数理解析研究所講究録
第458巻 1982年 113-121

ON PERMUTATIONS OF WIRES AND STATES

Frank M. Brown[†] and Yoshihide Igarashi[††]

[†]Department of Electrical Engineering, University of Kentucky

[††]Department of Computer Science, Gunma University

1. Introduction

Consider a permutation network with N input terminals and N output terminals. We denote this permutation network by PERMU(N). Suppose that the i_1-th, i_2-th, . . ., and i_k-th input terminals of PERMU(N) are cyclically permuted to the i_2-th, i_3-th, . . . ,i_k-th and i_1-th output terminals. We call this sub-permutation on PERMU(N) a wire-cycle. Any permutation connection between input terminals and output terminals of PERMU(N) can be expressed as a product of wire-cycles.

We assume that a signal at each terminal of PERMU(N) is any of integer from the modulo-p field J_p= (0, 1, . . .,p-1), where p is a prime integer. The usual binary case then corresponds to p=2. Then we can interpret that PERMU(N) is a converter from p-nary numbers of N digits at input terminals to p-nary numbers of N digits at output terminals. A p-nary number or its corresponding decimal number at the input terminals (at the output terminals) under this interpretation is called an input-state (output-state). A state transformation induced by a permutation connection on PERMU(N) is a permuta-

tation on $\{0, 1, \ldots, p^N-1\}$. Suppose that input-states $A_1, A_2, \ldots, A_k$ are cyclically permuted to output-states $A_2, A_3, \ldots, A_k, A_1$ by a permutation connection on PERMU(N). We call this sub-permutation of states a state-cycle. Any state permutation induced by a permutation connection can be expressed as a product of state-cycles.

