数理解析研究所講究録
第716巻 1990年 101-116

On the Complexity of Three-Level Logic Circuits

Tsutomu SASAO
笹尾　勤

Department of Computer Science and Electronics
Kyushu Institute of Technology, Iizuka 820, Japan
九州工業大学情報工学部

phone　+81-948-28-5551 ext.323 (office)　+81-92-716-1417 (home)
fax　+81-948-28-5582 (office)　+81-92-722-0507 (home)

Abstract: This paper considers the number of gates to realize logic functions by OR-AND-OR three-level circuits under the condition that both true and complemented variables are available, and each gate has no fan-in and fan-out constraints. We show that an arbitrary n-variable function can be realized by an OR-AND-OR three-level circuit with at most $2^{r+1}+1$ gates, where n=2r and r are integers. We also prove that for sufficiently large n, regardless of the number of levels, we need at least $2^{r+1}(1-\xi)$ gates to realize almost all functions of n variables by an AND-OR multi-level circuit, where ξ is an arbitrarily small positive number. We developed a heuristic algorithm to design OR-AND-OR three-level circuits, realized various functions, and compared the number of gates for three-level circuits with two-level ones.
For arithmetic functions of 8 variables, three-level circuits require, on the average, 40% fewer gates than AND-OR two-level ones. For control functions of 13 to 83 variables, three-level circuits required 20% fewer gates. For randomly generated functions of 10 variables, three-level circuits required 50% fewer gates.

I. Introduction

In this paper, we consider the number of gates to realize arbitrary functions by AND-OR multi-level circuits under the condition that both true and complemented variables are available as inputs, and each gate has unlimited fan-in and fan-out.

An arbitrary logic function can be realized by an AND-OR two-level circuit. In this case, $2^{n-1}+1$ gates are necessary and sufficient to realize a parity function of n variables, which requires the largest number of gates in two-level realizations. In a similar way, an arbitrary function can be realized by multi-level circuits such as OR-AND-OR three-level circuits, AND-OR-AND-OR

four-level circuits, etc. Then, how many gates are necessary to realize an arbitrary function by multi-level circuit? And, which function requires the largest number of gates in multi-level realizations? This paper considers the number of gates to realize arbitrary functions, and try to find the most complex function in multi-level circuits.

First, we show that an arbitrary function of n variables (n=2r) can be realized by an OR-AND-OR three-level circuits by using at most $2^{r+1}+1$ gates. Second, we prove that, regardless of the number of levels, we need at least $2^{r+1}(1-\xi)$ gates to realize almost all functions of n variables by multi-level AND-OR circuits, where n is a sufficiently large integer, and ξ is an arbitrarily small positive number. From these facts, we know that there is a distinct difference between two-level circuits and three-level ones, but not so much difference between three-level ones and multi-level ones with more than three levels.

This paper is organized as follows:

In II, we will show that an arbitrary n-variable function (n=2r) can be realized with an OR-AND-OR three-level circuit with at most $2^{r+1}+1$ gates.

In III, we introduce the concept of almost all functions and prove that we need at least $2^{r+1}(1-\xi)$ gates to realize almost all functions of n variables by multi-level AND-OR circuits.

In IV, we show the number of gates to realize various functions obtained by computer experiments.

II. Upper Bound on the Number of Gates

In this section, we derive the upper bound on the number of gates to realize an arbitrary n-variable function by OR-AND-OR three-level circuits.

Theorem 2.1: An arbitrary n-variable function (n=2r) can be realized by an OR-AND-OR three-level circuit with at most $2^{r+1}+1$ gates.

(Proof) An arbitrary n-variable function is represented by

$$f(x_1, x_2, \ldots, x_n) = \bigvee g_{\underline{a}}(x_1, x_2, \ldots, x_k) x_{k+1}^{a_{k+1}} x_{k+2}^{a_{k+2}} \cdots x_n^{a_n}, \quad \text{---(2.1)}$$

where $1<k<n$, and $a_i=0$ or 1 $(i=k+1, k+2, \ldots, n)$, $x_i^0=\bar{x}_i$, $x_i^1=x_i$, $g_{\underline{a}}(x_1, x_2, \ldots, x_k)=f(x_1, x_2, \ldots, x_k, a_{k+1}, a_{k+2}, \ldots, a_n)$, and $\underline{a}=(a_{k+1}, a_{k+2}, \ldots, a_n)$. Note that the logical sum is taken for 2^{n-k} different combinations. $g_{\underline{a}}(x_1, x_2, \ldots, x_k)$ can be represented by a canonical product-of-sums expression:

