Efficient Computation of Exact and Simplified Observability Don't care Sets for Multiple-Level Combinational Networks

Maurizio Damiani *and* Giovanni De Micheli

Technical Report: CSL - TR - 90 - 424

April 1990

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, CA 94305-4055

Abstract

This work presents new algorithms for the extraction of exact and approximate Observability Don't Care sets (ODC sets) in a combinational multiple-level logic network. The proposed algorithms are efficient because they use local information, *i.e.* the computation of the ODC for a vertex in the network requires only the knowledge of the *don't care* sets at the adjacent vertices.

Two approaches are proposed. The former computes the exact ODC sets. The latter computes ODC subsets and can be used when partial information on the *don't care* sets is available. The algorithms for the simplified computation finds the largest subsets of the actual ODC sets, given the information available at the adjacent vertices.

Key Words and Phrases: Combinational Logic Synthesis, Logic Minimization, Boolean Networks, Don't Care sets, Testing

Efficient Computation of Exact and Simplified Observability Don't care Sets for Multiple-Level Combinational Networks

1 Introduction

Over the past few years, the problem of computing efficiently and correctly the Observability Don't Care (ODC) sets has emerged as a central one in the synthesis of combinational networks [1], [2], [3]. The knowledge of the ODC sets is important in several respects, namely: 1) local minimization of functions in a Boolean network, 2) synthesis of 100% testable networks, 3) test pattern generation [4].

Bartlett proposed in [1] a computation of the ODC sets requiring the representation of the primary output expression in terms of the network intermediate variables. Such a representation may be subject to the explosion in size of the representation. Muroga proposed in [3] exhaustive simulations of the circuit for determining the observability *don't care* sets of vertices with reconvergent fanout.

Other authors pointed out that it would be desirable and computationally much more efficient to derive the ODC set of a vertex of a Boolean network from the ODC sets of its direct fanout vertices [5], [7]. They showed, however, that a straightforward application of this idea could lead to erroneous results, because of the effect of reconvergent fanout. Therefore only approximate solutions have been proposed to compute the ODC sets [7], [8].

In this paper we show how an exact computation of ODC sets is indeed possible by using only local information. We propose an algorithm for such a computation that traverses the network backward from the primary outputs to the primary inputs processing each vertex only once.

Unfortunately this method, leading to an exact computation of the ODC sets, involves implicitly Boolean complementations and it is consequently prone to the well-known phenomenon of "combinational explosion". We therefore propose two other algorithms for the computation of subsets of the actual ODC sets, still based on local information. The first algorithm computes an ODC subset at each vertex of the Boolean network from the ODC subsets of its direct fanout vertices. The second one computes *both* subsets of the actual *care* and *don't care* sets from those of its fanout. It will be shown that the subsets computed by the second algorithm include those computed by the first one, thus providing a better approximation.

2 Definitions and Notations

2.1 Combinational Boolean Networks.

In this paper we model multiple-output combinational circuits by Boolean networks [1]. A Boolean network N with n input vertices and m output vertices realizes a function $\underline{\mathcal{F}} : B^n \rightarrow B^m$ [1], where B is the Boolean set $\{0, 1\}$. Underlining is used for denoting vector quantities throughout this paper.

The network is specified by its acyclic network graph $G = (V, E)$. The elements of the vertex set $V = V^I \cup V^G \cup V^O = \{v\}$ are in one-to-one correspondence with primary inputs, logic gates, and primary outputs, respectively. There is a directed edge e from a vertex μ to a vertex ν if the output of the gate in μ is connected to

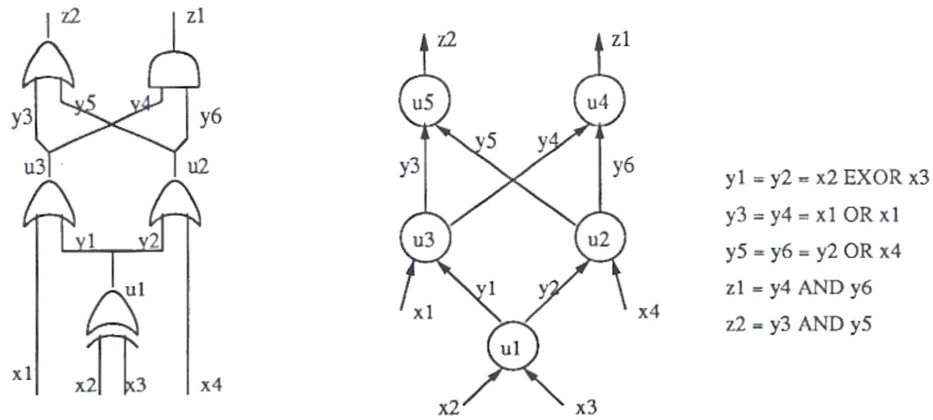

Figure 1: A multiple-level combinational circuit and its associated Boolean network representation.

an input of the gate in ν. In contrast to [1], and similarly to [3], we associate a network variable y_i to each edge $e_i \in E$. An example of a circuit and of its associated Boolean network is shown in Fig. (1).

