Observability Relations for Multi-Output Nodes *

Hamid Savoj Robert K. Brayton
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720

February 16, 1993

Abstract

The observability relation $\mathcal{O}(\mathbf{x}, \mathbf{z})$ (as defined by Cerny [5]) or Boolean relation (discussed in [3]) provides a description of all the flexibility available in implementing a Boolean network $\mathcal{N}$. In this paper, we develop techniques for finding observability relations for each component of a Boolean network decomposed into a set of multi-output nodes. The multi-output node can be a Boolean network itself. The original decomposition can be obtained in a variety of ways. For example, multi-output nodes can be obtained by clustering a set of single-output nodes in a regular Boolean network. Or, a Boolean network may be partitioned into a hierarchy of smaller networks such that each network satisfies some specific criteria. We show how to obtain maximum flexibility for implementing each element of a partition by computing its observability relation. Compatible observability relations can also be computed for a given topological ordering of the nodes. A Boolean relation minimizer (such as [11]) can then be used to find a good two-level implementation from the observability relation computed for the node. Alternately, compatible don't cares can be derived from these observability relations and then used to optimize the multi-level network at the node using a conventional two-level minimizer.

*This project was supported in part by NSF/DARPA under contract number MIP-8719546 and grants from DEC, IBM, and Intel.

1 Introduction

An important part of logic synthesis is the node simplification phase where the local function at a node in a Boolean network is minimized using a two-level minimizer (such as ESPRESSO [2]) and using don't cares derived from the environment of the node. These don't cares arise from various sources; external don't cares (EDC), satisfiability don't cares (SDC), and observability don't cares (ODC). It is shown in [1] that if a node is minimized using all three types of don't cares, then each connection to and inside that node is irredundant (and hence there exists a test for both stuck-at-1 and stuck-at-0 for each connection). In this sense the don't cares so defined are *complete*.

This theory is explicitly for the case where each node in the Boolean network is a single output node. For multiple output nodes, a corresponding theory is not available. In particular, one can view the Boolean network itself as a single multiple-output node, for which a complete don't care theory is missing. This lack becomes especially apparent when attempting to specify and use external don't cares for combinational logic minimization. In practice, each separate output of the network is given an external don't care set $d_i(\mathbf{x})$. If these are specified as functions of primary inputs **x**, they must be mutually independent or *compatible* [7]. On the other hand, in [3], it was pointed out that such external don't cares can never be complete in the sense that they provide a description of *all* the flexibility allowed in simplifying a circuit. To circumvent this, Boolean relations were defined and techniques for finding minimal sum-of-products representations implementing a Boolean relation were given. Previously, Cerny [5] had defined the observability relation for a circuit. It turns out that combinations of these ideas are the seed for a complete theory of don't cares for multiple-output nodes of logic networks. This paper expands on these ideas and provides techniques for computing all the flexibility allowed for minimizing multi-output nodes of a logic network.

In [9], an observability network $\mathcal{N}'$ is defined for a given Boolean network $\mathcal{N}$ by adding a **single** node whose logic function is the observability relation $\mathcal{O}(\mathbf{x}, \mathbf{z})$ for $\mathcal{N}$. It is shown that the regular treatment of ODC's for $\mathcal{N}'$ includes all the flexibility allowed by the Boolean relation.

We develop techniques for computing observability relations for each multi-output node of a Boolean network given the observability relation $\mathcal{O}(\mathbf{x}, \mathbf{z})$ of the network. Both maximal and compatible observability relations are derived. Finally, we present algorithms for visiting each of the multi-output nodes of the network, computing its maximal or compatible observability relation, and simplifying it using a Boolean relation minimizer.

The main contribution of this paper is to round out the extensive theory of don't cares developed for single output nodes to networks of multiple-output nodes. In theory, this should provide more flexibility for minimizing networks. In particular, such additional optimization will depend on clustering techniques and the power of the Boolean relatoion minimizer used.

2 Terminology and Notation

Let $x_1, x_2, \ldots, x_n$ be the variables of the space B^n. We use $\mathbf{x}$ to represent a vector of variables or a vertex in B^n. The primary inputs of a Boolean network are represented by $\mathbf{x}$, and its primary outputs by $\mathbf{z}$. We assume the network is decomposed into a set of multi-output nodes.

2.1 Observability Relations

Observability relations or characteristic functions were introduced by Cerny [5]. Later the notion of a general Boolean relation [3] was discussed and derived for a hierarchy of networks. These concepts apply equally to networks that are completely specified, those with external don't cares, and those with specified Boolean relations. For the most general specification we define Boolean relations.

Definition 2.1 *A Boolean relation is a one-to-many multi-output Boolean mapping $\mathcal{R} : B^n \longrightarrow B^l$. In general, for each $\mathbf{x} \in B^n$, $\mathcal{R}(\mathbf{x}) \subseteq B^l$ is a set.*

Definition 2.1 *A Boolean relation is well-defined if there exists at least one minterm $\mathbf{z} \in B^l$ for every $\mathbf{x} \in B^n$ such that $\mathbf{z} \in \mathcal{R}(\mathbf{x})$.*

We represent the Boolean relation by an observability relation [5], $\mathcal{O} : B^{n+l} \longrightarrow B$, defined as $\mathcal{O}(\mathbf{x}, \mathbf{z}) = \{(\mathbf{x}, \mathbf{z}) | \mathbf{x} \in B^n, \mathbf{z} \in B^l, \mathbf{z} \in \mathcal{R}(\mathbf{x})\}$. External don't cares are special cases of Boolean relations; they are specified for each output and are input combinations under which that output can have any value.

For example, let $g_1, \ldots, g_l$ be the global functions of a network $\mathcal{N}$, expressing each primary output in terms of primary inputs. In case of a completely specified circuit, the observability relation (equivalently, the specification) of $\mathcal{N}$ is

$$\mathcal{O} = (z_1 \overline{\oplus} g_1)(z_2 \overline{\oplus} g_2) \ldots (z_l \overline{\oplus} g_l).$$

If the external don't cares $d_1, \ldots d_l$, are also given in terms of primary inputs, then the observability relation of $\mathcal{N}$ is

$$\mathcal{O}(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^{l} (d_i + z_i \overline{\oplus} g_i).$$

To represent and use this flexibility in a synthesis system we propose the following.

Definition 2.2 *The observability network $\mathcal{N}'$ of $\mathcal{N}$ is derived by adding one additional node $\mathcal{O}$ to $\mathcal{N}$. The logic function for this node is the observability relation $\mathcal{O}(\mathbf{x}, \mathbf{z})$ of $\mathcal{N}$. $\mathcal{N}'$ has only one output, namely $\mathcal{O}$, and $\mathcal{O}$ has $n + l$ inputs, namely all primary inputs and outputs of $\mathcal{N}$.*

2.2 Set Operations and BDD's

The *cofactor* of a sum-of-products f with respect to a literal $x_i(\overline{x_i})$, denoted by $f_{x_i}(f_{\overline{x_i}})$, is a new function obtained by substituting $1(0)$ for $x_i(\overline{x_i})$ in every cube in f which contains $x_i(\overline{x_i})$.

The Shannon's expansion of a Boolean function f with respect to a variable x is

$$f = x f_x + \overline{x} f_{\overline{x}}.$$

BDD's [4] are compact representations of a recursive Shannon decomposition. They are unique for a given variable ordering and hence are canonical forms for representing Boolean functions. They can be constructed from the Shanon's expansion of a Boolean function by **1)** deleting a node whose two child edges point to the same node, and **2)** sharing isomorphic subgraphs. Technically the result is a reduced ordered BDD, (ROBDD), which we shall just call BDD.

The *consensus operator or universal abstraction* applied to a function f with respect to a variable x_i is

$$\mathcal{C}_{x_i} f \equiv f_{x_i} f_{\overline{x}_i}.$$

This is the largest Boolean function independent of x_i contained in f.

The *smoothing operator or existential abstraction* applied to a function f with respect to a variable x_i is

$$\mathcal{S}_{x_i} f \equiv f_{x_i} + f_{\overline{x}_i}.$$

This is the smallest Boolean function independent of x_i which contains f. Thus

$$\mathcal{C}_{x_i} f \subseteq f \subseteq \mathcal{S}_{x_i}.$$

3 Previous Work

In [5], an approach is developed for a unified synthesis of combinational and sequential circuits using characteristic functions. Each circuit is composed of a set of multi-output blocks. The input-output behaviour of the whole circuit is described by a Boolean relation called the Output Characteristic Function (OCF). It is assumed that the inputs to the circuit denoted by $\mathbf{x}$ are also inputs to all the multi-output blocks. Techniques are developed for finding Boolean relations describing input-output behaviour of some multi-output node when the OCF of the circuit and the characteristic functions of all other nodes are known. It is shown that the OCF can be computed in two different ways if all the blocks in the circuit are combinational implementations since the characteristic function of a combinational implementation of a block allows a unique output for each input. Here we build on the work in [5] and relate these ideas to don't care theory. In particular, we derive formulas for compatible and maximal observability relations for a multi-output node n in a combinational circuit, given the observability relation (OCF in [5]) for the circuit and Boolean relations for all other nodes. We also show that if the Boolean relations given for all nodes except node n, are the characteristic functions of an implementation at those nodes, then for node n, the compatible and maximal observability relations are equivalent in all cases where a Boolean relation can be computed. As in [5], we first develop techniques for computing Boolean relations for a two-way partitioned circuit and then generalize these techniques for any other decomposition of a circuit into multi-output nodes. The techniques used to show the correctness of the formulas for computing observability relations of multi-output nodes are different from that of [5] and clarify the distinction between maximal and compatible Boolean relations.

