

# THE DESIGN AND CONTROL OF DYNAMIC MEMORY NETWORKS

M.A. Fiol, J. Fàbrega and J.L.A. Yebra

Dep. de Matemàtica Aplicada i Telemàtica  
Universitat Politècnica de Catalunya  
08034 Barcelona, SPAIN

## ABSTRACT

A dynamic memory is a storage medium constituted by an array of cells and an interconnection network between them. It is characterized by the constant circulation of the stored data. The concern is to design the interconnection network in order to have small access times and a simple control of the memory. Several interconnection schemes have been proposed in the literature.

This paper presents a quite general model for such structures that greatly facilitates both the design and the control of the memory. It is shown that most known proposals of dynamic memory interconnection networks are particular instances of the model and that it leads to some new ones.

## 1. INTRODUCTION

Bulk memories are traditionally implemented as magnetic disks and drums, where data are stored at fixed locations on a magnetic surface that moves respect to a read/write mechanism. The use of dynamic storage technologies—such as magnetic bubble or CCD memories—has been proposed to allow a sharp reduction of the access times. The reason is that, instead of a mechanical movement, dynamic memories require the movement of the data within the storage medium itself. This already happens in MOS shift registers, where data are cyclically displaced at unit-time intervals until they reach a read/write port. But with such a structure it is impossible to obtain reduced values of the access time for large memories. Stone [13] was the first to propose a more general model for dynamic memories. Since then there have been many different proposals: Aho and Ullman [1], Kluge [9], Lenfant [10], Morris, Valiere III and Wisniewski [11], Wong and Tang [15], Iyer and Sinclair [7] and the authors [3], [4]. Besides, several patents have been registered [2], [5] and [6]. Most of these organizations are compared in the tutorial of Iyer and Sinclair [8].

A dynamic memory is intended for a set of fixed size data items. The memory consists of cells, each storing one datum, and an interconnection network between them. The network allows the movement of the data from a cell to a neighbouring one at unit-time intervals. The result

This work has been supported by the "Comisión Asesora para la Investigación Científica y Técnica" (CAICYT) under projects 1180-84 and PA86-0173.

of all the data transfers during such an interval is a permutation of the contents of the memory. Only a small number of these permutations are available. They are called *memory transformations*. Therefore, in a dynamic memory we must distinguish between the *physical address* of a datum, which is its current physical location, and its *logical address* which can be thought of as its initial location.

To access a requested datum, it must be sent to a specific cell, called the *read/write (r/w) cell*, where data can be read or written. To this end the following steps must be carried out:

(1) to find its physical address from its logical address and some additional information about the memory transformations that have been applied to the memory;

(2) to determine the sequence of memory transformations that route it to the r/w cell.

The *control problem* in a dynamic memory consists of this two steps and its study is the main objective of this paper.

In many instances, access to a block of contiguous data items is required. Aho and Ullman propose in [1] the use of a pair of memory transformations for a memory of  $N=2^k-1$  cells that allows *sequential access* to a block of data. The blocks can be as large as the size of the memory, and the corresponding sequence of memory transformations is then called a *tour*. The existence of tours for dynamic memories of size  $N=2^k$  has been studied by Stone in [14] and by Morris, Valiere III and Wisniewski in [11].

The plan of the paper is as follows. In Section 2 we present the mathematical formulation of the problem together with the mathematical background required through the paper. Section 3 contains the main theoretical results. Finally, the aim of Section 4 is twofold: firstly, to show how the theory developed in Section 3 applies to most known proposals of dynamic memory organizations, and secondly to illustrate its power in the designing of new ones.

## 2. MATHEMATICAL MODEL

In our formulation, the dynamic memory network is modelled by a directed graph, called a digraph for short, in which the vertices represent the storing cells and the arcs the links between them.

