Minsk, Belarus, October 2-4, 2001

INFORMATION MEASURES TO SUPPORT LOGIC NETWORK SYNTHESIS

V. Cheushev[1], C. Moraga[2]

[1]– Center of Information Resources and Communications, Belarusian State University, 220050, Minsk, BELARUS, tel. (37517) 220–64–13, e-mail: cheushev@niks.by
[2]– Department of Computer Science, University of Dortmund, 44221, Dortmund, GERMANY, e-mail: moraga@cs.uni-dortmund.de

ABSTRACT

We investigate the problem of the synthesis of combinational multi-level networks supported by information measuring. Using these measures, we can detect the possibility to use efficiently in the design already designed sub-networks and circuits from a given library. The proposed approach can be applied to drive the search in various methods, e.g., in evolutionary synthesis and other strategies.

1. INTRODUCTION

Logic design of binary and multiple-valued circuits has to face two challenges: *effectiveness* and *efficiency*. Effectiveness is a measure of the quality of the obtained circuit in terms of e.g. number of gates and connections, regularity of structure, depth, and energy consumption. Efficiency is a measure of the required computing time and memory in order to achieve a certain level of effectiveness.

Evolutionary strategies applied to logic design support efforts for effectiveness, since different aspects of the quality of a circuit may be evaluated independently and combined to give the fitness of the circuit under evolution [1, 2]. The value of the fitness drives the evolution within the problem space. This, however is obtained at the price of low efficiency. Evolutionary design of non-trivial digital circuits is characterized by high demands on computing time. The real problem is then to achieve effectiveness without loosing efficiency. In what follows we discuss a different strategy. We propose a way to support evolutionary and other search strategies of the design with quantitative information measuring. The proposed measures allow to detect, whether or not some part of the network "keeps the target functionality", to estimate a "progress" in the design, to compare several steps with its "quality". As a result, we can stop, in proper time, deadlock branches of the search and also choose the best branch.

We consider efficient using already designed parts of the network. Moreover, we assume the existence of a library of good circuits that may be used as building blocks to realize the goal circuit and provide with information measurements to detect when this is possible.

In this paper, we continue our recent research on application of Information Theory Methods to solve Logic Design problems [3 – 7]. In addition to already reported Information Theory measures we introduce, for the first time, new information estimations, namely *information potential of a network* and *information potential of a function with respect to a network* and discuss, how these measures can be used in the synthesis.

The rest of the paper is organized as follows. The problem under consideration is viewed in Section 2. In Section 3 we discuss its relationship to flexibility of the synthesis. The general outline of our approach is given in Section 4. Section 5 introduces the set of information estimations and related concepts. In Section 6 we discuss one example of application of the proposed approach. In Section 7, an information model of the design process is given, and Section 8 concludes the paper.

2. PROBLEM FORMULATION

We consider the following problem: *"Given a logic function, synthesize a network to implement it."* Assuming this problem to be solved with some step-by-step strategy, we discuss the sub-problem, how to use efficiently already designed parts of some bigger network, sub-networks, obtained on previous iterations or/and circuits from a given library $\mathcal{L}_C$.

Example 1 *Suppose, we have to implement the one-bit full adder, whose primary inputs are a, b, and c_{in}, and outputs are sum = [01101001], c_{out} = [00010111]. We can synthesize a network in a direct way namely find a simple form for each function, e.g. $sum = a \oplus b \oplus c_{in}$, $c_{out} = ab + ac_{in} + bc_{in}$ and then realize them (Fig. 1,a).*

