Some Results on Syntheses of Sorting Networks Based on the Two-dimensional Arrangement of Signal Lines

Toru KAWATA*, Toshiaki CHIBA** and Hiroshi OZAKI*

Abstract

In this paper, we investigate some conservative properties of partial odering generated in the process of synthesis of sorting networks that use comparators. An improved synthesis algorithm for 2^{2r+1} inputs and an algorithm suitable for the synthesis of sorting networks that use 4-sorters are derived from these properties.

1. Introduction

The study of sorting networks that use comparators is motivated by such areas of research as permutation networks, nonadaptive sorting algorithms for pipeline computers and parallel sorting algorithms, as well as by the desire to build special hardware for sorting. Synthesis algorithms for sorters usually give rather economical construction for the case where the number of inputs has some special forms, but lose their effectiveness for the case where the number of inputs has not such forms.

In this paper, we study some improvements on N-sorter construction based on some preservation of partial ordering relations generated in the process of synthesis.

2. Preliminaries

Without loss of generality, in this paper, it is assumed that we study the synthesis of switching circuits which sort any given input sequence, which is a permutation of $I=\{1, 2, \ldots, N\}$, in ascending order. Furthermore, for clarity of discussion, the mutual ordering of signal lines is fixed through the network, as shown in Fig. 1. A denotes the set of line numbers and is equal to $\{0, 1, \ldots, N-1\}$. $\phi(i)$, $(i \in A)$ denotes the datum $(\in I)$ on the signal line i. Thus, ϕ is a function mapping A onto I and is called the data function. Let S_N be an N-input sorting network, or N-sorter. $(i : j)$ denotes a comparator, or C_2-cell, which manipulates the data on lines i and j. The equation $X(i : j) = Y$ means that Y is the resulting sequence which is obtained by applying $(i : j)$ to the sequence X. A C_2-network is a network with N

This paper first appeared in Japanese in Joho—Shori (Journal of the Information Processing Society of Japan), Vol. 17, No. 11 (1976), pp. 1033~1040.

* Department of Electronic Engineering, Faculty of Engineering, Osaka University
** Research and Development Center, Tokyo Shibaura Electric Co., Ltd.

inputs and N outputs constructed only by C_2-cells. $|\theta|$ denotes the number of C_2-cells in C_2-network θ and ϕ_θ denotes the data function at the outputs of C_2-network θ.

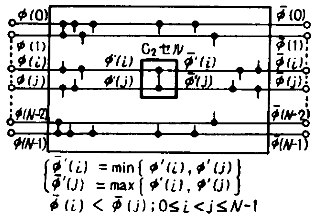


Fig. 1 Sorting network S_N with N inputs and its component (C_2-cell, usually called comparator).

<u>Definition 1</u> Let α be a C_2-network, and $T = \{t_i \mid t_i$ is a nonnegative integer, and $t_i < t_j$ for $i<j\}$. $S(\alpha, T)$ is defined the C_2-network achieved by replacing each cell $(i : j)$ in α with a cell $(t_i : t_j)$.

" p/q " and " $p//q$ " represent the quotient and the remainder that result from the integer division of p by q, respectively.

If P is a partial ordering relation, the relation C_P is defined as follows: $C_P = \{ (x, y) \varepsilon P \mid x \neq y$ and $(x, y) \varepsilon P$ and $(z, y) \varepsilon P \Rightarrow x = y$ or $y = z \}$.

<u>Definition 2</u> The function ϕ is consistent with a partial ordering relation P over A, if " iPj and $i \neq j \Rightarrow \phi(i) < \phi(j)$ ".

<u>Definition 3</u> A partial ordering relation F over A is called a shift if (i) $(i, j) \varepsilon F$ and $(i', j) \varepsilon F \Rightarrow i = i'$, (ii) $(i, j) \varepsilon F$ and $(i, j') \varepsilon F \Rightarrow j = j'$, and (iii) $(i, j) \varepsilon F^k$ and $(i, j) \varepsilon F^{k'} \Rightarrow k = k'$. A maximal sequence of signal line numbers i_1, i_2, ... , i_k ($i_j \varepsilon A$, $j = 1, \ldots, k$) such that $F(i_j) = i_{j+1}$ ($0 \leq j \leq k-1$) is called a chain of the shift F.

