Comparing Two Rewiring Models

March 15, 2004

Abstract

The concept of SPFDs (Sets of Pairs of Functions to be Distinguished) was introduced in context of FPGA synthesis [10]. Subsequently, SPFDs were used for rewiring networks of FPGAs [10, 11, 12]. It was experimentally shown [11, 12] that SPFD-based rewiring techniques could outperform ATPG-based rewiring techniques in terms of rewiring ability, i.e. it could find alternates wires for a larger number of wires in the network.

In this paper, we provide formal arguments to strengthen the claim that rewiring using SPFDs is indeed more powerful than rewiring based on ATPG methods.

1 Introduction

Rewiring seeks to replace one wire with a new set of previously non-existing wires without changing the functionality of the network. Rewiring has a number of interesting applications, such as replacing a wire on the critical path with another wire that is not on the critical path, or a wire in a heavily congested routing area with another in a less congested area. Minor gate functionality changes are allowed to accomodate for the modified wiring. An interesting feature of rewiring is that it minimally perturbs the original circuit and hence can be effective in the later stages of logic synthesis. Most previous work in this area used ATPG-based methods [1, 2, 3, 7]. More recently, a new formalism for expressing flexibility during logic synthesis, called SPFDs, was introduced in the context of FPGA synthesis. Subsequently, they were used for rewiring FPGA networks [10, 11]. SPFDs were also used for rewiring technology-independent Boolean networks [17] with promising results. In the sequel, we refer to rewiring using SPFDs and rewiring using ATPG-based methods as SPFD-rewiring and ATPG-rewiring, respectively.

In this paper, the capabilities of SPFD-rewiring and the opportunities for using it are explored further. It has been postulated that SPFD-rewiring is more powerful than ATPG-rewiring. We present some formal results to support this claim. We also present some initial ideas for using SPFD-rewiring and re-mapping in conjuction.

In Section 2, some previous work in rewiring, particularly the ATPG-based techniques, are described. Section 3 introduces the basic notation that will be used in the rest of the paper. The connection between SPFDs and rewiring is discussed in Section 4. Section 5 provides formal arguments to establish that SPFD-rewiring is more powerful than the ATPG-based techniques. The possibility of using SPFD-based rewiring and re-mapping at the same time for standard-cell networks is discussed in Section 6. The paper concludes with some directions for future work in Section 7.

2 Previous Work

As mentioned before, most previous work in rewiring used ATPG-based methods [1, 2, 3, 4, 5, 7]. The common idea is the notion of *redundancy addition and removal* (RAR): the ability of adding a redundant wire and in the process, making some of the other wires redundant, which can then be removed. Several efficient heuristics have been proposed to quickly identify redundant wires that can make the target wires redundant. More recently, the requirement that the added wire had to be redundant was dropped i.e. the new wire w_a need not be redundant but if the original wire w_t is removed and the new wire w_a is added, an equivalent circuit is obtained. The work in [6, 13] explores this concept. In [13], the authors used the concept of error correction to find a replacement for a particular wire. The problem with this approach was it required the use of a formal verification tool for guaranteeing the correctness of the modified circuit. The work in [6] attempted to eliminate the use of formal verification by finding some necessary and sufficient conditions under which a new wire could replace an original wire without affecting network functionality. However, no experimental results, comparing the advantage of their technique over RAR methods, were presented.

Some related work used the concept of global flow analysis [8, 9]. This technique modeled the rewiring problem using a flow graph and then solved it using the maxflow-mincut algorithm on the corresponding flow graph. The advantage of this approach over ATPG-based methods was that it could simultaneously add and remove many redundant wires at the same time. However, these methods are similar to ATPG-based methods in that they try to make the wires redundant by making them untestable. Hence they allow very limited functionality changes of the nodes in the network.

More recently, SPFDs were used for rewiring, for delay improvement of FPGA networks [10, 11, 12]. The authors of [11, 12] also presented some experimental evidence to show that SPFD-rewiring can indeed outperform ATPG-rewiring. The basic idea of SPFD-rewiring will be explained in detail in Section 4.2.

