vFPGA: Towards Sub-μs Reconfiguration via 3D FPGA and Packaging Co-Design

Nikhil K. Cherukuri*
cheru054@umn.edu
University of Minnesota
Minneapolis, MN, USA

Sharad Nag
nag00010@umn.edu
University of Minnesota
Minneapolis, MN, USA

Pragnya S. Nalla
nalla052@umn.edu
University of Minnesota
Minneapolis, MN, USA

Ashish K. Kola
kola0161@umn.edu
University of Minnesota
Minneapolis, MN, USA

Chetan S. Gadireddi
cgadired@asu.edu
Arizona State University
Tempe, AZ, USA

Kevin Dai
dai00175@umn.edu
University of Minnesota
Minneapolis, MN, USA

Jae-sun Seo
js3528@cornell.edu
Cornell Tech
New York, NY, USA

Zhenman Fang
zhenman@umn.edu
University of Minnesota
Minneapolis, MN, USA

Jeff Zhang
jeffzhang@asu.edu
Arizona State University
Tempe, AZ, USA

Yu Cao*
yucao@umn.edu
University of Minnesota
Minneapolis, MN, USA

Abstract

Traditional FPGAs suffer from long reconfiguration latencies, ranging from milliseconds to seconds, which significantly limits their usage in applications that require rapid multi-context switching and dynamic partial reconfiguration. To overcome this barrier, we propose vertical FPGA (vFPGA), a novel 3D FPGA and packaging co-design methodology that leverages advanced packaging to achieve sub-μs full-chip reconfiguration, while maximizing the on-chip storage capacity for multiple configuration contexts. In vFPGA, we partition a 2D FPGA die into two dies: a bottom die with the conventional programmable fabric, and a top die with memory banks and control logic to store multiple configuration contexts. These two dies are interconnected face-to-face through high density, vertical connections, such as micro bumps, Cu pillars, or hybrid bonding, which enable high-throughput, parallel programming paths that rapidly reconfigure the bottom die. Using the OpenFPGA framework with GlobalFoundries (GF) 22nm FDSOI process technology, we design both dies and conduct a comprehensive design space exploration (DSE) in light of packaging technology scaling. Our DSE focuses on the co-design of the pitch and dimensions of vertical interconnects, logic fabric partitioning, scan-chain partitioning, and memory bank selection and layout, with the objective of minimizing reconfiguration latency. In the physical implementation of 3D die-to-die integration, we also address critical signal integrity issues including clock synchronization and electrostatic discharge (ESD), along with power delivery challenges. Our simulation results demonstrate that, with 22nm dies and 45μm Cu pillar pitch, our new 3D FPGA achieves a sub-μs reconfiguration latency, independent of the FPGA die size, yielding a 1,000× faster reconfiguration over traditional 2D FPGAs. Moreover, with hybrid bonding as vertical interconnects, we achieve sub-10 ns reconfiguration latency.

CCS Concepts

• **Hardware → Reconfigurable logic and FPGAs**; **3D integrated circuits**; **VLSI packaging**; • **Computer systems organization → Reconfigurable computing**.

Keywords

3D FPGA, Rapid Reconfiguration, Multi-context Reconfiguration, 3D Packaging, Memory-on-Logic

ACM Reference Format:
Nikhil K. Cherukuri, Sharad Nag, Pragnya S. Nalla, Ashish K. Kola, Chetan S. Gadireddi, Kevin Dai, Jae-sun Seo, Zhenman Fang, Jeff Zhang, and Yu Cao. 2026. vFPGA: Towards Sub-μs Reconfiguration via 3D FPGA and Packaging Co-Design. In *Proceedings of the 2026 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '26), February 22–24, 2026, Seaside, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3748173.3779206

1 Introduction

Field-Programmable Gate Arrays (FPGAs) are reconfigurable semiconductor devices composed of programmable logic blocks, configurable interconnects, and heterogeneous resources such as block RAMs (BRAMs), digital signal processing (DSP) units, and high-speed I/Os [12]. They have become a versatile computing platform across cloud infrastructure, embedded systems, networking, adaptive deep learning, and scientific computing [1, 5, 14, 29, 37, 43]. Their post-fabrication programmability bridges the gap between the flexibility of software and the efficiency of application-specific integrated circuits (ASICs) [12]. This adaptability allows FPGAs to deliver efficient hardware acceleration, rapid prototyping, and long-term support for evolving application requirements. FPGAs further exploit massive fine-grained parallelism through thousands

*Corresponding authors

FPGA '26, Seaside, CA, USA

ACM ISBN 979-8-4007-2079-6/2026/02
https://doi.org/10.1145/3748173.3779206

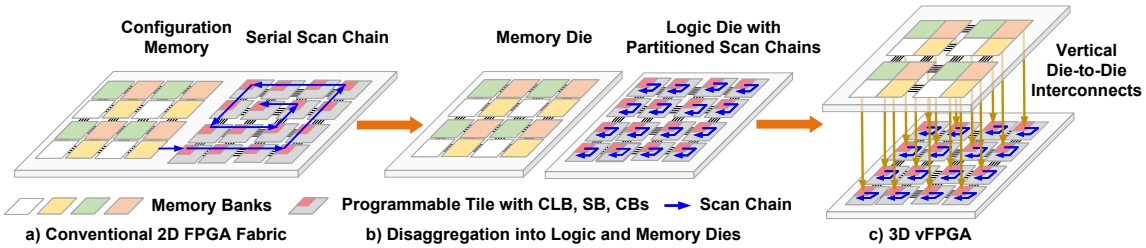

Figure 1: 2D FPGA to 3D vFPGA. (a) Conventional 2D FPGA fabrics integrate configuration memory and logic with long serial scan chains. (b) Disaggregation separates memory and logic into distinct dies, introducing localized, partitioned scan chains. (c) Proposed vFPGA vertically stacks memory and logic dies using high-density VIs, enabling scalable and parallel reconfiguration.

of concurrently active logic elements and custom data paths, delivering high-throughput and energy-efficient execution without the overheads of general-purpose processors [30, 31].

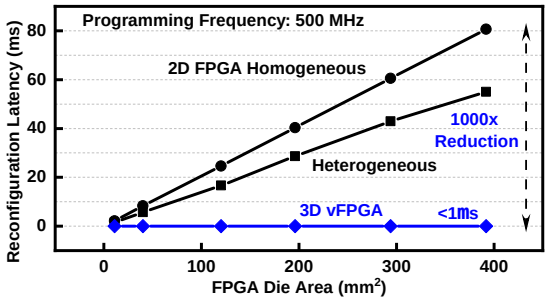

Figure 2: Conventional 2D FPGAs incur ms-scale reconfiguration latency that increases with CLB count, while the proposed vFPGA sustains sub-µs latency independent of area.

Despite these advantages, FPGA reconfigurability introduces substantial configuration overhead that limits fast context switching and partial reconfiguration (PR) [12, 36]. Programming an FPGA requires loading a bitstream into distributed storage cells, typically flip-flops or SRAM, embedded within configurable logic blocks (CLBs), switch blocks (SBs), and connection blocks (CBs) [7, 36, 43]. Conventional devices rely on a global serial shift-register chain (SRC) that loads one bit per clock cycle, causing full-chip reconfiguration to span milliseconds (ms) to seconds depending on bitstream size and interface bandwidth. Figure 2 illustrates that reconfiguration latency increases linearly with the effective die area for both homogeneous (CLB-only) and heterogeneous (CLB + DSP/BRAM) 2D island-style fabrics, and remains in the millisecond range even at high programming frequencies. Both fabrics were generated with fixed layout dimensions using OpenFPGA [34], with die area estimated from the corresponding homogeneous fabrics synthesized in GF22. In this setup, heterogeneous fabrics replace 50% of CLB columns with DSP/BRAM tiles without altering the physical footprint, isolating the impact of architectural heterogeneity on bitstream size and reconfiguration latency.

To address long reconfiguration times, FPGA vendors and researchers have explored architectural optimizations [42] such as parallel scan chains and frame-based programming, which provide programming throughput and support PR. Yet, these approaches incur significant I/O, routing, and area overheads [13], and their benefits diminish as bitstream sizes scale with device capacity.

Complex applications within heterogeneous workloads often require rapid switching or reconfiguration across multiple algorithms or workload phases, including real-time multi-model AI inference, multi-step robotics navigation and planning, and data-adaptive high-frequency trading (HFT). For instance, robotic workloads follow the sequential sense–plan–act paradigm but must adapt computation across perception, localization, planning, and control layers, each with considerable algorithmic variability requiring high-speed PR [26]. Even in domains where FPGAs provide substantial speedup over CPUs and improved energy efficiency compared with GPUs, such as video analytics, long configuration latencies remain a major bottleneck for rapid run-time reconfiguration (RTR) [38]. Similar constraints arise in HFT, where sub-µs responsiveness is required to adapt to market dynamics without disrupting active pipelines [27]. Modern AI workloads also increasingly rely on multiple specialized models. Hierarchical inference frameworks such as Super-Sub [40] first run a lightweight classifier to identify a superclass and then invoke a fine-grained model for subclass prediction [43]. Supporting such multi-model pipelines requires fast, selective context switching and the ability to activate many hardware configurations with minimal overhead. Together, these trends motivate the need for FPGA architectures that achieve sub-µs reconfiguration, support efficient multi-context (MC) storage, and enable scalable PR.

