Delay Uncertainty and Signal Criticality Driven Routing Channel Optimization for Advanced DRAM Products

Samyoung Bang[§], Kwangsoo Han[†], Andrew B. Kahng[†‡] and Mulong Luo[‡]
CSE[‡] and ECE[†] Departments, UC San Diego, La Jolla, CA, USA
Samsung Electronics Co. Ltd.[§], Hwaseong-si, Gyeonggi-do, Republic of Korea
eva.bang@samsung.com, {kwhan, abk, muluo}@ucsd.edu

Abstract— Signal delay uncertainty induced by crosstalk is a critical challenge to the physical design of long interconnect channels in DRAM products at the 2× and 1× technology nodes. Due to severe cost challenges in a high-volume, commodity market, layout resources including channel width, buffers, and number of metal routing layers are extremely scarce. We describe a new channel optimizer that reduces crosstalk-induced delay uncertainty, weighted by signal criticality and aware of signal activity correlations (e.g., to reduce delay uncertainty by mutual shielding). Instead of the typical signal net permutation strategy, we apply (pessimistic) timing-driven swizzling to minimize the delay uncertainty cost function. Contributions of this work include (1) an accurate and efficient analytical crosstalk delay calculator, (2) scalability up to hundreds of signals and tracks in the routing channel through use of greedy and decomposition strategies as well as a pair-swapping approach, and (3) experimental studies that demonstrate up to 24% reduction of the worst-case criticality-weighted delay uncertainty (or, 34ps of absolute delay uncertainty reduction) compared with the typical signal permutation approach.

I. Introduction

Signal delay uncertainty induced by crosstalk is a critical challenge to the physical design of long interconnect channels in advanced DRAM products. This challenge arises due to resource scarcity: channel width, buffers, and number of metal layers. The problem is especially severe in the interconnect channel which lies in the control block of the DRAM chip, as shown in Figure 1. Crosstalk between signals in the channel can cause signal integrity issues and timing violations, leading to failures in DRAM read or write operations. It is necessary to route signals carefully to minimize the crosstalk effect. Previous works mainly focus on crosstalk analysis [2][3][4][8][9][11][12][14][16] and crosstalk-aware design [17][18][19][20][21][23][24][25] in logic circuits, with little attention to design automation of a resource-constrained, long channel as seen in DRAM [22]. Manual design is still the dominant methodology for this challenging DRAM interconnect; hence, while techniques such as swizzling or twisted structures might be introduced, applications of such levers are based on fixed patterns and may be far from optimal. With these motivations, we develop an automated DRAM channel layout optimizer to minimize the crosstalk effects among different signals, according to given classes of signal criticalities and corresponding layout rules (i.e., buffer size and buffer distance, and non-default wire width and spacing).

Our contributions are summarized as follows.

- We develop a timing-driven DRAM channel optimizer that takes as inputs a set of signals and routing track resources, along with delay and slew constraints and *signal criticality class* information. The optimizer comprehends correlations of signal activities, probabilities of worst-case delay uncertainties, and impacts of multiple criticality classes of signals. It outputs a layout of the DRAM channel that minimizes the maximum criticality-weighted delay uncertainty induced by crosstalk.

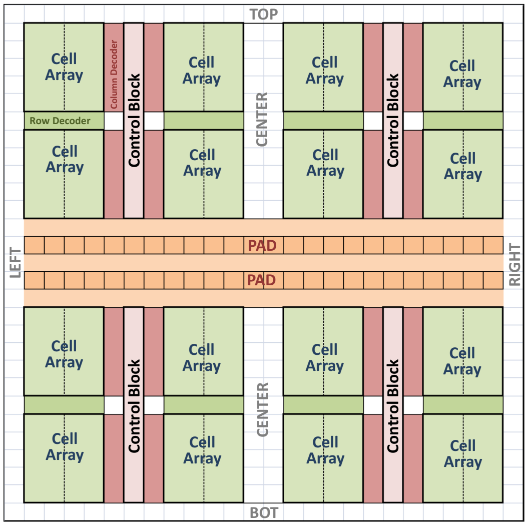

Fig. 1: Schematic of a DRAM die, adapted from [1]. "Control Block" indicates the interconnect channel of interest in our work.

- To guide the channel layout optimization, we develop an accurate, closed-form analytical delay calculator based on the previous works of [9] and [13].
- We go beyond today's typical signal permutation strategies, and integrate swizzling per segment in the channel to exploit signal correlations and achieve reduced worst-case delay uncertainties.
- We explore several approaches to achieve a high-quality, scalable channel layout optimization: (i) an optimal mixed integer-linear program (MILP) formulation for track permutation in a given segment of the channel; (ii) decomposition of the signals into subsets for scalability; and (iii) iterative improvement of the signal permutations within a segment of the channel, by greedy pair-swapping. We provide SPICE simulations and other studies to support our choices of approaches.
- We study testcases with signal criticality class distributions and metal design rules based on inputs from a leading DRAM manufacturer. Our experiments assess the tradeoff of channel area versus uncertainty, as well as the sensitivity of solution quality to decomposition for large testcases.
- We achieve a 24% reduction of maximum weighted delay uncertainty versus the conventional signal permutation solution for a testcase with 200 signals; this corresponds to 34ps absolute delay uncertainty reduction.

In the following, Section II provides a review of related previous works. In Section III, we give a detailed problem formulation, along with our delay calculation and overall optimization approach. In Section IV we describe the testcases that we use to test our layout optimization. In Section V, we describe our design of experiments

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

along with the computational results, and we compare them against the conventional track permutation-based solutions and verify results with SPICE simulations. We conclude and provide ongoing research directions in Section VI.

II. Previous Work

Previous works have addressed various aspects of our problem, including crosstalk-aware routing, accurate crosstalk modeling, aggressor and victim arrival time alignment, and design levers to reduce crosstalk in interconnect design. We briefly review these works as follows.

Crosstalk-aware MILP-based detailed routing. Gao et al. [17] formulate an MILP for detailed channel routing using track permutation to minimize the sum of the slacks of all signals, given the required arrival time of each signal and the constraint that no signal has negative slack. While the MILP enables an optimal solution, coupling length is the only factor that affects timing, i.e., the formulation does not take into account the positions of aggressors, the driver strengths, and the load capacitances of aggressors or victims, or the switching activity correlations between different signals. In advanced nodes, it is necessary to consider these factors for crosstalk delay estimation and the overall channel optimization.

Analytical modeling of crosstalk-induced delay and noise. Xiao et al. [8] and Dartu et al. [10] use analytical (two-pole) models for crosstalk waveforms. While accurate, the computational complexities are high due to Newton-Raphson iteration. Other works use a one-pole model to reduce the complexity. For example, Cong et al. [13] provide a closed-form expression for the crosstalk noise waveform between two parallel signal routes using a 2-π equivalent circuit by only considering the dominant pole. The method can handle multiple aggressors.

Arrival time alignment of aggressor and victim for worst-case victim delay and noise. Arrival time alignments of aggressor and victim signals strongly affect the crosstalk-induced delay variation. When the arrival times of two signals are close to each other, the delay effect is larger than when the arrival times are widely separated. Gross et al. [6] and Sirichotiyakul et al. [12] present methods to align aggressors for worst-case crosstalk-induced delay or noise experienced by the victim net. However, these methods typically require iterations to converge as well as expensive SPICE-based calibration. Sato et al. [9] derive the delay change of the victim as a function of the arrival time difference with respect to the aggressor(s), based on the crosstalk-induced waveforms. The worst-case delay change is derived analytically based on this function.

