How to Consider Shorts and Guarantee Yield Rate Improvement for Redundant Wire Insertion

Fong-Yuan Chang\textsuperscript{1,2} \ fongyuan.chang@gmail.com \ Ren-Song Tsay\textsuperscript{1} \ rtsay@cs.nthu.edu.tw \ Wai-Kei Mak\textsuperscript{1} \ wkmak@cs.nthu.edu.tw

ABSTRACT
This paper accurately considers wire short defects and proposes an algorithm to guarantee IC chip yield rate improvement for redundant wire insertion. Without considering yield rate degradation caused by shorts, traditional methods may even lead to yield rate loss. However, shorts are more complicated to analyze than opens. Moreover, since any two points of a routed net can be connected by a redundant wire, the number of possible insertion patterns for a chip is un-tractable. To maximize yield rate improvement and to make the problem tractable, we identify a key insight, tolerance-ratio, as an effective guide for choosing insertion patterns and insertion order. Finally, to guarantee yield rate improvement, only positive gain redundant wires are committed. Experimental results show that, compared with unprocessed cases, all yield rate improvements in the proposed algorithm are positive, and the defect rates are reduced by up to 65\% and by 24\% on average. On the other hand, without considering shorts, the defect rate can increase as much as 7\%.

Categories and Subject Descriptors: B.7.2
[Integrated Circuits]: Design Aids - Layout, Placement and Routing.


Keywords: Redundant Wire Insertion, Shorts, Opens, Yield Rate.

1. INTRODUCTION
In current complex chip designs, fully functional interconnections are essential to chip yield. A chip is considered good in this paper if there are no opens or shorts, i.e. wiring defects. A net is open if any pin pair on the net is unconnected. A short occurs when any part of a net connects to different nets. As semiconductor process advances, yield rate formerly dominated by opens is now becoming more short critical\textsuperscript{12} [13]. Hence, to increase chip yield, opens and shorts must both be considered.

The two main types of wiring defects are manufacturing defects and electromigration effects [1]. Manufacturing defects are mainly caused by random defect and process variation [2]. One source of these random defects may be undesired particles falling on the wafer and causing wire interruptions or shorts. Similarly, process variation can distort wire shapes and result in faults. Furthermore, electromigration effect [3] is the diffusion of metal atoms in a wire along the direction of electron flow. With higher operating signal frequencies and narrower wire widths, current densities are increased and hence are more likely to cause wire open faults due to severe electromigration effects.

To mitigate open and short problems, numerous researchers have proposed different approaches. For instance, non-tree routing approaches [4] [5] [6] add redundant wires to a routed net to tolerate open faults. Double via insertion approaches [7] [8] add redundant vias to ease via fault problems, and the local loop method [9] is another approach to tolerate via faults. Furthermore, the wire spreading and widening approach [10] is proposed to reduce the possibilities of wiring open and short. The same concept is also applied at the track routing stage [11].

\begin{center}
\includegraphics[width=0.5\textwidth]{figure1.png}
\end{center}

\textbf{Figure 1:} With the insertion of redundant wire $r$, the opens of wire $w$ can be tolerated but the two short-critical areas, $a1$ and $a2$, are created.

Most approaches [4-9] mentioned above adopt either a redundant wire or via insertion method for yield rate improvement. However, these proposed methods either concern only opens or do not accurately consider yield rate loss caused by shorts. Therefore, the yield rate may actually deteriorate with improper insertion of redundant wires. As Figure 1 shows, the opens of $w$ can be tolerated by inserting $r$, and hence, yield rate should increase. However, at the same time, areas $a1$ and $a2$ become critical to shorts and yield rate will decrease. Depending on the process and actual wiring situations, the yield rate may even end up deteriorating. Hence, the effects of shorts have to be considered carefully.

This work is partially supported by National Science Council under Grant No. NSC 98-2220-E-007-012

\textsuperscript{1}Department of Computer Science National Tsing-Hua University, Hsinchu, Taiwan
\textsuperscript{2}Springsoft Inc.
Moreover, since any two points of a routed net can be connected by a redundant wire, the number of possible wire insertion patterns for a chip is untractable. Thus, the choice of redundant wires is a major problem. Additionally, the order of insertions is observed to be critical to the final yield rate. Therefore, an intelligent and efficient algorithm is needed.