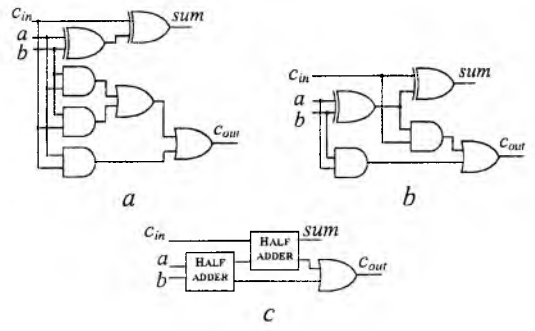

Figure 1. Network solutions for one-bit full adder

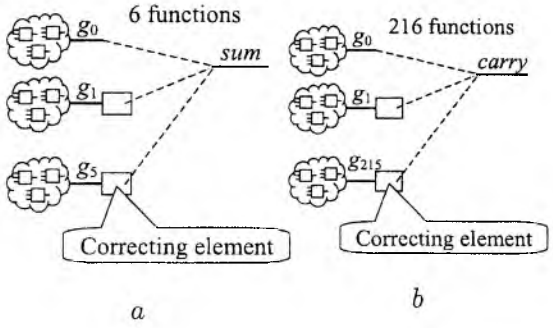

Figure 2. Illustration for flexibility of the design: instead of looking for exact implementations, we can find a network for any of 6 functions, which may be easily corrected to *sum* and a network for any of 216 functions, which can be easily corrected to *carry*

The first part of the obtained network (the implementation for sum) is a good solution in terms of number of gates but the second part seems to be not a good one.

An experienced designer asks himself: "If I would use a signal from the first part of the network, could I simplify the second one?" For many cases, the answer is "Yes!" (Fig. 1,b). Moreover, an experienced designer knows that in the library of well-designed circuits there should be a so-called Half-adder. It may be optimized for the target technology and it is good practice to use it (Fig. 1,c).

So, in simple words we can outline the discussed problem as follows: *"If we have something already designed, how to use it more efficiently?"*

Traditional methods of the synthesis work with very limited view. Up to now, only designer's experience can overcome these limitations and allows to see more than a confined class of logic manipulations. Evolutionary methods seems to be a good alternative thanks to random nature of the search, but such a search (only slightly limited) is time consuming. Our goal is to propose a systematic way to drive the search (not only evolutionary one) with information measuring.

Here, the following observation is important: *To be efficiently used in the synthesis, a sub-network should implement a part of the target functionality.*

Based on this observation, we can divide the problem into two parts. The first one is *Whether or not a sub-network Net implements a part of the target functionality?* and the second one is *How to transform a "partial" functionality of a sub-network Net into the full functionality of the target network?*

In order to detect a possibility to use already designed sub-network Net or circuits from a library $\mathcal{L}_C$, we apply *Information Theory Measures* (ITMs). With these measures, we look for a sub-network, which is a "nearest neighbor" for the target network. If such a neighbor is found, we look for a possibility to transform (or "correct") it to achieve the target functionality. So, the idea is that instead of searching in the huge space of concrete functional transformation of circuits, we first look, using ITMs, for a *possibility* of a transformation. Only if such a possibility exists, we look for a proper transformation. The benefit is extreme reducing the search space and more opportunities to find good solutions. Partly, this idea was reported in [5, 6, 7]. Here, we propose to reinforce information support of the synthesis and look not only for nearest neighbor networks but also for "remote" ones. The basis is the concept of *keeping functionality* and the information measure called *information potential of a network.*

3. FLEXIBILITY OF THE SYNTHESIS

The question, *Whether or not some sub-network can be efficiently used in the design*, is closely related to *flexibility of the synthesis.*

Different approaches have been proposed to express functional flexibility and flexibility of the design (see, for example, [8, 9]).

One aspect of flexibility is the question *"Given the target function, how many networks can be recognized to be solutions for it?"* Certainly, there are lot of networks, such that any of them implements the target function. This answer is right, but it is not exhaustive for flexible design. Say, for a Boolean function we can find a network, implementing this function exactly, but we can also find a network, implementing the complement of the target function and put the inverter to the its output. There are much more possibilities to do something similar in the multi-valued functions domain.

Example 2 *Assume we need to implement the ternary adder with primary inputs x_1, x_2 and outputs sum = [012120201] and $carry$ = [000001011].*

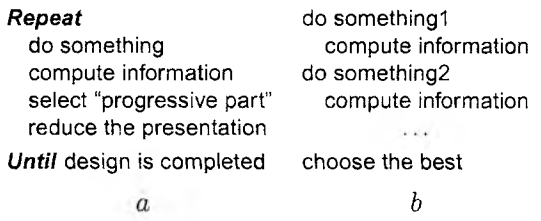

Figure 3. Supporting the Logic Design by information measuring

Suppose, we are testing an unlimited set of networks and look for solutions for these functions.

