

Near-Optimal TDM Ratio Assignment for Die-Level Routing in Multi-FPGA Systems

Jiawei Lin, Longkun Guo*, and Weijie Fang

School of Mathematics and Statistics, Fuzhou University, Fujian, China

lkguo@fzu.edu.cn

Abstract—Modern multi-FPGA systems often integrate multiple dies to expand logic capacity and address the increasing complexity of integrated circuit designs. To overcome the limitations of physical I/O pins, these systems typically employ time-division multiplexing (TDM) technology. However, higher TDM ratios introduce considerable signal delays, resulting in higher critical connection delays. This paper focuses on optimizing the TDM ratio to tackle this challenge. We formulate the TDM ratio assignment problem as a block-angular convex program and solve it using Lagrangian decomposition, obtaining a $(1 + \epsilon)$ -approximate solution for any given $\epsilon > 0$. We further introduce a delay-aware TDM wire assignment scheme to achieve efficient signal assignment. Experimental results demonstrate that our method enables efficient, high-quality die-level routing in modern multi-FPGA systems, achieving up to 10.8% reduction in critical connection delay compared to the state-of-the-art approaches.

Index Terms—Die-level multi-FPGA systems, Time-division multiplexing, Lagrangian decomposition

I. INTRODUCTION

With the growing complexity of large-scale integrated circuit (IC) designs [1], a single field-programmable gate array (FPGA) can no longer provide sufficient logic capacity or performance. Consequently, multi-FPGA systems have become an important platform for IC logic emulation and prototyping [2]. To further align with larger design scales and higher operating frequencies, 2.5D packaging technology has integrated multiple dies within a single FPGA package [3], forming die-level multi-FPGA systems that significantly enhance logic resources.

Fig. 1(a) illustrates a die-level multi-FPGA system, where each FPGA comprises multiple dies interconnected through a limited number of Super Long Lines (SLLs) [4], [5]. Inter-FPGA communication occurs through physical inter-die connections (Fig. 1(b)), constrained by the limited I/O pins [6]. To improve bandwidth utilization, time-division multiplexing (TDM) is widely employed in multi-FPGA systems [7]. TDM partitions each clock cycle into multiple time slots and assigns them to different signals, so that multiple signals can share the same physical wires and pins, resulting in significantly enhanced channel utilization. Fig. 1(c) illustrates that TDM slot allocation may vary across different physical wires.

In multi-FPGA systems, system routing is a critical step for generating the routing topology and assigning TDM ratios to inter-FPGA signals [8]. While traditional FPGA-level routing

This work is supported by National Natural Science Foundation of China (No. 12271098), Guangzhou Leading Science and Technology Talent Program (No. 2025A04J7076), and Key Project of the Natural Science Foundation of Fujian Province (No. 2025J02011).

*Corresponding author.

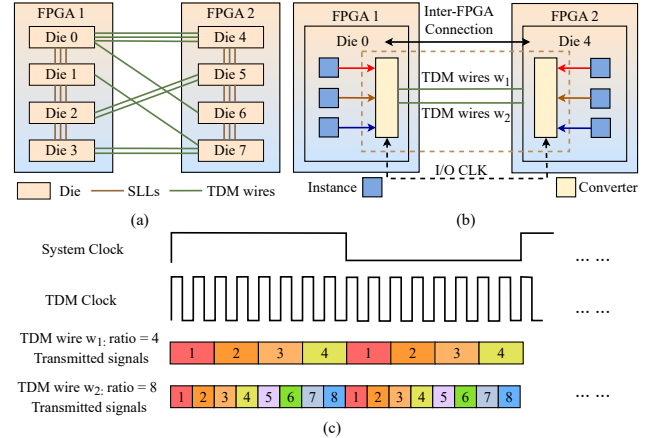


Fig. 1. (a) A multi-FPGA system with 2 FPGAs and 8 dies, interconnected via multiple physical wires (SLLs or TDM wires); (b) TDM I/O implementation; (c) Clock waveform and transmission schedule of signals on the two wires connecting Die 0 and Die 4.

treats each FPGA as a single monolithic device and focuses solely on inter-FPGA connections, die-level routing requires precise signal routing across multiple dies. Moreover, inter-FPGA communication typically relies on TDM to share limited channels, which introduces additional signal delays [9] and increases the overall problem complexity.

Most existing work focuses on FPGA-level routing [10] and on optimizing the maximum total TDM ratio of NetGroups. Lin et al. [11] used an approximate Steiner tree [12] for initial routing and Lagrangian relaxation for TDM ratio assignment. Zou et al. [13] proposed a weighted Steiner tree routing with a redistribution strategy to address net imbalance during TDM ratio assignment. Zheng et al. [14] proposed a hybrid maze [15] and fast minimum terminal spanning tree [16] routing for multi-FPGA systems, with competition-based TDM ratio assignment. Zhuang et al. [17] employed Dijkstra-based routing for inter-FPGA nets and system-level delay optimization to minimize the maximum TDM ratio of NetGroups. Lin et al. [18] introduced a sequential method that first generates high-quality routing topologies through pre-routing and congestion evaluation, followed by network-based TDM ratio allocation.

In the academic literature on die-level routing, Huang et al. [3] combined the minimum Steiner tree algorithm with a maze routing algorithm for the routing and applied dynamic programming to assign TDM ratios. However, their method lacks scalability for large designs. Wang et al. [8] proposed a balanced routing algorithm that jointly optimizes connection

delay and resource usage on SLL and TDM edges, using Lagrangian relaxation [19] and margin-aware TDM ratio legalization, though without approximation guarantees.

