Orthogonal Ray Graphs and Nano-PLA Design

Anish Man Singh Shrestha, Satoshi Tayu, and Shuichi Ueno
Department of Communications and Integrated Systems
Tokyo Institute of Technology, Tokyo 152-8550-S3-57, Japan

Abstract—The logic mapping problem and the problem of finding a largest square sub-crossbar with no defects in a nano-crossbar with nonprogrammable crosspoint defects and disconnected wire defects have been known to be NP-hard. This paper shows that for nano-crossbars with only disconnected wire defects, the former remains NP-hard, while the latter can be solved in polynomial time.

I. INTRODUCTION

The problem of mapping a logic function onto a defective nano-crossbar with nonprogrammable crosspoint defects and disconnected wire defects was first considered by Rao, OraIologlu, and Karri [4]. They proposed several heuristics since the problem is NP-hard. The problem of finding a maximum defect-free square sub-crossbar in a nano-crossbar with nonprogrammable crosspoint defects and disconnected wire defects was first investigated by Tahoori [6]. Since the problem is also NP-hard, several heuristics have been proposed [1], [6]. This paper considers the complexity of the problems for nano-crossbars with only disconnected wire defects.

I-A. LOGIC MAPPING

Let $f$ be a logic function in a sum-of-product form. The problem of implementing $f$ in a surviving sub-crossbar $S$ of a nano-crossbar with disconnected wire defects is formulated as LOGIC MAPPING, which is the problem of assigning the literals and product terms of $f$ to surviving nano-wires of $S$ so that containment relationships among the literals and product terms can be represented by crosspoint connections in $S$. A graph model of LOGIC MAPPING can be obtained as follows.

Let $L_f$ be the set of literals of $f$, and $P_f$ be the set of product terms of $f$. A logic function graph $G_f$ for $f$ is a bipartite graph defined as follows: $V(G_f) = L_f \cup P_f$, and $(L_f, P_f)$ is a bipartition of $G_f$; vertices $l \in L_f$ and $p \in P_f$ are connected by an edge if and only if literal $l$ is contained in product term $p$.

Let $W_h$ be the set of surviving horizontal nano-wires, and $W_v$ be the set of surviving vertical nano-wires of $S$. A surviving sub-crossbar graph $G_S$ for $S$ is a bipartite graph defined as follows: $V(G_S) = W_h \cup W_v$ and $(W_h, W_v)$ is a bipartition of $G_S$; vertices $x \in W_h$ and $y \in W_v$ are connected by an edge if and only if nano-wires $x$ and $y$ have a crosspoint. Then, LOGIC MAPPING can be modeled as the subgraph isomorphism problem, which is to find a subgraph of $G_S$ isomorphic to $G_f$. An example of a logic function $f$, a defective crossbar $S$, and their corresponding bipartite graphs $G_f$ and $G_S$ is shown in Figure 1.

I-B. SQUARE SUB-CROSSBAR

SQUARE SUB-CROSSBAR is the problem of finding a maximum defect-free square sub-crossbar within the original nano-crossbar with disconnected wire defects. SQUARE SUB-CROSSBAR can be modeled as the balanced complete bipartite subgraph problem, which is to find a complete bipartite graph $K_{k,k}$ contained in $G_S$.

I-C. OUR RESULTS

Although it is well known that both the subgraph isomorphism problem and the balanced complete bipartite subgraph problem are NP-hard for bipartite graphs [2], [3], the complexity of LOGIC MAPPING and SQUARE SUB-CROSSBAR is not clear since the graphs representing surviving sub-crossbars are a special kind of bipartite graphs.

A bipartite graph $G$ with a bipartition $(U, V)$ is called an orthogonal ray graph if there exist a family of non-intersecting rays (half-lines) $R_u, u \in U$, parallel to the $x$-axis in the $xy$-plane, and a family of non-intersecting rays $R_v, v \in V$, parallel to the $y$-axis such that for any $u \in U$ and $v \in V$, $(u, v) \in E(G)$ if and only if $R_u$ and $R_v$ intersect.