Using a traditional verificator, we will detect any network implementing the given function exactly (functions g_0 in Fig. 2). However, in [5] we showed that there exist 6 so-called 1-neighbor functions such that any of them can be easily corrected to the function sum and 216 such functions for carry. We found all possible corrections of this type for ternary logic and concluded that any correction can be performed with at most two standard gates [5]. Using verification based on information measuring, we can detect any network from a large set as shown in Fig. 2 and correct it in case of need. So, we have a chance to use in our design much more well designed solutions, especially to implement both functions with one combined network.

4. Outline of the Approach

In this section, we outline our approach without formulas and mathematic justification. We do it, based on two reasons. The first one is that it is better to explain a new idea from the beginning in the simplest words. The second one is that information measures can be defined in various manners, and the way proposed in Section 5, is not unique, whereas the general approach is "more universal".

Information measuring can be applied to support various strategies and methods of the network synthesis. For example, we developed algorithms to build decision diagrams [4], we use the entropy-based fitness function in evolutionary design [6], the strategy is proposed to detect so-called "neighbor" functionality to stop the evolutionary synthesis and efficiently complete the design in a systematic way [5, 6]. In [7] we discuss, how ITMs allow to detect the possibility of a transformation of a circuit from a given library to obtain the target network.

Here, we consider a way to apply information measuring in some step-by-step strategy of the design of multi-level digital network. We consider a step of the design as follows (Fig. 3,*a*).

Do something to realize a part of the network (put or replace elements, change interconnections, etc.). This step results in some network Net.

Compute information namely

– information potential $Q(Net)$ of the network;

– information potential $Q(f|Net)$ of the target function with respect to the network;

– entropy $\mathbf{H}(f|Net)$ of the target function with respect to the network.

Select "progressive part" of the performed step namely a part, which increases the information potential $Q(Net)$ of the network.

Reduce the presentation, that means decreasing the number of arguments of the function, selecting the best sub-network and removing redundant parts, etc. To choose the best sub-network, the criterion

$$Q(Net) = \max \tag{1}$$

can be used.

Above actions can be repeated ***Until the design is finished.*** In this point the information potential of the network is equal to the initial information potential of the goal function:

$$Q(Net) = Q(f). \tag{2}$$

A way to apply information measures is to use them to be optimization criteria (see Fig. 3,*b*). The best action, e.g., a branch of a search, can be selected according to (1) or a more complex criterion, using (1) to be the base.

So, we are now in the position to discuss, how to estimate information potentials of a network and a logic function and how its behavior reflects some events in a synthesis process.

5. Information Measures and Related Concepts

5.1. Variety source

Informally, the **variety source** or, in other words, **variable source** over the set X of logic variables is something whose values change in connection with changes of variables.

For example, a Boolean variable is a single variety source taking values 0 and 1. A couple of variables is a variety source, too, and it takes four different values 00, 01, 10, 11. A function over these variables is also a variety source, but it takes at most two values. We can say, a role of a logic function is to reduce the variety from initial number of values, that is the number of patterns, to m values of the function. This observation is an important basis for information estimations and its using.

So, we consider any combination of variables or any combination of signals in the network to be a variety source.

Definition 1 *Given a network, the variety source V is any set of primary inputs, outputs, and accessible internal signals of this network.*

5.2. Information measures of variety sources

Let $\mathcal{F}$ be a finite, non-empty superset of logic values such that a value $\alpha \in \mathcal{A} = \{0, \ldots, r-1\}$ occurs in $\mathcal{F}$ k_α times. The superset $\mathcal{F}$ can be characterized by the distribution $\Lambda_\mathcal{F} = \{k_0, \ldots, k_{r-1}\}$.