For these reasons, to date yield rate improvement by redundant wire insertion has been of theoretical interest only. Hence, this paper focuses on the redundant wire insertion problem and devises an equation to estimate yield rate change. Moreover, the equation identifies a key aspect, tolerance-ratio, as an effective guide to dealing with the issues mentioned above. In the proposed algorithm, only positive gain redundant wires are committed. Hence, the proposed algorithm guarantees yield rate improvement. Experimental results show that, compared with unprocessed cases, all yield rate improvements in the algorithm are positive and the defect rates are reduced by an average of 24%, up to 65%. In comparison, without considering shorts, the defect rate can increase as much as 7%.

The rest of this paper is organized as follows: Section 2 shows how previous methods may result in yield rate loss. Section 3 introduces an equation to estimate yield rate changes and the key insight, tolerance-ratio. Section 4 formulates the redundant wire insertion problem. Then Section 5 analyzes problems observed for redundant wire insertion and proposes an algorithm. Finally, experimental results of four real industrial cases are given to demonstrate the efficacy of the proposed algorithm.

2. YIELD RATE LOSS

This section describes the reasons why traditional methods result in yield rate loss.

First, we show that how methods [4] [5] [6] of redundant wire insertion may induce yield rate loss. Take the method proposed by Kahng, et al. [4] for example. Its goal is to tolerate wire opens by inserting redundant wires with a given maximum total redundant wire length. However, this method does not consider the short problem and realistic wiring issues, like wiring detours. In Figure 1, a redundant wire is inserted to connect two pins of a net. Depending on process situations, the yield rate decrease caused by shorts may be more than the yield rate increase caused by tolerance of opens. Hence, the chip yield rate can be reduced.

Moreover, methods [8][9] for via open tolerance could also cause yield rate loss. For example, the method proposed by J. Bickford, et al. [9] is to tolerate opens for dead vias, i.e., the vias that cannot be replaced by double vias, by creating local loops containing dead vias. Nevertheless, the creation of local loops may generate new short-critical areas which would decrease the yield rate. Similarly, the yield rate decrease by shorts may larger than the increase by tolerance of dead vias. Thus, chip yield rate is reduced. Likewise, for the method proposed by Cheok-Kei, et al. [8], it uses a wire spreading technique to increase double via insertion rate. However, when spreading wires, new wires and new short-critical areas may be created, which would decrease the yield rate.

In this paper, we focus on the redundant wire insertion problem. We start from devising a simplified equation to estimate yield rate changes caused by inserting redundant wires.

3. YIELD RATE CHANGE EQUATIONS

In this section, we analyze the short problem and devise an equation to estimate yield rate changes caused by inserting redundant wires.

### Table 1: Notations.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o$</td>
<td>The open rate per unit wire length.</td>
</tr>
<tr>
<td>$s$</td>
<td>The short rate per parallel run length.</td>
</tr>
<tr>
<td>$r$</td>
<td>A redundant wire/edge.</td>
</tr>
<tr>
<td>$r_l$</td>
<td>The length of a redundant wire/edge $r$.</td>
</tr>
<tr>
<td>$L$</td>
<td>The total wire length of a chip.</td>
</tr>
<tr>
<td>$C$</td>
<td>The total length of wires on loops of a chip.</td>
</tr>
<tr>
<td>$C_r$</td>
<td>The increase of $C$ caused by inserting $r$.</td>
</tr>
<tr>
<td>$P$</td>
<td>The total parallel run length of a chip.</td>
</tr>
<tr>
<td>$T$</td>
<td>The total routing resource (length) of a chip.</td>
</tr>
<tr>
<td>$Y$</td>
<td>The yield rate for a chip.</td>
</tr>
</tbody>
</table>

![Figure 2: An example of parallel run length. Seven wire segments are shown in a partial layout with dashed lines indicating wiring tracks. The parallel run length is $p_1 + p_2 + p_3$.](image)

### 3.1 Short Problem Analysis

A short is a connection between wire segments of different nets. The short rate for two wire segments depends on the interleaving spacing and the defect size distribution. A widely-used model of defect size distribution and the short rate of two wire segments are discussed by W. Maly [15]. Generally in digital designs, wires are routed on tracks to maintain the required spacing. Figure 2 illustrates an example, where dashed lines represent wiring tracks and the wires are routed on tracks. Since the short rate between neighboring wires is much larger than non-neighboring ones, we considered only neighboring short effects in this paper.