3 Notation

A combinational network $\mathcal{N}$ can be denoted as a tuple $< I_{\mathcal{N}}, O_{\mathcal{N}}, \mathcal{V}_{\mathcal{N}}, \mathcal{W}_{\mathcal{N}} >$, where $I_{\mathcal{N}}$, $O_{\mathcal{N}}$, $\mathcal{V}_{\mathcal{N}}$ and $\mathcal{W}_{\mathcal{N}}$ denote the primary inputs, primary outputs, nodes and directed edges between the nodes, respectively. Let η_j denote a node in $\mathcal{N}$. The global function of η_j is denoted as g_j. The fanin and fanout nodes of η_j are collectively denoted as $FI(\eta_j)$ and $FO(\eta_j)$, respectively. The nodes in the transitive fanin of η_j are collectively denoted as $TFI(\eta_j)$. The nodes and the primary output nodes in the transitive fanout of η_j are collectively denoted as $TFO(\eta_j)$ and $PO(\eta_j)$, respectively. Each primary input η_i of $\mathcal{N}$ is associated with two variables, x_i and x'_i. The variables associated with the primary inputs of $\mathcal{N}$ are collectively denoted as X or X', depending on whether the unprimed or primed variables are used. Both X and X' are referred to as the primary space of $\mathcal{N}$. A directed connection from η_i to η_j is called a wire and is denoted as $w_{\eta_i \to \eta_j}$. Given a wire $w_{\eta_i \to \eta_j}$, η_i and η_j are called the source and destination of $w_{\eta_i \to \eta_j}$, respectively.

4 SPFDs

In this section, we briefly review SPFDs and their ability to represent the information content of a node (see [17] for a detailed overview).

An SPFD, $R = \{(g_{1a}, g_{1b}), (g_{2a}, g_{2b}), \cdots, (g_{na}, g_{nb})\}$, denotes a set of pairs of functions that have to be distinguished i.e. for each pair $(g_{ia}, g_{ib}) \in R$, the minterms in g_{ia} have to produce a different value from the minterms in g_{ib}.

Example 1 $\{(ab, \overline{a}b), (\overline{a}\overline{b}, a\overline{b})\}$ *is an example of an SPFD.*

The functions contained in an SPFD are all the functions that can satisfy the SPFD. A function f is said to satisfy an SPFD, $R = \{(g_{1a}, g_{1b}), (g_{2a}, g_{2b}), \cdots, (g_{na}, g_{nb})\}$, if for each pair $(g_{ia}, g_{ib}) \in R$, $f(g_{ia}) \neq f(g_{ib})$.

Example 2 *The function $f_1 = a$ satisfies the SPFD $\{(ab, \overline{a}b), (\overline{a}\overline{b}, a\overline{b})\}$, since it distinguishes each pair in the set. However, the function $f_2 = b$ cannot distinguish the second pair in the SPFD, and hence does not satisfy the SPFD.*

An SPFD , $R = \{(g_{1a}, g_{1b}), \ldots, (g_{na}, g_{nb})\}$, can also be represented as a graph, $G = (V, E)$, where

$$\begin{aligned} V &= \{m_k | m_k \in g_{ij}, 1 \leq i \leq n, j = \{a, b\}\} \\ E &= \{(m_i, m_j) | ((m_i \in g_{pa}) \wedge (m_j \in g_{pb})) \vee \\ &\quad ((m_i \in g_{pb}) \wedge (m_j \in g_{pa})), 1 \leq p \leq n\} \end{aligned}$$

Every $e \in E$ is referred to as an SPFD edge. It is easy to see that any valid coloring of the SPFD graph is a function that satisfies the SPFD. This function can be multi-valued, in general. In the sequel, we will only consider binary functions.

An SPFD is best thought of as a graph that encapsulates information. For combinational logic, information is the ability to distinguish one primary input minterm from another. An SPFD attached to a node specifies which pairs of primary input minterms can be or have to be distinguished by the node. This can be thought of as the information content of the node, since it denotes what information the node passes to its fanouts and hence to its surrounding network. In a later section, we will discuss how this property can be exploited for rewiring.

Next, we introduce a few more concepts related to SPFDs. Later, we prove that SPFD-rewiring can emulate the core techniques of most ATPG-based techniques.

4.1 Theory

Definition 1 *Given $\mathcal{N}$, the* **global SPFD of node** η_j *specifies that the minterms in $g_j(X)$ have to be distinguished from the minterms in $\overline{g_j(X)}$.*

For a network with binary nodes, the global SPFDs of the nodes are bipartite.

Definition 2 *The* **separator of a node** η_j*, denoted as $\mathcal{Y}_j$, is the set of nodes in $\mathcal{N}$ that do not belong to $TFO(\eta_j)$ but have a direct fanout to a node in $TFO(\eta_j)$. It is given as*

$$\mathcal{Y}_j = \{\eta_s | \eta_s \in FI(\eta_p), \text{ where } \eta_p \in TFO(\eta_j) \text{ and } \eta_s \notin TFO(\eta_j)\}$$

Figure 1 illustrates the separator $\mathcal{Y}_j$ of η_j. $\mathcal{Y}_j$ contains the node η_j and includes all the nodes in $\mathcal{N}$ that directly fanout to a node in $TFO(\eta_j)$. Hence, it forms a complete cut for the primary outputs in $PO(\eta_j)$ i.e. if the nodes in $\mathcal{Y}_j$ are removed, all the nodes in $PO(\eta_j)$ are disconnected from the primary inputs. Thus, $\mathcal{Y}_j$ denotes the set of nodes that provide all the required information to the nodes in $PO(\eta_j)$.