To address these requirements, we propose a vertical FPGA (vFPGA) that overcomes the limitations of 2D FPGAs without incurring substantial planar silicon area or footprint overhead. The vFPGA employs a novel 3D FPGA and packaging co-design methodology that leverages advanced packaging technologies to achieve sub-µs full-chip reconfiguration while maximizing on-chip storage for multiple configuration contexts. As shown in Figure 1, the proposed architecture partitions a traditional 2D FPGA into two stacked dies: a bottom logic die (LD) containing the programmable fabric, and a top memory die (MD) integrating dense configuration memory banks and control circuitry. The two dies are connected face-to-face using high-density vertical interconnects (VIs), such as µ-bumps, copper (Cu) pillars, or hybrid bonding [17, 20, 22, 24, 32], providing high-throughput, parallel configuration paths that replace long serial scan chains (Figure 1c) with localized partitions and significantly reduce reconfiguration latency. The LD is partitioned into independent regions aligned with corresponding MD partitions (Figure 3), enabling coarse-grained static PR. Moreover, the MD stores

multiple configuration contexts, allowing rapid context switching without the silicon area and latency overheads associated with 2D memory-based FPGA designs. By co-optimizing VI pitch, memory organization, and 3D integration parameters, the vFPGA achieves sub-µs full-chip reconfiguration and efficient multi-context operation with improved runtime adaptability. To enable scalable sub-µs multi-context reconfiguration in vFPGAs, this work makes the following key contributions:

- Proposes a **two-tier 3D FPGA and packaging co-design methodology** that partitions a traditional 2D FPGA into stacked programmable logic and memory dies.
- Develops a **2D-to-3D bitstream mapping algorithm** that segments a 2D bitstream into tile-aligned parallel scan-chain segments, enabling high-bandwidth vertical reconfiguration.
- Introduces a **scalable, partitioned** reconfiguration scheme with independent clock and control domains across aligned inter-die regions, supporting both coarse-grained static partial reconfiguration and synchronous full-chip programming.
- Designs **memory bank organization and selection** strategies that maximize multi-context capacity while minimizing latency and preserving design simplicity.
- Performs comprehensive **DSE**, demonstrating sub-µs full-chip reconfiguration, independent of FPGA size, achieving up to a **1,000**× speedup over conventional 2D FPGAs.
- Analyzes and addresses critical **3D integration challenges**, including clock synchronization, ESD protection, and power delivery in face-to-face die stacking.

The remainder of this paper is organized as follows. Section 2 reviews background and related work. Section 3 describes the vFPGA architecture and associated DSE. Section 4 presents 3D circuit-level design considerations, while Section 5 outlines 3D programming methodologies. Section 6 reports implementation details and experimental results, and Section 7 concludes the paper.

2 Background and Related Work

This section reviews 2D FPGA reconfiguration mechanisms, advances in 2.5D and 3D integration, and prior 3D FPGA architectures, and identifies the limitations that motivate the proposed vFPGA.

2.1 2D FPGA Reconfiguration

FPGA reconfiguration latency is the time required to load a configuration bitstream, determined primarily by bitstream size and available configuration bandwidth. Over time, FPGA configuration mechanisms have evolved to balance flexibility, performance, power, and area. Early devices relied on SRC configuration, in which bits propagate sequentially through distributed flip-flops. Although SRC offers simplicity and minimal area overhead, it scales poorly, resulting in high reconfiguration latency. Modern commercial FPGAs instead employ SRAM-based frame configuration, where addressable volatile memory frames enable faster access, parallel loading, and PR, but at the cost of higher static power, larger silicon area, and reliance on external non-volatile storage [23]. Despite these advantages, scan-chain based approaches remain attractive for semi-custom or resource-constrained FPGAs where area and leakage power are critical [13, 23]. Advanced techniques such as PR allow selective runtime updates to FPGA regions, improving hardware reuse and dynamic adaptability, but are limited by tools, routing constraints, and floorplanning complexity [36, 43]. Multi-context FPGAs, in contrast, provide near-instant switching between multiple stored configurations, enabling cycle-level reconfiguration with a limited number of contexts, but incurring significant area and energy overheads even with architectural optimizations [11, 35]. As modern applications including adaptive signal processing, cryptography, high-performance computing, and AI/ML demand increasingly dynamic behavior, 2D FPGAs face growing bottlenecks due to limited configuration bandwidth and the large area footprint required for on-chip MC storage.

2.2 3D Packaging Technologies

Advanced packaging methods have enabled 2.5D and 3D heterogeneous integration of large-scale systems with improved data movement and energy efficiency. Several bonding approaches have progressively reduced the pitch of VIs. Early methods relied on solder bumps at coarse pitches greater than 100 µm, but these structures scale poorly at finer dimensions. Copper pillar-based µ-bumps were introduced to overcome this limitation, consisting of a cylindrical Cu column topped with a solder cap. They are fabricated by electroplating Cu, followed by solder deposition; during reflow, the solder forms intermetallic compound (IMC) bonds, enabling robust face-to-face or face-to-back stacking [18]. Hybrid bonding eliminates solder entirely and instead forms direct Cu-Cu contacts embedded within dielectric-to-dielectric bonded layers. This method requires chemical mechanical planarization (CMP) to ensure intimate wafer or die surface contact [4, 16]. Scaling trends show that µ-bump pitch has reduced from ∼80 µm pitch in early flip-chip designs to ∼40-20 µm today, with research targeting 10 µm and below [10]. Hybrid bonding has advanced from ∼10 µm down to sub-5 µm, with roadmaps projecting 1–2 µm for dense 3D IC integration [17, 20, 24, 32]. Compared with µ-bumps, hybrid bonding supports finer pitch, lower parasitics, and higher interconnect density, at the cost of increased process complexity. In this work, we consider the scaling capabilities of both µ-bumps and hybrid bonding in the context of vFPGA integration and DSE.

2.3 3D Design and 3D FPGAs

As conventional 2D FPGA fabrics approach scaling limits, researchers have increasingly explored 3D integration to address density, performance, and interconnect bottlenecks. Commercial devices employing 2.5D integration, such as Xilinx Virtex-7 [42] and Intel Stratix-10 [19], place multiple chiplets on passive interposers and integrate them to increase logic capacity. While effective for capacity scaling, these designs suffer from limited inter-die bandwidth and increased latency due to long interposer routing. True 3D stacking enabled by through-silicon vias (TSVs) or face-to-face bonding offers significantly higher interconnect density, lower latency, and improved power efficiency. Unlike 2.5D systems, which are constrained by edge-based interconnects, fully stacked 3D designs can exploit dense Cu interconnects distributed across the entire die surface. Prior work has explored both logic-on-logic stacking and logic-on-heterogeneous integration, supported by modeling and evaluation frameworks for 3D FPGA architectures [6, 9, 39, 44]. These studies examine trade-offs among stacking multiple FPGA

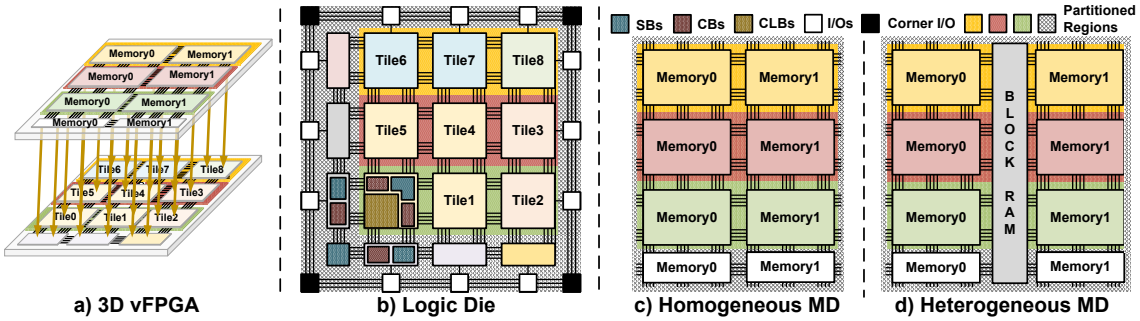

Figure 3: Overview of the proposed vFPGA architecture: (a) logic die (LD) and memory die (MD) stacked using Cu pillars, (b) tile-level layout of the baseline FPGA fabric on LD, showing CLBs, CBs, SBs, and I/Os, with partitioned domains, (c) homogeneous MD with domain-aligned configuration memories, (d) heterogeneous MD incorporating BRAM and memory banks.

fabrics to expand logic capacity, integrating logic with memory or ASIC accelerators for heterogeneous systems, and embedding configuration memory within back-end-of-line layers using monolithic 3D approaches to improve area efficiency, power, total wire length and critical path delay relative to 2D FPGAs, but not full-chip MC reconfiguration latency. Recent demonstrations further highlight the potential of fine-pitch 3D integration [2, 18, 41]. For example, [33] prototyped a 3D ASIC system-on-chip in TSMC 7nm using hybrid bonding with sub-2 µm pitch, achieving substantial improvements in on-chip capacity and performance for augmented reality workloads within the same area footprint. Their design addresses critical 3D challenges such as clock forwarding, clock-tree synthesis (CTS) balancing, power delivery, and IR-drop analysis, underscoring the maturity of 3D integration for high-performance systems. In contrast, commercial multi-die FPGAs primarily adopt 2.5D or multi-die integration to scale logic capacity rather than to accelerate reconfiguration. Rapid context switching in these devices is typically achieved through logic duplication and output multiplexing, trading area for ns-scale switching latency, but supporting only a limited number of contexts. Collectively, these observations motivate fully stacked 3D integration with dense die-to-die VIs as a promising direction to enable high-bandwidth reconfiguration and scalable MC operation, and sub-µs latency, while preserving the benefits of 2D FPGA fabrics with minimal design refactoring.