Definition 2 *Let the* **information potential** $Q(\mathcal{F})$ *of the superset $\mathcal{F}$ be*

$$Q(\mathcal{F}) = Q(\Lambda_\mathcal{F}) = \sum_{i=0}^{r-1} k_i \log k_i. \qquad (3)$$

Proposition 1 *A variety source V can be characterized by the distribution $\Lambda_V = \{k_{V=0}, \ldots, k_{V=m-1}\}$ with the information potential*

$$Q(V) = Q(\Lambda_V).$$

A complex variety source $V = \{V_1, \ldots, V_r\}$ can be characterized by the distribution $\Lambda_V = \{k_{V=00\ldots0}, \ldots, k_{V=m^r-1}\}$ with the information potential

$$Q(V) = Q(\Lambda_V).$$

Definition 3 *Given the variety source V_i, the* **integrated conditional entropy** *of the variety source V_j is*

$$\mathbf{H}(V_j|V_i) = Q(V_i) - Q(V_i, V_j). \qquad (4)$$

Example 3 *Let $f(x_1, x_2) = [012120200]$ be a ternary function.*

The distribution of values of the function f is

$$\Lambda_f = \{k_{f=0}, k_{f=1}, k_{f=2}\} = \{4, 2, 3\}$$

and the information potential is

$$\begin{aligned} Q(f) &= 4\log_3 4 + 2\log_3 2 + 3\log_3 3 \\ &= 9.31 \; ternary \; units. \end{aligned}$$

The variety source x_1 is characterized by the truth vector [000111222] with the distribution

$$\Lambda_{x_1} = \{k_{x_1=0}, k_{x_1=1}, k_{x_1=2}\} = \{3, 3, 3\}$$

and the information potential

$$Q(x_1) = 9 \; ternary \; units.$$

Being considered jointly, x_1 and f form the superset

$$(x_1, f) = \begin{bmatrix} 000111222 \\ 012120200 \end{bmatrix}$$

of values with the distribution

$$\Lambda_{(x_1,f)} = \{k_{00}, \ldots, k_{22}\} = \{1, 1, 1, 1, 1, 1, 2, 0, 1\}$$

and the information potential

$$Q(x_1, f) = 1.26 \; ternary \; units.$$

The integrated conditional entropy of the function f with respect to the variable x_1 is

$$\mathbf{H}(f|x_1) = Q(x_1) - Q(x_1, f) = 7.74 \; ternary \; units.$$

5.3. Keeping the functionality

We say **the variety source keeps the target functionality** if the target function may be expressed to be a function over this variety source.

Example 4 *Let a switching function f be $f = f(a, b, c) = [01010111]$ and let $x = ab$, $y = a + b$.*

Obviously, the variety source $\{a, b, c\}$ keeps the functionality f. The variety source $\{x, c\}$ keeps the target functionality, too, because f may be expressed to be a function over (x, c), e.g. $f = x + c$. The variety source $\{y, c\}$ does not keep the functionality, since f does not a function over (y, c).

Let $\mathcal{A}$ be the superset of values of the variety source V and $\mathcal{B}$ bc the superset of values of the function f.

We will denote by $V \backsimeq f$ the statement "the variety of V is equal to the variety of f". It is shown in [5], in this case there exists a *bijective mapping* from $\mathcal{A}$ into $\mathcal{B}$, and $Q(V) = Q(f)$.

$V \rhd f$ will mean "the variety of V can be reduced to the variety of f". For this case there exists a *surjective mapping* from $\mathcal{A}$ into $\mathcal{B}$, and $Q(V) < Q(f)$.

$V \trianglerighteq f$ will mean "the variety of V is equal or can be reduced to the variety of f". Here, there exists a *bijective or a surjective mapping* from $\mathcal{A}$ into $\mathcal{B}$, and $Q(V) \leq Q(f)$ (see [5] for details).

Example 5 *For functions shown on Fig. 2,a, $g \backsimeq f$, $g \in \{g_0, \ldots, g_5\}$, for functions on Fig. 2,b, $g_0 \backsimeq f$ and $g \trianglerighteq f$, $g \in \{g_1, \ldots, g_{215}\}$.*

The following affirmation explains the importance of the introduced concept: *If V keeps the functionality f (i.e. $V \trianglerighteq f$) then the synthesis of the network for f can be finished using signals from V only.*

The following theorem provides a way to detect such a variety source using information measure (see [5] for details and proof).