Swizzling-based interconnect design to reduce crosstalk-induced delay and noise. Several previous works study *swizzling*, i.e., patterns of track permutations to mitigate crosstalk effects on given victim nets. Zhong et al. [23] propose a twisted-bundle layout structure to reduce coupling. The work mainly focuses on inductive noise reduction and does not address capacitive noise in detail. Gupta et al. [7] present a swizzling pattern to reduce capacitive crosstalk effect based on the Elmore delay model. Yu et al. [25] present a twisted pattern to reduce both capacitive and inductive delay impacts, and verify their results on silicon. Although the authors show that the twisted pattern can reduce delay and improve signal integrity, their fixed pattern does not consider either signal activity correlation or signal criticalities, as our work does.

III. Crosstalk-Aware Layout Optimization

We formulate the crosstalk-aware layout optimization problem as follows.

- A set of *signals* to be routed is given in a long and narrow rectangular *channel* that is represented by a set of *tracks*.
- A channel is divided into a set of *segments*, where each segment has the same length. Crosstalk delay uncertainty in a given signal is from coupling with the signals on its left and right neighboring tracks.[1]
- Signals are assigned to different *criticality classes* (e.g., CLK has the highest criticality, CMD has the second highest criticality). Signals with a given criticality belong to the same class. For each distinct signal criticality, the timing constraint is different, e.g., less critical signals generally have later required arrival times than higher critical signals.
- We apply different layout rules (e.g., metal pitch) to each signal criticality classes. Accordingly, the inter-buffer lengths (distances between two consecutive buffers of the same signal) differ between criticality classes.

Under the given constraints, we seek to assign signals to tracks in each channel segment to minimize crosstalk delay uncertainty.[2] Our objective is to route all signals while minimizing the maximum weighted delay uncertainty over all signals.

A. Notations and Problem Statement

We denote the set of signals s_i by S, the set of tracks t_k by T, and the set of segments g_j by G. We also denote the set of classes of signals γ_l by Γ, and the most critical class of signals by γ_0.

Figure 2 shows an example with signals $s_0, s_1 \in \gamma_0$ and signals $s_2, s_3 \in \gamma_1$. The inter-buffer length of γ_0 (resp. γ_1) is two (resp. four) units of segments.

We use binary variable $q_{i,k}^j$ to indicate whether the signal s_i is on track t_k at the segment g_j of the channel ($q_{i,k}^j = 1$) or not ($q_{i,k}^j = 0$). Given $q_{i,k}^j$ for all $s_i \in S$, $t_k \in T$, $g_j \in G$, we can determine the layout of the channel. We then calculate the delay uncertainty d_i^j of signal s_i at the endpoint of g_j in the channel. We define the maximum delay uncertainty of the signals in class γ_l as $D_l^{max} = \max_{s_i \in \gamma_l}(d_i^{|G|})$. We define the weighting factor for class γ_l as λ_l. Our crosstalk-aware layout optimization problem is formally stated as follows.

Objective: $\max_{\gamma_l \in \Gamma}(\lambda_l \cdot D_l^{max})$
Input: set of tracks T; set of signals S; set of classes Γ; design rules for each class; inter-buffer lengths for each class; and sizes of buffers.
Output: track assignments for each signal $s_i \in S$.

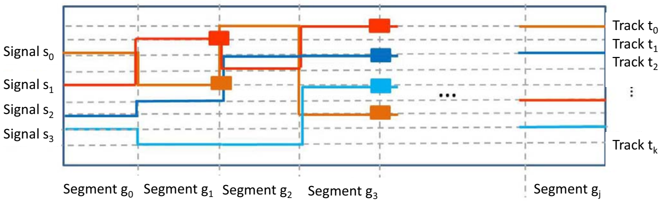

Fig. 2: A channel with four signals that are assigned to different tracks at different segments. The square blocks indicate buffers.

[1] Based on feedback from our our industry collaborators, the length of the channel is around $8000\mu m$, while the width of the channel is around $100\mu m$ to $200\mu m$. Due to the aspect ratio (length/width) of the channel being very high, the coupling effect from vertical jogs (segments that are parallel to the width direction of the channel) is much smaller than that on segments that are parallel to the length direction. Thus, we ignore the coupling effect from vertical jogs.

[2] We consider the delay uncertainty when the aggressors switch in the opposite direction from the victim. Thus, in our work the delay uncertainty of a victim is computed as the maximum delay with aggressors minus delay without aggressors.

B. Our Approach

To address the crosstalk-aware layout optimization problem described above, we optimize the entire channel segment by segment. We study two approaches: (i) an MILP-based approach, and (ii) a pair-swapping approach. Figure 3 shows our entire optimization flow. We validate our flow on artificial testcases which comprise the specifications of each given channel, including length, width, number of signals, number of tracks, etc. To optimize each segment, we first pre-calculate the delay uncertainty of all pairs of signals; then, we apply either the pair-swapping approach or the MILP-based approach to assign signals to tracks in the current segment. We iteratively apply this optimization method until the end of the channel is reached.

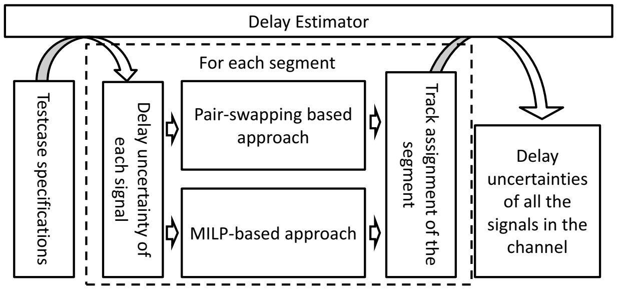
Delay Estimator
For each segment
Testcase specifications
Delay uncertainty of each signal
Pair-swapping based approach
MILP-based approach
Track assignment of the segment
Delay uncertainties of all the signals in the channel

Fig. 3: Our optimization flow.

Segment-by-segment optimization. Given a channel with a set of signals S, a set of tracks T and a set of segments G, the number of binary variables $q^j_{i,k}$ is $|T| \cdot |S| \cdot |G|$, and the solution space has size $P(|T|, |S|)^{|G|}$, where $P(|T|, |S|) = \frac{|T|!}{(|T|-|S|)!}$. Because it is not feasible to enumerate all solutions, we propose a greedy method that optimizes the layout segment by segment from the leftmost segment g_1 to the rightmost segment $g_{|G|}$. Thus, we only need to enumerate at most $P(|T|, |S|)$ solutions per segment, and for a channel with $|G|$ segments, at most $P(|T|, |S|) \cdot |G|$ solutions need to be enumerated. Note that when we optimize segment g_j, the routing solutions for segments $g_{j'}$, where $j' > j$, are not yet determined. Therefore, the delay uncertainties of the signals at segment $g_{j'}$ cannot be accurately calculated. To resolve this issue, we make the pessimistic assumption that there are no coupling capacitances between any two signals s_i and $s_{i'}$ at any segment $g_{j'}$, where $j' > j$. This leads to the worst-case (maximum) delay uncertainty d^j_i at the segment g_j.[3]

A small example in Figure 4 illustrates how our method can achieve relatively good solution quality compared to the optimal solution. We consider four signals and four tracks, with the entire channel divided into four segments and each segment's length being $100\mu m$. The coupling capacitance between neighboring tracks in each segment is 2.7fF, the ground capacitance of each track segment is 8fF, and the resistance per track segment is 10.3Ω. We assume load capacitance to be 20fF and driver on-resistance to be 300Ω, with the starting slew (at the output pin of the driving buffer) of all signals to be 100ps. Figure 4 compares the optimal solution, our greedy (segment-by-segment permutation) solution and the typical (whole-track permutation) solution for comparison. The maximum delay uncertainties for optimal, our, and typical solutions are 7.16ps, 7.19ps, and 9.57ps, respectively. We use the delay uncertainty calculator described in Section III-C. The suboptimality of our solution is $\frac{7.19}{7.16} - 1 = 0.4\%$.