3 Architecture Exploration and Definition

Modern heterogeneous workloads such as multi-model AI inference, robotics, and HFT, require rapid RTR with minimal system disruption, motivating a holistic 3D FPGA co-design of 2D architectures and packaging technologies. In particular, reducing RTR requires careful selection of VI density, partitioning of signal and power VIs, and memory organization on the MD to maximize MC capacity. To guide these design choices, we conduct a comprehensive DSE that evaluates the trade-offs among these parameters.

3.1 Baseline 2D FPGA and Fabric Generation

We adopt an island-style homogeneous, tileable FPGA architecture as the baseline for exploration. A simplified 3×3 instance of this fabric is shown in Figure 3b to illustrate the organization of CLBs, CBs and SBs; in our study the architecture is scaled to a 23×23 grid of CLBs, each containing ten 6-input look-up tables (LUTs), with programmable routing implemented via CBs that connect routing tracks to CLB inputs and SBs that provide inter-track connectivity. The device contains 1.1×10^6 configuration bits, programmed serially through a single scan chain, resulting in full-chip reconfiguration latencies of 3.6 ms and 2.18 ms at frequencies of 300 MHz and 500 MHz, respectively. This baseline provides a foundation for evaluating VI pitch, memory organization, and programming methodologies prior to detailed circuit- and system-level design. We use the open-source OpenFPGA framework [34], built on the Versatile Place and Route (VPR) tool [28], to generate 2D FPGA fabrics with different architectural configurations using a flexible CAD flow. OpenFPGA takes as input the VPR architecture XML, OpenFPGA-specific architectural annotations, benchmark RTLs, timing and design constraint files, switching activity data, and optional verification or optimization settings. The framework supports FPGA-Verilog and FPGA-Bitstream subflows, enabling automated generation of 2D fabric RTLs, SDC files, and native bitstream. Our experiments employ both homogeneous and heterogeneous FPGA fabric templates derived from VPR architecture definitions, with key parameters including fabric dimensions (auto or fixed), routing channel width (auto or fixed), heterogeneous block placement, and configuration protocols. Figure 4 illustrates the overall flow. Because the VPR flow automatically scales fabric size and routing channel width based on RTL benchmark requirements, resulting in variable bitstream sizes, we enforce a fixed fabric size and channel width across all RTL benchmarks. This ensures consistent bitstream

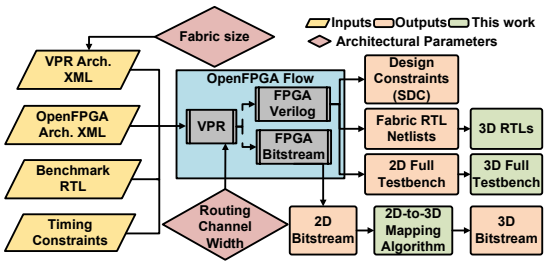

Figure 4: OpenFPGA-based design flow illustrating the generation of 3D-aware vFPGA RTLs, bitstreams, and testbenches.

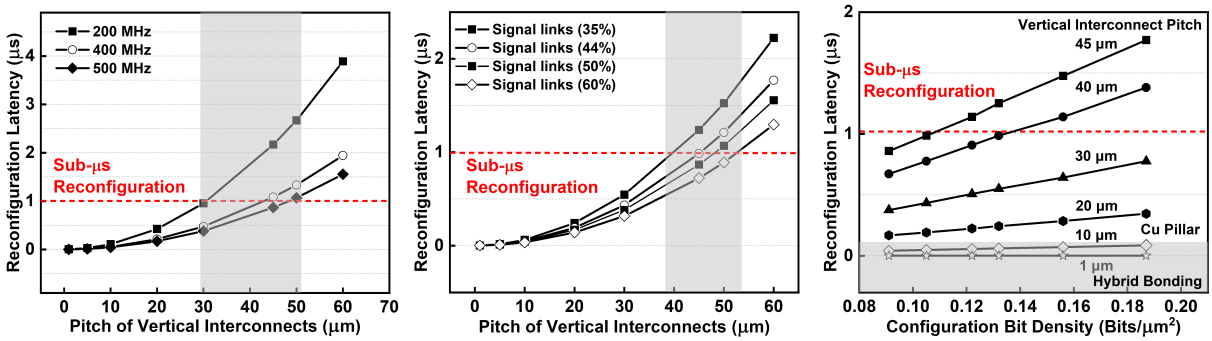

Figure 5: Reconfiguration latency versus (a) VI pitch across programming frequencies (50% signal-links, fixed bit density), (b) VI pitch with varying signal-link allocations (500 MHz, fixed bit density), and (c) bit density across VI pitches (500 MHz, 44% signal-links). Sub-µs operation is maintained at ≤30 µm pitches, with 500 MHz extending scalability to 50 µm (Cu pillars).

sizes and enables fair evaluation of multi-context bitstreams across different benchmark circuits, which is essential for analyzing vertical and multi-context RTR in vFPGA. The generated RTL fabrics are synthesized in GF22 FDSOI technology to obtain area and timing estimates, which are used to derive VI counts across different pitches and corresponding 3D reconfiguration latencies.

3.2 VI pitch and Reconfiguration Latency

The VI pitch directly influences reconfiguration latency by determining the density of parallel programming links between LD and MD. To quantify this, we analyze three key parameters: programming frequency, signal-link allocation, and configuration bit density. Our initial analysis assumes linear scaling of the memory bit-cells and peripheral circuitry with the logic fabric, and sufficient memory bandwidth to fully utilize available signal links at fine VI pitches.

3.2.1 Across Programming Frequencies. We begin by assuming a CLB pitch of 140 µm derived from synthesis results. The total number of VIs within this area is evenly partitioned between signal and power links (50% each), consistent with common practices in 3D integration analysis [45]. Figure 5a illustrates the impact of VI pitch on reconfiguration latency across different programming frequencies, assuming fixed bit density and a 50/50 signal–power VI partition. As VI pitch decreases, interconnect density increases, enabling higher programming parallelism and reduced reconfiguration latency. The observed quadratic trend arises because the number of VIs scales inversely with the square of pitch, while reconfiguration latency scales inversely with the available VI count. For coarse pitches above 50 µm, reconfiguration latency remains above the sub-µs regime even at higher programming frequencies. In contrast, pitches below 30 µm consistently achieve sub-µs reconfiguration across evaluated frequencies. At 500 MHz, sub-µs latency is also attainable with pitches below 50 µm using Cu pillars, making this frequency-pitch combination a practical near-term design point (shaded region in Figure 5a). Overall, increasing programming frequency from 200 to 500 MHz yields modest latency reductions compared to the substantial gains achieved through aggressive VI pitch scaling. With advanced integration such as 1 µm pitch hybrid bonding, latency can be reduced to sub-10 ns range, underscoring the potential of fine-pitch VIs for future vFPGA designs.

3.2.2 Signal VI Allocation. Figure 5b evaluates the impact of signal-link allocation on reconfiguration latency, with programming frequency fixed at 500 MHz and configuration bit density held constant. Based on the prior analysis, VI pitches below 50 µm were identified as a practical operating region, and are therefore retained here for further exploration. As the fraction of VIs allocated to signal links increases, programming throughput improves, leading to reduced reconfiguration latency. At a 35% signal-link allocation, programming latency is highest, with several configurations exceeding 1 µs once the pitch approaches ~40 µm. Increasing the allocation to 44% reduces latency and allows sub-µs operation to persist up to 45 µm. These results reinforce the earlier observation that both programming frequency and VI partitioning strongly influence reconfiguration latency. While higher programming frequency and increased signal-link allocation extend the sub-µs operating envelope, pitches of 50 µm and above generally transition into the multi-µs regime regardless of link allocation. This behavior underscores the importance of aggressive VI pitch scaling in combination with sufficient signal-link provisioning to achieve low-latency reconfiguration.