Parallel run length is used to compute the short rate. The parallel run length of two wire segments is the length of their parallel portions. For example, the total parallel run length $P$ of the example in Figure 2 is $p_1 + p_2 + p_3$. For a given per parallel run length short rate $s$, the total short rate of a chip is the total parallel run length $P$ multiplied by $s$.

### 3.2 Yield Rate Change Equations

Here, we develop an equation to estimate yield rate changes. According to Kahng, et al. [4], we assume open rate is directly proportional to the total wire length. Additionally, short rate is proportional to the total parallel run length. As a result, the chip yield rate can be expressed as

$$Y = (1 - o)^L \cdot (1 - s)^P,$$

where $L$, $P$, $Y$, $o$, and $s$ are the total wire length, the total parallel run length of the chip, the open rate per unit wire length, and the short rate per parallel run length as defined in Table 1. In general, $o$ and $s$ are much smaller than one. Then, Equation (2) can be approximated as
or the total length of wires on loops of the chip, the yield rate is reduced. Increased total parallel run length, and the yield rate may be increased. Since double-open redundancy is shown in Figure 2, each wire point has two neighboring routing points. For example, the wire point a has two neighboring routing points, a₁ and a₂. The probability that a routing point is occupied by a wire is the chip routing density, i.e., L/T, where T is the total routing resource. Then, the number of occupied neighboring points is 2L (L/T). Since a parallel run point is assembled by two occupied neighboring routing points, the total parallel run length P is equal to 2L (L/T) · s. Thus, Equation (4) can be rewritten as

\[ Y \equiv L \cdot o - L \cdot (L/T) \cdot s. \]  

Clearly, the above equation indicates that when there are more routing wires, the shorting effect will increase in magnitude. Traditional approaches of redundant wire insertion for open rate reduction may actually cause increased short rates due to increased total parallel run length, and the yield rate may be reduced.

We will now discuss the effects of adding redundant wires. The example in Figure 3 shows two redundant wires, the resulting loops and newly added parallel run length. Since double-open faults are much less likely to occur than single-open faults and a loop requires at least two opens to break, we may assume that loops are free from open faults. Considering loops, with C as the total length of wires on loops of the chip, the yield rate is

\[ Y \equiv 1 - (L-C) \cdot o - L \cdot (L/T) \cdot s. \]

Therefore, the yield rate change, \( \Delta Y \), after inserting a redundant wire r is as follows:

\[ \Delta Y = (C_r - |r|) \cdot o - [(L + |r|)^2 - L^2]/T \cdot s \]  

where \( C_r \) is the increase of C by inserting r. Assume \( |r| \ll L \), then

\[ \Delta Y \equiv (C_r - |r|) \cdot o - 2 \cdot |r| \cdot (L/T) \cdot s, \]  

or

\[ \Delta Y/|r| \equiv (C_r/|r| - 1) \cdot o - 2 \cdot L/T \cdot s. \]

Equation (8) is the average yield rate change caused by inserting a redundant wire r. To guarantee yield rate improvement, Equation (8) can also be used to compute the real yield rate change.

The most important equation is Equation (9), which leads to a key aspect of our approach. Equation (9) is the average yield rate change per unit length. Since the short rate increase per unit length, 2L/T · s, is the same for any redundant wire, the average yield rate improvement per unit length depends only on the change in open rate. Most importantly, the ratio \( C_r/|r| \) alone determines how much improvement can be made. We call the ratio \( C_r/|r| \) tolerance-ratio, which can serve as an effective guide in designing our redundant wire insertion algorithm.

4. PROBLEM FORMULATION

Before presenting our proposed approach, we first formulate the redundant wire insertion problem and introduce a net projection-graph model for finding redundant wires.

4.1 The Redundant Wire Insertion Problem

As we have discussed, redundant wires can decrease the open rate, but increase the short rate. Thus, the redundant wire insertion problem is to insert a set of redundant wires to a routed design for the maximum chip yield rate, considering both opens and shorts.

As mentioned, since any two points of a routed net can be connected by a redundant wire, the number of possible wire insertion patterns for a chip is untractable. Hence, to make this problem tractable, we first design two redundant wire types which determine the possible connection points for nets. Then, to have greater yield rate improvement, problems of redundant wire insertion are discussed and the corresponding solutions are proposed.

