FlexiCTS: CPPR-Aware 3D Clock Tree Synthesis for Face-to-Face Bonded ICs

Shanyi Li[1], Leilei Jin[1⋆], Siyuan Liang[1], Zhen Zhuang[1], Rongmei Chen[2], Bei Yu[1], Tsung-Yi Ho[1]
[1] The Chinese University of Hong Kong, [2] Peking University,
⋆: Corresponding Author

Abstract

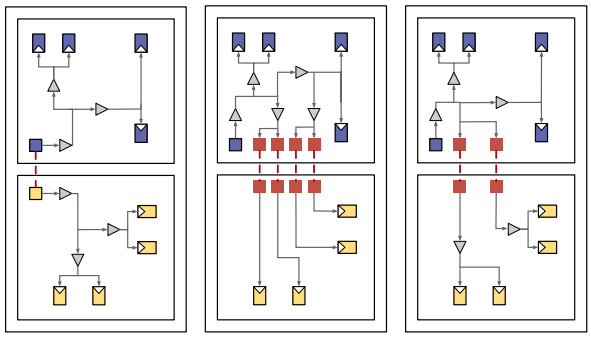
Face-to-Face 3D ICs enable advanced vertical integration but pose critical challenges for clock tree synthesis (CTS). Existing pseudo-3D flows rigidly partition clock paths across dies, fragmenting common paths and crippling Common Path Pessimism Removal (CPPR), resulting in excessive skew and inefficient hybrid bonding terminal (HBT) utilization. We present FlexiCTS, a correct-by-construction CPPR-aware framework featuring adaptive cross-die buffer assignment to maximize path sharing and optimize HBT allocation. Experimental results show that FlexiCTS achieves 4.1× skew reduction and 88% fewer HBTs than state-of-the-art methods, while simultaneously matching 2D timing quality with superior resource efficiency.

1 Introduction

Face-to-Face (F2F) stacking, enabled by sub-micron wafer-to-wafer (W2W) hybrid bonding, has emerged as a transformative technology in advanced 3D-ICs [1]. Its deployment in commercial products like AMD's 3D V-Cache [2] and META's AR SoC [3] demonstrates an industry-wide shift towards fine-grained integration. Unlike traditional Through-Silicon Via (TSV) approaches limited to block-level stacking with large interconnect pitch (typically 5–10μm), F2F hybrid bonding achieves sub-micron pitch (<1μm) [4]. This dramatic pitch reduction enables vertical connections with electrical characteristics approaching those of local on-chip wiring, fundamentally altering the design space for system-level optimization.

However, this architectural evolution exposes critical limitations in established design methodologies, particularly in Clock Tree Synthesis (CTS). In 3D-ICs, CTS transitions from a single-die problem to a complex multi-die co-optimization challenge that governs system-wide timing, power, and reliability [5–7]. The designer must manage clock skew across physically distinct dies while navigating uncorrelated inter-die process variations, which is a challenge absent in 2D designs. Yet, within this complexity, a unique optimization opportunity emerges.

The dense, low-parasitic interconnects of F2F bonding enable *cross-die common clock paths*, unlocking substantial benefits through Common Path Pessimism Removal (CPPR). To understand this opportunity, consider that modern static timing analysis (STA) introduces pessimism through conservative On-Chip Variation (OCV) modeling [8, 9]. When launching and capturing flip-flops share a common clock path segment, delay variations along this segment affect both paths identically and thus cancel out in timing analysis.



(a) Cascade-2D [13] (b) Pin-3D [14] (c) FlexiCTS

Figure 1: Pseudo-3D CTS strategies for F2F ICs: (a) Port partitioning strategy: independent clock trees are built on each die for their respective clock sinks. (b) Centralized buffering strategy: all clock buffers are constrained to a single die. (c) FlexiCTS: A unified clock tree is built with buffers placed across both dies.

CPPR recovers this artificially pessimistic timing margin by crediting the correlated variation [10–12]. In F2F architectures, the ability to route clock paths across dies with near-local-wire characteristics means that buffers on one die can serve as common path elements for flip-flops (FFs) on both dies, which is a capability fundamentally unattainable with high-parasitic TSV interconnects. This presents a critical design dilemma: aggressively extending common paths across dies maximizes CPPR credit but risks degrading clock skew and exhausting limited hybrid bonding terminals (HBTs). Conversely, partitioning clock trees early to minimize skew forfeits valuable timing margin. This multi-objective trade-off renders existing CTS strategies inadequate.

Prior research has not fully addressed this co-optimization challenge, with academic and industrial approaches taking fundamentally different paths. Academic work has largely focused on TSV-based architectures, with algorithms like 3D-MMM [15], DME [16], and inductance-aware methods [17] targeting wirelength minimization and TSV cost reduction. These methods assume high-cost vertical links and cannot be directly applied to hybrid bonding, where the cost model for cross-die connections is fundamentally different due to their low parasitics.

