L_int=3) -->
    <sg_link name="UP" y_offset="3" mux="2.5D_SB_MUX"
      seg_type="int_wire"/>
  </sg_link_list>
</sg_pattern>

<!-- Horizontal cut; two equally sized dies -->
<interposer_cut y="H/2">
  <!-- Instantiate N_sg=2 UP links -->
  <interdie_wire sg_name="2.5D_conn" sg_link="UP" num="2"
    offset_start="-1" offset_end="-2"/>
</interposer_cut>

```

Fig. 5: SG XML syntax for 2.5D interposer-based FPGAs.

B. Multi-die Aware FPGA CAD Flow

1) *Routing Resource Graph Construction*: Each SG connection is realized in the routing resource (RR) graph as a new node with F_g incoming edges from local routing wires at the source switch block and F_{sc} outgoing edges fanning out at the destination. SG nodes are appended as a post-processing step, keeping the implementation modular and independent of base graph construction.

For 3D stacked architectures, we introduce a new RR node type, CHANZ, characterized by a *layer range* encoding the range of stacked layers it connects. This is more explicit and expressive than the approaches taken by prior work. Boutros et al. [16] connected OPIN nodes directly to CHANX/CHANY nodes on the adjacent layer, limiting inter-layer connections to those driven solely by output pins and ruling out bidirectional links. VTR 9 introduced 3D switch blocks and models inter-layer connections as two CHANX nodes joined by a single edge [47]. While this allows switch block connections between layers, it misleads the router lookahead into treating vertical die crossings as horizontal displacements, and the 3D switch block specification does not allow detailed control over the multiplexing to and from the scarce 3D interconnect. Lazagna [17] augments the VTR 9 code base with an external RR graph generator to insert 3D connections, and hence inherits the same restrictions. The CHANZ node, combined with the new SG syntax, eliminates all of these limitations.

In interposer-based 2.5D FPGAs, intra-die wires must not cross the die boundary defined by the `<interposer_cut>` tag. Prior works used an RR graph post-processing step to detect and shorten straddling wires [31], reducing connectivity near the boundary. Our approach instead enforces the constraint during RR graph generation: any wire that would cross the `<interposer_cut>` boundary is split into two shorter wires, one on each side.

2) *Router Lookahead*: VPR uses an A^* -based router that requires a heuristic cost estimate from each RR graph node to the sink. Recent VPR versions [47] compute this estimate via a *lookahead table* [48]: for each wire type and each $(|\Delta x|, |\Delta y|)$ offset, the table stores the minimum expected resource cost and delay to reach a node that far away. The table is filled by sampling routes from many starting wires to many target distances. For a

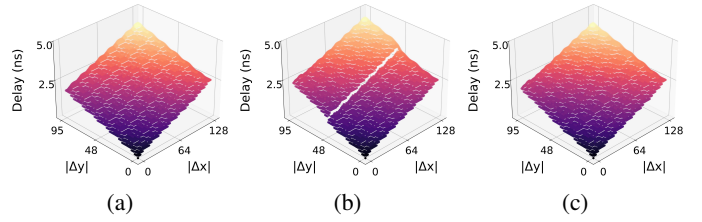


Fig. 6: (a) Default router lookahead for a 2D architecture. (b) Default router lookahead for the same architecture but with an interposer cut in the middle. (c) Our modified router lookahead for the 2.5D architecture.

2D device with a uniform routing fabric, the resulting table is *translation-invariant*: the estimated cost of moving $(|\Delta x|, |\Delta y|)$ is independent of the starting position, as illustrated in Fig. 6a.

In contrast, Nasiri et al. [31] used a simpler, formula-based lookahead suited to routing architectures with a single wire type: it computed the (x, y) distance from the current node to the sink, divided each component by the wire’s length, and multiplied by the average delay per wire traversal. A fixed die-crossing penalty was added when the current node and sink resided on different dice, yielding a $31\times$ routing speedup. This approach is insufficient for modern architectures, which mix short, medium, and long wire segments with complex inter-wire connectivity patterns, necessitating the table-based lookahead described above.

VPR’s lookahead table cannot be used directly with 2.5D architectures either, however. Sampling does not yield a translation-invariant table in 2.5D devices because interposer wires introduce additional delay whenever a sample route crosses a die boundary, causing some table entries to be inflated. The result is a visible discontinuity at the interposer cut, as shown in Fig. 6b for a device with an interposer cut at $y = 48$. This breaks translation invariance and degrades both routing quality and speed.

We solve this problem by decoupling the heuristic into two components: a *translation-invariant intra-die lookahead* and a *die-crossing cost table*. To build the intra-die lookahead, we modify the sampling procedure by setting the cost of all interposer wires to zero during sampling, preventing die-crossing delays from contaminating the table. As shown in Fig. 6c, the resulting table closely matches the regular structure of a 2D lookahead. For the die-crossing component, we compute, for each interposer cut, the minimum delay among all inter-die wire types that cross it, and store these minima in the die-crossing cost table. During routing, we add the die-crossing penalty for each cut that lies between the current node and the sink to the intra-die lookahead value. This decomposition resulted in a $2.5\times$ reduction in routing time vs. the default VPR lookahead with no reduction in result quality.