Although our segment-by-segment optimization reduces the solution space, it is still infeasible to find the best solution from $P(|T|, |S|)$ permutations for one segment when $|T|, |S|$ are large. As noted above, to optimize the track assignments for each segment, we study two approaches: (i) an MILP-based approach and (ii) a pair-swapping approach.

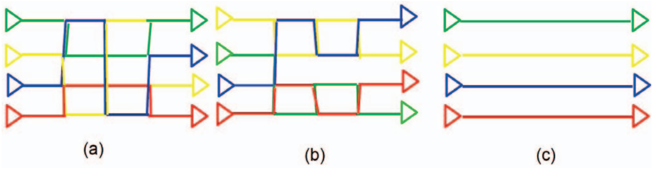

Fig. 4: Schematic layouts of (a) optimal solution, (b) our solution, and (c) the typical solution.

MILP-based approach. Delay uncertainty of a signal is mainly affected by the coupling capacitances with its nearest neighbors. For an accurate delay uncertainty calculation, it is essential to consider both the left and the right adjacent signals at the same time. However, this would induce $O(|S|^3)$ binary variables in a straightforward MILP formulation.[4] Thus, we propose an MILP that considers the left and right signals separately, and superposes the induced delay uncertainties.[5] Once we obtain the solution for the current segment, we update the delay uncertainties of all signals by considering their neighbors on both sides at the same time, and then move on to optimize the next segment. The details of our MILP formulation are as follows.

Input: $d^{j-1}_i \quad \forall s_i \in S, g_{j-1} \in G$

$\Delta d^j_{i,i'} \quad \forall s_i, s_{i'} \in S, g_j \in G$

$\lambda_l \quad \forall \gamma_l \in \Gamma$

Output: $q^j_{i,k} \quad \forall s_i \in S, g_j \in G, t_k \in T$

Minimize: $\max\limits_{\gamma_l \in \Gamma} (\lambda_l \cdot \hat{D}^j_l)$

Subject to:

$$\hat{D}^j_l = \max_{s_i \in \gamma_l} \hat{d}^j_i \quad \forall \gamma_l \in \Gamma \tag{1}$$

$$\hat{d}^j_i = d^{j-1}_i + \sum_{s_{i'} \in S} \Delta d^j_{i,i'} \cdot p^j_{i,i'} \tag{2}$$

$$\sum_{s_i \in S} q^j_{i,k} \leq 1 \quad \forall t_k \in T \tag{3}$$

$$\sum_{t_k \in T} q^j_{i,k} = 1 \quad \forall s_i \in S \tag{4}$$

$$q^j_{i,k} + q^j_{i,k-1} + q^j_{i',k} + q^j_{i',k-1} \leq 1 + p^j_{i,i'} \quad \forall s_i, s_{i'} \in S, t_k \in T \tag{5}$$

$$\sum p^j_{i,i'} \leq |S| - 1 \quad \forall s_i, s_{i'} \in S,\ i < i' \tag{6}$$

[3] Intuitively, larger ground-cap-to-coupling-cap ratio C_g/C_c leads to reduction of crosstalk effects. Here, C_c is the total coupling capacitance with the active aggressors. When we consider the crosstalk coupling at the current segment, the coupling capacitance $C_{c,other}$ at other segments is regarded as inactive and is added to C_g. Thus, if we do not consider possible coupling capacitance $C_{c,other}$ in segment $g_{j'}$, $j' > j$, we actually underestimate C_g/C_c and in consequence overestimate the crosstalk effect.

[4] For each signal considering its left and right neighbor combinations, there will be $C(|S|, 3) = \frac{|S|!}{3! \cdot (|S|-3)!}$ combinations, and $O(|S|^3)$ binary variables to indicate all the combinations, which does not scale for large $|S|$. There are $C(|S|, 2) = \frac{|S|!}{2! \cdot (|S|-2)!}$ combinations and $O(|S|^2)$ binary variables if we only consider the neighbor on one side.

[5] Because the track assignment of current segment is not decided, if we consider the aggressor on the left (resp. right) neighboring track of the current segment, we assume that there is no aggressor on the right (resp. left) neighboring track of the current segment.

where $\hat{D_l^j}$ is an estimate of the maximum delay uncertainty for signals in class γ_l at the endpoint of segment g_j, and $\hat{d_i^j}$ estimates delay uncertainty for signal s_i at the endpoint of segment g_j. We assume there is no coupling capacitance on the other side of the victim of the segment when we calculate $\hat{D_l^j}$ and $\hat{d_i^j}$. Note that $\hat{D_l^j}$ and $\hat{d_i^j}$ are different from D_l^j and d_i^j which are accurate delay uncertainties updated after solving the current MILP instance for segment g_j. For the calculation of D_l^j and d_i^j, we consider both the left and the right coupling capacitances of segment g_j.

Our objective is to minimize the maximum weighted delay uncertainty over all signals. λ_l is the weighting factor for signal delay uncertainty of class γ_l. In Equation (1), we give an estimate of the maximum delay uncertainty for each class. Equation (2) estimates the delay uncertainty of each signal considering its neighboring signals. We use the binary variable $p_{i,i'}^j$ to indicate whether signal s_i and signal $s_{i'}$ are adjacent in segment g_j. The number of such p variables is $|S| \cdot (|S| - 1)$ for each segment. $\Delta d_{i,i'}^j$ is the delay uncertainty of signal s_i induced by signal $s_{i'}$ in segment g_j, and is given as input to the optimization. Equation (3) ensures that no more than one signal can be assigned to the same track. Equation (4) ensures that each signal must be assigned to exactly one track. Equation (5) enforces $p_{i,i'}^j = 1$ if signals s_i and $s_{i'}$ are neighbors. Equation (6) bounds the number of neighboring signal-pairs to be $|S| - 1$.

For each segment, the number of $p_{i,i'}^j$ variables is $|S| \cdot (|S| - 1)$ and the number of $q_{i,k}^j$ variables is $|S| \cdot |T|$. There are $|T|$ Equations (3), $|S|$ Equations (4), $|S| \cdot (|S| - 1) \cdot (|T| - 1)/2$ Equations (5), and $|S| \cdot (|S| - 1)/2$ Equations (6). Assuming $|S| \approx |T|$, the number of binary variables is $O(|S|^2)$, while the number of constraints is $O(|S|^3)$.

Based on this MILP formulation, we can solve the MILP instance for 20 signals and 20 tracks within five minutes using 30 threads on a 2.5GHz Intel Xeon server. For large testcases, we decompose the channel into small subsets of tracks and solve the MILP instances corresponding to these subsets in parallel. Our solver is *CPLEX* v12.5 [26].