In the commercial context, the absence of native 3D CTS tools has led to *pseudo-3D* frameworks that adapt conventional 2D flows through rigid partitioning strategies. As illustrated in Figure 1, these approaches fall into two categories, each with inherent limitations. The *clock port partitioning* strategy (Figure 1(a)), employed by Cascade-2D [13] and AnchorGrid [18], builds independent clock trees on each die. This minimizes HBT usage but forces early path divergence, severely limiting CPPR opportunities and degrading timing performance. Conversely, the *centralized buffering* strategy (Figure 1(b)),

DAC '26, Long Beach, CA, USA

ACM ISBN 979-8-4007-2254-7/2026/07
https://doi.org/10.1145/3770743.3803984

adopted by Pin-3D [14], constrains all clock buffers to a single die to maintain 2D tool compatibility. While this maximizes common path length, it creates extreme imbalance in cross-die delays and consumes excessive HBTs, leading to significant clock skew degradation. The fundamental flaw in both strategies is their reliance on static partitioning. This approach involves a predetermined assignment of buffers to dies, which prevents adaptation to the local characteristics of different clock tree regions.

To overcome these limitations, we introduce **FlexiCTS**, a CPPR-aware 3D CTS framework that dynamically navigates the trade-off space through adaptive buffer assignment. As shown in Figure 1(c), FlexiCTS abandons fixed partitioning in favor of an intelligent synthesis framework that makes buffer decisions based on local subtree characteristics, timing criticality, and resource constraints. This adaptive approach achieves superior timing closure through holistic co-optimization of clock topology, CPPR credit, and bonding resource allocation. Our key contributions are summarized as follows:

- We propose a flexible optimization strategy that dynamically assigns and replicates buffers across dies, thereby overcoming the limitations of rigid, pre-partitioned approaches.
- We introduce a comprehensive, CPPR-aware cost model that accurately evaluates the trade-offs among clock skew, path pessimism, and power to guide the optimization.
- We develop an efficient algorithm for hybrid bonding resource management that minimizes cross-die interconnect overhead.
- Experimental results on industrial 3D-IC benchmarks demonstrating significant timing improvements over existing flows, with competitive area and power metrics.

2 Preliminaries

2.1 Hybrid Bonding for F2F 3D-ICs

F2F hybrid bonding represents a paradigm shift in 3D integration technology. Unlike traditional TSV-based stacking, where dies are connected face-to-back through thick silicon vias, F2F bonding joins two dies F2F at the wafer level, creating a dense array of interconnects at the bonding interface [19]. Its dense grid of Hybrid Bonding Terminals (HBTs) at the die-to-die interface features ultra-low parasitic loading, enabling cross-die signals to be treated as on-chip global routes and thus eliminating the need for bulky I/O circuits.

This architecture, however, creates a fundamental interdependency between CTS and HBT assignment. The set of inter-die clock nets is not a static input but rather a dynamic outcome of CTS decisions. For instance, replicating a buffer to improve timing generates a new net requiring an HBT. The relationship is deeply intertwined, such that CTS's topological choices define the assignment problem, while HBT's resource constraints limit the viable solution space for CTS. Consequently, a robust 3D CTS framework must replace this sequential paradigm with a flexible approach that co-optimizes topology synthesis and HBT assignment.

2.2 CPPR in 3D-IC Clock Tree Synthesis

Static Timing Analysis (STA) identifies the most critical paths in a design but is inherently pessimistic due to *common path pessimism*. This pessimism arises because the clock path segment shared by both launching and capturing flip-flops (FFs) is assumed to have maximum delay for setup checks and minimum delay for hold checks. Common Path Pessimism Removal (CPPR) corrects this by calculating a credit

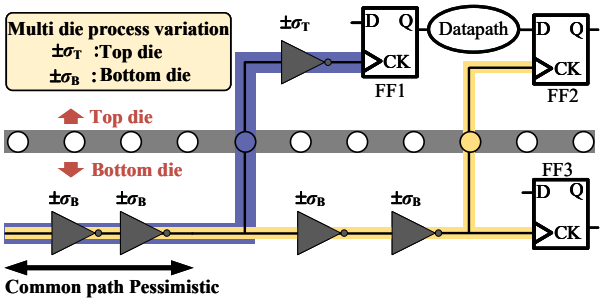


Figure 2: CPPR challenge in a Face-to-face hybrid bonded 3D-IC due to uncorrelated inter-die process variations.

based on the delay difference in the common path, which cannot simultaneously be both fast and slow. The CPPR credit is defined as:

$$\text{CPPR}_{\text{credit}} = \sum_{e \in \text{Common Path}} \left[d^{\max}(e) - d^{\min}(e) \right], \tag{1}$$

where $d^{\max}(e)$ and $d^{\min}(e)$ are the maximum and minimum delays of an edge e in the common path.

In 3D-ICs, this concept becomes even more important. Because different dies can operate at disparate Process, Voltage, and Temperature (PVT) corners, cross-die clock paths inherently possess large delay variations, making them the most significant source of potential CPPR credit. As illustrated in Figure 2, splitting the clock path early on the source die restricts the common path, thereby forfeiting the substantial timing margin that could be reclaimed from the cross-die link.

This creates a fundamental tension in 3D CTS. A conventional, skew-driven approach favors localized subtrees, forcing an early clock split for cross-die paths and sacrificing valuable CPPR credit. Conversely, a CPPR-focused strategy extends a shared clock path across the die interface to maximize timing slack, yet this deliberately unbalanced topology is notoriously difficult to integrate and risks degrading global skew. This inherent conflict exposes the inadequacy of rigid CTS methodologies, which are ill-equipped to address it. Therefore, the central challenge is to develop a sophisticated, CPPR-aware 3D CTS framework capable of navigating this trade-off by intelligently selecting topologies to co-optimize for both maximum timing benefit and robust clock network quality.