3) *Placement Cost Function*: VPR’s placer uses simulated annealing to minimize a weighted sum of timing and wirelength costs. Wirelength is approximated by the half-perimeter wirelength (HPWL) of each net’s bounding box. The timing cost is computed using a placement delay model, which is tied to the router lookahead; making the lookahead multi-die aware therefore makes the timing cost multi-die aware as well.

For interposer-based architectures, we include a *cut* cost as in prior work on silicon-interposer FPGAs [31], [49]. Each net is penalized when its bounding box crosses an inter-die boundary, with the penalty growing linearly with the span of that box perpendicular to the boundary. In our early experiments, this term

reduced the total number of nets crossing a die boundary, but did not encode *where* along the boundary demand is concentrated. We still observed localized hotspots that caused routing failures. Ravishankar et al. [28] report the same issue and address it by inserting placeable inter-die wire driver instances into the netlist for each die-crossing net; the partition-driven placer then treats inter-die wires as capacity bins, guaranteeing no inter-die channel is oversubscribed. Instead of restructuring our annealing-based flow around a partition-driven approach, we adopt the congestion-aware placement cost of [50], which estimates channel demand via a wirelength per area model and penalizes nets whose average demand-to-capacity ratio within their bounding box exceeds a threshold. We restrict this congestion term to inter-die routing channels only, so the annealer directly discourages hotspot formation at die boundaries without interfering with intra-die placement.

V. EXPERIMENTAL RESULTS

This section evaluates the impact of key inter-die routing architectural parameters including the fan-in/fan-out (F_g and F_{sc}) of inter-die connections and the number of such connections per logic block (N_{sg}) on routability, critical path delay (CPD), routed wirelength (WL), and area in 3D/2.5D multi-die 7nm FPGAs.

Each die uses the 7nm FPGA architecture from [16] (10 fracturable 6-LUTs per LB, $W = 300$, length-4/16 wires). Inter-die connectivity is modeled using the SG syntax from Section IV-A, parameterized by varying N_{sg} , fan-in/fan-out (F_g/F_{sc}), and delays from Section III-B. Benchmarks are drawn from the Koios benchmark suite [51]. For each circuit, the FPGA fabric is sized to the smallest square grid that fits the circuit, stressing inter-die routing resources.

As shown in Section III-A, sufficient 2.5D inter-die connectivity requires a minimum interposer wire length L_{int} , which sets a minimum device size of $2L_{int} \times 2L_{int}$ tiles. Our maximum evaluated $L_{int} = 37 H_{LB}$ excludes circuits smaller than this threshold; the Koios subset we use spans 12k to 759k primitives and maps to 78×78 to 335×335 tile devices.

A. 2.5D Homogeneous Integration

In this section, we evaluate a family of 2.5D architectures consisting of two dice placed side by side, separated by a horizontal interposer cut that creates a discontinuity along the column direction; Fig. 7 illustrates an example 2.5D routing architecture. Based on the feasibility study of Section III-A, we evaluate pitch sizes of $10\mu\text{m}$ and $5\mu\text{m}$ for inter-die connections, with varying interposer wire lengths and driver fan-in. Table II summarizes the results of our 2.5D architecture exploration.

1) *Interposer Wire Length*: For a fixed bump pitch, increasing interposer wire length (L_{int}) raises the inter-die connectivity ratio α at the cost of higher wire delay. Using an architecture with a bump pitch of $10\mu\text{m}$ and interposer wire fan-in of 48, we test $L_{int} \in \{5, 9, 13, 17, 21, 25, 29, 33, 37\}$. The shortest L_{int} that achieves routing success across all benchmarks is $L_{int} = 17$, corresponding to $\alpha = 0.32$. This is consistent with results from [31] showing that routability is not severely impacted when $\alpha > 0.25$. For routable architectures, increasing L_{int} improves α but directly degrades CPD due to higher wire delay. Beyond CPD, longer L_{int} also increases WL and route time, which we attribute to the behavior of placement and routing algorithms. The placement cost function minimizes HPWL by placing connected netlist primitives close together; however, when two connected blocks are on opposite sides of an interposer cut, placing them near

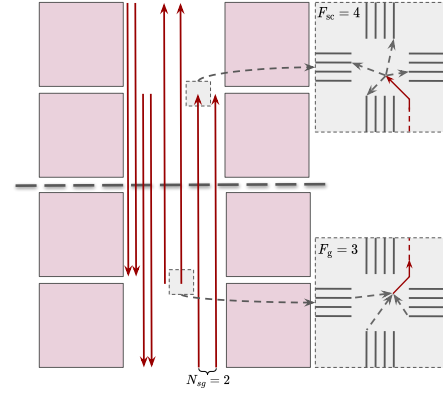


Fig. 7: 2.5D FPGA routing architecture ($N_{sg} = 2$, $F_g = 3$, $F_{sc} = 4$, $L_{int} = 3$). Zoomed-in sections show fan-in/fan-out between a single interposer wire and a switch-block location.