A network variable y_i is said to be a fanout (fanin) variable of a vertex $\nu \in V$ if e_i is an edge whose tail (head) end-point is ν. We associate to each vertex ν an expression $f_\nu(y_1, y_2, \cdots, y_n)$ of its fanin edge variables. The expression f_ν, describing the functionality of the gate in ν, specifies all the fanout variables of ν in terms of the fanin variables.

2.2 Observability *don't care* sets.

By cutting an edge e_i and by considering y_i as a primary input variable, the new network N^c realizes a function $\underline{\mathcal{F}}^c(\underline{x}, y) : B^{n+1} \rightarrow B^m$. Given a primary input assignment $\underline{x}_0$, the variable y_i of N is not observable if the *vector* equality [6]

$$\underline{\mathcal{F}}^c(\underline{x}_0, 0) \;=\; \underline{\mathcal{F}}^c(\underline{x}_0, 1). \tag{1}$$

is satisfied. Recall that the cofactor $f|_{y_i}$ of a function f with respect to y_i is the function obtained by assuming $y_i = 1$. Similarly, $f|_{\overline{y}_i}$ is the function obtained by setting $y_i = 0$. The *vector* function:

$$\underline{ODC}_{y_i} \;=\; \underline{\mathcal{F}}^c|_{\overline{y}_i} \,\overline{\oplus}\, \underline{\mathcal{F}}^c|_{y_i} \;\stackrel{\text{def}}{=}\; \overline{\left(\frac{\partial \underline{\mathcal{F}}}{\partial y_i}\right)}. \tag{2}$$

therefore describes the observability of y_i. In particular, y_i will be observable at the k^{th} network output ($k = 1, \cdots, m$) if the k^{th} component of $\underline{ODC}_{y_i}$ is 0. The quantity $\partial\underline{\mathcal{F}}/\partial y_i$ is usually termed *Boolean difference* [6] of $\underline{\mathcal{F}}$ with respect to the (possibly internal) variable y_i.

The observability *care* set of y_i ($\underline{OC}_{y_i}$) is defined by the complement of $\underline{ODC}_{y_i}$, *i.e.* by $\overline{\underline{ODC}_{y_i}}$. Its k^{th} component describes the network configurations that make the variable y_i observable at the k^{th} output.

Given a network, it is possible in principle to compute $\underline{ODC}_y$ for any internal variable y by flattening the network N^c and applying Eq. (2). We show here that it is possible to avoid the flattening operation on the network and compute exact and approximate versions of the functions $\underline{ODC}_y$ with a single traversal of the network.

For example, if the *don't care* set of a vertex ν is known, then it is easy to obtain an expression of $\underline{ODC}_y$ for any fanin variable y of ν from [1]:

$$\underline{ODC}_y = \underline{ODC}_\nu + (1, 1, \quad, 1)^T \overline{\left(\frac{\partial f_\nu}{\partial y}\right)}. \tag{3}$$

[1]Notice that Eq. (3) may contain internal network variables. These, however, may be resolved by back-substitution to obtain an expression of primary input variables only, so that there is no real contrast with the definition (2).

The vector $(1, 1, \cdots, 1)^T$ is used to add $\partial f_\nu / \partial y$ to all the components of the vector $\underline{ODC}_y$.

If the network has a tree structure, then it is possible to obtain all the ODC sets by traversing backwards the network and applying iteratively Eq. (3). If a vertex has reconvergent fanout, however, the observability conditions of the vertex do not coincide with those of its fanout variables. We present here how the observability *don't care* of a vertex can be derived from those of its fanout variables.

2.3 Observability *don't care* sets in presence of reconvergent fanout

Consider a vertex ν with reconvergent fanout, and suppose that all the edges $e_1, e_2, \cdots, e_n$, whose tail-end point is ν, are cut. Let $\underline{\mathcal{F}}^c(\underline{x}, y_1, y_2, \cdots, y_n)$ be the function realized by the new network N^c, obtained by adding the variables $y_1, \cdots, y_n$, corresponding to the cut edges, to the primary inputs. Then, the observability *don't care* set of ν is described by the function

$$\underline{ODC}_\nu = \underline{\mathcal{F}}^c(\underline{x}, 0, \quad, 0) \overline{\oplus} \underline{\mathcal{F}}^c(\underline{x}, 1, \quad 1) \tag{4}$$

or, equivalently, by

$$\underline{ODC}_\nu = \underline{\mathcal{F}}^c|_{\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_n} \overline{\oplus} \underline{\mathcal{F}}^c|_{y_1, y_2, \cdots, y_n}. \tag{5}$$

For the sake of simplicity, we describe first the case in which $n = 2$, so that there are only two fanout variables, y_1 and y_2. We will generalize the result down below.

The observability *don't care* set is described by the function

$$\underline{ODC}_\nu = \underline{\mathcal{F}}^c|_{\bar{y}_1, \bar{y}_2} \overline{\oplus} \underline{\mathcal{F}}^c|_{y_1, y_2} \tag{6}$$

By manipulating Eq. (6), $\underline{ODC}_\nu$ can be rewritten as

$$\underline{ODC}_\nu = \left(\underline{\mathcal{F}}^c|_{\bar{y}_1, \bar{y}_2} \overline{\oplus} \underline{\mathcal{F}}^c|_{y_1, \bar{y}_2} \right) \overline{\oplus} \left(\underline{\mathcal{F}}^c|_{y_1, \bar{y}_2} \overline{\oplus} \underline{\mathcal{F}}^c|_{y_1, y_2} \right) \tag{7}$$

where the term $\underline{\mathcal{F}}^c|_{y_1, \bar{y}_2}$ has been "added and subtracted" in Eq. (6).