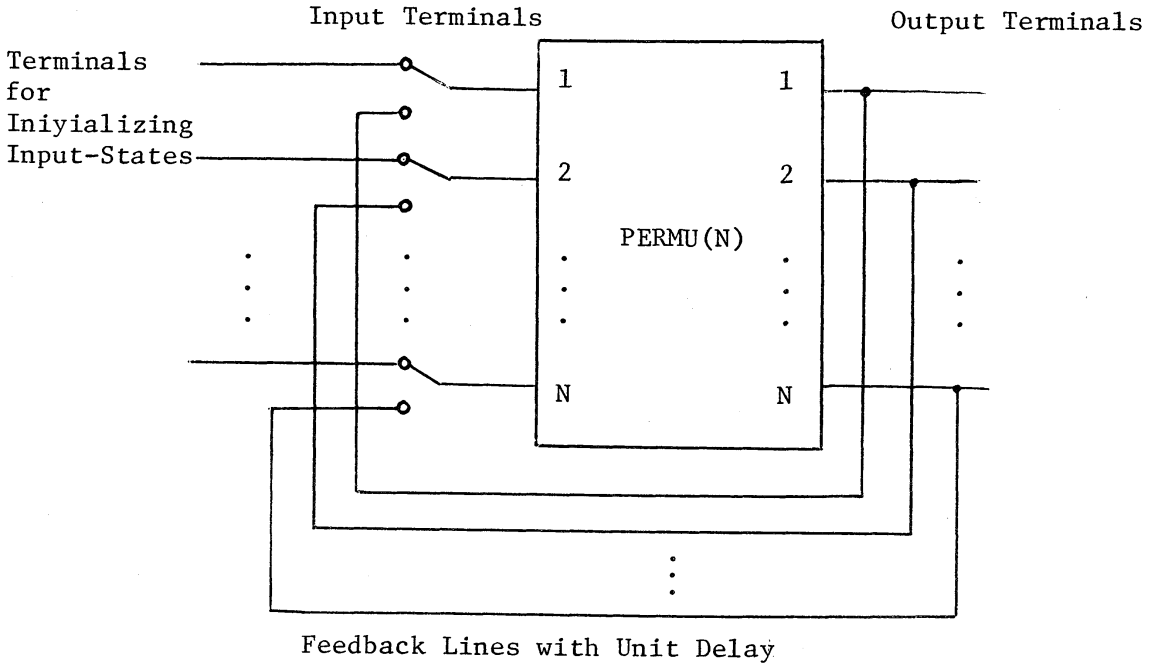

Fig. 1 A cyclic state generator

PERMU(N) can be used as a cyclic state generator as shown in Fig. 1. For this purpose cycle periods of state-cycles generated by the cyclic state generator are important characters of a permutation connection. The study discussed in this paper was motivated by a desire to achieve a basic understanding of the relation between a permutation connection and cyclic behavior of signals of cyclic periods. In particular, we study the relation between cycle periods of wire-cycles and cycle periods of state-cycles.

2. Characterization and Combinatorial Studies

We characterize each permutation connection by cycle periods of wire-cycles and state-cycles. We call these characteristics a wire-sequence and

a state-sequence, respectively. The wire-sequence of permutation π is denoted by $W(\pi)$, and the state-sequence induced by π on PERMU(N) is denoted by $S_p(\pi)$. $W(\pi)$ and $S_p(\pi)$ are formally defined as follows: Let $\theta_\pi(i)$ be the number of wire-cycles of period i in the expression of the product of wire-cycles for π. Let $\overline{\theta}_\pi(i)$ be the number of state-cycles of period i in the expression of the product of state-cycles for the state permutation induced by π. Let $MPW(\pi)$ be the maximum period of wire-cycles for π, and $MPW(\pi)$ be the maximum period of state-cycles induced by π. Then $W(\pi)$ is a sequence consisting of $\theta_\pi(i)$ i's in non-descending order, where i = 1, 2, . . . ,$MPW(\pi)$. $S_p(\pi)$ is a sequence consisting of $\overline{\theta}_\pi(i)$ i's in non-descending order, where i = 1, 2,. . .,$MPS(\pi)$. Note that subscript p of S_p denotes that signals are from J_p.

Example 1. Let π be (1, 2)(3). Then $W(\pi)$ = (1, 2) and $S_2(\pi)$ = (1, 1, 1, 1, 2, 2).

A modulo-p linear sequential network is composed of arbitrary interconnections of three kind of elements (1) unit delay elements (2) modulo-p adders and (3) modulo-p scalar multipliers, where the delay elements are separated out from the combinational logic. Any wire in such a network is capable of being, at each instance of time, in any one of p states represented by the digits 0, 1, 2, . . . ,p-1. A cyclic state generator constructed from PERMU(N) can be considered a special type of autonomous linear sequential networks such that neither modulo-p adders nor modulo-p multipliers are used in the network. Therefore, some properties concerned with cyclic state generators can be derived using matrix algebra as various authors have studied for linear sequential networks (Booth[3], Elspas[5], Zieler[8]). Although the matrix algebraic approach is valid, it is usually inefficient for our purpose. Since the structure of the cyclic state generators is very simple, it may be better to devise an individual method for solving each problem

concerning the cyclic state generators. We shall give results without their proofs. The reader may find the proofs in [4].

We define Γ_N (N = 1, 2, . . .) as follows: $\Gamma_N = \{(a_1, \ldots, a_r) \mid$ r is an integer, a_i (i = 1, . . .,r) is an positive integer, $\sum_{i=1}^{r} a_i = N$, and for each i ($1 \leq i \leq r-1$) $a_i \leq a_{i-1}\}$. From the definition of $W(\pi)$ the next proposition is immediate.

Proposition 1. For any N ($N \geq 1$) $\Gamma_N = \{W(\pi) \mid \pi$ is a permutation on $\{1, \ldots, N\}\}$.