Definition 3 *The* **separator of a node** η_j **wrt** η_o*, denoted as $\mathcal{Y}_j^o$, is the set of nodes in $\mathcal{Y}_j$ that belong to $TFI(\eta_o)$.*

Thus, $\mathcal{Y}_j^o$ denotes all the nodes in $\mathcal{Y}_j$ that have a path to η_o.

Definition 4 *Given $\mathcal{N}$ and two nodes η_j and η_o such that $\eta_o \in TFO(\eta_j)$, the* **minimum SPFD of a node** η_j **wrt** η_o *is the set of edges in the global SPFD of η_o that is not contained in the global SPFDs of the other nodes in $\mathcal{Y}_j^o$.*

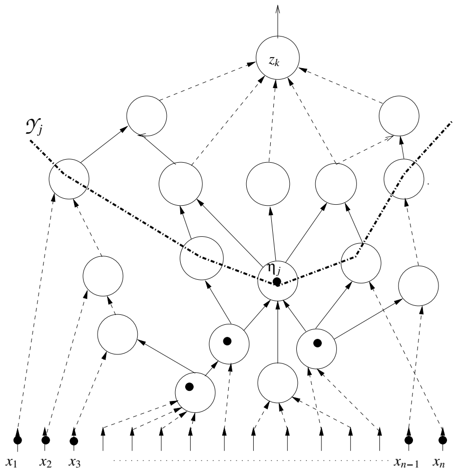


Figure 1: The set of nodes marked by dots denotes the separator $\mathcal{Y}_j$: The solid edges indicate direct fanouts.

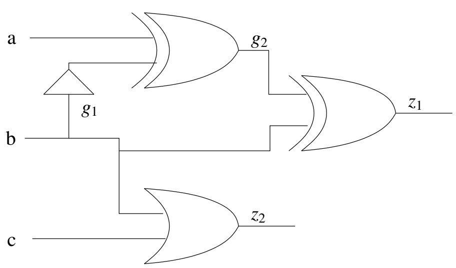


Figure 2: Rewiring example.

Thus the minimum SPFD of a node η_j wrt to another node η_o denotes the unique information that η_j provides to η_o. The **minimum SPFD of a node** η_j is simply the union of the minimum SPFDs of η_j wrt the nodes in $PO(\eta_j)$. Thus, it denotes the unique information provided by a node to the primary outputs of a network. Once η_j is optimized using its minimum SPFD, the functionalties of the nodes in its transitive fanout may have to be altered to account for the modified flow of information. However, the functionalities of the remaining nodes remain unaltered.

Assuming that the network $\mathcal{N}$ only has a single primary output η_o, the minimum SPFD of a node η_j can easily computed as follows[1]:

1. For each node η_i in $\mathcal{Y}_j$, compute its global SPFD R_i.
2. The minimum SPFD of η_j, denoted as R_j^{min}, is

$$R_j^{min} = R_o \wedge \overline{\cup_{\eta_i \in \mathcal{Y}_j; \eta_i \neq \eta_j}(R_i)},$$

where R_o is the global SPFD of the primary output η_o.

[1]The given algorithm can be very easily generalized for a network with multiple primary outputs

Example 3 *Consider the circuit shown in Figure 2. The separator of node g_1 is $\{a, g_1, b\}$. Node g_1 has a single primary output z_1 in its transitive fanout. Hence, its minimum SPFD is equal to its minimum SPFD wrt z_1. The global SPFD of z_1 is equal to $\{1-, 0-\}$ (the minterms are in the form ab). The global SPFD of a and b are $\{1-, 0-\}$ and $\{-1, -0\}$, respectively. Note that all the edges of the global SPFD of z_1 are contained in the global SPFD of a. Hence, the minimum SPFD of g_1 is empty.*

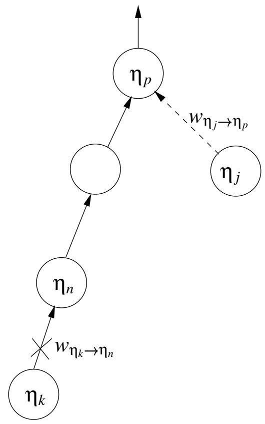
If the minimum SPFD of a node is empty, then it doesn't provide any unique information to the primary outputs. Hence, it can be removed. However, it should be noted that the node may not be s-a-0 or s-a-1 redundant and cannot be replaced with a 0 or 1 without modifying the functionalities of the nodes in its transitive fanout. For instance, the minimum SPFD of g_1 is empty (as derived in the above example) but it is neither s-a-0 or s-a-1 redundant. This is because the nodes in the transitive fanout of g_1 depend on the information flowing in from g_1.If the functionalities of g_2 and z_1 are suitably altered, then g_1 can be set to a 0 or 1. Alternatively, if the minimum SPFD of a node is not empty, then the node can't be set to 0 or 1 by simply altering the functionalities of the nodes in its transtive fanout. It will be necessary to alter the topology of the network (i.e. add new wires to provide additional channels of information) to maintain correct functionality.