In this paper, we propose a scheme for TDM ratio and TDM wire assignment in die-level multi-FPGA systems, aiming to minimize the critical connection delay. The main contributions of this work are summarized as follows:

- We formulate the initial TDM ratio assignment problem as a block-angular convex program, ensuring a provable $(1 + \epsilon)$ -approximation guarantee for any given $\epsilon > 0$, and employ a Lagrangian decomposition algorithm to approximate the optimal solution.
- We propose a delay-aware TDM wire assignment algorithm that assigns signals to physical TDM wires while minimizing local delay increments.
- Compared with the state-of-the-art method, our approach achieves a 10.8% reduction in critical connection delay on die-level multi-FPGA routing contest benchmarks.

The rest of this paper is organized as follows: Section II introduces the preliminaries. Section III presents the die-level routing method. Section IV details the initial TDM ratio assignment and TDM wire assignment algorithms. Section V reports experimental results, and Section VI concludes the paper.

II. PRELIMINARIES

This section introduces the die-level multi-FPGA system model, presents its design constraints, defines the routing problem, and outlines our proposed framework.

A. Die-Level Multi-FPGA System Model

In this work, the die-level multi-FPGA system is represented as an undirected graph $G(V, E)$, with a corresponding set of netlists N specified as follows:

- Each vertex $v \in V$ represents a distinct die in the system. Each edge $e \in E$ corresponds to a physical connection between two dies, with capacity Cap_e denoting the number of physical wires in this connection. We define \mathbb{E}_{SLL} and \mathbb{E}_{TDM} as the sets of SLL and TDM edges, respectively.
- The netlist set N contains the nets to be routed, where each net $n \in N$ has a single driver node and a set of load nodes. The dies hosting these nodes must be connected through edges $e \in E$.
- For each net n , the sequence of routing edges traversed from its driver node to a load node forms a routing path p . The set of all such paths for net n is denoted by P_n , and the set of all paths in the system is $P = \bigcup_{n \in N} P_n$.

B. Design Rules

The die-level multi-FPGA routing problem is subject to the following constraints:

- (1) **SLL edge constraints.** Each physical wire on an SLL edge can carry at most one signal. Thus, for any $e \in \mathbb{E}_{\text{SLL}}$, the number of routed signals must not exceed Cap_e .
- (2) **TDM ratio constraints.** Let r_{ne} denote the TDM ratio of the signal associated with net n on TDM edge e . Each

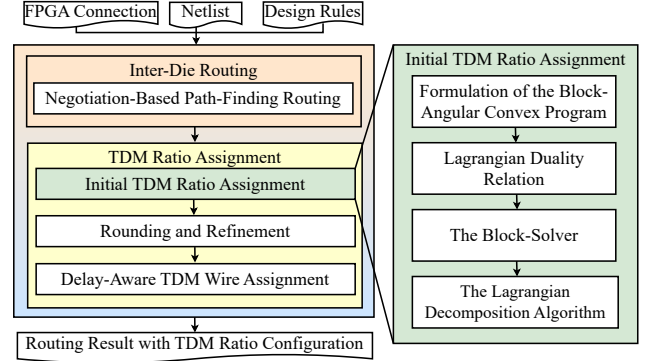


Fig. 2. Overview of the algorithmic framework.

r_{ne} must be an integer multiple of the TDM ratio step Δ_{TDM} . For an edge e with capacity Cap_e , the sum of the reciprocals of the TDM ratios of all multiplexed signals must not exceed Cap_e .

- (3) **TDM wire constraints.** An edge $e \in \mathbb{E}_{\text{TDM}}$ consists of Cap_e parallel physical wires. For each physical wire, the signals assigned to it must have the same r_{ne} , and the sum of the reciprocals of these ratios must not exceed 1. All signals on the same physical TDM wire must be transmitted in the same direction.
- (4) **Delay rules.** The delay of a path $p \in P$ is defined as the sum of the delays of all edges composing that path. For $e \in \mathbb{E}_{\text{SLL}}$, the signal delay is a constant d_{SLL} . For $e \in \mathbb{E}_{\text{TDM}}$, the signal delay is expressed as $d_0 + d_1 \cdot r_{ne}$, where n is the net to which path p belongs and d_0 and d_1 are constants.

C. Die-Level Multi-FPGA Routing Problem Formulation

The problem can be formulated as follows:

Given: A graph $G(V, E)$, a set N of nets.

Task: Find the routing paths from driver to load nodes for each net, assign TDM ratios to the signals on edges $e \in \mathbb{E}_{\text{TDM}}$, and assign the signals to specific physical wires, with the objective of satisfying the design rules and minimizing the critical connection delay:

$$\min \max_{p \in P} \left(\sum_{e \in (p \cap \mathbb{E}_{\text{TDM}})} (d_0 + d_1 \cdot r_{ne}) + d_{\text{SLL}}^p \right),$$

$$\text{where } d_{\text{SLL}}^p = \sum_{e \in (p \cap \mathbb{E}_{\text{SLL}})} d_{\text{SLL}}.$$

D. Overview

The overall workflow of our approach is illustrated in Fig. 2 and consists of two main stages:

- **Inter-Die Routing:** Each multi-fanout net is decomposed into die-to-die connections and routed using the negotiation-based path-finding algorithm, which satisfies the **SLL edge constraints**.
- **TDM Ratio Assignment:** We formulate the initial TDM ratio assignment as a block-angular convex program and solve it using a Lagrangian decomposition method, yielding a $(1 + \epsilon)$ -approximate solution. Then, a delay-aware

TDM wire assignment algorithm is employed to assign signals to physical wires, subject to the **TDM ratio constraint** and **TDM wire constraint**.