4.2 Redundant Wire Types

Here, we classify redundant wires into two types according to the connection points. One is the widely-used pin-pin redundant wire, which connects two points that are pins or Steiner points. The other is the pin-wire redundant wire, which connects a pin and one of the pin’s projection points. The projection points of a pin are the closest points on its left, right, top and bottom wire segments. Figure 4 shows the two types of redundant wires.

Equation (9) implies that to have larger yield rate improvement per unit length, redundant wires should have larger tolerance-ratio. In general, a pin-wire redundant wire can have a larger tolerance-ratio.

4.3 The Graph Modeling of Nets

To precisely describe issues of redundant wire insertion and the proposed algorithm, we model a net as a projection-graph. We first introduce the planar graph of a net. Assuming a rectilinear

\[ \text{Figure 4: Two types of redundant wires. } n1 \text{ is a net, } r1 \text{ is a pin-pin redundant wire and } r2 \text{ is a pin-wire redundant wire.} \]
Steiner tree implements a net connection on a Manhattan plane, the corresponding planar graph of the net is formed by the pins and Steiner points as vertices, and rectilinear wire segments as edges. Thus, we define projection-graph.

Definition 1: A projection-graph of a planar graph is a graph with extra vertices that are the horizontal or vertical projection points of all vertices onto neighboring edges. An example is illustrated in Figure 5.

Each redundant wire can be represented on a projection graph as an extra edge, also called a redundant edge, connecting two vertices.

5. THE INSERTION ALGORITHM

5.1 Observations and Methods

Before designing the proposed algorithm, we carefully analyze a few more issues of redundant wire insertion. In this paper, we model a net as a projection-graph and afterward use a possible redundant wire as a new edge of the graph. Then, the number of possible redundant wires for a net is equal to \( C(n_i, 2) \), where \( n_i \) is the number of vertices of a projection-graph. The number of possible choices of insertion patterns for a net is \( 2^n C(n_i, 2) \). Assume that a design has \( N \) nets, and then the number of all possible choices for the design is \( \prod_{i=1}^{N} 2^n C(n_i, 2) \). Since the number of possible choices could be large for today’s complex designs, it is necessary to have an effective method for determining which redundant wires should be inserted. The following are three problems observed, illustrated in Figure 6.

1. **Yield Rate Loss**: It is critical to know that the yield rate may be reduced after inserting a redundant wire. For example, the insertion of \( r1 \) will cause yield rate loss if \( \alpha S < 1 \), i.e., the per unit length short potential is higher than the per unit length open potential. To overcome this issue, we apply Equation (8) to estimate the real yield rate changes, and only positive gain redundant wires are committed.

2. **Insertion Pattern**: The opens of a partial net can be tolerated by different choices of redundant wires. Clearly, to reduce the total redundant wire length, only one of these choices has to be taken. For the example in Figure 6, the opens of the partial net consisting of \( w1, w2 \), and \( w3 \) can be tolerated by inserting either \( r2 \) or \( r3 \). Thus, if \( r2 \) is inserted, \( r3 \) does not need to be inserted since no openings for other parts of the net can be tolerated. Our method is to choose a set of redundant wire candidates with minimum wire length to cover opens for all nets. For this example, the minimum set is \{ \( r1, r2, r4, r5, r6 \} \}. Equivalently, by minimizing the redundant wire length of the set, these candidates have the maximum average tolerance-ratio.

3. **Insertion order**: Since the wiring result changes dynamically along with the insertion process, the actual yield rate gain for each insertion depends on the insertion order. For example, if \( r4 \) is inserted after \( r5 \), the parallel run length increases by \( p \) when inserting \( r4 \). Hence, the yield rate gain for \( r4 \) is reduced due to the increased parallel run length. For the same reason, if \( r5 \) is inserted after \( r4 \), the yield rate gain for \( r5 \) is reduced. To decide the insertion order, we follow the decreasing order of the tolerance-ratio, \( C_i/|r| \), the average yield rate change per unit length. According to Equation (9), the higher the tolerance-ratio, the better the estimated per unit insertion wire length yield rate improvement can be. Hence, for the example in Figure 6, we will insert \( r4 \) before \( r5 \). Later, if the actual yield rate gain of \( r5 \) is negative, then the insertion of \( r5 \) is skipped.