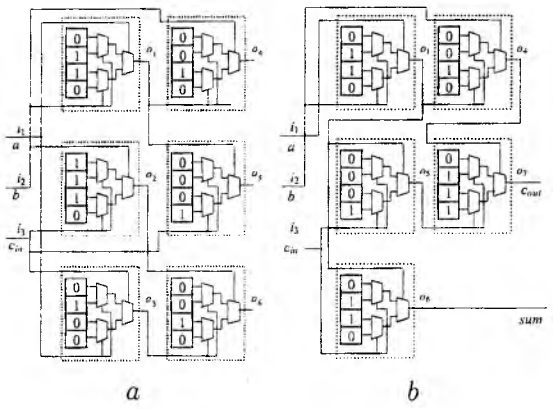

Figure 4. Evolved network (a) and network after correction (b)

Theorem 1 *The equality*

$$\mathbf{H}(f|V) = 0 \qquad (5)$$

is a necessary and sufficient condition to V keeps the functionality f.

So, computing the entropy $\mathbf{H}(f|V)$, we check, whether the design of a target network can be finished using signals from V only. Let us demonstrate on example, how we used this property in our recent experiments with an evolutionary algorithm [6].

Example 6 *The evolutionary algorithm puts at random elements into a given circuit-box and creates interconnections between them. Then verification is performed to check, whether the obtained network implements the given function. If not, the algorithm changes something in the network (by means of evolution operations) and verifies it again. And so on, for millions of iterations. In many cases, success is not achieved (within the predefined number of circuit-boxes or within a reasonable amount of computing time).*

We improved the algorithm in the following way.

Assume we need to realize the one-bit full adder on the basis of FPGAs and after some iterations of the evolutionary synthesis we have the network, shown on Fig. 4,a. This network does not realize the target functions on its outputs. Before destroying this network and building a new one, we test, whether the functionality we need is present inside the network. For this aim we check pairs of signals by computing the conditional entropy of the target function with respect to each pair. If the entropy is zero, it means the pair keeps the target functionality and we can obtain the desired function using one additional FPGA cell.

For this example, we have found such pairs for both functions:

$$\mathbf{H}(c_{out}|o_5, o_4) = 0 \Rightarrow V = \{o_5, o_4\} \trianglerighteq c_{out}$$
$$c_{out} = \varphi(o_5, o_4) = [0111],$$
$$\mathbf{H}(sum|o_1, i_3) = 0 \Rightarrow V = \{o_1, i_3\} \trianglerighteq sum$$
$$sum = \varphi(o_1, i_3) = [0110].$$

Here, φ denotes the so-called correcting function. During entropy computation, we obtain all necessary data to express it. So, after implementing two correcting functions with FPGA cells and removing redundant cells, we obtain the target network (Fig. 4,b). This approach allowed us to improve efficiency of the evolutionary synthesis dozens of times (see [5, 6]). Moreover, only 5 FPGA cells were needed.

In the above example we exploited only one information estimation and the condition (5) to support the synthesis. This is a simple limited way to apply information measuring. Our further goal is to enlarge the advantages of information measuring. To achieve this goal, we propose to analyze the *dynamic behavior* of information potentials during the process of the synthesis.

5.4. Information potential of a network

To estimate a network with partial functionality, we introduce the following concepts.

Given network Net, let V^F denote a variety source, keeping the target functionality.

Definition 4 *Given a network Net, the **maximal informative variety source** of this network is a variety source V^{Fm}, whose information potential is maximal among all V^F:*

$$V^{Fm} = \arg\max_{V^F} Q(V_i^F). \qquad (6)$$

Proposition 2 *The **information potential of the network** Net is the information potential of a maximal informative variety source of this network:*

$$Q(Net) = Q(V^{Fm}). \qquad (7)$$

Proposition 3 *The **information potential of the target function** with respect to a network is its information potential with respect to a maximal informative variety source:*

$$Q(f|Net) = Q(f|V^{Fm}). \qquad (8)$$

Using information potential of a maximal informative variety source and information potential of the target function with respect to a network, we obtain the opportunity to compare several partial network solutions (for example, several branches of a search).

The next section demonstrates our idea using a simple example, and Section 7 collects introduced measures in a form of an information model.