III. DIE-LEVEL ROUTING IN MULTI-FPGA SYSTEMS

In the die-level routing stage, we follow the weight and cost calculation formulations proposed by Wang [8]. In this approach, each multi-fanout net is decomposed into die-to-die connections, which are routed in descending order of their routing weights. These weights are precomputed using the Floyd–Warshall algorithm [20].

Routing is performed iteratively using a negotiation-based path-finding algorithm [21], [22], which determines a path for each connection while estimating the costs of both SLL and TDM edges to balance resource utilization and connection delay. The edge cost in [8] is defined as:

$$\text{cost}_e = \begin{cases} \kappa w_e, & e \in \mathbb{E}_{\text{SLL}} \\ \kappa \left(d_0 + \Delta_{\text{TDM}} + \frac{\text{demand}_e}{\text{Cap}_e} \right), & e \in \mathbb{E}_{\text{TDM}} \end{cases}$$

where the SLL edge weight w_e is set to 1 if the number of nets on the die is less than half of the SLL capacity, and to $|V|+1$ otherwise. The factor $\kappa \in (0, 1]$ balances edge reuse and connection delay, while demand_e denotes the number of signals routed through edge e . By iteratively increasing the costs of congested edges, the algorithm produces routing solutions that satisfy the SLL edge capacity constraint.

IV. TDM RATIO ASSIGNMENT OPTIMIZATION

In this section, we formulate the initial TDM ratio assignment problem as a block-angular convex program and then solve it using the Lagrangian decomposition method with theoretical guarantees. Subsequently, the proposed delay-aware TDM wire assignment algorithm is applied to assign signals to physical TDM wires.

A. Block-Angular Convex Programming

For each TDM edge $e \in \mathbb{E}_{\text{TDM}}$, the feasible TDM ratio is

$$X^e = \left\{ x^e = (r_{ne})_{n \in \mathbb{N}_e} \mid r_{ne} \in [\Delta_{\text{TDM}}, \infty) \right\},$$

where \mathbb{N}_e denotes the set of nets traversing edge e . Each vector $x^e \in X^e$ represents a possible assignment of TDM ratios r_{ne} for the nets using edge e . The global assignment is

$$x = (x^1, \dots, x^{|\mathbb{E}_{\text{TDM}}|})^\top, \quad x \in X = \prod_{e \in \mathbb{E}_{\text{TDM}}} X^e.$$

Here, \prod denotes the Cartesian product of X^e .

For each path p , let $m_p = |p \cap \mathbb{E}_{\text{TDM}}|$. If $m_p > 0$ define

$$\tilde{d}_c^e = d_0 + \frac{d_{\text{SLL}}^p}{m_p}, \quad f_p^e(x^e) = \tilde{d}_c^e + d_1 r_{ne},$$

Accordingly, the connection delay of path p is expressed by:

$$f_p(x) = \sum_{e \in (p \cap \mathbb{E}_{\text{TDM}})} (d_0 + d_1 \cdot r_{ne}) + d_{\text{SLL}}^p = \sum_{e \in (p \cap \mathbb{E}_{\text{TDM}})} f_p^e(x^e).$$

Introducing an auxiliary variable λ , the problem can be equivalently written as:

$$\lambda^* = \min \left\{ \lambda \mid \begin{array}{l} \sum_{e \in (p \cap \mathbb{E}_{\text{TDM}})} f_p^e(x^e) \leq \lambda, \quad \forall p \in P, \\ x^e \in X^e, \quad \forall e \in \mathbb{E}_{\text{TDM}} \end{array} \right\}$$

Lemma 1: The above optimization problem exhibits a block-angular convex structure.

This follows directly from the conditions for block-angular convex programs established in [23], [24].

B. Lagrangian Duality Relation and Block Solver

Let $f^e(x^e) = (f_1^e(x^e), \dots, f_P^e(x^e))^T : X^e \rightarrow \mathbb{R}^{|P|}$ and $f(x) = \sum_{e \in \mathbb{E}_{\text{TDM}}} f^e(x^e) : X \rightarrow \mathbb{R}^{|P|}$. Then the problem can be written as

$$\begin{aligned} \lambda^* &= \min_{x \in X} \{ \lambda(x) = \max \{ f_1(x), \dots, f_P(x) \} \} \\ &= \min_{x \in X} \{ \lambda \mid f(x) \leq \lambda \mathbf{1}_P \}. \end{aligned} \quad (1)$$

By von Neumann's minimax theorem [25], λ^* admits the following equivalent saddle-point representation:

$$\lambda^* = \min_{x \in X} \max_{y \in Y} L(x, y) = \max_{y \in Y} \min_{x \in X} L(x, y),$$

where the dual variable y belongs to the unit simplex

$$Y = \{ y \in \mathbb{R}^{|P|} \mid \mathbf{1}_P^T y = 1, y \geq 0 \}.$$

The Lagrangian is

$$L(x, y) = \sum_{e \in \mathbb{E}_{\text{TDM}}} L^e(x^e, y) = \sum_{e \in \mathbb{E}_{\text{TDM}}} y^T f^e(x^e) = y^T f(x).$$

For each block e , define the local problem

$$I^e(y) = \min_{x^e \in X^e} L^e(x^e, y) = \min_{x^e \in X^e} y^T f^e(x^e).$$

Then the dual value decomposes:

$$\lambda^* = \max_{y \in Y} I(y) = \max_{y \in Y} \sum_{e \in \mathbb{E}_{\text{TDM}}} I^e(y).$$