From Eq.(2), the first term within parentheses is $\underline{ODC}_{y_1}|_{\bar{y}_2}$, while the second parentheses describe $\underline{ODC}_{y_2}|_{y_1}$ It then follows that

$$\underline{ODC}_\nu = \underline{ODC}_{y_1}|_{\bar{y}_2} \overline{\oplus} \underline{ODC}_{y_2}|_{y_1} \tag{8}$$

Notice that in Eq. (5), $\underline{ODC}_\nu$ can be also rewritten as

$$\underline{ODC}_\nu = \left(\underline{\mathcal{F}}^c|_{\bar{y}_1, \bar{y}_2} \overline{\oplus} \underline{\mathcal{F}}^c|_{\bar{y}_1, y_2} \right) \overline{\oplus} \left(\underline{\mathcal{F}}^c|_{\bar{y}_1, y_2} \overline{\oplus} \underline{\mathcal{F}}^c|_{y_1, y_2} \right) = \underline{ODC}_{y_2}|_{\bar{y}_1} \overline{\oplus} \underline{ODC}_{y_1}|_{y_2} \tag{9}$$

from which we obtain the identity

$$\underline{ODC}_\nu = \underline{ODC}_{y_1}|_{\bar{y}_2} \overline{\oplus} \underline{ODC}_{y_2}|_{y_1} = \underline{ODC}_{y_1}|_{y_2} \overline{\oplus} \underline{ODC}_{y_2}|_{\bar{y}_1} \tag{10a}$$

Similarly, for the *care* set we obtain

$$\underline{OC}_\nu = \underline{OC}_{y_1}|_{\bar{y}_2} \oplus \underline{OC}_{y_2}|_{y_1} = \underline{OC}_{y_1}|_{y_2} \oplus \underline{OC}_{y_2}|_{\bar{y}_1}. \tag{10b}$$

These identities will be used in Sect. (3), when deriving approximations to $\underline{ODC}_\nu$.

The algebraic manipulation above can be extended to the general case of n fanout variables as follows. It can be easily verified that for $n \geq 2$ the following identity can be derived from Eq. (5):

$$\underline{ODC}_y = \left(\underline{\mathcal{F}}^c|_{\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_n} \overline{\oplus} \underline{\mathcal{F}}^c|_{y_1, \bar{y}_2, \quad, \bar{y}_n} \right) \overline{\oplus}$$

$$\left(\mathcal{F}^c|_{y_1,\overline{y}_2,\cdots,\overline{y}_n} \overline{\oplus} \mathcal{F}^c|_{y_1,y_2,\cdots,\overline{y}_n}\right) \overline{\oplus} \quad \overline{\oplus}\left(\mathcal{F}^c|_{y_1,y_2,\cdots,y_{n-1},\overline{y}_n} \overline{\oplus} \mathcal{F}^c|_{y_1,y_2,\cdots,y_{n-1},y_n}\right)$$

This can be rewritten as:

$$\underline{ODC_\nu} = \overline{\bigoplus}_{i=1}^{n} \mathcal{F}^c|_{y_1,\cdots,y_{i-1},\overline{y}_i,\cdots,\overline{y}_n} \overline{\oplus} \mathcal{F}^c|_{y_1,\cdots,y_i,\cdots,\overline{y}_{i+1},\cdots,\overline{y}_n}$$

From Eq. (2), each term of the sum in Eq. (11) is easily recognized to be $\underline{ODC_{y_i}}|_{y_1,\cdots,y_{i-1},\overline{y}_{i+1},\cdots,\overline{y}_n}$. Eq. (12) can be thus rewritten as

$$\underline{ODC_\nu} = \overline{\bigoplus}_{i=1}^{n} \underline{ODC_{y_i}}|_{y_1,\cdots,y_{i-1},\overline{y}_{i+1},\cdots,\overline{y}_n} \tag{13}$$

For the *care* set, we have:

$$\underline{OC_\nu} = \bigoplus_{i=1}^{n} \underline{OC_{y_i}}|_{y_1,\cdots,y_{i-1},\overline{y}_{i+1},\cdots,\overline{y}_n}. \tag{14}$$

Similarly to the case of 2 fanout variables, changing the order in which the fanout variables are complemented results in different expressions of $\underline{OC_\nu}$. For n fanout variables, there are $n!$ possible orderings, hence $n!$ different possible expressions for $\underline{ODC_\nu}$. In particular, let $(i_1, i_2, \cdots, i_n)$ denote a permutation of $(1, 2, \cdots, n)$. Then, the following identities hold:

$$\underline{ODC_\nu} = \overline{\bigoplus}_{j=1}^{n} \underline{ODC_{y_{i_j}}}|_{y_{i_1},\cdots,y_{i_{j-1}},\overline{y}_{i_{j+1}},\cdots,\overline{y}_{i_n}}.$$

$$\underline{OC_\nu} = \bigoplus_{j=1}^{n} \underline{OC_{y_{i_j}}}|_{y_{i_1},\cdots,y_{i_{j-1}},\overline{y}_{i_{j+1}},\cdots,\overline{y}_{i_n}}$$

2.4 An exact algorithm for the ODC sets computation

Given the ODC set of a vertex ν, it is possible to compute the ODC set of all its fanin edges by means of Eq. (3). In turn, Eq. (13) allows us to compute a vertex ODC set, given those of its fanout variables. It is thus now possible to visit the Boolean network backwards from the primary outputs to its inputs and to determine the ODC sets of each vertex also in presence of reconvergent fanout.

