IEEE TC 86

Multilevel Logical Networks

M. KARPOVSKY

Abstract—**In this correspondence we present a design technique for implementation of systems of Boolean functions in the form of multilevel AND-OR networks. We show that for a given system of Boolean functions, the transition from the traditional two-level AND-OR implementation to multilevel AND-OR implementations results in considerable savings in gate counts and delays. We discuss gate-array implementations of these multilevel networks and their space and time complexities. Experimental data for 11 different components of peripheral control units for VAX computers indicate that the transition from the two-level implementations to multilevel implementations results in average savings of about 40 percent in gate counts, of about 25 percent in required silicon areas and of about 25 percent in delays, which illustrate a good potential of the proposed techniques for design of cost-efficient gate arrays.**

Index Terms—**AND-OR implementations of systems of Boolean functions, delays, gate arrays, gate counts, multilevel logical networks, time and space complexity of gate arrays.**

I. Introduction

In this correspondence we consider a problem of implementation of systems of k Boolean functions of m arguments in the form of gate arrays. We suppose that every Boolean function is represented as an OR of some products of arguments, with or without negations. Let N denote the total number of these product terms for the whole system. We will refer to such a system as an (m, N, k) system. Implementation of (m, N, k) systems in the form of logical networks is one of the most important problems in logical design.

We present a method for minimizing the number of gates for multilevel AND-OR implementations of (m, N, k) systems using the gate arrays, standard cells or full custom circuits technologies. (PLA implementations of (m, N, k) systems will not be discussed in the correspondence.) This method is similar to the approaches developed in [1], [2]. Another approach for implementation of (m, N, k) systems was proposed in [3]. This approach requires about $2^r/r$ two-input gates where $r = m + \log_2 k$ and may be efficient only for very large N (close to 2^m). In [4] multilevel implementations of PLA's with decoders and PLA's in a form of three-level OR-AND-OR networks have been considered. Estimations on sizes of the corresponding AND and OR arrays are also presented in [4].

Let us suppose that the space complexity $G(r)$ (gate count) of an r-input gate is $r - 1$ and delay $D(r, f)$ (time complexity) for an r-input gate with the output fan-out f is $D(r, f) = D_W f + D_G r$, where D_G and D_W are constants for a given technology (D_G characterizing delays in gates and D_W delays in wires). These assumptions about space and time complexities are reasonably accurate for many technologies [5]. For example, they are applicable to CMOS technologies. (Similar assumptions for space complexities have been used in [1]–[3].) Usage of buffers for fan-out reductions and reductions of delays will not be discussed in this correspondence.

It is difficult to introduce a simple criterion for routability of a design to estimate the complexity of its layout. Experimental data (see Tables IV–VI) for different components of peripheral control units for VAX computers indicate that the transition from the traditional two-level implementations to the proposed multilevel implementations results in considerable savings in gate counts, required silicon areas, and delays.

For a path T from a primary input to a primary output delay $D(T)$ is estimated as a sum of all delays of gates on this path plus $D_W f_{in}$ where f_{in} is the input fanout. Let $D(T) = D_1(T)D_W + D_2(T)D_G$ and $\max_T (D_1(T) + D_2(T)) = D_1(T_0) + D_2(T_0)$. Then the delay of the whole network is estimated as $D(T_0) = D_1(T_0)D_W + D_2(T_0)D_G$.

For the traditional two-level AND-OR implementations we have for the space complexity

$$G_2(m, N, k) \leq (m-1)N + (N-2)k \qquad (1)$$

and for the time complexity

$$D_2(m,\ N,\ k) \le D_W(N+k) + D_G(m+N-1). \tag{2}$$

For example, given a (16, 64, 16) system we have by (1), (2)

$$G_2(16,\ 64,\ 16) \le 1952, \quad D_2(16,\ 64,\ 16) \le 80D_W + 79D_G.$$

(Upper bounds (1) and (2), as well as all upper bounds below, represent the "worst case" analysis. For example, in (1) and (2) we assume that all product terms are minterms and every output is an OR of $N - 1$ minterms.)

In the next sections we present design techniques for multilevel AND-OR implementations which result in a drastic decrease of space complexity (1) and in time complexity (2).

Let $x_1, \cdots, x_m$ be the inputs, $P_1, \cdots, P_N$ the product terms and $f_1, \cdots, f_k$ the outputs of a (m, N, k) system ($N \le 2^{m-1}$, $k \le 2^N$).

Example 1: Suppose we have a system of 6 Boolean functions $f_1, \cdots, f_6$, each with 6 arguments $x_1, \cdots, x_6$ and 8 product terms $P_1, \cdots, P_8$ as follows:

$$P_1 = x_1x_2 \quad P_2 = x_1x_3\bar{x}_4 \quad P_3 = \bar{x}_2\bar{x}_3\bar{x}_4 \quad P_4 = \bar{x}_1\bar{x}_3\bar{x}_4\bar{x}_5$$

$$P_5 = x_2x_5\bar{x}_6 \quad P_6 = \bar{x}_3\bar{x}_4x_6 \quad P_7 = \bar{x}_2x_3\bar{x}_4x_6 \quad P_8 = x_3\bar{x}_4x_5\bar{x}_6$$

$$f_1 = P_1 \vee P_2 \vee P_3 \vee P_4 \vee P_6 \quad f_2 = P_2 \vee P_3 \vee P_4 \vee P_7 \vee P_8 \quad f_3 = P_3 \vee P_5 \vee P_7$$

$$f_4 = P_1 \vee P_2 \vee P_5 \vee P_6 \quad f_5 = P_2 \vee P_5 \vee P_8 \quad f_6 = P_1 \vee P_2 \vee P_4 \vee P_7 \vee P_8.$$

For this (6, 8, 6) system, counting the literals in the terms, we have $G_2(6, 8, 6) = 37$, and for delay we have $D_2(6, 8, 6) = 12D_W + 8D_G$($T_0 = \bar{x}_4P_2f_1$).

II. Universal AND/Universal OR Networks

First let us construct a network with α inputs implementing all possible products of at most α arguments, with or without negations. We will call this network a universal AND network (UAN) for α arguments. This network has α inputs and $3^\alpha - 1$ outputs corresponding to all nontrivial products of α arguments.

The UAN for $\alpha = 4$ arguments is presented in Fig. 1(a). The recursive procedure to construct a universal AND network for α arguments from the networks for $\lceil \alpha/2 \rceil$ and $\lfloor \alpha/2 \rfloor$ arguments is illustrated by Fig. 1(b) ($\lfloor y \rfloor$ ($\lceil y \rceil$) is the greatest (smallest) integer smaller (greater) or equal to y). Thus, for the space complexity, $G_{AND}(\alpha)$ of UAN we have

$$G_{AND}(1) = 0, \quad G_{AND}(\alpha) = G_{AND}(\lceil \alpha/2 \rceil) + G_{AND}(\lfloor \alpha/2 \rfloor) + (3\lfloor \alpha/2 \rfloor - 1)(3\lceil \alpha/2 \rceil - 1), \tag{3}$$

and from (3)

$$G_{AND}(\alpha) = 3^\alpha - 2\alpha - 1. \tag{4}$$

It is easy to see that the UAN represented by Fig. 1 is optimal from the point of view of space complexity $G_{AND}(\alpha)$, since there are $3^\alpha - 2\alpha - 1$ nontrivial products of α arguments (arguments and their negations considered to be trivial products) and we are using exactly one two-input gate per output. For the time complexity $D_{AND}(\alpha)$ of UAN with α inputs we have from Fig. 1(b)

$$D_{AND}(\alpha) \simeq D_W 3^{\lceil \alpha/2 \rceil}(\lceil \log_2 \alpha \rceil + 1) + D_G(1 + 2\lceil \log_2 \alpha \rceil)). \tag{5}$$

Similarly, we can introduce universal OR networks (UON's) for β arguments implementing all possible $2^\beta - 1$ OR's of at most β inputs. These UON's are represented by Fig. 2, and in this case

$$G_{OR}(\beta) = 2^\beta - \beta - 1 \tag{6}$$

and

$$D_{OR}(\beta) = D_W 2^{\lceil \beta/2 \rceil} \lceil \log_2 \beta \rceil + D_G 2 \lceil \log_2 \beta \rceil . \quad (7)$$

These universal OR networks are also optimal from the point of view of the space complexity $G_{OR}(\beta)$.

III. Optimal Multilevel Implementations

In this section we will use universal AND networks and universal OR networks to construct optimal three-level and four-level AND-OR networks.

Let us first consider three-level networks consisting of two levels of AND gates and one level of OR gates. We will refer to these networks as AND-AND-OR networks.

Partition the set $X = \{x_1, \cdots, x_m\}$ of inputs into disjoint subsets $R_1, R_2, \cdots, R_a$ where $R_i \cap R_j = \emptyset$, $\cup_{i=1}^{a} R_i = X$ and $|R_j| = \alpha_j$, $\Sigma_{j=1}^{a} \alpha_j = m$. For every subset R_i we construct the universal AND network for all the arguments from R_i. Then every product term may be obtained by ANDing of outputs of the corresponding universal AND networks. In this case, every AND gate will have at most one input coming from every universal AND network, i.e., the number of inputs for AND gates at the second level is at most "a." The resulting AND-AND-OR structure is represented in Fig. 3.

The three-level AND-OR-OR network based on the partition $\pi = \{\pi_1, \cdots, \pi_b\}$ of the set $P = \{P_1, \cdots, P_N\}$ of product terms ($\pi_i \cap \pi_j = \phi$, $\cup_{i=1}^{b} \pi_i = P$, $|\pi_i| = \beta_i$, $\Sigma_{i=1}^{b} \beta_i = N$) is given in Fig. 4.

A four-level AND-AND-OR-OR implementation of a (m, N, k) system based on two partitions R and π is represented by Fig. 5.

Example 2: Let us construct a four-level AND-AND-OR-OR implementation for the (6, 8, 6) system described in Example 1.

Select the following partitions: $R = \{R_1, R_2, R_3\}$, $R_1 = \{x_1, x_2\}$, $R_2 = \{x_3, x_4\}$, $R_3 = \{x_5, x_6\}$ ($\alpha_1 = \alpha_2 = \alpha_3 = 2$, $a = 3$) and $\pi = \{\pi_1, \pi_2, \pi_3\}$, $\pi_1 = \{P_1, P_2\}$, $\pi_2 = \{P_3, P_4, P_5\}$, $\pi_3 = \{P_6, P_7, P_8\}$ ($\beta_1 = 2$, $\beta_2 = \beta_3 = 3$, $b = 3$).

The four-level implementation based on these partitions is represented by Fig. 6. In this case, for the space and time complexities we have $G_4(6, 8, 6) = 28$ and $D_4(6, 8, 6) = 11D_W + 10D_G$ (for three-level AND-AND-OR and AND-OR-OR we have correspondingly for this (6, 8, 6) system $G_3(6, 8, 6) = 32$, $D_3(6, 8, 6) = 9D_W + 10D_G$ and $G_3(6, 8, 6) = 33$, $D_3(6, 8, 6) = 12D_W + 8D_G$). Comparing these numbers to the complexities of the two-level implementations $G_2(6, 8, 6) = 37$, $D_2(6, 8, 6) = 12D_W + 8D_G$, we can see that considerable savings are realized in gate numbers without any substantial increase in delays if we replace the two-level implementation by the three-level or four-level implementations.

IV. Space and Time Complexities of Multilevel Implementations

We will now present upper bounds for the space and time complexities $G_4(m, N, k)$ and $D_4(m, N, k)$ of four-level AND-AND-OR-OR implementations of (m, N, k) systems. (Complexities of three-level AND-AND-OR and AND-OR-OR implementations may be estimated in a similar manner).