To solve these block problems efficiently, we propose a block solver which takes $y \in \mathbb{R}_+^{|P|}$ as input and returns $x^e \in X^e$ such that $I^e(y) = y^T f^e(x^e) \leq \sigma \cdot \text{opt}_e(y)$, where $\text{opt}_e(y)$ denotes the optimal block value and $\sigma \geq 1$ is the approximation factor. The block solver approximately minimizes the block Lagrangian associated with edge e :

$$\begin{aligned} \min y^\top f^e(x^e) &= \min \sum_{p \in P} y_p \cdot f_p^e(x^e) \\ &= \min \sum_{n \in \mathbb{N}_e} \sum_{p \in P_n^e} y_p \cdot (\tilde{d}_c^e + d_1 r_{ne}) \\ &= \min \sum_{n \in \mathbb{N}_e} \sum_{p \in P_n^e} y_p \cdot d_1 r_{ne} + C. \end{aligned}$$

where y_p is the dual weight of path p , $P_n^e = \{ p \in P_n \mid e \in p \}$ and C is constant with respect to x^e .

Algorithm 1: Lagrangian Decomposition Algorithm

Input: An initial solution $x_0 \in X$ and a threshold parameter $\epsilon > 0$.

Output: TDM ratio assignment r_{ne} for each edge e with each net n .

```

1 Initialize  $x = x_0, \underline{\lambda} = 0, \bar{\lambda} = \lambda(x)$ ;
2 while  $(\bar{\lambda} - \underline{\lambda}) > \epsilon \bar{\lambda}$  do
3    $\delta = \bar{\lambda} - \underline{\lambda}, \lambda_1 = \underline{\lambda} + \frac{\delta}{3}, \lambda_2 = \underline{\lambda} + \frac{2\delta}{3}$ ;
4    $\nu = \frac{1}{2\underline{\lambda} + \lambda}, t = \frac{\nu\delta}{7}, \tau = \frac{4t^2}{P(25+10t)}$ ;
5   while True do
6      $\Phi(\theta, f) = \theta - \frac{t}{P} \sum_{p \in P} \ln(\theta - f_p(x))$ ;
7      $\theta(f) = \arg \min\{\Phi(\theta, f) \mid \lambda(x) < \theta < +\infty\}$ ;
8      $y_p = \frac{t}{P} \frac{1}{\theta(f) - f_p(x)}, \forall p = 1, \dots, P$ ;
9     Compute  $\hat{x}^e \in X^e$  for each  $e \in \mathbb{E}_{\text{TDM}}$ ;
10    if  $y^\top f(\hat{x}) \geq \theta(f) - 2t$  then
11      break;
12    else
13       $x = (1 - \tau)x + \tau \cdot \hat{x}$ ;
14  if  $\lambda(\hat{x}) \leq \lambda_2$  then
15     $x = \hat{x}, \bar{\lambda} = \lambda(\hat{x})$ ;
16  else
17     $\underline{\lambda} = \lambda_1$ ;
```

Let $\eta_{ne} = \sum_{p \in P_e} y_p \cdot d_1$. Incorporating the edge capacity constraint, the optimization problem solved by the block solver can be written as:

$$\begin{aligned} \min \quad & \sum_{n \in \mathbb{N}_e} \eta_{ne} \cdot r_{ne} + C \\ \text{s.t.} \quad & \sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}} \leq \text{Cap}_e, \\ & r_{ne} \geq \Delta_{\text{TDM}}, \forall n \in \mathbb{N}_e. \end{aligned}$$

Lemma 2: The above optimization problem admits an optimal solution given by

$$r_{ne}^* = \frac{\sum_{n' \in \mathbb{N}_e} \sqrt{\eta_{n'e}}}{\sqrt{\eta_{ne}} \text{Cap}_e}, \quad \forall n \in \mathbb{N}_e,$$

provided that $r_{ne}^* \geq \Delta_{\text{TDM}}$ for all net $n \in \mathbb{N}_e$. If some variables violate this condition, i.e., $r_{ne}^* < \Delta_{\text{TDM}}$, they are fixed at Δ_{TDM} , and the residual capacity is updated accordingly. Reapplying the same formula to the reduced set yields the optimal solution. Under this procedure, the block solver achieves exact optimality.

Proof: Introducing the Lagrangian multiplier $\mu \geq 0$ for the capacity constraint, we obtain

$$L(r, \mu) = \sum_{n \in \mathbb{N}_e} \eta_{ne} \cdot r_{ne} + C + \mu \left(\sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}} - \text{Cap}_e \right).$$

The first-order condition is

$$\frac{\partial L}{\partial r_{ne}} = \eta_{ne} - \frac{\mu}{r_{ne}^2} = 0 \quad \Rightarrow \quad r_{ne} = \sqrt{\frac{\mu}{\eta_{ne}}}.$$

Enforcing the constraint $\sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}} \leq \text{Cap}_e$ gives

$$\sum_{n' \in \mathbb{N}_e} \frac{\sqrt{\eta_{n'e}}}{\sqrt{\mu}} = \text{Cap}_e \quad \Rightarrow \quad \sqrt{\mu} = \frac{\sum_{n' \in \mathbb{N}_e} \sqrt{\eta_{n'e}}}{\text{Cap}_e}.$$