The following algorithm performs the computation of the ODC sets. It uses the subset S of the vertices whose ODC set is known. Initially S is the set of primary output vertices with empty fanout set.

```
OBSERVABILITY(G);
S := {primary output vertices with empty fanout set};
while ( S ≠ V ) {
        select ν ∈ {V − S} such that FO(ν) ⊆ S;
        foreach fanout variable y_i of ν {
                μ = head vertex of edge e_i
                ODC_{y_i} = ODC_ν + ‾(∂f_μ/∂y_i)          /* compute ODC_y by Eq. (3) */
        }
        ODC_ν = ‾⊕_{i=1}^n ODC_{y_i}|_{y_1,…,y_{i-1},‾y_{i+1}…‾y_n}   /* compute ODC_ν by Eq. (13) */
        S := S ∪ {ν};
}
```

The algorithm is linear in the number of edges. We illustrate here its operation on the circuit shown in Fig. (1). At the beginning, $S = \{u_4, u_5\}$, and

$$\underline{ODC}_{u_4} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} ; \underline{ODC}_{u_5} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

The first and second component of the vectors $\underline{ODC}$ describe the observability with respect to z_1 and z_2, respectively.

First the vertices u_2, u_3 are considered. By applying Eqs. (3) and (13)

$$\underline{ODC}_{u_3} = \underline{ODC}_{y_3}|_{\overline{y}_4} \overline{\oplus} \underline{ODC}_{y_4}|_{y_3} = \begin{pmatrix} 1 \\ y_5 \end{pmatrix} \overline{\oplus} \begin{pmatrix} \overline{y}_6 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{y}_6 \\ y_5 \end{pmatrix}$$

$$\underline{ODC}_{u_2} = \underline{ODC}_{y_6}|_{\overline{y}_5} \overline{\oplus} \underline{ODC}_{y_5}|_{y_6} = \begin{pmatrix} \overline{y}_4 \\ 1 \end{pmatrix} \overline{\oplus} \begin{pmatrix} 1 \\ y_3 \end{pmatrix} = \begin{pmatrix} \overline{y}_4 \\ y_3 \end{pmatrix}$$

Then $S = \{u_2, u_3, u_4, u_5\}$ and u_1 is selected. Its fanout variables are y_1, y_2, and according to Eq. (3) :

$$\underline{ODC}_{y_1} = \begin{pmatrix} \overline{y}_6 + x_1 \\ y_5 + x_1 \end{pmatrix} = \begin{pmatrix} \overline{y}_2 \overline{x}_4 + x_1 \\ y_2 + x_4 + x_1 \end{pmatrix}$$

$$\underline{ODC}_{y_2} = \begin{pmatrix} \overline{y}_4 + x_4 \\ y_3 + x_4 \end{pmatrix} = \begin{pmatrix} \overline{y}_1 \overline{x}_1 + x_4 \\ y_1 + x_1 + x_4 \end{pmatrix}$$

so that, using Eq. (13),

$$\underline{ODC}_{u_1} = \underline{ODC}_{y_1}|_{\overline{y}_2} \overline{\oplus} \underline{ODC}_{y_2}|_{y_1} = \begin{pmatrix} \overline{x}_4 + x_1 \\ x_4 + x_1 \end{pmatrix} \overline{\oplus} \begin{pmatrix} x_4 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1 x_4 \\ x_1 + x_4 \end{pmatrix}$$

which is the correct result.

Finally the algorithm computes the ODC sets of the primary inputs, that may be used as external ODC sets for the minimization of the logic feeding the circuit.

The product of all the components of $\underline{ODC}_\nu$ gives the conditions for which the gate output at vertex ν is not observable at *any* output. For example, in the case of the gate in u_1, the product of the components of $\underline{ODC}_{u_1}$ yields $x_1 x_4$.

3 Computation of approximate ODC sets

It is of practical interest to consider the case in which the ODC sets are approximated by subsets, because of their size. Note that excess approximations of the ODC sets are of no practical value for Boolean minimization. Therefore it is useful to derive ODC subsets at a vertex from the ODC sets of its fanout vertices. Unfortunately, Eq. (7) may not yield an ODC subset from subsets of the actual ODC sets of each edge variable.

For example, in the circuit of Fig. (1), if we assume

$$\underline{ODC}_{y_1} = \begin{pmatrix} x_1 \\ x_1 \end{pmatrix}$$

(actually a subset of the true $\underline{ODC}_{y_1}$), we find for $\underline{ODC}_{u_1}$ the estimate:

$$\underline{ODC}_{u_1} = \begin{pmatrix} x_1 x_4 + \overline{x}_1 \overline{x}_4 \\ x_1 \end{pmatrix}$$

which is no longer a subset of the true $\underline{ODC}_{u_1}$.