2.3 Problem Formulation

Within the context of a pseudo-3D physical design flow for Face-to-Face bonding, the problem is to construct a clock tree T. Given a set of clock sinks S distributed across two dies, a clock root r, a set of available HBTs H for cross-die routing, and multi-corner timing constraints, the objective is to jointly optimize several interdependent factors. This includes determining the tree topology, the die assignment for all clock tree buffers, selective replication across the die interface, and an efficient HBT resource allocation for the set of cross-die edges, $E_{3D} \subseteq T$, via an injective assignment function $f : E_{3D} \rightarrow H$. The goal of this co-optimization is to mitigate timing pessimism by maximizing common path sharing, minimizing clock skew and latency, and ultimately maximizing the available timing margin for data paths.

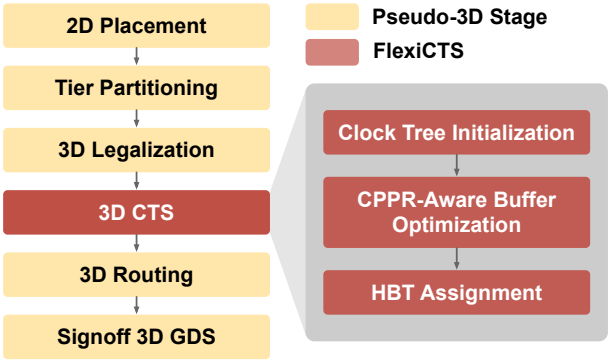


Figure 3: The pseudo-3D physical design flow, integrating FlexiCTS as a critical optimization stage for 3D CTS.

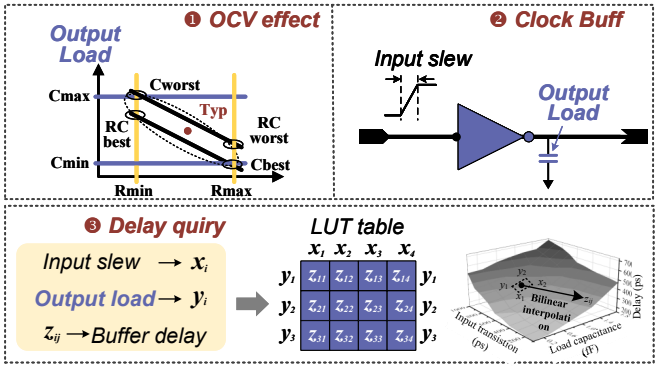


Figure 4: Buffer delay characteristics as a function of output load and input slew, influenced by OCV and hybrid bonding terminal capacitance. The 3D surface illustrates the non-linear relationship captured by the timing LUTs.

3 Methodology

3.1 Overall Flow

To address the CPPR-aware 3D CTS problem, we introduce FlexiCTS, a novel optimization framework specifically designed to address the challenges of CTS within modern 3D IC physical design flows. The overall physical design methodology in which FlexiCTS operates is illustrated in Figure 3. To enable the use of mature and powerful 2D commercial EDA tools for 3D ICs, our flow adopts the *pin-projection* technique, a paradigm first pioneered by Pin-3D [14]. This technique creates a virtual 2D layout view of the entire 3D stack, allowing for a die-by-die optimization strategy within a full 3D context.

Our flow begins by implementing the pin-projection methodology to generate an initial physical design. This preparatory phase involves standard 2D placement, followed by tier partitioning and 3D legalization. During this process, when optimizing an active die, cells on the inactive die are treated as fixed objects and are masked from the placer's view. To establish a well-defined starting point for our framework, we then synthesize an initial clock tree. Following a methodology similar to that of Pin-3D, this is accomplished by constraining all clock buffers to the top die. However, this strategy, while ensuring compatibility with 2D-based commercial tools, creates a fundamentally flawed initial state. It results in a severely unbalanced clock tree, leading to significant skew and overutilization of valuable HBT resources.

It is precisely this well-defined but flawed initial state that our FlexiCTS framework is designed to optimize. Taking the partitioned and legalized design as input, FlexiCTS executes its core optimization algorithms. The centerpiece of our contribution is the CPPR-Aware Buffer Optimization, which intelligently redistributes, replicates, and reassigns clock buffers across the two dies to minimize timing pessimism. This is followed by the HBT Assignment stage, which maps the newly determined inter-die clock connections to physical bonding resources in a timing-aware manner. Upon completion, FlexiCTS generates an optimized clock topology by providing the final 3D locations for clock buffers and the definitive HBT assignments. This pre-optimized information is then fed back into the commercial EDA flow. A 3D-aware router can then perform the final signal and clock net routing, leveraging the deterministic, low-parasitic HBT connections established by FlexiCTS. In this manner, FlexiCTS acts as a critical, CTS optimization engine that seamlessly integrates into a pin-projection-based flow, enhancing it to overcome the inherent limitations of die-by-die CTS and achieve superior 3D timing closure.

3.2 CPPR-Aware Cost Model Formulation

Our adaptive buffer assignment algorithm (detailed in Section 3.3) relies on a comprehensive cost model to navigate the complex 3D design space. This model quantifies the impact of buffer topology decisions on key performance metrics. It integrates four core dimensions: clock performance (skew and latency), CPPR pessimism, and physical resource cost (area), creating a unified framework to manage their inherent trade-offs. The foundation of our cost model is an accurate, 3D-aware delay calculation. This approach explicitly accounts for the parasitic capacitance of HBTs, a critical factor in 3D-ICs. As illustrated in Figure 4, a buffer's propagation delay is a non-linear function of its input slew and total output capacitive load, a relationship pre-characterized in standard timing library look-up tables (LUTs).