3.2.3 Impact of Configuration Bit Density. Figure 5c shows the effect of configuration bit density on reconfiguration latency across VI pitches, assuming a fixed 44% signal-link allocation and a programming frequency of 500 MHz. Bit density is derived from the baseline design by scaling the CLB pitch from 140 µm to 105 µm, capturing area scaling trends expected in advanced technology nodes and custom FPGA implementations. As bit density increases, reconfiguration latency becomes increasingly sensitive to VI pitch. At coarse pitches (40-45 µm), latency rapidly exceeds the sub-µs target with increasing bit density, whereas finer pitches (≤35 µm) consistently sustain sub-µs reconfiguration. At the extreme of 1 µm hybrid bonding (shaded region in Figure 5c), latency theoretically approaches the nanosecond regime, enabling near single-cycle full-chip reconfiguration that is effectively independent of bit density.

The combined analysis establishes clear design guidelines for vFPGAs. The VI pitches of ≤30 µm consistently achieve sub-µs reconfiguration across programming frequencies, signal-link allocations, and bit densities, rendering latency largely insensitive to other parameters. For VI pitches up to 50 µm, a 500 MHz programming frequency with ~44% signal links provides a practical balance

that sustains sub-µs reconfiguration under moderate increases in bit density. Building on these insights, we next examine memory scaling and bank organization, which are critical for sustaining sub-µs performance and enabling scalable multi-context reconfiguration.

3.3 Memory Scaling and Bank Organization

While VI density sets the upper limit of inter-die reconfiguration bandwidth, memory scaling and bank organization determine how efficiently this bandwidth is utilized by the memory die at fine VI pitches to achieve lowest possible reconfiguration latency and high multi-context capacity. In 3D FPGAs, the memory die must therefore support both dense context storage and fast, parallel access to the logic die, making careful bank partitioning, alignment with logic regions, and inter-bank organization critical for sustaining high context capacity. We use GF22 SRAM memories to evaluate the selection of different memory bank sizes under different partitioning granularities and study their impact on achievable context capacity and operating frequency in the slow–slow (SS) corner.

We consider five partitioning granularities: per tile, per half region, per one region, and across two and three regions. As shown in Figure 3b, a tile comprises CLBs, SBs, and CBs. A sub-region corresponds to half the tiles within a region, one region spans one full row of tiles, and multi-region configurations cover two or three complete rows. The architecture scales accordingly across these partitioning levels based on architectural choices. The scan-chain partitioned regions on the LD are aligned with corresponding memory partitions on the MD, as illustrated in Figure 3c for homogeneous fabrics and Figure 3d for heterogeneous fabrics incorporating BRAMs. These scalable partitioning strategies enable static PR at multiple granularities and can be naturally extended to support dynamic PR. Fine-grained partitions, such as tile-level regions, require smaller memory banks to reconfigure localized portions of the logic die, whereas coarser partitions spanning multiple regions allow the use of larger memory banks with higher bandwidth to reconfigure multiple tiles concurrently. From a scaling perspective, multiple small memory banks incur higher area overhead per bit due to the peripheral circuitry, whereas fewer, larger banks reduce area cost but typically operate at lower frequencies, introducing potential performance constraints. This analysis therefore examines the trade-offs among bank size, operating frequency, and architectural partitioning to quantify their impact on supported context capacity and overall MC reconfiguration efficiency.

Figure 6 illustrates the trade-offs between the normalized area per bit and achievable clock frequency (SS corner) for different memory partitioning granularities aligned with scan-chain regions on the LD (Figure 3). Fine-grained partitions (tile or sub-region level) sustain high operating frequencies but incur substantial area overhead due to a larger fraction of peripheral circuitry, which limits the number of storable contexts and constrains the ability to drive all signal VIs at finer pitches (≤20 µm). As partitioning scales to multiple larger banks spanning one to three regions, area efficiency improves substantially, enabling higher context capacity. However, operating frequency degrades and falls below the 500 MHz target when banks span three or more regions; this trade-off is acceptable at fine VI pitches below 30 µm, where reconfiguration latency is largely insensitive to programming frequency.

These results underscore a fundamental design trade-off: finer-grained partitioning with smaller banks sustains higher access frequencies but incurs area overhead penalties, whereas larger banks improve area efficiency at the cost of reduced operating frequency. Balancing these factors is critical for maintaining sub-µs reconfiguration while supporting scalable multi-context storage. For VI pitches in the 30-50 µm range, partitioning at the one-region or two-region level provides a practical balance, with the final choice guided by specific design requirements, including PR.

3.4 Balancing Context Capacity and PR

As illustrated in Figure 6, memory bank partitioning directly influences both context capacity and the granularity of static or dynamic PR. Fine-grained partitioning at the tile or sub-region level offers flexible reconfiguration of small fabric regions but limits the number of pre-stored contexts, as configuration memory is fragmented across many small banks. In contrast, coarser partitioning at the one- or two-region level supports higher context capacity by amortizing peripheral overhead, at the expense of reduced PR granularity. This trade-off is further influenced by VI overhead, since finer-grained partitioning requires additional clock and control VIs. Although smaller VI pitches help mitigate this overhead by providing more routing density, the fundamental balance between PR granularity and context storage remains. For VI pitches in the 30-50 µm range, partitioning at the one- or two-region level offers the most practical compromise, enabling high context capacity while supporting coarse-grained PR with manageable interconnect overhead.

3.5 Heterogeneous Architecture Exploration

In OpenFPGA-generated heterogeneous 2D fabrics, architectural variation is introduced by replacing selected CLB tile columns with specialized IP blocks such as BRAMs, DSP units, or adder chains. Relative to homogeneous fabrics, this substitution reduces the effective configuration bit density on the LD, which correspondingly lowers reconfiguration latency when extended to proposed vFPGA. Heterogeneous resources can also be placed on the MD (Figure 3d), increasing the logic density on the LD but imposing additional demands on VI allocation and reducing available context capacity. In such cases, VIs must be provisioned not only for configuration data delivery but also for access to associated memory banks and heterogeneous resources. Although finer VI pitches increase the available

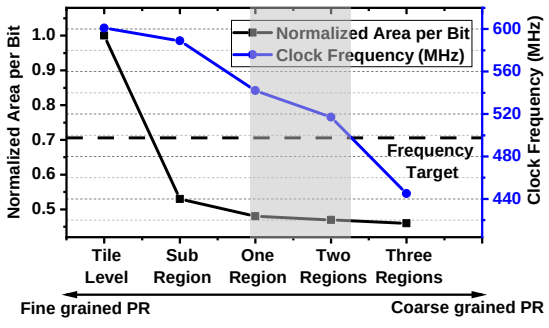

Figure 6: Area–frequency trade-off across memory bank partitioning levels, from fine- to coarse-grained PR.

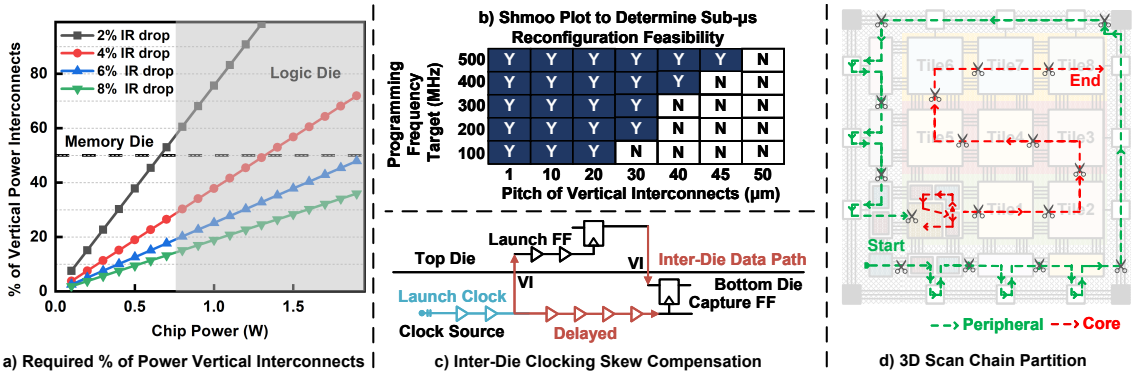

Figure 7: 3D integration challenges in vFPGA design: (a) power VI requirements, (b) sub-µs feasibility shmoo plot, (c) inter-die clock skew management, and (d) core/peripheral scan-chain partitioning of 2D bitstream.

inter-die connectivity, careful VI allocation is required to balance reconfiguration bandwidth, context capacity, and heterogeneous resource connectivity. Effectively managing these trade-offs enables denser heterogeneous fabrics that preserve the performance and flexibility advantages of these FPGA architectures while sustaining low reconfiguration latency and scalable multi-context support.

4 3D FPGA Circuit Design Analysis

This section analyzes key circuit-level challenges in implementing the proposed vFPGA with fine-pitch 3D integration, focusing on ESD reliability, power delivery, and inter-die clock synchronization.