Two different approaches for the computation of ODC subsets are considered here. In particular we show two formulae that can replace Eq. (13) in the previous algorithm. The first formula computes *both* subsets of the actual *care* and *don't care* sets of a vertex from those of its fanout variables. The second one, simpler but less accurate, computes ODC subsets only.

We give first formulae for computing ODC_ν in the case of two fanout variables. We then present their generalization to larger fanout.

3.1 Propagation of *care* and *don't care* subsets

Let y_1 and y_2 denote the variables corresponding to the edges in the fanout of a vertex ν. Assume that only subsets $\underline{\widetilde{ODC}}_{y_i}$, $\underline{\widetilde{OC}}_{y_i}$ of $\underline{ODC}_{y_i}, \underline{OC}_{y_i}$ are available. Then, $\underline{\widetilde{ODC}}_\nu$ and $\underline{\widetilde{OC}}_\nu$ computed by:

$$\begin{aligned}\underline{\widetilde{ODC}}_\nu = {} & \underline{\widetilde{ODC}}_{y_1}|_{\overline{y}_2}\underline{\widetilde{ODC}}_{y_2}|_{y_1} + \underline{\widetilde{ODC}}_{y_1}|_{y_2}\underline{\widetilde{ODC}}_{y_2}|_{\overline{y}_1} + \\ & + \underline{\widetilde{OC}}_{y_1}|_{\overline{y}_2}\underline{\widetilde{OC}}_{y_2}|_{y_1} + \underline{\widetilde{OC}}_{y_1}|_{y_2}\underline{\widetilde{OC}}_{y_2}|_{\overline{y}_1}\end{aligned} \tag{17a}$$

and by

$$\begin{aligned}\underline{\widetilde{OC}}_\nu = {} & \underline{\widetilde{OC}}_{y_1}|_{\overline{y}_2}\underline{\widetilde{ODC}}_{y_2}|_{y_1} + \underline{\widetilde{OC}}_{y_1}|_{y_2}\underline{\widetilde{ODC}}_{y_2}|_{\overline{y}_1} + \\ & + \underline{\widetilde{ODC}}_{y_1}|_{\overline{y}_2}\underline{\widetilde{OC}}_{y_2}|_{y_1} + \underline{\widetilde{ODC}}_{y_1}|_{y_2}\underline{\widetilde{OC}}_{y_1}|_{\overline{y}_1}\end{aligned} \tag{17b}$$

are subsets of the true $\underline{ODC}_\nu$ and $\underline{OC}_\nu$, respectively.

To verify Eq. (17), it is sufficient to observe that, by expanding Eq. (10a) in sum of products form,

$$\underline{ODC}_\nu = \underline{ODC}_{y_1}|_{\overline{y}_2}\underline{ODC}_{y_2}|_{y_1} + \underline{OC}_{y_1}|_{\overline{y}_2}\underline{OC}_{y_2}|_{y_1} + \underline{ODC}_{y_2}|_{\overline{y}_1}\underline{ODC}_{y_1}|_{y_2} + \underline{OC}_{y_2}|_{\overline{y}_1}\underline{OC}_{y_1}|_{y_2}$$

Similarly, it can be verified from Eq. (10b) that

$$\underline{OC}_\nu = \underline{OC}_{y_1}|_{\overline{y}_2}\underline{ODC}_{y_2}|_{y_1} + \underline{ODC}_{y_1}|_{\overline{y}_2}\underline{OC}_{y_2}|_{y_1} + \underline{OC}_{y_2}|_{\overline{y}_1}\underline{ODC}_{y_1}|_{y_2} + \underline{ODC}_{y_2}|_{\overline{y}_1}\underline{OC}_{y_1}|_{y_2}. \tag{18b}$$

Since we assumed $\underline{\widetilde{ODC}}_{y_i} \subseteq \underline{ODC}_{y_i}$; $\underline{\widetilde{OC}}_{y_i} \subseteq \underline{OC}_{y_i}$, $i = 1,2$, the relations $\underline{\widetilde{ODC}}_\nu \subseteq \underline{ODC}_\nu$ and $\underline{\widetilde{OC}}_\nu \subseteq \underline{OC}_\nu$ are immediately verified.

3.2 Computation of *don't care* subsets only

If only subset of *don't care* conditions $\underline{\widetilde{ODC}}_{y_i}$ are available, then, $\underline{\widetilde{ODC}}_\nu$ can be computed from:

$$\underline{\widetilde{ODC}}_\nu = \underline{\widetilde{ODC}}_{y_1}|_{\overline{y}_2}\underline{\widetilde{ODC}}_{y_2}|_{y_1} + \underline{\widetilde{ODC}}_{y_1}|_{y_2}\underline{\widetilde{ODC}}_{y_2}|_{\overline{y}_1} \tag{19}$$

To verify Eq. (19) it is sufficient to observe that its right member matches the first elements in the sum (17a). Therefore $\underline{\widetilde{ODC}}_\nu$, as computed by Eq. (19), is also contained in $\underline{ODC}_\nu$.

Note that the ODC sets computed by Eq. (17a) always include those derived from Eq. (19). Therefore, the propagation of both *care* and *don't care* sets provide a better approximation to the exact ODC sets.