The effective capacitive load, $C_{eff}(b)$, for a buffer b driving cross-die fanouts must incorporate this parasitic load:

$$C_{eff}(b) = \sum_{i \in \text{fanouts}(b)} [C_{wire}(i) + C_{input}(i)] + n_{hbt}(b) \cdot C_{hbt}, \quad (2)$$

where $n_{hbt}(b)$ is the number of HBTs driven by buffer b, and C_{hbt} is the parasitic capacitance per HBT. This effective capacitance determines the buffer's exact operating point on its delay surface as shown in Figure 4.

With $C_{eff}(b)$ determined, we can query the propagation delays under slow and fast process corners from multi-dimensional LUTs:

$$\begin{aligned} d^{\text{slow}}(b) &= \text{LUT}_{\text{slow}}(\text{slew}_{in}(b), C_{eff}(b), \text{type}(b)) \\ d^{\text{fast}}(b) &= \text{LUT}_{\text{fast}}(\text{slew}_{in}(b), C_{eff}(b), \text{type}(b)). \end{aligned} \quad (3)$$

Based on these delays, we compute the first two components of our cost function: **clock skew**, the maximum arrival time difference among all sinks, and **maximum latency**, the longest root-to-sink path delay.

The next component of our model quantifies the impact on CPPR, which is critical for timing closure. The CPPR credit (CC) for a timing path between a launch flop F_l and a capture flop F_c is the sum of OCV-induced delay differences along their common clock path, $CP(F_l, F_c)$:

$$\text{CC}(F_l, F_c) = \sum_{b \in CP(F_l, F_c)} [d^{\text{slow}}(b) - d^{\text{fast}}(b)]. \quad (4)$$

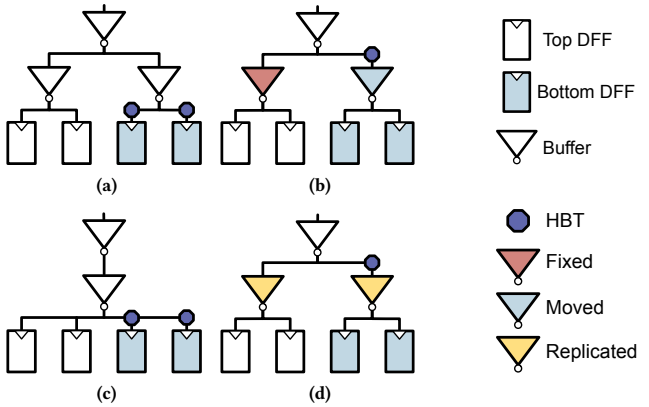


Figure 5: Illustration of the adaptive buffer assignment strategies in FlexiCTS.

A key insight of our work is that CPPR credit is not a static property but is dynamically altered by CTS decisions. Specifically, buffer replication, a standard technique for skew reduction, can be detrimental to CPPR. When a buffer b is replicated into b_{top} and b_{bot}, the clock path divergence point moves upstream. Consequently, b is removed from the common path of any cross-die timing endpoints it previously served, directly reducing their CPPR credit.

To formally capture this critical trade-off, we introduce the **CPPR Loss** metric. For a proposed topological change (e.g., replicating buffer b), the loss is the aggregated reduction in CPPR credit across all affected timing paths $\mathcal{P}_{\text{affected}}$:

$$\text{CPPR}_{\text{Loss}} = \sum_{(F_l,F_c)\in\mathcal{P}_{\text{affected}}} [\text{CC}_{\text{before}}(F_l,F_c) - \text{CC}_{\text{after}}(F_l,F_c)]. \quad (5)$$

This term quantifies the timing pessimism introduced by the replication. Our cost model uses $\text{CPPR}_{\text{Loss}}$ to penalize decisions that sacrifice valuable timing margin for marginal skew improvements, thereby enforcing a more globally optimal optimization.

Finally, we unify these metrics into a single, weighted cost function to guide our buffer assignment algorithm. For any proposed buffer configuration S, the total cost is defined as:

$$\text{Cost}(S) = w_s \cdot \text{Skew} + w_l \cdot \text{Lat}_{\text{max}} + w_c \cdot \text{CPPR}_{\text{Loss}} + w_a \cdot \Delta\text{Area}, \quad (6)$$

where ΔArea represents the change in total buffer area resulting from the configuration S. The coefficients w_s, w_l, w_c, and w_a are user-definable weights that allow designers to tune the optimization focus. This scalar cost function enables an efficient search for a globally superior solution by resolving the trade-offs between skew, latency, CPPR, and area at each decision point.

3.3 Adaptive Buffer Assignment via Dynamic Programming

This section details our algorithm for 3D clock buffer assignment, which solves the inherent multi-objective optimization problem using a bottom-up dynamic programming (DP) framework, detailed in Algorithm 1. To manage interconnect costs, we enforce a *single die-crossing constraint*, which prunes the search space by disallowing any root-to-sink path from traversing the die interface more than once. This constraint is critical for minimizing HBT usage and associated parasitics.

