Proceedings of the 2005 International Conference on Foundations of Computer Science (FCS'05), pp. 24-28, Las Vegas, Nevada, June, 2005

K-Map Analysis of Digital Circuits with Uncertain Inputs

Houssain Kettani
Department of Computer Science
Jackson State University
Jackson, MS 39217
houssain.kettani@jsums.edu

Abstract

Unlike the classical deterministic digital circuit analysis, we consider the analysis of digital circuits with uncertain inputs. Thus, given a binary function of n uncertain input binary variables, we introduce a new way of using K-maps to express the probability of this binary function in terms of the probabilities of the corresponding input binary variables. This in turn, allows us to estimate appropriate probabilistic measure of the output of a digital circuit with uncertain input parameters and answer typical questions that arise in stochastic optimization.

1 Introduction

The theory of deterministic digital circuits has been studied extensively — See [3] for instance. Typically, a binary variable x_j is allowed to take only two values; 0 and 1. A binary function of n binary variables $f(x_n, x_{n-1}, \ldots, x_1)$, is also allowed to take only the values 0 and 1. These values may stand for false and true, or low and high. The latter is implemented through voltage levels; low for low voltage level, and high for high voltage level. We typically have a threshold level that separates the two levels to avoid uncertainty.

However, due to noise, temperature variation, signal delay, and other factors, voltage level can be a random variable. Thus, it makes sense to consider the case when the binary variable x_j is a random variable too. Furthermore, if we consider a binary function of n binary variables $f(x_n, x_{n-1}, \ldots, x_1)$, then this in turn would be a random variable.

The introduction of uncertainty in digital circuits has been used in different areas to model complex systems. For example, see [4] for the introduction of probabilistic Boolean networks to model gene regulatory networks. In such a model, the x_j's represent the state of gene j, where $x_j = 1$ denotes the fact that gene j is expressed and $x_j = 0$ means it is not expressed. The binary function $f_j(x_n, x_{n-1}, \ldots, x_1)$, on the other hand, is referred to as a predictor, and is used to determine the value of x_j in terms of some other gene states.

Analysis of digital circuits with uncertain inputs was introduced in [2]. In such analysis, multi-input mono-output digital networks as depicted in Figure 1 was studied. The case when the binary variables x_j, $j = n, n-1, \ldots, 1$, are random was considered. Thus, a binary function of the n random binary variables, $f(x_n, x_{n-1}, \ldots, x_1)$, is a Bernoulli random variable. A formula expressing the probability of the output in terms of the probabilities of the input variables was presented in [2]. In this paper, we accomplish this task by presenting a practical algorithm that exploits K-maps.

Hence, Throughout this paper, we consider the case when the x_j's are independent random variables with probabilities

$$P(x_j = 1) \doteq p_j = E[x_j].$$

Next, we consider the probability or expectation

$$\begin{aligned} \mathcal{P} &\doteq P(f(x_n, x_{n-1}, \ldots, x_1) = 1) \\ &= E[f(x_n, x_{n-1}, \ldots, x_1)]. \end{aligned}$$

We then pose and answer the following questions: Given a binary logic function, $f(x_n, x_{n-1}, \ldots, x_1)$, with known probabilities p_j's, what can we say about the probability $\mathcal{P}$?

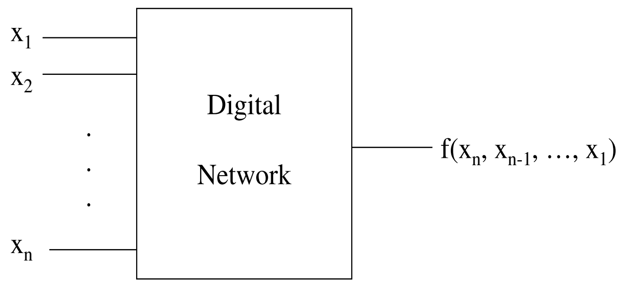

Figure 1. Network Configuration

The flow of this paper is as follows. In Section 2, we present and prove a result that expresses the probability $\mathcal{P}$ in terms of the probabilities p_j's, with $j = n, n-1, \ldots, 1$. To accomplish this goal, an algorithm that uses K-maps is introduced in Section 3. We end this paper by presenting a summary in Section 4.

2 Stochastic Measures

Suppose the binary variables x_j's, where $j = n, n-1, \ldots, 1$, are independent random variables with given probabilities

$$P(x_j = 1) \doteq p_j = E[x_j].$$

Given a binary logic function of the n binary random variables, $f(x_n, x_{n-1}, \ldots, x_1)$, let us define

$$\begin{aligned} \mathcal{P} &\doteq P(f(x_n, x_{n-1}, \ldots, x_1) = 1) \\ &= E[f(x_n, x_{n-1}, \ldots, x_1)]. \end{aligned}$$

Clearly $\mathcal{P}$ is a function of the x_j's. It would be of interest, however, to know the exact form of this function. Such function was presented in [2], where also the questions of maximizing it or minimizing it were addressed. We present the theorem here together with its proof for the sake of completeness.