From the block-diagram of Fig. 6 and (4), (6), and (7), we have

$$\begin{aligned} G_4(m, N, k) &\le \sum_{i=1}^{a} (3^{\alpha_i} - 2\alpha_i - 1) + N(a-1) \\ &\quad + \sum_{j=1}^{b} (2^{\beta_j} - \beta_j - 1) + k(b-1) \\ &= \sum_{i=1}^{a} 3^{\alpha_i} - 2m - a + N(a-1) \\ &\quad + \sum_{j=1}^{b} 2^{\beta_i} - N - b - k(b-1) \end{aligned} \quad (8)$$

for any $\alpha_1, \cdots, \alpha_a$ and $\beta_1, \cdots, \beta_b$ such that

$$\sum_{i=1}^{a} \alpha_i = m \text{ and } \sum_{j=1}^{b} \beta_j = N.$$

Thus, we have from (8)

$$+ \min_{\beta_1 + \cdots + \beta_b = N} \left(\sum_{j=1}^{b} (2^{\beta_j} - \beta_j - 1) + k(b-1) \right). \quad (9)$$

It is very difficult to find analytical solutions $\alpha_1^*, \cdots, \alpha_a^*$ and $\beta_1^*, \cdots, \beta_b^*$, minimizing the right-hand part of (9). In Appendix I we present the optimal values for $\alpha_1, \cdots, \alpha_a$ for $1 \le m < 40$ and $1 \le N \le 80$, together with the corresponding upper bounds on numbers of AND gates in the two levels of the AND array. Similarly, in Appendix II we present the optimal values for $\beta_1, \cdots, \beta_b$ for $1 \le N \le 80$ and $1 \le k \le 40$, and also upper bounds on numbers of OR gates in the two levels of the OR array ($N \le 2^m$, $k \le 2^N$). We also note that

$$\max_i \alpha_i^* - \min \alpha_i^* \le 1$$

$$\max_j \beta_j^* - \min \beta_j^* \le 1. \quad (10)$$

In Appendix I (Appendix II) the notation $\overset{\lambda s}{\lambda+1, t}$ stands for $\alpha_1^* = \cdots = \alpha_s^* = \lambda, \alpha_{s+1}^* = \cdots = \alpha_{s+t}^* = \lambda + 1, s + t = a, \lambda s + (\lambda + 1)t = m$ ($\beta_1^* = \cdots = \beta_s^* = \lambda, \beta_{s+1}^* = \cdots = \beta_{s+t}^* = \lambda + 1, s + t = b, \lambda s + (\lambda + 1)t = N$). Data in Appendix I and Appendix II have been generated by exhaustive computer search of all $\alpha_1, \cdots, \alpha_a (\sum_{i=1}^{a} \alpha_i = m)$ and $\beta_1, \cdots, \beta_b (\sum_{j=1}^{b} \beta_j = N)$.

To estimate the time complexity $D_4(m, N, k)$ of four-level implementations we note that different outputs of a universal AND (OR) network are inputs for different AND (OR) gates of the second (fourth) level. Denote $n_{i,j}^{\text{AND}}$ ($n_{i,j}^{\text{OR}}$) a number of AND gates of the second level (OR gates of the fourth level) interconnected with the jth output of the ith universal AND (OR) network. Then, for any i

$$\sum_{j=1}^{3^{\alpha_j - 1}} {}^{\text{AND}}_{n_{i,j}} \le N, \quad i = 1, \cdots, a \quad (11)$$

$$\sum_{j=1}^{2^{\beta_j - 1}} {}^{\text{OR}}_{n_{i,j}} \le k, \quad i = 1, \cdots, b. \quad (12)$$

Thus, we have in a view of (5), (7), (11), and (12)

$$D_4(m, N, k) \le D_W(N + k + \max_i (3^{\lceil \alpha_i/2 \rceil} (\lceil \log_2 \alpha_i \rceil + 1)$$

$$+ \max_j (2^{\lceil \beta_j/2 \rceil} \lceil \log_2 \beta_j \rceil))$$

$$+ D_G(1 + 2 \max_i \lceil \log_2 \alpha_i \rceil$$

$$+ 2 \max_j \lceil \log_2 \beta_j \rceil + a + b). \quad (13)$$

Example 3: Suppose $m = 16$, $N = 64$, $k = 16$. Then from Appendix I $a = 6$, $\alpha_1^* = \alpha_2^* = 2$, $\alpha_3^* = \alpha_4^* = \alpha_5^* = \alpha_6^* = 3$ and from Appendix II $b = 21$, $\beta_1^* = \cdots = \beta_{20}^* = 3$, $\beta_{21}^* = 4$. Thus, finally we have from Appendix I and Appendix II and (8), (13)

$$G_4(16, 64, 16) \le 408 + 411 = 819$$

$$D_4(16, 64, 16) \le D_W(64 + 16 + 3^2(2 + 1) + 2^2 \cdot 2)$$

$$+ D_G(1+2\cdot 2+2\cdot 2+6+21) = 115D_W + 36D_G.$$

(Note, that $G_2(16, 64, 16) \leq 1{,}952$ and $D_2(16, 64, 16) \leq 80D_W + 79D_G$).

As a reasonably good approximation for $\alpha_1^*, \cdots, \alpha_a^*$ and $\simeq \beta_1^*, \cdots, \beta_b^*$ for big m, N, and k we can take

$$\alpha_1^* = \cdots = \alpha_a^* = \lfloor \log_3 N - \log_3 \log_3 N \rfloor,$$

$$a = \left\lceil \frac{m}{\log_3 N - \log_3 \log_3 N} \right\rceil \quad (14)$$

and

$$\beta_1^* = \cdots = \beta_b^* = \lfloor \log_2 k - \log_2 \log_2 k \rfloor,$$

$$b = \left\lceil \frac{D}{\log_2 k - \log_2 \log_2 k} \right\rceil. \quad (15)$$

In this case, we have from (8), (13), (14) and (15)

$$\begin{aligned} G_4(m, N, k) \leq & \frac{Nm}{\log_3 N - \log_3 \log_3 N} \\ & + \frac{Nm}{\log_3 N (\log_3 N - \log_3 \log_3 N)} \\ & - 2m + \frac{Nk}{\log_2 k - \log_2 \log_2 k} \\ & + \frac{Nk}{\log_2 k (\log_2 k - \log_2 \log_2 k)} - 2N - k \end{aligned} \quad (16)$$

and

$$\begin{aligned} D_4(m, N, k) \leq & D_W \left(N + k + \sqrt{\frac{N}{\log_3 N}} (\lceil \log_2 \log_3 N \rceil + 1) \right. \\ & + \sqrt{\frac{k}{\log_2 k}} \lceil \log_2 \log_2 k \rceil) \\ & + D_G(1 + 2 \lceil \log_2 \log_3 N \rceil \\ & + 2 \lceil \log_2 \log_2 k \rceil + \left\lceil \frac{m}{\log_3 N - \log_3 \log_3 N} \right\rceil \\ & \left. + \left\lceil \frac{N}{\log_2 k - \log_2 \log_2 k} \right\rceil \right). \end{aligned} \quad (17)$$

Let us now estimate an asymptotical behavior of $G_4(m, N, k)$ and $D_4(m, N, k)$ as $m, N, k \to \infty$ and $m/\log_3 N, N/\log_2 k \to \infty$. From (16) and (17) we have in this case

$$G_4(m, N, k) \lesssim \frac{mN}{\log_3 N} + \frac{Nk}{\log_2 k} \quad (18)$$

and

$$\begin{aligned} D_4(m, N, k) \lesssim & D_W(N+k) + D_G \left(\frac{m}{\log_3 N} + \frac{N}{\log_2 k} \right. \\ & \left. + 2 \log_3 \log_2 N + 2 \log_2 \log_2 k \right) \end{aligned} \quad (19)$$

($\rho(n) \lesssim \tau(n)$ iff $\lim_{n\to\infty} \frac{\rho(n)}{\tau(n)} \leq 1$, $\rho(n) \sim \tau(n)$ iff $\rho(n) \lesssim \tau(n)$ and $\tau(n) \lesssim \rho(n)$.).

Note, that for two-level implementations

$$G_2(m,\ N,\ k) \leq mN+Nk \tag{20}$$

$$D_2(m,\ N,\ k) \leq D_W(N+k)+D_G(m+N). \tag{21}$$

Thus, comparing (16), (17) with (1), (2) or (18), (19) with (20), (21) one can see that the transition from traditional two-level AND-OR implementations to multilevel AND-OR implementations results in considerable savings both in gate counts and delays.

We note also, that formulas (8), (16), and (18) present the upper bounds for gate counts since for a given (m, N, k) system probably not all $3^{\alpha_i}-1$ outputs of universal AND networks $(i=1, \cdots, a)$ and not all $2^{\beta_i}-1$ outputs of universal OR networks $(j=1, \cdots, b)$ have to be implemented.

Let us estimate now numbers of gates in universal AND (OR) networks UAN_q (UON_q), $q=1, \cdots, a$ ($q=1, \cdots, b$), required for a given (m, N, k) system.

Any (m, N, k) system may be represented by two matrices (x_{ij}) and (P_{ij}) where

$$x_{ij}=\begin{cases}1, & x_j \in P_i;\\ -1, & \bar{x}_j \in P_i;\ (i=1, \cdots, N,\ j=1, \cdots, m)\\ 0, & \text{otherwise;}\end{cases} \tag{22}$$

$$P_{ij}=\begin{cases}1, & P_j \in f_j;\\ 0, & P_j \notin f_j;\end{cases}\quad (i=1, \cdots, k,\ j=1, \cdots, N). \tag{23}$$

These matrices for the (6, 8, 6) system from Example 1 are

$$(x_{ij})=\left(\begin{array}{cc|cc|cc}
1 & 1 & 0 & 0 & 0 & 0\\
1 & 0 & -1 & -1 & 0 & 0\\
0 & -1 & -1 & -1 & 0 & 0\\
-1 & 0 & -1 & -1 & -1 & 0\\
0 & 1 & 0 & 0 & 1 & -1\\
0 & 0 & -1 & -1 & 0 & 1\\
0 & -1 & 1 & -1 & 0 & 1\\
0 & 0 & 1 & -1 & 1 & -1
\end{array}\right)$$

(α_1^*, α_2^*, α_3^*)

$$(P_{ij})=\left(\begin{array}{cc|ccc|ccc}
1 & 1 & 1 & 1 & 0 & 1 & 0 & 0\\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1\\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0\\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1\\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1
\end{array}\right)$$

(β_1^*, β_2^*, β_3^*)

Every partition $R=\{R_1, \cdots, R_a\}$, $|R_i|=\alpha_i^*$ (or $\pi=\{\pi_1, \cdots, \pi_b\}$, $|\pi_j|=\beta_j^*$) of the set of inputs $X=\{x_1, \cdots, x_m\}$ (or the set of product terms $P=\{P_1, \cdots, P_N\}$) defines the partition of the matrix (x_{ij}) (or (P_{ij})) into a (or b) submatrices containing, correspondingly, $\alpha_1^*, \cdots, \alpha_a^*$ or $\beta_1^*, \cdots, \beta_b^*$ columns. The number of outputs of UAN_q (or UON_q) required for the realization of a given (m, N, k) system is equal to the number of different nonzero rows of the qth submatrix of (x_{ij}) (or (P_{ij})). It was shown in Section II, that the implementation of a nontrivial output (not equal to any input or its negation) of UAN_q (or UON_q) requires on average one gate.