A digraph  $D=(V,A)$  consists of a set  $V=V(D)$  of points called *vertices* and a set  $A=A(D)$  of arcs or directed lines between vertices. The number of vertices  $N=|V|$  is the order of the digraph. If  $[x,y]$  is an arc from a vertex  $x$  to a vertex  $y$ , it

is said that  $x$  is adjacent to  $y$  and also that  $y$  is adjacent from  $x$ . We allow  $D$  to have loops, that is  $[x,x]$  arcs, but not parallel arcs, so that there is at most one arc from each vertex to any other. The sets of vertices adjacent to and from a vertex  $x$  are denoted by  $\Gamma^-(x)$  and  $\Gamma^+(x)$  respectively, and their cardinalities  $\delta^-(x)=|\Gamma^-(x)|$  and  $\delta^+(x)=|\Gamma^+(x)|$  are the in- and out-degree of  $x$ . A digraph is regular of degree  $\delta$  or  $\delta$ -regular if the in- and out-degree of all vertices equals  $\delta$ . For instance, the complete symmetric digraph on  $\delta$  vertices  $K_\delta^+$ , in which each vertex is adjacent to any other one including itself, is  $\delta$ -regular. See  $K_3^+$  in Fig. 1.

Only strongly connected digraphs are considered here; this means that for any pair of vertices  $x, y$  there is a (directed) path from  $x$  to  $y$ . The number of arcs of a shortest path from a vertex  $x$  to another vertex  $y$  is called the distance  $d(x,y)$  from  $x$  to  $y$ . Finally, we recall that in a  $\delta$ -regular digraph ( $\delta > 1$ ) there are at most

$$1 + \delta + \delta^2 + \dots + \delta^k = (\delta^{k+1} - 1) / (\delta - 1) \quad (1)$$

vertices at distance at most  $k$  of a given one.

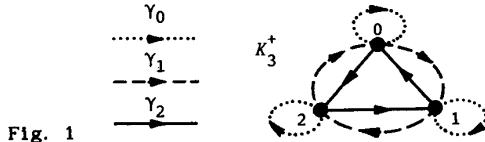


Fig. 1

The distance from one vertex to another measures the access time required to send a datum between the corresponding cells of the memory. The networks to be considered correspond to regular digraphs of degree  $\delta$  and the memory transformations to a decomposition into permutations of the digraph in the sense defined next.

A decomposition into permutations of a  $\delta$ -regular digraph  $D=(V,A)$  is a set  $\{\gamma_i; 0 \leq i \leq \delta-1\}$  of permutations of the vertices of  $V$  that satisfies

$$(a) \gamma_i(x) \in \Gamma^+(x) \quad (2)$$

$$(b) \gamma_i(x) \neq \gamma_j(x) \text{ for } i \neq j. \quad (3)$$

It is easily shown that any  $\delta$ -regular digraph can be decomposed (usually in several ways) into permutations. For instance, if we identify the set of vertices of  $K_\delta^+$  with the set of integers modulo  $\delta$ ,  $V(K_\delta^+) = \mathbb{Z}_\delta$ , the permutations defined by

$$\gamma_i(x) = x+i, \quad 0 \leq i \leq \delta-1 \quad (4)$$

form a decomposition of  $K_\delta^+$ . Fig. 1 shows this decomposition of  $K_3^+$ . See [3] for more details. A decomposition associates a permutation to every arc of  $D$ :

$$[x,y] \mapsto \gamma_i \text{ if } \gamma_i(x) = y$$

which is actually an (arc-)coloring of  $D$ , since different permutations are associated to the  $\delta$  arcs to and from any vertex.

As it is seen in the next section, a useful way to obtain a digraph decomposed into permutations consists in identifying its vertex set  $V$  with the set of elements of a group  $G$ . Besides the standard notions, the following concepts from group theory are required.