Hence,

$$r_{ne}^* = \frac{\sum_{n' \in \mathbb{N}_e} \sqrt{\eta_{n'e}}}{\sqrt{\eta_{ne}} \text{Cap}_e}.$$

If $r_{ne}^* < \Delta_{\text{TDM}}$ for some n , set $r_{ne}^* = \Delta_{\text{TDM}}$ and reduce the effective capacity as:

$$\text{Cap}' = \text{Cap}_e - \frac{|F_\Delta|}{\Delta_{\text{TDM}}},$$

where F_Δ is the set of fixed indices. Reapplying the formula to the remaining variables preserves feasibility and optimality. Consequently, the block solver achieves 1-optimality. \blacksquare

C. Lagrangian Decomposition Algorithm

We formulate the TDM ratio assignment problem as a block-angular convex program, and design a block solver that optimally solves each subproblem therein. Leveraging Lagrangian decomposition [24], we solve the primal-dual formulation with a provable approximation guarantee. Specifically, for any $\epsilon > 0$, a $(1 + \epsilon)$ -approximate solution can be obtained within $O\left(P(\ln P + \epsilon^{-2} \ln \epsilon^{-1})\right)$ iterations [24], and the resulting solution $x \in X$ satisfies $f(x) \leq (1 + \epsilon)\lambda^* \mathbf{1}_p$.

At each iteration of the Lagrangian decomposition algorithm, a dual vector $y \in Y$ is computed from the current solution $x \in X$ according to the coupling constraints. Each block problem is then solved independently using the block solver to obtain $\hat{x}^e \in X^e$, and the current solution is updated through a convex combination $(1 - \tau)x^e + \tau \cdot \hat{x}^e$ with a suitable step $\tau \in (0, 1]$.

The complete procedure is summarized in Algorithm 1. As the initial solution, we set $r_{ne} = \max\left(\frac{|\mathbb{N}_e|}{\text{Cap}_e}, \Delta_{\text{TDM}}\right)$. The ternary search procedure (lines 2–4 and 14–17) iteratively narrows the interval $[\underline{\lambda}, \bar{\lambda}]$ to approximate the optimal λ . The function $f(x)$ is scaled iteratively using the dual vector y (lines 5–13), which is calculated using a logarithmic potential function based on the current solution x (lines 6–8). The block solver then computes the updated solution \hat{x} corresponding to y (line 9). Finally, the current solution x is updated as a convex combination of x and \hat{x} (line 13) until the stopping condition on $f(\hat{x})$ in line 10 is satisfied.

Lemma 3: Let $x_i^e, x_j^e \in X^e$ be two feasible TDM ratio assignments for edge e , satisfying $\sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}} \leq \text{Cap}_e$. Then their convex combination $x^e = (1 - \tau)x_i^e + \tau \cdot x_j^e$, for any $\tau \in [0, 1]$, also satisfies the same constraint.

Proof: The combined TDM ratio $n \in \mathbb{N}_e$ is:

$$r_{ne} = (1 - \tau)r_{ne}^i + \tau \cdot r_{ne}^j.$$

Since $h(t) = \frac{1}{t}$ is convex for $t \in [\Delta_{\text{TDM}}, \infty)$, we have:

$$\begin{aligned} \frac{1}{r_{ne}} &\leq (1 - \tau) \cdot \frac{1}{r_{ne}^i} + \tau \cdot \frac{1}{r_{ne}^j}, \\ \Rightarrow \sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}} &\leq (1 - \tau) \cdot \sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}^i} + \tau \cdot \sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}^j}, \\ \Rightarrow \sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}} &\leq (1 - \tau) \cdot \text{Cap}_e + \tau \cdot \text{Cap}_e = \text{Cap}_e. \end{aligned}$$

Thus, the convex combination x^e remains feasible. \blacksquare

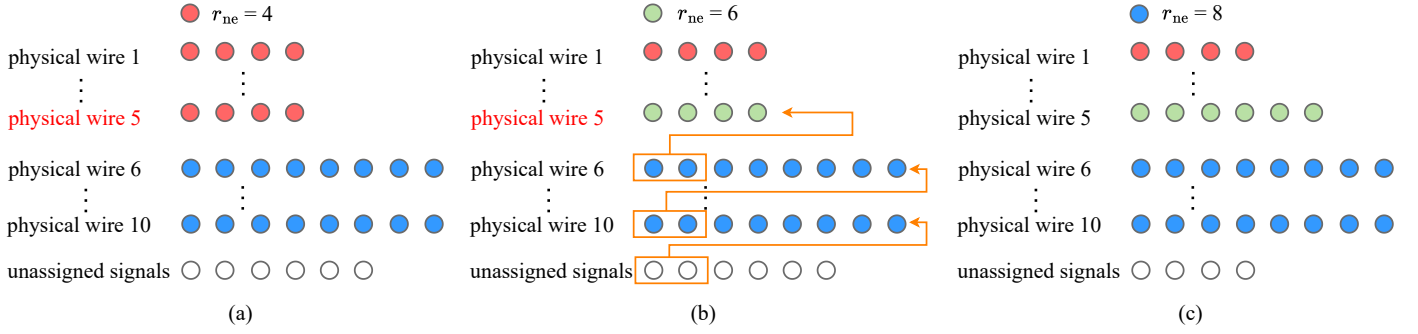


Fig. 3. Illustration of delay-aware TDM wire assignment: (a) Initial assignment with physical wire 5 selected for a TDM ratio increment; (b) Increasing the TDM ratio on wire 5 by Δ_{TDM} (assuming $\Delta_{TDM} = 2$), with subsequent wires sequentially accommodating the next Δ_{TDM} signals; (c) Result after adjustment.