Definition 5 *Given $\mathcal{N}$ and a wire $w_{\eta_k \to \eta_j}$, let $\mathcal{N}'$ denote the network derived from $\mathcal{N}$ by adding a buffer η_b on the wire $w_{\eta_k \to \eta_j}$. Thus, $\mathcal{N}' = < I_{\mathcal{N}}, O_{\mathcal{N}}, \mathcal{V}_{\mathcal{N}} \cup \{\eta_b\}, \mathcal{W}_{\mathcal{N}} \setminus \{w_{\eta_k \to \eta_j}\} \cup \{w_{\eta_k \to \eta_b}, w_{\eta_b \to \eta_j}\} >$. Given $\eta_o \in TFO(\eta_j)$, the* **minimum SPFD of the wire** *$w_{\eta_k \to \eta_j}$ wrt η_o in $\mathcal{N}$ is equal to the minimum SPFD of η_b wrt η_o in $\mathcal{N}'$.*

It can be proved that the minimum SPFD of wire $w_{\eta_k \to \eta_j}$ wrt η_o is the set of edges, in the minimum SPFD of η_j wrt η_o, that can only be distinguished by η_k. The minimum SPFD of a wire can similarly be generalized from the minimum SPFD of a node.

It is often useful to define the notion of the minimum SPFD of η_j relative to a special node η_p in the network, where η_p is such that all paths from η_j to the primary outputs have to pass through η_p. In previous ATPG literature, η_p is often referred to as the (absolute) dominator of η_j [14]. Note that this is slightly different from the classical definition of a dominator [15, 16], where a dominator is defined for a pair of vertices. It is also possible to define the notion of the dominator of a wire using the concept of the dominator of a node.

Definition 6 *The dominator of a wire $w_t = w_{\eta_k \to \eta_j}$ in $\mathcal{N}$ is equal to the dominator of the node η_b in $\mathcal{N}' = < I_{\mathcal{N}}, O_{\mathcal{N}}, \mathcal{V}_{\mathcal{N}} \cup \{\eta_b\}, \mathcal{W}_{\mathcal{N}} \setminus \{w_{\eta_k \to \eta_j}\} \cup \{w_{\eta_k \to \eta_b}, w_{\eta_b \to \eta_j}\} >$.*



η_p is a dominator of η_n

Figure 3: General Rewiring Model

It should be pointed out that the destination of a wire is also its dominator. Thus, η_j is a dominator of the wire $w_{\eta_k \to \eta_j}$.

The concept of the minimum SPFD of the node/wire wrt to a dominator node is useful since any simplification of η_j using this minimum information will restrict the changes in $\mathcal{N}$ to the nodes between the transitive fanout cone of η_j and the transitive fanin cone of η_p. Note that these SPFDs can be very easily obtained using the definitions given earlier.

4.2 Rewiring using SPFDs

Due to its ability to represent the information content of a node/wire, SPFDs provide a powerful tool for rewiring. Here, we describe the general model of SPFD-rewiring and argue that rewiring using this model gives us back an equivalent network.

The most general model of SPFD-rewiring looks as shown in Figure 3. The node η_p is a dominator of the wire $w_t = w_{\eta_k \to \eta_n}$. If the minimum SPFD of w_t wrt to η_p is a subset of the SPFD of a node η_j[2], then w_t can be replaced by a wire $w_a = w_{\eta_j \to \eta_p}$. Of course, if the replacement is made, the functions of all the nodes that lie between η_n and η_p (i.e η_p, η_n and nodes that belong to $\{TFI(\eta_p) \cap TFO(\eta_n)\}$) have to be altered to account for the different flow of information. This replacement can be done by modifying the SPFDs of these nodes to reflect the different flow of information. It can be proved that after replacing w_t with w_a and suitably modifying the functions of the nodes in between, the node η_p can still distinguish all the edges that belonged to its global SPFD before rewiring. Thus, the global functionality of η_p remains unchanged (modulo don't cares) after rewiring. Since η_p is a dominator of w_t and the global function of η_p remains unchanged as a result of the rewiring, all the effects of replacing w_t with w_a are suppressed beyond η_p. Hence, the functionality of the network remains unchanged as a result of this wire replacement. It should be noted that η_n is also a dominator of the wire $w_{\eta_k \to \eta_n}$. Hence it is also possible to replace $w_{\eta_k \to \eta_n}$ with a new wire from η_m to η_n. In this case, we will only need to modify the functionality at η_n to account for the wire replacement.