$g_{\underline{a}}(x_1, x_2, \ldots, x_k) = \bigwedge [g_{\underline{a}}(b_1, b_2, \ldots, b_k) \vee x_1^{\bar{b}_1} \vee x_2^{\bar{b}_2} \vee \ldots \vee x_k^{\bar{b}_k}]$,

where $g_{\underline{a}}(b_1, b_2, \ldots, b_k)$ is a binary constant, $\underline{a} \in B^k$, $b_j \in B$ $(j=1, 2, \ldots, k)$, and $B=\{0, 1\}$. A circuit which generates all the maxterms of the k-variable is called a k-bit decoder. By using a k-bit decoder and 2^{n-k} AND gates, we can realize 2^{n-k} functions as follows:

$g_{\underline{a}}(x_1, x_2, \ldots, x_k)\, x_{k+1}^{a_{k+1}} x_{k+2}^{a_{k+2}} \ldots x_n^{a_n}$, where $a_i=0$ or 1 for $i=k+1, \ldots, n$.

By summing these functions by an OR gate, we have a circuit for the function f. The designed circuit contains 2^k OR gates for the k-bit decoder, 2^{n-k} AND gates, and one OR gate. So the total number of gates is $\alpha(n) = 2^k + 2^{n-k} + 1$. Let $k=n/2$, and we have $\alpha(n) = 2^{r+1} + 1$. Hence, we have the theorem. (Q.E.D)

Example 2.1: The function shown in Fig. 2.1 can be represented as

$f = g_{(0,0)} \bar{x}_3 \bar{x}_4 \vee g_{(0,1)} \bar{x}_3 x_4 \vee g_{(1,0)} x_3 \bar{x}_4 \vee g_{(1,1)} x_3 x_4$,

where $g_{(0,0)}$, $g_{(0,1)}$, $g_{(1,0)}$, and $g_{(1,1)}$ are represented by canonical product-of-sums expression :

$$g_{(0,0)} = x_1 x_2 \vee \bar{x}_1 \bar{x}_2 = (\bar{x}_1 \vee x_2)(x_1 \vee \bar{x}_2),$$

$$g_{(0,1)} = x_1 \vee \bar{x}_2 = (x_1 \vee \bar{x}_2),$$

$$g_{(1,0)} = \bar{x}_1 = (\bar{x}_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2), \text{ and}$$

$$g_{(1,1)} = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 = (\bar{x}_1 \vee \bar{x}_2)(x_1 \vee x_2).$$

Therefore, the given function is represented by

$$f = (\bar{x}_1 \vee x_2)(x_1 \vee \bar{x}_2)\bar{x}_3 \bar{x}_4 \vee (x_1 \vee \bar{x}_2)\bar{x}_3 x_4 \vee (\bar{x}_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2) x_3 \bar{x}_4 \vee (\bar{x}_1 \vee \bar{x}_2)(x_1 \vee x_2) x_3 x_4.$$

Fig. 2.2 shows the realization of this function. (End of Example).

Theorem 2.2: Let $(n + \log_2 m)/2$ and $(m)^{1/2}$ be integers. Then, an arbitrary n-input m-output function can be realized by an OR-AND-OR three-level circuit with at most $(m)^{1/2} \cdot 2^{(n/2)+1} + m$ gates.

(Proof) Consider a set of m expressions like (2.1), and use k-bit decoders. Then, the total number of AND gates is at most $m \cdot 2^{n-k}$. We need m OR gates for m outputs. Therefore, the total number of gates is

$$m \cdot 2^{n-k} + 2^k + m \quad \text{----(2.3)}$$

Let $k=(n+\log_2 m)/2$, and (2.3) becomes $m^{(1/2)} \cdot 2^{(n/2)+1} + m$.

Hence, the theorem. (Q.E.D.)

III. Lower Bound on the Number of Gates

It is well known that, among n-variable functions, the parity functions of n variables are the most complex ones to realize in two-level circuits. In fact, the parity functions of n variables require $2^{n-1}+1$ gates, and other functions require the same numeber of gates or less. So the lower bound on the number of gates to realize arbitrary function of n variables in two-level circuit is $2^{n-1}+1$. Then, the problem arises how many gates are necessary to realize an arbitrary function of n variables by multi-level circuits. To answer this problem, we have to find an n-variable function rquiring the largest number of gates in multi-level realization. Unfortunately, it is extremely difficult to find such a function. This is partly due to the fact that the problem to realize a particular function by a multi-level circuit using the least number of gates is extremely difficult. So we use the concept of almost all functions, which was used to derive the complexity of series-parallel relay circuits [SHA 49].

Definition 3.1: A property A is said to hold for almost all functions if the proportion of n-variable functions which do not satisfy A tends to zero as $n\to\infty$. In other words, let w(n) be the number of n-variable functions which do not satisfy A . Then $w(n)/2^{2^n} \to 0$ as $n\to\infty$.

