A Multistage Construction for Substitution Networks

Christian RONSE
Philips Research Laboratory Brussels

Author's address:
Philips Research Laboratory Brussels
2 Av. Van Becelaere,
B-1170 Bruxelles
BELGIUM

Abstract : *Substitution networks are switching networks which realize one to one mappings from their input alphabet to their output alphabet. They are used for example in cryptology for the enciphering of the cleartext and the deciphering of the ciphertext. We describe here a multistage construction for the design of larger substitution networks from smaller ones, in such a way that this construction allows the realization of any input-output bijection.*

Computer and Control Abstracts classification:
42.30 switching theory; 51.20 logic and switching circuits; 52.10 logic design.
Electrical and Electronics Abstracts classification:
61.50 communication switching theory.

Running head: *Multistage substitution networks*

Symbols:

Δ : capital letter Delta.

Λ : capital letter Lambda.

Σ : capital letter Sigma.

$\mathcal{S}$: script S.

$\mathcal{C}$: script C.

Number of Figures: 6

Figure captions:

Figure 1: *A substitution network on* Σ.

Figure 2: *A* Σ *to* Σ' *CULM*.

Figure 3: *A total substitution network on* $\Sigma \times \Sigma'$.

Figure 4: *The Clos Network*

Figure 5: *A total substitution network on* $\Sigma_1 \times \ldots \times \Sigma_n$ *(n=3).*

No caption on Figures 6.

1. Introduction.

Substitution networks are a basic tool in cryptology. Indeed they realize bijections between an input and an output written in a given alphabet; they are thus used for the enciphering of the cleartext and the deciphering of the ciphertext [4].

Generally the alphabet can be expressed as a set of binary n-tuples. This poses the important problem of the design of such networks with logical gates, and in particular the design of networks which can realize all possible bijections on that alphabet, what we call *total substitution networks.*

Due to the relative complexity of straightforward designs for large substitution networks and to the actual limitation of the available chip sizes, one must often build them from smaller components. In particular it may be necessary to build large substitution networks by interconnecting smaller ones. This is what we call *expansion.*

In this paper we describe a three-stage design which builds from total substitution networks on the alphabets Σ and Σ' (of respective sizes N and N') and from *Canonical Universal Logical Modules (CULM)* (i.e., switching networks that can realize any input-output function), a total substitution network on the alphabet $\Sigma \times \Sigma'$ (of size $N \cdot N'$). This design is obtained by a correspondence with the well-known network of Clos which is used to build permutation networks (see [1]). By iteration, we can thus build a total substitution network on the alphabet Σ^n (the set of n-tuples in Σ) with a $(2n-1)$-stage construction. Given that a total substitution network on the binary alphabet $\{0,1\}$ is a modulo two adder, and that the corresponding CULM is a multiplexer, total substitution networks on the alphabet $\{0,1\}^n$ can be constructed with $(2n-1)$ modulo 2 adders and $(2n-1)$ multiplexers of size 2^{n-1}.

2. Definitions.

Write $\{x_0, \ldots, x_n\}$ for the set containing $x_0, \ldots, x_n$. Given m sets $A_1, \ldots, A_M$ ($m \geq 2$), write $A_1 \times \cdots \times A_m$ for the set of M-tuples $(a_1, \ldots, a_m)$ with a_i in A_i for $i = 1, \ldots, m$. In particular write A^m for the product $A \times \ldots \times A$ of m copies of A, that is the set of m-tuples in A.

Let Σ be an alphabet of size N. Then a *substitution network* on Σ is a switching network having an N-valued input I (with values in Σ), an N-valued output O (with

values in Σ) and an M-valued control input C. For each value of the control input C, the network realizes a bijection between the values of the input I and those of the output O, in other words a permutation of the alphabet Σ. Every such bijection is called a *substitution*. We illustrate that network in Figure 1 and write it S_N.