The algorithm iteratively refines the clock tree by processing each buffer in a reverse topological order. For each buffer, it evaluates

Algorithm 1 CPPR-Aware Adaptive Buffer Assignment

Input: Clock tree T, delay LUTs, cost weights W
Output: Optimal buffer configuration s^*

$S \leftarrow \emptyset$
for each node $v \in T$ in a bottom-up traversal **do**
  $C_v \leftarrow \emptyset$ ▷ Candidate states for v
  DFFs $\leftarrow$ Downstream_DFFs(v)
  strategies $\leftarrow$ {Top, Bottom, Replicate} ▷ Prune by DFF distribution
  for p $\in$ strategies **do**
    $state_p \leftarrow$ Eval(v, DFFs, p, S) ▷ Compute Objective Metrics
    if Valid($state_p$) **then** ▷ Check single die-crossing
      $C_v \leftarrow C_v \cup state_p$
    end if
  end for
  $S[v] \leftarrow$ Pareto_Prune(C_v) ▷ Remove dominated solutions
end for
return $\arg\min_{\{s\in S[\text{root}]\}}$

three primary placement strategies, as illustrated by the example in Figure 5(a) and Figure 5(b). Given an initial state for a subtree (Figure 5(a)), one option is to keep the buffer on its current die (the top die, highlighted in red in Figure 5(b)), which maintains the existing connection to its fanout. Alternatively, the *Move* strategy relocates the buffer to the fanout sinks' die (the bottom die, highlighted in blue). This move is valid only if all sinks reside on the same target die. By doing so, the cross-die connection is shifted one level upstream, which can reduce the total number of HBTs and their associated delay. The most complex option is the *Replicate* strategy (Figure 5(d)), which duplicates the buffer, creating instances and placing them at the same (x, y) coordinates. This allows for independent, localized skew balancing on each die. However, this approach incurs an area penalty and, more critically, shortens the common clock path for any cross-die timing endpoints. This shortening leads to a quantifiable penalty known as *CPPR Loss*, which is rigorously evaluated by our cost model (Section 3.2). The figure illustrates how replication also moves the cross-die connection upstream to feed the newly created buffer instance.

The selection among these strategies is governed by our Pareto-based DP engine, which optimizes the 3D clock tree by propagating cost information from sinks to the root. At each node, rather than prematurely scalarizing costs with a weighted-sum function, our approach embraces the problem's multi-objective nature by employing Pareto pruning. This method maintains a set of candidate solutions, preserving valuable trade-offs between multiple objectives. This process continues until the root is reached, at which point the final, globally optimal configuration s^* is selected from the root's Pareto-optimal set by applying the weighted-sum cost function from Equation (6). This process ensures the final topology is derived from a richer set of high-quality global candidates.

3.4 Hybrid Bonding Terminal Assignment

Following adaptive buffer optimization, the resultant cross-die clock nets must be assigned to physical HBTs. This assignment step is critical, as it determines the physical inter-die connection points and profoundly impacts wirelength, power, and timing. An optimal assignment minimizes RC delay, whereas a suboptimal one can introduce long routes that influence timing closure. The overall

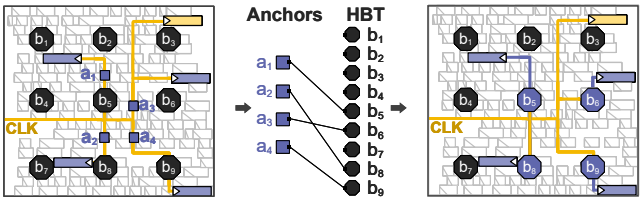


Figure 6: Schematic of the hybrid bonding assignment process.

process, illustrated schematically in Figure 6, aims to find a timing-aware assignment that minimizes wirelength and timing costs.

Let $\mathcal{N} = \{n_1, n_2, \ldots, n_k\}$ be the set of k cross-die clock nets requiring inter-die connection, and let $\mathcal{H} = \{h_1, h_2, \ldots, h_m\}$ be the set of m available HBTs. In modern F2F designs, the density of HBTs is high, leading to a scenario where the number of nets is much smaller than the available terminals ($k \ll m$). Our goal is to find an optimal assignment that balances wirelength and timing. We formulate this as a minimum-weight bipartite matching problem. The objective is to find an injective (one-to-one) mapping $\phi : \mathcal{N} \to \mathcal{H}$ that minimizes the total assignment cost:

$$\min \sum_{n_i \in \mathcal{N}} C(n_i, h_{\phi(i)}), \tag{7}$$

where $\phi(i)$ is the index of the HBT assigned to net n_i. To capture the trade-off between distance and timing, we define a timing-aware cost function, $C(n_i, h_j)$, for assigning net n_i to HBT h_j:

$$C(n_i, h_j) = d(c_i, p_j) \cdot (1 + w_t \cdot \tau_i), \tag{8}$$

Here, $d(c_i, p_j)$ is the Manhattan distance between the net's geometric centroid c_i and the HBT's position p_j, representing the wirelength cost. The centroid c_i is calculated as the center of mass of the net's driver and sink cells, representing its ideal connection point at the die interface. The term τ_i quantifies the timing criticality of net n_i. It is defined as the non-negative slack of the most critical timing path associated with the net, i.e., $\tau_i = \max(0, -S_i)$, where S_i is the slack obtained from STA. This model ensures that nets with higher timing criticality (i.e., a larger τ_i, corresponding to more negative slack) incur a significantly higher penalty for being assigned to distant HBTs. The weight w_t is a user-defined parameter to control the relative importance of timing over pure wirelength minimization.