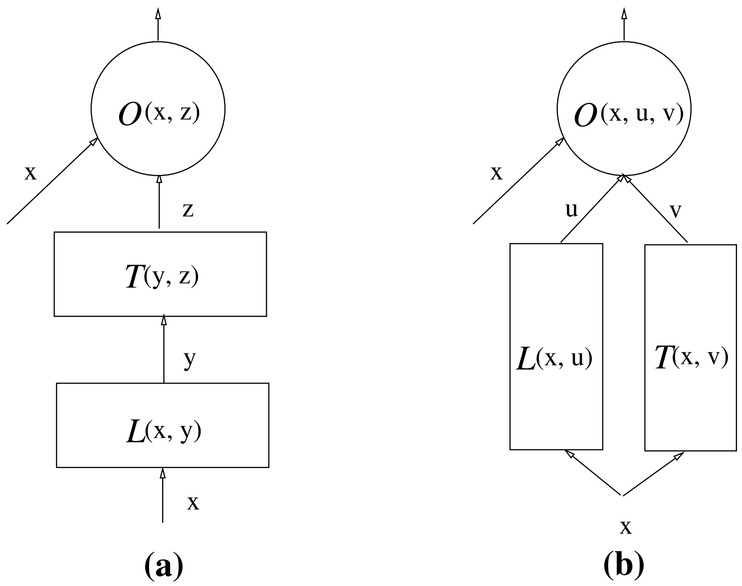

Figure 1: Decomposition of Observability Relation

4 Two-Way Partitioning of a Boolean Relation

First, techniques for two-way serial and parallel decomposition of an observability relation are discussed. These ideas are then used to find the observability relation for a general multi-output node in a network. Consider Figure 1. An external observability relation or Boolean relation $\mathcal{O}$ is specified for a network $\mathcal{N}$. The observability network $\mathcal{N}'$ is shown. $\mathcal{N}$ is decomposed into two parts serially in **(a)** and in parallel in **(b)**. The objective is to find maximal and compatible Boolean relations for the components of the network.

Definition 4.1 *A function* $\mathbf{y} = f(\mathbf{x})$ *is allowed by an observability relation* $\mathcal{O}(\mathbf{x}, \mathbf{y})$, *written* $f \prec \mathcal{O}$, *if* $\mathbf{y}\overline{\oplus} f(\mathbf{x}) \subseteq \mathcal{O}(\mathbf{x}, \mathbf{y})$.

Definition 4.2 *For a given set of nodes, an associated set of observability relations is* compatible *if each function at each node can be changed (as allowed by its observability relation) independent of all allowable changes in the functions at other nodes in the set.*

Compatible observability relations computed for the components of a network have the extra advantage that each node can be optimized irrespective of other nodes since the optimum choice at one node does not affect the set of functions compatible at the other nodes. However, use of compatible relations restricts the flexibility available at the given node.

The term *implementation* is used in this paper with the following meaning.

Definition 4.3 *An* implementation *is an observability relation with exactly one output minterm for each input minterm. Thus an implementation is a completely specified function.*

Definition 4.4 *Given relations $\mathcal{T}$, $\mathcal{L}$, $\mathcal{O}$, and a series or parallel structure (Figure 1*(**a**) *or 1*(**b**)*), $\mathcal{T}$ is* consistent *with $((\mathcal{L}, \mathcal{O})$ if for each $f \prec \mathcal{T}$, there exists $g \prec \mathcal{L}$ such that $(f, g) \prec \mathcal{O}$.*

A similar definition holds for $\mathcal{L}$ consistent with $(\mathcal{T}, \mathcal{O})$. For the case $\mathcal{L} = 1$, we say that $\mathcal{T}$ is consistent with $\mathcal{O}$, if for each $f \prec \mathcal{T}$ there exists some function g such that $(f, g) \prec \mathcal{O}$.

Definition 4.5 *Given relations $\mathcal{T}$, $\mathcal{L}$, $\mathcal{O}$, and a series or parallel structure, $(\mathcal{L}, \mathcal{T})$ is* compatible *with $\mathcal{O}$ if for all $f \prec \mathcal{T}$ and all $g \prec \mathcal{L}$, $(f, g) \prec \mathcal{O}$.*

Definition 4.6 *A* minimal separating set of nodes *is a set of nodes which separates all primary outputs from all primary inputs if all nodes in the set are removed.*

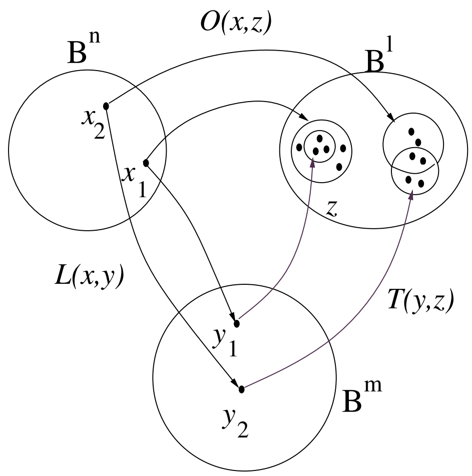

Figure 2: Observability Relation for a Separating Set of Variables

4.1 Serial Decomposition

We first consider the case where $\mathcal{T}$ and $\mathcal{O}$ are given and we want to derive relations for $\mathcal{L}$. Let $\mathcal{O}(\mathbf{x}, \mathbf{z})$ be the observability relation (specification) for the network, and $\mathcal{T}(\mathbf{y}, \mathbf{z})$ an observability relation for the top part of the network consistent with $\mathcal{O}(\mathbf{x}, \mathbf{y})$ as shown in Figure 1(a). First, we derive a relation $\mathcal{L}$ for the lower part compatible with $\mathcal{T}$. Second, we find a relation $\mathcal{L}$ which is maximal (but not necessarily compatible) with respect to $\mathcal{T}$. In both cases $\mathcal{L}$ is consistent with $(\mathcal{T}, \mathcal{O})$. The mapping represented by $\mathcal{L}(\mathbf{x}, \mathbf{y})$ is shown in Figure 2. It gives the minterms in the separating set $\mathbf{y}$ allowed for some particular minterm $\mathbf{x}_i$.

The set of $\mathbf{z}$'s in relation with $\mathbf{x}_i$ allowed by $\mathcal{O}$ is $\mathcal{O}_{\mathbf{x}_i}$ and the set of $\mathbf{z}$'s allowed by $\mathcal{T}$ for a particular $\mathbf{y}_j$ is $\mathcal{T}_{\mathbf{y}_j}$. We investigate three possibilities; $\mathcal{T}_{\mathbf{y}_j} \subseteq \mathcal{O}_{\mathbf{x}_i}$ (represented by $\mathbf{x_1}$ and $\mathbf{y}_1$ in Figure 2), $\mathcal{T}_{\mathbf{y}_j} \not\subseteq \mathcal{O}_{\mathbf{x}_i}$ but $\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} \neq 0$ (represented by $\mathbf{x}_2$ and $\mathbf{y}_2$ in Figure 2), and finally $\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} = 0$.

If $\mathcal{T}_{\mathbf{y}_j} \subseteq \mathcal{O}_{\mathbf{x}_i}$, $\mathbf{x}_i$ can be put in relation with $\mathbf{y}_j$ ($(\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{L}$), and no constraint is imposed on $\mathcal{T}$. This is because $\mathbf{y}_j$ can accept any $\mathbf{z}_k$ allowed by $\mathcal{T}$ and still $(\mathbf{x}_i, \mathbf{z}_k) \in \mathcal{O}$. The relation $\mathcal{L}$ computed this way is compatible with $\mathcal{T}$ and consistent with $\mathcal{O}$. This means an implementation can be found from $\mathcal{L}$ independent of that found from $\mathcal{T}$ (this is the case for $\mathbf{x}_1$ and $\mathbf{y}_1$ shown in Figure 2).

If $\mathcal{T}_{\mathbf{y}_j} \not\subseteq \mathcal{O}_{\mathbf{x}_i}$, $\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} \neq 0$, we can put $(\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{L}$, but some constraint is now imposed on $\mathcal{T}$. If an implementation based on $\mathcal{L}$ gives $\mathbf{y}_j$ for input $\mathbf{x}_i$, any $\mathbf{z}_k$ allowed for $\mathbf{y}_j$ (by T) must be in $\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i}$. The relations $\mathcal{T}$ and $\mathcal{L}$ are not compatible in this case. Still, $\mathcal{L}$ can be used to find an implementation for the lower part of the circuit. However, $\mathcal{T}$ must be made compatible to this implementation before it is used to find an implementation for the upper part of the circuit. We will show that $\mathcal{L}$ obtained this way is the maximal observability relation consistent with $(\mathcal{T}, \mathcal{O})$.

Finally, if $\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} = 0$, $(\mathbf{x}_i, \mathbf{y}_j)$ cannot be in $\mathcal{L}$. Any function g such that $\mathbf{y}_j = g(\mathbf{x}_i)$ is inconsistent with $(\mathcal{T}, \mathcal{O})$.

Lemma 4.1 *(**compatible**) Given relations $\mathcal{T}(\mathbf{y}, \mathbf{z})$ and $\mathcal{O}(\mathbf{x}, \mathbf{z})$,*

$$\mathcal{L}^{cm}(\mathbf{x}, \mathbf{y}) = \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{y}, \mathbf{z}) + \mathcal{O}(\mathbf{x}, \mathbf{z}))$$

is the maximal relation compatible with $\mathcal{T}$ and consistent with $\mathcal{O}$.

Proof We first show that $\mathcal{L}$ is maximally compatible with $\mathcal{T}$ if and only if $\mathcal{L} = \{(\mathbf{x}_i, \mathbf{y}_j) | \mathcal{T}_{\mathbf{y}_j} \subseteq \mathcal{O}_{\mathbf{x}_i}\}$. Assume $(\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{L}$ and $\mathcal{T}_{\mathbf{y}_j} \subseteq \mathcal{O}_{\mathbf{x}_i}$. Then for all $\mathbf{z}_k$ such that $(\mathbf{y}_j, \mathbf{z}_k) \in \mathcal{T}$ we have $(\mathbf{x}_i, \mathbf{z}_k) \in \mathcal{O}$. Thus any function $f \prec \mathcal{L}$ where $\mathbf{y}_j = f(\mathbf{x}_i)$ is compatible with any $g \prec \mathcal{T}$ since $g(\mathbf{y}_j) = \mathbf{z}_k$ for some $\mathbf{z}_k$ and $(\mathbf{x}_i, \mathbf{z}_k) \in \mathcal{O}$. Conversely, assume $(\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{L}$ and $\mathcal{T}_{\mathbf{y}_j} \not\subseteq \mathcal{O}_{\mathbf{x}_i}$. Then, there is $\mathbf{z}_k$ such that $(\mathbf{y}_j, \mathbf{z}_k) \in \mathcal{T}$ and $(\mathbf{x}_i, \mathbf{z}_k) \notin \mathcal{O}$. As a result, there could exist $f \prec \mathcal{L}$ where $\mathbf{y}_j = f(\mathbf{x}_i)$, and $g \prec \mathcal{T}$ where $g(\mathbf{y}_j) = \mathbf{z}_k$, but $(\mathbf{x}_i, \mathbf{z}_k) \notin \mathcal{O}$. f is not consistent with $(g, \mathcal{O})$. Thus $\mathcal{L}$ and $\mathcal{T}$ are not compatible.