[2]We can either look at the global SPFD of η_j or some pre-determined SPFD of η_j, depending on whether η_j remains the same or is independently optimized.

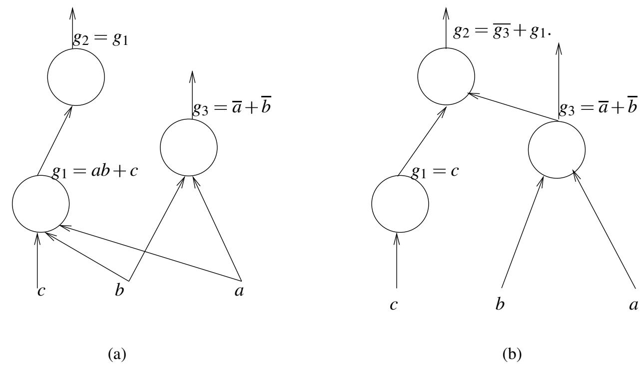


Figure 4: Example of General Rewiring Model

Example 4 *Consider the example circuit shown in Figure 4(a). Suppose we want to replace $w_t = w_{a \to g_1}$ with a new wire to g_2. The minimum SPFD of w_t wrt to g_2 is $\{110, 010\}$. Node $g_3 = \overline{a} + \overline{b}$ can distinguish this minimum SPFD since $g_3(110) = 0$ and $g_3(010) = 1$. So, we can replace w_t with a new wire $w_a = w_{g_3 \to g_1}$. The functionalities of the nodes g_1 and g_2 have to be modified to reflect this new flow of information. Figure 4(b) shows the new functionalities of g_1 and g_2 after replacing w_a with w_t.*

It is also possible to use the above model of SPFD rewiring to replace w_a with w_t. For this situation, we check if the minimum SPFD of w_a wrt to dominator η_p is contained in some node η_j. The main thing that will change is the set of nodes whose functionalities have to be modified. To minimize the amount of functionality changes, it will be useful to restrict η_j to be a dominated node of η_p. In that case, it will suffice to modify the functionalities of the nodes in between η_j and η_p.

If the minimum SPFD of $w_{\eta_k \to \eta_n}$ wrt η_p is empty, it does not provide any unique information to node η_p and hence can be removed. However, the functionalities of all the nodes between η_n and η_p (including η_n and η_p) may have to be changed to account for the different flow of information.

Multiple-wire replacement can also be done using SPFD-rewiring. In that case, we use a virtual node, added between a suitable set S of nodes to which we want to add the replacement wires and their fanouts, as a dominator.

5 Comparing SPFD-Rewiring to ATPG-rewiring

A systematic exploration of the link between SPFD-rewiring and other rewiring techniques would be of interest. It has been postulated that SPFDs provide more rewiring opportunities than the ATPG-based rewiring approaches and the global flow techniques proposed by Berman et. al. In the rest of the section, we explore the connection between SPFD-rewiring and the following ATPG-rewiring techniques:

- Redundant Wire Addition - Add a redundant wire w_a that makes an exisiting wire w_t redundant. This is the technique used in RAR.
- Simultaneously Redundant Wires - Simultaneously add an irredundant wire w_a and remove another wire w_t to get an equivalent network.

Then, we provide a simple example where SPFD-rewiring can find a candidate but the other two techniques fail.

Definition 7 *Given $\mathcal{N}$, a wire $w_{\eta_k \to \eta_j}$ is a* **replacement** *of $w_{\eta_m \to \eta_n}$ if the new network $\mathcal{N}'$ formed by removing $w_{\eta_k \to \eta_j}$ and adding $w_{\eta_m \to \eta_n}$ is functionally equivalent to $\mathcal{N}$.*

Removing a wire $w_{\eta_k \to \eta_j}$ is equivalent to setting the input of η_j connected to η_k to a constant. The value should be chosen so that the other inputs don't get eliminated. For instance, if the original function is $f = ab + c$, then a should be set to 1. Setting it to 0 would effectively remove b as well.

In the remainder of this section, we will develop the theory to show that SPFD-rewiring can emulate redundant wire addition and simultaneous wire replacement.