4.1 ESD Considerations

Electrostatic Discharge (ESD) remains a critical reliability challenge in 3D IC packaging as die stacking and fine-pitch VIs become increasingly prevalent. The shrinking VI pitch significantly constrains the available I/O footprint, limiting the integration of conventional pad-based ESD protection structures. This concern is more severe in advanced technology nodes, where reduced gate-oxide thickness lowers breakdown voltage thresholds and increases susceptibility to ESD-induced oxide failure. Traditional ESD protection circuits with diode networks and RC-clamp paths are effective in planar ICs with sufficient pad pitch; however, in 3D ICs they incur excessive silicon area overhead and introduce parasitic capacitance that degrades high-speed I/O signaling. Moreover, standard ESD models such as the human body model (HBM) and charged device model (CDM), which involve the application of high-voltage pulses to external pads or internal nodes, do not fully capture ESD discharge mechanisms in stacked-die environments, where VIs are embedded within the die. Recent studies indicate that wafer-to-wafer (W2W) hybrid bonding offers natural ESD mitigation through a voltage suppression effect, whereby elevated inter-wafer capacitance can reduce electrostatic potential by up to 98% [25]. When combined with the inherent series resistance of metal routing, this capacitance significantly attenuates discharge currents, reducing the need for dedicated ESD devices. In such assemblies, CDM events during stacking dominate, while HBM events are largely irrelevant. In contrast, die-to-die bonding does not benefit from comparable capacitive suppression and therefore requires more careful design and reliance on interconnect resistance for current damping [15]. Although the RC characteristics of interconnect structures remain largely invariant with decreasing pitch generations [15], the intrinsic series resistance of metal routing and interconnects plays a pivotal role in attenuating peak discharge currents. For larger voltage transients, this necessitates reliance on foundry-level cleanroom integration practices to ensure safe 3D die stacking and to minimize the likelihood of ESD events during prototype fabrication. Thus, in this work, ESD robustness during face-to-face stacking is ensured through reliance on foundry-level cleanroom practices.

4.2 vFPGA Power Considerations

Power delivery is a critical challenge in face-to-face stacked architectures where the top die relies on VIs from the bottom die for power supply. In vFPGA, this challenge is amplified because the finite number of VIs within a given die area must be shared between power delivery and signal connectivity for parallel programming, clock distribution, and control. Allocating a larger fraction of VIs to power delivery directly reduces the available signal VIs, thereby constraining achievable reconfiguration latency, especially for VI pitches in the 30-50 µm range, as observed in the DSE (Figure 5b). Consequently, accurate characterization of the power requirements of both LD and MD, along with a careful choice of die stacking order, is essential to balance power delivery integrity and reconfiguration performance. Figure 7a illustrates the VI requirements for power delivery as a function of chip power under different IR-drop limits. The shaded region represents the power envelope of the 2D LD, which scales with CLB count, while the MD's power demand, derived from activity-driven analysis of SRAM bank operations, is significantly lower. The fraction of VIs allocated to power delivery increases with the total chip power of the die placed on top, as maintaining a voltage drop within 2-6% of V_{DD} requires progressively more power VIs. These requirements are estimated by modeling IR drop across the Cu pillar and via stack in GF22. Stricter IR-drop constraints (2-4%) drive power VI requirements beyond 40-60% at higher chip power levels, while relaxed limits (6-8%) reduce the power VI demand but at the expense of power integrity margins.

Placing the MD on top is therefore advantageous, as its lower power demand consumes fewer power VIs, leaving a larger fraction

for signal routing and enabling sustained sub-µs reconfiguration or even lower latency at finer pitches, provided that memories can supply sufficient bandwidth. Once sub-µs reconfiguration is achieved, additional VIs can be reassigned to power, to further strengthen the power delivery network without impacting latency, depending on design requirements. Increasing the number of power VIs benefits both dies by reducing current density per interconnect and improving overall supply robustness. In contrast, placing the LD on top, significantly increases power VI requirements due to its higher power consumption, reducing the number of signal VIs and limiting reconfiguration flexibility for pitches above 30 µm. Consequently, stacking the lower-power MD on top minimizes contention between power and signal VIs while providing greater opportunity to reinforce power integrity and sustained fast reconfiguration.

4.3 Clock Distribution and Skew Management

To address timing challenges in 3D circuits, we implement a high-speed source-synchronous inter-die clocking architecture in which the bottom die serves as the primary timing reference (Figure 7c). The source-synchronous method is preferred for high-speed 3D ICs as it mitigates the impact of uncorrelated process variations between stacked dies [3, 21, 33]. In vFPGA, each die is synthesized with an independent clock tree to support local operations. The cross-die synchronization is achieved by forwarding the bottom-die clock to the top die, which uses this to launch return data. The bottom die then samples the returning data using a locally delayed capture clock. This capture clock is explicitly calibrated to compensate for the worst-case round-trip RC latency, accounting for both the forward clock path and the return data path across Cu pillars. This methodology minimizes the skew accumulation and is critical for ensuring timing closure during parallel 3D programming, enabling robust, sub-µs reconfiguration in vFPGA architectures.

5 3D FPGA Programming Methodologies

In this section, we present a bitstream segmentation algorithm that converts a 2D bitstream generated by OpenFPGA into a 3D bitstream for vFPGA. The resulting bitstream is stored on the top MD and delivered to the bottom LD via VIs that drive multiple short, parallel scan chains, significantly reducing reconfiguration latency compared to a single long scan chain. The algorithm requires three categories of inputs. The bitstream inputs include the 2D binary configuration bitstream ($fabric\_bitstream.bit$) and the structural mapping of bits to FPGA tiles extracted from the XML netlist ($fabric\_bitstream.xml$). The DSE inputs specify the total number VIs (L_{3D}) within the die area and the percentage of signal links range (R_s). The design inputs describe the physical layout of the FPGA fabric and the corresponding 2D scan-chain routing. The algorithm outputs the generated 3D bitstream and the total number of reconfiguration clock cycles for both the core and periphery ($Cyc\_reconfig\_core$, $Cyc\_reconfig\_peri$). A key feature of the algorithm is that scan-chain partitioning is performed at tile granularity rather than uniformly slicing the bitstream across signal VIs. This enables scalable and repeatable tile blocks across the layout, simplifying hardware generation, RTL design and programming workflows. Separating the bitstream into core and peripheral

Algorithm 1 2D Bitstream Segmentation for 3D vFPGA

Bitstream Inputs: 2D bitstream (bit_{2d}), XML netlist (xml)
DSE Inputs: Total VIs (L_{3D}), % Signal VIs range (R_s)
Design Inputs: Physical layout, 2D scan-chain route
Outputs: 3D bitstream, Total reconfiguration clock cycles for core ($Cyc\_reconfig\_core$) and periphery ($Cyc\_reconfig\_peri$)

Step 1: Define groups
Read xml and bit_{2d}
Extract $N_{tiles}, N_{peri}, N_{io}$ from layout
$split\_loc \leftarrow I\_O\_tile\_last$ from xml
Group 1: $Nbit\_peri \leftarrow |bit_{2d}[0 : split\_loc]|$
Group 2: $Nbit\_core \leftarrow |bit_{2d}[split\_loc{+}1 : \text{end}]|$
$Nbits\_tile \leftarrow \max_i\big(|bit_{2d}[core\_tile\_i\_indices]|\big)$

Step 2: Balanced Link Allocation
$L_{range} \leftarrow \{1, \ldots, \lfloor R_s * L_{3D} \rfloor\}$; $prev\_imbalance \leftarrow \infty$
for $L \in L_{range}$ **do**
    $L\_peri \leftarrow \lfloor \frac{L*Nbit\_peri}{Nbit\_peri+Nbit\_core} \rfloor$; $L\_core \leftarrow L_{3D} - L\_peri$
    $imbalance \leftarrow |\frac{Nbit\_peri}{L\_peri} - \frac{Nbit\_core}{L\_core}|$
end for
Store and select the L_core and L_peri with minimal imbalance
$\lambda_{core} \leftarrow \frac{Nbit\_core}{L\_core}$
$\lambda_{peri} \leftarrow \frac{Nbit\_peri}{L\_peri}$

Step 3: Core Tile Assignment
$L\_per\_tile\_core = \lceil \frac{Nbits\_tile}{\lambda_{core}} \rceil$
$Cyc\_reconfig\_core = \lceil \frac{Nbits\_tile}{L\_per\_tile\_core} \rceil$
Execute Algorithm 2 to segment the core 2D bitstream

Step 4: Peripheral Assignment
Assign one link per periphery tile
$Cyc\_reconfig\_peri \leftarrow \max(|bit_{2d}[peri\_tile\_i\_indices]|)$

$L_{io} \leftarrow \lceil \frac{\sum(|bit_{2d}[io\_i\_indices]|)}{Cyc\_reconfig\_peri} \rceil$
Pad smaller tiles with zeros as specified in Algorithm 2

Step 5: Validation
$L_{total} \leftarrow L\_per\_tile\_core * N_{tiles} + 1 * N_{peri} + L_{io} * 1$
if $L_{total} > \max(L_{range})$ **then**
    Rechoose the next-optimal configuration (Line 18)
    (L_{core}, L_{peri}) and repeat Steps 3–5
end if