Now $\mathcal{T}_{\mathbf{y}_j} \subseteq \mathcal{O}_{\mathbf{x}_i}$ if and only if $\overline{\mathcal{T}}_{\mathbf{y}_j} + \mathcal{O}_{\mathbf{x}_i} = 1$ i.e. $\mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}_{\mathbf{y}_j} + \mathcal{O}_{\mathbf{x}_i}) = 1$. $\mathcal{L}^{cm}$ is then

$$\begin{aligned}
\mathcal{L}^{cm}(\mathbf{x}, \mathbf{y}) &= \sum_{(\mathbf{x}_i, \mathbf{y}_j) \in B^{n+m}} \mathbf{x}_i \mathbf{y}_j \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}_{\mathbf{y}_j} + \mathcal{O}_{\mathbf{x}_i}) \\
&= \sum \mathcal{C}_{\mathbf{z}}(\mathbf{x}_i \mathbf{y}_j \overline{\mathcal{T}}_{\mathbf{y}_j} + \mathbf{x}_i \mathbf{y}_j \mathcal{O}_{\mathbf{x}_i}) \\
&= \sum \mathcal{C}_{\mathbf{z}}(\mathbf{x}_i \mathbf{y}_j \overline{\mathcal{T}} + \mathbf{x}_i \mathbf{y}_j \mathcal{O}) \\
&= \sum \mathbf{x}_i \mathbf{y}_j \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}} + \mathcal{O}) \\
&= \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}} + \mathcal{O}) \sum_{(\mathbf{x}_i, \mathbf{y}_j) \in B^{n+m}} \mathbf{x}_i \mathbf{y}_j \\
&= \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{y}, \mathbf{z}) + \mathcal{O}(\mathbf{x}, \mathbf{z})).
\end{aligned}$$

■

For $\mathcal{L}^{cm}$ to be a well-defined Boolean relation it is necessary to have $\mathcal{S}_{\mathbf{y}}\mathcal{L}^{cm}(\mathbf{x}, \mathbf{y}) = 1$. $\mathcal{L}^{cm}$ obtained as in Lemma 4.1 may not be a well-defined Boolean relation because there can be $\mathbf{x}$'s for which no $\mathbf{y}$ is allowed. This simply says that $\mathcal{T}$ may be too large, in the sense that for some $\mathbf{x}$ there is no $\mathbf{y}$ whose image $\mathcal{T}_{\mathbf{y}}(\mathbf{z})$ is contained in $\mathcal{O}_{\mathbf{x}}(\mathbf{z})$.

The concept of compatible observability relation is similar to compatible don't cares. As we know, one can also compute maximal don't cares. Likewise, the relation $\mathcal{L}$ can be computed in a way which results in a maximal Boolean relation. The restriction that $\mathcal{L}$ must be compatible with $\mathcal{T}$ is relaxed. But it is required that for any implementation based on $\mathcal{L}$ there is at least one possible implementation based on $\mathcal{T}$ allowed by $\mathcal{O}$ i.e. $\mathcal{L}$ is consistent with $(\mathcal{T}, \mathcal{O})$.

Lemma 4.2 *(***maximal and consistent***) Given relations $\mathcal{T}(\mathbf{y}, \mathbf{z})$ and $\mathcal{O}(\mathbf{x}, \mathbf{z})$, the maximal relation $\mathcal{L}^m(\mathbf{x}, \mathbf{y})$ consistent with $(\mathcal{T}, \mathcal{O})$ is*

$$\mathcal{L}^m(\mathbf{x}, \mathbf{y}) = \mathcal{S}_{\mathbf{z}}\mathcal{O}(\mathbf{x}, \mathbf{z})\mathcal{T}(\mathbf{y}, \mathbf{z}).$$

Proof We first prove that $\mathcal{L} = \{(\mathbf{x}_i, \mathbf{y}_j) | \mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} \neq 0\}$ is the maximal relation consistent with $(\mathcal{T}, \mathcal{O})$. If $\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} \neq 0$, then there exists at least one $\mathbf{z}_k$ such that $\mathbf{z}_k \in \mathcal{T}_{\mathbf{y}_j}$ and $\mathbf{z}_k \in \mathcal{O}_{\mathbf{x}_i}$. As a result, for any $\mathcal{L}$ where $\mathbf{y}_j = f(\mathbf{x}_i)$ there is at least one $g \prec \mathcal{T}$ with $\mathbf{z}_k = g(\mathbf{y}_i)$ and $(\mathbf{x}_i, \mathbf{z}_k) \in \mathcal{O}$. On the other hand, if $\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} = 0$ for $(\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{L}$, there is no $\mathbf{z}_k$ where $(\mathbf{y}_j, \mathbf{z}_k) \in \mathcal{T}$ and $(\mathbf{x}_i, \mathbf{z}_k) \in \mathcal{O}$, so a function f would exist such that $f \prec \mathcal{L}$ but no $g \prec \mathcal{T}$ would exist where $(f, g) \prec \mathcal{O}$. Thus $\mathcal{L}$ is not consistent with $\mathcal{T}$ and $\mathcal{O}$.

$\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i} \neq 0$ if and only if $\mathcal{S}_{\mathbf{z}}(\mathcal{T}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{x}_i}) = 1$. Therefore we can compute $\mathcal{L}^m(\mathbf{x}, \mathbf{y})$ by

$$\begin{aligned}
\mathcal{L}^m(\mathbf{x}, \mathbf{y}) &= \sum_{(\mathbf{x}_i, \mathbf{y}_j) \in B^{n+m}} \mathbf{x}_i \mathbf{y}_j \mathcal{S}_{\mathbf{z}}(\mathcal{O}_{\mathbf{x}_i}\mathcal{T}_{\mathbf{y}_j}) \\
&= \sum \mathcal{S}_{\mathbf{z}}(\mathbf{x}_i \mathbf{y}_j \mathcal{O}_{\mathbf{x}_i}\mathcal{T}_{\mathbf{y}_j}) \\
&= \sum \mathcal{S}_{\mathbf{z}}(\mathbf{x}_i \mathbf{y}_j \mathcal{O}\mathcal{T}) \\
&= (\mathcal{S}_{\mathbf{z}}\mathcal{O}\mathcal{T}) \sum_{(\mathbf{x}_i, \mathbf{y}_j) \in B^{n+m}} \mathbf{x}_i \mathbf{y}_j \\
&= \mathcal{S}_{\mathbf{z}}(\mathcal{O}(\mathbf{x}, \mathbf{z})\mathcal{T}(\mathbf{y}, \mathbf{z})).
\end{aligned}$$

∎

Note the difference between Lemmas 4.1 and 4.2; Lemma 4.1 requires compatibility with respect $(\mathcal{T}, \mathcal{O})$ while 4.2 only consistency with $(\mathcal{T}, \mathcal{O})$. Clearly, $\mathcal{L}^{cm} \subseteq \mathcal{L}^m$; hence the latter is called maximal.

Again, $\mathcal{L}^m$ may not be well-defined if $\mathcal{S}_{\mathbf{y}}\mathcal{L}^m(\mathbf{x}, \mathbf{y}) \neq 1$. This just means that $\mathcal{T}$ was not consistent with $\mathcal{O}$, because there exist no function for the lower part of the circuit that is consistent with $\mathcal{T}, \mathcal{O}$.

Lemma 4.3 *If $\mathcal{T}(\mathbf{y}, \mathbf{z})$ is an implementation (i.e. for each $\mathbf{y}$ there exists a unique $\mathbf{z}$ such that $\mathcal{T}(\mathbf{y}, \mathbf{z}) = 1$), then $\mathcal{L}^{cm}(\mathbf{x}, \mathbf{y}) = \mathcal{L}^m(\mathbf{x}, \mathbf{y})$.*

Proof Since $\mathcal{T}$ is an implementation ($\mathbf{z} = g(\mathbf{y})$), $\mathcal{T} = \mathbf{z}g + \overline{\mathbf{z}g}$. Then

$$\begin{aligned}
\mathcal{S}_{\mathbf{z}}\mathcal{O}\mathcal{T} &= \mathcal{S}_{\mathbf{z}}\mathcal{O}(\mathbf{z}g + \overline{\mathbf{z}g}) \\
&= \mathcal{O}_{\mathbf{z}}g + \mathcal{O}_{\overline{\mathbf{z}}}\overline{g}.
\end{aligned}$$

In the same way,

$$\begin{aligned}\mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}+\mathcal{O}) &= \mathcal{O}_{\mathbf{z}}g+\mathcal{O}_{\overline{\mathbf{z}}}\overline{g}+\mathcal{O}_{\mathbf{z}}\mathcal{O}_{\overline{\mathbf{z}}}\\ &= \mathcal{O}_{\mathbf{z}}g+\mathcal{O}_{\overline{\mathbf{z}}}\overline{g}.\end{aligned}$$

Therefore $\mathcal{L}^{cm}=\mathcal{L}^{m}$. ■

We can find a relation $\mathcal{L}^{g_i}=\mathcal{O}_{\mathbf{z}}g_i+\mathcal{O}_{\overline{\mathbf{z}}}\overline{g_i}$ that is maximal and compatible with respect to g_i for any function $g_i\prec\mathcal{T}$ because $\mathcal{T}$ is consistent with $\mathcal{O}$. Notice that $\mathcal{L}^{cm}$ can also be expressed as

$$\mathcal{L}^{cm} = \prod_{\forall g_i\prec\mathcal{T}}\mathcal{L}^{g_i}.$$

We now consider deriving maximal compatible and maximal consistent relations for $\mathcal{T}$ given relations $\mathcal{L}$ and $\mathcal{O}$. These results are different from the case where $\mathcal{T}$ and $\mathcal{O}$ are given since $\mathcal{T}$ and $\mathcal{L}$ are not symmetrically interchangeable. $\mathcal{T}$ is in series after $\mathcal{L}$.