Denote $G_A(R_q)$ ($G_{\mathrm{OR}}(\pi_q)$) a number of different rows containing more than one nonzero element in the qth submatrix of (x_{ij}) ((P_{ij}). (For the (6, 8, 6) system considered above $G_A(R_1)=1$, $G_A(R_2)=2$, $G_A(R_3)=1$, $G_{\mathrm{OR}}(\pi_1)=1$, $G_{\mathrm{OR}}(\pi_2)=2$, $G_{\mathrm{OR}}(\pi_3)=1$). Then

$$G_4(m,\ n,\ k)=\sum_{q=1}^{a} G_A(R_q)+N(a-1)+\sum_{q=1}^{b} G_{\mathrm{OR}}(\pi_q)+k(b-1). \tag{24}$$

Since $G_A(R_q)$ and $G_{OR}(\pi_q)$ depend on the choice of partitions R and π, we have

$$\min_{R,\pi} G_4(m,\ N,\ k) = \min_{R}\left(\sum_{q=1}^{a} G_A(R_q) + N(a-1)\right) + \min_{\pi}\left(\sum_{q=1}^{b} G_{OR}(\pi_q) + k(b-1)\right). \quad (25)$$

where R is any partition $R = \{R_1, \cdots, R_a\}$, $|R_q| = \alpha_q^*$, $\Sigma_{q=1}^{a} \alpha_q^* = m$ and $\pi = \{\pi_1, \cdots, \pi_b\}$, $|\pi_q| = \beta_q^*$, $\Sigma_{q=1}^{b} \beta_q^* = N$.

Numbers of different partitions R and π are increasing with the increase of m, N, and k.

Thus, network minimization by optimizing partitions R and π for the given parameters $\alpha_1^*, \cdots, \alpha_a^*$ and $\beta_1^*, \cdots, \beta_b^*$ may be time consuming and, in most cases, does not result in considerable additional savings.

Table I summarizes the results presented in previous sections for the optimal sizes of partitions α^* or β^*, corresponding gate counts G and delays D. In Table II these results are presented for (16, 64, 16) systems and in Table III these results are presented in the asymptotic form. In Table IV we present experimental data on gate counts for two-level AND-OR, three-level AND-AND-OR, and four-level AND-AND-OR-OR implementations for 11 components of peripheral control units for VAX computers. In Table V, for the first 6 devices from Table IV, we present total gate counts and delays for two-level and four-level implementations. One can see from these tables that the transition from two-level implementations to four levels results in an average saving in gate counts of about 43 percent (which translates in considerable savings in required silicon areas, see also Table VI) and an average savings in delays of about 26 percent.

TI
TII
TIII
TIV
TV, TVI

V. Multilevel Implementations for Special Classes of (m, N, k) Systems

In previous sections we presented general techniques which are applicable to any (m, N, k) system. If an additional information about the system is available to a designer than time and space complexities may be decreased. In this section we are going to illustrate this situation by several important examples.

By *density* d of an (m, N, k) system we mean fraction of nonzero entries in (x_{ij}) matrix defined by (22), i.e.,

$$d = m^{-1}N^{-1}\sum_{i,j}|x_{ij}| \quad 0 \le d \le 1. \quad (26)$$

Efficiency of techniques presented in previous sections is increasing with an increase in density d. The case $d = 1$ (all product terms for an (m, N, k) system are minterms) is of a special interest. In this case, we can replace in the four-level implemtation of Fig. 5 universal AND networks for α arguments by ($\alpha \times 2^{\alpha}$) decoders, and α^* can be chosen as

$$\alpha^* = \alpha_1^* = \cdots = \alpha_a^* = \lfloor \log_2 N - \log_2 \log_2 N \rfloor$$

$$a = \left\lceil \frac{m}{\log_2 N - \log_2 \log_2 N} \right\rceil. \quad (27)$$

Then we have for space and time complexities

$$G_4(m,\ N,\ k) \lesssim \frac{mN}{\log_2 N} + \frac{Nk}{\log_2 k} \quad (28)$$

and

$$D_4(m,\ N,\ k) \lesssim D_W(N+k) + D_G\left(\frac{m}{\log_2 N}\right.$$

$$+\frac{N}{\log_2 k}+2\log_2\log_2 N+2\log_2\log_2 k\Bigg). \quad (29)$$

Thus, comparing (28) and (29) to (18) and (19), one can see that if all products terms are minterms ($d = 1$) then we have asymtotical savings in time and space complexities of AND arrays by the factor of $\log_2 3 = 1.58$.

In Table VI we present experimental results on savings from the transition from two-level implementations by simulation of layouts for two systems of Boolean functions ($m = k = 16, N = 64, d = 1$, and $d = 0.66$) in the gate array with 2200 gates. These experimental results illustrate savings in gate counts and required silicon areas with the increase in a density.

Similar results may be obtained for the case of unary (m, N, k) systems. Remind that a (m, N, k) system is positive (negative) with respect to input x_i if all outputs of the system depend on x_i and do not depend on $\bar{x}_i$ (depend on $\bar{x}_i$ and do not depend on x_i). A system is unary if it is either positive or negative with respect to every input. It is easy to check that for a unary system formulas (27), (28), and (29) remain valid.

Let us summarize now the above asymtotical results. Suppose that for a given (m, N, k) system, m_1 arguments are unary, and from the remaining $m - m_1$ arguments there are m_2 arguments such that a submatrix of (x_{ij}) formed by the corresponding m_2 columns has density equal to one. Then for $m_1, m_2, m_3 = m - m_1 - m_2, N, k \to \infty$, $m_i/\log_3 N \to \infty (i = 1, 2, 3)$, $N \to \infty$, $k \to \infty$, $k/\log_2 k \to \infty$

$$G_4(m_1, m_2, m_3, N, k) \lesssim \frac{(m_1+m_2)N}{\log_2 N}+\frac{m_3 N}{\log_3 N}+\frac{Nk}{\log_2 k} \quad (30)$$

and

$$D_4(m_1, m_2, m_3, N, k) \lesssim D_W(N+k)+D_G\left(\frac{m_1+m_2}{\log_2 N}+\frac{m_3}{\log_3 N}\right.$$

$$\left.+\frac{N}{\log_2 k}+2\log_2\log_2 N+2\log_2\log_2 k\right). \quad (31)$$

Another important special case of (m, N, k) systems is the class of single-output systems ($k = 1$). In this case, the approach described in Section IV results in three-level AND-AND-OR implementations with

$$G_3(m, N, 1) \lesssim \min_{\alpha_1+\cdots+\alpha_a=m}\left(\sum_{i=1}^{a}(3^{\alpha_i}-2\alpha_i-1)+Na\right), \quad (32)$$

$$D_3(m, N, 1) \lesssim D_W(N+\max_i(3^{\lceil\alpha_i/2\rceil}(\lceil\log_2\alpha_i\rceil+1)+1)$$

$$+D_G(1+2\max_i\lceil\log_2\alpha_i\rceil+a+N) \quad (33)$$

(see (8) and (13)).

We shall present now another approach which results in considerable savings in time and space complexities (32) and (33).

Let us represent Boolean function $f(x_1, \cdots, x_m)$ which should be implemented in the form

$$f(x_1, \cdots, x_{\alpha_1}, x_{\alpha_1+1}, \cdots, x_m)$$

$$=\bigvee_{i=1}^{h} Q_i(x_1, \cdots, x_{\alpha_1})\varphi_i(x_{\alpha_1+1}, \cdots, x_m) \quad (34)$$

where $h = 3^{\alpha_1}$, $Q_i(x_1, \cdots, x_{\alpha_1})$ $(i = 1, 2, \cdots, h)$ are all possible products of at most α_1 arguments $x_1, \cdots, x_{\alpha_1}$ with or without negations and $\varphi_i(x_{\alpha_1+1}, \cdots, x_m)$ are sums of products of $x_{\alpha_1+1}, \cdots, x_m$ with or without negations. Then $f(x_1, \cdots, x_m)$ may be implemented by the block diagram of Fig. 7.

For the network represented by Fig. 7 we have

Fig 7

$$G(m,\ N,\ 1) \leq \min_{\alpha_1+\cdots+\alpha_a=m} \left(\sum_{i=1}^{a} (3^{\alpha_i} - 2\alpha_i - 1) + N(a-1) + 3^{\alpha_i} \right) \tag{35}$$

$$D(m,\ N,\ 1) \leq D_W(N + \max_{i>1} (3^{\lceil \alpha_i/2 \rceil} \lceil \log_2\ \alpha_i \rceil + 1) + 1)$$

$$D_G(1 + 2 \max_{i>1} \lceil \log_2\ \alpha_i \rceil + a + N). \tag{36}$$

As a reasonably good approximation to the optimal values of $\alpha_1, \cdots, \alpha_a$ minimizing the right hand of (35) one can take again $\alpha_1^* = \cdots = \alpha_a^* = \lfloor \log_3 N - \log_3 \log_3 N \rfloor$ and $a = \lceil m/(\log_3 N - \log_3 \log_3 N) \rceil$.

Comparing (35) and (36) to (32) and (33) one can see that usage of block diagram of Fig. 7 may result in considerable savings in gate counts without increases in delays (see also Example 4 below).

Example 4: Suppose $m = 16$, $N = 64$, $x = 1$. Then from Appendix I $a = G$, $\alpha_1^* = \alpha_2^* = 2$, $\alpha_3^* = \alpha_4^* = \alpha_5^* = \alpha_6^* = 3$. By (32), (33) and (35), (36) we have that the transition from the three-level implementation of Fig. 3 to the four-level implementation of Fig. 7 results in the decrease in the gate count from 472 to 417 without any increase in delay.

Acknowledgment

The author wishes to thank A. Jones, Cambridge University, Cambridge, UK for computing data in Appendix I and Appendix II and for many useful discussions, J. Morse, the manager of Microsystem Advanced Development Group, Digital Equipment Corporation, Maynard, MA, and T. Bryan from this group for many very helpful comments.

Computed 1-07 8611448

Gal.10

Appendix I

Optimal Values LAMBDA, S LAMBDA + 1, t for Parameters α^* (LAMBDA·S + (LAMBDA + 1)·t = m), Gate Counts G and Delays $D0$, $D1$ ($D = D_W D0 + D_G D1$) for and Arrays