Algorithm 2: Delay-aware TDM Wire Assignment

Input: TDM edge e , TDM ratio r_{ne} .

Output: Assignment of signals to physical wires.

- 1 Compute directional capacity Cap_e^d ;
 - 2 **for** each direction of TDM edge e **do**
 - 3 Sort signals by ascending r_{ne} , breaking ties by descending maximum connection delay;
 - 4 **while** the capacity Cap_e^d is not reached **do**
 - 5 Select signal s with smallest r_{ne} ;
 - 6 Assign s to a new physical wire w ;
 - 7 Assign the next $(r_{ne} - 1)$ signals to w ;
 - 8 Set the TDM ratio of these signals to r_{ne} ;
 - 9 Reorder signals with identical TDM ratios on physical wires;
 - 10 **while** unassigned signals exist **do**
 - 11 Select a physical wire with minimal connection delay upon TDM ratio expansion;
 - 12 Expand the TDM ratio of the selected wire and assign additional signals;
 - 13 Maintain signal order on the physical wire;
-

the total capacity Cap_e of a TDM edge e is divided into two directional capacities Cap_e^d , allocated in proportion to the traffic demand of each direction. The demand is quantified as the sum of the reciprocals of the corresponding TDM ratios. Formally,

$$\text{Cap}_e^d = \left\lceil \text{Cap}_e \cdot \frac{\sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}^d}}{\sum_{n \in \mathbb{N}_e} \frac{1}{r_{ne}}} \right\rceil,$$

where $\lceil \cdot \rceil$ denotes rounding to the nearest integer.

In the TDM wire assignment process, all signals mapped to the same physical wire must share an identical TDM ratio, and the sum of the reciprocals of these ratios must not exceed 1. Equivalently, the assigned TDM ratios collectively determine the maximum number of signals that a physical wire can accommodate. Due to limited capacity, this assignment often necessitates increasing the TDM ratios of specific signals, leaving a margin between demand and available capacity on each TDM edge. However, such adjustments inevitably introduce additional delay for some connections. To efficiently utilize physical wires while minimizing critical connection delay, we introduce a delay-aware TDM wire assignment strategy. The core idea is to balance the feasibility of wire capacity with delay optimization. The TDM ratios are expanded only when necessary, and when capacity becomes insufficient, the algorithm selects the wire whose expansion results in the minimum connection delay.

Algorithm 2 outlines the procedure. For each direction of edge e , signals are first sorted in ascending order of their TDM ratios r_{ne} ; ties are broken by prioritizing signals with larger maximum connection delays (line 3). Unassigned signals are then grouped and allocated to physical wires until the directional capacity Cap_e^d is reached. Specifically, the signal with the smallest r_{ne} is chosen as the baseline, and the next $(r_{ne} - 1)$ signals are grouped onto the same physical wire, with their TDM ratios adjusted to r_{ne} (lines 4–8). This ensures that all signals mapped to the same wire share an identical TDM ratio. Note that aligning larger TDM ratios downward to smaller ones consumes more physical wire capacity. Nevertheless, this approach prioritizes minimizing the maximum connection delay. Smaller TDM ratios typically correspond to signals on high-delay connections; if instead they were aligned upward to larger ratios, more wire capacity would be freed, but at the

After the initial TDM ratio assignment, each TDM ratio is rounded to the nearest integer multiple of the TDM step Δ_{TDM} , without violating the capacity constraint Cap_e . For edges with remaining capacity, a refinement procedure iteratively reduces the critical TDM ratios by Δ_{TDM} following the guidance of a max-heap prioritized by the maximum connection delay. In each iteration, the top element is extracted and decreased if the edge's remaining capacity allows, then reinserted into the heap. Refinement terminates when no further reductions are possible, producing locally delay-optimized TDM ratios.

D. Delay-Aware TDM Wire Assignment

Upon finalizing the TDM ratio assignment under the TDM ratio constraints, the process advances to the TDM wire assignment stage. Each TDM edge consists of Cap_e parallel physical wires, and signals on the edge must be mapped to specific wires while strictly adhering to the TDM wire constraints. Since each physical wire supports only unidirectional transmission,

TABLE I: Benchmark statistics and comparison of critical connection delay with the contest winners and state-of-the-art methods.

Benchmark	Dies	Nets	SLL Wires Cap.	TDM Wires Cap.	1st	2nd	3rd	Adapted [26]	[3]	[8]	Ours
design1	8	5	122,040	200	6.5	6.5	6.5	6.5	6.5	6.5	6.5
design2	8	86	122,040	200	7.5	7.5	7.5	12	7.5	7.5	7.5
design3	8	84	122,040	10	11.5	14.5	11.5	28	11.5	11.5	11.5
design4	8	449	122,040	20	19.5	19.5	18.5	32	18.5	18.5	9.5
design5	12	5,083	183,060	220	135.5	136	132.5	185	131	136	131
design6	12	145,660	183,060	5,100	213	260	287	FAIL ^a	211.5	163	160.5
design7	16	76,258	244,080	4,250	84.5	102.5	76	270	78.5	74	70.5
design8	16	86,139	244,080	3,450	124	145.5	123	505	113.5	107.5	104.5
design9	16	871,588	244,080	71,000	150	196.5	177	FAIL	154.5	122.5	122.5
design10	20	3,324,963	305,100	37,500	4657.5	4745	5700	FAIL	4739.5	4207.5	4200
Normalized					1.213	1.355	1.280	2.639	1.185	1.108	1.000

^a "FAIL" indicates that the routing result is invalid due to violation of the SLL edge constraint.

cost of excessively increasing the maximum connection delay. Therefore, our strategy at this step is to favor preserving delay performance over maximizing wire capacity utilization.