Lemma 4.4 *(***compatible***) Given relations $\mathcal{L}(\mathbf{x},\mathbf{y})$ and $\mathcal{O}(\mathbf{x},\mathbf{z})$, the relation maximally compatible with $(\mathcal{L},\mathcal{O})$ is*

$$\mathcal{T}^{cm}(\mathbf{y},\mathbf{z})=\mathcal{C}_{\mathbf{x}}(\mathcal{O}(\mathbf{x},\mathbf{z})+\overline{\mathcal{L}}(\mathbf{x},\mathbf{y}))$$

Proof We first prove that $\mathcal{T}=\{(\mathbf{y}_j,\mathbf{z}_i)|\mathcal{L}_{\mathbf{y}_j}\subseteq\mathcal{O}_{\mathbf{z}_i}\}$ is maximally compatible with $\mathcal{L}(\mathbf{x},\mathbf{z})$ and consistent with $\mathcal{O}(\mathbf{x},\mathbf{z})$. If $\mathcal{L}_{\mathbf{y}_j}\subseteq\mathcal{O}_{\mathbf{z}_i}$, then any $\mathbf{x}_k$ in relation with $\mathbf{y}_j$ from $\mathcal{L}$ is also in relation with $\mathbf{z}_i$ allowed by $\mathcal{O}$, therefore, $\mathbf{y}_j\mathbf{z}_i\in\mathcal{T}$ consistent with $\mathcal{O}(\mathbf{x},\mathbf{z})$. There is no restriction on $\mathcal{L}$. On the other hand, if $\exists\mathbf{x}_k|\mathbf{x}_k\in\mathcal{L}_{\mathbf{y}_j},\mathbf{x}_k\notin\mathcal{O}_{\mathbf{z}_i}$, but $\mathbf{y}_j\mathbf{z}_i\in\mathcal{T}$, then one can choose an implementation producing output $\mathbf{z}_i$ for $\mathbf{y}_j$ from $\mathcal{T}$ and an implementation producing $\mathbf{y}_j$ for input $\mathbf{x}_k$ from $\mathcal{L}$. These two implementations are not allowed by $\mathcal{O}$ because they produce output $\mathbf{z}_i$ for input $\mathbf{x}_k$ and $(\mathbf{x}_k,\mathbf{z}_i)\notin\mathcal{O}$.

$\mathcal{L}_{\mathbf{y}_j}\subseteq\mathcal{O}_{\mathbf{z}_i}$ if and only if $\mathcal{C}_{\mathbf{x}}(\mathcal{O}_{\mathbf{z}_i}+\overline{\mathcal{L}}_{\mathbf{y}_j})=1$. The term $\mathbf{y}_j\mathbf{z}_i\mathcal{C}_{\mathbf{x}}(\mathcal{O}_{\mathbf{z}_i}+\overline{\mathcal{L}}_{\mathbf{y}_j})$ is equal to $\mathbf{y}_j\mathbf{z}_i$ if $\mathcal{C}_{\mathbf{x}}(\mathcal{O}_{\mathbf{z}_i}+\overline{\mathcal{L}}_{\mathbf{y}_j})=1$ and 0 otherwise. Therefore, we can compute $\mathcal{T}^{cm}$ by

$$\begin{aligned}\mathcal{T}^{cm}(\mathbf{y},\mathbf{z}) &= \sum_{(\mathbf{y}_j,\mathbf{z}_i)\in B^{m+l}}\mathbf{y}_j\mathbf{z}_i\mathcal{C}_{\mathbf{x}}(\mathcal{O}_{\mathbf{z}_i}+\overline{\mathcal{L}}_{\mathbf{y}_j})\\ &= \sum\mathbf{y}_j\mathbf{z}_i\mathcal{C}_{\mathbf{x}}(\mathbf{z}_i\mathcal{O}_{\mathbf{z}_i}+\mathbf{y}_j\overline{\mathcal{L}}_{\mathbf{y}_j})\\ &= \sum\mathbf{y}_j\mathbf{z}_i\mathcal{C}_{\mathbf{x}}(\mathbf{z}_i\mathcal{O}+\mathbf{y}_j\overline{\mathcal{L}})\\ &= \mathcal{C}_{\mathbf{x}}(\mathcal{O}+\overline{\mathcal{L}})\sum_{(\mathbf{y}_j,\mathbf{z}_i)\in B^{m+l}}\mathbf{y}_j\mathbf{z}_i\\ &= \mathcal{C}_{\mathbf{x}}(\mathcal{O}(\mathbf{x},\mathbf{z})+\overline{\mathcal{L}}(\mathbf{x},\mathbf{y})).\end{aligned}$$

■

If $\mathcal{L}$ is an implementation, $\mathcal{T}^{cm}$ is also maximal. It would seem from the previous lemmas that a relation $\mathcal{T}$ which is maximally consistent with $(\mathcal{L},\mathcal{O})$ is

$$\mathcal{T}(\mathbf{y},\mathbf{z})=\mathcal{S}_{\mathbf{x}}(\mathcal{O}(\mathbf{x},\mathbf{z})\mathcal{L}(\mathbf{x},\mathbf{y})).$$

This formula is obtained from the assumption that a relation is maximally consistent with $(\mathcal{L}, \mathcal{O})$ if and only if $\mathcal{L}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{z}_i} \neq 0$ for any $\mathbf{y}_j\mathbf{z}_i$ in the relation. However, this relation may not be well-defined. Suppose $\mathcal{L}_{\mathbf{y}_j} = 0$ ($\mathbf{y}_j$ cannot be obtained by any $\mathbf{x}$). Then $\mathcal{L}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{z}_i} = 0$ and no $\mathbf{z}$ is allowed for such $\mathbf{y}$. We can attempt to add another term to the above equation to take care of all the $\mathbf{y}$'s that cannot be obtained from $\mathcal{L}$; Thus add $(\mathbf{y}, \mathbf{z})$ for any $\mathbf{z}$ if $\mathcal{L}_{\mathbf{y}} = 0$

$$\mathcal{T}(\mathbf{y}, \mathbf{z}) = \mathcal{S}_{\mathbf{x}}(\mathcal{O}(\mathbf{x}, \mathbf{z})\mathcal{L}(\mathbf{x}, \mathbf{y})) + \overline{\mathcal{S}_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \mathbf{y})}. \tag{1}$$

$\overline{\mathcal{S}_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \mathbf{y})}$ is all the $\mathbf{y}$'s not obtained from any $\mathbf{x}$, therefore all $\mathbf{z}$'s are allowed for any such $\mathbf{y}$.

The above formula is also not correct becasue $\mathcal{T}$ may not be consistent with $(\mathcal{L}, \mathcal{O})$. Assume an implementation based on $\mathcal{T}$ gives $\mathbf{z}_i$ for $\mathbf{y}_j$. Then any implementation for the lower part of the circuit must choose $\mathbf{x}_k \in \mathcal{L}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{z}_i}$ to give $\mathbf{y}_j$, but there is no guarantee that all the $\mathbf{x}$'s can be covered this way. There could be an $\mathbf{x}$ that is in no $\mathcal{L}_{\mathbf{y}_j}\mathcal{O}_{\mathbf{z}_i}$ where $\mathbf{z}_i$ is the output for $\mathbf{y}_j$ in the implementation based on $\mathcal{T}$. The following is an example.

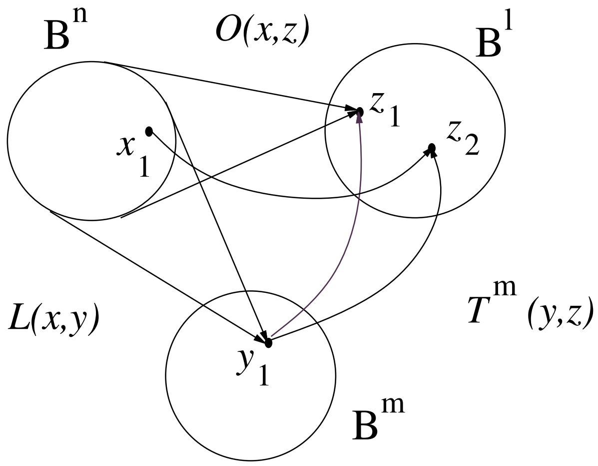

Figure 3: Example

<u>**Example:**</u>

The example is illustrated in Figure 3. Assume $\mathcal{L}$ is an implementation which gives $\mathbf{y}_1$ for every possible input $\mathbf{x}$. Also assume $\mathcal{O}$ is such that it allows $\mathbf{z}_1$ for every input $\mathbf{x}$ and $\mathbf{z}_2$ only for input $\mathbf{x}_1$ as shown in Figure 3. Clearly, there is an implementation for the circuit consistent with $(\mathcal{L}, \mathcal{O})$; the implementation for $\mathcal{T}$ can give $\mathbf{z}_1$ for every $\mathbf{y}$; the implementation for $\mathcal{L}$ is $\mathcal{L}$ itself; therefore, $\mathcal{L}$ is consistent with $\mathcal{O}$. If $\mathcal{T}$ is computed as in (1), then $\mathbf{y}_1\mathbf{z}_1, \mathbf{y}_1\mathbf{z}_2 \in \mathcal{T}$. All $\mathbf{z}$'s are allowed for any other $\mathbf{y}$. A possible implementation based on $\mathcal{T}$ can give $\mathbf{z}_2$ for all $\mathbf{y}$, but in such case, no implementation from $\mathcal{L}$ can be found which satisfies $\mathcal{O}$, therefore, (1) is not consistent with $(\mathcal{L}, \mathcal{O})$.