m	2	4	6	8	10	12	14	16	18	20
N = 4										
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	2, 1	2, 2	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	4	12	20	28	36	44	52	60	68	76
D0,D1	10, 4	10, 6	10, 6	10, 7	10, 8	10, 8	10,10	10,11	10,12	10,13
N = 6										
LAMBDA,S	0, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	0, 0	2, 2	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	0	16	28	40	52	64	76	88	100	112
D0,D1	0, 0	14, 5	14, 6	14, 7	14, 8	14, 8	14,10	14,11	14,12	14,13
N =12										
LAMBDA,S	0, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	0, 0	2, 2	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	0	20	36	52	68	84	100	116	132	148
D0,D1	0, 0	18, 5	18, 6	18, 7	18, 8	18, 9	18,10	18,11	18,12	18,13
N =16										
LAMBDA,S	0, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	0, 0	2, 2	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	0	24	44	64	84	104	124	144	164	184
D0,D1	0, 0	22, 5	22, 6	22, 7	22, 8	22, 9	22,10	22,11	22,12	22,13
N =20										
LAMBDA,S	0, 0	0, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	0, 0	0, 0	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	0	0	52	76	100	124	148	172	196	220
D0,D1	0, 0	0, 0	26, 6	26, 7	26, 8	26, 9	26,10	26,11	26,12	26,13
N =24										
LAMBDA,S	0, 0	0, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	0, 0	0, 0	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	0	0	60	88	116	144	172	200	228	256
D0,D1	0, 0	0, 0	30, 6	30, 7	30, 8	30, 9	30,10	30,11	30,12	30,13
N =28										
LAMBDA,S	0, 0	0, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	0, 0	0, 0	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	0	0	68	100	132	164	196	228	260	292
D0,D1	0, 0	0, 0	34, 6	34, 7	34, 8	34, 9	34,10	34,11	34,12	34,13
N =32										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	72	108	144	178	212	246	280	318
D0,D1	0, 0	0, 0	38, 7	38, 8	38, 9	38, 9	38,10	38,11	38,11	38,12
N =36										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	76	116	156	186	226	266	300	340
D0,D1	0, 0	0, 0	43, 7	43, 8	43, 9	43, 9	43,10	43,11	43,11	43,12
N =40										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	80	124	168	200	244	288	320	364
D0,D1	0, 0	0, 0	47, 7	47, 8	47, 9	47, 9	47,10	47,11	47,11	47,12
N =44										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	84	132	180	212	260	308	340	388
D0,D1	0, 0	0, 0	51, 7	51, 8	51, 9	51, 9	51,10	51,11	51,11	51,12
N =48										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	88	140	192	224	276	326	360	412
D0,D1	0, 0	0, 0	54, 7	54, 8	54, 9	54, 9	54,10	54,11	54,11	54,12
N =52										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	92	148	204	236	292	348	380	436
D0,D1	0, 0	0, 0	58, 7	58, 8	58, 9	58, 9	58,10	58,11	58,11	58,12
N =56										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	96	156	216	248	308	368	400	460
D0,D1	0, 0	0, 0	61, 7	61, 8	61, 9	61, 9	61,10	61,11	61,11	61,12
N =60										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	100	164	228	260	324	388	420	484
D0,D1	0, 0	0, 0	67, 7	67, 8	67, 9	67, 9	67,10	67,11	67,11	67,12
N =64										
LAMBDA,S	0, 0	0, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	3, 2	3, 2	3, 2	3, 4	3, 4	3, 4	3, 6	3, 6
G	0	0	104	172	240	272	340	408	440	504
D0,D1	0, 0	0, 0	71, 7	71, 8	71, 9	71, 9	71,10	71,11	71,11	71,12
N =68										
LAMBDA,S	0, 0	0, 0	0, 0	2, 1	3, 2	2, 0	2, 1	3, 4	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	0, 0	3, 2	4, 1	3, 4	3, 4	4, 1	3, 6	3, 6
G	0	0	0	180	248	284	356	424	460	532
D0,D1	0, 0	0, 0	0, 0	75, 8	75, 8	75, 9	75,10	75,10	75,11	75,12
N =72										
LAMBDA,S	0, 0	0, 0	0, 0	2, 1	3, 2	2, 0	2, 1	3, 4	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	0, 0	3, 2	4, 1	3, 4	3, 4	4, 1	3, 6	3, 6
G	0	0	0	188	268	296	372	440	480	556
D0,D1	0, 0	0, 0	0, 0	80, 8	80, 8	80, 9	80,10	80,10	80,11	80,12
N =76										
LAMBDA,S	0, 0	0, 0	0, 0	2, 1	3, 2	2, 0	2, 1	3, 4	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	0, 0	3, 2	4, 1	3, 4	3, 4	4, 1	3, 6	3, 6
G	0	0	0	196	284	308	388	456	500	580
D0,D1	0, 0	0, 0	0, 0	103, 8	103, 8	103, 9	103,10	103,10	103,11	103,12
N =80										
LAMBDA,S	0, 0	0, 0	0, 0	2, 1	3, 2	2, 0	2, 1	3, 4	2, 0	2, 1
LAMBDA+1,t	0, 0	0, 0	0, 0	3, 2	4, 1	3, 4	3, 4	4, 1	3, 6	3, 6
G	0	0	0	204	272	320	404	472	520	604
D0,D1	0, 0	0, 0	0, 0	107, 8	107, 8	107, 9	107,10	107,10	107,11	107,12

Computers 1-07 8611448

Sal. 11

APPENDIX I (*continued*)

m	22	24	26	28	30	32	34	36	38
N = 4									
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,E	2,11	2,12	2,13	2,14	2,15	2,16	2,17	2,18	2,19
G	84	92	100	108	116	124	132	140	148
DO,D1	10,14	10,15	10,16	10,17	10,18	10,19	10,20	10,21	10,22
N = 8									
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,E	2,11	2,12	2,13	2,14	2,15	2,16	2,17	2,18	2,19
G	124	136	148	160	172	184	196	208	220
DO,D1	14,14	14,15	14,16	14,17	14,18	14,19	14,20	14,21	14,22
N =12									
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,E	2,11	2,12	2,13	2,14	2,15	2,16	2,17	2,18	2,19
G	164	180	196	212	228	244	260	276	292
DO,D1	18,14	18,15	18,16	18,17	18,18	18,19	18,20	18,21	18,22
N =16									
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,E	2,11	2,12	2,13	2,14	2,15	2,16	2,17	2,18	2,19
G	204	224	244	264	284	304	324	344	364
DO,D1	22,14	22,15	22,16	22,17	22,18	22,19	22,20	22,21	22,22
N =20									
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,E	2,11	2,12	2,13	2,14	2,15	2,16	2,17	2,18	2,19
G	244	268	292	316	340	364	388	412	436
DO,D1	26,14	26,15	26,16	26,17	26,18	26,19	26,20	26,21	26,22
N =24									
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,E	2,11	2,12	2,13	2,14	2,15	2,16	2,17	2,18	2,19
G	284	312	340	368	396	424	452	480	508
DO,D1	30,14	30,15	30,16	30,17	30,18	30,19	30,20	30,21	30,22
N =28									
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,E	2,11	2,12	2,13	2,14	2,15	2,16	2,17	2,18	2,19
G	324	356	388	420	452	484	516	548	580
DO,D1	34,14	34,15	34,16	34,17	34,18	34,19	34,20	34,21	34,22
N =32									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	352	384	420	456	488	524	560	592	628
DO,D1	58,13	58,13	58,14	58,15	58,15	58,16	58,17	58,17	58,18
N =36									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	380	412	452	492	524	564	604	636	676
DO,D1	63,13	63,13	63,14	63,15	63,15	63,16	63,17	63,17	63,18
N =40									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	408	440	484	528	560	604	648	680	724
DO,D1	67,13	67,13	67,14	67,15	67,15	67,16	67,17	67,17	67,18
N =44									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	436	468	516	564	596	644	692	724	772
DO,D1	71,13	71,13	71,14	71,15	71,15	71,16	71,17	71,17	71,18
N =48									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	464	496	548	600	632	684	736	768	820
DO,D1	75,13	75,13	75,14	75,15	75,15	75,16	75,17	75,17	75,18
N =52									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	492	524	580	636	668	724	780	812	868
DO,D1	79,13	79,13	79,14	79,15	79,15	79,16	79,17	79,17	79,18
N =56									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	520	552	612	672	704	764	824	856	916
DO,D1	83,13	83,13	83,14	83,15	83,15	83,16	83,17	83,17	83,18
N =60									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	548	580	644	708	740	804	868	900	964
DO,D1	87,13	87,13	87,14	87,15	87,15	87,16	87,17	87,17	87,18
N =64									
LAMBDA,S	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,E	3, 6	3, 8	3, 6	3, 8	3,10	3,10	3,10	3,12	3,12
G	576	608	676	744	776	844	912	944	1012
DO,D1	91,13	91,13	91,14	91,15	91,15	91,16	91,17	91,17	91,18
N =68									
LAMBDA,S	3, 6	2, 0	2, 1	3, 8	2, 0	2, 1	3,10	2, 0	2, 1
LAMBDA+1,E	4, 1	3, 8	3, 6	4, 1	3,10	3,10	4, 1	3,12	3,12
G	600	636	708	776	812	884	952	988	1060
DO,D1	95,12	95,13	95,14	95,14	95,15	95,16	95,16	95,17	95,18
N =72									
LAMBDA,S	3, 6	2, 0	2, 1	3, 8	2, 0	2, 1	3,10	2, 0	2, 1
LAMBDA+1,E	4, 1	3, 8	3, 6	4, 1	3,10	3,10	4, 1	3,12	3,12
G	824	864	740	808	848	924	992	1032	1108
DO,D1	99,12	99,13	99,14	99,14	99,15	99,16	99,16	99,17	99,18
N =76									
LAMBDA,S	3, 6	2, 0	2, 1	3, 8	2, 0	2, 1	3,10	2, 0	2, 1
LAMBDA+1,E	4, 1	3, 8	3, 6	4, 1	3,10	3,10	4, 1	3,12	3,12
G	848	882	772	840	884	964	1032	1076	1156
DO,D1	103,12	103,13	103,14	103,14	103,15	103,16	103,16	103,17	103,18
N =80									
LAMBDA,S	3, 6	2, 0	2, 1	3, 8	2, 0	2, 1	3,10	2, 0	2, 1
LAMBDA+1,E	4, 1	3, 8	3, 6	4, 1	3,10	3,10	4, 1	3,12	3,12
G	872	720	804	872	820	1004	1072	1120	1204
DO,D1	107,12	107,13	107,14	107,14	107,15	107,15	107,16	107,17	107,18

Appendix II

Optimal Values LAMBDA, S LAMBDA + 1, t for Parameters β^* (LAMBDA·S + (LAMBDA + 1)·t = N), Gate Counts G and Delays $D0$, $D1$, ($D = D_W D0 + D_C D1$) for or Arrays

N	2	4	6	8	10	12	14	16	18	20
K = 4										
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	2, 1	2, 2	2, 3	2, 4	2, 5	2, 6	2, 7	2, 8	2, 9	2,10
G	1	5	11	18	27	38	51	68	89	114
D0,D1	6, 3	6, 4	6, 5	6, 6	6, 7	6, 8	6, 9	6,10	6,11	6,12
K = 8										
LAMBDA,S	1, 0	1, 0	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	2, 1	2, 2	3, 2	3, 2	3, 3	3, 4	3, 4	3, 4	3, 5	3, 5
G	1	10	15	25	34	40	49	58	64	73
D0,D1	10, 3	10, 4	16, 6	16, 7	16, 8	16, 8	16, 9	16,10	16,10	16,11
K =12										
LAMBDA,S	1, 0	1, 0	2, 0	2, 1	3, 2	2, 0	2, 1	3, 4	2, 0	2, 1
LAMBDA+1,t	2, 1	2, 2	3, 2	3, 2	4, 1	3, 4	3, 4	4, 1	3, 5	3, 5
G	1	14	20	33	45	52	63	75	84	97
D0,D1	14, 3	14, 4	20, 6	20, 7	20, 7	20, 8	20, 8	20, 9	20,10	20,11
K =16										
LAMBDA,S	0, 0	0, 0	2, 0	3, 0	3, 2	2, 0	3, 2	3, 4	2, 0	3, 4
LAMBDA+1,t	0, 0	0, 0	3, 2	4, 2	4, 1	3, 4	4, 2	4, 1	3, 5	4, 2
G	0	0	24	31	51	64	75	91	104	119
D0,D1	0, 0	0, 0	24, 6	24, 6	24, 7	24, 8	24, 8	24, 9	24,10	24,10
K =20										
LAMBDA,S	0, 0	0, 0	2, 0	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	0, 0	0, 0	3, 2	4, 2	4, 1	4, 3	4, 2	4, 4	4, 3	4, 5
G	0	0	28	42	58	73	80	104	121	125
D0,D1	0, 0	0, 0	28, 6	28, 6	28, 7	28, 7	28, 8	28, 8	28, 9	28, 9
K =24										
LAKEDA,S	0, 0	0, 0	2, 0	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	0, 0	0, 0	3, 2	4, 2	4, 1	4, 3	4, 2	4, 4	4, 3	4, 5
G	0	0	32	46	67	81	102	116	137	151
D0,D1	0, 0	0, 0	32, 6	32, 6	32, 7	32, 7	32, 8	32, 8	32, 9	32, 9
K =28										
LAMBDA,S	0, 0	0, 0	2, 0	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	0, 0	0, 0	3, 2	4, 2	4, 1	4, 3	4, 2	4, 4	4, 3	4, 5
G	0	0	36	50	78	89	114	126	153	157
D0,D1	0, 0	0, 0	36, 6	36, 6	36, 7	36, 7	36, 8	36, 8	36, 9	36, 9
K =32										
LAMBDA,S	0, 0	0, 0	2, 0	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	0, 0	0, 0	3, 2	4, 2	4, 1	4, 3	4, 2	4, 4	4, 3	4, 5
G	0	0	40	54	83	97	126	140	169	183
D0,D1	0, 0	0, 0	40, 6	40, 6	40, 7	40, 7	40, 8	40, 8	40, 9	40, 9
K =36										
LAMBDA,S	0, 0	0, 0	2, 0	3, 0	4, 0	3, 0	4, 1	3, 0	4, 2	3, 0
LAMBDA+1,t	0, 0	0, 0	3, 2	4, 2	5, 2	4, 3	5, 2	4, 4	5, 2	4, 5
G	0	0	44	58	88	106	135	152	182	198
D0,D1	0, 0	0, 0	44, 6	44, 6	60, 8	44, 7	60, 8	44, 8	60,10	44, 9
K =40										
LAMBDA,S	0, 0	0, 0	2, 0	3, 0	4, 0	3, 0	4, 1	3, 0	4, 2	3, 0
LAMBDA+1,t	0, 0	0, 0	3, 2	4, 2	5, 2	4, 3	5, 2	4, 4	5, 2	4, 5
G	0	0	48	62	82	113	143	164	194	215
D0,D1	0, 0	0, 0	48, 6	48, 6	64, 8	48, 7	64, 9	48, 8	64,10	48, 8