If  $G$  is a finite group generated by  $\Delta = \{a_0, a_1, \dots, a_{\delta-1}\}$ , its Cayley diagram with respect to  $\Delta$ ,  $D_\Delta(G)$ , is the  $\delta$ -regular digraph whose vertices represent the elements of  $G$  and where there is an arc  $[g,h]$  labelled (or colored)  $a_i$  iff  $h = a_i g$  for some  $a_i \in \Delta$ .

Given two groups  $G$  and  $H$  together with an homomorphism of  $H$  into the set of automorphisms of  $G$ ,  $\Pi: H \rightarrow \text{Aut } G$ ,  $\Pi(h) = \pi_h$ , the (external) semidirect product  $G \rtimes H$  is the group with set of elements  $\{(g,h); g \in G, h \in H\}$  and composition rule

$$(g_1, h_1) * (g_2, h_2) = (g_1 \pi_{h_1}(g_2), h_1 h_2). \quad (5)$$

It coincides with the direct product  $G \times H$  iff  $\Pi = 0$ , that is, when  $\Pi(h)$  is always the identity  $\iota$ . In this paper  $H$  will already be a subgroup of  $\text{Aut } G$ , so that, with  $\Pi$  the canonical embedding, (5) becomes

$$(g_1, \pi_1) * (g_2, \pi_2) = (g_1 \pi_1(g_2), \pi_1 \pi_2). \quad (5')$$

A permutation group  $\Sigma$  on a set  $V$  is called transitive if, given any pair of elements  $x, y \in V$ , there exists a permutation  $\sigma \in \Sigma$  such that  $\sigma(x) = y$ . The permutations of  $\Sigma$  that fix a given element  $x \in V$  form a subgroup  $\text{St}_\Sigma(x)$ , called the stabilizer of  $x$  in  $\Sigma$ . If  $\Sigma$  is transitive its order is given by  $|\Sigma| = |V| |\text{St}_\Sigma(x)|$  for any  $x \in V$ .

### 3. CONTROL OF THE MEMORY

The number of different states of a dynamic memory is the order of the group generated by the memory transformations. We begin this section by characterizing this group.

Given a set  $V$  together with a set  $\{\gamma_i; 0 \leq i \leq \delta-1\}$  of permutations of  $V$  that satisfy (3), we can consider the  $\delta$ -regular digraph  $(V, \{\gamma_i\})$  that has  $V$  as set of vertices and where, in view of (2), each vertex  $x$  is adjacent to the vertices  $\gamma_i(x)$ ,  $0 \leq i \leq \delta-1$ . A useful way to exploit this idea consists in considering  $V$  as the set of elements of a group  $G$ , that is, each vertex of  $V$  stands for an element of  $G$ . Then we have the following natural ways of defining permutations in  $V$ :

- (a) multiplication by an element  $g$  of  $G$ :  $\gamma(x) = gx$ ;
- (b) by means of an automorphism  $\pi$  of  $G$ :  $\gamma(x) = \pi(x)$ .

Of course, we can also combine them. Thus, given  $g_i \in G$  and  $\pi_i \in \text{Aut } G$  for  $0 \leq i \leq \delta-1$ , we can form the permutations

$$\gamma_i(x) = g_i \pi_i(x) \quad \forall x \in V, 0 \leq i \leq \delta-1 \quad (6)$$

and the digraph  $D=(V, \{\gamma_i\})$  that has the decomposition into permutations  $\{\gamma_i; 0 \leq i \leq \delta-1\}$ . We assume that the  $g_i$  and the  $\pi_i$  have been so chosen that  $D$  is strongly connected.

Let  $\Sigma = \langle \gamma_0, \gamma_1, \dots, \gamma_{\delta-1} \rangle$  be the permutation group generated by  $\gamma_0, \gamma_1, \dots, \gamma_{\delta-1}$ . In order to characterize  $\Sigma$ , let  $H = \langle \pi_0, \pi_1, \dots, \pi_{\delta-1} \rangle$ .