Lemma 1 *The global SPFD of a node η_p is a subset of the union of the global SPFDs of its fanins.*

Proof Proof by contradiction: Assume there exists a minterm pair $e = (m_1, m_2)$ in the global SPFD on η_p that does not exist in the global SPFDs of the nodes in $FI(\eta_p)$. Since m_1 and m_2 are not distinguished by any fanin of η_p, hence m_1 and m_2 must map to the same minterm m in the local fanin space of η_p. Since, by assumption, m_1 is distinguished from m_2 in the global SPFD of η_p, it would mean that the function at η_p has to non-deterministically map m to both 0 and 1, which is not possible for a deterministic network. □

The above lemma implies that if some information is not available at the inputs of a node, it cannot be available at the output of a node. Thus, information available at the output of a node is limited by the information available at its inputs.

Lemma 2 *Given $\mathcal{N}$ and the wire $w_t = w_{\eta_k \to \eta_j}$, let $\mathcal{N}'$ denote the network derived from $\mathcal{N}$ by removing w_t. The node in $\mathcal{N}'$ corresponding to the primary output $\eta_o \in PO(\eta_j)$ in $\mathcal{N}$ does not distinguish the edges in the minimum SPFD of w_t wrt η_o.*

Proof Let η_b denote buffer added to the wire $w_{\eta_k \to \eta_j}$ in $\mathcal{N}$. By definition, the minimum SPFD of w_t wrt η_o is equal to the minimum SPFD of η_b wrt η_o. Note that in $\mathcal{N}$, $TFO(\eta_b) = TFO(\eta_j) \cup \{\eta_j\}$ and $\eta_o \in PO(\eta_b)$. Let e denote a minterm pair in the minimum SPFD of w_t wrt η_o. By definition of the minimum SPFD, e is in the global SPFD of η_o and e is not distinguished by any node in $\mathcal{Y} = \mathcal{Y}_b^o \setminus \{\eta_b\}$ in $\mathcal{N}$. Let $\mathcal{Y}'$ denote the nodes corresponding to $\mathcal{Y}$ in $\mathcal{N}'$. Since the global functions of the nodes in $\mathcal{Y}'$ are identical to their counterparts in $\mathcal{Y}$, e is not distinguished by the nodes in $\mathcal{Y}'$. Given $T = TFO(\eta_b) \cap TFI(\eta_o)$, let T' denote the corresponding nodes in $\mathcal{N}'$. By definition of a separator, a node in T can only have nodes in $\mathcal{Y}_b^o$ or nodes in T as their immediate fanins. Correspondingly,a node in T' can only have either the nodes in $\mathcal{Y}'$ or the nodes in T' as its immediate fanin. Hence, it is possible to levelize the nodes in T' by starting at the nodes in $\mathcal{Y}'$. We prove by induction on the level of the nodes in T' that e will not be distinguished by the primary output in $\mathcal{N}'$ corresponding to η_o in $\mathcal{N}$.
Base Case - Level 0 : Level 0 nodes belong to $\mathcal{Y}'$ and hence cannot distinguish e, by definition.
Inductive Case - Given it is true for all nodes of level $\leq i$, prove that it is true for level $i+1$. Consider a level $(i+1)$ node η_p. It will only have fanins with levels $\leq i$. By assumption, e is not distinguished by any node with levels $[0, \cdots, i]$. Hence, by Lemma 1, e cannot be distinguished by η_p.
Thus, e will not be distinguished by the node in $\mathcal{N}'$ corresponding to η_o in $\mathcal{N}$. □

Lemma 2 quantifies the information that is lost at a node as a result of a wire being removed from one of the nodes in its transitive fanin.

Theorem 1 *If a wire $w_a = w_{\eta_j \to \eta_p}$ is a replacement of wire $w_t = w_{\eta_k \to \eta_n}$ (Figure 3), where η_p is a dominator of w_t, then the minimum SPFD of w_t wrt η_p is contained in the global SPFD of η_j. The minimum SPFD of w_t wrt η_p is not empty in the original network $\mathcal{N}$.*

Proof Let $\mathcal{N}'$ denote the network derived from $\mathcal{N}$ by removing w_t and adding w_a. Let's consider the following three networks: $\mathcal{N}_1$ is the subnetwork of $\mathcal{N}$ consisting of η_p and all its transitive fanins, $\mathcal{N}_2$ is derived from $\mathcal{N}_1$ by removing w_t and $\mathcal{N}_3$ is derived from $\mathcal{N}_2$ by adding w_a. Note that $\mathcal{N}_3$ is the sub-network in $\mathcal{N}'$ that corresponds to the sub-network $\mathcal{N}_1$ in $\mathcal{N}$. Let e denote a minterm pair in the minimum SPFD of w_t wrt to η_p (at least one such e exists as the minimum SPFD of w_t wrt to η_p is non-empty in $\mathcal{N}$). By definition of minimum SPFD, e has to be present in the global SPFD of η_p in $\mathcal{N}$. By Lemma 2, e is not distinguished by the primary output in $\mathcal{N}_2$ corresponding to η_p in $\mathcal{N}_1$. However, the functionality of η_p is $\mathcal{N}'$ is equal to the functionality of η_p in $\mathcal{N}$ (modulo don't cares), as the functionalities of the nodes in $\mathcal{N}'$ are not modified beyond η_p (by definition of wire replacement) and the networks $\mathcal{N}$ and $\mathcal{N}'$ are equivalent. Hence e must be present in the global SPFD of η_p in $\mathcal{N}_3$. The only difference between $\mathcal{N}_2$ and $\mathcal{N}_3$ is that η_p in $\mathcal{N}_3$ has a new fanin η_j due to the addition of w_a. Hence e has to be present in the global SPFD of η_j (by Lemma 1). □