Proposition 2. Let π be a permutation on $\{1, \ldots, N\}$ expressed as the product of wire-cycles $\alpha_1, \ldots, \alpha_k$, where the period of α_i is w_i ($1 \leq i \leq k$). Then the maximum period of state-cycles induced by π on PERMU(N) is LCM(w_1, . . . , w_k), where LCM($w_1, \ldots, w_k$) means the least common multiplier of $w_1, \ldots, w_k$.

The next proposition is immediate from Proposition 2.

Proposition 3. $\{S_p(\pi) \mid \pi$ is a permutation on $\{1, \ldots, N\}\}$ is a proper subset of Γ_t, where $t = p^N$ and $N \geq 1$.

We now define equivalence relations on permutations induced by wire-sequences and state-sequences respectively as follows:

(1) π and π' are equivalent under wire-sequences if and only if $W(\pi) = W(\pi')$.

(2) π and π' are equivalent under state-sequences if and only if $S_p(\pi) = S_p(\pi')$.

We conjecture that the equivalence relation defined by (1) above is the same as the equivalence relation defined by (2) above. At present we can only show that the equivalence relation induced by wire-sequences is a refinement of the equivalence relation induced by state-sequences. That is , we have the next theorem.

Theorem 1. Let π and π' be permutations on $\{1, \ldots, N\}$ respectively. If $W(\pi) = W(\pi')$, then $S_p(\pi) = S_p(\pi')$.

We do not know at present whether there exists a pair of permutations π and π' such that $W(\pi) \neq W(\pi')$ but $S(\pi) = S(\pi')$. We shall give a necessary condition that for a pair of permutations π and π' on $\{1, \ldots, N\}$ $S_p(\pi)$ is equal to $S_p(\pi')$.

Let α and β be length-N sequences of integers from J_p respectively. If α is obtained from β by applying an appropriate number of cyclic shifts, we say that α and β are cyclic equivalent. Let $C_p(N)$ be the set of length-N sequences of integers from J_p such that their cycle period is N, and let $\overline{C_p}(N)$ be the set of cyclic equivalence classes of $C_p(N)$. $\#\overline{C_p}(N)$ denotes the cardinality of $\overline{C_p}(N)$.

Example 2. $\overline{C_2}(1) = \{0, 1\}$, $\overline{C_2}(2) = \{10\}$, $\overline{C_2}(3) = \{100, 110\}$, $\overline{C_2}(4) = \{1000, 1100, 1110\}$, $\overline{C_2}(5) = \{10000, 11000, 10100, 11100, 11010, 11110\}$, where each equivalence class is expressed by its representative member.

Theorem 2. Let $W(\pi)$ be a wire-sequence consisting of m components, and let the number of multipliers of q in $W(\pi)$ be r, where q is a prime integer. Then the number of q in $S_p(\pi)$ is $p^{m-r}((q\ \#\overline{C_p}(q) + p)^r - p^r)/q$.

Proposition 4. Let $W(\pi) = (w_1, \ldots, w_t)$ and $W(\pi') = (w_1', \ldots, w_u')$. If $S_p(\pi) = S_p(\pi')$, the following two conditions are satisfied:

(1) $t = u$ (i.e., the number of components in $W(\pi)$ is equal to the number of components in $W(\pi')$).

(2) For any prime integer q, the number of multipliers of q in $W(\pi)$ is equal to the number of multipliers of q in $W(\pi')$.

Unfortunately, the two conditions given in Proposition 4 are not sufficient to be $W(\pi) = W(\pi')$ for a pair of π and π' such that $S_p(\pi) = S_p(\pi')$. The next example shows this fact.

Example 3. Let π and π' be a pair of permutations on {1, . . .,30} such that $W(\pi) = (5,5,6,14)$ and $W(\pi') = (3,7,10,10)$. These two wire-sequences satisfy the two conditions in Proposition 4. However, the number of 6's in $S_2(\pi)$ is 152 whereas the number of 6's in $S_2(\pi')$ is 24 as shown in Table 1.

i	Number of i's in $S_2(\pi)$	Number of i's in $S_2(\pi')$
1	16	16
2	24	24
3	16	16
4	0	0
5	816	816
6	152	24
7	144	144
.	.	.
.	.	.
.	.	.