Computers 1-07 8611448 Gal.13

APPENDIX II (*continued*)

N	22	24	26	28	30	32	34	36	38	40
K = 4										
LAMBDA,$	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,£	2,11	2,12	2,13	2,14	2,15	2,15	2,17	2,18	2,19	2,20
G	51	56	61	66	71	76	81	86	91	96
DO,D1	6,13	6,14	6,15	6,16	6,17	6,18	6,19	6,20	6,21	6,22
K = 8										
LAMBDA,$	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2
LAMBDA+1,£	3, 6	3, 8	3, 8	3, 8	3,10	3,10	3,10	3,12	3,12	3,12
G	82	86	87	105	112	121	130	135	145	154
DO,D1	16,12	16,12	16,13	16,14	16,14	16,15	16,16	16,16	16,17	16,18
K =12										
LAMBDA,$	3, 6	2, 0	2, 1	3, 6	2, 0	2, 1	3,10	2, 0	2, 1	3,12
LAMBDA+1,£	4, 1	3, 8	3, 8	4, 1	3,10	3,10	4, 1	3,12	3,12	4, 1
G	107	116	125	138	148	161	171	180	193	203
DO,D1	20,11	20,12	20,13	20,13	20,14	20,15	20,15	20,16	20,17	20,17
K =16										
LAMBDA,$	3, 6	2, 0	3, 6	3, 6	2, 0	3, 8	3,10	2, 0	3,10	3,12
LAMBDA+1,£	4, 1	3, 8	4, 2	4, 1	3,10	4, 2	4, 1	3,12	4, 2	4, 1
G	131	144	158	171	184	198	211	224	238	251
DO,D1	24,11	24,12	24,12	24,13	24,14	24,14	24,15	24,16	24,16	24,17
K =20										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 4	4, 8	4, 6	4, 7	4, 8	4, 8	4, 7	4, 8	4, 8	4,10
G	162	155	163	197	214	228	245	258	276	290
DO,D1	28,10	28,10	28,11	28,11	28,12	28,12	28,13	28,13	28,14	28,14
K =24										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 4	4, 8	4, 8	4, 7	4, 8	4, 8	4, 7	4, 8	4, 8	4,10
G	172	186	207	221	242	256	277	291	312	326
DO,D1	32,10	32,10	32,11	32,11	32,12	32,12	32,13	32,13	32,14	32,14
K =28										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 4	4, 8	4, 8	4, 7	4, 8	4, 8	4, 7	4, 8	4, 8	4,10
G	192	206	231	245	270	284	308	323	348	362
DO,D1	36,10	36,10	36,11	36,11	36,12	36,12	36,13	36,13	36,14	36,14
K =32										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 4	4, 8	4, 8	4, 7	4, 8	4, 8	4, 7	4, 8	4, 8	4,10
G	212	226	253	268	298	312	341	355	384	398
DO,D1	40,10	40,10	40,11	40,11	40,12	40,12	40,13	40,13	40,14	40,14
K =36										
LAMBDA,$	4, 3	3, 0	4, 4	3, 0	4, 5	3, 0	4, 6	3, 0	4, 7	3, 0
LAMBDA+1,£	5, 2	4, 8	5, 2	4, 7	5, 2	4, 8	5, 2	4, 8	5, 2	4,10
G	229	246	276	293	323	340	370	387	417	434
DO,D1	60,11	44,10	60,12	44,11	60,13	44,12	60,14	44,13	60,15	44,14
K =40										
LAMBDA,$	4, 3	3, 0	4, 4	3, 0	4, 5	3, 0	4, 6	3, 0	4, 7	3, 0
LAMBDA+1,£	5, 2	4, 8	5, 2	4, 7	5, 2	4, 8	5, 2	4, 8	5, 2	4,10
G	245	266	298	317	347	365	398	419	449	470
DO,D1	64,11	48,10	64,12	48,11	64,13	48,12	64,14	48,13	64,15	48,14

N	42	44	46	48	50	52	54	56	58	60
K = 4										
LAMBDA,$	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,£	2,21	2,22	2,23	2,24	2,25	2,26	2,27	2,28	2,29	2,30
G	101	106	111	116	121	126	131	136	141	146
DO,D1	8,23	8,24	8,25	8,26	8,27	8,28	8,29	8,30	8,31	8,32
K = 8										
LAMBDA,$	2, 0	2, 1	3, 2	2, 0	2, 1	2, 2	2, 0	2, 1	3, 2	2, 0
LAMBDA+1,£	3,14	3,14	3,14	3,16	3,16	3,16	3,18	3,18	3,18	3,20
G	150	169	178	184	193	202	206	217	226	232
DO,D1	16,16	16,18	16,20	16,20	16,21	16,22	16,22	16,23	16,24	16,24
K =12										
LAMBDA,$	2, 0	2, 1	3,14	2, 0	2, 1	3,16	2, 0	2, 1	3,18	2, 0
LAMBDA+1,£	3,14	3,14	4, 1	3,16	3,18	4, 1	3,18	3,18	4, 1	3,20
G	212	225	235	244	257	267	276	289	299	308
DO,D1	20,16	20,18	20,18	20,20	20,21	20,21	20,22	20,23	20,23	20,24
K =16										
LAMBDA,$	3, 0	3,12	3,14	2, 0	3,14	3,16	2, 0	3,18	3,18	2, 0
LAMBDA+1,£	3,14	4, 2	4, 1	3,16	4, 2	4, 1	3,18	4, 2	4, 1	3,20
G	264	278	291	304	318	331	344	358	371	384
DO,D1	24,16	24,18	24,18	24,20	24,20	24,21	24,22	24,22	24,23	24,24
K =20										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 9	4,11	4,10	4,12	4,11	4,13	4,12	4,14	4,13	4,15
G	307	321	338	352	369	383	400	414	431	445
DO,D1	28,16	28,18	28,18	28,18	28,17	28,17	28,18	28,18	28,18	28,18
K =24										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 9	4,11	4,10	4,12	4,11	4,13	4,12	4,14	4,13	4,15
G	347	361	382	396	417	431	452	466	487	501
DO,D1	32,16	32,16	32,18	32,18	32,17	32,17	32,18	32,18	32,18	32,19
K =28										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 9	4,11	4,10	4,12	4,11	4,13	4,12	4,14	4,13	4,15
G	387	401	426	440	465	478	504	518	543	557
DO,D1	36,16	36,16	36,18	36,18	36,17	36,17	36,18	36,18	36,18	36,18
K =32										
LAMBDA,$	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,£	4, 9	4,11	4,10	4,12	4,11	4,13	4,12	4,14	4,13	4,15
G	427	441	470	484	513	527	556	570	599	613
DO,D1	40,15	40,16	40,16	40,18	40,17	40,17	40,18	40,18	40,18	40,18
K =36										
LAMBDA,$	4, 8	3, 0	4, 8	3, 0	4,10	3, 0	4,11	3, 0	4,12	3, 0
LAMBDA+1,£	5, 2	4,11	5, 2	4,12	5, 2	4,13	5, 2	4,14	5, 2	4,15
G	484	481	511	528	556	575	606	622	652	668
DO,D1	60,16	44,16	60,17	44,18	60,18	44,17	60,19	44,18	60,20	44,18
K =40										
LAMBDA,$	4, 8	3, 0	4, 8	3, 0	4,10	3, 0	4,11	3, 0	4,12	3, 0
LAMBDA+1,£	5, 2	4,11	5, 2	4,12	5, 2	4,13	5, 2	4,14	5, 2	4,15
G	500	521	551	572	602	623	653	674	704	725
DO,D1	64,16	48,16	64,17	48,18	64,18	48,17	64,19	48,18	64,20	48,18

Computers 1.07 8611448

Sol. 14

APPENDIX II (*continued*)

N	62	64	66	68	70	72	74	76	78	80
K = 4										
LAMBDA,S	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0
LAMBDA+1,t	2,31	2,32	2,33	2,34	2,35	2,36	2,37	2,38	2,38	2,40
G	151	156	161	166	171	176	181	186	191	196
DO,D1	8,33	6,34	6,35	6,36	6,37	8,38	6,39	6,40	8,41	8,42
K = 8										
LAMBDA,S	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1	2, 2	2, 0	2, 1
LAMBDA+1,t	3,20	3,20	3,22	3,22	3,22	3,24	3,24	3,24	3,26	3,26
G	241	250	258	265	274	280	289	296	304	313
DO,D1	16,25	16,25	16,26	16,27	16,28	16,24	16,28	16,30	16,30	16,31
K =12										
LAMBDA,S	2, 1	3,20	2, 0	2, 1	3,22	2, 0	2, 1	3,24	2, 0	2, 1
LAMBDA+1,t	3,20	4, 1	3,22	3,22	4, 1	3,24	3,24	4, 1	3,26	3,26
G	321	331	340	353	363	372	386	395	404	411
DO,D1	20,25	20,25	20,26	20,27	20,27	20,28	20,28	20,29	20,30	20,31
K =16										
LAMBDA,S	3,18	3,20	2, 0	3,20	3,22	2, 0	3,22	3,24	2, 0	3,24
LAMBDA+1,t	4, 2	4, 1	3,22	4, 2	4, 1	3,24	4, 2	4, 1	3,28	4, 2
G	398	411	424	438	451	464	478	491	504	518
DO,D1	24,24	24,25	24,26	24,25	24,27	24,28	24,28	24,29	24,30	24,30
K =20										
LAMBDA,S	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	4,14	4,16	4,15	4,17	4,18	4,18	4,17	4,19	4,18	4,20
G	462	476	493	507	524	538	555	568	586	600
DO,D1	28,20	28,20	28,21	28,21	28,22	28,22	28,23	28,23	28,24	28,24
K =24										
LAMBDA,S	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	4,14	4,16	4,15	4,17	4,18	4,18	4,17	4,19	4,18	4,20
G	522	536	557	571	582	606	627	641	662	675
DO,D1	32,20	32,20	32,21	32,21	32,22	32,22	32,23	32,23	32,24	32,24
K =28										
LAMBDA,S	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	4,14	4,16	4,15	4,17	4,16	4,18	4,17	4,19	4,18	4,20
G	582	596	621	635	660	674	689	713	736	752
DO,D1	36,20	36,20	36,21	36,21	36,22	36,22	36,23	36,23	36,24	36,24
K =32										
LAMBDA,S	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0	3, 2	3, 0
LAMBDA+1,t	4,14	4,16	4,15	4,17	4,18	4,18	4,17	4,19	4,18	4,20
G	642	656	685	698	728	742	771	785	814	828
DO,D1	40,20	40,20	40,21	40,21	40,22	40,22	40,23	40,23	40,24	40,24
K =36										
LAMBDA,S	4,13	3, 0	4,14	3, 0	4,15	3, 0	4,16	3, 0	4,17	3, 0
LAMBDA+1,t	5, 2	4,16	5, 2	4,17	5, 2	4,18	5, 2	4,19	5, 2	4,20
G	689	716	746	763	783	810	840	857	887	904
DO,D1	60,21	64,20	60,22	44,21	60,23	44,22	60,24	44,23	60,24	44,24
K =40										
LAMBDA,S	4,13	3, 0	4,14	3, 0	4,15	3, 0	4,16	3, 0	4,17	3, 0
LAMBDA+1,t	5, 2	4,16	5, 2	4,17	5, 2	4,18	5, 2	4,18	5, 2	4,20
G	755	776	806	827	857	878	908	925	959	980
DO,D1	64,21	48,20	64,22	48,21	64,23	48,22	64,24	48,23	64,25	48,24

ACKNOWLEDGMENT

The author wishes to thank A. Jones, Cambridge University, Cambridge, UK for computing data in Appendix I and Appendix II and for many useful discussions, J. Morse, the manager of Microsystem Advanced Development Group, Digital Equipment Corporation, Maynard, MA, and T. Bryan from this group for many very helpful comments.