In practice, one often has $N = p^k$ and $\Sigma = \Lambda^k$ for an alphabet Λ of size p, and the input and output can be seen as sets of k p-valued inputs and outputs respectively. One generally takes $p = 2$. Thus in a concrete design, one can consider that we have $log_2(N)$ binary inputs and outputs.

If the substitution network S_N can realize all possible $N!$ substitutions, then we say that S_N is *total*. In this case it is necessary that $M \geq N!$.

Given two alphabets Σ and Σ' (of respective sizes N and N'), a Σ to Σ' *Canonical Universal Logical Module* (in brief, a *CULM*) [3] is a switching network having an input I with values in Σ, an output O with values in Σ' and an M-valued control input C such that every mapping from Σ to Σ' can be realized as input-output function of that network under an appropriate setting of the value of C. Note that we must have $M \geq N'^N$. We illustrate that network in Figure 2 and write it $C_{N,N'}$.

3. The three-stage expansion of total substitution networks.

We are now ready to attack our problem. The expansion of total substitution networks means the construction of larger total substitution networks from smaller ones. We will describe here a three-stage construction of a total substitution network on the alphabet $\Sigma \times \Sigma'$. The design is organized as follows:

Let Λ and Λ' be the respective alphabets of the control inputs of Σ and Σ'. Then we obtain the following three stages:

— The first stage contains a total substitution network on Σ and a Σ' to Λ CULM.

— The second stage contains a total substitution network on Σ' and a Σ to Λ' CULM.

— The third stage is similar to the first one.

This construction is illustrated in Figure 3.

We will now explain how this construction has been found and why that substitution network is total. This will be done by an analogy with permutation networks.

A *permutation network* on n lines is a switching circuit with n inputs and n outputs, which can realize any one to one connection between the inputs and the outputs. This type of network is used for example in telephony (see [1]). If each line traversing that network carries signals in a p-valued logic, then that permutation network can be seen as a substitution network on an alphabet of size p^n.

However, it is not in this way that we will make our analogy, but rather as follows: A permutation network can realize all permutations on the set of input lines, while a total substitution network can realize all permutations of the input alphabet. We will thus relate the input lines of a permutation network to the values of the input of a total substitution network. Given a design for the construction of a permutation network (for example Clos's network), it can be expressed mathematically as a law for the decomposition of permutations, and this law can be translated in terms of the alphabet as a design for total substitution networks.

Consider indeed Clos's network. It is a three-stage design for constructing a permutation network on $a \cdot b$ lines with $2a$ copies of a permutation network B on b lines and b copies of a permutation network A on a lines. We illustrate it on Figure 4 for $a = 3$ and $b = 2$, and we number the lines with elements of $Z_b \times Z_a$, where $Z_a = \{0, ..., a-1\}$ and $Z_b = \{0, ..., b-1\}$. It is well-known that this construction forms a permutation network (see [1]).

Now this fact means that a permutation of $Z_b \times Z_a$ can be decomposed as the product of three permutations, corresponding to the three stages of the network:

— The first permutation maps every (x, y) of $Z_b \times Z_a$ on some $(\pi_y(x), y)$, where π_y is a permutation of Z_b corresponding to y.

— The second permutation maps every (x, y) of $Z_b \times Z_a$ on some $(x, \rho_x(y))$, where ρ_x is a permutation of Z_a corresponding to x.

— The third permutation is of the same type as the first one.

Now if we replace Z_b and Z_a by Σ and Σ' and line permutations by substitutions, then we obtain the construction of Figure 3. Indeed, a substitution π_y on

Σ determined by some y in Σ' can be obtained by making y "act" on the control input of a total substitution network on Σ. But as the correspondence between the values of the control input and the resulting substitution is not predefined (and can be arbitrary), we need to use a Σ' to Λ CULM to realize the action of y on the control input of that total substitution network on Σ. This explains thus the first stage of the design of Figure 3. The two remaining stages can be understood similarly.