Every valid implementation $t \in \mathcal{T}$ must be consistent with $(\mathcal{L}, \mathcal{O})$. From Lemma 4.2

$$l(\mathbf{x}, \mathbf{y}) = \mathcal{S}_{\mathbf{z}}\mathcal{O}(\mathbf{x}, \mathbf{z})t(\mathbf{y}, \mathbf{z})$$

represents all the implementations for the lower part of the circuit consistent with $(\mathcal{T}, \mathcal{O})$. t is consistent with $\mathcal{O}$ if l is well-defined, i.e. $\mathcal{S}_{\mathbf{y}}\mathcal{S}_{\mathbf{z}}\mathcal{O}(\mathbf{x},\mathbf{z})t(\mathbf{y},\mathbf{z}) = 1$. t is consistent with $(\mathcal{L}, \mathcal{O})$ if $l\mathcal{L}$ is well-defined, i.e. $\mathcal{S}_{\mathbf{y}}(\mathcal{L}(\mathbf{x},\mathbf{y})\mathcal{S}_{\mathbf{z}}\mathcal{O}(\mathbf{x},\mathbf{z})t(\mathbf{y},\mathbf{z})) = 1$. As a result, the set of all implementations possible for the top part of the circuit is all $t \in T$ (given by (1)) such that $\mathcal{S}_{\mathbf{y}}(\mathcal{L}(\mathbf{x},\mathbf{y})\mathcal{S}_{\mathbf{z}}\mathcal{O}(\mathbf{x},\mathbf{z})t(\mathbf{y},\mathbf{z})) = 1$.

Unfortunately, we do not know how to express this as a Boolean relation. The above discussion seems to lead to a constrained form of Boolean relation minimization:

$$\begin{aligned} \min\ t &\leq \mathcal{S}_{\mathbf{x}}(\mathcal{O}(\mathbf{x},\mathbf{z})\mathcal{L}(\mathbf{x},\mathbf{y})) + \overline{\mathcal{S}_{\mathbf{x}}\mathcal{L}(\mathbf{x},\mathbf{y})} && (2)\\ s.t.\ & \quad \mathcal{S}_{\mathbf{y}}(\mathcal{L}(\mathbf{x},\mathbf{y})\mathcal{S}_{\mathbf{z}}\mathcal{O}(\mathbf{x},\mathbf{z})t(\mathbf{y},\mathbf{z})) \equiv 1. && (3)\end{aligned}$$

It is unknown if this can be rewritten in a simpler form.

4.2 Parallel Decomposition

Given the relation $\mathcal{T}$ consistent with the relation $\mathcal{O}$ as shown in Figure 1(b), we develop formulas for computing maximal and compatible $\mathcal{L}$.

Lemma 4.5 *(***compatible***) Given the relations $\mathcal{T}(\mathbf{x},\mathbf{v})$ and $\mathcal{O}(\mathbf{x},\mathbf{u},\mathbf{v})$, the relation maximally compatible with $(\mathcal{L}, \mathcal{O})$ is*

$$\mathcal{L}^{cm}(\mathbf{x},\mathbf{u}) = \mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}}(\mathbf{x},\mathbf{v}) + \mathcal{O}(\mathbf{x},\mathbf{u},\mathbf{v})).$$

Proof We first prove that $\mathcal{L} = \{(\mathbf{x}_i, \mathbf{u}_j) | \mathcal{T}_{\mathbf{x}_i} \subseteq \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}\}$. If $\mathcal{T}_{\mathbf{x}_i} \subseteq \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}$, then any $\mathbf{v}_k$ in relation with $\mathbf{x}_i$ from $\mathcal{T}$ also satisfies $(\mathbf{x}_i, \mathbf{u}_j, \mathbf{v}_k) \in \mathcal{O}$; therefore, $(\mathbf{x}_i, \mathbf{u}_j) \in \mathcal{L}$ is compatible with $\mathcal{T}$ and consistent with $\mathcal{O}$. On the other hand, if $\{\exists \mathbf{v}_k | \mathbf{v}_k \in \mathcal{T}_{\mathbf{x}_i}, \mathbf{v}_k \notin \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}\}$, then $(\mathbf{x}_i, \mathbf{u}_j) \in \mathcal{L}$ and $\mathbf{x}_i\mathbf{v}_k \in \mathcal{T}$ may put $(\mathbf{v}_k, \mathbf{u}_j)$ in relation with $\mathbf{x}_i$ but $(\mathbf{x}_i, \mathbf{u}_j, \mathbf{v}_k) \notin \mathcal{O}(\mathbf{x},\mathbf{u},\mathbf{v})$. As a result an implementation based on $\mathcal{T}$ and $\mathcal{L}$ could result in a circuit that violates $\mathcal{O}$.

$\mathcal{T}_{\mathbf{x}_i} \subseteq \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}$ if and only if $\mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}}_{\mathbf{x}_i} + \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}) = 1$. The term $\mathbf{x}_i\mathbf{u}_j\mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}}_{\mathbf{x}_i} + \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j})$ is equal to $\mathbf{x}_i\mathbf{u}_j$ if $\mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}}_{\mathbf{x}_i} + \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}) = 1$ and 0 otherwise. As a result, $\mathcal{L}^{cm}$ can be written as

$$\begin{aligned}\mathcal{L}^{cm}(\mathbf{x},\mathbf{u}) &= \sum \mathbf{x}_i\mathbf{u}_j\mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}}_{\mathbf{x}_i} + \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j})\\ &= \sum \mathcal{C}_{\mathbf{v}}(\mathbf{x}_i\mathbf{u}_j\overline{\mathcal{T}}_{\mathbf{x}_i} + \mathbf{x}_i\mathbf{u}_j\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j})\\ &= \sum \mathcal{C}_{\mathbf{v}}(\mathbf{x}_i\mathbf{u}_j\overline{\mathcal{T}} + \mathbf{x}_i\mathbf{u}_j\mathcal{O})\\ &= \sum \mathbf{x}_i\mathbf{u}_j\mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}} + \mathcal{O})\\ &= \mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}} + \mathcal{O})\sum \mathbf{x}_i\mathbf{u}_j\\ &= \mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}}(\mathbf{x},\mathbf{v}) + \mathcal{O}(\mathbf{x},\mathbf{u},\mathbf{v})).\end{aligned}$$

■

Lemma 4.6 *(***maximal and consistent***) Given relations $\mathcal{T}(\mathbf{x},\mathbf{v})$ and $\mathcal{O}(\mathbf{x},\mathbf{u},\mathbf{v})$, the maximal relation consistent with $(\mathcal{T}, \mathcal{O})$ is*

$$\mathcal{L}^{m}(\mathbf{x},\mathbf{u}) = \mathcal{S}_{\mathbf{v}}\mathcal{O}(\mathbf{x},\mathbf{u},\mathbf{v})\mathcal{T}(\mathbf{x},\mathbf{v}).$$

Proof We first prove that $\mathcal{L} = \{(\mathbf{x}_i, \mathbf{u}_j) | \mathcal{T}_{\mathbf{x}_i}\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j} \neq 0\}$ is the maximal relation consistent with $(\mathcal{T}, \mathcal{O})$. If $\mathcal{T}_{\mathbf{x}_i}\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j} \neq 0$, then there exists at least one $\mathbf{v}_k$ such that $\mathbf{v}_k \in \mathcal{T}_{\mathbf{x}_i}$ and $\mathbf{v}_k \in \mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}$. As a result, for any implementation based on $\mathcal{L}$ that gives $\mathbf{u}_j$ for $\mathbf{x}_i$, there is at least one implementation allowed by $\mathcal{T}$ that satisfies $\mathcal{O}$. On the other hand, if $\mathcal{T}_{\mathbf{x}_i}\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j} = 0$, then no $\mathbf{v}_k$ is allowed for any implementation that gives $\mathbf{u}_j$ for $\mathbf{x}_i$.

$\mathcal{T}_{\mathbf{x}_i}\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j} \neq 0$ if and only if $\mathcal{S}_{\mathbf{v}}(\mathcal{T}_{\mathbf{x}_i}\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}) = 1$. $\mathbf{x}_i\mathbf{u}_j\mathcal{S}_{\mathbf{v}}(\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}\mathcal{T}_{\mathbf{x}_i})$ is equal to $\mathbf{x}_i\mathbf{u}_j$ if $\mathcal{S}_{\mathbf{v}}(\mathcal{T}_{\mathbf{x}_i}\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}) = 1$ and 0 otherwise. As a result, we can write

$$\begin{aligned}
\mathcal{L}^m(\mathbf{x}, \mathbf{u}) &= \sum \mathbf{x}_i\mathbf{u}_j\mathcal{S}_{\mathbf{v}}(\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}\mathcal{T}_{\mathbf{x}_i}) \\
&= \sum \mathcal{S}_{\mathbf{v}}(\mathbf{x}_i\mathbf{u}_j\mathcal{O}_{\mathbf{x}_i\mathbf{u}_j}\mathcal{T}_{\mathbf{x}_i}) \\
&= \sum \mathcal{S}_{\mathbf{v}}(\mathbf{x}_i\mathbf{u}_j\mathcal{O}\mathcal{T}) \\
&= \mathcal{S}_{\mathbf{v}}(\mathcal{O}\mathcal{T})\sum \mathbf{x}_i\mathbf{u}_j \\
&= \mathcal{S}_{\mathbf{v}}(\mathcal{O}(\mathbf{x}, \mathbf{u}, \mathbf{v})\mathcal{T}(\mathbf{x}, \mathbf{v})).
\end{aligned}$$

■

Lemma 4.7 *If $\mathcal{T}(\mathbf{x}, \mathbf{v})$ is an implementation (i.e. for each $\mathbf{x}$ there exists a unique $\mathbf{v}$ such that $\mathcal{T}(\mathbf{x}, \mathbf{v}) = 1$), then $\mathcal{L}^{cm}(\mathbf{x}, \mathbf{u}) = \mathcal{L}^m(\mathbf{x}, \mathbf{u})$.*

Proof Since $\mathcal{T}$ is an implementation ($\mathbf{v} = g(\mathbf{x})$) $\mathcal{T} = \mathbf{v}g + \overline{\mathbf{v}g}$. Then

$$\begin{aligned}
\mathcal{S}_{\mathbf{v}}\mathcal{O}\mathcal{T} &= \mathcal{S}_{\mathbf{v}}\mathcal{O}(\mathbf{v}g + \overline{\mathbf{v}g}) \\
&= \mathcal{O}_{\mathbf{v}}g + \mathcal{O}_{\overline{\mathbf{v}}}\overline{g}.
\end{aligned}$$

In the same way,

$$\begin{aligned}
\mathcal{C}_{\mathbf{v}}(\overline{\mathcal{T}} + \mathcal{O}) &= \mathcal{O}_{\mathbf{v}}g + \mathcal{O}_{\overline{\mathbf{v}}}\overline{g} + \mathcal{O}_{\mathbf{v}}\mathcal{O}_{\overline{\mathbf{v}}} \\
&= \mathcal{O}_{\mathbf{v}}g + \mathcal{O}_{\overline{\mathbf{v}}}\overline{g}.
\end{aligned}$$