From here, we will show that we need at least $2^{r+1}(1-\xi)$ gates to realize almost all function of n variables, where n=2r. We will not show that any particular function requires $2^{r+1}(1-\xi)$ gates, but it is impossible for almost all functions to require less. In other words, there exist not enough circuits to represent almost all n variable function with less than $2^{r+1}(1-\xi)$ gates. To enumerate the number of meaningful circuits only, we will define a class of AND-OR multi-level circuits which do not contain a certain type of redundant connections.

Definition 3.2: A multi-level circuit consisting of AND and OR gates is said to be normal if no output of any AND (OR) gate is connected to the input of other AND (OR) gates.

Lemma 3.1: An arbitrary AND-OR multi-level circuit can be converted into a normal one without increasing the number of gates.

(Proof) If the output of an AND gate A is connected to an input of other AND gate B (Fig. 3.1), then delete the connection between A and B, and connect all the input signals of A to the input of B in addition to the original signals (Fig. 3.2). Apply this operation to all AND gates. Apply the similar operations to all OR gates. And, we have a normal AND-OR multi-level circuit.

It is clear that these operations do not increase the number of gates. (Q.E.D.)

Lemma 3.2: Let $\mu(n,m,N)$ be the number of different n-input m-output normal AND-OR multi-level circuits with at most N gates. Then

$$\mu(n,m,N) \leqq (3^{nN})\cdot(2^{N^2/4}).$$

(Proof) Consider normal AND-OR multi-level circuits with p levels. Assume that all the output functions are obtained from OR gates. Suppose that p is an odd number. In this case, the first-level gates (ones which are farthest from the outputs) are ORs, the second-level ones are ANDs, third-level ones are ORs,..., the (p-1)-th ones are ANDs, and the p-th (output) ones are ORs, as shown in Fig. 3.2. Let a_i be the number of i-th level gates (i=1, 2,..., p).

Because the total number of gates is at most N, we have $\sum_{i=1}^{p} a_i \leqq N$.

First, consider the number of different patterns of connections in an OR gate in the first level. To each OR gate, either a true variable is connected, the complement is connected , or neither of them is connected, for each variable x_i. So, three possible cases exist for each variable x_i. Therefore, the number of different patterns of connections in an OR gate in the first level is 3^n.

Second, consider the number of different patterns of connections in an AND gate in the second level. To each AND gate, either x_i or $\bar{x}_i$ or none of them is connected for each x_i. Also, either an output of an OR gate in the first level is connected or unconnected. Therefore, the number of different patterns of connections in an AND gate in the second level is $3^n \cdot 2^{a_1}$.

Third, consider the number of different patterns of connections in an OR gate in the third level. To each OR gate, either x_i or $\bar{x}_i$ or none of them is connected for each x_i. Also, either an output of an AND gate in the second level is connected or unconnected. Therefore, the number of different patterns of connections in an OR gate in the third level is $3^n \cdot 2^{a_2}$.

Lastly, consider the number of different patterns of connections in an OR gate in the p-th level (output). To each output OR gate, either the outputs of the AND gates in the 2nd, 4-th,..., and (p-1)-th level are connected or unconnected, in addition to the patterns of an OR gate in the first level. Therefore, the number of different patterns of connections in the output

OR gate is $3^n \cdot 2^{(a_2+a_4+\ldots+a_{p-1})}$

So, the numbers of different patterns of connections in the gates are :

First level	3^n
Second level	$3^n \cdot 2^{a_1}$
Third level	$3^n \cdot 2^{a_2}$
Fourth level	$3^n \cdot 2^{(a_1+a_3)}$
Fifth level	$3^n \cdot 2^{(a_2+a_4)}$
Sixth level	$3^n \cdot 2^{(a_2+a_4+a_6)}$

(p-1)th level	$3^n \cdot 2^{(a_1+a_3+\ldots+a_{p-2})}$
p-th level	$3^n \cdot 2^{(a_2+a_4+\ldots+a_{p-1})}$

Therefore, the total number of different connections in the circuit is

$$\mu(n,m,N) = (3^n)^{a_1} \cdot (3^n \cdot 2^{a_1})^{a_2} \cdot (3^n \cdot 2^{a_2})^{a_3} \cdot (3^n \cdot 2^{(a_1+a_3)})^{a_4} \cdot (3^n \cdot 2^{(a_2+a_4)})^{a_5}$$

$$\ldots\ldots \cdot (3^n \cdot 2^{(a_2+a_4+\ldots+a_{p-1})})^{a_p}.$$

$$= 3^{n(a_1+a_2+\ldots+a_p)} \cdot 2^{\{a_1a_2 + a_2a_3 + (a_1+a_3)a_4 + \ldots + (a_2+a_4+\ldots+a_{p-1})a_p\}}$$