Table 1. The target function and intermediate signals

a	b	c	f	$x = b + c$	$y = ab$	$z = ac$
1	2	3	4	5	6	7
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1

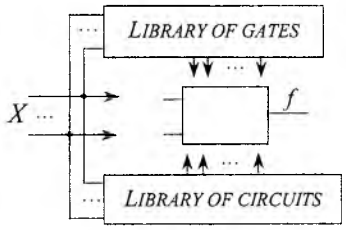

Figure 5. Signals to synthesize a network for f

6. AN EXAMPLE

In this section we discuss, based on a simple example, how information estimations can be used in the design of a logic network.

Suppose, we know nothing about methods of conventional synthesis and we need to synthesize a network for the switching function given in Table 1 (columns 1-4), using primary inputs, gates from the given library, and networks from another library (Fig. 5). (Certainly, in a real-life design the library of gates can be extended with a set of already designed parts of the network.)

Now, we try to design a network for the given function based on information estimations only. We use the following operations:

– Select a variety source V.

– Check, whether the variety source V keeps the target functionality f according to the criterion (5).

– Compute the information potential $Q(V)$.

– Choose the "best variety source" according to the criterion (1).

Assume, the library of circuits contains nothing to realize our target function directly. Thus, we need to do something to simplify or change this function.

Let us examine 3 gates from the library to check, whether or not they may be efficiently used in our design.

I. First, we check an OR gate with inputs b and

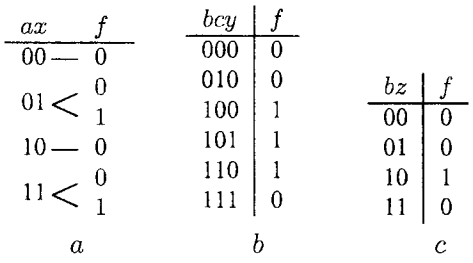

Figure 6. Variety sources: $\{a, x\}$ does not keep the target functionality, $\{b, c, y\}$ and $\{b, z\}$ keep it

c (the signal x in Table 1). We look for a *minimal variety source keeping functionality*. We start from the smallest set namely the pair $\{a, x\}$. Entropy computation shows that this set does not keep the functionality. It is easy to check that the target function cannot be expressed to be a function of these two arguments (see Fig. 6,a). Then, we check bigger sets, namely $V = \{a, b, x\}$, $V = \{a, c, x\}$, $V = \{b, c, x\}$. The result is the same: entropy $\mathbf{H}(f|V) > 0$ that implies, according to Theorem 1, that $V \not\supseteq f$ (V does not keep the functionality f). Only the set $V = \{a, b, c, x\}$ keeps the target functionality. But this set contains all primary inputs! This means, the gate we checked is useless for our design. The value of the information potential confirms our conclusion:

$$Q(a, b, c, x) = 0 \qquad \text{– no progress.}$$

II. Second, we check the signal y (Table 1, column 6), namely the gate AND with inputs a and b:

$$V = \{c, y\}: \quad \mathbf{H}(f|V) > 0 \;\Rightarrow\; V \not\supseteq f$$
$$V = \{a, c, y\}: \quad \mathbf{H}(f|V) > 0 \;\Rightarrow\; V \not\supseteq f$$
$$V = \{b, c, y\}: \quad \mathbf{H}(f|V) = 0 \;\Rightarrow\; V \supseteq f$$

We found the minimal variety source keeping the functionality with 3 signals, b, c, and y (we can make sure from Fig. 6,b that f can be expressed to be $f(b, c, y)$). The information potential

$$Q(b, c, y) = 4\ bit \quad \text{– there is some progress.}$$

III. Third, we test the gate AND with inputs a and c (the signal z in Table 1). In this case, the minimal variety source keeping functionality is two signals only, b and z. It follows from Fig. 6,c that f can be expressed as

$$f = b \cdot \overline{z} \tag{9}$$

that means, the design can be finished. So, this is the best result in our search, and the value of the information potential points to this fact:

$$Q(b, z) = 9.5\ bit \quad \text{– it is the best progress.}$$

Now, let us discuss an interpretation of the obtained information estimations. Our goal here is to

Table 2. The target function, expressed to be the function $f(b, c, y)$

b	c	y	f				
0	0	0	0	0	0	0	0
0	0	1	–	0	0	1	1
0	1	0	0	0	0	0	0
0	1	1	–	0	1	0	1
1	0	0	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	0	0	0	0
				more flexibility			

show that values of information measures reflect (in an implicit way) various aspects of real-life design.