After the initial assignment, signals with identical TDM ratios are sorted in descending order of their maximum connection delay, and the physical wires are adjusted accordingly (line 9). Specifically, if the physical wires w_1, w_2, w_3 are assigned the same TDM ratio, then the maximum connection delays of the signals on these wires satisfy $\text{delay}(w_1) \leq \text{delay}(w_2) \leq \text{delay}(w_3)$. At this stage, both the TDM ratio of each physical wire and the maximum delay among the signals it carries are determined. This maximum delay is then used as the criterion for wire selection in subsequent steps. Remaining unassigned signals are accommodated by incrementally expanding the TDM ratios of selected wires. At each iteration, the wire whose expansion yields the minimal connection delay is selected, and new signals are drawn from the higher-ratio group, prioritizing those with the largest delays (lines 10–13).

Fig. 3 presents an example of delay-aware TDM wire assignment. In the initial configuration (Fig. 3(a)), signals with identical TDM ratios are arranged sequentially. The algorithm selects the physical wire for TDM ratio expansion that incurs the minimum connection delay after expansion. Since signals within each ratio group are pre-sorted by delay, only wires with differing TDM ratios are considered. For example, in Fig. 3(a), wires 5 and 10 are considered for expansion. Suppose that increasing the TDM ratio of wire 5 incurs a smaller connection delay than increasing the TDM ratio of wire 10, then wire 5 is selected. In Fig. 3(b), increasing the TDM ratio of wire 5 by a step size Δ_{TDM} enables it to accommodate Δ_{TDM} additional signals. Then, the first Δ_{TDM} signals on wire 6 (those with the largest delays in the TDM ratio-12 group) are reassigned to wire 5, and the updated assignment is shown in Fig. 3(c).

V. EXPERIMENTS

We implemented our algorithm in C++ and evaluated it on Windows with an AMD Ryzen 5 4600H CPU (3.00 GHz) and 16 GB RAM. Benchmark datasets from the Die-Level Routing Contest 2023 [27], provided by an industrial vendor [28]. To assess the effectiveness of our approach, we compared it against the top-3 winning solutions from the contest as well as recent state-of-the-art methods [3], [8]. Additionally, we refer to [8], which adapts the FPGA-level routing method of [26] to the die-level setting, showing that such simple adaptation is ineffective due to fundamental problem differences.

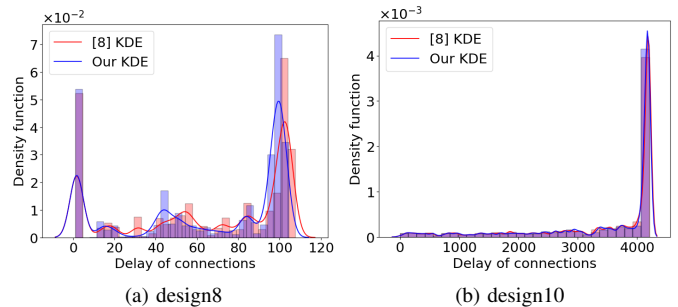


Fig. 4. Delay distribution of connections. KDE: Kernel Density Estimate.

Table I summarizes the benchmark statistics and compares the critical connection delays, with the best results highlighted in bold. Compared with the top-3 contest entries, our approach reduces the critical connection delay by 21.3%, 35.5%, and 28.0%, respectively. Furthermore, relative to the state-of-the-art die-level multi-FPGA routing methods [3], [8], our method achieves reductions of 18.5% and 10.8% in critical connection delay. Notably, since we adopt the same routing strategy as [8], **these improvements directly reflect the effectiveness of our TDM ratio assignment and delay-aware TDM wire assignment strategies, which provide a $(1 + \epsilon)$ -approximation guarantee in the TDM ratio assignment phase and minimize additional delay during wire assignment.**

Fig. 4 presents the kernel density estimates (KDE) of connection delays for design8 and design10. Compared with [8], our approach achieves a substantial reduction in critical connection delay and yields a tighter overall distribution, resulting in more pronounced and concentrated peaks. **This improvement is attributed to our balanced TDM ratio assignment, which enables more efficient utilization of TDM resources across connections.**

VI. CONCLUSION

This paper tackles the die-level routing problem in multi-FPGA systems with a focus on the TDM stage. We formulate the TDM ratio assignment problem as a block-angular convex program and apply Lagrangian decomposition to obtain a $(1 + \epsilon)$ -approximate solution. Furthermore, we introduce a delay-aware TDM wire assignment strategy that maps signals to physical wires while minimizing incremental local delay. Experimental results show that, under the same routing method, our TDM optimization framework outperforms the SOTA approach, reducing critical connection delay by up to 10.8%.