Algorithm 1 Decomposition method to solve large testcase

```
1: for all $g_j \in G$ do
2:    for subset $v_r$ do
3:       $v_r \leftarrow \emptyset$;
4:       if $j$ is even then
5:          $L_r \leftarrow max\{r \cdot V, 0\}$;
6:          $U_r \leftarrow min\{(r+1) \cdot V, |T|\}$;
7:       else
8:          $L_r \leftarrow max\{\lfloor (r - \frac{1}{2}) \cdot V \rfloor, 0\}$;
9:          $U_r \leftarrow min\{\lfloor (r + \frac{1}{2}) \cdot V \rfloor, |T|\}$;
10:      end if
11:      for all $s_i \in S$ do
12:         if $j = 0$ then
13:            $r \leftarrow i\%(\lceil |T|/V \rceil)$; $v_r \leftarrow v_r \cup \{s_i\}$
14:         else
15:            if $q_{i,k}^{j-1} = 1$, $L_r \leq k < U_r$ then
16:               $v_r \leftarrow v_r \cup \{s_i\}$;
17:            end if
18:         end if
19:      end for
20:      $MILP(v_r, g_j)$;
21:      Update track assignments based on the MILP solution;
22:   end for
23: end for
```

Algorithm 1 shows the details of our decomposition method. We use v_r to denote the r^{th} subset of tracks. V is a prescribed maximum number of tracks in each subset. $MILP(v_r, g_j)$ is the MILP instance comprising the signals in v_r and segment g_j. For each segment, we initialize all the subsets to be empty (Line 3). In Lines 4-10, we determine L_r and U_r, which are respectively the minimum and maximum indices of tracks in subset v_r. When j is even, we set the L_r to be $r \cdot V$ or 0 if it is the first subset, and we set U_r to be $(r+1) \cdot V$ or $|T|$ for the last subset. When j is odd, we offset the L_r and U_r by $\frac{V}{2}$ so that signals are grouped differently in segments g_{j-1} and g_j. Figure 5 subsets of tracks in even- and odd-indexed channel segments.

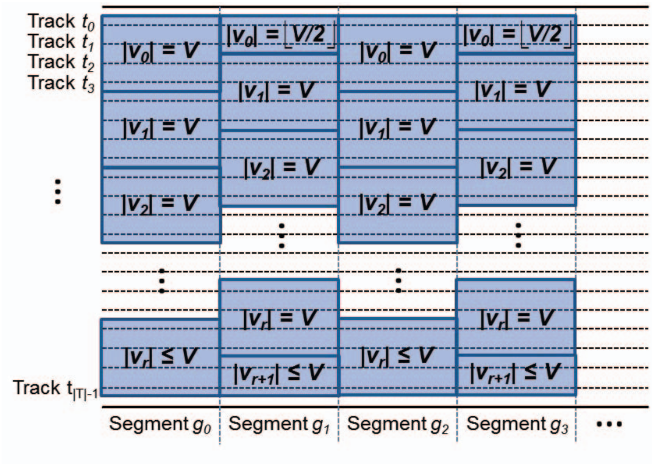

Fig. 5: Subsets of tracks on even- and odd-indexed channel segments. Note that the odd-indexed segments can have one more subset than the even-indexed segments, as illustrated.

We then decompose all the signals in v_r (Lines 11-19) based on the track assignment. For the first segment ($j = 0$), we assign signals based on their indices (Line 13). For the segment g_j ($j > 0$), we assign them based on track assignment on segment g_{j-1} (Lines 15-17). After we assign all signals into subsets, we solve each MILP instance in Line 20. In Line 21, we update the track assignments based on the solution.

The numbers of tracks and signals in the subsets will determine the tradeoff between solution quality and runtime. We study this tradeoff in our experiments in Section V.

Pair-swapping approach. Although the MILP-based approach can handle the large testcase by decomposing it, the decomposition can highly degrade the solution quality. To address this, we propose the pair-swapping approach which shows better solution quality and scalability.

We first select the signal with the maximum delay uncertainty. We attempt to swap it with the remaining signals in ascending order of weighted delay uncertainty. If one swap does not reduce the delay uncertainty, we revert the swap. Otherwise, we keep the swap and update the delay uncertainty of each signal. We keep swapping the signal with the maximum delay uncertainty with other signals until there is no delay uncertainty reduction for any swap.

Algorithm 2 shows the details of our pair-swapping approach. In Line 1, we use the solution of a previous segment g_{j-1} as our initial track assignment for the current segment g_j.[6] We then update the delay uncertainties over all signals (Line 2) and sort the signals in ascending order of weighted delay uncertainty (Line 3). We denote by $d_{pre\_max}^j$ the weighted maximum delay uncertainty in the previous iteration, and by $d_{cur\_max}^j$ the maximum weighted delay uncertainty in the current iteration. We use s_{max} to represent the signal with maximum weighted delay uncertainty

[6] For $j = 0$, i.e., the first segment, we assign all the signals to each subset in a round-robin fashion.

Algorithm 2 Pair-swapping approach for track assignment at g_j $(j > 0)$

```
1: Copy the solution of $g_{j-1}$ for the initial track assignments;
2: Update $d_i^j$, $\forall s_i \in S$;
3: Sort $S$ in ascending order of weighted delay uncertainty;
4: $d^j_{pre\_max} \leftarrow \infty$; $d^j_{cur\_max} \leftarrow max_{s_i \in S} \lambda_l d_i^j$;
5: $s_{max} \leftarrow$ the signal with the maximum weighted delay uncertainty $d^j_{cur\_max}$;
6: while $d^j_{pre\_max} > d^j_{cur\_max}$ do
7:     $d^j_{pre\_max} \leftarrow d^j_{cur\_max}$;
8:     for all $s_i \in S$ do
9:         Swap the signals $s_{max}$ and $s_i$;
10:        for all $s_{i'} \in S$ do
11:            Update $d^j_{i'}$
12:        end for
13:        $d^j_{cur\_max} \leftarrow max_{s_i \in S} \lambda_l d_i^j$;
14:        if $d^j_{pre\_max} > d^j_{cur\_max}$ then
15:            Break;
16:        else
17:            Swap the signals $s_{max}$ and $s_i$;
18:        end if
19:    end for
20:    Sort the signals $s_i \in S$ in ascending order of weighted delay uncertainty;
21:    $d^j_{cur\_max} \leftarrow max_{s_i \in S} \lambda_l d_i^j$;
22: end while
```

by the current solution. In Lines 4-5, we initialize the values of $d^j_{pre\_max}$, $d^j_{cur\_max}$ and s_{max}. We then iteratively run Lines 6-22 until there is no further reduction of maximum delay uncertainty. To be specific, we first save the current max delay uncertainty in $d^j_{pre\_max}$ (Line 7). Then, in Lines 8-19, we swap the signals s_{max} and s_i (Line 9) to check whether the current maximum weighted delay uncertainty can be reduced (Lines 14-18). Note that in the first iteration, we pick the signal that has the minimum delay uncertainty to swap. If there is no max delay uncertainty reduction in the first iteration, we pick the signal that has the second minimum delay uncertainty to swap, we iterate such operation until there is no max delay uncertainty reduction. We break out of the **for** loop (Lines 8-19) and update $d^j_{cur\_max}$ if there is max delay uncertainty reduction, and terminate the optimization (Lines 6-22) if there is no max delay uncertainty reduction.[7]

Note that this pair-swapping approach requires that the number of signals $|S|$ be equal to the number of tracks $|T|$. Thus, in the case that there are empty tracks (i.e., $|S| < |T|$), we add pseudo signals that have no coupling capacitances with other neighbor signals, and ignore these signals' delay uncertainties in our optimization. In other words, these pseudo signals do not have any electrical effects.