The ODC subsets computed by Eq. (17a), (17b) and (19) have a local maximality property, in the sense that no other cube of the cofactors $\underline{\widetilde{ODC}}_{y_i}|_{y_j}$, $\underline{\widetilde{OC}}_{y_i}|_{y_j}$ appearing in those equations can be proved to be always contained in $\underline{ODC}_\nu$.

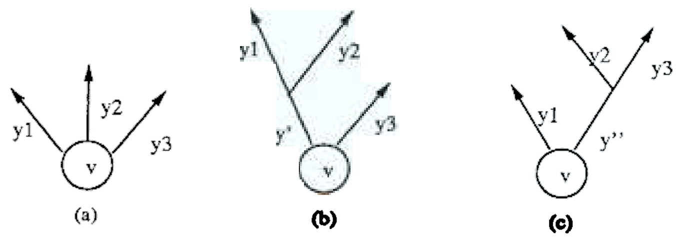

Figure 2: Multi-way fork decompositions of a fanout tree.

3.3 Generalization to larger fanout

A vertex ν with fanout $n > 2$ can always be replaced by a multi-way fork of interconnections, as shown in Fig. (2). Eqs. (17) (or, more simply, Eq. (19)) could be applied iteratively to this fork. In this way a subset of ODC_ν can be obtained in an $O(n)$ number of applications of Eqs. (17) or (19).

The result, however, depends on the selected tree. Consider, for example, the case of three fanout variables, shown in Fig. (2), and suppose, for simplicity, to apply Eq. (19) only. If the fanout tree of ν is decomposed as in Fig. (2b), the resulting expression of $\underline{\widetilde{ODC}}_\nu$ is

$$\begin{aligned}\underline{\widetilde{ODC}}_\nu = {} & \underline{\widetilde{ODC}}_{y_1}|_{\bar{y}_2\bar{y}_3}\underline{\widetilde{ODC}}_{y_2}|_{y_1\bar{y}_3}\underline{\widetilde{ODC}}_{y_3}|_{y_1y_2} + \underline{\widetilde{ODC}}_{y_2}|_{\bar{y}_1\bar{y}_3}\underline{\widetilde{ODC}}_{y_1}|_{y_2\bar{y}_3}\underline{\widetilde{ODC}}_{y_3}|_{y_1y_2} + \\ & \underline{\widetilde{ODC}}_{y_1}|_{\bar{y}_2y_3}\underline{\widetilde{ODC}}_{y_2}|_{y_1y_3}\underline{\widetilde{ODC}}_{y_3}|_{\bar{y}_1\bar{y}_2} + \underline{\widetilde{ODC}}_{y_2}|_{\bar{y}_1\bar{y}_3}\underline{\widetilde{ODC}}_{y_1}|_{y_2\bar{y}_3}\underline{\widetilde{ODC}}_{y_3}|_{\bar{y}_1\bar{y}_2}\end{aligned} \tag{20}$$

If it is decomposed as in Fig. (2c), we instead obtain the following expression, different from the previous one:

$$\begin{aligned}\underline{\widetilde{ODC}}_\nu = {} & \underline{\widetilde{ODC}}_{y_2}|_{\bar{y}_3\bar{y}_1}\underline{\widetilde{ODC}}_{y_3}|_{y_2\bar{y}_1}\underline{\widetilde{ODC}}_{y_1}|_{y_2y_3} + \underline{\widetilde{ODC}}_{y_3}|_{\bar{y}_2\bar{y}_1}\underline{\widetilde{ODC}}_{y_2}|_{y_3\bar{y}_1}\underline{\widetilde{ODC}}_{y_1}|_{y_2y_3} + \\ & \underline{\widetilde{ODC}}_{y_2}|_{\bar{y}_3y_1}\underline{\widetilde{ODC}}_{y_3}|_{y_2y_1}\underline{\widetilde{ODC}}_{y_1}|_{\bar{y}_2\bar{y}_3} + \underline{\widetilde{ODC}}_{y_3}|_{\bar{y}_2\bar{y}_1}\underline{\widetilde{ODC}}_{y_2}|_{y_3\bar{y}_1}\underline{\widetilde{ODC}}_{y_1}|_{\bar{y}_2\bar{y}_3}\end{aligned}$$

The largest expression that can be obtained, starting from the variable *don't care* subsets, is

$$\begin{aligned}\underline{\widetilde{ODC}}_\nu = {} & \underline{\widetilde{ODC}}_{y_1}|_{\bar{y}_2\bar{y}_3}\underline{\widetilde{ODC}}_{y_2}|_{y_1\bar{y}_3}\underline{\widetilde{ODC}}_{y_3}|_{y_1y_2} + \underline{\widetilde{ODC}}_{y_1}|_{\bar{y}_3\bar{y}_2}\underline{\widetilde{ODC}}_{y_3}|_{y_1\bar{y}_2}\underline{\widetilde{ODC}}_{y_2}|_{y_1y_3} + \\ & \underline{\widetilde{ODC}}_{y_2}|_{\bar{y}_1\bar{y}_3}\underline{\widetilde{ODC}}_{y_1}|_{y_2\bar{y}_3}\underline{\widetilde{ODC}}_{y_3}|_{y_2y_1} + \underline{\widetilde{ODC}}_{y_2}|_{\bar{y}_3\bar{y}_1}\underline{\widetilde{ODC}}_{y_3}|_{y_2\bar{y}_1}\underline{\widetilde{ODC}}_{y_1}|_{y_2y_3} + \\ & \underline{\widetilde{ODC}}_{y_3}|_{\bar{y}_1\bar{y}_2}\underline{\widetilde{ODC}}_{y_1}|_{y_3\bar{y}_2}\underline{\widetilde{ODC}}_{y_2}|_{y_3y_1} + \underline{\widetilde{ODC}}_{y_3}|_{\bar{y}_2\bar{y}_1}\underline{\widetilde{ODC}}_{y_2}|_{y_3\bar{y}_1}\underline{\widetilde{ODC}}_{y_1}|_{y_3y_2}\end{aligned} \quad (22$$

larger than (or equal to) both estimates (20) and (21).