Algorithm 2 Step 3: Split the 2D core bitstream into tile segments

$L \leftarrow$ links_per_core_tile
segments$[1..L] \leftarrow \epsilon$ ▷ per-link streams (empty)
for all t in core_tile **do**
    base $\leftarrow \lfloor \lambda_{core}/L \rfloor$; rem $\leftarrow \lambda_{core}$ mod L
    seg_len $\leftarrow base + 1$ ▷ length of the longer segments
    for $k = 1$ to L **do**
        segments[k] $\leftarrow$ next $(base + 1)$ if $k \leq$ rem else next $base$ bits
        Pad segments[k] with zeros to seg_len ▷ pad shorter segments
    end for
end for

groups provides two additional advantages: (i) the core can be partitioned into evenly sized regions to support balanced PR or flexibly adapted to design requirements, and (ii) peripheral I/O tiles can be programmed independently, which is beneficial for static PR requiring stable I/O contexts during dynamic updates. The scan-chain segmentation begins by partitioning the 2D bitstream into peripheral (Group 1) and core (Group 2) regions using the XML file. As shown in Step 1 of Algorithm 1, the split point is defined by the last I/O tile entry ($I\_O\_tile\_last$). In this step, the maximum number of bits across all core tiles ($Nbits\_tile$) is identified. Under parallel scan-chain programming, this value determines the total number of reconfiguration clock cycles, and therefore directly sets the reconfiguration latency. In Step 2, the average number of configuration bits per VI, for the core and periphery (λ_{core} and λ_{peri}) is evaluated over a range of signal-link fractions (R_s). The signal-link allocation that minimizes the imbalance between core and peripheral bits is selected. Guided by the DSE results in Section 3, R_s is constrained to the range of 44-50%. The core tile assignment is performed in Step 3. The core 2D bitstream is traversed tile by tile (Figure 7d) and divided into tile-level segments. For each tile containing λ_{core} bits and assigned L_{core} links, bits are distributed as uniformly as possible across the links, with any remainder allocated to a subset of links, as given in Algorithm 2. To enable synchronous shifting across parallel scan chains, shorter segments are right-padded with zeros so that all links associated with a tile shift an equal number of bits. The total number of reconfiguration cycles for the core ($Cyc\_reconfig\_core$) is computed as $Nbits\_tile/L\_per\_tile\_core$. Step 4 assigns VIs for peripheral tiles. Since peripheral tiles contain fewer configuration bits than core tiles, dedicated links (one link each) are assigned to the bottom-left, bottom-edge, and left-edge tiles. The remaining I/O and corner tiles share a common link pool sized to match the maximum peripheral programming cycle count (Figure 7d). This maximum is determined by the peripheral tile with highest number of configuration bits, and padding is applied as needed to equalize cycle counts across all peripheral links. Finally, Step 5 validates the total number of assigned links does not exceed the available VI budget. If this constraint is violated, the next optimal signal-link allocation from Step 2 is selected and the subsequent steps are repeated. Overall, the algorithm partitions the 2D bitstream into multiple parallel short scan-chains driven by VIs from MD, enabling efficient and scalable vFPGA reconfiguration.

6 Results and Implementation

Through our exploration, we derive a set of design principles that guide scalable vFPGA architectures. Our DSE shows that VI pitch is the dominant factor governing reconfiguration latency, with pitches ≤30 µm consistently achieving sub-µs operation across evaluated parameters. Although advanced hybrid bonding at 1-10 µm pitch can theoretically enable sub-10 ns switching, practical implementations are constrained by memory scaling, where peripheral circuitry overhead limits the number of ports available per tile or region, preventing full utilization of the increased signal VI count. As a result, the primary bottleneck shifts from VI density to memory periphery scalability, which limits the achievable configuration bandwidth despite abundant signal VIs. Therefore, for hybrid-bonded vFPGAs, architectures that jointly balance sub-µs reconfiguration latency, multi-context storage, and partial reconfiguration, while allocating remaining signal VIs to support high-performance heterogeneous FPGA fabrics, represents a practical and scalable design strategy. At intermediate pitches of 30-50 µm, a 500 MHz programming frequency with ~44% signal-link allocation provides a practical operating point, sustaining sub-µs reconfiguration even under moderate increases in configuration density for semi-custom designs. Memory scaling further introduces trade-offs among operating frequency, area efficiency, and multi-context capacity, with one- or two-region memory partitioning offering the most effective balance between coarse-grained PR and context storage. Collectively, these results define cohesive guidelines for achieving fast, scalable, and power-aware reconfiguration in future vFPGAs.

6.1 vFPGA Implementation

We implement a vFPGA prototype guided by insights from the DSE.

6.1.1 Prototype Architecture, Partitioning, and Clocking. A 12×8 tileable island-style homogeneous 2D FPGA fabric is generated using OpenFPGA. Using the proposed 2D-to-3D bitstream mapping algorithm, signal VIs are optimally allocated, and the bitstream is partitioned into peripheral and core groups, with further segmentation at inter- and intra-tile granularity to form multiple short scan chains on the LD. The vFPGA RTLs are designed for the MD, integrating SRAM banks and control logic, and for the LD, incorporating scan-chain partitioning and control logic to support two-region static PR. Only minimal modifications to the original 2D FPGA RTLs are required, enabling a practical extension of existing architectures to vFPGA. For testing purposes, the prototype supports both 2D and the proposed 3D parallel programming modes. The vFPGA is synthesized in GF22 using a 45 µm Cu pillar pitch and implements a 10×6 CLB fabric (Figure 9a). The peripheral and core groups are divided into seven regions, which are combined into three static PR domains by grouping two core regions per domain, while the peripheral region forms a separate static PR domain. Each domain can be reconfigured independently without affecting the others. This capability is enabled through distributed multi-clock architecture, in which separate on-chip clocks drive individual PR domains under a centralized clock control block driven from the on-chip global clock. Synchronous full-chip reconfiguration is supported by activating all clocks simultaneously. In this prototype, three core PR domains operate synchronously, while the peripheral PR domain is asynchronous and completes earlier. This design choice reduces padding overhead during bitstream partitioning. Figure 9d

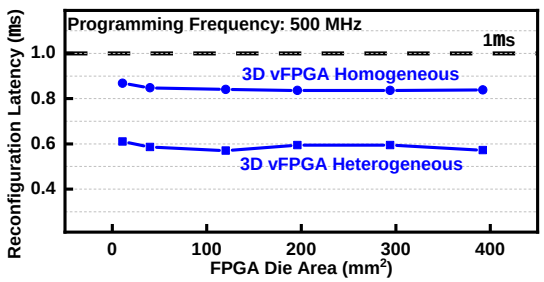

Figure 8: RTR of vFPGA fabrics across area at 45 µm VI pitch.

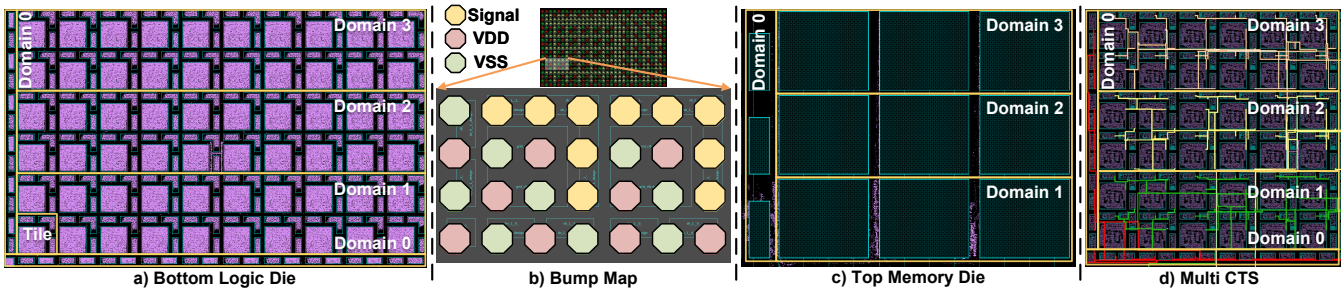

Figure 9: Post-layout views of the 12×8 vFPGA with defined PR domains: (a) bottom LD (2.0 mm × 1.25 mm), (b) bump map highlighting the distribution of signal, V_{DD}, and V_{SS} interconnects, and (c) top MD (1.8 mm × 1.05 mm), (d) multi-domain CTS.

illustrates the multi-clock tree synthesis on the LD, with the same clocking strategy mirrored on the MD to ensure inter-die domain alignment. The MD supports 27 pre-loaded configuration contexts using ping-pong operation, provisioned with three 4K × 100 SRAM banks per core PR domain and three 2880 × 18 banks for the peripheral domain. The prototype employs four parallel 100 Mbps shift-register scan chains, achieving an on-chip SRAM fill time of ∼0.44 ms per context from off-chip DRAM. A custom multi-port shift-register scan-chain interface is designed to support this off-chip to on-chip data transfer. The top two metal layers (IA and IB) handle signal redistribution to VIs, minimizing routing impact on each die. Computation results are stored in LD's on-chip memory and read out off-chip through serializers.