In the above, the expression inside of { } is simplified as follows:

$$a_1a_2 + a_2a_3 + (a_1+a_3)a_4 + (a_2+a_4)a_5 + (a_1+a_3+a_5)a_6 + (a_2+a_4+a_6)a_7$$

$$+\ldots+(a_1+a_3+a_5+\ldots+a_{p-2})a_{p-1} + (a_2+a_4+\ldots+a_{p-1})a_p$$

$$= (a_1+a_3+a_5+\ldots+a_p)\cdot(a_2+a_4+a_6+\ldots+a_{p-1}) = N_1 \cdot N_2 \text{ , where}$$

$N1 = a_1+a_3+a_5+\ldots+a_p$ and $N2 = a_2+a_4+a_6+\ldots+a_{p-1}$

Because, $N1 \cdot N2 \leqq (N1+N2)^2/4 = N^2/4$, we have $\mu(n,m,N) \leqq (3^{nN}) \cdot (2^{(N^2/4)})$. Thus, we have the lemma when p is an odd number. When p is an even number, we can prove it in a similar way. Hence the theorem. (Q.E.D.)

Definition 3.3: Let $\eta(n,m)$ be the necessary and sufficient number of gates to realize an arbitrary n-input m-output function by a multi-level AND-OR circuit.

Theorem 3.1: For an arbitrary positive small number ξ, there exists a positive integer $n(\xi)$ satisfying the following condition:
when $n \geqq n(\xi)$, $\eta(n,m) > m^{(1/2)} \cdot 2^{(n/2)+1} \cdot (1-\xi)$.
(Proof) The number of n-input m-output function realized by AND-OR multi-level circuits with at most N gates does not exceed $\mu(n,m,N)$, the number of normal AND-OR multi-level circuits with at most N gates.

There are 2^{m2^n} different n-input m-output functions.

From here, we will show that if $N=(m)^{1/2} \cdot 2^{(n/2)+1} \cdot (1-\xi)$

then $\mu(n,m,N)/(2^{m2^n}) < 1$, for sufficiently large n.
By taking the logarithm of the left hand side of the inequality, we have

$$\log[\mu(n,m,N)/(2^{m2^n})]$$
$$\leqq nN(\log 3) + N^2(\log 2)/4 - m2^n(\log 2)$$
$$= n\{(m)^{1/2} \cdot 2^{(n/2)+1} \cdot (1-\xi)\}(\log 3) + m \cdot 2^{n+2} \cdot (1-\xi)^2 \cdot (\log 2)/4 - m \cdot 2^n \cdot (\log 2)$$
$$= n\{(m)^{1/2} \cdot 2^{(n/2)+1} \cdot (1-\xi)\}(\log 3) + \{m \cdot 2^n(-2\xi+\xi^2)\}(\log 2)$$
$$= m \cdot 2^n(\log 2)\{n \cdot (m)^{-1/2} \cdot 2^{-(n/2)+1} \cdot [(\log 3)/(\log 2)](1-\xi) - \xi(2-\xi)\}$$
$$= m \cdot 2^n(\log 2)[2 \cdot A(1-\xi) - \xi(2-\xi)], \qquad \text{------(3.1)}$$

where $A = \dfrac{n}{m^{(1/2)} \cdot 2^{(n/2)} \cdot (\log 2)} \cdot (\log 3)$

The above expression becomes negative when

$$2 \cdot A < \frac{\xi(2-\xi)}{(1-\xi)} = \xi + \xi(1+\xi^2+\xi^3+\ldots) = 2\xi + \xi^2 + \xi^3 + \ldots$$

Dropping the higher order terms $\xi^2+\xi^3+\ldots$, we have $A<\xi$. In this case, the logarithm still remains negative. In other words, there exists $n(\xi)$, and if $n > n(\xi)$, then not all the functions can be realized by using at most $N = m^{(1/2)} \cdot 2^{(n/2)+1} \cdot (1-\xi)$ gates. Hence the theorem. (Q.E.D.)

Theorem 3.2: Let ξ be an arbitrary small positive number. In order to realize almost all functions of n variables,

$N=(m)^{1/2} \cdot 2^{(n/2)+1} \cdot (1-\xi)$ gates are necessary.

(Proof) Let ξ be a constant such that $0<\xi<1$, then the left hand side of (3.1) in Theorem 3.2 tends to $-\infty$ as $n \to \infty$. Therefore,

$$\log[\mu(n,m,N)/2^{(m2^n)}] \to -\infty \text{ as } n \to \infty.$$

In other words,

$$\mu(n,m,N)/2^{(m2^n)} \to 0 \text{ as } n \to \infty.$$

Thus the fractions of functions realized by N or less gates tends to zero as $n \to \infty$. This implies we need at least N gates to realize almost all functions. (Q.E.D.)