Given the data function ϕ_1 and the shift F over A, for any chain $i_1\, i_2 \ldots i_k$ of F there exists a unique data function ϕ_2 such that (i) $\phi_2(i_1), \ldots, \phi_2(i_k)$ is a permutation of $\phi_1(i_1), \ldots, \phi_1(i_k)$ and (ii) $p < q \Rightarrow \phi_2(i_p) < \phi_2(i_q)$. Then, if the above-mentioned data function ϕ_1 and ϕ_2 are consistent with the partial ordering relation P over A, it is said that F preserves P.

<u>Definition 4</u> $A(d_1, d_2)$ denotes a two-dimensional arrangment of A with d_1 rows and d_2 columns. Then, the element of A corresponding to line number $x_1+d_2 \cdot x_1$ is represented by an ordered pair (x_1, x_2).

<u>Definition 5</u> Let a partial ordering relation P over $A(d_1, d_2)$ be defined as follow: for any two elements (x_1, x_2) and (x_1', x_2'), if $x_1 = x_1'$ and $x_2<x_2'$, or if $x_2 = x_2'$ and $x_1<x_1'$, then $(x_1, x_2)\ P\ (x_1', x_2')$.

If the data function $\phi : A \rightarrow I$ is consistent with this partial ordering relation P, it is said that the set A is in the two-dimensional grid relation. In this paper, we will derive synthesis algorithms based on this relation; and the following is its basis.

Theorem 1 [2] Let shifts F_a and F_b be defined over $A(d_1, d_2)$ as follows: for $X = (x_1, x_2)$ and $Y = (y_1, y_2)$, $(X, Y \in A)$, $(X, Y) \in F_a \Leftrightarrow x_i + a_i = y_i$ $(i = 1, 2)$; $(X, Y) \in F_b \Leftrightarrow x_i + b_i = y_i$ $(i = 1, 2)$, where $a = (a_1, a_2)$, $b = (b_1, b_2)$, $a_1^2 + a_2^2 \neq 0$, and $b_1^2 + b_2^2 \neq 0$.

Table 1 Numerical results of comparison with three typical synthesis methods

方法＼入力数	32	128	512	2048	8192
I	191	1471	9727	58376	327679
II	198	1455	9888	60628	340636
III	187	1419	9347	54533	309972

I: Batcher's method [1]
II: The iterative application of Theorem 3 to Van Voorhis' method
III: The method in this paper

Then the following are euivalent : (i) F_a preserves F_b^* and F_b preserves F_a^* ; (ii) $a_i b_i \geq 0$ $(i = 1, 2)$, where R^* is the reflexive transitive closure of the relation R over A.

Corollary 2 Column-sorting preserves row-sorting and vice versa.

Theorem 3 [4], [5] Let α be an N-sorter. Let β be the C_2-network obtained by deleting the C_2-cells, through which the maximum datum input at signal line $(N-1)$ should pass, from α. Then β is an $(N-1)$-sorter.

Given an algorithm for some special form of the number of inputs, we can derive a synthesis algorithm for the case where the number of inputs has not such a form, by the iterative application of this theorem. But, when this is applied to the Van Voorhis' algorithm for 2^{2r+1} inputs, it is less effective than the Batcher's one, as shown in Table 1.