Based on these three observations, we present the proposed algorithm in the next subsection.

5.2 The Algorithm Flow

In this subsection, we propose a highly effective redundant wire insertion algorithm. We note that actual wiring through a router may find a redundant wire infeasible, and possible detours can actually spoil an expected yield improvement. Therefore, only those redundant wires that can actually improve yield rate are realized.

The proposed four-step algorithm flow is shown in Figure 7. **Step 1** is to construct a projection-graph for each net. Any redundant wire of a net is a redundant edge connecting two vertices of the corresponding projection-graph. Hence, the algorithm can support pin-pin and pin-wire redundant wire types.

To deal with the insertion pattern problem mentioned, **Step 2** is to generate a set of redundant wire candidates with the minimum wire length to tolerate opens for all nets. We apply a minimum-length 2-edge-connected graph algorithm on the projection-graph and generate a set of recommended redundant wire candidates. Heuristic algorithms [4] [16] [17] can be applied to build 2-edge connected graphs and in our implementation, we adopted a 1.5-approximation minimum length 2-edge connected graph algorithm [16]. In the next two steps, we further fine tune the initial solution.
Table 2: The statistics of four industrial test cases. (let $a = 10^{-7} \mu m$)

<table>
<thead>
<tr>
<th>Cases</th>
<th>#Net</th>
<th>L(mm)</th>
<th>P(mm)</th>
<th>$\alpha/s=3$</th>
<th>$\alpha/s=2$</th>
<th>$\alpha/s=1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>6,933</td>
<td>1,738</td>
<td>507</td>
<td>0.25</td>
<td>0.809</td>
<td>0.801</td>
</tr>
<tr>
<td>M1</td>
<td>28,421</td>
<td>1,279</td>
<td>591</td>
<td>0.42</td>
<td>0.852</td>
<td>0.843</td>
</tr>
<tr>
<td>M2</td>
<td>14,343</td>
<td>2,146</td>
<td>1,247</td>
<td>0.53</td>
<td>0.744</td>
<td>0.723</td>
</tr>
<tr>
<td>H1</td>
<td>14,107</td>
<td>2,118</td>
<td>1,425</td>
<td>0.64</td>
<td>0.741</td>
<td>0.717</td>
</tr>
</tbody>
</table>

Table 3: The defect rate improvement comparison. $\Delta L$ is the increase of total wire length. $\Delta P$ is the increase of total parallel run length. $\Delta Y$ is the yield rate improvement, and $d$ is the percentage of defect rate improvement, or $d = \Delta Y/(1 - Y)$. The unit of measurement is mm.

<table>
<thead>
<tr>
<th>Cases</th>
<th>$\Delta L$</th>
<th>C</th>
<th>$\Delta P$</th>
<th>$\Delta Y$</th>
<th>d</th>
<th>$\Delta L$</th>
<th>C</th>
<th>$\Delta P$</th>
<th>$\Delta Y$</th>
<th>d</th>
<th>$\Delta L$</th>
<th>C</th>
<th>$\Delta P$</th>
<th>$\Delta Y$</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1,476</td>
<td>3,077</td>
<td>1,112</td>
<td>0.123</td>
<td>65%</td>
<td>1,444</td>
<td>3,018</td>
<td>1,062</td>
<td>0.104</td>
<td>52%</td>
<td>1,129</td>
<td>2,411</td>
<td>694</td>
<td>0.048</td>
<td>26%</td>
</tr>
<tr>
<td>M1</td>
<td>346</td>
<td>878</td>
<td>345</td>
<td>0.042</td>
<td>28%</td>
<td>336</td>
<td>861</td>
<td>333</td>
<td>0.036</td>
<td>22%</td>
<td>255</td>
<td>727</td>
<td>246</td>
<td>0.022</td>
<td>12%</td>
</tr>
<tr>
<td>M2</td>
<td>374</td>
<td>1,123</td>
<td>450</td>
<td>0.060</td>
<td>23%</td>
<td>359</td>
<td>1,070</td>
<td>375</td>
<td>0.052</td>
<td>19%</td>
<td>213</td>
<td>726</td>
<td>222</td>
<td>0.026</td>
<td>9%</td>
</tr>
<tr>
<td>H1</td>
<td>266</td>
<td>652</td>
<td>411</td>
<td>0.025</td>
<td>10%</td>
<td>226</td>
<td>565</td>
<td>329</td>
<td>0.017</td>
<td>8%</td>
<td>31</td>
<td>116</td>
<td>39</td>
<td>0.005</td>
<td>1%</td>
</tr>
</tbody>
</table>

Figure 8: The statistics of redundant edges for test case S1 generated by Step-2 of the proposed algorithm.