C. *Worst-Case Delay Uncertainty Calculation*

Although SPICE simulations can accurately assess the delay uncertainty, this is impractical for our optimization due to the long runtime. We now present an accurate, closed-form delay uncertainty model which is much faster than SPICE simulation. Recall from Section II that Cong et al. [13] propose a closed-form expression for crosstalk noise on a 2-π circuit. Also, recall that Sato et al. [9] propose a closed-form expression to transform the noise waveform into a *Delay Change Curve* (DCC). Our delay uncertainty calculator essentially combines these works according to the flow shown in Figure 6. To estimate the delay uncertainty, we first convert a given aggressor-victim pair in the routing channel into a 2-π circuit. We then convert the noise waveform from the 2-π circuit into an equivalent DCC; we calculate the maximum delay uncertainty using DCC.[8] For multiple aggressors, we calculate the total delay uncertainty by summing the delay uncertainties induced by each of the individual aggressors.

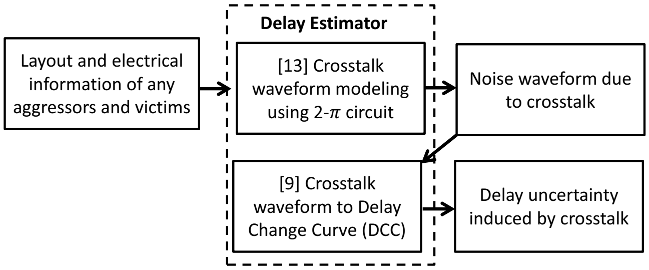
Delay Estimator
Layout and electrical information of any aggressors and victims
[13] Crosstalk waveform modeling using 2-π circuit
Noise waveform due to crosstalk
[9] Crosstalk waveform to Delay Change Curve (DCC)
Delay uncertainty induced by crosstalk

Fig. 6: Delay change calculation flow.

To demonstrate the accuracy of our delay uncertainty estimation, we compare it with SPICE simulations on a testcase with five signals and five tracks, and channel length of $8000\mu m$. We randomly generate 300 swizzling patterns and calculate the delay uncertainty using both our model and SPICE simulations. We also compare SPICE simulation results with results based on Elmore delay and on switching factors based on the model in [7].

Figure 7 shows rank correlations of the delay uncertainties for the five signals over all 300 swizzling patterns (i.e., 1500 data points). In terms of the deviation of ranks, our model has maximum deviation of 9.87% (i.e., maximum rank difference of 148, out of the 1500 instances) versus the uncertainty computed by SPICE, while the model used in [7] has a maximum deviation of 32.5% (maximum rank difference of 487).

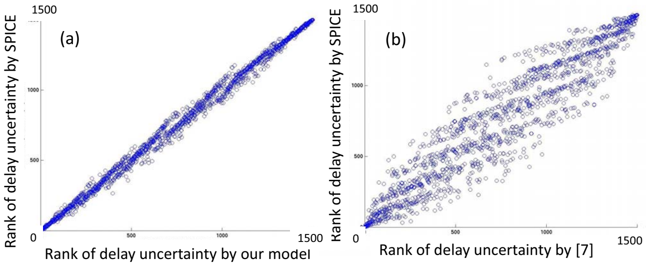

Fig. 7: Rank correlations (a) between our delay uncertainty model and SPICE simulation, and (b) between the model in [7] and SPICE simulation.

IV. Artificial Testcases

To our knowledge, there is no public benchmark available for the purpose of DRAM channel routing optimization. Furthermore, obtaining detailed layout problem instances for leading-edge products is difficult. We therefore develop artificial testcases based on information provided by collaborators at a leading DRAM manufacturer.

We classify the parameters of our developed artificial testcases into three categories: (i) general parameters that are independent of (criticality) classes or signals, (ii) class-specific parameters, and (iii) signal-specific parameters. The details of each category are shown in Table I.

[7] Note that iteratively applying our pair-swapping approach for the same segment does not change the solution. In our pair-swapping method, the j^{th} segment's initial solution is determined by the delay uncertainties of signals at the end of the $(j-1)^{st}$ segment. If we were to apply our pair-swapping method to the same j^{th} segment, the solution is the same since the delay uncertainties at the end of the $(j-1)^{st}$ segment are the same.

[8] If the timing window of each signal is given, we can accurately calculate the maximum delay uncertainty according to the timing windows. However, in practice designers cannot obtain the exact timing windows of each signal in the DRAM channel. Thus, for the purpose of our optimization, we use the peak point of the DCC as the maximum delay uncertainty.

Note that signals in different classes have different design rules (i.e., width and space). Also, signals in different classes have different required distances between buffers. In Table I we show the width and space values, and the buffer distances, for different signal criticality classes. We further assume the thickness of all wires as $0.5\mu m$. Based on the geometry information, we perform simulation using RAPHAEL [27] to derive the class-specific parameters (e.g., $C_{g,l}$, R_i, $C_{c,l,l'}$). We extract parasitic information of buffers (i.e., $C_{buf,l,q}$, $R_{buf,l,q}$) from SPICE models. For signal-specific parameters, we use $R_i = 500\Omega$, τ_i = 130ps, $C_{L,i}$ = 4fF.

TABLE I: Parameters of Our Testcases.

Info.	Symbol	Definition
General param.	$\lvert S\rvert$	number of signals in the channel
	$\lvert G\rvert$	number of segments of the channel
	$\lvert T\rvert$	number of tracks in the channel
	V_{DD}	supply voltage of the system
	clk	clock period of the system
	$\chi_{l,l'}$	probability that a signal in γ_l is correlated with a signal in $\gamma_{l'}$
Class-specific param.	$\lvert\gamma_l\rvert$	number of signals in class γ_l
	$C_{g,l}$	ground capacitance per unit wire length of signals in γ_l
	R_l	resistance per unit length of signals in γ_l
	$C_{c,l,l'}$	coupling capacitance per unit wire length between signals in γ_l and signals in $\gamma_{l'}$
	$L_{buf,l,q}$	distance of the q^{th} buffer from the input of the channel of a signal in γ_l
	$C_{buf,l,q}$	input capacitance of the q^{th} buffer of a signal in γ_l
	$R_{buf,l,q}$	output on-resistance of the q^{th} buffer of a signal in γ_l
Signal-specific param.	$C_{L,i}$	load capacitance of signal s_i
	τ_i	input slew of signal s_i
	R_i	driver resistance of signal s_i
	$\eta_{i,i'}$	activity correlation between signals s_i and $s_{i'}$

We also set the correlations for all pairs of signals.[9] If two signals s_i and $s_{i'}$ are correlated ($\eta_{i,i'} = 1$), then we assume that they are active (i.e., can switch) at the same time. Otherwise, two uncorrelated signals s_i and $s_{i'}$ are assumed not to switch at the same time, and hence can mutually shield each other. Table II shows the probability of the correlation for a given signal in class γ_l to the other signals in class $\gamma_{l'}$.[10]

TABLE II: Probabilities of activity correlation among classes.