In the general case, by expanding Eq. (13) in sum-of-products form, it is immediate to verify that the product

$$\underline{\widetilde{ODC}}_{y_1}|_{\bar{y}_2,\cdots,\bar{y}_n}\underline{\widetilde{ODC}}_{y_2}|_{y_1,\bar{y}_3,\cdots,\bar{y}_{n-1},\bar{y}_n}\cdots\underline{\widetilde{ODC}}_{y_n}|_{y_1,y_2,\cdots,y_{n-1}}$$

is certainly contained in $\underline{ODC}_\nu$. Let $(i_1, i_2, \cdots, i_n)$ denote a permutation of $(1, 2, \cdots, n)$. From the expression (15) of the exact $\underline{ODC}_\nu$ associated to that permutation it follows that

$$\underline{\widetilde{ODC}}_{y_{i_1}}|_{\bar{y}_{i_2},\cdots,\bar{y}_{i_n}}\underline{\widetilde{ODC}}_{y_{i_2}}|_{y_{i_1},\bar{y}_{i_3},\cdots,\bar{y}_{i_n}}\cdots\underline{\widetilde{ODC}}_{y_{i_n}}|_{y_{i_1},y_{i_2},\cdots,y_{i_{n-1}}} \tag{24}$$

is also always contained in $\underline{ODC}_\nu$. Let π_n denote the set of all possible permutations $\underline{i} = (i_1, i_2, \quad, i_n)$ of $(1, 2, \cdots, n)$. The sum

$$\sum_{\underline{i}\in\pi_n}\underline{\widetilde{ODC}}_{y_{i_1}}|_{\bar{y}_{i_2},\cdots,\bar{y}_{i_n}}\underline{\widetilde{ODC}}_{y_{i_2}}|_{y_{i_1},\bar{y}_{i_3},\cdots,\bar{y}_{i_n}}\quad\underline{\widetilde{ODC}}_{y_{i_n}}|_{y_{i_1},y_{i_2},\cdots,y_{i_{n-1}}}$$

is therefore a subset of $\underline{ODC}_\nu$.

Similarly to the case of two fanout variables, it can be shown that no other product of the variables appearing in Eq. (25) can be added, while maintaining the first member of Eq. (25) an ODC subset. From the expansion of Eq. (13), all candidate products involve in fact the complement of at least one fanout variable ODC subset, *i.e.* an OC *superset*. If such a product is added, it is no longer possible to guarantee $\widetilde{ODC}_\nu \subseteq \underline{ODC}_\nu$. Eq. (25) therefore represents the largest possible estimate.

Unfortunately, its computation requires a factorial amount of time, and is therefore not applicable to large fanouts. Note, however, that for most circuits the number of fanout stems for a node is typically small.

When subsets of the true $\underline{OC}_{y_i}$ sets are also available, then the largest possible subsets of $\underline{ODC}_\nu, \underline{OC}_\nu$ can be computed as follows. A sum-of products expression of $\underline{ODC}_\nu$, $\underline{OC}_\nu$ can be obtained by expanding the expressions (13) and (14). By substituting each $\overline{ODC}_{y_i}$ and $\overline{OC}_{y_i}$ with $\widetilde{OC}_{y_i}$ and $\widetilde{ODC}_{y_i}$, respectively, an expression of ODC and OC subsets is obtained. Different subsets are obtained by repeating the same procedure using a permutation of the variables $y_1, \cdots, y_n$. The largest possible subset is the sum of these subsets.

4 Relation to previous works.

Methods for computing observability *don't care* sets have been proposed in [8], [3] and [7]. None of these methods maintains separate components for the observability at different outputs, and therefore must either make some form of approximations in case of reconvergent fanout, or abandon the network traversal method.

The most "conservative" approach is taken in the program MIS [8]. There, the observability *don't care* of a gate in a vertex ν is computed by assuming the complete observability of its fanout gates. The observability $ODC_{y_i}^{MIS}$ of each fanout variable y_i of ν is thus obtained from Eq. (3) by neglecting the first term of the sum. MIS computes the observability *don't care* set of ν as

$$ODC_\nu^{MIS} = \prod_{y_i} \frac{\overline{\partial f_{\mu_i}}}{\partial y_i} \tag{26}$$

$$ODC_{y_i} = ODC_z + \frac{\partial z}{\partial y_i} = ODC_z + \sum_{j=1, j\neq i}^{n} y_j \tag{27}$$

which corresponds (with a different formalism) to Eq. (3.1) in [3]. The cases of not-reconvergent and reconvergent fanout are treated separately. In the case of not reconvergent fanout, the observability of the gate output ν is computed (cfr. Eq. (3.4) in [3]) from those of the fanout variables y_i as

$$ODC_\nu = \prod_{i=1}^{n} ODC_{y_i} \tag{28}$$

In the case of reconvergent fanout, since the different components of the observability are not kept separate, the network traversal paradigm must be abandoned. Muroga proposes an approach based upon *two exahustive simulations* of the circuit, with the reconvergent gate output stuck-at 0 and stuck-at 1, respectively. The observability *don't care* set for the gate is then computed as the set of primary input configurations that produce the same output in the two simulations. In other words, Eq. (1) is solved by "brute force".