Corollary 3.1: Regardless of number of levels, AND-OR multi-level circuits require at least $2^{r+1}(1-\xi)$ gates to realize almost all functions of n variables, where $n=2r$ and ξ is an arbitrary small positive number.
(Proof) From Theorem 3.2. (Q.E.D.)

Corollary 3.2: For almost all functions, there is no design method for AND-OR multi-level circuits which produces much better circuits than the one in Theorem 2.1. (Proof) By Theorems 2.2 and 3.2. (Q.E.D.)

Thus, for almost all functions, it is sufficient to consider the AND-OR circuits of levels not greater than three. Hence, three-level OR-AND-OR circuits (and their duals) are very important multi-level circuits.

Note that there exists some class of functions whose minimum AND-OR multi-level realizations require many fewer gates than ones obtained by the method in Theorem 2.1. Such a class of functions include parity functions, partially symmetric functions and adders. Of course, the fraction of such functions tends to zero as $n \to \infty$. Table 3.1 compares the numbers of gates to realize various classes of functions.

From here, we will show the number of gates to realize almost all functions by AND-OR two-level circuits.

Lemma 3.3: The number of different n-variable AND-OR two-level circuits using at most N gate is at most 3^{nN}.
(Proof) Consider the number of different patterns of connections in an AND gate. To each AND gate, either a true variable is connected, the complement is connected, or neither of them is connected for each variable. So, three possible cases exist for each variable. Therefore, the number of different patterns of connections in an AND gate is 3^n. Hence, the total number of different patterns of connections is 3^{nN}.

Theorem 3.3: For sufficiently large n, almost all functions of n variables require at least $(\log_3 2)\cdot(1-\xi)2^n/n$ gates in AND-OR two-level circuits, where ξ is an arbitrarily small positive number.
(Proof) The number of n-variable function realized by AND-OR two-level circuits with at most N gates does not exceeds the number of different n-variable AND-OR circuits with N gates. From here, we will show that if $N=(\log_3 2)\cdot(1-\xi)2^n/n$ then the fraction of functions requiring at most N

gates approaches to zero as $n \to \infty$. i.e., $\gamma = 3^{nN}/(2^{2^n}) \to 0$.
By taking the logarithm of γ, we have

$\log_2 \gamma = nN(\log_2 3) - 2^n = (\log_3 2)(\log_2 3)(1-\xi)2^n - 2^n = -\xi 2^n$.

Therefore, $\gamma \to 0$ as $n \to \infty$. Hence, we have the theorem. (Q.E.D.)

From Theorems 2.1 and 3.3, we see that OR-AND-OR circuits require many fewer gates than AND-OR circuits to realize almost all functions.

IV. Experimental Results

We developed an algorithm to obtain OR-AND-OR circuits with near minimum number of gates[SAS 88]. We coded it in FORTRAN and implemented it on a Sun 3/50 workstation. We designed OR-AND-OR three-level circuits, and compared the number of gates with AND-OR two-level ones. Table 4.1 shows the number of gates for 9 arithmetic circuits. This results shows that the OR-AND-OR circuits require, on the average, 40% fewer gates than AND-OR circuits. For example, to realize WGT8 (which is also called as RD84), 259 gates are necessary for a two-level circuit, but only 49 gates for a three-level circuit.

We also designed 18 control circuits by OR-AND-OR three-level ones. The original circuits are AND-OR two-level with 13 to 83 inputs, 6 to 94 outputs, and 63 to 637 products. On the average, OR-AND-OR circuits require 20% fewer gates than two-level AND-OR circuits. In particular, in the case of TIALU, the three-level circuit required 57% fewer gates than two-level one. Table 4.2 shows the part of the results.

We also realized randomly generated functions of 10-variables. Table 4.3 shows average numbers of gates to realize functions with various number of minters.

Prof. Muroga's group obtained all the optimum AND-OR multi-level circuits for functions of four or fewer variables[CUL 79]. Their optimality means minimization of the number of gates as the primary objective and the number of connections as the secondary objective, regardless of the number of levels. They obtained optimum circuits for 222 representative functions of NPN-equivalence classes of four or fewer variables by the branch and bound method. Their result is very interesting: Of all the representative functions, only two representative functions have optimal circuits exclusively of four levels. No representative functions have optimum circuits of five or more levels. In other words, in the case of functions with four or fewer variables, most optimum circuits are three or less levels.

V. Conclusion and Comments

In this paper, we considered the number of gates to realize arbitrary function by OR-AND-OR three-level circuits under the condition that fan-in and fan-out of the gates are unlimited. We showed that there exists a distinct difference between multi-level realizations and two-level realizations. In two-level realization, we need $2^{n-1}+1$ gates to realize arbitrary n-variable functions, while in multi-level realization, we need $2^{(n/2)+1}(1-\xi)$ gates. We have the following results for almost all functions:

1. OR-AND-OR circuits require many fewer gates than AND-OR two-level circuits. However, four or more level AND-OR circuits require the same order of gates as OR-AND-OR circuits.
2. No algorithm produces circuits with many fewer gates than one in this paper.