To solve this large-scale $k \times m$ assignment problem efficiently, we employ a scalable two-stage approach. First, we perform Candidate Pruning by creating a candidate list of the top-K nearest HBTs for each net, drastically reducing the search space. Second, for the Optimal Assignment, we construct a bipartite graph from these candidates and solve it using the Jonker-Volgenant (JV) algorithm. With a worst-case complexity of $O(k^3)$, the JV algorithm is highly efficient for industrial-scale problems. We first represent each cross-die net by its anchor, based on the physical locations of its sinks (Figure 6(a)). This transforms the physical layout into an abstract bipartite matching problem between these anchors and the available HBTs (Figure 6(b)). Figure 6(c) shows the final, optimized assignment produced by our algorithm, where mappings minimize the global timing-aware cost. This assignment method integrates seamlessly with the preceding buffer optimization stage, ensuring that resource allocation decisions holistically support the overall goal of achieving timing closure.

4 Experimental Results

4.1 Experimental Setup

We evaluate three designs: AES128, JPEG [20], and ARM Cortex-A7 [21], a power-efficient commercial CPU widely deployed in mobile and embedded systems requiring balanced power-performance tradeoffs. All designs are implemented in a commercial 7*nm* technology library (VDD = 0.75V, Temp = 25°*C*) using Synopsys Design Compiler [22] for synthesis, Cadence Innovus [23] for place-and-route, and Synopsys PrimeTime [24] for timing analysis with back-annotated parasitics.

The proposed methodology was implemented using a combination of C++ for algorithms and Tcl for scripting automation. Experimental validation was conducted on a high-performance computing cluster running Rocky Linux 8.10. The computational platform was powered by a dual-processor system featuring Intel Xeon Platinum 8480+ CPUs, providing a total of 112 cores. The system was equipped with two NVIDIA L40S GPUs, each with 46GB GDDR6 memory. To ensure a consistent and controlled comparative analysis, we implemented reference versions of the two primary pseudo-3D CTS strategies, Cascade-2D [13] and Pin-3D [14], within the same experimental environment. These implementations were carefully constructed to adhere to the fundamental principles described in their respective publications.

4.2 Clock Metric Comparison

As shown in Table 1, FlexiCTS simultaneously achieves superior timing quality and minimal resource overhead, resolving the fundamental tradeoff in 3D clock tree synthesis.

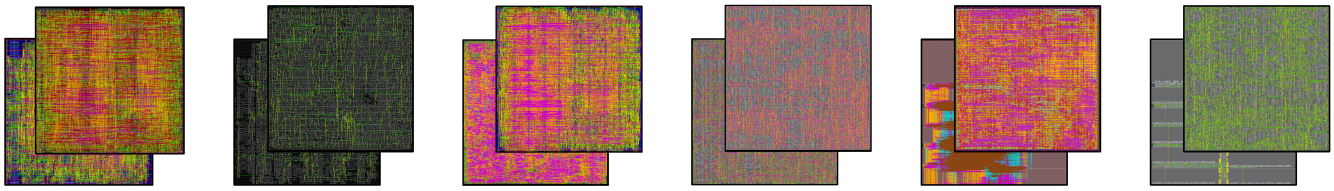
Timing superiority. FlexiCTS delivers significant improvements in clock skew control (Table 1). While the Pin-3D CTS strategy degrades skew by up to 4.3× versus the 2D baseline, FlexiCTS reverses this trend. On AES128, it achieves just 8 ps skew, reducing Cascade-2D's 26 ps by 69% and **nearly matching the 2D quality**. Across all benchmarks, FlexiCTS outperforms Pin-3D by an average factor of 4.1× (geometric mean), demonstrating the effectiveness of CPPR-aware adaptive buffering. This timing advantage extends to full-chip metrics (Table 2). On ARM Cortex-A7, where Pin-3D degrades setup/hold TNS to 2.86×/4.59× of baseline, FlexiCTS *improves* both to 0.86×/0.81×, **surpassing even the 2D baseline** despite 50% area reduction.

Resource efficiency. The Pin-3D CTS strategy demands 16,954 HBTs for ARM Cortex-A7's clock network, while FlexiCTS requires just 2,034, an **88% reduction**. This frees thousands of critical HBTs for signal/power routing, directly improving routability. Simultaneously, FlexiCTS minimizes silicon overhead: its 122 μm^2 buffer area on ARM is 46% lower than Cascade-2D's 227 μm^2, while delivering superior timing.

The paradigm shift. Unlike the Cascade-2D strategy (which sacrifices timing for HBT savings) or Pin-3D strategy (which sacrifices resources for common paths), FlexiCTS optimizes both through CPPR-aware co-design. As confirmed in Table 2, FlexiCTS consistently achieves the lowest power (0.85–0.87×), best timing margins, and competitive wirelength across all benchmarks.

Table 1: Comparison of clock metrics among different CTS strategies: 2D baseline, Cascade-2D (C2D) [13], Pin-3D (P3D) [14], and FlexiCTS (Ours).