Let us say a few words on the control of CULM's in our construction. A detailed analysis of their structure can be found in [3]. We can however state a few elementary facts. If the alphabets Σ and Σ' have respective size σ and σ', then the control of a Σ to Σ' CULM must have size σ'^{σ}, and so we can choose the alphabet Σ'^{σ} for the control; given a bijection $\tau : \Sigma \to \{0, \ldots, \sigma - 1\}$, we associate to a function $\phi : \Sigma \to \Sigma'$ realized by the CULM the control input $\left(\phi(\tau^{-1}(0)), \ldots, \phi(\tau^{-1}(\sigma - 1))\right)$.

4. Some applications.

Since Beneš [1], it is customary to generate permutation networks on n^k lines with $2n-1$ stages of n^{k-1} permutation networks on n lines, by an iterative application of Clos's three-stage decomposition. Using our construction of Section 3 (from a preliminary draft of this paper), M. Davio [2] did a similar thing for total substitution networks. We show in Figure 5 (for $n = 3$) his design for a total substitution network on the alphabet $\Sigma_1 \times \ldots \times \Sigma_n$ built with $2n - 1$ stages using each one CULM and one total substitution network. Note the regular interconnection pattern between the stages: the first $n - 1$ interconnections consist in a rotation $(\Sigma_n, \ldots, \Sigma_1)$ on the n alphabets, and the remaining $n - 1$ interconnections form the inverse rotation $(\Sigma_1, \ldots, \Sigma_n)$.

In the same way as one builds permutation networks on 2^n lines from binary cells with a $(2n - 1)$-stage construction [1)5], one can build with total substitution networks on the binary alphabet $\Delta = \{0, 1\}$ a total substitution network on the alphabet Δ^n. In fact, a total substitution network on Δ is a binary adder . Moreover, the corresponding Δ^{n-1} to Δ CULM is simply a multiplexer of size 2^{n-1}. The form taken by every stage (without taking into account the interconnection with the preceding and following stages) of the resulting total substitution network is shown in Figure 6.

One further interest in using the basis 2 for our construction is a property of the corresponding permutation network of Beneš and Waksman, that there is a simple control algorithm for the construction of any permutation, called "looping" [5]. In fact, the looping algorithm can be translated to that type of total substitution networks. This is done in an example in [2]. In this reference, other properties of the multistage design for total substitution networks are analysed (for example asymptotical cost and delay, comparison with other designs, etc.).

5. Conclusion.

We have presented here a multistage design for the construction of large total substitution networks. In particular, it is possible to build a total substitution network on the alphabet $\{0,1\}^n$ with $2n-1$ stages containing each an adder modulo 2 and a multiplexer of size 2^{n-1}, and the "looping algorithm" of Waksman [5] can be used as control algorithm of such a network.

REFERENCES:

[1] V.E. BENEŠ, *Mathematical theory of switching networks and telephone traffic*, Academic Press, New York (1965).

[2] M. DAVIO, "Substitution networks", *Proceedings of the summer school on security of transmissions (C.I.S.M.)*, Springer Verlag *(to appear)* (1982).

[3] M. DAVIO, J.P. DESCHAMPS, A. THAYSE, *Discrete and switching functions*, McGraw-Hill (1978).

[4] J. B. KAM, G. I. DAVIDA, "Structured Design of Substitution-Permutation Encryption Networks", *IEEE Trans. on Computers*, **Vol. C-28**, *n° 10*, (1979).

[5] A. WAKSMAN, "A permutation network", *J. ACM* **15**, 159–163, 340 (1968).

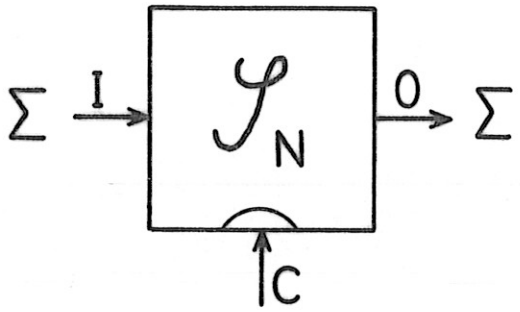


FIG. 1 A substitution network on Σ.