$\chi_{l,l'}$	$l=0$	$l=1$	$l=2$	$l=3$	$l=4$
$l'=0$	1	1	1	0.8	0.5
$l'=1$	1	1	0.8	0.5	0.5
$l'=2$	1	0.8	0.5	0.5	0.5
$l'=3$	0.8	0.5	0.5	0.5	0.5
$l'=4$	0.5	0.5	0.5	0.5	0.5

V. Experimental Setup and Results

In this section, we first show several confirmations of the reasonable behaviors of our optimization method, i.e., reasonable sensitivities of the maximum weighted delay uncertainty objective to number of signals, number of tracks, and correlations between signals. We then compare the results from our MILP-based approach (with and without decomposition), our pair-swapping approach, and the typical signal permutation approach. We also show the results of a small testcase using the method of [17] for comparison.

A. Confirmation of Our MILP-based Approach

We have executed three experiments to confirm the reasonable behavior of our MILP-based approach. These respectively study:

- Experiment 1: the impact of the numbers of signals and tracks (where increased number of tracks or reduced number of signals is expected to reduce the objective function value);
- Experiment 2: the impact of the percentage of the signals of each class (where increasing the percentage of higher-criticality signals is expected to increase the objective function value); and
- Experiment 3: the impact of the activity correlation between signals (where increasing the activity correlation is expected to increase the objective function value).

1) Impact of Numbers of Signals and Tracks: To study the impact of the numbers of signals and tracks, Experiment 1 varies $|T|$ and $|S|$, and evaluates the induced changes of objective function values. Table III shows testcases E1T1-E1T6 with respective values of $|T|$, $|S|$ and the number of signals in each criticality class. For the weighting factor, we set $\lambda_0 = 10$, $\lambda_1 = 6.7$, $\lambda_2 = 4$, $\lambda_3 = 2$ and $\lambda_4 = 1$ based on the guidance received from collaborators at a DRAM product company. The channel length is $8000\mu m$, the number of segments is 16, and all signals are fully correlated.

TABLE III: Results for different numbers of signals and tracks.

Testcase	$\lvert S\rvert$	$\lvert T\rvert$	$\lvert\gamma_0\rvert$	$\lvert\gamma_1\rvert$	$\lvert\gamma_2\rvert$	$\lvert\gamma_3\rvert$	$\lvert\gamma_4\rvert$	Objective	Runtime (s)
E1T1	10	10	2	2	2	2	2	544	16
E1T2	10	11	2	2	2	2	2	433	62
E1T3	10	12	2	2	2	2	2	347	80
E1T4	20	20	4	4	4	4	4	796	1156
E1T5	20	21	4	4	4	4	4	633	1274
E1T6	20	22	4	4	4	4	4	545	1408

From Table III, we observe that the objective function value (given in units of picoseconds) increases as the number of signals increases and the number of empty tracks decreases, which follows our expectations. Figure 8(a) shows the first eight segments of the result layout for testcase E1T1 on the first row in Table III. We represent each signal by a different color, and the delay uncertainty by the width of the line (i.e., wider line indicates larger delay uncertainty). In the figure, we observe that the higher-criticality signals are mostly on the boundary of the channel and interleave with the lower-criticality signals, whereby their delay uncertainties can be made smaller than those of the lower-criticality signals.

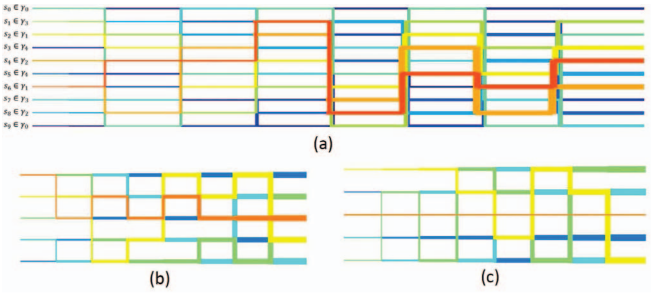

Fig. 8: Part (a): Layout of the optimized channel for testcase E1T1 of Experiment 1. Parts (b) and (c): Layouts of channels for testcases E3T1 and E3T2, respectively, in Experiment 3.

Figure 9(a) shows the changes of weighted delay uncertainties of the 10 signals over the channel. We see that the lower-criticality signals have larger delay uncertainty than higher-criticality signals. Figure 9(b) shows the change of objective values over the channel.

[9] Without having simulation data, correlations of signals can only be related to their timing windows, i.e., two correlated signals tend to have overlapped timing windows, while uncorrelated signals do not have overlapped timing windows. Because it is cumbersome to export timing window information in our flow, we use given correlation information in our optimization. However, if timing windows are given, our method is capable of handling the timing windows.

[10] More precisely, if there are j signals in class γ_l and k signals in class $\gamma_{l'}$, then there are $j \cdot k$ pairs of signals in these two classes. Table II gives a probability of correlation $\chi_{l,l'}$ for these two classes. We choose $\lceil j \cdot k \cdot \chi_{l,l'} \rceil$ out of the $j \cdot k$ signal pairs at random and set their correlations to $\eta = 1$.

Each line indicates the result for each testcase. As the number of empty tracks increases, we see the expected reduction of the maximum weighted delay uncertainty.

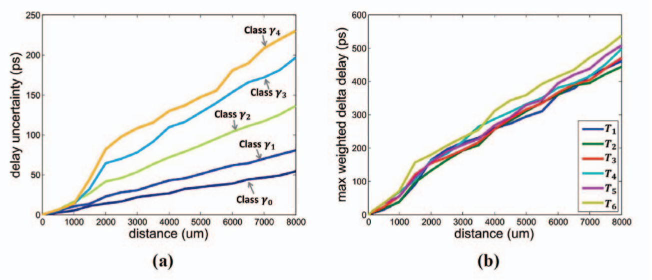

Fig. 9: (a) Delay uncertainties of five signal criticality classes for testcase E1T1 in Experiment 1. (b) Maximum weighted delay uncertainty for each of testcases E1T1-E1T6 in Experiment 1.

2) Impact of Percentage of Signals in Each Class: In Experiment 2, we fix the numbers of tracks and signals, but change the percentage of signals in each criticality class. Table IV shows our experimental setup and results. All the testcases use 20 signals and 20 tracks and the channel length is $8000\mu m$. We assume all signals are completely correlated in this experiment. We set $\lambda_0 = 10$, $\lambda_1 = 6.7$, $\lambda_2 = 4$, $\lambda_3 = 2$ and $\lambda_4 = 1$. For testcases E2T1-E2T4, we use only class γ_0 and class γ_1 signals, and change the ratio between the number of signals in γ_0 and the number of signals in γ_1. We observe that increasing the proportion of the lower-criticality signals reduces the objective function value and the maximum delay uncertainty of each class. Similarly, from testcases E2T3 and E2T5-E2T7, we see that replacing the higher-criticality signals with lower-criticality signals reduces the maximum weighted delay uncertainty. (The lower-criticality signals have less impact on the maximum weighted delay uncertainty than higher-criticality signals.)

TABLE IV: Results for different percentages of signals in each class.