2.1 Theorem

Let $f(x_n, x_{n-1}, \ldots, x_1)$ be a binary function of n independent binary random variables with $P(x_j = 1) = p_j$. Let I be the set of minterm indices for which $f(x_n, x_{n-1}, \ldots, x_1)$ is 1. Then

$$\mathcal{P} = \sum_{i \in I} \prod_{j=1}^{n} P(x_j = \lfloor i2^{-j+1} \rfloor - 2\lfloor i2^{-j} \rfloor). \quad (1)$$

2.2 Proof of Theorem

We devote this section to proving Theorem 2.1. Let $f(x_n, x_{n-1}, \ldots, x_1)$ be a binary function of n variables. Let I be the set of minterm indices for which $f(x_n, x_{n-1}, \ldots, x_1)$ is 1. Clearly we have $0 \leq i < 2^n$ for $i \in I$. Now note that any binary function can be written as a sum of its minterms — see [3] for details. Thus, we write

$$f(x_n, x_{n-1}, \ldots, x_1) = \sum_{i \in I} m_i.$$

Let us now write $(i)_{10} = (i_n i_{n-1} \dots i_1)_2$, where $i_j \in \{0, 1\}$. Hence, we have

$$m_i = x_n^{i_n} x_{n-1}^{i_{n-1}} \dots x_1^{i_1} = \prod_{j=1}^{n} x_j^{i_j},$$

where we adopt the notion that $x_i^0 = \overline{x}_j$; which is the complement of x_j, and $x_j^1 = x_j$. Suppose now that $P(x_j = 1) = p_j$, and let

$$\mathcal{P} \doteq P(f(x_n, x_{n-1}, \dots, x_1) = 1).$$

Now, since the events $\{m_i = 1\}$ are mutually exclusive, we have

$$P(f(x_n, x_{n-1}, \dots, x_1) = 1) = \sum_{i \in I} P(m_i = 1).$$

Next, note that

$$P(m_i = 1) = \prod_{j=1}^{n} P(x_j = i_j),$$

since x_j's are independent. Now, by applying the base conversion theorem [1], we have

$$i_j = \lfloor i 2^{-j+1} \rfloor - 2 \lfloor i 2^{-j} \rfloor.$$

Thus, (1) follows and this concludes the proof.

3 Application of K-Maps

The result of Theorem 2.1 gives a general formula to relate $\mathcal{P}$ to the probabilities p_i. However, it would be of interest to have a simpler method to come up with such formula. Fortunately this can be done by implementing Karnaugh maps.

Karnaugh maps or K-maps for short, were invented by Maurice Karnaugh, while he was working as a telecommunications engineer at Bell Labs in 1953. He was studying the application of digital logic to the design of telephone circuits.

A K-map is a graphical tool to reduce a Boolean function to its simplest expression without the need to go through the headache of Boolean algebra. A K-map is a reorganization of the truth table in a table of squares or cells. Each cell represents a minterm. So a K-map for n variables would have 2^n cells. See [3] for more details on K-maps.

We say that two cells in a K-map are *adjacent* if they share the same line. The number of cells combined for a K-map of n variables are in $2^m, m = 0, 1, \dots, n$. The larger the number the better, since it would produce a simpler Boolean expression of the Boolean function in question. The objective is to have a collection of the largest groupings of adjacent squares that include 1. 2^m such adjacent cells will give a product term of $(n - m)$ literals. So one cell that has a value one will give a product term of n literals which is a minterm. Similarly, 2^n adjacent cells that have the value one will give a product term of 0 literals or a constant 1, which means that the function in question is equal to 1.

We can use K-maps to analyze the type of circuits described in this paper. From the proof of Theorem 2.1 we can see that one step of the proof required that the events $\{m_i = 1\}$ be mutually exclusive. This can be generalized to require the events of product terms equal one be mutually exclusive instead. Consequently, this can be explained K-maps as expanding the groups of cells as described by Karnaugh but not overlap these groups.

3.1 Algorithm

In summary, to find the probability $\mathcal{P}$ of a given binary function f, we introduce the following algorithm:

- Gather groups of 2^m, $m = 0, 1, \dots, n$, adjacent cells with the largest possible m such that such groups do not overlap;
- Write down the corresponding expression of the binary function f; then

- The expression of $\mathcal{P}$ is similar to that of f with each x_i is replaced by p_i and each $\overline{x}_i$ is replaced by $(1 - p_i)$.

