Rewiring Using IRredundancy Removal and Addition*

Chun-Chi Lin
Dept. of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
Email: idk.ccling@gmail.com

Chun-Yao Wang
Dept. of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
Email: wcyao@cs.nthu.edu.tw

Abstract—Redundancy Addition and Removal (RAR) is a restructuring technique used in the synthesis and optimization of logic designs. It can remove an existing target wire and add an alternative wire in the circuit such that the functionality of the circuit is intact. However, not every irredundant target wire can be successfully removed due to some limitations. Thus, this paper proposes a new restructuring technique, IRredundancy Removal and Addition (IRRA), which successfully removes any desired target wire by constructing a rectification network which exactly corrects the error caused by removing the target wire.

I. INTRODUCTION

Redundancy Addition and Removal (RAR) is a technique for reconstructing a circuit by adding some redundant wires or gates, named alternative wire/gates, and results in the removal of given target wires. In the process of RAR, the addition and the removal of redundant wires will restructure a circuit without changing the functionality. This circuit transformation technique is applicable to achieving optimization objectives such as area, timing, power, or reliability [2] [3] [4] [6] [7] of VLSI circuits.

One of the most commonly used approaches to RAR is Automatic Test Pattern Generation (ATPG)-based algorithm. ATPG-based approaches can be divided into two-stage algorithms and one-stage algorithms. In two-stage algorithms [2] [3] [4] [6] [7], a set of candidate wires that make the target wire become redundant is built up by finding the Mandatory Assignments (MAs). Then, redundancy tests which require much effort on each candidate wire are performed. On the other hand, one-stage algorithms [1] [5] identify alternative wires without redundancy tests which can significantly reduce the CPU time. However, the rewiring capability is not as good as that of two-stage algorithms.

A target wire has an alternative wire if the target wire become redundant after adding a redundant wire to the circuit. The capability of ATPG-based RAR for finding alternative wire is limited by the identified MAs. If a target wire does not have an alternative wire since the condition of MA is unsatisfied, the target wire cannot be removed [3] [6].

In this work, the RAR technique is viewed in an opposite way where it removes an irredundant target wire first and then adds an irredundant wire to rectify the functionality of the circuit. From this point of view, a technique of IRredundancy Removal and Addition (IRRA) for circuit restructuring is proposed. IRRA is an ATPG-based technique that can remove any desired irredundant target wire and rectify the error by adding some wires/gates. These added wires/gates are named rectification network. Thus, RAR is a special case of IRRA where the rectification network is just an alternative wire.

II. NOTATIONS AND BACKGROUND

A Boolean network is a Directed Acyclic Graph where each node $ni$ is associated with a Boolean variable $yi$ and a Boolean function $fi$. There exists a connection directed from node $ni$ to node $nj$ if the function $fj$ depends on the variable $yi$. An input to a gate has a controlling value if the value of the gate’s output is determined by the input regardless of the other inputs. The noncontrolling value is the inverse of the controlling value.

The dominators [8] of a wire $w$ is the set of gates $G$ such that all paths from $w$ to any primary outputs have to pass through all gates in $G$. Consider the dominators of $w$, the fault propagating inputs of a dominator are the inputs in the transitive fanout of $w$, and the other inputs are side inputs of the dominator. In the process of test generation for a stuck-at fault at a wire $w(gi \rightarrow gj)$, $gi$ is assigned to a controlling value to activate the fault effect and all side inputs of $w$’s dominators are assigned to noncontrolling values to propagate the fault effect.

The MAs are the unique value assignments required for a test to exist. The logic implication is a process of computing MAs for a test. The MAs for a stuck-at fault test on $w$ can be computed from setting the fault-activating value and setting noncontrolling values on the side inputs of $w$’s dominators. Then the MAs can be propagated forward or backward to obtain more MAs. Recursive learning [9] can be applied to find more MAs. Forced MA [4] is an MA that causes the target fault untestable while violating it.

III. IRREdundANCY REMOVAL AND ADDITION

For the process of IRRA, as seen in Fig. 1, it removes an irredundant target wire $wt$ first, and then it adds another irredundant wire $wr$ to rectify the functionality of the circuit.

---

*This work was supported in part by the National Science Council of R.O.C. under Grant NSC97-2220-E-007-042 and NSC97-2220-E-007-034.
The most important step is how to recognize the rectification network with respect to the injected error. To generate a test vector for the fault at \( wt \), the MAs for \( wt \) are calculated. Among the MAs, Source MAs are defined as follows which will be used in our approach.

**Definition 1:** Given a set of MAs for a target fault, the source MA (SMA) is defined as an MA whose transitive fanin cone contains no other MAs.

After calculating MAs and SMAs, a destination gate \( gd \) is selected on which the rectification network is added. The destination is selected from the dominators of \( wt \) since a dominator is where the error must pass through. In this work, the gates of AND, OR, and INV are only considered and the error model is "missing single wire". Thus, the target wire \( wd \) is defined as the network having minterms changed from 1 to 0 at \( gd \). The functionality of \( gd \) is as shown in Fig. 2(a).