Pitch (N_{sg})	F_g/F_{sc}	L_{int} (LBs)	α	WL	CPD	Route Time	LB Area
2D	N.A.	N.A.	N.A.	1	1	1	1
$10\mu\text{m}$ (3) § V-A1	48/64	5	0.08	D.N.F.	D.N.F.	D.N.F.	1.041
	48/64	9	0.16	D.N.F.	D.N.F.	D.N.F.	1.042
	48/64	13	0.24	D.N.F.	D.N.F.	D.N.F.	1.041
	48/64	17	0.32	1.07	1.05	3.02	1.041
	48/64	21	0.40	1.07	1.06	5.73	1.042
	48/64	25	0.48	1.10	1.07	10.00	1.041
	48/64	29	0.56	1.14	1.08	18.30	1.041
	48/64	33	0.64	1.19	1.10	32.28	1.036
	48/64	37	0.72	1.21	1.11	48.09	1.036
$10\mu\text{m}$ (3) § V-A2	3/4	17	0.32	D.N.F.	D.N.F.	D.N.F.	1.016
	6/8	17	0.32	1.12	1.06	4.12	1.021
	12/16	17	0.32	1.07	1.05	3.28	1.024
	24/32	17	0.32	1.06	1.05	3.09	1.029
	48/64	17	0.32	1.07	1.05	3.04	1.041
$5\mu\text{m}$ (4) § V-A3 (3) (2)	12/16	5	0.32	D.N.F.	D.N.F.	D.N.F.	1.048
	12/16	9	0.32	1.02	1.04	1.62	1.025
	12/16	13	0.32	1.04	1.04	2.04	1.012
	12/16	17	0.32	1.06	1.05	3.18	1.008
	12/16	25	0.32	1.11	1.08	9.81	1.004
	12/16	37	0.32	1.11	1.08	9.81	1.004

TABLE II: Comprehensive study of interposer wire lengths, F_g/F_{sc} ratios, and pitch configurations. All results are the geometric mean of the benchmark suite normalized to the 2D baseline. Entries denoted as D.N.F. (did not finish) indicate architectures that were not able to route all benchmark circuits.

the boundary does not help, as they cannot be connected through short local wires and must instead use the long interposer wire spanning L_{int} tiles. On the routing side, VPR's router uses the bounding-box heuristic [48] to prune the RR graph search to a slightly expanded bounding box of each net. Long interposer wires can extend beyond this bounding box, causing search failures that force the router to repeatedly enlarge it and increase runtime. These observations highlight that placement and routing algorithms have significant room for improvement for 2.5D devices, beyond the multi-die aware enhancements introduced in Section IV-B.

2) *Interposer Wire Fan-in*: Since increasing L_{int} beyond the minimum required for reliable routability only worsens CPD, we select $L_{int} = 17$ and sweep the interposer wire fan-in, testing $F_g \in \{48, 24, 12, 6, 3\}$. In this architecture, the fan-in wires are gathered from the sides of each switch block that face away from the interposer cut, and F_{sc} is set to $\frac{4}{3}F_g$ at the destination.

At $F_g = 3$, each inter-die connection aggregates only 3 intra-die

wires, making it difficult for the router to find an interposer wire to cross dice with, resulting in unroutable circuits. Increasing F_g improves routability and reduces route time, as wider muxes make inter-die routing resources more accessible. Area overhead grows substantially with fan-in: increasing F_g from 3 to 48 more than doubles the LB area overhead due to significant multiplexer area. CPD and WL both decrease with larger F_g , as wider muxes reduce intra-die routing detours, though gains diminish beyond $F_g = 12$. Fig. 8 shows how the area-delay product of a 2.5D architecture with 10 μm bumps and an interposer wire length of 17 tiles varies with F_g . The best values of F_g are in the 8 to 32 range, which balance routability with the area and delay of the added mux.

3) *Inter-die Connection Density*: As shown in Table I, achieving full inter-die connectivity ($\alpha = 1$) is feasible with fine-pitch hybrid bonding. For a 5 μm pitch, $\alpha = 1$ requires $L_{int} = 13$, meaning that only 12 LB rows on each side of the boundary need to accommodate the interposer wire circuitry. However, since FPGAs are manufactured by tiling the same LB across the entire device, a column structure must be maintained: even if only a fraction of the tiles in a column require wider layout to host the additional driving circuitry and ESD protection, all tiles in that column must match that width. As a result, the area overhead extends to the full column height, not just the rows directly involved in inter-die connectivity.

This observation motivates using longer interposer wires to spread the interposer wire circuitry across a larger number of LB rows, reducing the per-tile area overhead by scattering buffers, drivers, and ESD circuitry over a larger region rather than concentrating them near the die boundary. To evaluate this trade-off, we fix the 5 μm pitch and hold $\alpha = 0.32$ while increasing L_{int} from 5 to 25. The results are shown in the bottom section of Table II. Note that $L_{int} = 5$ itself is not reliably routable: inter-die wires must be reached through intra-die routing resources, and concentrating all of them in a narrow band immediately adjacent to the boundary creates local routing congestion in that region. Longer interposer wires distribute this demand over a wider area, alleviating the congestion. Tile area decreases by 4.2% across this range, confirming that spreading the circuitry over more rows meaningfully reduces per-tile overhead. Delay increases for interposer wires longer than 13 LBs due to their higher wire delay, and the best area-delay trade-off occurs with L_{int} in the 13 to 17 range. As previously discussed, route time increases with long interposer wires, motivating future work in interposer-aware routing algorithms.