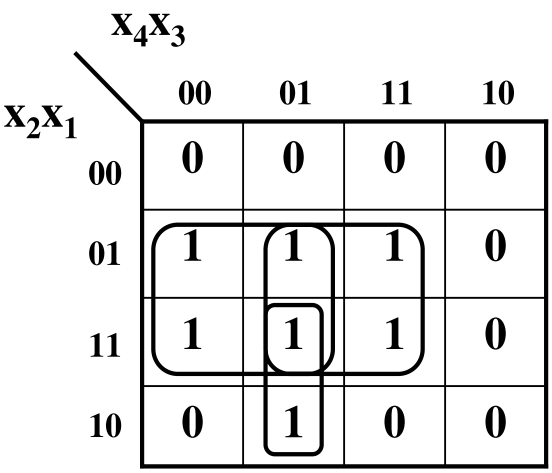

Figure 2. Optimal realization of the function in example 3.2

3.2 Illustrative Example

To illustrate the ideas of this section, we consider the following function:

$$f(x_4, x_3, x_2, x_1) = \sum m(1, 3, 5, 6, 7, 13, 15).$$

As illustrated in Figure 2, the minimum sum-of-product form for this function is

$$f(x_4, x_3, x_2, x_1) = \overline{x}_1 x_4 + x_2 x_4 + \overline{x}_1 x_2 x_3.$$

Now, to find $\mathcal{P}$, we use the K-map in Figure 3. We gather the maximum grouping of adjacent cells without overlapping them. Thus, we can rewrite f as the sum of product terms that are mutually exclusive as follows

$$f(x_4, x_3, x_2, x_1) = x_2 x_4 + \overline{x}_1 \overline{x}_2 x_4 + \overline{x}_1 x_2 x_3 \overline{x}_4.$$

Hence we get

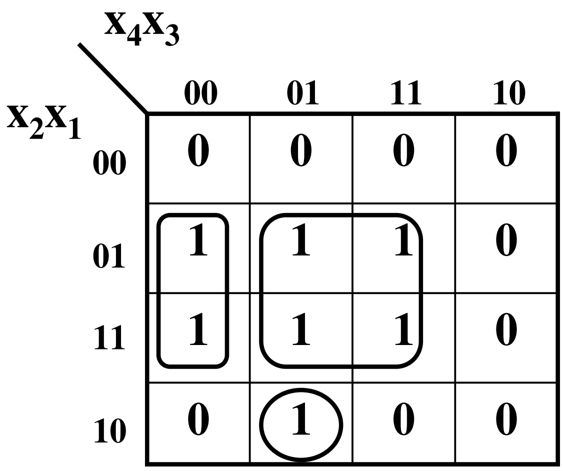

Figure 3. Optimal realization of the function in example 3.2 to find the corresponding $\mathcal{P}$

$$\begin{aligned}\mathcal{P} &= p_2 p_4 + (1 - p_1)(1 - p_2) p_4 \\ &\quad + (1 - p_1) p_2 p_3 (1 - p_4) \\ &= p_4 + p_2 p_3 - p_1 p_4 + p_1 p_2 p_4 - p_2 p_3 p_4 \\ &\quad - p_1 p_2 p_3 + p_1 p_2 p_3 p_4.\end{aligned}$$

Note that other variations in the selection of the non-overlapping collections of adjacent squares will result in the same $\mathcal{P}$.

4 Summary

We considered the case of digital circuits with uncertain input variables. We have presented an algorithm that uses K-maps to analyze such circuits. This algorithm presents the probability of the output binary function n terms of the probabilities of the input binary variables.

References

[1] H. Kettani (2004), "*On the Conversion Between Number Systems,*" proceedings of the 2004 International Conference on Algorithmic Mathematics and Computer Sci-

ence (AMCS'04), pp. 317-320, Las Vegas, Nevada, June, 2004.

[2] H. Kettani (2004), *"On the Analysis of Digital Circuits with Uncertain Inputs,"* proceedings of the 2004 Annual IASTED International Conference on Circuits, Signals, and Systems (CSS 2004), pp. 90-95, Clearwater Beach, Florida, November, 2004.

[3] M. M. Mano and C. R. Kime (2001), *"Logic and Computer Design Fundamentals,"* second edition, Prentice Hall.

[4] I. Shnulevich, E. R. Dougherty, S. Kim, and W. Zhang (2002), *"Probabilistic Boolean Networks: A Rule-Based Uncertainty Model for Gene Regulatory Networks,"* Bioinformatics, Vol. 18, pp. 261–274.