REFERENCES

- [1] C. Constantinescu, "Trends and challenges in vlsi circuit reliability," *IEEE Micro*, vol. 23, no. 4, pp. 14–19, 2003.
- [2] W. N. Hung and R. Sun, "Challenges in large fpga-based logic emulation systems," in *Proceedings of the 2018 International Symposium on Physical Design*, ISPD '18, p. 26–33, 2018.
- [3] C. Huang, P. Chu, S. Bi, R. Sun, and H. You, "System routing and tdm assignment optimization in multi-2.5d fpga-based prototyping systems," in *Proceedings of the 2024 International Symposium of Electronics Design Automation*, ISEDA '24, pp. 324–331, 2024.
- [4] R. Raikar and D. Stroobandt, "Multi-die heterogeneous fpgas: How balanced should netlist partitioning be?," in *Proceedings of the 2023 ACM/IEEE Workshop on System Level Interconnect Pathfinding*, SLIP '22, 2023.
- [5] S. Ravishankar, D. Gaitonde, and T. Bauer, "Placement strategies for 2.5d fpga fabric architectures," in *Proceedings of the 2018 International Conference on Field Programmable Logic and Applications*, FPL '18, pp. 16–164, 2018.
- [6] M. Inagi, Y. Takashima, and Y. Nakamura, "Globally optimal time-multiplexing of inter-FPGA connections for multi-FPGA prototyping systems," *IPSS Transactions on System and LSI Design Methodology*, vol. 3, pp. 81–90, 2010.
- [7] J. Babb, T. Russell, and A. Anant, "Virtual wires: overcoming pin limitations in FPGA-based logic emulators," in *Proceedings of the 1993 IEEE Workshop on FPGAs for Custom Computing Machines*, pp. 142–151, 1993.
- [8] J. Wang, Y. Liu, and Y. Lin, "Synergistic die-level router for multi-fpga system with time-division multiplexing optimization," in *Proceedings of the 2025 ACM/IEEE Design Automation Conference*, DAC '25, 2025.
- [9] C.-W. Pui and E. F. Y. Young, "Lagrangian relaxation-based time-division multiplexing optimization for multi-fpga systems," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 25, Feb. 2020.
- [10] Y. Su, R. Sun, and P.-H. Ho, "2019 cad contest: System-level FPGA routing with timing division multiplexing technique," in *Proceedings of the 2019 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '19, pp. 1–2, 2019.
- [11] T. Lin, W. Tai, Y. Lin, and I. H. Jiang, "Routing topology and time-division multiplexing co-optimization for multi-FPGA systems," in *Proceedings of the 2020 ACM/IEEE Design Automation Conference*, DAC '20, pp. 1–6, 2020.
- [12] S. E. Dreyfus and R. A. Wagner, "The steiner problem in graphs," *Networks*, vol. 1, no. 3, pp. 195–207, 1971.
- [13] P. Zou, Z. Lin, X. Shi, Y. Wu, J. Chen, J. Yu, and Y.-W. Chang, "Time-division multiplexing based system-level FPGA routing for logic verification," in *Proceedings of the 2020 ACM/IEEE Design Automation Conference*, DAC '20, pp. 1–6, 2020.
- [14] D. Zheng, X. Zhang, C.-W. Pui, and E. F. Young, "Multi-fpga co-optimization: Hybrid routing and competitive-based time division multiplexing assignment," in *Proceedings of the 2021 Asia and South Pacific Design Automation Conference*, ASPDAC '21, p. 176–182, 2021.
- [15] W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. Chao, "Nctu-gr 2.0: Multi-threaded collision-aware global routing with bounded-length maze routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 5, pp. 709–722, 2013.
- [16] K. Mehlhorn, "A faster approximation algorithm for the steiner problem in graphs," *Information Processing Letters*, vol. 27, no. 3, pp. 125–128, 1988.
- [17] Z. Zhuang, X. Huang, G. Liu, W. Guo, W. Qian, and W.-H. Liu, "Alifrouter: A practical architecture-level inter-fpga router for logic verification," in *Proceedings of the 2021 Design, Automation and Test in Europe Conference*, DATE '21, pp. 1570–1573, 2021.
- [18] W. Lin, H. Wu, P. Gao, W. Luo, S. Cai, and X. Xiong, "Sequential routing-based time-division multiplexing optimization for multi-FPGA systems," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 28, p. 10, oct 2023.
- [19] C. Lemaréchal, "Lagrangian relaxation," in *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions* (M. Jünger and D. Naddef, eds.), pp. 112–156, Springer, 2001.
- [20] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, pp. 345–345, 1962.
- [21] L. McMurchie and C. Ebeling, "Pathfinder: a negotiation-based performance-driven router for fpgas," in *Proceedings of the 1995 ACM Third International Symposium on Field-Programmable Gate Arrays*, FPGA '95, p. 111–117, 1995.
- [22] Y. Zha and J. Li, "Revisiting pathfinder routing algorithm," in *Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '22, p. 24–34, 2022.
- [23] M. D. Grigoriadis and L. G. Khachiyan, "Fast approximation schemes for convex programs with many blocks and coupling constraints," *SIAM Journal on Optimization*, vol. 4, no. 1, pp. 86–107, 1994.
- [24] M. D. Grigoriadis and L. G. Khachiyan, "Coordination complexity of parallel price-directive decomposition," *Mathematics of Operations Research*, vol. 21, no. 2, pp. 321–340, 1996.
- [25] V. Klee, "Convex analysis (r. tyrrell rockafellar)," *SIAM Review*, vol. 13, no. 2, pp. 233–238, 1971.
- [26] W.-K. Liu, M.-H. Chen, C.-M. Chang, C.-C. Chang, and Y.-W. Chang, "Time-division multiplexing based system-level fpga routing," in *Proceedings of the 2021 IEEE/ACM International Conference On Computer Aided Design*, pp. 1–6, 2021.
- [27] S2C, "Fpga die-level system routing algorithm design," technical report, S2C, 2023.
- [28] S2C Inc., "S2C prototyping: FPGA ASIC SoC IP verification, validation, emulation," 2024. Accessed: 2025-08-21.