In [7] ODC subsets are computed as follows. Variables are associated to gate outputs rather than edges. Eq. (3) is used to determine the observability of fanin variables for each vertex. In presence of reconvergent fanout lines, the observability *don't care* sets associated to each variable are considered as estimates of the true ODC of the vertex. Note that, because of the different definition of variables, in presence of reconvergent fanout the observability of a vertex seems to depend on itself. Consider, for example, the circuit of Fig. 1, and assume that the gate in u_4 does not exist. The method finds two estimates of the observability of u_1 as

$$ODC'_{u_1} = u_1 + x_1; \;\; ODC''_{u_1} = u_1 + x_4$$

This apparent dependency is eliminated by invoking the RESTRICT operation on the expressions ODC'_{u_1}, ODC''_{u_1} which drops all the cubes containing u_1 or $\overline{u}_1$. The RESTRICT operation on an expression $expr$ with respect to a variable var can be defined as follows:

$$RESTRICT(expr, var) \stackrel{\text{def}}{=} expr|_{var} expr|_{\overline{var}}$$

In the example of Fig. 1,

$$RESTRICT(ODC'_{u_1}, u_1) = ODC'_{u_1}|_{u_1} ODC'_{u_1}|_{\overline{u_1}} = x_1;$$

$$RESTRICT(ODC''_{u_1}, u_1) = ODC''_{u_1}|_{u_1} ODC''_{u_1}|_{\overline{u_1}} = x_4. \tag{31}$$

RESTRICT is therefore similar to the cofactor operation of Eqs. (8) and (17). Eq. (17) shows, however, that only part of the cubes (containing either u_1 or $\overline{u}_1$) actually needs to be dropped.

The estimates obtained after the application of RESTRICT are finally ANDed. In the above example, $ODC_{u_1} = x_1 x_4$. It has been proven in [7] that the final result is a subset of the true ODC set. The exact result is, in the above example, $ODC_{u_1} = x_1 + x_4 \supseteq x_1 x_4$.

5 Summary

In this paper we have presented a novel algorithm for the extraction of exact and approximate Observability Don't Care sets (ODC sets) in a combinational multilevel logic network. The proposed algorithms are efficient because they use local information, i.e. the computation of the ODC for a vertex requires only the knowledge of the *don't cares* at the vertex immediate fanout. Two approaches have been proposed. The former computes the exact ODC sets. The latter computes ODC subsets and can be used when partial information on the *don't cares* is available.

6 Acknowledgement

This research was supported in part by a fellowship of the Rotary Foundation and by AT & T and DEC, jointly with NSF, under a PYI award program. We also acknowledge support from NSF under contract # MIP 8719546.

The helpful discussions with Frederic Mailhot and Michiel Ligthart are here gratefully acknowledged.

References

[1] K. A. Bartlett, R. K. Brayton, G. D. Hachtel, R. M. Jacoby, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, " Multilevel Logic Minimization Using Implicit Don't Cares", *IEEE Transactions on CAD/ICAS*, vol. CAD-7, No. 6, pp. 723-739, June 1988.

[2] G. D. Hachtel and M. R. Lightner, " Top-Down Synthesis of Multilevel Logic Networks" *Proc. ICCAD 1987*, pp. 316-319, S. Clara, Nov. 1987.

[3] S. Muroga, Y.Kambayashi, H.Lai and J.Culliney, "The Transduction Method - Design of Logic Networks Based on Permissible Functions ", *IEEE Trans. Comp.*, vol. 38, No. 10, pp. 1404-1424, 1989.

[4] D. Bostick, G. D. Hachtel, R. M. Jacoby, M. R. Lightner, P. Moceyunas, C. R. Morrison, and D. Ravenscroft, " The Boulder Optimal Logic Design System", *Proc. ICCAD 1987*, pp. 62-65, S. Clara, Nov. 1987.

[5] R. K. Brayton and F. Somenzi, "Boolean Relations and the Incomplete Specification of Logic Networks", *Proc. VLSI '89*, pp. 231-240, Munich, August 1989.

[6] H. Fujiwara, *Logic Design and Design for Testability*, MIT Press, Cambridge, 1985.

[7] G. D. Hachtel, R. M. Jacoby, P. H. Moceyunas, " On Computing and Approximating the Observability Don't Care Set ", *Proceedings on the International Workshop on Logic Synthesis*, Research Triangle Park, May 1989.

[8] R. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, A. Wang , "MIS: A Multiple-Level Logic Optimization System", *IEEE Transactions on CAD/ICAS*, Vol. CAD-6, No. 6, pp. 1062-1081, November 1987.