3. Preservations of Partial Ordering

Two shifts F_1 and F_2 over $A(d_1, d_2)$ are defined as follows :

$(i, j)\ F_1\ (i, j+1)$; $0 \leq i \leq d_1-1$, $0 \leq j \leq d_2-2$

$(i, j)\ F_2\ (i+1, j)$; $0 \leq i \leq d_1-2$, $0 \leq j \leq d_2-1$

Then, by the corollary, F_2 preserves F_1^*. Thus, we can generate the grid relation by cascading a C_2-network α_2 which realizes F_2 to a C_2-network α_1 which realizes F_1. α_1 and α_2 are given in the equations (2) and (3).

$$\alpha_1 = \alpha(d_2)\ S(\alpha(d_2), \{\lambda \mid (\lambda/d_2) = 1\}) \ldots S(\alpha(d_2), \{\lambda \mid (\lambda/d_2) = d_2-1\}) \quad (2)$$

$$\alpha_2 = S(\alpha(d_1), \{\lambda \mid (\lambda//d_2) = 0\}) \ldots S(\alpha(d_1), \{\lambda \mid (\lambda//d_2) = d_2-1\}) \quad (3)$$

After generating the grid relation over $A(d_1, d_2)$ by $\alpha_1\alpha_2$, the set A is partitioned into p subsets as follows : $Q_k^p = \{X \mid ((X/d_2)//p) = k\}$, $0 \leq k \leq p-1$ (4)

In the following discussion, we assume that $d_1 = d_1' \cdot p$; but its extension to the general case is easy.

Corresponding to Q_k^p, let the C_2-network β_2 be defined in the equation (5) and let the shift F_p be defined in the equation (6) and (7). Then β_2 is a realization of F_p.

$\beta_p = S(\ \theta_2(\ d_1',\ d_2\ ),\ Q_0^p\ )\ \ldots\ S(\ \theta_2(\ d_1',\ d_2\ ),\ Q_i^p\ )\ \ldots\ (\ \theta_2(\ d_1',\ d_2\ ),\ Q_{p-1}^p)$ (5)

$(\ i,\ j\ )\ F_p\ (\ i,\ j+1\ )\ ;\ 0 \leq i \leq d_1-1,\ 0 \leq j \leq d_2-2$ (6)

$(\ i,\ d_2-1\ )\ F_p\ (\ i+p,\ 0\ )\ ;\ 0 \leq i \leq d_1-p-1$ (7)

<u>Theorem 4</u> The shift F_p preserves the partial ordering relation P (= C_p^+) specified by C_p, defined as follows : $(\ i,j\ )\ C_p\ (\ i,\ j+1\ )\ ;\ 0 \leq i \leq d_1-1,\ 0 \leq j \leq d_2-2$, $(\ i,\ j\ )\ C_p\ (\ i+1,\ j\ )\ ;\ 0 \leq i \leq d_1-2,\ 0 \leq j \leq d_2-1$.

4. An Economical Construction of $S_{2^{2r+1}}$

In this section, it is assumed that $N = 2^{2r+1}$, $d_1 = 2^r$, and $d_2 = 2^{r+1}$. After generating the two-dimensional grid relation over $A(\ 2^r,\ 2^{r+1}\ )$ at the outputs of $\alpha_1\alpha_2$, the set $A(\ 2^r,\ 2^{r+1}\ )$ is partitioned into two subsets as follows : One of them is the set $B(\ 2^r,\ 2^r\ ) = \{\ (\ i,\ j\ ) \epsilon A |\ j$ is even $\}$, and the other is the set $C(\ 2^r,\ 2^r\ ) = \{\ (\ i,\ j\ ) \epsilon A |\ j$ is odd $\}$. Clearly, each of B and C is in the grid relation because of the transitive law of the partial ordering.

Let the C_2-networks β_2' and δ be defined as follows :

$\beta_2' = S(\ \theta_2(\ 2^r,\ 2^r\ ),\ Q_0^2\ )\ S(\ \theta_2(\ 2^r,\ 2^r\ ),\ Q_1^2\ )$, where $\theta_2(\ 2^r,\ 2^r\ )$ is a C_2-network which satisfies such a condition that $\alpha_1\alpha_2\theta_2$ is a sorter for $A(\ 2^r,\ 2^r\ )$.

$\delta = \delta_{r-1}\delta_{r-2}\ \ldots\ \delta_0$, where $\delta_j = (1:2^{j+1})\ (3:2^{j+1}+2)\ \ldots\ (2^{2r+1}-2^{j+1}-1:2^{2r+1}-2)$, $0 \leq j \leq r-1$. Then, the following property is obtained.