REFERENCES

[1] E. I. Nechiporuk, "On the synthesis of gate networks," *Problemy Kibernetiki*, no. 9, 1963 (Russian).

[2] M. G. Karpovsky, "Finite Orthogonal Series," in *Analysis and Design of Digital Devices*. New York: Wiley, 1976, ch. 3.

[3] D. E. Muller, "Complexity of electronic switching circuits," *IRE Trans. Electron. Comput.*, pp. 15–19, Mar. 1956.

[4] 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.

[5] G. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.

> Au: Is [3] correct?

Manuscript received May 31, 1985; revised August 20, 1985. This work was supported in part by the National Science Foundation under Grant DCR-8317763 and by the Digital Equipment Corporation.

The author is with the College of Engineering, Boston University, Boston, MA 02215.

IEEE Log Number 8611448.

Computers 1-07 8611448 Dalley 15

TABLE I
SUMMARY OF THE RESULTS ON TWO-LEVEL AND FOUR-LEVEL IMPLEMENTATIONS

	Two-level Impl.		Four-level Implementations	
	AND ARRAY	OR ARRAY	AND ARRAY	OR ARRAY
Size of a block in a partition of inputs, α^* or β^*	1	1	$\lfloor \log_3 N - \log_3 \log_3 N \rfloor$	$\lfloor \log_2 k - \log_2 \log_2 k \rfloor$
Maximal gate count, G	$(m-1)N$	$(N-2)k$	$\frac{mN}{\log_3 N - \log_3 \log_3 N} + \frac{mN}{\log_3 N(\log_3 N - \log_3 \log_3 N)} - 2m - N$	$\frac{Nk}{\log_2 k - \log_2 \log_2 k} + \frac{Nk}{\log_2 k(\log_2 k - \log_2 \log_2 k)} - N - k$
* maximal delay, D	$(N+k, m)$	$(0, N-1)$	$(N+k+\sqrt{\frac{N}{\log_3 N}}(\lceil \log_2 \log_3 N \rceil + 1, 1 + 2\lceil \log_2 \log_3 N \rceil + \lceil \frac{m}{\log_3 N - \log_3 \log_3 N} \rceil)$	$(\sqrt{\frac{k}{\log_2 k}} \lceil \log_2 \log_2 k \rceil, 2\lceil \log_2 \log_2 k \rceil + \lceil \frac{N}{\log_2 k - \log_2 \log_2 k} \rceil)$

* $(x, y) = D_W x + D_G y$, delays resulting from input fan-outs are included in delays of AND arrays.

TABLE II
SUMMARY OF RESULTS FOR TWO-LEVEL AND FOUR-LEVEL IMPLEMENTATIONS OF A (16, 64, 16) SYSTEM

	Two-level Implementations		Four-level Implementations	
	AND ARRAY	OR ARRAY	AND ARRAY	OR ARRAY
Size of a block in a partition of inputs, α^* or β^*	1	1	$\alpha_1^*=\alpha_2^*=2$, $\alpha_3^*=\alpha_4^*=\alpha_5^*=\alpha_6^*=3$	$\beta_1^*=\ldots=\beta_{20}^*=3$, $\beta_{21}^*=4$
maximal gate count, G	960	992	408	411
maximal delay, D	(80,16)	(0,63)	(107,11)	(8,25)

TABLE III
SUMMARY OF ASYMPTOTICAL RESULTS (m, N, $k \to \infty$, $m/\log_3 N$, $N/\log_2 k \to \infty$) FOR TWO-LEVEL IMPLEMENTATIONS AND FOUR-LEVEL IMPLEMENTATIONS

	Two-level Implementations		Four-level Implementations	
	AND ARRAY	OR ARRAY	AND ARRAY	OR ARRAY
Size of a block in a partition of inputs, α^* or β^*	1	1	$\log_3 N - \log_3 \log_3 N$	$\log_2 k - \log_2 \log_2 k$
maximal gate count, G	mN	Nk	$\frac{mN}{\log_3 N} + 2\log_2 \log_3 N$	$\frac{Nk}{\log_2 k} + 2\log_2 \log_2 k$
maximal delay, D	$(N+k, m)$	$(0, N)$	$(N+k, \frac{m}{\log_3 N})$	$(0, \frac{N}{\log_2 k})$

TABLE IV
GATE COUNTS OF TWO- THREE- AND FOUR-LEVEL IMPLEMENTATIONS OF PERIPHERAL CONTROL UNITS FOR VAX COMPUTERS

#	Number of inputs, m	Number of product terms, N	Number of outputs, k	Gate count for 2-levels			Gate count for 3-levels			Gate count for 4-levels		
				AND	OR	TOTAL	AND	OR	TOTAL	AND	OR	TOTAL
1	18	51	28	265	80	345	149	80	229	149	54	203
2	22	48	28	274	63	337	180	63	243	180	48	228
3	21	48	28	235	48	283	113	48	161	113	35	148
4	20	48	26	280	57	337	159	57	216	153	43	296
5	14	48	24	319	113	432	151	113	264	151	92	243
6	16	57	28	477	150	627	203	150	353	203	124	327
7	18	51	14	325	100	425	149	100	249	161	75	236
8	22	48	14	386	121	407	204	121	325	204	90	294
9	48	48	14	237	60	297	114	60	174	114	45	159
10	20	48	13	308	75	383	166	75	241	166	58	224
11	14	38	6	237	31	268	121	31	152	121	25	146

Computers 1-07 8611448

Gallery 16

TABLE V
TOTAL GATE COUNTS AND DELAYS FOR TWO-LEVEL AND FOUR-LEVEL IMPLEMENTATIONS OF PERIPHERAL CONTROL UNITS FOR VAX COMPUTERS

#	2 - level		4 - level	
	Gate Count	Delay (D_W, D_G)	Gate Count	Delay (D_W, D_G)
1	345	(56,32)	203	(37,22)
2	337	(35,30)	228	(25,25)
3	283	(44,27)	148	(27,29)
4	337	(41,26)	202	(28,24)
5	432	(51,26)	237	(31,25)
6	627	(41,33)	327	(32,32)

TABLE VI
RESULTS OF A SIMULATION OF LAYOUTS IN THE 2200 GATE ARRAY FOR TWO (16, 64, 16) SYSTEMS WITH $d = 1$ AND $d = 0.66$

	d = 1		d = 0.66	
	2-level	4-level	2-level	4-level
Number of cells	1,177	730	1,087	708
Max pins/net	36	10	36	19
Number of nets with > 10 pins/net	53	0	52	13
Number of nets with > 20 pins/net	32	0	22	0
Number of gates	1,712	996	1,576	1,117
Gate usage (%)	76.98	44.78	70.86	50.22
Area usage (%)	77.25	54.14	71.18	60.79
Number of signal nets	1,161	714	1,071	693
Average pins/net	3,293	3,062	3,144	3,029

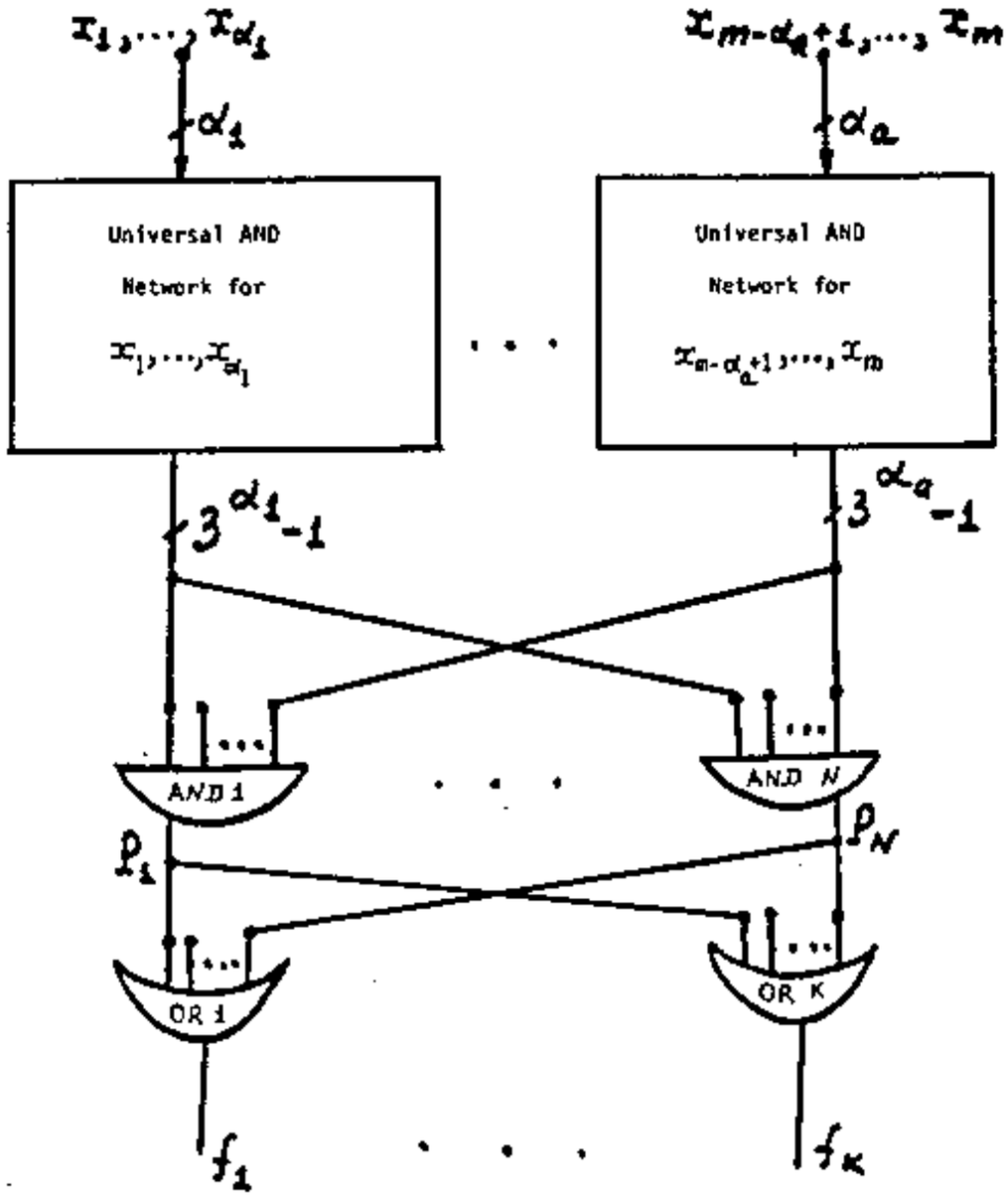


Fig. 3. AND-AND-OR implementation of an (m, N, k) system.