**Theorem 1.** The permutation group  $\Sigma$  is isomorphic to a subgroup of the semidirect product  $G \rtimes H$ .

**Proof.** Since

$$\begin{aligned} \gamma_j(\gamma_j(x)) &= g_j \pi_j(g_j \pi_j(x)) = g_j \pi_j(g_j) \pi_j \pi_j(x) = \\ &= g' \pi'(x) \end{aligned} \quad (7)$$

with  $g' = g_j \pi_j(g_j)$ ,  $\pi' = \pi_j \pi_j$ , every permutation  $\sigma \in \Sigma$  is given by

$$\sigma(x) = g\pi(x) \quad (8)$$

for some  $g \in G$ ,  $\pi \in H$ . Then the injection

$$\Sigma \longrightarrow G \rtimes H, \quad \sigma \longmapsto (g, \pi) \quad (9)$$

is easily seen to be well defined and is a group homomorphism since, from the above calculation, if  $\sigma_1 \longmapsto (g_1, \pi_1)$  and  $\sigma_2 \longmapsto (g_2, \pi_2)$  then

$$\sigma_1 \sigma_2 \longmapsto (g_1 \pi_1(g_2), \pi_1 \pi_2) = (g_1, \pi_1) * (g_2, \pi_2). \quad \blacksquare$$

For convenience we identify  $\Sigma$  and its image  $\Sigma'$  in  $G \rtimes H$ , that is, we identify every permutation  $\sigma \in \Sigma$  with the pair  $(g, \pi) \in G \rtimes H$  given by (8). Since the digraph  $D$  is supposed to be strongly connected, the group  $\Sigma'$  is transitive. It follows that its order satisfies

$$|\Sigma'| = N\ell \quad (10)$$

where  $N = |V| = |G|$  and  $\ell = |\text{St}_{\Sigma'}(x)|$  for any  $x \in G$ . For  $x = e$ , the identity of  $G$ , this subgroup  $L' = \text{St}_{\Sigma'}(e)$

consists of the elements of  $\Sigma'$  of the form  $(e, \pi)$ , since  $(g, \pi)(e) = e \Leftrightarrow g = e$ . From (10) it follows that the number of right (or left) cosets of  $L'$  in  $\Sigma'$  equals  $N$ .

At every instant the state of the memory is completely determined by a permutation or *memory address map*  $\sigma \in \Sigma$ , such that  $\sigma(x)$  is the physical address of the datum with logical address  $x$ . By (8),  $\sigma$  is determined in turn by the pair  $g \in G$ ,  $\pi \in H$ . To obtain  $\pi$  we need a simulator of the group  $H$ , but  $g$  can then be deduced from the knowledge of the logical address  $s$  of the datum which is presently at the r/w cell  $w$ , for  $\sigma(s) = g\pi(s) = w$  leads to  $g = w(\pi(s))^{-1} = w\pi(s^{-1})$ . In other words, the memory acts as a simulator of the group  $G$ . Now the physical address of a datum with logical address  $x$  is given by

$$\sigma(x) = w\pi(s^{-1})\pi(x) = w\pi(s^{-1}x) \quad (11)$$

Reciprocally, with  $y = \sigma(x)$ , this results in

$$\sigma^{-1}(y) = s\pi^{-1}(w^{-1}y) \quad (12)$$

for the *memory configuration*  $\sigma^{-1}$  which gives the logical address of the datum in any cell  $y$ . Note also that when  $\Sigma'$  is a proper subgroup of  $G \rtimes H$  this calculation uses redundant information.

Once the physical address  $y = \sigma(x)$  of the requested datum  $x$  is known, the next problem is how to transfer it to the r/w cell  $w$ . From the considerations above there are  $\ell$  different permutations  $\tau = (g_\tau, \pi_\tau) \in \Sigma'$  such that  $(g_\tau, \pi_\tau)(y) = g_\tau \pi_\tau(y) = w$ . This means  $g_\tau = w(\pi_\tau(y))^{-1} = w\pi_\tau(y^{-1})$ , so that they all have the form

$$(w\pi(y^{-1}), \pi) \in \Sigma' \quad (13)$$

for  $\ell$  choices of  $\pi \in H$ .