6.1.2 VI Allocation, RTR, Power Analysis. The VI budget is partitioned such that 43% support parallel RTR, while approximately 55% are allocated to power delivery for the top MD. The remaining VIs are used for distributing clocks and control signals to the MD. Within the core, each tile is provisioned with five signal VIs (Figure 9b). In the Cu pillar bump map, power and ground VIs are prioritized near the periphery, close to inline wire-bonding pads on the bottom die, while ensuring that signal VIs are surrounded by sufficient power and ground bumps. During vertical programming, inactive regions are clock-gated. This VI allocation strategy scales naturally to larger fabrics. With these design choices, the vFPGA achieves a RTR of 0.83 µs at a 500 MHz programming frequency. As shown in Figure 10, reconfiguration latency decreases with increasing frequency for the allocated signal VIs. Power analysis indicates that the logic-die clock network dominates total consumption

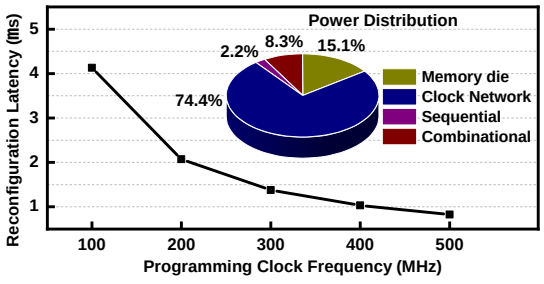

Figure 10: RTR versus programming clock frequency, with a power distribution breakdown of 12×8 vFPGA design.

(∼74.4%), due to dense scan-chain flip-flops distributed within each tile and across the die. These results underscore the importance of balanced VI allocation, power-aware bump mapping, and optimized clocking. Once reconfiguration latency targets are met, reallocating additional VIs to power and clock distribution offers an effective path toward improved reliability in larger-scale implementations.

6.1.3 Functional Verification and Comparison. We perform RTL simulations by integrating both dies into a top-level module to validate RTR of pre-loaded multi-contexts using customized testbenches and bitstreams generated by the mapping algorithm. Benchmarks include a 16-bit MAC, a 16-bit multiplier, a 128-bit counter, and an 8-bit reconfigurable processing element supporting both MAC and FFT operations. In all cases, the vFPGA reconfigures correctly and produces the expected outputs, confirming functional correctness across diverse multi-context workloads.

For context, [8] demonstrated a 3D multi-context FPGA in UMC 90 nm using 10 µm Cu–Cu bonding, reporting 8.42 µs RTR with DRAM-based configuration storage. By comparison, the proposed vFPGA in GF22 achieves 0.83 µs RTR using SRAM-based multi-context storage with no tile-area overhead at a 45 µm Cu-pillar pitch. Based on our DSE, extrapolating the vFPGA to a 10 µm pitch reduces RTR to ∼0.04 µs, independent of fabric size.

7 Conclusion

In this work, we present a two-tier 3D FPGA and packaging co-design methodology that partitions logic and memory dies to enable scalable, high-speed RTR. Through memory bank optimization, partitioned reconfiguration, and a comprehensive DSE, the proposed vFPGA architecture achieves sub-µs full-chip reconfiguration, and an up to 1,000× speedup over 2D FPGAs. By jointly addressing inter-die clocking, ESD robustness, and power delivery in face-to-face stacking, this work establishes a practical balance between latency, multi-context capacity, and system reliability, providing a scalable foundation for future 3D FPGA architectures targeting adaptive and complex heterogeneous computing systems.

Acknowledgments

This work was supported in part by CoCoSys, one of the seven centers sponsored by the Semiconductor Research Corporation (SRC) and DARPA under the Joint University Microelectronics Program 2.0 (JUMP 2.0).

References

[1] Kamaraj A, Vijaya Kumar, P Vishnu Prasanth, Ritam Dutta, Bhaskar Roy, and Prabal Chakraborty. 2024. FPGAs as Hardware Accelerators in Data Centers: A Survey From the Data Centric Perspective. In *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*. 1–6. doi:10.1109/DICCT61038.2024.10533053

[2] Alchip Technologies. 2025. Alchip 3DIC Test Chip Tape Out Validates Ecosystem Readiness. https://www.alchip.com/en/Newsroom/Alchip_3DIC_Test_Chip_Tape_Out

[3] Samyoung Bang, Kwangsoo Han, Andrew B. Kahng, and Vaishnav Srinivas. 2015. Clock clustering and IO optimization for 3D integration. In *2015 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*. 1–8. doi:10.1109/SLIP.2015.7171709

[4] Wenhan Bao, Jieqiong Zhang, Hei Wong, Jun Liu, and Weidong Li. 2025. Emerging Copper-to-Copper Bonding Techniques: Enabling High-Density Interconnects for Heterogeneous Integration. *Nanomaterials* 15, 10 (2025). doi:10.3390/nano15100729

[5] Christophe Bobda, Joel Mandebi Mbongue, Paul Chow, Mohammad Ewais, Naif Tarafdar, Juan Camilo Vega, Ken Eguro, Dirk Koch, Suranga Handagala, Miriam Leeser, Martin Herbordt, Hafsah Shahzad, Peter Hofste, Burkhard Ringlein, Jakub Szefer, Ahmed Sanaullah, and Russell Tessier. 2022. The Future of FPGA Acceleration in Datacenters and the Cloud. *ACM Trans. Reconfigurable Technol. Syst.* 15, 3, Article 34 (Feb. 2022), 42 pages. doi:10.1145/3506713

[6] Andrew Boutros, Fatemehsadat Mahmoudi, Amin Mohaghegh, Stephen More, and Vaughn Betz. 2023. Into the Third Dimension: Architecture Exploration Tools for 3D Reconfigurable Acceleration Devices. In *2023 International Conference on Field Programmable Technology (ICFPT)*. 198–208. doi:10.1109/ICFPT59805.2023.00027

[7] Stephen D. Brown, Robert J. Francis, Jonathan Rose, and Zvonko G. Vranesic. 1992. *Field-programmable gate arrays*. Kluwer Academic Publishers, USA.

[8] Alesandro Cevrero, Panagiotis Athanasopoulos, Hadi Parandeh-Afshar, Maurizio Skerlj, Philip Brisk, Yusuf Leblebici, and Paolo Ienne. 2009. Using 3D integration technology to realize multi-context FPGAs. 507 – 510. doi:10.1109/FPL.2009.5272454

[9] S. Chiricescu, M. Leeser, and M.M. Vai. 2001. Design and analysis of a dynamically reconfigurable three-dimensional FPGA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 9, 1 (2001), 186–196. doi:10.1109/92.920832

[10] Do Hoon Cho, Seong Min Seo, Jang Baeg Kim, Sri Harini Rajendran, and Jae Pil Jung. 2021. A Review on the Fabrication and Reliability of Three-Dimensional Integration Technologies for Microelectronic Packaging: Through-Si-via and Solder Bumping Process. *Metals* 11, 10 (2021). doi:10.3390/met11101664

[11] W. Chong, S. Ogata, M. Hariyama, and M. Kameyama. 2005. Architecture of a multi-context FPGA using reconfigurable context memory. In *19th IEEE International Parallel and Distributed Processing Symposium*. 7 pp.–. doi:10.1109/IPDPS.2005.112

[12] Katherine Compton and Scott Hauck. 2002. Reconfigurable computing: a survey of systems and software. *ACM Comput. Surv.* 34, 2 (June 2002), 171–210. doi:10.1145/508352.508353

[13] Ganesh Gore, Xifan Tang, and Pierre-Emmanuel Gaillardon. 2023. A Scalable and Area-Efficient Configuration Circuitry for Semi-Custom FPGA Design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 31, 8 (2023), 1128–1139. doi:10.1109/TVLSI.2023.3270448

[14] Kaiyuan Guo, Shulin Zeng, Jincheng Yu, Yu Wang, and Huazhong Yang. 2019. [DL] A Survey of FPGA-based Neural Network Inference Accelerators. *ACM Trans. Reconfigurable Technol. Syst.* 12, 1, Article 2 (March 2019), 26 pages. doi:10.1145/3289185

[15] Emad Haque, Pragnya Sudershan Nalla, Jeff Zhang, Sachin S. Sapatnekar, Chaitali Chakrabarti, and Yu Cao. 2025. Tiny Chiplets Enabled by Packaging Scaling: Opportunities in ESD Protection and Signal Integrity. arXiv:2511.10760 [cs.AR] https://arxiv.org/abs/2511.10760

[16] Han-Wen Hu and Kuan-Neng Chen. 2021. Development of low temperature CuCu bonding and hybrid bonding for three-dimensional integrated circuits (3D IC). *Microelectronics Reliability* 127 (2021), 114412. doi:10.1016/j.microrel.2021.114412

[17] IEEE Electronics Packaging Society. 2024. *Heterogeneous Integration Roadmap (HIR)*. Technical Report. https://eps.ieee.org/technology/heterogeneous-integration-roadmap/2024-edition.html Accessed: 2025-10-01.

[18] 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.F. Petersburg, M. Phen-givoni, A. M. Pillai, A. Sairam, P. Shekhar, P. Sinha, P. Stover, A. Telang, and Z. Zell. 2019. Foveros: 3D Integration and the use of Face-to-Face Chip Stacking for Logic Devices. In *2019 IEEE International Electron Devices Meeting (IEDM)*. 19.6.1–19.6.4. doi:10.1109/IEDM19573.2019.8993637

[19] Intel Corporation. 2017. Intel Stratix 10 Device Overview. https://www.intel.com/content/www/us/en/docs/programmable/683729/current/gx-sx-device-overview.html. White Paper / Device Overview.