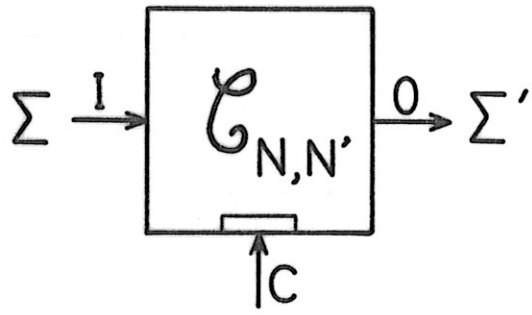


FIG. 2 A Σ to Σ′ CULM.

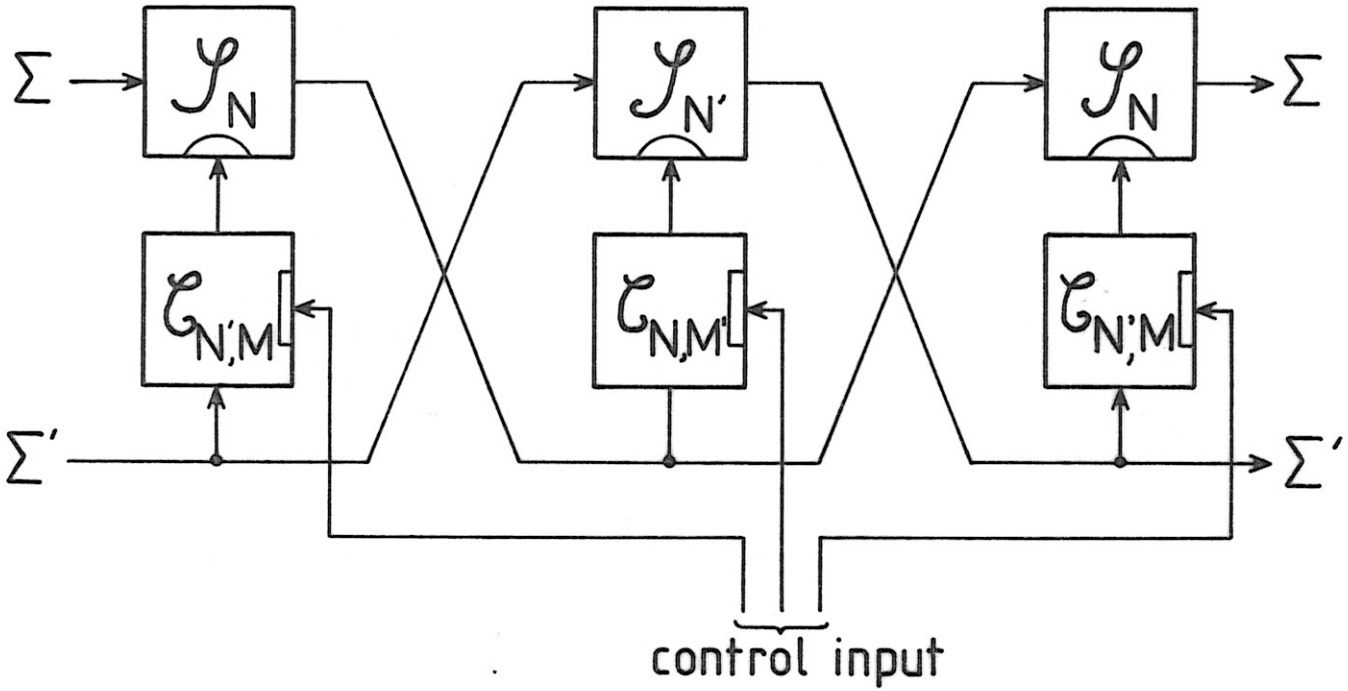


FIG. 3 A total substitution network on $\Sigma \times \Sigma'$.