B. 3D-stacked Homogeneous Integration

The 3D FPGA architecture evaluated in this study comprises two homogeneous dice stacked atop each other. For inter-layer connectivity, we consider three die stacking technologies: 1 μm and 5 μm pitch hybrid bonds, and 10 μm μ bumps. There are multiple possible routing architectures to leverage these inter-die links. One option is to cross between dice only at output pins, as in [16]; we call an architecture where all block output pins can reach the CHANZ nodes that model die-crossing a *full output pin connectivity* architecture. Another possibility is to have routing wires on one layer drive the bumps (CHANZ nodes) that reach routing wires on a different die layer; we call this option *switch-block connectivity*. The choice between full output pin or switch-block connectivity depends on the available inter-layer connection density. For the 5 μm and 10 μm pitches, where the inter-layer connection density is insufficient to support full output pin connectivity, the inter-layer interface is realized by N_{sg} vertical connections per direction;

Pitch (N_{sg})	F_g/F_{sc}	3D Nets	WL	CPD	Route Time	LB Area	ESD Area
2D	N.A.	N.A.	1.00	1.00	1.00	N.A.	N.A.
1 μm (40)	1/64	1.00	0.90	0.95	0.92	1.104	6.29%
5 μm (12)	64/64	0.84	0.86	0.96	1.07	1.091	1.91%
	32/32	0.83	0.87	0.96	0.99	1.068	1.95%
	16/16	0.82	0.90	0.96	1.00	1.051	1.98%
	8/8	0.82	0.92	0.95	1.04	1.043	2.00%
	4/4	0.83	0.93	0.96	1.11	1.034	2.01%
10 μm (3)	64/64	0.61	0.89	0.96	0.92	1.023	0.51%
	32/32	0.61	0.90	0.95	0.94	1.017	0.51%
	16/16	0.60	0.92	0.95	0.98	1.013	0.51%
	8/8	0.61	0.93	0.96	1.01	1.011	0.51%
	4/4	0.61	0.94	0.97	1.07	1.008	0.52%
10 μm (1)	64/64	0.54	0.91	0.95	1.00	1.008	0.17%
	32/32	0.55	0.92	0.94	1.00	1.006	0.17%
	16/16	0.54	0.93	0.95	1.07	1.004	0.17%
	8/8	0.54	0.95	0.98	1.11	1.004	0.17%
	4/4	0.54	0.96	0.98	1.19	1.003	0.17%

TABLE III: Impact of pitch, N_{sg} , F_g and F_{sc} on QoR and area. All QoR metrics are normalized to the 2D baseline. LB Area is reported with ESD included; the ESD Area column lists the percentage of LB area due to ESD.

here, each vertical connection is driven by a multiplexer aggregating F_g length-4 wires from all four cardinal sides of the source switch block. Driving CHANZ nodes from both x- and y-directed routing wires via the gather multiplexer maximizes use of the scarce die-crossing bumps, as each bump can be driven by any of several candidate wires and signals can cross dice anywhere along a net's path. At the 1 μm pitch, the inter-layer connection density suffices for full output pin connectivity: LB output pins drive CHANZ nodes directly ($F_g = 1$) without a gather multiplexer. In both full output pin and switch-block connectivity, upon reaching the adjacent die, the inter-layer signal fans out into F_{sc} length-4 wires, which are distributed across the four sides of the destination switch block. Length-16 wires and input pins do not connect to the inter-layer interconnects directly.

For the 5 μm and 10 μm pitches, we integrate the maximum possible number of vertical resources, yielding N_{sg} values of 12 and 3 connections per direction per LB, respectively. For the 1 μm pitch, rather than maximizing the number of connections (which would yield up to $N_{sg} = 302$), we set $N_{sg} = 40$ so that every output pin can cross to the other die via its own dedicated inter-layer connection. To investigate the threshold of routability and the impact of inter-layer routing resource scarcity, we further evaluate a constrained 10 μm case where N_{sg} is reduced to 1. Furthermore, we modeled inter-layer connectivity with a 25 μm μ bump pitch, which translates to a density of $N_{sg} = 0.5$ connections per direction per LB. In this configuration, vertical connections are staggered such that the connection direction alternates at each LB. More than one-third of the benchmark circuits failed to route with a 25 μm pitch, suggesting that this inter-layer connection density falls well below the threshold of routability. Across the remaining viable combinations of pitch and N_{sg} , we sweep inter-layer connection fan-in and fan-out, testing $F_g, F_{sc} \in \{64, 32, 16, 8, 4\}$. Results are reported in Table III.