Design Metrics	AES128				JPEG				ARM Cortex-A7			
	2D	C2D	P3D	FlexiCTS	2D	C2D	P3D	FlexiCTS	2D	C2D	P3D	FlexiCTS
#FFs (Top/Bot.)	889	312/577	312/577	312/577	37340	20386/16954	20386/16954	20386/16954	14053	8431/5622	8431/5622	8431/5622
#Buffers (Top/Bot.)	180	85/135	180/0	87/102	7670	5234/4619	7670/0	4735/3990	1497	908/1061	1497/0	825/806
Area Over. (μm^2)	0	4.63	0	**1.15**	0	9.63	0	**5.21**	0	227.16	0	**122.12**
Clock Skew (ps)	9	26	39	**8**	68	88	126	**81**	86	97	147	**94**
Clock Latency (ps)	57	196	179	**60**	192	256	468	**205**	250	399	299	**258**
#HBTs (Clock)	0	0	577	**81**	0	0	5622	**844**	0	0	16954	**2034**



(a) AES128 (b) Clock tree of AES128 (c) JPEG (d) Clock tree of JPEG (e) ARM Cortex-A7 (f) Clock tree of ARM

Figure 7: Final layouts and corresponding clock trees for the (a,b) AES128, (c,d) JPEG, and (e,f) ARM Cortex-A7 benchmarks, as generated by the FlexiCTS framework.

Table 2: Full-chip PPA comparison across different CTS strategies, with all metrics normalized to the 2D baseline.

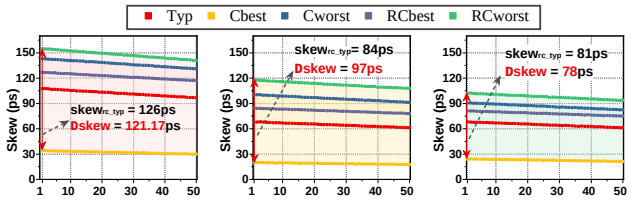
Design	Method	Footprint (μm^2)	WL (μm)	Total Power (mW)	Setup WNS (ps)	Setup TNS (ns)	Hold WNS (ns)	Hold TNS (ns)
AES128	2D	1	1	1	1	1	1	1
	C2D	0.52	0.92	0.92	0.91	1.72	1.73	1.24
	P3D	0.52	0.77	0.93	0.89	1.65	1.73	1.41
	FlexiCTS	0.52	0.74	0.85	0.76	0.71	0.59	0.72
JPEG	2D	1	1	1	1	1	1	1
	C2D	0.55	0.98	0.90	1.03	1.23	0.99	1.09
	P3D	0.55	0.96	0.91	0.91	2.35	2.83	2.69
	FlexiCTS	0.55	0.94	0.87	0.74	0.82	0.69	0.71
Cortex-A7	2D	1	1	1	1	1	1	1
	C2D	0.50	0.95	0.94	0.96	1.22	1.09	1.14
	P3D	0.50	0.89	0.91	0.87	2.86	4.59	5.25
	FlexiCTS	0.50	0.88	0.86	0.77	0.86	0.81	0.72

4.3 Design-Level PPA Comparison

The full-chip PPA results in Table 2 confirm the holistic benefits of FlexiCTS. While all pseudo-3D methods achieve the expected ~50% footprint reduction by stacking dies, FlexiCTS consistently achieves the lowest total power and wirelength across all benchmarks. Most critically, FlexiCTS overcomes the severe timing degradation problem that was common in previous 3D-CTS strategies. For instance, on the ARM Cortex-A7 benchmark, the Pin-3D CTS strategy degrades setup and hold TNS to 4.59× and 5.25× the 2D baseline, indicating a design that would fail timing closure. In contrast, our CPPR-aware optimization actively improves timing, reducing setup and hold TNS to 0.81× and 0.72× of the baseline. Figure 7 shows the final layouts and clock tree structures generated by our framework, showcasing the physical implementation of our optimizations.

4.4 CPPR-Awareness Validation

To validate our CPPR-aware optimization, we conducted an ablation study on the JPEG benchmark, focusing on clock skew's robustness across five standard signoff corners (RC). As shown in Figure 8, the baseline Pin-3D strategy is highly susceptible to process corners, yielding a large cross-corner skew variation (Δskew) of 121.17 ps.



(a) Pin-3D (b) Ours w.o. CPPR (c) Ours w. CPPR

Figure 8: Illustration of the impact of CPPR-aware optimization on clock skew across process corners.

A CPPR-agnostic version of our framework, relying only on flexible topology, reduced this to 97 ps, demonstrating that flexibility alone is insufficient. In contrast, the complete FlexiCTS framework, guided by our CPPR-aware model, strategically maximizes common path length. This leverages CPPR to mitigate variation, achieving a low Δskew of just 78 ps, which achieves a 19.6% reduction over the agnostic version. This result confirms that our CPPR-aware model is not merely an optimization but an essential component for achieving robust and predictable timing in 3D ICs.

5 Conclusion

This paper introduces FlexiCTS to address the trade-off between CPPR and bonding resources in pseudo-3D CTS flows for F2F hybrid-bonded ICs. Our novel framework uses a multi-objective dynamic programming algorithm to optimize buffer placement and replication across dies, maximizing common clock path sharing. Complemented by an efficient terminal assignment, FlexiCTS holistically improves the 3D clock network. Experiments confirm our approach yields significant clock skew reduction and substantial PPA improvements over existing methods.

6 Acknowledgements

This research was partially conducted by ACCESS-AI Chip Center for Emerging Smart Systems, supported by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government. This work is jointly supported by the Research Grants Council of Hong Kong SAR No. CUHK14211324 and No. T46-415/25-R.