Therefore $\mathcal{L}^{cm} = \mathcal{L}^m$. ■

Example:

We find maximum and compatible observability relations for nodes u and v using the observability relation of the separating set u, v shown in Figure 4. These nodes have a single output which allows comparison with compatible observability don't cares obtained for the nodes. The observability relation for the circuit is $\mathcal{O}(\mathbf{z}, \mathbf{x}) = z_1\overline{\oplus}(x_1x_2 + x_3x_4)$. The observability relation for the separating set u, v is $\mathcal{O}(u, v, \mathbf{x}) = (u + v)\overline{\oplus}(x_1x_2 + x_3x_4)$. The current implementation at v is $\mathcal{T}(v, \mathbf{x}) = v\overline{\oplus}(x_3 + x_4)$. Thus,

$$\begin{aligned}
\mathcal{L}^m(u, \mathbf{x}) &= \mathcal{S}_v\mathcal{O}(u, v, \mathbf{x})\mathcal{T}(v, \mathbf{x} \\
&= \mathcal{S}_v(u + v)\overline{\oplus}(x_1x_2 + x_3x_4)v\overline{\oplus}(x_3 + x_4) \\
&= x_3x_4 + u\overline{\oplus}(x_1x_2).
\end{aligned}$$

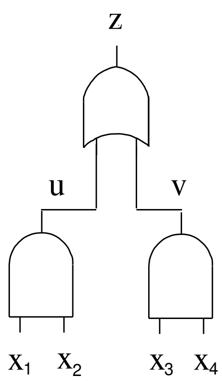

Figure 4: Example

The full ODC at node u from the observability relation at u is

$$\begin{aligned} ODC_u &= \mathcal{C}_u \mathcal{L}^m(u, \mathbf{x}) \\ &= \mathcal{C}_u(x_3 x_4 + u \overline{\oplus}(x_1 x_2)) \\ &= x_3 x_4 \end{aligned}$$

which is equal to what one gets with a direct computation of ODC's. Having computed $\mathcal{L}^m(u, \mathbf{x})$ we know find $\mathcal{T}^c(v, \mathbf{x})$ which is compatible to it.

$$\begin{aligned} \mathcal{T}^c(v, \mathbf{x}) &= \mathcal{C}_u(\mathcal{O}(u, v, \mathbf{x}) + \overline{\mathcal{L}}^m(u, \mathbf{x})) \\ &= \mathcal{C}_u((u+v)\overline{\oplus}(x_1 x_2 + x_3 x_4) + (\overline{x}_3 \overline{x}_4 (u \oplus (x_1 x_2))) \\ &= v \overline{\oplus}(x_3 x_4) + x_1 x_2 (\overline{x}_3 + \overline{x}_4). \end{aligned}$$

The CODC at v is

$$\begin{aligned} CODC_v &= \mathcal{C}_v(v \overline{\oplus}(x_3 x_4) + x_1 x_2 (\overline{x}_3 + \overline{x}_4)) \\ &= x_1 x_2 (\overline{x}_3 + \overline{x}_4) \end{aligned}$$

5 Compatible and Maximal Observability Relations for Multi-Output Nodes

Here we address the problem of finding *complete don't cares* for a multi-output node n and then simplifying it. This complete don't care set is effectively the observability relation for the node defined in the same way as the observability relation of a normal Boolean network. This can be either a local observability relation $\mathcal{O}^l(\mathbf{y}, \mathbf{u})$ where the $\mathbf{y}$ are the local fanins for the node and the $\mathbf{u}$ are the node outputs, or the global observability relation $\mathcal{O}^g(\mathbf{x}, \mathbf{u})$ giving the relation required between the primary inputs $\mathbf{x}$ and the node outputs $\mathbf{u}$.

Each multi-output node n has a set of inputs v which are unique to that node, a set of inputs y which are shared with other nodes, and a set of outputs u. The rest of the variables in the

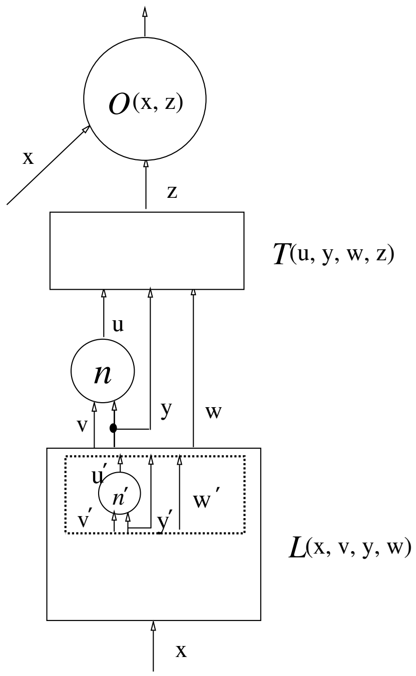

Figure 5: Observability Network for a Network of Multi-Output Nodes

two separating sets of variables before and after n which divide the network into two partitions are represented by $\mathbf{w}$ (a typical node is shown in Figure 5). The function at n is represented by $F(\mathbf{v}, \mathbf{y})$. The variables $(\mathbf{u}, \mathbf{y}, \mathbf{w})$ and $(\mathbf{v}, \mathbf{y}, \mathbf{w})$ represent separating sets of variables in the network just after and just before node n (see Figure 5). Note that the variables $\mathbf{y}$ which are fanins of n can be inputs to other nodes in the fanout network as shown in Figure 5. The characteristic function for the top part of the network $\mathcal{T}(\mathbf{u}, \mathbf{y}, \mathbf{w}, \mathbf{z})$ represents the present implementation after n, and the characteristic function for the lower part of the network $\mathcal{L}(\mathbf{x}, \mathbf{v}, \mathbf{y}, \mathbf{w})$, the implementation before n. $\mathcal{L}(\mathbf{x}, \mathbf{v}, \mathbf{y}, \mathbf{w})$ is also called the controllability function [6] for the separating set $(\mathbf{v}, \mathbf{y}, \mathbf{w})$, and gives the image computation from $\mathbf{x}$ to $(\mathbf{v}, \mathbf{y}, \mathbf{w})$.

Using the lemmas for parallel and serial decomposition of a Boolean relation, the observability relation for a multi-output node n shown in Figure 5 can be computed. Let $\mathcal{L}_1$ be the relation between $\mathbf{y}, \mathbf{w}$ and inputs $\mathbf{x}$ and $\mathcal{L}_2$ be the relation between $\mathbf{v}$ and inputs $\mathbf{x}$.

Lemma 5.1 **(compatible, global**) *Given the relations $\mathcal{L}_1(\mathbf{x}, \mathbf{y}, \mathbf{w})$ and $\mathcal{T}(\mathbf{u}, \mathbf{y}, \mathbf{w}, \mathbf{z})$, the compatible global observability relation for n is given by*

$$\mathcal{O}^{cg}(\mathbf{x}, \mathbf{u}) = \mathcal{C}_{\mathbf{y},\mathbf{w}}(\overline{\mathcal{L}_1}(\mathbf{x}, \mathbf{y}, \mathbf{w}) + \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{u}, \mathbf{y}, \mathbf{w}, \mathbf{z}) + \mathcal{O}(\mathbf{x}, \mathbf{z}))).$$

In particular, if $\mathcal{L}(\mathbf{x}, \mathbf{v}, \mathbf{y}, \mathbf{w}) = \mathcal{L}_1(\mathbf{x}, \mathbf{y}, \mathbf{w})\mathcal{L}_2(\mathbf{x}, \mathbf{v})$,

$$\mathcal{O}^{cg}(\mathbf{x}, \mathbf{u}) = \mathcal{C}_{\mathbf{y},\mathbf{w},\mathbf{v},\mathbf{z}}(\overline{\mathcal{L}}(\mathbf{x}, \mathbf{v}, \mathbf{y}, \mathbf{w}) + \overline{\mathcal{T}}(\mathbf{u}, \mathbf{y}, \mathbf{w}, \mathbf{z}) + \mathcal{O}(\mathbf{x}, \mathbf{z})).$$

Proof $\mathcal{O}'(\mathbf{x},\mathbf{u},\mathbf{y},\mathbf{w}) = \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z}) + \mathcal{O}(\mathbf{x},\mathbf{z})))$ is the compatible observability relation for the network at the separating set $(\mathbf{u},\mathbf{y},\mathbf{w})$ (serial decomposition, Lemma 4.1). $\mathcal{C}_{\mathbf{y},\mathbf{w}}(\overline{\mathcal{L}}_1 + \mathcal{O}')$ is the compatible parallel decomposition for node n (Lemma 4.5). If $\mathcal{L} = \mathcal{L}_1\mathcal{L}_2$, then $\mathcal{S}_{\mathbf{v}}\mathcal{L}_2(\mathbf{x},\mathbf{v}) = 1$ resulting in $\mathcal{C}_{\mathbf{v}}\overline{\mathcal{L}}_2(\mathbf{x},\mathbf{v}) = 0$. The global observability relation of n can be written as

$$\begin{aligned}
\mathcal{O}^{cg}(\mathbf{x},\mathbf{u}) &= \mathcal{C}_{\mathbf{y},\mathbf{w}}(\overline{\mathcal{L}}_1(\mathbf{x},\mathbf{y},\mathbf{w}) + \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z}) + \mathcal{O}(\mathbf{x},\mathbf{z})))\\
&= \mathcal{C}_{\mathbf{y},\mathbf{w}}(\overline{\mathcal{L}}_1(\mathbf{x},\mathbf{y},\mathbf{w}) + \mathcal{C}_{\mathbf{v}}\overline{\mathcal{L}}_2(\mathbf{x},\mathbf{v}) + \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z}) + \mathcal{O}(\mathbf{x},\mathbf{z})))\\
&= \mathcal{C}_{\mathbf{v},\mathbf{y},\mathbf{w}}(\overline{\mathcal{L}}_1(\mathbf{x},\mathbf{y},\mathbf{w}) + \overline{\mathcal{L}}_2(\mathbf{x},\mathbf{v}) + \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z}) + \mathcal{O}(\mathbf{x},\mathbf{z})))\\
&= \mathcal{C}_{\mathbf{v},\mathbf{y},\mathbf{w}}(\overline{\mathcal{L}}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w}) + \mathcal{C}_{\mathbf{z}}(\overline{\mathcal{T}}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z}) + \mathcal{O}(\mathbf{x},\mathbf{z})))\\
&= \mathcal{C}_{\mathbf{v},\mathbf{y},\mathbf{w},\mathbf{z}}(\overline{\mathcal{L}}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w}) + \overline{\mathcal{T}}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z}) + \mathcal{O}(\mathbf{x},\mathbf{z})).
\end{aligned}$$

■

We can also compute $\mathcal{O}^{mg}(\mathbf{x},\mathbf{u})$ which is maximal with respect to $\mathcal{L}_1$, $\mathcal{T}$, and consistent with $\mathcal{O}$.