Table 1. $S_2(\pi)$ and $S_2(\pi')$, where $W(\pi) = (5,5,6,14)$ and $W(\pi') = (3,7,10,10)$

We now describe an algorithm for computing $\#\overline{C}_p(k)$ for k = 1, 2, . . . Suppose that a wire-sequence consists of a single component k. Then we say its corresponding state-sequence to be "simple" and denote it by $SEQ_p(k)$.

Let π be a permutation such that $W(\pi) = (k)$. Then for a factor f of k a state-cycle of period f for π is obtained by the following way: Choose a representative element, say α in p-nary form, of an equivalence class of $\overline{C}_p(f)$. The k/f times repetition of α is allocated to input terminals in the way that the i-th digit of the k/f times repetition of α is given to the input terminal appearing at the i-th position in the wire-cycle for π. Starting from this initial state a sequence of states appearing sequentially on the cyclic state generator with π interconnection is a state-cycle of period f for π. Any state-cycle of period f for π can be obtained in this

way. For an integer j which is not a factor of k there does not exist a state-cycle of period j for π. We therefore have the next proposition.

Proposition 5. Let $f_1, f_2, \ldots, f_t$ be all the factors of k. Then $SEQ_p(k)$ consists of $\#\overline{C}_p(f_i)$ f_i's (i = 1, 2, . . . ,t).

Example 4. $SEQ_2(1) = (1, 1)$, $SEQ_2(2) = (1, 1, 2)$, $SEQ_2(3) = (1, 1, 3, 3)$, $SEQ_2(4) = (1, 1, 2, 4, 4, 4)$, $SEQ_2(5) = (1, 1, 5, 5, 5, 5, 5, 5)$, $SEQ_2(6) = (1, 1, 2, 3, 3, 6, 6, 6, 6, 6, 6, 6, 6, 6)$.

Since the number of states which are expressed as k digits in p-nary form is p^k, the next theorem is immediate from Proposition 5.

Theorem 3. $\sum_{f \mid k} f \#\overline{C}_p(f) = p^k$ for k = 1, 2, . . . ,

where $\sum_{f \mid k}$ means the summation of terms for all f which can divide k without remainder.

From the equations given in Theorem 3, we can easily compute $\#\overline{C}_p(k)$ for k = 1, 2, . . ., Möbius function $\mu(d, k)$ is defined by (see Berge[1])

$$\mu(d, k) = \begin{cases} 1 & \text{if } k = d, \\ (-1)^t & \text{if } k = p_1 p_2 \ldots p_t d, \text{ where the } p_i \ (i=1, \ldots, t) \text{ are distinct primes,} \\ 0 & \text{other case.} \end{cases}$$

Then from Möbius inversion formula (Proposition 2 in Rota[7]) we have the next theorem. A computer program for computing Möbius function is given in (Nijehuis and Wilf[6]).

Theorem 4. $\#\overline{C}_p(k) = \left(\sum_{d \mid k} p^d \mu(d, k) \right)/k$ for any $k \geq 1$.

Let I_p be the number of irreducible polinomials of degree k over modulo-p field. Then we have the following equations (Elspas[4]): $\sum_{j \mid k} j I_p(j) = p^k$. Therefore, we have the next theorem.

Theorem 5. For each j (j = 1, 2, . . .) $\#\overline{C}_p(j)$ is equal to the number of irreducible polinomials of degree j over modulo-p field.