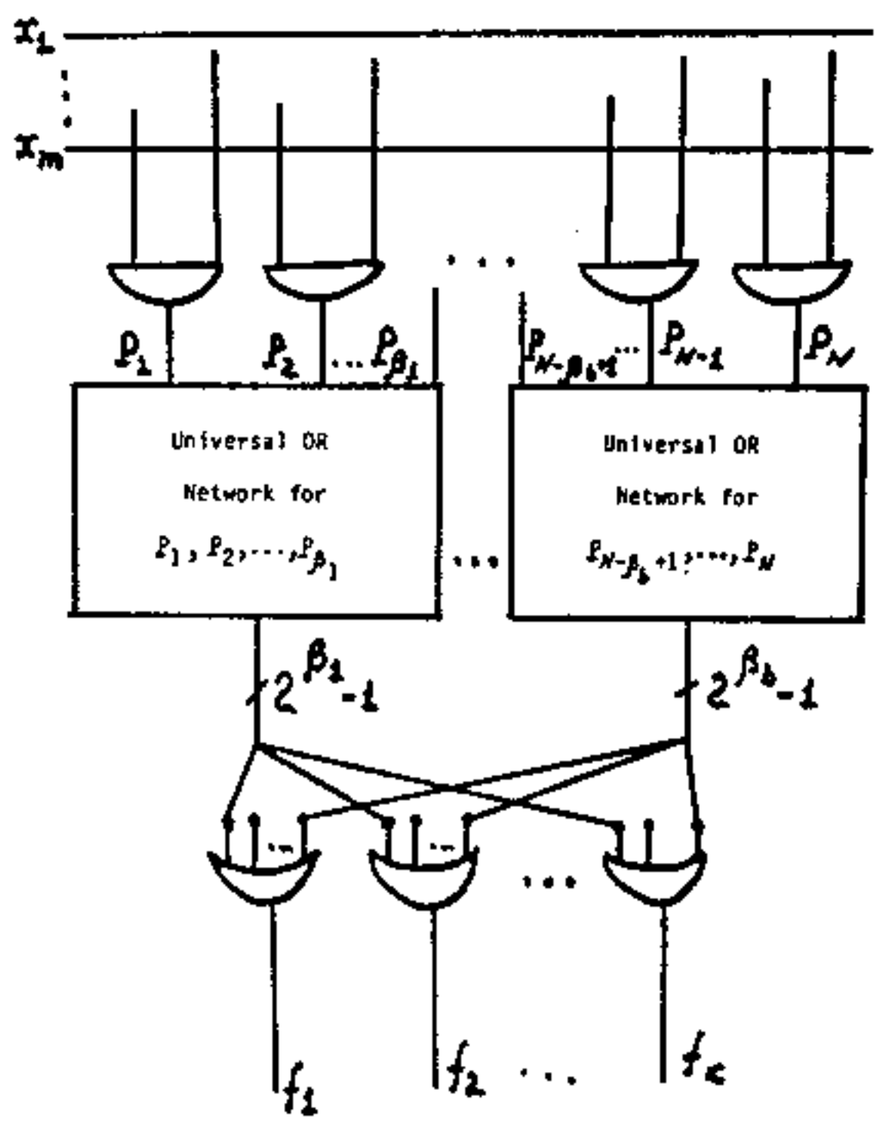
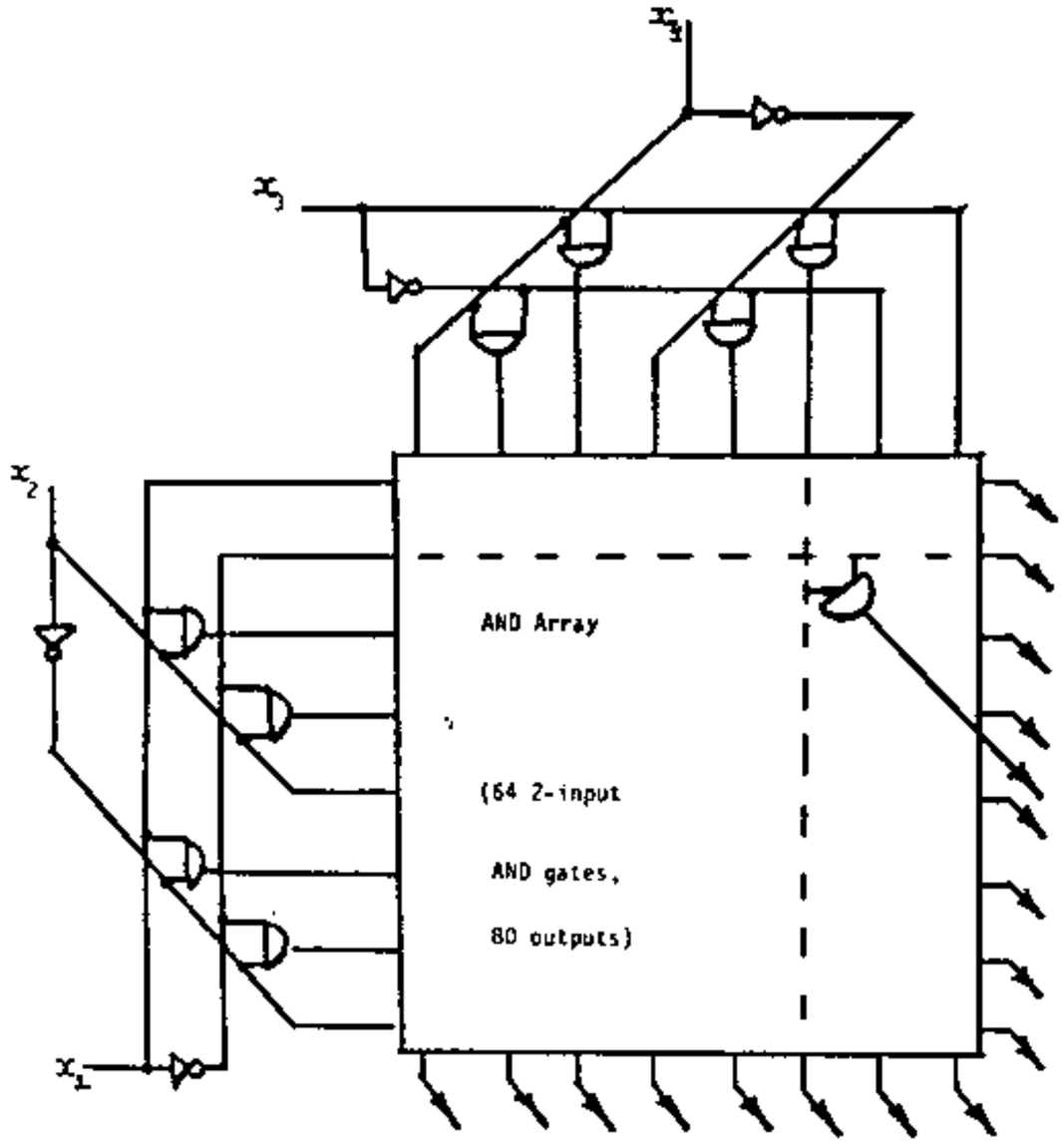


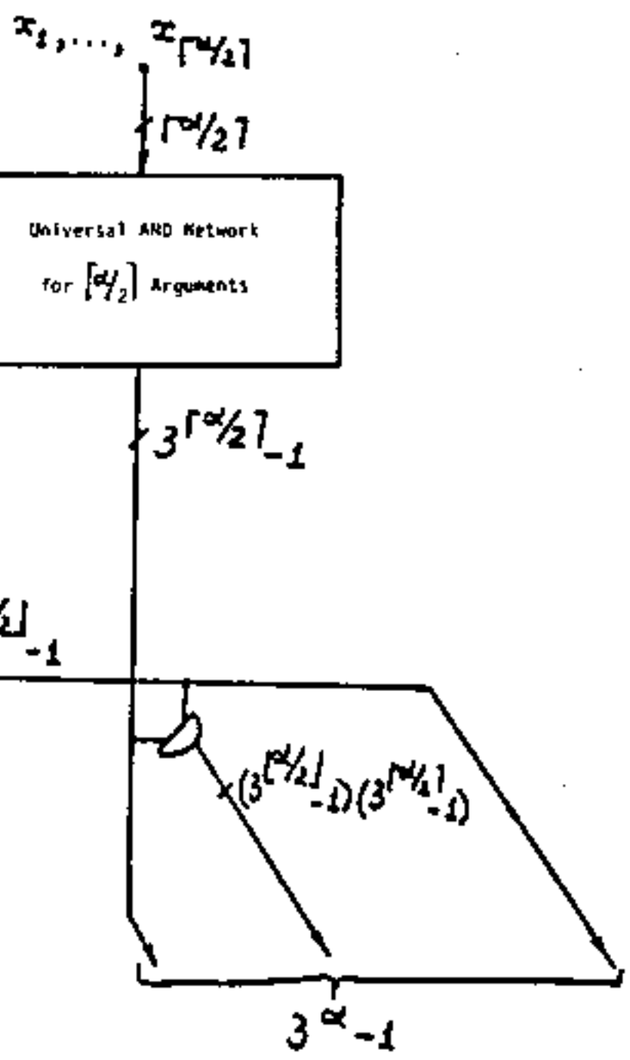
Fig. 4. AND-OR-OR implementation of a (m, N, k) system.

Computer 1-07 861448

Galley 17

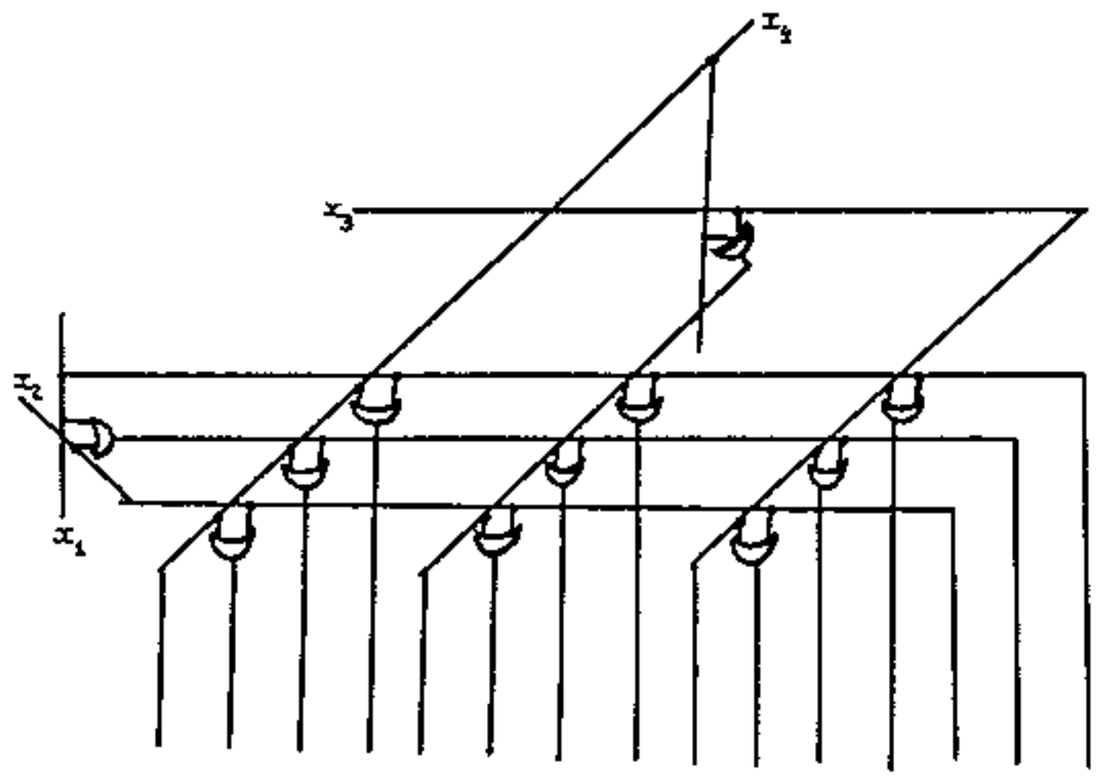


(a)