Nano-wires such as $m$ and $n$ of a defective nano-crossbar shown in Figure 2 cannot be controlled as they do not touch the
boundary of the originally intended nano-crossbar. Since we cannot use such nano-wires, a graph representing a surviving sub-crossbar must be an orthogonal ray graph. The orthogonal ray graph was introduced by Shrestha, Kobayashi, Tayu, and Ueno [5] as a graph model for a surviving sub-crossbar.

We show in Section III that LOGIC MAPPING is NP-hard by showing that the subgraph isomorphism problem is NP-hard even for orthogonal ray graphs. We also show in Section IV that SQUARE SUB-CROSSBAR can be solved in polynomial time provided that the vertices of the orthogonal ray graph representing a surviving sub-crossbar are ordered so as to reflect the position of nano-wires relative to each other, which is a quite natural condition.

II. ORTHOGONAL RAY GRAPHS

Let \( G \) be an orthogonal ray graph with a bipartition \((U, V)\). \( G \) is called a two-directional orthogonal ray graph if \( R_u = \{ (x, b_v) \mid x \geq a_u \} \) for each \( u \in U \), and \( R_v = \{ (a_v, y) \mid y \geq b_v \} \) for each \( v \in V \), where \( a_w \) and \( b_w \) are real numbers for any \( w \in U \cup V \). The 3-claw is a tree obtained from a complete bipartite graph \( K_{1,3} \) by replacing each edge with a path of length 3. (See Figure 3(a).)

Although the following characterization of two-directional orthogonal ray trees was shown in [5], we show complete proofs to make the paper self-contained.

**Lemma 1:** The 3-claw is not a 2-directional orthogonal ray graph.

**Proof:** Assume to the contrary that the 3-claw is a 2-directional orthogonal ray graph. Let the vertices of the 3-claw be named as in Figure 3(a). We shall refer to the endpoint of the ray corresponding to a vertex \( v \) as \((a_v, b_v)\). Without loss of generality, suppose \( R_{u1} \) is a horizontal ray and that \( R_{u1}, R_{v2}, R_{v3} \) intersect with \( R_{u1} \) such that \( R_{v2} \) lies to the right of \( R_{u1} \), and to the left of \( R_{v3} \) as shown in Figure 3(b). It is easy to observe that \( b_{v2} > b_{v3} \) or else it is not possible to define \( R_{u2}, R_{u3}, \) and \( R_{u4} \). Since \( R_{u3} \) has to be defined such that \( a_{u3} > a_{v1} \) and \( b_{u3} < b_{v1} \), it is not possible to define \( R_{u3} \) such that it intersects with \( R_{u3} \) but not with \( R_{u1} \), a contradiction.

A path \( P \) in a tree \( T \) is called a spine of \( T \) if every vertex of \( T \) is within distance two from at least one vertex of \( P \).

**Theorem 1:** A tree \( T \) has a spine if and only if \( T \) does not contain 3-claw as a subtree.

**Proof:** The necessity is obvious. To prove the sufficiency, assume \( T \) does not contain a 3-claw. Let \( P \) be a longest path in \( T \). We claim that \( P \) is a spine. Assume it is not. Let \( V(P) = \{ v_1, v_2, \ldots, v_p \} \), and \( (v_i, v_{i+1}) \in E(P) \), \( 1 \leq i \leq p - 1 \). Let \( F \) be a forest obtained from \( T \) by deleting the edges in \( E(P) \). Let \( T_i \) be a tree in \( F \) containing \( v_i, 1 \leq i \leq p \). Since \( P \) is a longest path in \( T \), \( T_i \) consists of only one vertex, \( v_1 \), and \( T_{p-1} \) consists of only one vertex, \( v_p \). Also all vertices in \( T_2 \) and \( T_{p-1} \) are within distance one from \( v_2 \) and \( v_{p-1} \), respectively; and all vertices in \( T_3 \) and \( T_{p-2} \) are within distance two from \( v_3 \) and \( v_{p-2} \), respectively. Since we assumed that \( P \) is not a spine, there exists an integer \( j \) (\( 4 \leq j \leq p - 3 \)) such that \( T_j \) contains a vertex \( w_j \) whose distance from \( v_j \) is three. Let \( P' \) be the path from \( v_j \) to \( w_j \). Then the subgraph of \( T \) induced by the vertices in \( \{ v_i \mid j - 3 \leq i \leq j + 3 \} \cup V(P') \) is a 3-claw. This contradicts the assumption that \( T \) does not contain 3-claw as a subtree, and therefore \( P \) is a spine.