Let us suppose, an experienced designer obtained for the target function the following solution, which is a minimal one in terms of number of gates:

$$f = b \cdot \overline{c \cdot a}. \tag{10}$$

It is not hard to see, we have also obtained the equivalent solution (9) that for this simple example is a final one.

Let us consider now a *partial solution*. Using the signal y, we can obtain

$$f = b \cdot \overline{c \cdot y}. \tag{11}$$

The solution (11) seems to be not better than (10). But it is right in a "local sense". In a real-life design "locally better" is not always the same that "globally better". We have seen it on Example 1.

Really, in comparison with (10), the solution (11) contains the additional gate to implement y. But what if the signal y exists in another part of the network? Then it is "free" namely using this signal we do not increase the cost of the network. If so, why the second solution may be better than the first one?

It may be better because it is *more flexible*. In the second case we have the *incompletely specified function* (see Table 2), which may be "covered" by one of 4 completely specified functions. Any implementation for any of these 4 functions can be solution for it. If one of these implementations is presented in the given library of circuits, we can use it and complete the design. So, we obtain *more freedom* for our synthesis and more possibilities to use well-designed solutions.

Now, let us examine the behavior of our information measures (information potentials) for this example. If we use the signal x, we have

$$Q(Net) = 0, \qquad Q(f|Net) = 16.4\ bit,$$

for the signal y we have

$$Q(Net) = 4\ bit, \qquad Q(f|Net) = 9.5\ bit,$$

and for the signal z:

$$Q(Net) = 9.5\ bit, \quad Q(f|Net) = 4.75\ bit.$$

With x no improving is achieved, with y we reduce the number of specified patterns for the target function (see Fig. 6,*b*), and with z we have benefit in reducing the number of arguments of the target function (Fig. 6,*c*). So, we have compared three different implementations for a step of the synthesis and we can conclude that information measuring give us an opportunity to estimate the quality for each implementation.

7. INFORMATION MODEL

In the previous section, we have analyzed the behavior of information potentials on the example. In this section we will form an information model, which reflects the process of the synthesis in terms of information measures.

In addition to information measures, we will estimate the complexity of a network by the number of gates N_G and the complexity of the target function by a combined characteristic $\langle representation \rangle$ (it can be the number of arguments and the number of specified patterns, the number of nodes of BDD, etc.).

Now, we can describe a step-by-step synthesis process in terms of introduced concepts.

1. In the initial state of the synthesis, when $Net = \{primary\ inputs\}$, the information potential of the network is 0, V^{Fm} coincides with X, the number of gates is 0, the information potential of the target function is maximal, the initial representation is maximal, too:

$$Q(Net) = Q(x_1, \ldots, x_n) = 0,$$
$$V^{Fm} = \{X\}, \quad N_G = 0,$$
$$Q(f|Net) = Q(f|x_1, \ldots, x_n) = \max,$$
$$\langle representation \rangle = \max.$$

2. During a (successful) synthesis, the information potential of the network increases. In contrary, the information potential of the target function with respect to the network decreases, a representation of the target function via signals from a maximal informative variety source reduces:

$$Q(Net) \nearrow, \quad N_G \nearrow$$
$$Q(f|Net) \searrow, \quad \langle representation \rangle \searrow.$$

3. In the final state of the synthesis, when Net implements f, the information potential of the network is equal to the initial information potential of

the target function and the information potential of the target function with respect to the network is 0:

$$Q(Net) = Q(f|x_1, \ldots, x_n), \quad Q(f|Net) = 0,$$
$$V^{Fm} = \{f\}, \quad \langle representation \rangle = f.$$

4. At any point of the synthesis the network and the target function can be characterized by their information potentials.