Lemma 5.2 *(***maximal, global***) Given the relations $\mathcal{L}_1(\mathbf{x},\mathbf{y},\mathbf{w})$ and $\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})$, the maximal global observability relation for n is given by*

$$\mathcal{O}^{mg}(\mathbf{x},\mathbf{u}) = \mathcal{S}_{\mathbf{y},\mathbf{w},\mathbf{z}}\mathcal{L}_1(\mathbf{x},\mathbf{y},\mathbf{w})\mathcal{O}(\mathbf{x},\mathbf{z})\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z}).$$

In particular, if $\mathcal{L}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w}) = \mathcal{L}_1(\mathbf{x},\mathbf{y},\mathbf{w})\mathcal{L}_2(\mathbf{x},\mathbf{v})$ ($\mathcal{L}$ can be decomposed if it is an implementation), then

$$\mathcal{O}^{mg}(\mathbf{x},\mathbf{u}) = \mathcal{S}_{\mathbf{y},\mathbf{w},\mathbf{z}}\mathcal{L}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w})\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})\mathcal{O}(\mathbf{x},\mathbf{z}).$$

Proof $\mathcal{O}'(\mathbf{x},\mathbf{u},\mathbf{y},\mathbf{w}) = \mathcal{S}_{\mathbf{z}}\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})\mathcal{O}(\mathbf{x},\mathbf{z})$ is the maximal observability relation for the network at the separating set $(\mathbf{u},\mathbf{y},\mathbf{w})$ (serial decomposition, Lemma 4.2). $\mathcal{S}_{\mathbf{y},\mathbf{w}}(\mathcal{L}_1\mathcal{O}')$ is the maximal parallel decomposition for node n (Lemma 4.6). If $\mathcal{L} = \mathcal{L}_1\mathcal{L}_2$, then $\mathcal{S}_{\mathbf{v}}\mathcal{L}_2(\mathbf{x},\mathbf{v}) = 1$ because $\mathcal{L}_2$ must be well-defined. The global observability relation of n can be written as

$$\begin{aligned}
\mathcal{O}^{mg}(\mathbf{x},\mathbf{u}) &= \mathcal{S}_{\mathbf{y},\mathbf{w},\mathbf{z}}\mathcal{L}_1(\mathbf{x},\mathbf{y},\mathbf{w})\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})\mathcal{O}(\mathbf{x},\mathbf{z})\\
&= \mathcal{S}_{\mathbf{y},\mathbf{w},\mathbf{z}}\mathcal{L}_1(\mathbf{x},\mathbf{y},\mathbf{w})\mathcal{S}_{\mathbf{v}}(\mathcal{L}_2(\mathbf{x},\mathbf{v}))\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})\mathcal{O}(\mathbf{x},\mathbf{z})\\
&= \mathcal{S}_{\mathbf{v},\mathbf{y},\mathbf{w},\mathbf{z}}\mathcal{L}_1(\mathbf{x},\mathbf{y},\mathbf{w})\mathcal{L}_2(\mathbf{x},\mathbf{v})\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})\mathcal{O}(\mathbf{x},\mathbf{z})\\
&= \mathcal{S}_{\mathbf{v},\mathbf{y},\mathbf{w},\mathbf{z}}\mathcal{L}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w})\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})\mathcal{O}(\mathbf{x},\mathbf{z}).
\end{aligned}$$

■

If $\mathcal{L}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w})$ and $\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})$ are implementations, $\mathcal{O}^{mg}(\mathbf{x},\mathbf{u}) = \mathcal{O}^{cg}(\mathbf{x},\mathbf{u})$ because of Lemmas 4.3 and 4.7.

Let $\mathcal{L}_3(\mathbf{x},\mathbf{v},\mathbf{y})$ be the relation between $\mathbf{v},\mathbf{y}$ and inputs $\mathbf{x}$ and $\mathcal{L}_4(\mathbf{x},\mathbf{w})$ be the relation between $\mathbf{w}$ and inputs $\mathbf{x}$.

Lemma 5.3 *(***compatible, local***) Given the relations $\mathcal{L}_3(\mathbf{x},\mathbf{v},\mathbf{y})$ and $\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})$, the compatible local observability relation for n is given by*

$$\mathcal{O}^{cl}(\mathbf{v},\mathbf{y},\mathbf{u}) \;=\; \mathcal{C}_{\mathbf{x}}(\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})+\overline{\mathcal{L}}_3(\mathbf{x},\mathbf{v},\mathbf{y}))$$

where $\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})$ is given by 5.1. In particular, if $\mathcal{L}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w}) = \mathcal{L}_3(\mathbf{x},\mathbf{v},\mathbf{y})\mathcal{L}_4(\mathbf{x},\mathbf{w})$ ($\mathcal{S}_{\mathbf{w}}\mathcal{L}_4(\mathbf{x},\mathbf{w}) = 1$ if $\mathcal{L}_4(\mathbf{x},\mathbf{w})$ is well-defined),

$$\mathcal{O}^{cl}(\mathbf{v},\mathbf{y},\mathbf{u}) \;=\; \mathcal{C}_{\mathbf{x},\mathbf{w}}(\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})+\overline{\mathcal{L}}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w})).$$

Proof $\mathcal{O}^{cl}(\mathbf{v},\mathbf{y},\mathbf{u}) = \mathcal{C}_{\mathbf{x}}(\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})+\overline{\mathcal{L}}_3(\mathbf{x},\mathbf{v},\mathbf{y}))$ is simply the compatible parallel decomposition made local by quantifying away $\mathbf{x}$. If $\mathcal{L} = \mathcal{L}_3\mathcal{L}_4$ and $\mathcal{C}_{\mathbf{w}}\overline{\mathcal{L}}_4(\mathbf{x},\mathbf{w}) = 0$, then the local compatible observability relation for n is

$$\begin{aligned}
\mathcal{O}^{cl}(\mathbf{v},\mathbf{y},\mathbf{u}) &= \mathcal{C}_{\mathbf{x}}(\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})+\overline{\mathcal{L}}_3(\mathbf{x},\mathbf{v},\mathbf{y}))\\
&= \mathcal{C}_{\mathbf{x}}(\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})+\overline{\mathcal{L}}_3(\mathbf{x},\mathbf{v},\mathbf{y})+\mathcal{C}_{\mathbf{w}}\overline{\mathcal{L}}_4(\mathbf{x},\mathbf{w})\\
&= \mathcal{C}_{\mathbf{x},\mathbf{w}}(\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})+\overline{\mathcal{L}}_3(\mathbf{x},\mathbf{v},\mathbf{y})+\overline{\mathcal{L}}_4(\mathbf{x},\mathbf{w})\\
&= \mathcal{C}_{\mathbf{x},\mathbf{w}}(\mathcal{O}^{cg}(\mathbf{x},\mathbf{u})+\overline{\mathcal{L}}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w})).
\end{aligned}$$

■

As is the case in equation 2, we do not know of a way to express $\mathcal{O}^{ml}(\mathbf{v},\mathbf{y},\mathbf{u})$ as a Boolean relation. However, if both $\mathcal{T}(\mathbf{u},\mathbf{y},\mathbf{w},\mathbf{z})$ and $\mathcal{L}(\mathbf{x},\mathbf{v},\mathbf{y},\mathbf{w})$ are implementations, $\mathcal{O}^{cl}(\mathbf{v},\mathbf{y},\mathbf{u})$ is also maximal.

$\mathcal{O}^{cl}(\mathbf{v},\mathbf{y},\mathbf{u})$ is a Boolean relation and can be minimized as a two-level function using the program GYOCRO [11]. The result can then be used to replace the present implementation for n, $\mathbf{u} = F(\mathbf{y})$. Alternately, if n is a multi-level network itself, compatible external don't cares can be derived as in [8] for each output of n and *full_simplify* [10] can be used to optimize n.

5.1 Node Optimization Using Maximal Observability Relations

The algorithm in Figure 6 shows the computation of the maximal observability relation for each multi-output node n_i of a Boolean network and the simplification of the node using GYOCRO. We first order all the nodes in topological order from outputs. $\mathcal{O}_0(\mathbf{x},\mathbf{z})$ is the given specification of the circuit. The inputs of n_i are $\mathbf{v}_i$ and $\mathbf{y}_i$ and its outputs are $\mathbf{u}_i$. $\mathbf{w}_i$ are the rest of variables shared by the separating sets of variables before and after n_i. For each node n_i, the observability relation $\mathcal{O}_i(\mathbf{x},\mathbf{u}_i,\mathbf{y}_i,\mathbf{w}_i)$ is computed from the old relation $\mathcal{O}_{i-1}(\mathbf{x},\mathbf{u}_{i-1},\mathbf{y}_{i-1},\mathbf{w}_{i-1})$ and the new implementation of n_{i-1} after its optimization

$$\begin{aligned}
\tilde{\mathcal{O}}_i(\mathbf{x},\mathbf{v}_{i-1},\mathbf{y}_{i-1},\mathbf{w}_{i-1}) &= \mathcal{S}_{\mathbf{u}_{i-1}}\mathcal{O}_{i-1}(\mathbf{x},\mathbf{u}_{i-1},\mathbf{y}_{i-1},\mathbf{w}_{i-1})(\mathbf{u}_{i-1}\equiv F_{i-1}(\mathbf{v}_{i-1},\mathbf{y}_{i-1}))\\
\mathcal{O}_i(\mathbf{x},\mathbf{u}_i,\mathbf{y}_i,\mathbf{w}_i) &= \tilde{\mathcal{O}}_i(\mathbf{x},\mathbf{v}_{i-1},\mathbf{y}_{i-1},\mathbf{w}_{i-1}).
\end{aligned}$$