Testcase	l	l'	$\|\gamma_l\|$	$\|\gamma_{l'}\|$	λ_l	$\lambda_{l'}$	D_l^{max}	$D_{l'}^{max}$	Objective	Runtime (s)
E2T1	0	1	20	0	10	6.7	99.0	0	544	16
E2T2	0	1	15	5	10	6.7	96.1	109.2	961	1000
E2T3	0	1	10	10	10	6.7	90.5	108.5	905	1036
E2T4	0	1	5	15	10	6.7	82.8	107.5	828	938
E2T5	0	2	10	10	10	4	88.2	141.1	882	1014
E2T6	0	3	10	10	10	2	85.0	186.4	850	1106
E2T7	0	4	10	10	10	1	84.0	215.5	840	880

3) Impact of Correlation of Signals: In Experiment 3, we study two testcases E3T1 and E3T2. Table V shows our experimental setup and results. For testcase E3T1, we assume that all the signals are correlated with each other. For testcase E3T2, we assume that a signal in class γ_1 is not correlated with (i.e., does not switch at the same time as) the signals in class γ_0.

TABLE V: Results for impact of correlation of signals.

Testcase	l	l'	$\|\gamma_l\|$	$\|\gamma_{l'}\|$	λ_l	$\lambda_{l'}$	D_l^{max}	$D_{l'}^{max}$	Objective	Runtime (s)
E3T1	0	1	4	1	10	6.7	67.4	67.2	674	1
E3T2	0	1	4	1	10	6.7	47.9	0	479	1

Figure 8(b) and Figure 8(c) show the optimized channel layouts for testcases E3T1 and E3T2. The orange line represents the signal in γ_1. In E3T2, we see that the orange line is in the center of the channel since it is not correlated with other signals. Due to this zero correlation in E3T2, the maximum delay uncertainties of γ_0 and γ_1 are smaller in E3T2 (47.9ps and 0ps) than in E3T1 (67.4ps and 67.2ps).

B. Comparison of MILP-Based, Pair-Swapping and Signal Permutation Approaches

We now describe experimental results comparing all of our methods on the artificial (realistic) channel instances. We generate testcase T1 with parameters as shown in Table VI to (i) study the variation of the solution quality for MILP-based approach according to the size of the decomposition subset, and (ii) compare the solutions across our two approaches along with the solution from the typical signal permutation approach. We also show the result using the method of [17] (i.e., MILP with linear delay uncertainty model) for the small testcase T1. For a typical signal permutation, we apply our pair-swapping approach to the entire channel at a time (i.e, the number of segments is equal to one). For the testcase T1, we use $8000\mu m$ channel length and determine the correlations between signals based on Table II.

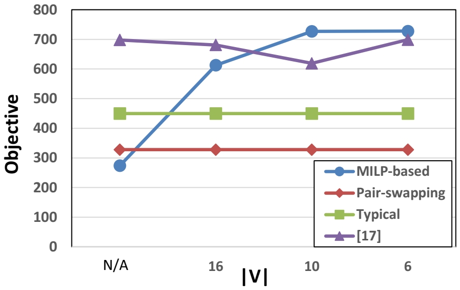

Fig. 10: Comparison of objective values among different methods.

Rows 1-10 of Table VI show the results from our MILP-based approach for different sizes of the decomposition subset (i.e., no decomposition (N/A), 16, 10, 6), along with the results from pair-swapping, typical signal permutation approaches, and the method in [17]. As visualized in Figure 10, as the size of the subset for decomposition decreases, the maximum weighted delay uncertainty increases. The MILP-based approach without decomposition achieves 39% reduction of weighted maximum delay uncertainty compared to the typical signal permutation result. However, once we decompose the testcase, the result from the MILP-based method can become even worse than that of typical signal permutation. Although pair-swapping is worse than the MILP-based approach without decomposition, it is much faster than the MILP-based approach. We also notice that the result using linear delay uncertainty model in [17] to calculate the objective function value is worse than the results from pair-swapping and typical signal permutation. This indicates that inaccurate modeling of delay uncertainty degrades solution quality (i.e., results in larger objective function value).

We also generate larger testcases T2-T5 to evaluate the scalability of our methods. The details of $|S|$, $|T|$, V and the number of signals in each class are shown in Table VI. The channel length is $8000\mu m$ and the number of segments is 16. We determine the correlations for all signals based on the probabilities in Table II.

For the overall results (Rows 11-22 of Table VI), pair-swapping always achieves better results than the MILP-based approach or typical signal permutation. For 200 signals on 200 tracks (T2), pair-swapping achieves 24% reduction of maximum weighted delay uncertainty compared with the typical signal permutation approach; this corresponds to 34ps of absolute delay uncertainty reduction for class γ_2. We see the benefit of additional area resource (i.e., empty tracks) when we compare the results between T2 and T3 (resp. T4 and T5): for pair-swapping, MILP-based and typical signal permutation approaches, we observe that assigning 10% empty tracks reduces the maximum weighted delay uncertainty by up to 15%, 4.5% and 19.9% (resp. 9.2%, 3.1% and 9.3%).

TABLE VI: Comparison of MILP-based approach, pair-swapping approach, typical signal permutation approach, and approach in [17].

		Testcase	$\|S\|$	$\|T\|$	V	$\|\gamma_0\|$	$\|\gamma_1\|$	$\|\gamma_2\|$	$\|\gamma_3\|$	$\|\gamma_4\|$	D_0^{max}	D_1^{max}	D_2^{max}	D_3^{max}	D_4^{max}	Objective	Runtime (s)
1	MILP-based approach	T1	30	30	N/A	5	12	9	3	1	26.8	40.2	68.5	133.0	148.5	274	3223
2		T1	30	30	16	5	12	9	3	1	61.3	63.2	131.1	183.2	213.6	613	598
3		T1	30	30	10	5	12	9	3	1	72.7	61.7	122.4	187.3	187.1	727	50
4		T1	30	30	6	5	12	9	3	1	72.8	67.5	131.8	217.8	139.4	728	12
5	[17]	T1	30	30	N/A	5	12	9	3	1	69.8	62.4	147.4	233.6	0	698	1339
6		T1	30	30	16	5	12	9	3	1	68.1	66.6	131.1	197.6	150.1	681	872
7		T1	30	30	10	5	12	9	3	1	61.9	59.5	142.5	190.4	181.0	619	116
8		T1	30	30	6	5	12	9	3	1	69.9	57.3	133.1	233.9	184.8	699	13
9	Pair-swapping	T1	30	30	N/A	5	12	9	3	1	31.3	48.4	81.9	163.5	140.0	328	12
10	Typical	T1	30	30	N/A	5	12	9	3	1	42.1	56.3	77.3	224.8	0.0	450	12
11	Pair-swapping approach	T2	100	100	N/A	15	40	30	10	5	39.0	58.0	98.3	191.9	390.5	393	982
12		T3	100	110	N/A	15	40	30	10	5	33.1	49.1	83.4	166.1	323.9	334	1600
13		T4	200	200	N/A	30	80	60	20	10	41.8	60.9	105.2	210.7	419.2	421	15404
14		T5	200	220	N/A	30	80	60	20	10	38.1	56.8	95.5	191.1	380.5	382	24224
15	MILP-based approach	T2	100	100	20	15	40	30	10	5	66.8	71.6	139.8	222.4	359.2	668	1846
16		T3	100	110	20	15	40	30	10	5	63.8	65.5	126.0	183.1	232.9	638	1744
17		T4	200	200	20	30	80	60	20	10	68.4	73.9	153.3	236.0	383.4	684	43476
18		T5	200	220	20	30	80	60	20	10	66.3	72.4	132.0	191.5	255.6	663	52804
19	Typical signal permutation	T2	100	100	N/A	15	40	30	10	5	53.4	71.7	139.2	252.4	442.5	557	52
20		T3	100	110	N/A	15	40	30	10	5	35.2	54.4	100.2	222.8	442.5	446	113
21		T4	200	200	N/A	30	80	60	20	10	53.4	71.7	139.2	252.4	447.1	557	825
22		T5	200	220	N/A	30	80	60	20	10	42.1	71.7	116.6	252.4	447.1	505	1758