We developed a heuristic algorithm for OR-AND-OR three-level circuits, designed many arithmetic and control circuits, and compared the number of gates for two level circuits with three-level ones. In the case of the arithmetic circuits, three-level realizations required 40% fewer gates than two-level ones. In the case of the control circuits, three-level ones required 20% fewer gates.

In this paper, we assume that each gate has unlimited fan-in. As an example of circuits where this assumption holds is a NAND array shown in Fig. 5.1. This is a programmable logic devices (PLDs) commercially available [SIG 86]. To realize a given function, first, design a three-level OR-AND-OR circuit and then transform it into a NAND three-level circuit. The NAND three-level circuit can be obtained by properly programming the NAND array. In this case, the fan-in of each gate is sufficiently large.

However, we cannot apply the present method to the circuits where each gate has fan-in limitation, such as ones in gate array LSIs. Most multi-level synthesis algorithms assume that each gate has fan-in limitation [KAR 87]. In such a case, the number of gates necessary to realize an n-variable function is proportional to $2^n/n$ [MUL 56].

Acknowledgement

The author thanks to M. Higashida, who made the minimizatioin program. [RBU 87] and [HAS 87] were mentioned by I. Stojmenovic and O. Watnabe, respectively.

VI. References

[BRU 87] B. Brustmann and I. Wegener, "The complexity of symmetric functions in bounded-depth circuits," Information Processing Letters 25, pp. 217-219, 1987.

[CUL 79] J. N. Culliney, M. H. Young, T. Nakagawa, and S. Muroga, "Results of the synthesis of optimal networks of AND and OR gates for four-variable switching functions," IEEE Trans. on Comput. Vol. C-27, No. 1, pp. 76-85, Jan. 1979.

[HAS 87] J. T. Hastad, "Computationnal Limitations for small-depth circuits," The MIT Press, Cambridge, 1989.

[KAR 87] M. Karpovsky, "Multilevel logical networks," IEEE Trans. on Comput., Vol. C-36, No. 2, pp. 215-226, Feb, 1987.

[LAW 64] E. L. Lawler, "An approach to multilevel Boolean minimization," Journal of ACM, Vol. 11, No. 3, pp. 283-295, July 1964.

[MUL 56] D. E. Muller, "Complexity in electronic switching circuits," IRE Trans. Elec. Computers, EC-5, 1, pp. 15-19, 1956.

[PAP 77] C. A. Papachristou, "Characteristic measure of switching functions," Information Sciences, 13, pp. 51-75, 1977.

[SAS 81] T. Sasao, "Multiple-valued decomposition of generalized Boolean functions and the complexity of programmable logic arrays," IEEE Trans. Comput. Vol. C-30, pp. 635-643, Sept. 1981.

[SAS 84] T. Sasao, "Input variable assignment and output phase optimization of PLA's," IEEE Trans. Comput. vol. C-33, No. 10, pp. 879-894, Oct. 1984.

[SAS 88] T. Sasao and M. Higashida, "A design method for three-level logic circuits," (in Japanese), The Technical Papers of IEICE Japan, VLD88-84, Dec. 1988.

[SHA 49] C. E. Shannon, "The synthesis of two-terminal switching circuits," Bell Syst. Tech. J. 28, 1, pp. 59-98, 1949.

[SIG 87] Signetics, "Designing with Programmable Macro Logic," 1987.

Table 3.1 Number of Gates to Realize Various Functions of n variables

Function Class	AND-OR	OR-AND-OR
Arbitrary function	$2^{n-1}+1$	$2^{r+1}+1$ (UB) $2^{r+1}(1-\xi)$ (LB)
Parity function	$2^{n-1}+1$	$k\cdot 2^{t}+2^{k-1}+1$ $(n=k\cdot t)$
Symmetric function	$2^{n-1}+1$	$k\cdot 2^{t}+(t+1)^{k-1}+1$ $(n=k\cdot t)$
n-bit Adder	$6\cdot 2^{n}-3n-4$	$n^{2}+5n+2$

UB: Upper Bound
LB: Lower Bound

Table 4.1 Number of Gates to Realize Arithmetic circuits

Input Data			# of gates			
Name	IN	OU	two-level (A)	three-level (B)	B/A (%)	CPU SEC
ADR4	8	5	80	38	48	19
INC8	8	9	46	41	89	12
LOG8	8	8	136	84	62	175
MLP4	8	8	135	84	62	149
NRM4	8	5	125	77	62	139
RDM8	8	8	84	54	64	33
ROT8	8	5	62	48	77	24
SQR8	8	16	196	123	63	393
WGT8	8	4	259	49	19	144