1) *Routability under Scarce Inter-layer Connectivity*: A key finding is that 3D homogeneous integration does not require the most advanced and expensive packaging technologies to achieve routability. Prior work assumed high inter-layer connectivity: Boutros et al. [16] used full output-pin connectivity ($N_{sg} = 40$), and Lazagna [17] demonstrated routability with $N_{sg} \approx 26$ using

3D switch blocks. In both cases, the required connection density can only be realized with a $1\mu\text{m}$ hybrid-bond pitch. Our results show that circuits remain routable down to $N_{sg} = 1$, which is achievable with $10\mu\text{m}$ μbumps , a more cost-effective packaging technology. Beyond reducing manufacturing cost, lower inter-layer connectivity also shrinks the per-tile area overhead of the multiplexers, buffers, and ESD protection needed to implement the inter-layer interface, further improving the overall area efficiency of the 3D FPGA fabric. Achieving routability under scarce inter-layer connectivity, however, does not depend on interconnect density alone; placement must also be aware of how limited these resources are. This is handled through the HPWL cost, which scales displacement along each dimension by the inverse of available routing resources [52]. As N_{sg} decreases, the relative cost of crossing to another die rises, encouraging the placer to keep net endpoints within a single die. This is reflected in the results: the number of inter-layer nets is sensitive to N_{sg} but largely independent of F_g and F_{sc} , since F_g and F_{sc} affect routing flexibility rather than resource density.

2) *QoR and Route Time Impact*: Reduced inter-layer connectivity imposes a measurable cost on both QoR and route time. As N_{sg} decreases, the router faces a scarcer pool of die-crossing routing resources, increasing the likelihood of congestion at die boundaries. When local inter-layer connections are occupied, the router must detour to find available CHANZ nodes further away, driving up both WL and route time. CPD is also affected, though less severely, as the timing-driven router reserves local die-crossing connections for timing-critical nets and reroutes non-critical nets along longer paths. Inter-layer connection density alone, however, does not fully determine QoR, as the accessibility of those connections matters equally. As F_g or F_{sc} decreases, each inter-layer connection aggregates fewer intra-die wires, making it harder for the router to find a suitable die-crossing resource for a given net. This reduced accessibility leads to longer routing detours, increasing WL, CPD, and route time even when the total number of die-crossing connections remains fixed. The most accessible configuration is full output-pin connectivity ($1\mu\text{m}$ pitch, $F_g = 1$), where every output pin has its own dedicated inter-layer wire and can immediately cross to the adjacent die without competing for a shared multiplexer. This yields the best route time and CPD among all evaluated architectures, with 23% faster routing and 3% lower CPD compared to the most resource-scarce configuration ($10\mu\text{m}$, $N_{sg} = 1$, $F_g = F_{sc} = 4$).

3) *Area-Delay Trade-off*: Tile area moves in the opposite direction to WL and CPD, revealing a fundamental trade-off. Reducing N_{sg} not only enables 3D integration with less advanced packaging technologies, as discussed above, but also lowers the silicon area overhead: fewer inter-layer connections mean fewer buffers, multiplexers, and ESD protection circuits per tile. Reducing F_g , which makes connections less accessible by narrowing the input multiplexers, has the same effect, as smaller muxes occupy less area. Thus, the architectural choices that improve QoR and reduce routing runtime (higher N_{sg} , larger F_g) come at the cost of increased tile area, while leaner configurations trade some QoR for a smaller silicon footprint. The area cost of larger F_g is more pronounced in architectures with finer bump pitches and higher N_{sg} . Fig. 8 shows the area-delay product of the evaluated 3D architectures. The $10\mu\text{m}$ architecture with $N_{sg} = 1$ achieves the best area-delay trade-off among all evaluated architectures, meaning that the area savings in this architecture outweigh the resulting CPD degradation.

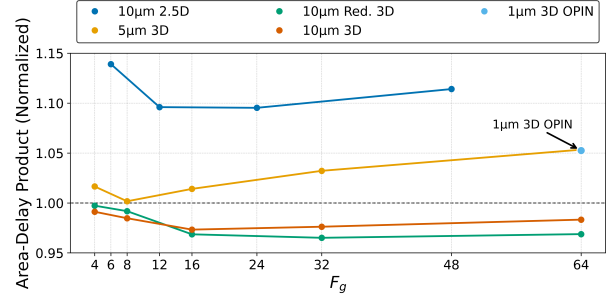


Fig. 8: Area-delay products of the explored multi-die architectures.

Recent packaging technology advancements have demonstrated that ESD protection circuitry can be simplified or even eliminated [53], [54]. As Table III shows, the area saved by removing ESD protection is most significant in architectures with a high density of inter-layer connections. For the $1\mu\text{m}$ pitch ($N_{sg} = 40$), removing ESD yields an area saving of 6.29%, whereas these savings diminish as vertical connections become sparser, reaching as low as 0.17% for the $10\mu\text{m}$ pitch ($N_{sg} = 1$) cases.

VI. CONCLUSION

Advanced packaging enables higher-capacity multi-die FPGAs, but inter-die routing must be tailored to each technology's constraints, motivating versatile CAD modeling tools. To address this need, we presented detailed circuit models of inter-die connections in 2.5D and 3D FPGAs, enabling accurate delay and area estimates for a range of die-crossing technologies and fan-in/fan-out configurations. We also introduced a first-order feasibility analysis for 2.5D architectures that links bump pitch, interposer wire length, and inter-die connectivity ratio, enabling rapid pruning of the design space before detailed simulation. We also extended the VTR architecture description language with the scatter-gather (SG) syntax, modeling inter-die connections with independent control over fan-in, fan-out, wire length, and connectivity pattern.