VI. Conclusions

In this work, we have proposed a DRAM routing channel optimization to specifically target the layout design of long, resource-constrained channels in modern DRAM products. Our optimization is signal criticality-aware, and minimizes a maximum weighted delay uncertainty objective. We develop a new, accurate and efficient closed-form delay uncertainty expression, based on the crosstalk noise model in [13] and the Delay Change Curve from [9], to guide our optimization. Further, based on the swizzling technique [7] and the MILP formulation insight from [17], we propose an MILP-based channel optimization methodology, which permutes and swizzles signals with different criticalities in the routing channel – processing the channel in a greedy, segment-by-segment fashion – to reduce the worst-case weighted delay uncertainty. We observe that the optimality of the MILP is compromised when we add a decomposition strategy to maintain practical runtimes. Thus, we also propose a pair-swapping approach to optimize the permutation of signals within a given segment of the channel. In our experiments, the pair-swapping approach outperforms the MILP plus decomposition approach when instance complexities grow large. Overall, our experimental results show that the proposed methodologies can achieve up to 24% reduction of maximum weighted delay uncertainty (and, absolute reductions of up to 34ps of maximum delay uncertainty), compared to a typical track permutation methodology.

Finally, we note that our current studies fix the positions and sizes of buffers along the routing channel for each signal. However, flexibility of the buffer intervals would afford freedom to further reduce crosstalk-induced delay uncertainties. Furthermore, some DRAM design methodologies may permit use of inverting repeaters, rather than buffers, along the channel. Combined with more flexible buffer locations, the use of inverters can permit nearly noise-free layout designs through "inverter staggering" [5]. Our ongoing work extends our present work to include these design levers, while maintaining constraints on (repeaters, tracks, power, etc.) resources.

References

[1] T.-Y. Oh, Y.-S. Sohn, S.-J. Bae, M.-S. Park, J.-H. Lim, Y.-K. Cho, et al., "A 7Gb/s/pin GDDR5 SDRAM with 2.5ns Bank-to-Bank Active Time and No Bank-Group Restriction", *Proc. ISSCC*, 2010, pp. 434-436.

[2] S. S. Sapatnekar, "A Timing Model Incorporating the Effect of Crosstalk on Delay and its Application to Optimal Channel Routing", *IEEE T-CAD* 19(5) (2000), pp. 550-559.

[3] L. H. Chen and M. M. Sadowska, "Aggressor Alignment for Worst-Case Coupling Noise", *Proc. ISPD*, 2000, pp. 48-54.

[4] M. Redeker, B. F. Cockburn and D. G. Elliott, "An Investigation into Crosstalk Noise in DRAM Structures", *Proc. MTDT*, 2002, pp. 123-129.

[5] A. B. Kahng, S. Muddu, E. Sarto and R. Sharma, "Interconnect Tuning Strategies for High-Performance ICs", *Proc. DATE*, 1998, pp. 471-478.

[6] P. D. Gross, R. Arunachalam, K. Rajagopal and L. T. Pileggi, "Determination of Worst-Case Aggressor Alignment for Delay Calculation", *Proc. ICCAD*, 1998, pp. 212-219.

[7] P. Gupta and A. B. Kahng, "Wire Swizzling to Reduce Delay Uncertainty Due to Capacitive Coupling", *Proc. VLSI Design*, 2004, pp. 431-436.

[8] T. Xiao and M. Marek-Sadowska, "Efficient Delay Calculation in Presence of Crosstalk", *Proc. ISQED*, 2000, pp. 491-497.

[9] T. Sato, Y. Cao, K. Agarwal, D. Sylvester and C. Hu, "Bidirectional Closed-form Transformation Between On-Chip Coupling Noise Waveforms and Interconnect Delay-Change Curves", *IEEE T-CAD* 22(5) (2000), pp. 560-572.

[10] F. Dartu and L. T. Pillage, "Calculating Worst-Case Gate Delays Due to Dominant Capacitance Coupling", *Proc. DAC*, 1997, pp. 46-51.

[11] S. Nazarian and M. Pedram, "Crosstalk-affected Propagation Delay in Nanometer Technologies", *J. Electronics* 95(9) (2008), pp. 903-937.

[12] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov and J. Zuo, "Driver Modeling and Alignment for Worst-Case Delay Noise", *Proc. DAC*, 2001, pp. 720-725.

[13] J. Cong, D. Z. Pan and P. V. Srinivas, "Improved Crosstalk Modeling for Noise Constraint Interconnect Optimization", *Proc. ASP-DAC*, 2001, pp. 373-378.

[14] R. Gandikota, K. Chopra, D. Blaauw and D. Sylvester, "Victim Alignment in Crosstalk-Aware Timing Analysis", *IEEE T-CAD* 29(2) (2010), pp. 261-274.

[15] A. B. Kahng, S. Muddu and E. Sarto, "On Switch Factor Based Analysis of Coupled RC Interconnects", *Proc. DAC*, 2000, pp. 79-84.

[16] V. Zolotov and P. Feldmann, "Variation Aware Cross-Talk Aggressor Alignment by Mixed Integer Linear Programming", *Proc. DAC*, 2015, pp. 164-169.

[17] T. Gao and C. L. Liu, "Minimum Crosstalk Channel Routing", *Proc. ICCAD*, 1993, pp. 692-696.

[18] A. Vittal and M. Marek-Sadowska, "Crosstalk Reduction for VLSI", *IEEE T-CAD* 16(3) (1997), pp. 290-298.

[19] P. Saxena and C. L. Liu, "a Postprocessing Algorithm for Crosstalk-Driven Wire Perturbation", *IEEE T-CAD* 19(6) (2000), pp. 691-702.

[20] D. Wu, J. Hu, M. Zhao and R. Mahapatra, "Timing Driven Track Routing Considering Coupling Capacitance", *Proc. ASP-DAC*, 2005, pp. 1156-1159.

[21] L. Ding, D. Blaauw and P. Mazumder, "Efficient Crosstalk Noise Modeling Using Aggressor and Tree Reductions", *Proc. ICCAD*, 2002, pp. 595-600.

[22] C. Hwang and M. Pedram, "Interconnect Design Methods for Memory Design", *Proc. ASP-DAC*, 2004, pp. 438-443.

[23] G. Zhong, C.-K. Koh and K. Roy, "A Twisted-Bundle Layout Structure for Minimizing Inductive Coupling Noise", *Proc. ICCAD*, 2000, pp. 406-411.

[24] C.-C. Chang and J. Cong, "Pseudo Pin Assignment with Crosstalk Noise Control", *Proc. ISPD*, 2000, pp. 41-47.

[25] H. Yu, L. He and M.-C. F. Chang, "Robust On-Chip Signaling by Staggered and Twisted Bundle", *IEEE Design & Test of Computers*, 26(5) (2009), pp. 92-104.

[26] *IBM ILOG CPLEX*. www.ilog.com/products/cplex/

[27] *Synopsys RAPHAEL User Guide*. https://www.synopsys.com/TOOLS/TCAD/INTERCONNECTSIMULATION/Pages/Raphael.aspx