![Fig. 1. An example of IRredundancy Removal and Addition.](image)

**Theorem 1:** Given a Boolean network, a target wire \( wt \) and a destination gate \( gd \). Suppose the cofactors of \( gd \) with respect to SMA in good/faulty circuits are denoted as \( gd_{g(SMA)} \) and \( gd_{f(SMA)} \), respectively, and the product of all SMAs is denoted as \( AND(SMA) \). The Boolean function of EAN at \( gd \) is

\[
AND(SMA) \cdot \overline{gd_{g(SMA)}} \cdot \overline{gd_{f(SMA)}}
\]

The Boolean function of ERN at \( gd \) is

\[
AND(SMA) \cdot gd_{g(SMA)} \cdot gd_{f(SMA)}
\]

**Theorem 2:** Given a Boolean network, a target wire \( wt \), a destination gate \( gd \), the EAN at \( gd \), and the ERN at \( gd \). The functionality of \( (gd+ERN) \cdot EAN \) after removing \( wt \) is equivalent to that of the original Boolean network. The general scheme of the rectification network is as shown in Fig. 2(a).

**Theorem 3:** The EAN at \( gd \) can be simplified from (1) to

\[
AND(SMA) \cdot gd_{g(SMA)}
\]

The ERN at \( gd \) can be simplified from (2) to

\[
AND(SMA) \cdot gd_{g(SMA)}
\]

**IV. SINGLE ALTERNATIVE WIRE IDENTIFICATION USING THE IRRA APPROACH**

In the RAR, a redundant wire \( wr \) is a Forward Alternative Wire (FAW) of \( wt \) if the addition wire \( wr \) blocks the fault effect. On the other hand, if the addition wire \( wr \) violates a forced MA, this wire is a Backward Alternative Wire (BAW) of \( wt \).

In the IRRA, if the destination gate \( gd \) in Fig. 2(b) has an MA \( D \{ \overline{D} \} \), then \( D \{ \overline{D} \} \) represents 0/1 in the good/faulty circuit, the value of \( gd_{g(SMA)} \) will be 0 \{1\}. This causes the EAN \{ERN\} to be a constant 0 network and cause the EAN \{ERN\} to leave the \( AND(SMA) \) term. Thus, the scheme becomes Fig. 3(a) \{Fig. 3(b)\}. The \( AND(SMA) \) term in Fig. 3 can be seen as an MA 1 which blocks the fault effect at \( gn \). Thus, the wire \( (AND(SMA) \rightarrow gn) \) is the FAW of the target wire. On the other hand, if a gate which is not a dominator but has a forced MA 1 \{0\}, we can also violate the forced MA by adding the \( AND(SMA) \) term. The scheme is also shown as Fig. 3(a) \{Fig. 3(b)\}. The new gate \( gn \) will have a forced MA 1 \{0\}. But \( AND(SMA) \) term which has the MA 1 will cause the \( gn \) value become 0 \{1\}, this violates the forced MA at \( gn \). Thus, the wire \( (AND(SMA) \rightarrow gn) \) is the BAW of the target wire.
substituting for all SMAs. This MA is one end point of the alternative wire (g8 → g11) for wt(g1 → g2) in (a).

With the MAs computed from the wt and SMA classification, Fig. 5 lists all types of substitution relationships for an AND gate. Suppose a is an irredundant SMA and b, c, d, and e are known MAs. The purpose is to determine whether the value of a can be backward implied from the output of the gate. Fig. 5(a)-(c) are the trivial cases. In Fig. 5(d), a is a controlling value 0 and e has a noncontrolling value 1. If e is a redundant SMA, g4 = 0 can imply a = 0, and g4 can replace a. On the other hand, if e is a semi-redundant SMA, we cannot ensure whether g4 = 0 can imply a = 0. This is because if e depends on the irredundant SMA a, e is unknown before a = 0 is implied. Thus, a = 0 cannot be implied from g4 = 0. The types of substitution relationships for an OR gate are summarized in Fig. 6.

For the last example in Fig. 4(c), by referring Fig. 5 and Fig. 6, the substitution set for the irredundant SMA a is \{g6, g8\}. By the same manner as a, the substitution set for the irredundant SMA b is \{g7, g8\}. The intersection of these two sets is \{g8\}. Thus, the wire (g8 → g11) is a single alternative wire for wt. The final circuit is as shown in Fig. 4(d).

V. EXPERIMENTAL RESULTS

Two experiments are conducted to demonstrate the effectiveness of the proposed IRRA technique.

The proposed single alternative wire identification algorithm was implemented in C and the experiments were conducted over a set of ISCAS85 and MCNC benchmarks within SIS (10) environment on a Sun Blade 2500 workstation with 4GBytes memory. Since the circuits under consideration are only consist of AND, OR, and INV gates, we decompose the complex gates into these primitive 2-input gates by using decomp_tech_network command in SIS. Additionally, recursive learning technique is applied in these experiments with depth=1.

Table I shows the results for the single alternative wire identification that compared with the previous work [5] on the same platform. Nt represents the total number of target wires in each benchmark. % represents the percentage of target wires having alternative wires. Time(S) represents the CPU time measured in seconds. For example in c1908 circuit, the percentage of target wires having single alternative wires in [5] and ours are 47.95% and 66.15%, respectively. The CPU time needed are 169.85 seconds and 70.14 seconds, respectively.

According to Table I, our approach gets 16% improvement on the percentage of target wires having alternative wires,
This paper proposes a logic restructuring technique, IRRA. IRRA can remove any desired wire and rectify the error due to the removal of the wire. A single alternative wire identification procedure is also proposed from the IRRA technique. The experimental results show the effectiveness and efficiency of our approach as compared with the state-of-the-art work. The application of area optimization with the IRRA technique is also demonstrated in this paper. It is very promising that the characteristic of IRRA allows restructuring the circuits more widely such that different optimization objectives could be achieved.

VI. CONCLUSIONS

REFERENCES