Using these tools, we conducted a detailed design space exploration of 3D and 2.5D FPGA routing architectures. For 3D FPGAs, we show that homogeneous die stacking does not require the most advanced packaging technologies to achieve reliable routability. With $10\mu\text{m}$ μbump technology, 3D FPGAs remain routable even with as few as two inter-layer connections per logic block using a *switch block connectivity* 3D architecture. High-density $1\mu\text{m}$ hybrid bonding enables a *full output-pin connectivity* architecture, and 3D FPGAs achieve up to 10% wirelength reduction and 5% CPD improvement. For 2.5D FPGAs, we find that approximately 32% inter-die connectivity is sufficient for reliable routability, incurring only a 2% wirelength and 4% CPD overhead relative to a 2D baseline. The inter-die architecture specification syntax and CAD optimizations introduced in this work are open source and integrated with the VTR master branch¹, enabling future enhancements and work in this area.

ACKNOWLEDGEMENTS

The authors thank Altera, NSERC, and QuickLogic for research funding and Amin Mohaghegh for insightful technical discussions.

¹github.com/verilog-to-routing/vtr-verilog-to-routing

REFERENCES

- [1] A. Boutros, A. Arora, and V. Betz, "Field-programmable gate array architecture for deep learning: Survey and future directions," *Proceedings of the IEEE*, 2025.
- [2] F. Elsabbagh, S. Sheikhha, V. A. Ying, Q. M. Nguyen, J. S. Emer, and D. Sanchez, "Accelerating RTL simulation with hardware-software co-design," in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, 2023, pp. 153–166.
- [3] E. Perdomo, A. Kropotov, F. K. Cano Ladino, S. Zafar, T. Cervero, X. M. Bofill, and B. Salami, "Makinote: An FPGA-based HW/SW platform for pre-silicon emulation of RISC-V designs," in *Proceedings of the 16th Workshop on Rapid Simulation and Performance Evaluation for Design*, 2024, pp. 29–34.
- [4] H. Jeong, W. Lee, E. Baek, C. Kim, C. Jo, D. Chae, K. Choi, H. Jang, M. Elgammal, S. Hong *et al.*, "MangoBoost alice: Extremely fast, seamless, and versatile FPGA-accelerated DPU solutions," *IEEE Micro*, vol. 45, no. 5, pp. 43–55, 2025.
- [5] N. Tarafdar, N. Eskandari, V. Sharma, C. Lo, and P. Chow, "Galapagos: A full stack approach to FPGA integration in the cloud," *IEEE Micro*, vol. 38, no. 6, pp. 18–24, 2018.
- [6] W. F. Samayoa, M. L. Crespo, A. Cicuttin, and S. Carrato, "A survey on FPGA-based heterogeneous clusters architectures," *IEEE Access*, vol. 11, pp. 67 679–67 706, 2023.
- [7] G. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.
- [8] J. Wang, R. Tao, J. Mai, X. Jiang, S. Wang, C. Yang, H. Jie, K. Huang, R. Y. Sun, and Y. Lin, "TDM signal grouping and package pin assignment for 2.5 d multi-FPGA systems with lookahead placement," in *Proceedings of the 2026 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2026, pp. 201–211.
- [9] Y. Feng and K. Ma, "Chiplet actuary: A quantitative cost model and multi-chiplet architecture exploration," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 121–126.
- [10] S. Li, M.-S. Lin, W.-C. Chen, and C.-C. Tsai, "High-bandwidth chiplet interconnects for advanced packaging technologies in AI/ML applications: Challenges and solutions," *IEEE Open Journal of the Solid-State Circuits Society*, vol. 4, pp. 351–364, 2024.
- [11] TSMC, "Xilinx and TSMC reach volume production on all 28nm CoWoS-based all programmable 3D IC families," TSMC Press Release, Oct. 2013, accessed: 2025-04-03. [Online]. Available: <https://pr.tsmc.com/english/news/1792>
- [12] R. Chaware, K. Nagarajan, and S. Ramalingam, "Assembly and reliability challenges in 3D integration of 28nm FPGA die on a large high density 65nm passive interposer," in *2012 IEEE 62nd Electronic Components and Technology Conference*. IEEE, 2012, pp. 279–283.
- [13] A. Boutros and V. Betz, "FPGA architecture: Principles and progression," *IEEE Circuits and Systems Magazine*, vol. 21, no. 2, pp. 4–29, 2021.
- [14] B. Gaide, D. Gaitonde, C. Ravishankar, and T. Bauer, "Xilinx adaptive compute acceleration platform: Versal™ architecture," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '19, 2019, p. 84–93.
- [15] J. Chromczak, M. Wheeler, C. Chiasson, D. How, M. Langhammer, T. Vanderhoek, G. Zgheib, and I. Ganusov, "Architectural enhancements in Intel® Agilex™ FPGAs," in *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '20, 2020, p. 140–149.
- [16] A. Boutros, F. Mahmoudi, A. Mohaghegh, S. More, and V. Betz, "Into the third dimension: Architecture exploration tools for 3D reconfigurable acceleration devices," in *2023 International Conference on Field Programmable Technology (ICFPT)*, 2023, pp. 198–208.
- [17] I. Youssef, H. Yang, and C. Hao, "LaZagna: An open-source framework for flexible 3D FPGA architectural exploration," in *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2025, pp. 1–9.