IN : Number of inputs
OU : Number of outputs
CPU: SUN-3/50 Workstation

Table 4.2 Number of Gates to Realize Various Functons

Function Name	IN	OU	# of gates		B/A	CPU
			two-level (A)	three-level (B)	(%)	(sec)
INTB	15	7	638	239	37	3440
TIAL	14	8	595	255	43	3922
X7DN	66	15	553	300	54	4017

IN : Number of Inputs
OU : Number of Outputs
CPU: SUN-3/50 Workstation

Table 4.3 Number of Gates to Realize Randomly Generated Functons of 10 variables

# of minterms	# of gates		B/A (%)
	two-level (A)	three-level (B)	
128	87.1	64	73
256	134.8	65	48
384	155.9	65	42
512	159.6	65	41
640	153.5	65	42
768	130.9	65	50
896	92.4	65	70

Average of 10 functions.

(x3,x4) \ (x1,x2)	00	01	11	10
00	1		1	
01	1		1	1
11		1		1
10	1	1		

Fig. 2.1 Example 2.1

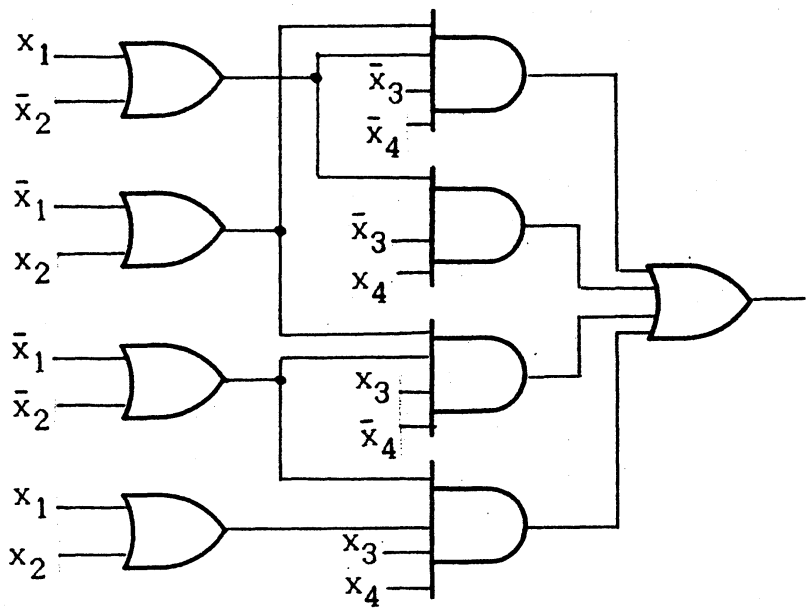

Fig. 2.2 Example 2.2

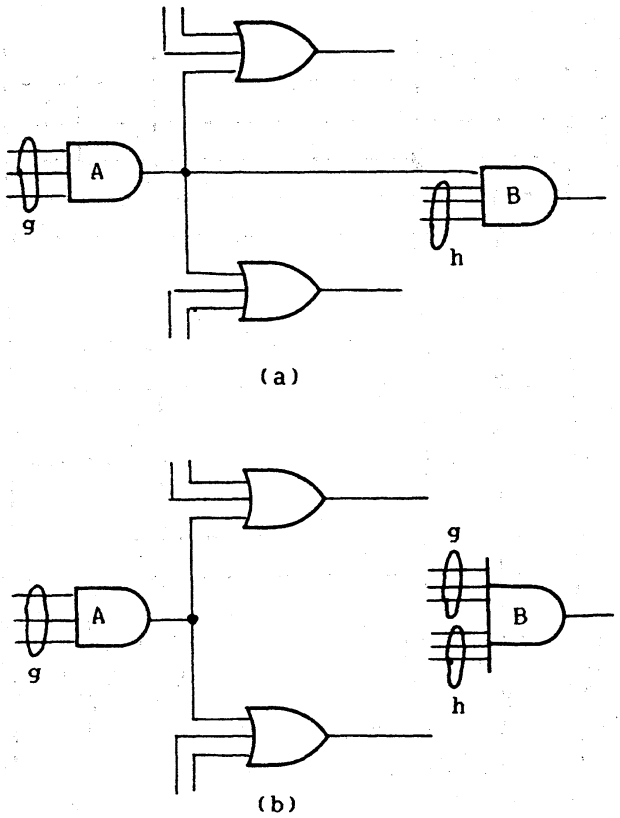

Fig.3.1 Transformation to Normal AND-OR circuit

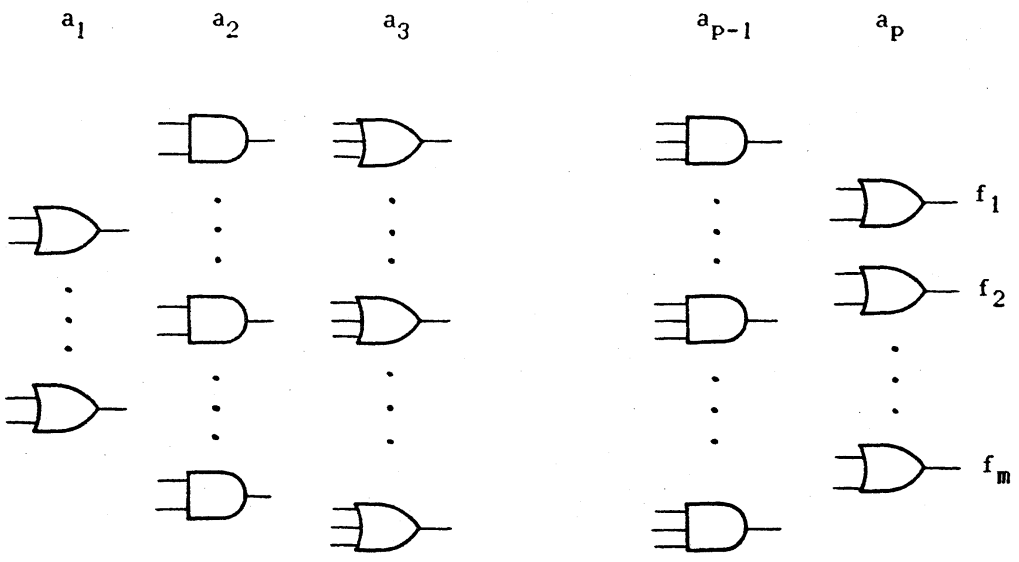

Fig.3.2 Normal AND-OR multi-level Circuit

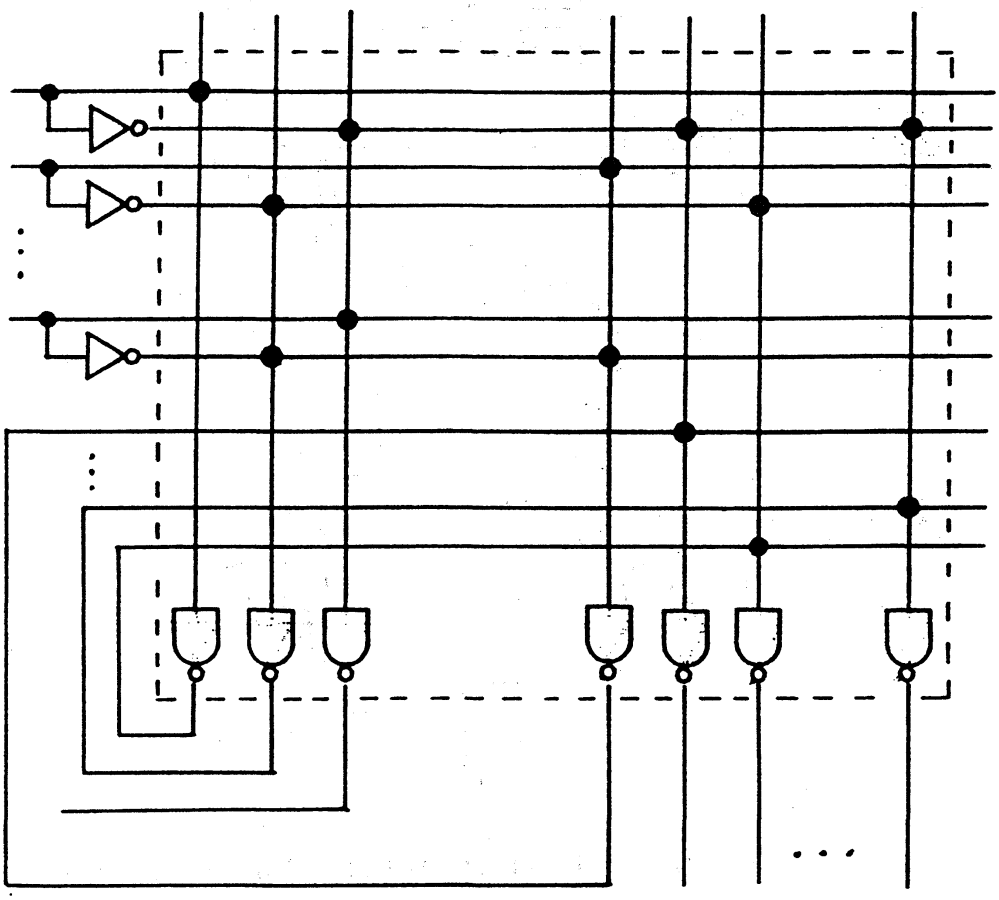

Fig. 5.1 A PLD consisting of a NAND array