<u>Theorem 5</u> Cascading the C_2-network δ_j to the C_2-network $\alpha_1\alpha_2\beta_2'\delta_{r-1}\delta_{r-2}\ \ldots\ \delta_j$ ($0 \leq j \leq r-2$), the data function $\bar{\phi}_{\alpha_1\alpha_2\beta_2'\delta_{r-1}\delta_{r-2}\ \ldots\ \delta_{j+1}\delta_j}$ preserves the partial ordering relation specified by the equations (2), (3) and (5), and the relations augmented by means of the applications of δ_{r-1}, ... , δ_{j+1}.

Applying this theorem in the case where $j = 0$, we can obtain the following inequalities: $\bar{\phi}(\ i\ ) < \bar{\phi}(\ i+1\ );\ i = 1,\ 3,\ \ldots\ ,\ 2^{2r+1}-3$. Since $\bar{\phi}$ preserves the relation specified by the equation (2), then $\bar{\phi}(\ i\ ) < \bar{\phi}(\ i+1\ );\ i = 0,\ 2,\ \ldots\ ,\ 2^{2r+1}-2$.

Thus, the C_2-network $\alpha_1\alpha_2\beta_2'\delta$ is a realization of $S_{2^{2r+1}}$.

Among the synthesis algorithms at whose first stages the k-dimensional grid relation over the k-dimensional arrangement of A is generated, it is known that every algorithm requires the number of C_2-cells represented as follows [3]: $\frac{1}{4}N(\log_2 N)^2 - \lambda N \log_2 N + O(N)$. Setting $\lambda = \lambda_1$ for our construction and $\lambda = \lambda_2$ for the Van Voorhis' one, we can derive that $\lambda_1 = \lambda_2$ from the following equalities.

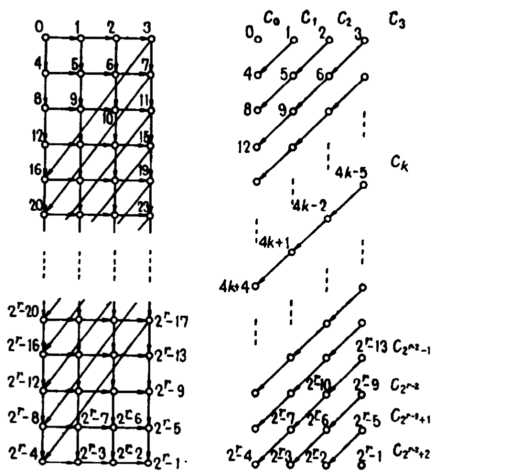
$$|\alpha_1\alpha_2\beta_2'\delta| = |\alpha_1| + |\alpha_2| + |\beta_2'| + |\delta| = \frac{1}{4}\ 2^{2r+1}(2r+1)^2 - \lambda_1 2^{2r+1}(2r+1) + O(2^{2r+1}).$$

Thus, the method introduced in this section keeps the efficiency of the Van Voorhis' algorithm. Actually, as shown in the numerical example of Table 1, this method (III) is superior to the Batcher's method (I) and the application of Theorem 3 to the Van Voorhis' one (II).

5. Construction of S_{2^r} by 4-sorter

In this section, based on the case where $p = 4$ and $d_2 = 4$ in Theorem 4, we will propose an efficient construction of S_{2^r} that uses 4-sorters.

First, the C_2-network $\alpha_1\alpha_2$, which is derived by $d_1 = 2^{r-2}$ and $d_2 = 4$ in the equations (2) and (3), is applied to generate the two-dimensional grid relation. Then, partitioning the set A into four subsets Q_0^4, Q_1^4, Q_2^4 and Q_3^4 by setting $p = 4$ in (4), and applying the C_2-network β_4 corresponding to (5), the data function $\bar{\phi}_{\alpha_1\alpha_2\beta_4}$ is consistent with the partial ordering P_4 specified by C_p in Fig 2(a). This corresponds to the case where $p = 4$ in Theorem 4.