References

[1] D. W. Fisher, S. Knickerbocker, D. Smith, R. Katz, J. Garant, J. Lubguban, V. Soler, and N. Robson, "Face to Face Hybrid Wafer Bonding for Fine Pitch Applications," in *IEEE Electronic Components and Technology Conference*, 2020.

[2] J. Wuu, R. Agarwal, M. Ciraula, C. Dietz, B. Johnson, D. Johnson, R. Schreiber, R. Swaminathan, W. Walker, and S. Naffziger, "3D V-Cache: the Implementation of a Hybrid-Bonded 64MB Stacked Cache for a 7nm x86-64 CPU," in *IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65, 2022.

[3] T. F. Wu, H. Liu, H. E. Sumbul, L. Yang, D. Baheti, J. Coriell, W. Koven, A. Krishnan, M. Mittal, M. T. Moreira *et al.*, "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 *IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67, 2024.

[4] S. Pentapati, A. Agnesina, M. Brunion, Y.-H. Huang, and S. K. Lim, "On Legalization of Die Bonding Bumps and Pads for 3D ICs," in *ACM International Symposium on Physical Design (ISPD)*, 2023.

[5] S. Chen, Y. Qiu, B. Xie, M. Chen, and X. Li, "A Fast, Iterative Clock Skew Scheduling Algorithm with Dynamic Sequential Graph Extraction," in *ACM/IEEE Design Automation Conference (DAC)*, 2025.

[6] W.-H. Liu, H.-T. Chang, A. Agnesina, R. Liang, A. Rajaram, and H. Ren, "Hybrid Structured Clock Network Construction with GPUs for Large Designs," in *International VLSI Symposium on Technology, Systems and Applications (VLSI TSA)*, 2025.

[7] D. K. Oh, M. J. Choi, and J. H. Kim, "Thermal-aware 3D Symmetrical Buffered Clock Tree Synthesis," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 24, no. 3, pp. 1–22, 2019.

[8] B. Lu, J. Hu, G. Ellis, and H. Su, "Process Variation Aware Clock Tree Routing," in *ACM International Symposium on Physical Design (ISPD)*, 2003.

[9] A. Rajaram, B. Lu, J. Hu, R. Mahapatra, and W. Guo, "Analytical Bound for Unwanted Clock Skew Due to Wire Width Variation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 25, no. 9, pp. 1869–1876, 2006.

[10] Z. Guo, M. Yang, T.-W. Huang, and Y. Lin, "A provably good and practically efficient algorithm for common path pessimism removal in large designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 41, no. 10, pp. 3466–3478, 2021.

[11] T.-W. Huang, P.-C. Wu, and M. D. Wong, "Fast Path-based Timing Analysis for CPPR," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014.

[12] T.-W. Huang and M. D. Wong, "UI-timer 1.0: An Ultrafast Path-based Timing Analysis Algorithm for CPPR," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 35, no. 11, pp. 1862–1875, 2016.

[13] K. Chang, S. Sinha, B. Cline, G. Yeric, and S. K. Lim, "Design-Aware Partitioning-Based 3-D IC Design Flow With 2-D Commercial Tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 41, no. 3, pp. 410–423, 2021.

[14] S. S. K. Pentapati, K. Chang, V. Gerousis, R. Sengupta, and S. K. Lim, "Pin-3D: A Physical Synthesis and Post-Layout Optimization Flow for Heterogeneous Monolithic 3D ICs," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020.

[15] X. Zhao, J. Minz, and S. K. Lim, "Low-Power and Reliable Clock Network Design for Through-Silicon Via (TSV) Based 3D ICs," *IEEE Transactions on Components, Packaging and Manufacturing Technology (TCPMT)*, vol. 1, no. 2, pp. 247–259, 2010.

[16] T.-Y. Kim and T. Kim, "Clock Tree Synthesis for TSV-based 3D IC Designs," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 16, no. 4, pp. 1–21, 2011.

[17] J. Zhou, Z. Ge, C. Xi, and P. Zhou, "Inductance-aware Clock Network Synthesis Considering Hierarchical Interconnects in 3D ICs," in *ACM Great Lakes Symposium on VLSI (GLSVLSI)*, 2025.

[18] P. Vanna-iampikul, J. Yoon, C. Park, G. Yeap, and S. K. Lim, "Placement-Aware 3D Net-to-Pad Assignment for Array-Style Hybrid Bonding 3D ICs," in *ACM International Symposium on Physical Design (ISPD)*, 2025.

[19] J. Kim, L. Zhu, H. M. Torun, M. Swaminathan, and S. K. Lim, "Micro-Bumping, Hybrid Nonding, or Monolithic? A PPA Study for Heterogeneous 3D IC Options," in *ACM/IEEE Design Automation Conference (DAC)*, 2021.

[20] "OpenCores." 2021, https://opencores.org/.

[21] Cortex-A7 Processor, https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a7.

[22] "Synopsys design compiler user guide," 2022, Guide,http://www.synopsys.com.

[23] Cadence, "Innovus Reference Manual", https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/soc-implementation-and-floorplanning/innovus-implementation-system.html, 2021.

[24] Synopsys, "Primetime User Guide", https://www.synopsys.com/implementation-and-signoff/signoff/primetime.html, 2021.