**Theorem 2:** A tree \( T \) is a 2-directional orthogonal ray tree if and only if \( T \) does not contain 3-claw as a subtree.

**Proof:** The necessity follows from Lemma 1. We will show the sufficiency. Assume \( T \) does not contain 3-claw as a subtree. Then from Theorem 1, \( T \) contains a spine \( P \). Let \( V(P) = \{ v_1, v_2, \ldots, v_p \} \), and \( (v_i, v_{i+1}) \in E(P) \), \( 1 \leq i \leq p - 1 \). Corresponding to each vertex \( v_i \) in \( P \), define ray \( R_{v_i} = \{ (i, y) \mid y \geq i - 1 \} \) if \( i \) is odd, and define ray \( R_{v_i} = \{ (i, x) \mid x \geq i - 1 \} \) if \( i \) is even. Let \( F \) be a forest obtained from \( T \) by deleting the edges in \( E(P) \). Let \( T_i \) be a tree in \( T \) containing \( v_i, 1 \leq i \leq p \). Consider \( T_i \) to be rooted at \( v_i \). Let \( w_{i1}, w_{i2}, \ldots, w_{iq(i)} \) be the children of \( v_i \) in \( T_i \), where \( q(i) \) is the number of children of \( v_i \) in \( T_i \). Let \( z_{ij}, z_{j}, z_{ij+1}, \ldots, z_{ij+r(i)} \) be the children of \( w_{ij} \) in \( T_i \), where \( r(i) \) is the number of children of \( w_{ij} \) in \( T_i \). The rays corresponding to \( w_{ij} \) and \( z_{jk} \), \( 1 \leq i \leq p, 1 \leq j \leq q(i), 1 \leq k \leq r(i) \), can be added as shown in Figure 4. Thus \( T \) is a 2-directional orthogonal ray graph.

III. INTRACTABILITY OF LOGIC MAPPING

We show in this section the following.

**Theorem 3:** LOGIC MAPPING is NP-hard.
Theorem 3 follows from Theorem 4 below. A decision problem associated with the subgraph isomorphism problem is stated as follows.

**SUBGRAPH ISOMORPHISM**

**INSTANCE:** Graphs $H$ and $G$.

**QUESTION:** Does $G$ contain a subgraph isomorphic to $H$, that is, does there exist a one-to-one mapping $\phi : V(H) \to V(G)$ such that if $(u, v) \in E(H)$ then $(\phi(u), \phi(v)) \in E(G)$?

**Theorem 4:** SUBGRAPH ISOMORPHISM is NP-complete even if $G$ is a 2-directional orthogonal ray tree and $H$ is a forest.

**Proof:** It is easy to see that the problem is in NP. We show a polynomial time reduction from 3-PARTITION, which has been shown to be strongly NP-complete in [2]. 3-PARTITION is defined as follows.

**3-PARTITION**

**INSTANCE:** A finite set $A$ of $3m$ elements, a bound $B \in \mathbb{Z}^+$, and a size $s(a) \in \mathbb{Z}^+$ for each $a \in A$, such that each $s(a)$ satisfies $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$.

**QUESTION:** Does $A$ have a 3-partition, that is, can $A$ be partitioned into $m$ disjoint sets $S_1, S_2, \ldots, S_m$ such that, for $1 \leq i \leq m$, $\sum_{a \in S_i} s(a) = B$?

Let $A = \{a_1, a_2, \ldots, a_{3m}\}$, $B \in \mathbb{Z}^+$, and $s(a_1), s(a_2), \ldots, s(a_{3m}) \in \mathbb{Z}^+$ be an instance of 3-PARTITION in which $\max_{a \in A} \{s(a)\}$ is bounded by a polynomial of the size of the instance. We shall construct a 2-directional orthogonal ray tree $G$ and a forest $H$ as follows.