- [18] F. Waqar, J. Zhang, A. Lu, Z. He, J. Cong, and S. Yu, "Monolithic 3d fpga design and synthesis with back-end-of-line configuration memories," in *Proceedings of the 62nd Annual ACM/IEEE Design Automation Conference*, ser. DAC '25. IEEE Press, 2025.
- [19] J. H. Lau, "Recent advances and trends in advanced packaging," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 12, no. 2, pp. 228–252, 2022.
- [20] D. B. Ingerly, S. Amin, L. Aryasomayajula, A. Balankutty, D. Borst, A. Chandra, K. Cheemalapati, C. S. Cook, R. Criss, K. Enamul, W. Gomes, D. Jones, K. C. Kolluru, A. Kandas, G.-S. Kim, H. Ma, D. Pantuso, C. Petersburg, M. Phen-givoni, A. M. Pillai, A. Sairam, P. Shekhar, P. Sinha, P. Stover, A. Telang, and Z. Zell, "Foveros: 3D integration and the use of face-to-face chip stacking for logic devices," in *2019 IEEE International Electron Devices Meeting (IEDM)*, 2019, pp. 19.6.1–19.6.4.
- [21] B. Banijamali, S. Ramalingam, H. Liu, and M. Kim, "Outstanding and innovative reliability study of 3D TSV interposer and fine pitch solder micro-bumps," in *2012 IEEE 62nd Electronic Components and Technology Conference*, 2012, pp. 309–314.
- [22] C. Peltó, R. Aggarwal, R. Ahan, M. Armstrong, M. Bebek, M. Blount, S. Chowdhury, J. Chuah, C. Connor, T. DeBonis, B. Dhayal, A. Dougless, S. Gokhale, A. Jain, V. Javvaji, K. Kamisetty, G. Kim, J. Kpetchoto, C. Kuan, C. Lin, G. Liu, Y. Ma, G. Mcpherson, S. Mokler, C. Perini, R. Ramaswamy, B. Sell, R. Subramaniam, J. Waldemer, D. Wei, Y. Yang, Y. Yang, J. Yaung, B. Sabi, and S. Natarajan, "Integration of Si-interposer and high density MIM capacitor on 2.5D Foveros face-to-face architecture," in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, 2024, pp. 1–2.
- [23] C.-H. Lee, Y. Hu, S. Chen, C. Lai, M. Liu, H. Chen, J. Lin, M. Yew, C. Hsu, M. Chiu, C. Chen, M. Chou, W. Chen, J. Chang, C. Hsieh, C. Chen, H. Chen, C. Wang, K. Yan, S.-P. Jeng, and J. He, "Next generation large size high interconnect density CoWoS-R package," in *2024 IEEE 74th Electronic Components and Technology Conference (ECTC)*, 2024, pp. 259–263.
- [24] IEEE International Roadmap for Devices and Systems (IRDS), "International roadmap for devices and systems (IRDS) 2024 edition: More moore tables," IEEE, Roadmap Report, 2024, accessed: 2026-03-25. [Online]. Available: <https://irds.ieee.org/editions/irds2024/>
- [25] H.-J. Chia, S.-P. Tai, J. J. Cui, C.-T. Wang, C.-H. Tung, K.-C. Yee, and D. C. Yu, "Ultra high density low temperature SoIC with sub-0.5 μ m bond pitch," in *2023 IEEE 73rd Electronic Components and Technology Conference (ECTC)*, 2023, pp. 1–4.
- [26] Y.-J. Mii, "Technology leadership," Presentation at the TSMC 2025 North America Technology Symposium, Taiwan Semiconductor Manufacturing Company (TSMC), Apr. 2025, available at <https://www.tsmc.com/english/node/223>.
- [27] Intel Corporation, "Foveros Direct 3D revolutionizes chip integration for AI and HPC," Intel Foundry, Technology Brief, 11 2025, accessed: March 26, 2026. [Online]. Available: <https://www.intel.com/content/dam/www/central-libraries/us/en/documents/2025-11/foveros-direct-3d-tech-brief.pdf>
- [28] C. Ravishankar, D. Gaitonde, and T. Bauer, "Placement strategies for 2.5D FPGA fabric architectures," in *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, 2018, pp. 16–164.
- [29] R. Raikar and D. Stroobandt, "Routing in 2.5D FPGAs: How long should interposer lines be?" in *Proceedings of the 2024 ACM International Workshop on System-Level Interconnect Pathfinding*, ser. SLIP '24, 2025.
- [30] I. Bolsens, "2.5D ICs: Just a stepping stone or a long term alternative to 3D?" Xilinx CTO Presentation, Xilinx, Inc., 2011, accessed: March 25, 2026. [Online]. Available: <https://www.xilinx.com/publications/about/3-D-Architectures.pdf>
- [31] E. Nasiri, J. Shaikh, A. Hahn Pereira, and V. Betz, "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, 2016.
- [32] M. A. Iyer, A. B. Kahng, J. Luu, B. Pramanik, K. Vorwerk, and G. Zgheib, "A partitioning-based CAD flow for interposer-based multi-die FPGAs," in *2025 IEEE 33rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2025, pp. 172–180.
- [33] R. Raikar and D. Stroobandt, "LiquidMD: Optimizing inter-die and intra-die placement for 2.5D FPGA architectures," in *Proceedings of the 14th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies*, ser. HEART '24, 2024, p. 90–98.