There is no previous wire insertion method that can accurately consider the short problem, instead of comparing with previous research work, we focus on analyzing yield rate changes among designs with different routing densities and $\alpha/s$ ratios.

Figure 8 shows the statistics of redundant edges in percentage of the total redundant edge count. Since the majority are two- or three-pin nets, a large percentage of redundant edges have a tolerance ratio of 2.

Table 3 summarizes the experimental results of all four test cases with different $\alpha/s$ ratios. For test cases of $\alpha/s=3$, where the open issue is more severe, the proposed algorithm tends to focus on open rate reduction by maximizing the $C$ value, and adds as many redundant wires as possible. Thus, for test case S1, with low wiring density, many redundant wires can be added to put almost all wires on loops. Hence, defect rate improves by 65% and the yield rate becomes as high as 0.93. On the contrary, for chip H1 with high wiring density, our algorithm inserts redundant wires with small $\Delta L$ to yield a large $C$ value ($C/\Delta L = 652/266 = 2.5$).

For $\alpha/s=1$, when the short issue is more significant, our algorithm still performs well, even when the wiring densities are larger than 50% (M2, H1). For H1, our algorithm chooses candidates of small $\Delta P$ to get a large value of $C/\Delta P = 116/39 = 2.97$ to maximize the yield rate improvement. The experimental results are summarized in Table 3, which shows that the defect rate can be reduced by up to 65% and that the average defect rate improvement is 24%.

The runtime of the proposed algorithm depends mainly on the performance of the router. Table 4 shows that the total runtimes range from 5 min (H1 with $\alpha/s=1$) to 167 min (M1 with $\alpha/s=3$). If the runtime of Step 4 (routing) is not included, these runtimes
would all be below 6 seconds.

Figure 9 shows a part of the final wiring results of redundant wires in S1 with \( o/s = 1 \) and a routed net with three redundant wires. To demonstrate the yield rate loss problem, our algorithm is modified to ignore shorts. Table 5 shows all yield rate improvements. In our experiment results, the worst case defect rate deteriorates by 7%. In addition, comparing to Table 3, all defect rate improvements are reduced and the gap becomes larger when \( o/s \) ratio decreases. That is because the short problem becomes more serious.

7. CONCLUSIONS

To the best of our knowledge, this is the first redundant wire insertion algorithm that accurately considers short defects. Our contributions are to devise an effective equation to estimate yield rate changes and use the tolerance-ratio to design our algorithm. The algorithm can guarantee the yield rate improvement and greatly increase the chip yield rate.

8. REFERENCES


Table 5: The yield rate loss problem. Without considering shorts, yield rate improvement can be negative. Notations are the same as in Table 3.

<table>
<thead>
<tr>
<th>Cases</th>
<th>( \Delta L )</th>
<th>( C )</th>
<th>( \Delta P )</th>
<th>( o/s=3 )</th>
<th>( o/s=2 )</th>
<th>( o/s=1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1500</td>
<td>3106</td>
<td>1135</td>
<td>65%</td>
<td>52%</td>
<td>21%</td>
</tr>
<tr>
<td>M1</td>
<td>553</td>
<td>1173</td>
<td>636</td>
<td>28%</td>
<td>19%</td>
<td>-1%</td>
</tr>
<tr>
<td>M2</td>
<td>650</td>
<td>1518</td>
<td>904</td>
<td>22%</td>
<td>15%</td>
<td>-1%</td>
</tr>
<tr>
<td>H1</td>
<td>442</td>
<td>910</td>
<td>707</td>
<td>9%</td>
<td>4%</td>
<td>-7%</td>
</tr>
</tbody>
</table>

Figure 9: (a) The wiring results with redundant wires for test case S1 with \( o/s = 1 \). (b) The final routed result of a sampled net. The three wires highlighted are redundant wires.