3. Construction of State-Sequences

In this section we discuss the following problem: "For a given wire-sequence, construct the state-sequence that corresponds to the wire-sequence." This problem is mathematically trivial, and can be solved by the following method: "Choose a permutation π such that $W(\pi)$ is equal to the given wire-sequence. Then we construct all possible state-cycles for π by state-by-state evaluation of the corresponding state diagram or the cyclic state generator with π interconnection." However, this näive method is obiously laborious. Although for any prime integer q, the number of q's in $S_p(\pi)$ can be immediately evaluated by the formula given in Theorem 2, it seems to be difficult to derive a general formula for evaluating the number of state-cycles of an given period. We shall describe an efficient method for solving this problem.

We define the product of state-sequences. Let $\alpha = (a_1, \ldots, a_r)$ and $\beta = (b_1, \ldots, b_t)$ be state-sequences that are not necessarily simple. The product of α and β is denoted by $\alpha \times \beta$ which is defined as

$$\mathrm{SORT}((a_1 \times b_1), (a_1 \times b_2), \ldots, (a_1 \times b_t), (a_2 \times b_1), \ldots, (a_2 \times b_t),, \ldots, (a_r \times b_1), \ldots, (a_r \times b_t)),$$

where for a pair of positive integers A and B $(A \times B)$ is a sequence of $A \cdot B/\mathrm{LCM}(A, B)$ times repetition of $\mathrm{LCM}(A, B)$ and $\mathrm{SORT}(d_1, \ldots, d_m)$ is the sorted sequence of $d_1, \ldots, d_m$ in non-descending order.

Example 5. $(1 \times 1) = (1)$, $(1 \times 2) = (2)$, $(2 \times 3) = (6)$, $(4 \times 6) = (12, 12)$, $(10 \times 15) = (30, 30, 30, 30, 30)$, $\mathrm{SEQ}_2(1) \times \mathrm{SEQ}_2(2) = (1, 1) \times (1, 1, 2) = (1, 1, 1, 1, 2, 2)$

It is obvious that the product of state-sequences is commutative and associateive (i.e., $\alpha \times \beta = \beta \times \alpha$ and $\alpha \times (\beta \times \gamma) = (\alpha \times \beta) \times \gamma$).

Theorem 6. If $W(\pi) = (w_1, w_2, \ldots, w_r)$, then $S_p(\pi) = \mathrm{SEQ}_p(w_1) \times \mathrm{SEQ}_p(w_2) \times \ldots \times \mathrm{SEQ}_p(w_r)$.

Example 6. Let $\pi_1 = (1, 2, 3)(4)$ and $\pi_2 = (1, 2)(3, 4)$. Then $W(\pi_1) = (1, 3)$ and $W(\pi_2) = (2, 2)$. From Theorem 6, $S_2(\pi_1) = SEQ_2(1) \times SEQ_2(3) = (1, 1) \times (1, 1, 3, 3) = (1, 1, 1, 1, 3, 3, 3, 3)$ and $S_2(\pi_2) = SEQ_2(2) \times SEQ_2(2) = (1, 1, 2) \times (1, 1, 2) = (1, 1, 1, 1, 2, 2, 2, 2, 2, 2)$.

References

(1) C. Berge, "Principle of Combinatorics", Academic Press, New York (1971).

(2) G. Birkhoff and S. Maclane, "A Survey of Modern Algebra", Macmillan, New York (1953).

(3) T. L. Booth, "Sequential Machines and Automata Theory", John Wiley and Sons, New York (1967).

(4) F. M. Brown and Y. Igarashi, "On permutations of wires and states", Technical Report CS-82-1, Gunma University (1982).

(5) B. Elspas, "The theory of autonomous linear sequential networks", IRE Trans. on Circuit Theory, CT-6, pp. 44-60 (1959).

(6) A. Nijenhuis and H. S. Wilf, "Combinatorial Algorithms", Academic Press, New York (1978).

(7) G. Rota, "On the foundations of combinatorial theory I. theory of Möbius functions", Z. Wahrscheinlichkeitstheorie, 2, pp. 340-368 (1964).

(8) N. Zierler, "Linear recurring sequences", J. Soc. Indust. Appl. Math. 3, pp. 31-48 (1959).