Both redundant wire addition and simultaneous wire replacement find replacements of exisiting wires in the network. By the above theorem, these replacements can also be found by the SPFD-rewiring model. According to redundant wire addition and simultaneous wire replacement, if w_a is an alternate of w_t, then the reverse is also true. This is also true in the SPFD model. It is easy to see that the minimum SPFD of w_a wrt to η_p has to be contained in the global SPFD of η_k. It is also possible to emulate multiple-wire replacement according to the above theorem. As mentioned before, a virtual node will be used as a dominator. Thus, we see that SPFD-rewiring can emulate redundant wire addition and simultaneous wire replacement, the core techniques of ATPG-rewiring.

Next we illustrate an example where SPFD-based rewiring can find an alternate but the ATPG-based techniques fail.

Example 5 *The minimum SPFD of wire* (g_2, z_1) *in the circuit in Figure 2 is* $A = \{(00, 10), (11, 01)\}$ *(the minterms in A are of the form ab). Primary input "a" can distinguish both pairs in A. Hence a fanout from "a" is a candidate wire for replacing* (g_2, z_1)*. Simplifying* z_1 *gives* $z_1 = a$*. In contrast, RAR cannot simplify the circuit. This is because there are no mandatory assignments for propagating a stuck-at-fault on* g_2 *(which is the main technique for discovering redundant wires), due to the presence of an XOR gate along the path, which is sensitive to both the inputs. The two wires are also not simultaneously replacable since we get an incorrect network if we simply remove* (g_2, z_1) *and add* (a, z_1)*.*

To re-iterate, SPFD-rewiring covers all situations where ATPG methods work and can also find cases when the ATPG-based techniques fail due to the following reasons:

- SPFD-rewiring relies on the concept of information flow rather than redundancy, which is related to *don't cares* and hence is less powerful than the notion of information flow.

- Unlike ATPG-rewiring, there is no restriction to keep the functionalities of all the nodes between η_k and η_p unchanged. This is both good and bad, but makes SPFDs more powerful. We will elaborate on this point a little more in the next section.

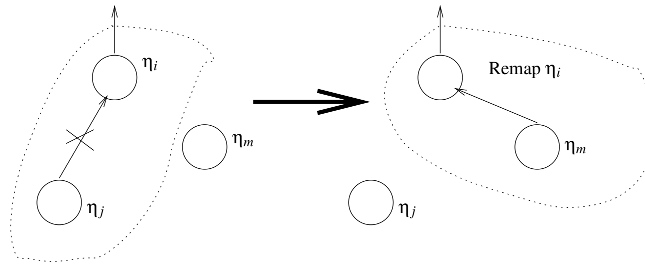


Figure 5: Restricted model of SPFD-rewiring combined with re-mapping.

6 SPFD-Rewiring and Remapping for Standard Cell Networks

As mentioned before, it was experimentally shown that SPFD-rewiring can find more wires with alternates than ATPG-rewiring for FPGA networks [11]. In a related work [12], the same authors show that the enhanced rewiring ability of SPFDs results in better networks (in terms of area and delay) after optimization, when compared with ATPG-rewiring.

Based on the conclusions of the previous section, we can conclude the SPFD-rewiring will outperform ATPG-rewiring in rewiring ability, even for standard-cell networks. However, this additional freedom provided by SPFD-rewiring may not gurantee better optimization opportunities. The main reason for this discrepancy is the following: After rewiring, the functionalities of nodes in between the destination of the original/new wire and the dominator of the original wire are forced to change to account for the modified flow of information. In FPGA networks, it is always possible to implement the new function at each node by simply modifying the functionality of the FPGA implementing it. This is not necessarily true for standard-cell networks. The new function at a node (whose original function was being implemented with a single gate from a given library) may now require a combination of gates to implement. This is not desirable as in effect we could end up increasing the area and/or delay after performing some wire replacements. So, in the worst case, we may be left with a situation where only the alternate wires returned by ATPG-rewiring can be used for optimization. However, there is also a strong possibility that the changes in the nodes of the network allows us to explore some alternate mappings for a small set of nodes, thereby enabling us to search the combined mapping and rewiring space.