Here, we need to give some comment on reducing the representation of the target function. This step is especially important for CAD, because for complex problems, it is a bad feature of a computer program to increase the size of the problem.

Example 7 *Return to the one-bit full adder. Suppose, we are synthesizing it with a step-by-step regular strategy. If we put the EXOR-gate at the first step, we can consider the function of 4 arguments (Fig. 7,a). If we will work with such increasing of the size at every step, we take the risk of an overflow. And vice versa, if we will subsequently reduce (or, at least, do not increase) the representation (Fig. 7,b), we have a chance to complete the design in a short time. Here, we cannot say that the solution (b) is better than (a) in terms of effectiveness, but it is better in terms of efficiency.*

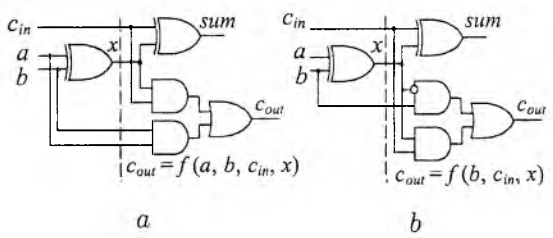

Figure 7. Illustration to increasing the representation of a target function

Information measuring allows to detect the possibility to reduce the presentation. Moreover, maximal information potential of a network usually is a feature of the maximal reducing of a representation of the target function via signals of maximal informative set of signals of this network.

8. Conclusion

The set of information measures is proposed, which can be used to drive the design of multilevel combinational networks. These measures allows without any functional transformations

- To ascertain, whether or not some combination of signals in the designed network keeps the target functionality
- To estimate a "progress" of the design
- To choose a better solution with "partial functionality"
- To detect that some network, not realizing the target functionality, realizes a "neighbor functionality" and can be corrected in a simple systematic way to achieve the target functionality

Application of proposed measures can be especially useful for multivalued logic because complex functional dependencies can be easier expressed via information ones [5, 7].

References

[1] T. Aoki, N. Homma, T. Higuchi, Evolutionary Design of Arithmetic network, *IEICE Transactions on Fundamentals*, vol. E-82-A, no. 5, 1999, pp. 798–806.

[2] C. Moraga, W. Wang, Evolutionary Methods in the Design of Quaternary Digital networks, *Proc. IEEE* 28^{th} *Int. Symp. on Multiple-Valued Logic*, 1998, pp. 89–94.

[3] V. Cheushev, V. Shmerko, D. Simovici. S. Yanushkevich, Functional Entropy and Decision Trees, *Proc. IEEE Int. Symp. on Multiple-Valued Logic*, 1998, pp. 357–362.

[4] V. Shmerko, D. Popel, R. Stanković, V. Cheushev, and S. Yanushkevich, AND/EXOR Minimization of Switching Functions Based on Information Theoretic Approach, *Facta Universitatis*, Vol. 13, No. 1, April 2000, pp. 11–25.

[5] V. Cheushev, S. Yanushkevich, C. Moraga. V. Shmerko, Flexibility in Logic Design. An Approach Based on Information Theory Methods, *Research Report 741*, Dept. Computer Science, Dortmund University, 2000, 50 p.

[6] V. Cheushev, S. Yanushkevich, V. Shmerko, C. Moraga, J. Kolodzieczyk, Remarks on Circuit Verification Trough the Evolutionary Circuit Design, *Proc.* 31^{st} *IEEE Int. Symp. on Multiple-Valued Logic*, 2001, pp. 201–206.

[7] C. Moraga, V. Cheushev, R. Stanković, Circuits Matching Using Information Measures, *Proc. of a Workshop on Computational Intelligence and Information Technologies*, Niš, Yugoslavia, June 20–21, 2001, pp. 91–95.

[8] S. Muroga, Y. Kambayashi, H.C. Lai, J.N. Culliney. The transduction method-design of logic networks based on permissible function, *IEEE Trans. on Computer Aided Design*, 1989, vol. 38, no. 10. pp. 1404–1424.

[9] S. Yamashita, H. Sawada, A. Nagoya, SPFD: A Method to Express Functional Flexibility, *IEEE Trans. on Computer Aided Design*, 2000, vol. 19. no. 8, pp. 840–849.