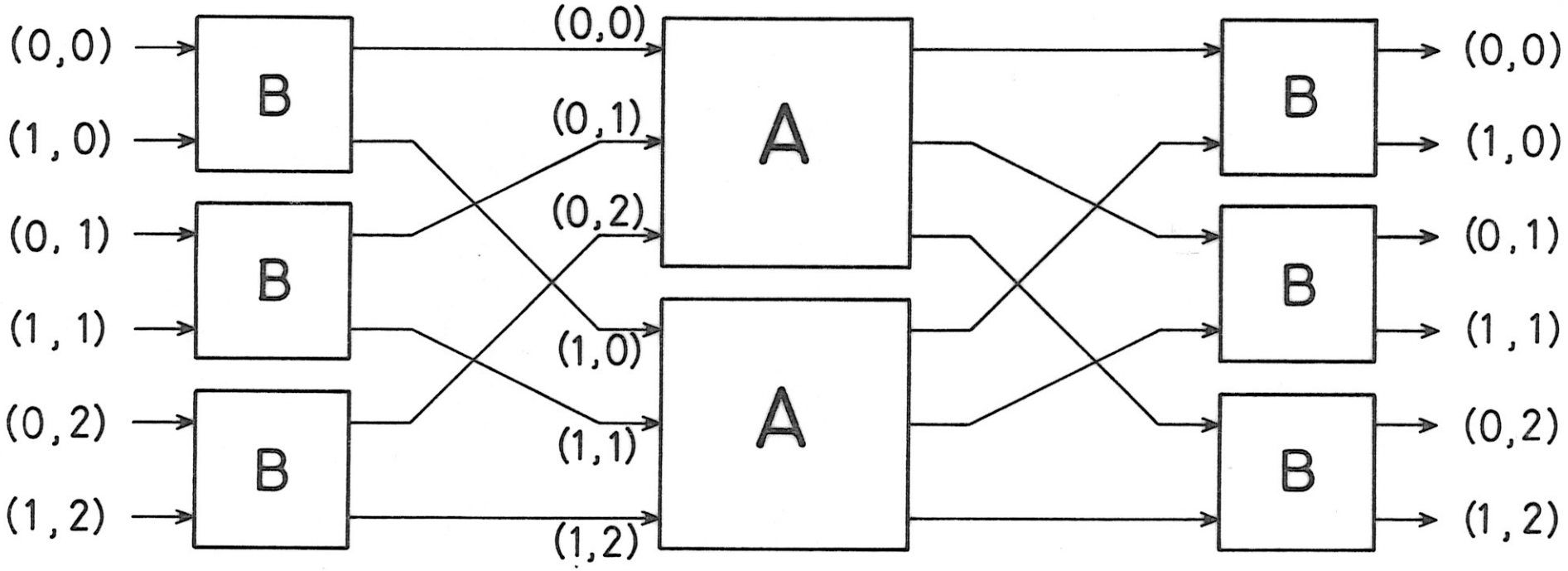
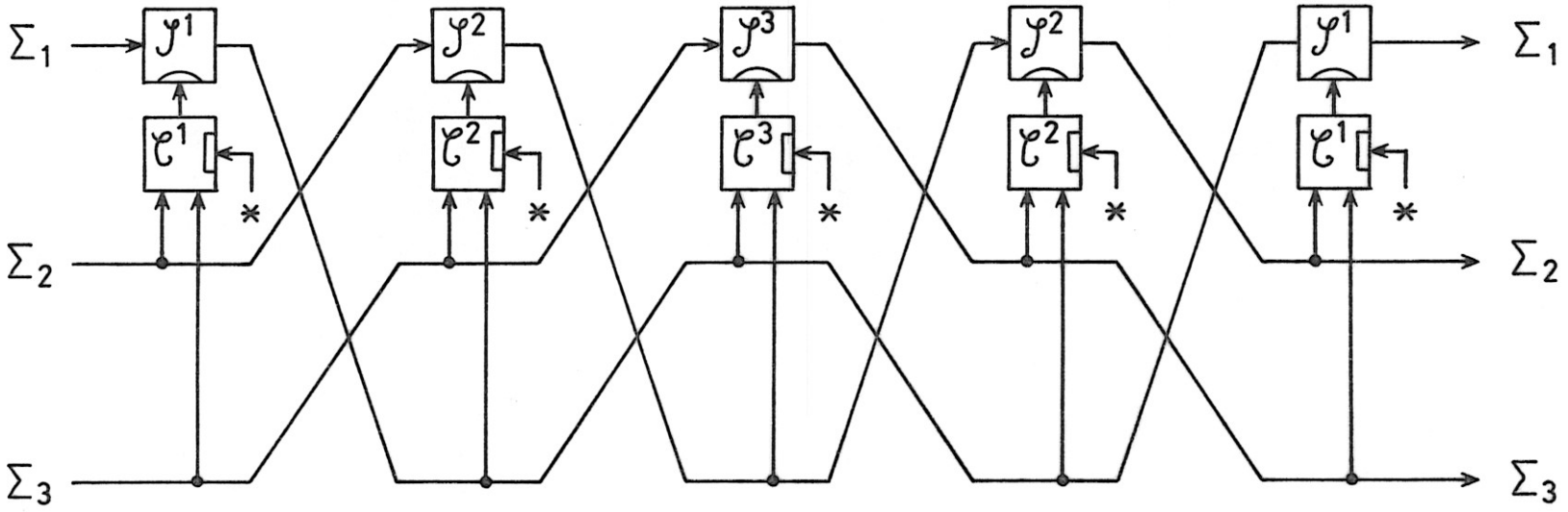