Example 6 *The circuit in Figure 2 gives a simple example where this combined action provides an advantage. The wire* (g_2, z_1) *can be replaced by* (a, z_1)*, only if the function at* z_1 *is re-mapped to* $z_1 = a$*.*

We have done some initial experiments combining a restricted model of SPFD-rewiring (where we replace a wire $w_{\eta_j \rightarrow \eta_i}$ with a new wire from η_m to η_i i.e the dominator of $w_{\eta_j \rightarrow \eta_i}$ is η_i) combined with incremental re-mapping at η_i. The basic idea is shown in Figure 5. This restricted SPFD-rewiring model does not fully cover ATPG-rewiring but it can be shown to explore a different space from ATPG-rewiring. These initial experiments indicate that combining rewiring with mapping does indeed provide additional optimization opportunities. Furthermore, SPFD-rewiring seems to be an appropriate tool for harnessing this opportunity, as it allows one to maintain the original connectivity of a set of nodes while exploring new mappings for them.

We are now investigating opportunities for conducting similar experiments for the more generalized SPFD-rewiring problem.

7 Conclusions and Future Work

In [11, 12], the authors had presented experimental evidence to show that SPFD-based rewiring can outperform ATPG-rewiring for FPGA networks, in terms of delay and area. We presented formal arguments to prove that the core techniques in ATPG-based methods for rewiring can be emulated using SPFD-rewiring. These strengthen the informal claim that SPFD-rewiring is more powerful than ATPG-rewiring, in terms of rewiring ability.

In the future, we plan to conduct experiments for determining how the added flexibility of SPFD-rewiring over ATPG-rewiring translates into more optimization opportunities for standard-cell networks.

References

[1] K-T. Cheng and L. Entrena. Multi-level logic optimization by redundancy addition and removal. In *Proceedings of European Conference Design Automation*, Feb 1993.

[2] S.C. Chang, M. Marek-Sadowska and K.T. Cheng. Perturb and Simplify: Multi-level Boolean Network Optimizer. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15 (no. 12),Nov 1996.

[3] S. Chang and K-T. Cheng. Postlayout Logic Restructuring Using Alternative Wires. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16 (no. 6), Jun 1997.

[4] Y.M. Jiang, A. Kristic, K.T. Cheng and M. Marek-Sadowska. Post layout logic restructing for performance optimization. In *Proceedings of Design Automation Conference*, June 1997.

[5] S.-C. Chang, L.P.P.P. van Ginneken and M. Marek-Sadowska. Circuit Optimization By Rewiring. In *IEEE Transactions on Computers*, vol 28, Sept 1999.

[6] C-W. Chang and M. Marek-Sadowska. XWire : Efficient Single Wire Addition and Removal Beyond Redundancy. In *International Workshop on Logic Synthesis*, Jun 2001.

[7] C-W. Chang, M-F. Hsiao, and M. Marek-Sadowska. A New Reasoning Scheme for Efficient Redundancy Addition and Removal. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, vol .22 (no. 7), Jun 2001.

[8] C. Berman and L. Trevillyan. Global Flow Optimization in Automatic Logic Design. In *IEEE Transactions on Computers*, vol. 10 (no. 5), May 1991.

[9] S. Chang, Z. Wu and H. Yu. Wire Reconnections Based on Implication Flow Graph. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, Nov 2000.

[10] S. Yamashita, H. Sawada, and A. Nagoya. A new method to express functional permissibilities for LUT based FPGAs and its applications. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, Nov 1996.

[11] J. Cong, Y. Lin and W. Long. SPFD-Based Global Rewiring. *Proceedings of the ACM International Symposium on Field-Programmable Gate Arrays*, Feb 2002.

[12] J. Cong and W. Long. Global SPFD-Based Rewiring. *UCLA Computer Science Department Technique Report No. 200026*, Nov 2000.

[13] A. Veneris, M.S. Abadir and I. Ting. Design Rewiring Based on Diagnosis Techniques. In *Proceedings of ASP-DAC*, Jan 2001.

[14] T. Kirkland and M.R. Mercer. A Topological Search Algorithm for ATPG. In *Proceedings of Design Automation Conference*, Jun 1987.

[15] R. Tarjan. Finding Dominators in Directed Graphs. In *SIAM Journal of Computing*, vol. 3, 1974.

[16] T. Lengauer and R.E. Tarjan. A Fast Algorithm for Finding Dominators in a Flowgraph. In *ACM Transactions on Programming Languages amd Systems*, vol. 1(no. 1), 1979.

[17] S. Sinha and R.K. Brayton. Implementation and Use of SPFDs in Optimizing Boolean Networks. In *International Conference on Computer-Aided Design*, Nov 1998.