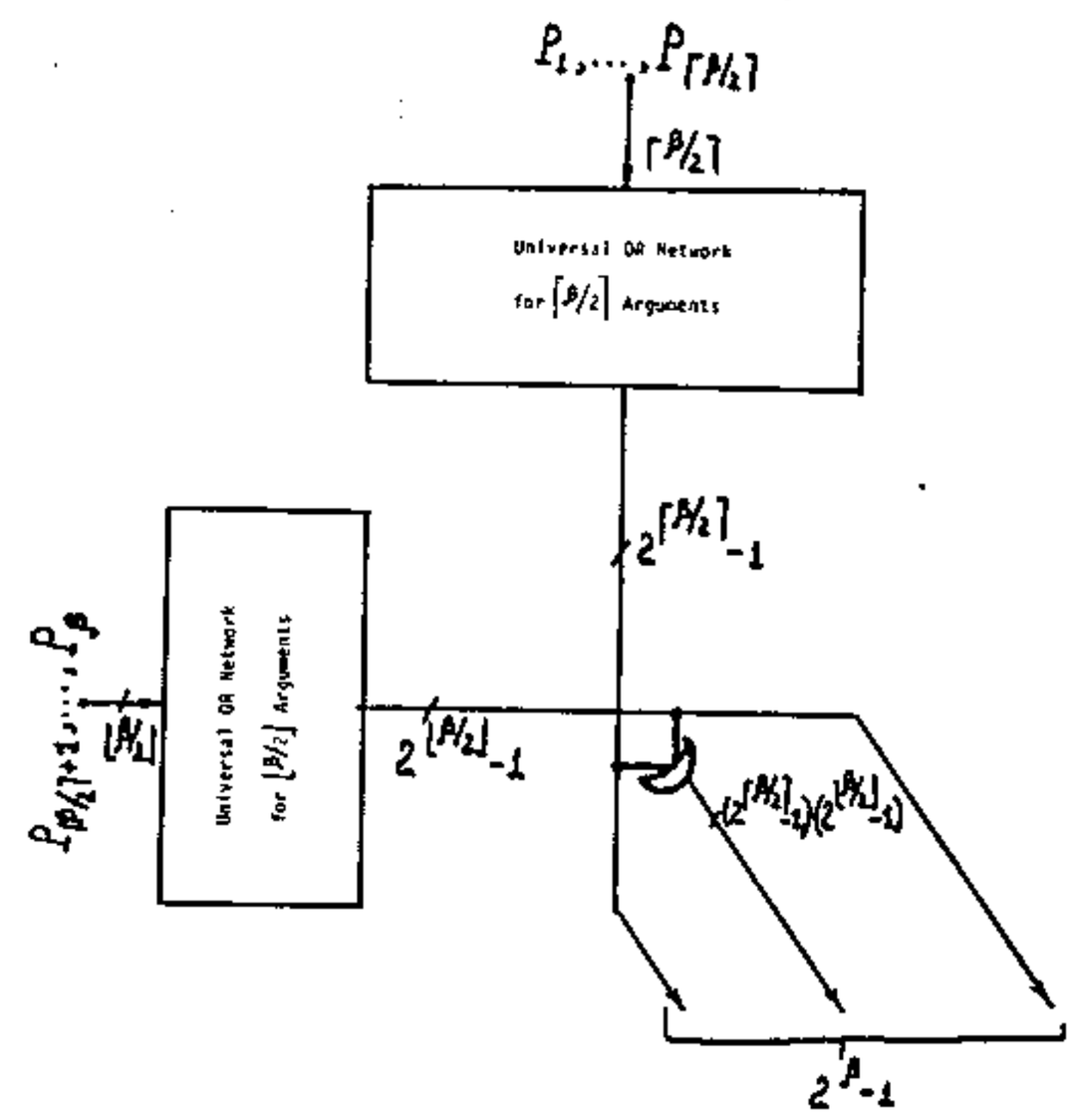


(b)

Fig. 1. (a) Universal AND network for 4 arguments. (b) Recursive procedure to construct universal AND networks for arguments $x_1, \cdots, x_\alpha$.



(a)



(b)

Fig. 2. (a) Universal OR network for 4 arguments. (b) Recursive procedure to construct the universal OR network for arguments $P_1, \cdots, P_\beta$.

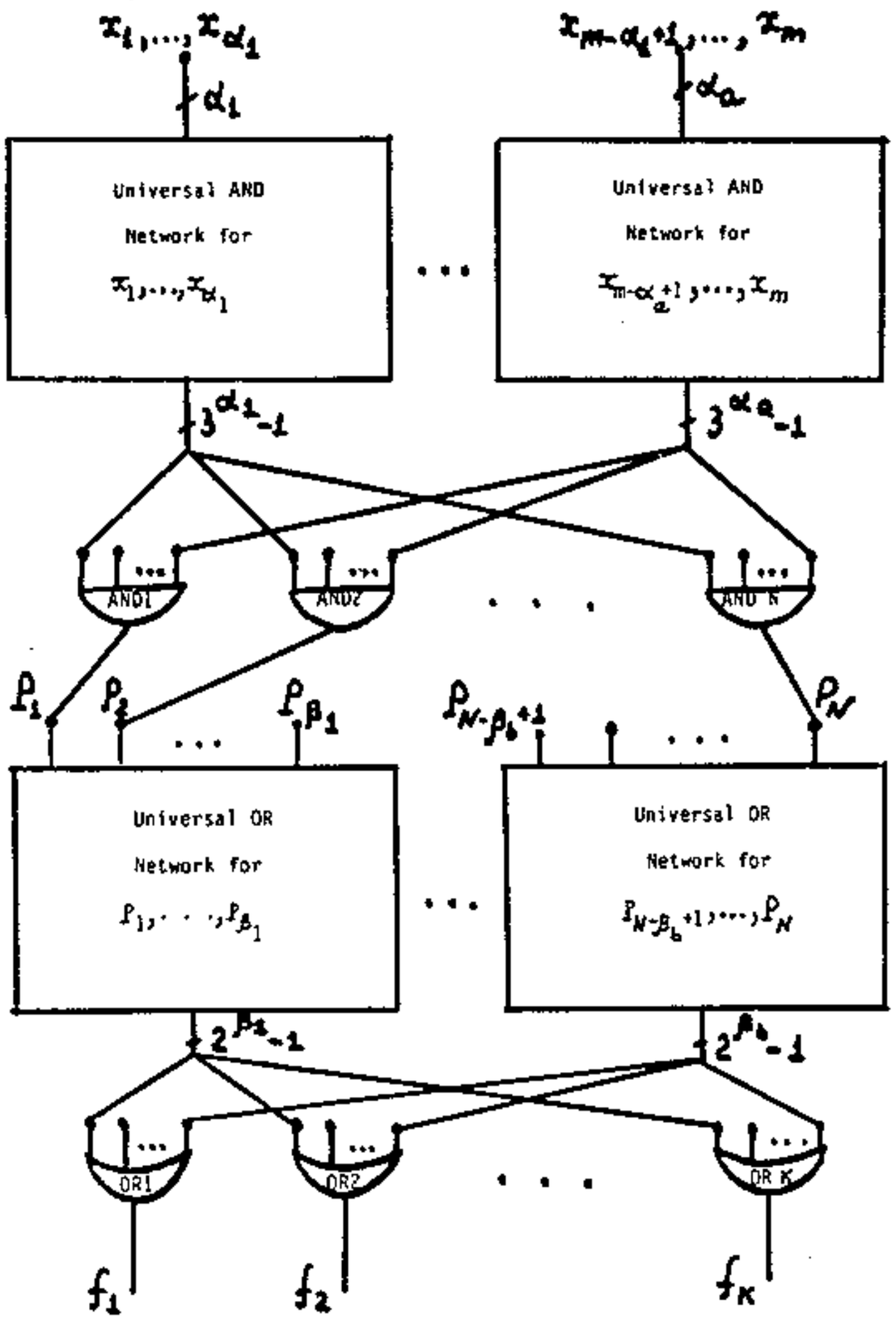


Fig. 5. AND-AND-OR-OR implementation of an (m, N, k) system.

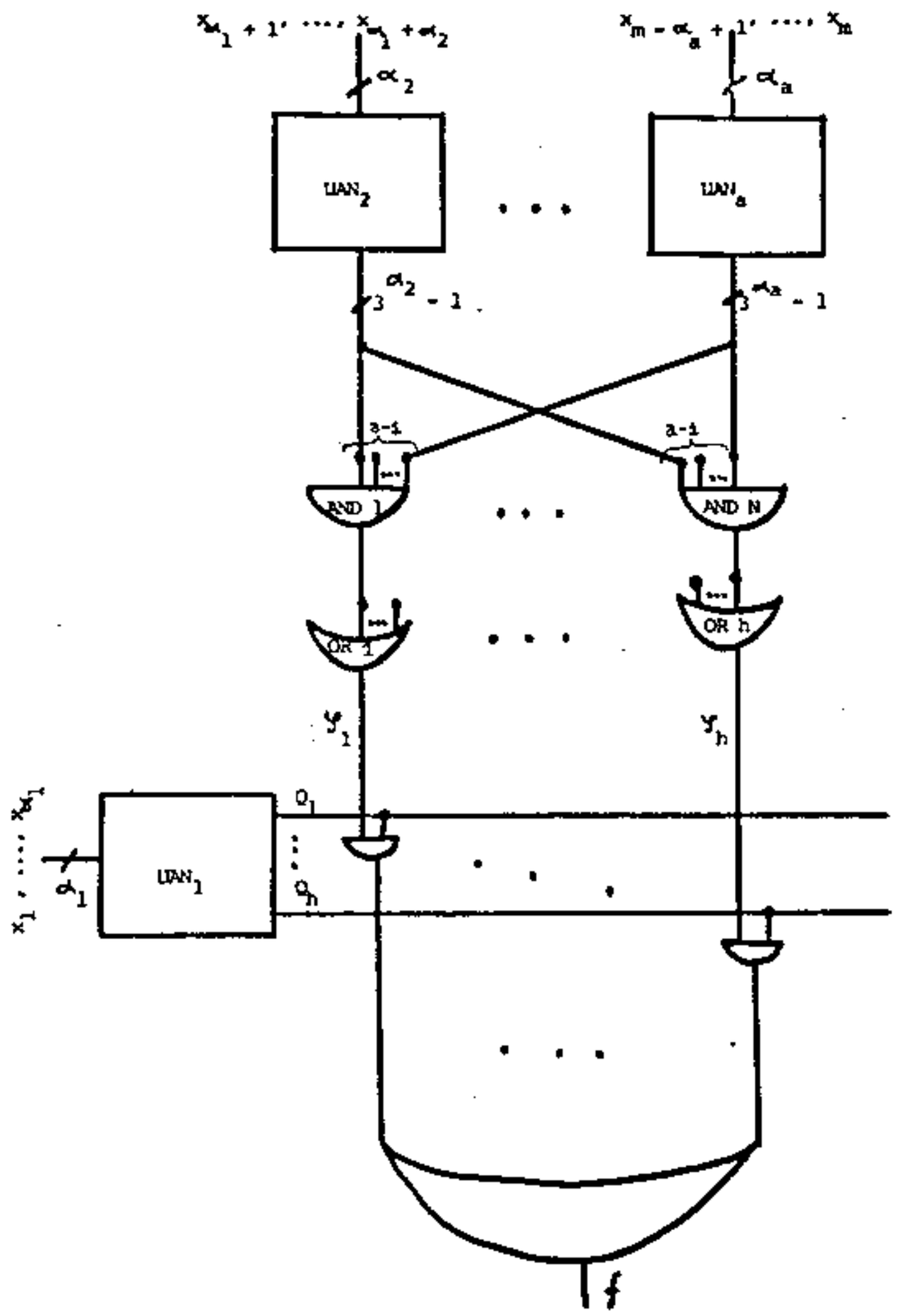


Fig. 7. Multilevel implementation of an (m, N, l) system.