FIG. 4 The Clos Network.



* = control inputs

FIG.5 A total substitution network on $\Sigma_1 \times \ldots \times \Sigma_n$ (n=3)

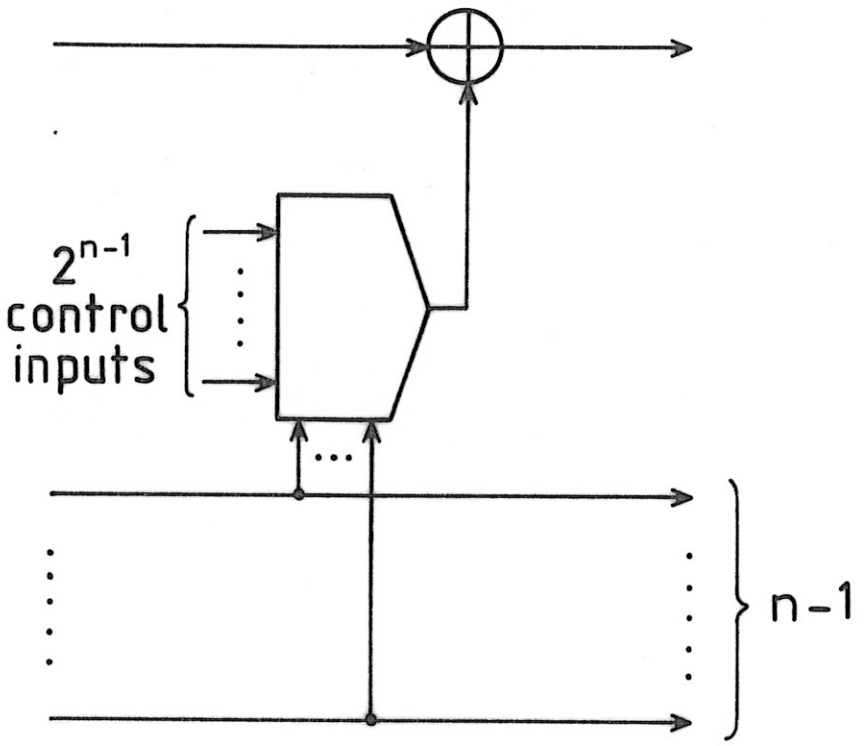


FIG. 6