Let $C_1, C_2, \ldots, C_m$ be $B$-vertex chains such that for each $i$ ($1 \leq i \leq m$), $V(C_i) = \{v_{i,j} \mid 1 \leq j \leq B\}$ and $E(C_i) = \{(v_{i,j}, v_{i,(j+1)}) \mid 1 \leq j \leq B - 1\}$. Let $T_1, T_2, \ldots, T_{m-1}$ be complete binary trees of height two rooted at vertices $r_1, r_2, \ldots, r_{m-1}$, respectively. Let $G$ be the graph defined as

$$V(G) = \left(\bigcup_{i=1}^{m} V(C_i)\right) \cup \left(\bigcup_{i=1}^{m-1} V(T_i)\right),$$

$$E(G) = \left(\bigcup_{i=1}^{m} E(C_i)\right) \cup \left(\bigcup_{i=1}^{m-1} E(T_i)\right) \cup \{(r_i, v_{i,B}), (r_i, v_{(i+1),1}) \mid 1 \leq i \leq m - 1\}.$$  

(See Figure 5(a).) Since the path in $G$ from $v_{1,1}$ to $v_{m,B}$ is a spine of $G$, it follows from Theorems 1 and 2 that $G$ is a two-directional orthogonal ray tree. Let $H$ be a forest consisting of $m - 1$ complete binary trees of height two $T'_1, T'_2, \ldots, T'_{m-1}$, and $3m$ chains $P_1, P_2, \ldots, P_{3m}$, each $P_j$ corresponding to element $a_j$ of $A$ and having $s(a_j)$ vertices. (See Figure 5(b).) $G$ and $H$ can be constructed in time polynomial in $m$ and $B$.

We next prove that $A$ has a 3-partition if and only if $G$ contains a subgraph isomorphic to $H$.

Suppose first that $A$ can be partitioned into $m$ disjoint subsets $S_1, S_2, \ldots, S_m$ such that for each $i$ ($1 \leq i \leq m$), $\sum_{a \in S_i} s(a) = B$. An isomorphism from $H$ to a subgraph of $G$ can be obtained as follows. Since each chain $C_i$ contains $B$ vertices, we can map the chains of $H$ corresponding to the elements of $S_i$ to the chain $C_i$ in $G$. Each $T'_i$ in $H$ can be mapped to $T_i$ in $G$. It is easy to see that this is indeed an isomorphism from $H$ to a subgraph of $G$.

Next suppose that $H$ is isomorphic to a subgraph of $G$. Each $T'_i$ ($1 \leq i \leq m - 1$) in $H$ contains two vertices which have degree three and are at a distance two from each other. For a pair of vertices in $G$, the same is true only if the two vertices are the children of vertex $r_i$ in $T_i$ for any $i$ ($1 \leq i \leq m - 1$). Therefore, each $T'_i$ in $H$ must be mapped to some $T_i$ in $G$. This means that chains $P_1, P_2, \ldots, P_{3m}$ in $H$ are mapped to chains $C_1, C_2, \ldots, C_m$ in $G$. For $1 \leq i \leq m$, let $S_i$ be the set of elements of $A$ corresponding to the paths of $H$ mapped to $C_i$. Since $C_i$ has $B$ vertices, $\sum_{a \in S_i} s(a) \leq B$.
for all $i$ ($1 \leq i \leq m$). Moreover, since the instance of 3-PARTITION satisfies $\sum_{a \in A} s(a) = mB$, we can conclude that $\sum_{a \in S_i} s(a) = B$ for all $i$ ($1 \leq i \leq m$). Therefore $A$ has a 3-partition.

IV. TRACTABILITY OF SQUARE SUB-CROSSBAR

IV-A. Two-Directional Orthogonal Rays

If we restrict the instance of SQUARE SUB-CROSSBAR such that all horizontal rays are directed towards the right and all vertical rays are directed upwards, we can solve the problem with a simple algorithm outlined in Figure 6, where we consider a decision problem associated with SQUARE SUB-CROSSBAR for simplicity. It is not difficult to see the following:

Theorem 5: Algorithm 1 solves a decision problem associated with SQUARE SUB-CROSSBAR in the instance restricted to rightward or upward rays in $O((|H| + |V|)^3)$ time.

IV-B. General Orthogonal Rays

We shall next extend Algorithm 1 to cover the case for general orthogonal rays.