The variables $\mathbf{v}_{i-1},\mathbf{y}_{i-1},\mathbf{w}_{i-1}$ are regrouped to form $\mathbf{u}_i,\mathbf{y}_i,\mathbf{w}_i$ (the same as $\mathbf{v},\mathbf{y},\mathbf{w}$ and $\mathbf{u}',\mathbf{y}',\mathbf{w}'$ in Figure 5). $\mathcal{L}_i(\mathbf{x},\mathbf{v}_i,\mathbf{y}_i,\mathbf{w}_i)$ is computed using the global functions g's at each of the inputs of n_i and the ones corresponding to $\mathbf{w}_i$. $\mathcal{O}_i^{mg}(\mathbf{x},\mathbf{u}_i)$ and $\mathcal{O}_i^{ml}(\mathbf{v}_i,\mathbf{y}_i,\mathbf{u}_i)$ are then

function node_simplify
/* multi-output node simplification using maximal observability relations */
begin

$\mathcal{O}_0(\mathbf{x}, \mathbf{z})$ = given specification for the circuit

NodeArray = nodes ordered in topological order from the outputs

for each node n_i in NodeArray in topological order **begin**

$\mathcal{O}_i(\mathbf{x}, \mathbf{u}_i, \mathbf{y}_i, \mathbf{w}_i) = \mathcal{S}_{\mathbf{u}_{i-1}} \mathcal{O}_{i-1}(\mathbf{x}, \mathbf{u}_{i-1}, \mathbf{y}_{i-1}, \mathbf{w}_{i-1})(\mathbf{u}_{i-1} \equiv F_{i-1}(\mathbf{v}_{i-1}, \mathbf{y}_{i-1}))$

$\mathcal{L}_i(\mathbf{x}, \mathbf{v}_i, \mathbf{y}_i, \mathbf{w}_i) = \prod_{k=1}^{|\mathbf{v}_i|}(g_{v_k} \overline{\oplus} v_k) \prod_{k=1}^{|\mathbf{y}_i|}(g_{y_k} \overline{\oplus} y_k) \prod_{k=1}^{|\mathbf{w}_i|}(g_{w_k} \overline{\oplus} w_k)$

$\mathcal{O}_i^{mg}(\mathbf{x}, \mathbf{u}_i) = \mathcal{S}_{\mathbf{y}_i, \mathbf{w}_i} \mathcal{L}_i(\mathbf{x}, \mathbf{v}_i, \mathbf{y}_i, \mathbf{w}_i) \mathcal{O}_i(\mathbf{x}, \mathbf{u}_i, \mathbf{y}_i, \mathbf{w}_i)$

$\mathcal{O}_i^{ml}(\mathbf{v}_i, \mathbf{y}_i, \mathbf{u}_i) = \mathcal{C}_{\mathbf{x}, \mathbf{w}_i}(\mathcal{O}_i^{mg}(\mathbf{x}, \mathbf{u}_i) + \overline{\mathcal{L}}_i(\mathbf{x}, \mathbf{v}_i, \mathbf{y}_i, \mathbf{w}_i))$

Optimize n_i with the relation $\mathcal{O}_i^l$ using GYOCRO

end

end

Figure 6: Maximal Observability Relation Computation and Node Simplification

computed and used to improve the current implementation at n_i. Lemma 5.3 is used to compute $\mathcal{O}_i^{ml}(\mathbf{v}_i, \mathbf{y}_i, \mathbf{u}_i)$ from $\mathcal{O}_i^{mg}(\mathbf{x}, \mathbf{u}_i)$. $\mathcal{O}_i^{ml}(\mathbf{v}_i, \mathbf{y}_i, \mathbf{u}_i)$ is maximal because $\mathcal{L}_i(\mathbf{x}, \mathbf{v}_i, \mathbf{y}_i, \mathbf{w}_i)$ is an implemetation.

5.2 Node Optimization Using Compatible Observability Relations

The algorithm in Figure 7 shows the computation of compatible observability relations for multi-output nodes of a Boolean network and the simplification of the nodes using GYOCRO. The nodes are ordered topologically from outputs as before. $\mathcal{O}_0(\mathbf{x}, \mathbf{z})$ is the given specification of the circuit. For each node n_i, the observability relation $\mathcal{O}_i(\mathbf{x}, \mathbf{u}_i, \mathbf{y}_i, \mathbf{w}_i)$ is computed from the old relation $\mathcal{O}_{i-1}(\mathbf{x}, \mathbf{u}_{i-1}, \mathbf{y}_{i-1}, \mathbf{w}_{i-1})$ and $\mathcal{O}_{i-1}^{cl}(\mathbf{v}_{i-1}, \mathbf{y}_{i-1}, \mathbf{u}_{i-1})$. This is then used to find $\mathcal{O}_i^{cg}(\mathbf{x}, \mathbf{u}_i)$ and $\mathcal{O}_i^{cl}(\mathbf{v}_i, \mathbf{y}_i, \mathbf{u}_i)$ as shown. Each node n_i can be optimized independent of the others using its observability relation because the observability relations are compatible.

6 Conclusion

In this paper, we have expanded the theory of don't cares which is for single-output nodes to a theory that can be applied to multi-output nodes. The flexibility at each node is represented by a Boolean relation. Techniques are provided for computing both maximal and compatible observability relations for multi-output nodes of a Boolean network as is the case with don't cares for single-output nodes. The maximal observability relation for a multi-output node of a combinational circuit is the maximum flexibility for manipulating that node. The compatible observability relations for a set of nodes ordered topologically allows optimization of each such node independent of the optimization done at other nodes in the set. The practicality of

function node_simplify
/* multi-output node simplification using compatible observability relations */
begin

$\mathcal{O}_0(\mathbf{x}, \mathbf{z})$ = given specification for the circuit

NodeArray = nodes ordered in topological order from the outputs

for each node n_i in NodeArray in topological order **begin**

$$\mathcal{O}_i(\mathbf{x}, \mathbf{u}_i, \mathbf{y}_i, \mathbf{w}_i) = \mathcal{C}_{\mathbf{u}_{i-1}}(\mathcal{O}_{i-1}(\mathbf{x}, \mathbf{u}_{i-1}, \mathbf{y}_{i-1}, \mathbf{w}_{i-1}) + \overline{\mathcal{O}}_{i-1}^{cl}(\mathbf{v}_{i-1}, \mathbf{y}_{i-1}, \mathbf{u}_{i-1}))$$
$$\mathcal{L}_i(\mathbf{x}, \mathbf{v}_i, \mathbf{y}_i, \mathbf{w}_i) = \prod_{k=1}^{|\mathbf{v}_i|}(g_{v_k} \overline{\oplus} v_k) \prod_{k=1}^{|\mathbf{y}_i|}(g_{y_k} \overline{\oplus} y_k) \prod_{k=1}^{|\mathbf{w}_i|}(g_{w_k} \overline{\oplus} w_k)$$
$$\mathcal{O}_i^{cg}(\mathbf{x}, \mathbf{u}_i) = \mathcal{C}_{\mathbf{y}_i, \mathbf{w}_i}(\overline{\mathcal{L}}_i(\mathbf{x}, \mathbf{v}_i, \mathbf{y}_i, \mathbf{w}_i) + \mathcal{O}_i(\mathbf{x}, \mathbf{u}_i, \mathbf{y}_i, \mathbf{w}_i))$$
$$\mathcal{O}_i^{cl}(\mathbf{v}_i, \mathbf{y}_i, \mathbf{u}_i) = \mathcal{C}_{\mathbf{x}, \mathbf{w}_i}(\mathcal{O}_i^{cg}(\mathbf{x}, \mathbf{u}_i) + \overline{\mathcal{L}}_i(\mathbf{x}, \mathbf{v}_i, \mathbf{y}_i, \mathbf{w}_i))$$

end

Nodes can be optimized using GYOCRO independent of each other

end

Figure 7: Compatible Observability Relation Computation and Node Simplification

these techniques for optimizing large circuits depends on how efficiently one can represent and manipulate observability relations in BDD or any other form. We are currently investigating such techniques.

References

[1] K. Bartlett, R. K. Brayton, G. D. Hachtel, R. M. Jacoby, C. R. Morrison a nd R. L. Rudell, A. Sangiovanni-Vincentelli, and A. R. Wang. Multi-level Logic Minimization Using Implicit Don't Cares. In *IEEE Transactions on CAD*, pages 723–740, June 1988.

[2] R. K. Brayton, G. D. Hachtel, C.T. McMullen, and A.L. Sangiovanni-Vincentelli. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers, 1984.

[3] R. K. Brayton and F. Somenzi. Boolean Relations and the Incomplete Specification of Logic Networks. In *VLSI'89*, August 1989.

[4] R. E. Bryant. Graph Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.

[5] E. Cerny. An approach to unified methodology of combinational switching circuits. *IEEE Transactions on Computers*, 27(8), 1977.

[6] E. Cerny and C. Mauras. Tautology Checking Using Cross-Controllability and Cross-Observability Relations. In *IEEE International Conference on Computer-Aided Design*, pages 34–37, November 1990.

[7] S. Muroga, Y. Kambayashi, H. C. Lai, and J. N. Culliney. The Transduction Method - Design of Logic Networks Based on Permissible Functions. In *IEEE Transactions on Computers*, October 1989.

[8] H. Savoj and R. Brayton. The Use of Observability and External Don't Cares for the Simplification of Multi-Level Networks. In *27th ACM/IEEE Design Automation Conference*, Orlando, June 1990.

[9] H. Savoj and R. K. Brayton. Observability Relations and Observability Don't Cares. In *IEEE International Conference on Computer-Aided Design*, November 1991.

[10] H. Savoj, H. Touati, and R. K. Brayton. Extracting Local Don't Cares for Network Optimization. In *IEEE International Conference on Computer-Aided Design*, November 1991.

[11] Y. Watanabe and R.K. Brayton. Heuristic Minimization of Boolean Relations. In *International Workshop on Logic Synthesis*, May 1991.