(a) C_{P_4} (Partial ordering relation $P_4 = C_{P_4}^+$ and $\bar{\phi}_{\alpha_1\alpha_2\beta_4}$ is consistent with P_4).

(b) A shift F_t, which consists of $(2^{r-2}+3)$ chains $C_0, C_1, \cdots, C_{2^{r-2}+2}$, and conserves the partial ordering relation P_4.

Fig. 2 Partial ordering relation C_{P_4} ($P_4 = C_{P_4}^+$, and P_4 is generated at the output of C_2-network $\alpha_1\alpha_2\beta_4$) and a shift F_t, which conserves P_4.

Now, introduce the shift F_t as follows: $(4i+1,\ 4i+4) \in F_t$; $0 \leq i \leq 2^{r-2}-2$, $(4i+2,\ 4i+5) \in F_t$; $0 \leq i \leq 2^{r-2}-2$, $(4i+3,\ 4i+6) \in F_t$; $0 \leq i \leq 2^{r-2}-2$.

This shift is depicted in Fig. 2(b).

<u>Theorem 6</u> The shift F_t preserves P_4.

The C_2-network γ which realizes the shift F_t is given as follows.

$$\gamma = (1{:}4)\ S(S_3, C_2)\ S(S_4, C_3)\ \ldots\ (S_4, C_{2^{r-2}-1})\ S(S_3, C_{2^{r-2}})\ (2^r-5{:}2^r-2).$$

After realizing F_t, only the relations among $\{4k-2,\ 4k-1,\ 4k,\ 4k+1\}$ $(1 \leq k \leq 2^{r-2}-1)$ are left ambiguous by the theorem. Then, let the C_2-network δ be as follows:

$$\delta = \delta_1\delta_2 \ldots \delta_{2^{r-2}-1}, \text{ where } \delta_k = (4k-2{:}4k)(4k-1{:}4k+1)(4k-1{:}4k).$$

Now, the C_2-network $\alpha_1\alpha_2\beta_4\delta\gamma$ is an N-sorter. The number $K_4(N)$ of 4-sorters required by the above-mentioned construction of an N-sorter satisfies such a condition that $K_4(2^r) = 8K_4(2^{r-2}) - 16K_4(2^{r-4}) + 2^{r-1} - 1$. This recursion can be solved as in the equation (8), and (9).

$$K_4(N) = \frac{1}{16}N(\log_2 N)^2 - \frac{1}{24}N\log_2 N + \frac{1}{9}(N-1); \text{ where } r \text{ is even} \tag{8}$$

$$K_4(N) = \frac{1}{16}N(\log_2 N)^2 - \frac{1}{48}N\log_2 N + \frac{1}{72}N - \frac{1}{9}; \text{ where } r \text{ is odd} \tag{9}$$

Besides, the direct application of the construction that uses C_2-cells to the construction that uses 4-sorters is based on pairing two adjacent signal lines.

The number of $V_4(N)$ of 4-sorters required by this metod is represented as follows:

$V_4(N) = \frac{1}{8}N(log_2N)^2 - (\frac{1}{4} + \frac{\lambda}{2})Nlog_2N + O(N)$, where $0.25 \leq \lambda \leq 0.395$.

This estimation shows that the method in this paper is rather efficient.

REFERENCES

[1] K. E. Batcher: Sorting networks and their applications, Proc. SJCC, p. 307 (1968).

[2] D. Gale and R. M. Karp: A phenomenon in the theory of sorting, IEEE Conf. Rec. of 11th Ann. Symp. on SWAT, p. 51 (1970).

[3] D. C. Van Voorhis: An economical construction for sorting networks, AFIPS Conf. Proc., Vol. 43, p. 921 (1974).

[4] D. E. Knuth: Sorting and searching, The Art of Computer Programming Vol.3, Addison Wesley, p. 220 (1973).

[5] M.W. Green: Some improvements in nonadaptive sorting algorithms, Proc. of 6th Ann. Princeton Conf. on ISS, p. 387 (1972).