- [34] D. Vercruyce, E. Vansteenkiste, and D. Stroobandt, "Liquid: High quality scalable placement for large heterogeneous FPGAs," in *2017 International Conference on Field Programmable Technology (ICFPT)*, 2017, pp. 17–24.
- [35] Z. Di, R. Tao, J. Mai, L. Chen, and Y. Lin, "LEAPS: Topological-layout-adaptable multi-die FPGA placement for super long line minimization,"

- IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 3, pp. 1259–1272, 2024.
- [36] L. Guo, Y. Chi, J. Wang, J. Lau, W. Qiao, E. Ustun, Z. Zhang, and J. Cong, “AutoBridge: Coupling coarse-grained floorplanning and pipelining for high-frequency HLS design on multi-die FPGAs,” in *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA ’21, 2021, p. 81–92.
- [37] A. Gayasen, V. Narayanan, M. Kandemir, and A. Rahman, “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.
- [38] S. A. Razavi, M. S. Zamani, and K. Bazargan, “A tileable switch module architecture for homogeneous 3d fpgas,” in *2009 IEEE International Conference on 3D System Integration*, 2009, pp. 1–4.
- [39] X. Tang, E. Giacomini, A. Alacchi, B. Chauviere, and P.-E. Gaillardon, “Openfpga: An opensource framework enabling rapid prototyping of customizable fpgas,” in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019, pp. 367–374.
- [40] N. K. Cherukuri, S. Nag, P. S. Nalla, A. K. Kola, C. S. Gadireddi, K. Dai, J.-s. Seo, Z. Fang, J. Zhang, and Y. Cao, “vfpga: Towards sub- μ s reconfiguration via 3d fpga and packaging co-design,” in *Proceedings of the 2026 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2026, pp. 189–200.
- [41] International Technology Roadmap for Semiconductors, “The international technology roadmap for semiconductors: 2005 edition - interconnect,” International Technology Roadmap for Semiconductors (ITRS), Technical Report, 2005, devised for technology assessment only.
- [42] E. Rosenbaum, V. Shukla, and M.-S. Keel, “ESD protection networks for 3D integrated circuits,” in *2011 IEEE International 3D Systems Integration Conference (3DIC), 2011 IEEE International*. IEEE, 2012, pp. 1–7.
- [43] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, “ASAP7: A 7-nm finFET predictive process design kit,” *Microelectronics Journal*, vol. 53, pp. 105–115, 2016.
- [44] S. Nikolić, F. Catthoor, Z. Tőkei, and P. Ienne, “Global is the new local: FPGA architecture at 5nm and beyond,” in *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2021, pp. 34–44.
- [45] L. Zhu, C. Jo, and S. K. Lim, “Power delivery solutions and PPA impacts in micro-bump and hybrid-bonding 3D ICs,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 12, no. 12, pp. 1969–1982, 2022.
- [46] J. Luu, J. H. Anderson, and J. S. Rose, “Architecture description and packing for logic blocks with hierarchy, modes and complex interconnect,” in *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*, 2011, pp. 227–236.
- [47] M. A. Elgammal, A. Mohaghegh, S. G. Shahrouz, F. Mahmoudi, F. Koşar, K. Talaei, J. Fife, D. Khadivi, K. Murray, A. Boutros, K. B. Kent, J. Goeders, and V. Betz, “VTR 9: Open-source CAD for fabric and beyond FPGA architecture exploration,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 18, no. 3, Aug. 2025.
- [48] K. E. Murray, S. Zhong, and V. Betz, “AIR: A fast but lazy timing-driven FPGA router,” in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 338–344.
- [49] A. Hahn Pereira and V. Betz, “CAD and routing architecture for interposer-based multi-FPGA systems,” in *Proceedings of the 2014 ACM/SIGDA international symposium on Field-programmable gate arrays*, 2014, pp. 75–84.
- [50] S. G. Shahrouz, S. Ho, and V. Betz, “Congestion-aware CAD optimizations for routing-constrained FPGAs,” in *2025 International Conference on Field Programmable Technology (ICFPT)*. IEEE, 2025, pp. 229–230.
- [51] A. Arora, A. Boutros, S. A. Damghani, K. Mathur, V. Mohanty, T. Anand, M. A. Elgammal, K. B. Kent, V. Betz, and L. K. John, “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*, vol. 42, no. 11, pp. 3895–3909, 2023.
- [52] A. Mohaghegh, “Expanding FPGA CAD and architecture exploration to new routing structures and stacked silicon,” Master’s thesis, University of Toronto (Canada), 2024.
- [53] S.-H. Lin, M. Simicic, N. Pantano, S.-H. Chen, G. Van Der Plas, E. Beyne, and P. Wambacq, “Toward 0 V ESD protection in 2.5D/3D advanced bonding technology,” in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2024, pp. 1–2.
- [54] E. Haque, P. S. Nalla, J. Zhang, S. S. Sapatnekar, C. Chakrabarti, and Y. Cao, “Tiny chiplets enabled by packaging scaling: Opportunities in ESD protection and signal integrity,” in *2025 IEEE 16th International Conference on ASIC (ASICON)*. IEEE, 2025, pp. 1–4.