[20] Ye Jin Jang, Ashutosh Sharma, and Jae Pil Jung. 2023. Advanced 3D Through-Si-Via and Solder Bumping Technology: A Review. *Materials* 16, 24 (2023). doi:10.3390/ma16247652

[21] David Kehlet. 2017. Accelerating Innovation Through a Standard Chiplet Interface: The Advanced Interface Bus (AIB). Intel White Paper. https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/accelerating-innovation-through-aib-whitepaper.pdf

[22] Jinwoo Kim, Lingjun Zhu, Hakki Mert Torun, Madhavan Swaminathan, and Sung Kyu Lim. 2021. Micro-bumping, Hybrid Bonding, or Monolithic? A PPA Study for Heterogeneous 3D IC Options. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 1189–1194. doi:10.1109/DAC18074.2021.9586229

[23] Dirk Koch. 2012. *Partial Reconfiguration on FPGAs: Architectures, Tools and Applications*. Springer. doi:10.1007/978-1-4614-1225-0

[24] John H. Lau. 2024. *Flip Chip, Hybrid Bonding, Fan-In, and Fan-Out Technology*. Springer. doi:10.1007/978-981-97-2140-5

[25] S.-H. Lin, M. Simicic, N. Pantano, S.-H. Chen, G. Van Der Plas, E. Beyne, and P. Wambacq. 2024. 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)*. 1–2. doi:10.1109/VLSITechnologyandCir46783.2024.10631467

[26] Shaoshan Liu, Zishen Wan, Bo Yu, and Yu Wang. 2021. *Robotic Computing on FPGAs*. doi:10.1007/978-3-031-01771-1

[27] John W. Lockwood, Adwait Gupte, Nishit Mehta, Michaela Blott, Tom English, and Kees Vissers. 2012. A Low-Latency Library in FPGA Hardware for High-Frequency Trading (HFT). In *2012 IEEE 20th Annual Symposium on High-Performance Interconnects*. 9–16. doi:10.1109/HOTI.2012.15

[28] Kevin E. Murray, Oleg Petelin, Sheng Zhong, Jia Min Wang, Mohamed Eldafrawy, Jean-Philippe Legault, Eugene Sha, Aaron G. Graham, Jean Wu, Matthew J. P. Walker, Hanqing Zeng, Panagiotis Patros, Jason Luu, Kenneth B. Kent, and Vaughn Betz. 2020. VTR 8: High-performance CAD and Customizable FPGA Architecture Modelling. *ACM Trans. Reconfigurable Technol. Syst.* 13, 2, Article 9 (June 2020), 55 pages. doi:10.1145/3388617

[29] Anouar Nechi, Lukas Groth, Saleh Mulhem, Farhad Merchant, Rainer Buchty, and Mladen Berekovic. 2023. FPGA-based Deep Learning Inference Accelerators: Where Are We Standing? *ACM Trans. Reconfigurable Technol. Syst.* 16, 4, Article 60 (Oct. 2023), 32 pages. doi:10.1145/3613963

[30] Tan Nguyen, Samuel Williams, Marco Siracusa, Colin MacLean, Douglas Doerfler, and Nicholas J. Wright. 2020. The Performance and Energy Efficiency Potential of FPGAs in Scientific Computing. In *2020 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*. 8–19. doi:10.1109/PMBS51919.2020.00007

[31] Murad Qasaimeh, Kristof Denolf, Jack Lo, Kees Vissers, Joseph Zambreno, and Phillip H. Jones. 2019. Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels. In *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*. 1–8. doi:10.1109/ICESS.2019.8782524

[32] Semiconductor Research Corporation (SRC). 2025. *Microelectronics and Advanced Packaging Technologies Roadmap (MAPT)*. Technical Report. https://srcmapt.org/ Accessed: 2025-10-01.

[33] H. Ekin Sumbul, Arne Symons, Lita Yang, Huichu Liu, Tony F. Wu, Matheus Trevisan Moreira, Debabrata Mohapatra, Abhinav Agarwal, Kaushik Ravindran, Chris Thompson, Yuecheng Li, and Edith Beigne. 2025. A 3D Design Methodology for Integrated Wearable SoCs: Enabling Energy Efficiency and Enhanced Performance at Iso-Area Footprint. In *2025 Design, Automation & Test in Europe Conference (DATE)*. 1–7. doi:10.23919/DATE64628.2025.10992815

[34] Xifan Tang, Edouard Giacomin, Baudouin Chauviere, Aurélien Alacchi, and Pierre-Emmanuel Gaillardon. 2020. OpenFPGA: An Open-Source Framework for Agile Prototyping Customizable FPGAs. *IEEE Micro* 40, 4 (July 2020), 41–48. doi:10.1109/MM.2020.2995854

[35] S. Trimberger, D. Carberry, A. Johnson, and J. Wong. 1997. A time-multiplexed FPGA. In *Proceedings. The 5th Annual IEEE Symposium on Field-Programmable Custom Computing Machines Cat. No.97TB100186)*. 22–28. doi:10.1109/FPGA.1997.624601

[36] Kizheppatt Vipin and Suhaib A. Fahmy. 2018. FPGA Dynamic and Partial Reconfiguration: A Survey of Architectures, Methods, and Applications. *ACM Comput. Surv.* 51, 4, Article 72 (July 2018), 39 pages. doi:10.1145/3193827

[37] Han Wang, Xiaoyue Liu, Yi Yang, Qin Liao, Shiyu Ke, and Yong Dong. 2025. FPGA-Based Acceleration of Deep Learning Networks: Techniques and Applications. In *Proceedings of the 2025 6th International Conference on Computer Information and Big Data Applications (CIBDA '25)*. Association for Computing Machinery, New York, NY, USA, 774–778. doi:10.1145/3746709.3746843

[38] Shang Wang, Chen Zhang, Yuanchao Shu, and Yunxin Liu. 2019. Live Video Analytics with FPGA-based Smart Cameras. In *Proceedings of the 2019 Workshop on Hot Topics in Video Analytics and Intelligent Edges* (Los Cabos, Mexico) *(HotEdgeVideo'19)*. Association for Computing Machinery, New York, NY, USA, 9–14. doi:10.1145/3349614.3356027

[39] Faaiq Waqar, Jiahao Zhang, Anni Lu, Zifan He, Jason Cong, and Shimeng Yu. 2025. Monolithic 3D FPGA Design and Synthesis with Back-End-of-Line Configuration Memories. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*. 1–7. doi:10.1109/DAC63849.2025.11132615

[40] Shixian Wen, Amanda Sofie Rios, Kiran Lekkala, and Laurent Itti. 2022. What can we learn from misclassified ImageNet images? arXiv:2201.08098 [cs.CV] https://arxiv.org/abs/2201.08098

[41] Tony F. Wu, Huichu Liu, H. Ekin Sumbul, Lita Yang, Dipti Baheti, Jeremy Coriell, William Koven, Anu Krishnan, Mohit Mittal, Matheus Trevisan Moreira, Max Waugaman, Laurent Ye, and Edith Beigné. 2024. 11.2 A 3D integrated Prototype System-on-Chip for Augmented Reality Applications Using Face-to-Face Wafer Bonded 7nm Logic at <2µm Pitch with up to 40% Energy Reduction at Iso-Area Footprint. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 67. 210–212. doi:10.1109/ISSCC49657.2024.10454529

[42] Xilinx Inc. 2011. Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency. https://docs.amd.com/v/u/en-US/wp380_Stacked_Silicon_Interconnect_Technology. White Paper: WP380 (v1.0).

[43] Yixin Xu, Zijian Zhao, Yi Xiao, Tongguang Yu, Halid Mulaosmanovic, Dominik Kleimaier, Stefan Duenkel, Sven Beyer, Xiao Gong, Rajiv Joshi, Xiaobo Hu, Shixian Wen, Amanda Sofie Rios, Kiran Lekkala, Laurent Itti, Eric Homan, Sumitha George, Vijaykrishnan Narayanan, and Kai Ni. 2024. Ferroelectric FET-based context-switching FPGA enabling dynamic reconfiguration for adaptive deep learning machines. *Science Advances* 10, 3 (2024), eadk1525. doi:10.1126/sciadv.adk1525

[44] Ismael Youssef, Hang Yang, and Cong Hao. 2025. LaZagna: An Open-Source Framework for Flexible 3D FPGA Architectural Exploration. arXiv:2505.05579 [cs.AR] https://arxiv.org/abs/2505.05579

[45] Lingjun Zhu, Nesara Eranna Bethur, Yi-Chen Lu, Youngsang Cho, Yunhyeok Im, and Sung Kyu Lim. 2022. 3D IC Tier Partitioning of Memory Macros: PPA vs. Thermal Tradeoffs. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design* (Boston, MA, USA) *(ISLPED '22)*. Association for Computing Machinery, New York, NY, USA, Article 19, 6 pages. doi:10.1145/3531437.3539724