Let $R_X$ be a set of horizontal rays and $R_Y$ be a set of vertical rays. Suppose two rays $R_x \in R_X$ and $R_y \in R_Y$ intersect at point $P$. Define $R^{xy}_Y \subseteq R_Y$ to be the set of rays that intersect with $R_x$ and are to the left of $P$. Similarly define $R^{xy}_X \subseteq R_X$ to be the set of rays that intersect with $R_y$ and are below $P$. Let $(x_L, y_L)$ be the point where the leftmost ray in $R^{xy}_Y$ intersects $R_x$, and let $(x_B, y_B)$ be the point where the bottommost ray in $R^{xy}_X$ intersects $R_y$. For each ray $R \in R^{xy}_Y$ with endpoint $(x_R, y_R)$, define ray $V_R$ as follows: $V_R = R$ if $R$ is an upward ray, and $V_R$ is an upward ray with endpoint $(x_R, y_B)$ if $R$ is a downward ray. And for each ray $R \in R^{xy}_X$ with endpoint $(x_B, y_R)$, define ray $H_R$ as follows: $H_R = R$ if $R$ is a rightward ray, and $H_R$ is a rightward ray with endpoint $(x_L, y_R)$ if $R$ is a leftward ray. Finally, define $V^{xy} = \{V_R \mid R \in R^{xy}_X\}$, and define $H^{xy} = \{H_R \mid R \in R^{xy}_X\}$.

The following observation is obvious from the definitions above.

Observation 1: Two rays in $V^{xy} \cup H^{xy}$ intersect if and only if their corresponding rays in $R^{xy}_Y \cup R^{xy}_X$ intersect.

Input: A set $H$ of rightward rays, a set $V$ of upward rays, and an integer $k$.

Output: YES if $H \cup V$ contains a $k \times k$ sub-crossbar, NO, otherwise.

Step 1: If $H$ or $V$ is empty, output NO and halt. Else, set $B$ to be the bottommost ray in $H$ and set $L$ to be the leftmost ray in $V$.

Step 2: Set $n_B$ to be the number of rays in $V$ that intersect with $B$, and set $n_L$ to be the number of rays in $H$ that intersect with $L$.

Step 3: If $n_B \geq k$ and $n_L \geq k$, output YES.

Step 4: If $n_B < k$, set $H = H - \{B\}$.

Step 5: If $n_L < k$, set $V = V - \{L\}$.

Step 6: Return to Step 1.

Fig. 6. Algorithm 1.

Fig. 7. Algorithm 2.

Observation 2: $R_X \cup R_Y$ contains a $k \times k$ surviving sub-crossbar if and only if there exists a pair of intersecting rays $R_x \in R_X$ and $R_y \in R_Y$ such that $H^{xy} \cup V^{xy}$ contains a $(k - 1) \times (k - 1)$ surviving sub-crossbar.

Proof: The sufficiency is immediate from Observation 1. To see the necessity, set $R_x$ and $R_y$ to be the topmost and rightmost rays, respectively of a $k \times k$ sub-crossbar. Figure 7 shows Algorithm 2 which uses Algorithm 1 as a subroutine.

Algorithm 2 exhaustively checks all pairs of intersecting rays to determine if there exists a pair $R_x \in R_X$ and $R_y \in R_Y$ such that $H^{xy} \cup V^{xy}$ contains a $(k - 1) \times (k - 1)$ surviving sub-crossbar. Therefore, from Observation 2 and Theorem 5, we obtain the following.

Theorem 6: Algorithm 2 solves a decision problem associated with SQUARE SUB-CROSSBAR in $O((|R_X| + |R_Y|)^4)$ time.

V. CONCLUDING REMARKS

It should be noted that Algorithm 2 can be easily modified for the search version and the original optimization version of SQUARE-CROSSBAR. It should also be noted that Algorithm 2 can be used to decide the presence of a $k \times k$ sub-crossbar even if the input sets $R_X$ and $R_Y$ contain line segments instead of rays. Moreover, Algorithm 2 can be easily modified to decide the presence of an $m \times n$ sub-crossbar for any positive integers $m$ and $n$. It is an interesting open question to reduce the complexity of Algorithms 1 and 2.

REFERENCES