In the digraph  $(V, \{\gamma_i\})$  that models the dynamic memory, the required path from  $y$  to  $w$  corresponds to any factorization of one of these  $\ell$  permutations as a product of the  $\gamma_i$ . Of course, in order to reduce the access time, the number of factors should be minimized both by an adequate choice of  $\pi$  in (13) and when factorizing. This is illustrated in the next section.

Besides sending the datum at  $y$  to  $w$ , it is often required that the memory attains a given state. For instance, when accessing a block of data its ordering should not be modified. This can be accomplished whenever the choice  $\pi = \iota$  is allowed in (13), for then the permutation applied to the memory is  $x \mapsto gx$ , with  $g = wy^{-1}$ . Other possibilities are illustrated in the next section.

#### 4. DYNAMIC MEMORY ORGANIZATIONS

The first part of this section shows how the theory developed in the preceding one applies to most known proposals of dynamic memories. We begin with a simple but interesting example.

##### Cayley diagrams

When  $G$  is a finite group generated by  $\Delta = \{a_0, a_1, \dots, a_{\delta-1}\}$  and we choose  $g_i = a_i$ ,  $\pi_i = \iota$ ,  $0 \leq i \leq \delta-1$ , so that  $\gamma_i(x) = a_i x$ ,  $0 \leq i \leq \delta-1$ , the corresponding digraph,  $D = (V, \{\gamma_i\})$ , which is  $\delta$ -regular and strongly connected, is the Cayley diagram  $D_\Delta(G)$ . Since  $H = \{\iota\}$  is the trivial group, we obtain  $\Sigma \cong G$  and so  $\ell = 1$ .

Therefore, the corresponding dynamic memory has the least possible number of states,  $N = |G|$ , and it can be controlled with just the information of the contents of the r/w cell. Indeed, if the datum at  $w$  was initially at  $s$ , the permutation applied to the memory has been  $ws^{-1}$ . It follows that any other datum that was initially at  $x$  must be now at  $\sigma(x) = ws^{-1}x$  or, reciprocally, that the contents of the cell  $y$  has logical address  $\sigma^{-1}(y) = sw^{-1}y$ . It can be sent to the r/w cell by the permutation  $wy^{-1}$ . As  $\Sigma \cong G$ , a shortest path corresponds to a minimal factorization of  $wy^{-1} \in G$  in terms of the generators. This is easy for some Cayley diagrams but is not solved for others.

##### De Bruijn digraphs

The well-known *de Bruijn digraphs*  $B(\delta, n)$  are set up on the set of  $N = \delta^n$  vertices  $V = \{x = x_0 x_1 \dots x_{n-1}; x_i \in \mathbb{Z}_\delta\}$  by the adjacency rules  $[x, y] \in A(B(\delta, n)) \Leftrightarrow y = x_1 x_2 \dots x_{n-1} x_n, x_n \in \mathbb{Z}_\delta$ .

It is noteworthy that the number of different states  $|\Sigma|$  of a dynamic memory modelled by these digraphs is at least  $n\delta^n$ . Indeed, because of the loop at vertex  $00\dots 0$ , there always exists a permutation  $\gamma$  that fixes it:  $\gamma(00\dots 0) = 00\dots 0$ . Then, the  $n$  permutations  $\gamma, \gamma^2, \dots, \gamma^n$  also fix this vertex and are necessarily different since  $\gamma^p(00\dots 01) \neq \gamma^q(00\dots 01)$  for  $p \neq q$ ,  $1 \leq p, q \leq n$ . Thus  $\ell \geq n$  and  $|\Sigma| \geq n\delta^n$  for any decomposition into permutations of  $B(\delta, n)$ . We next describe a structure that attains this bound.

In our context we can consider  $V$  as the set of elements of the direct product  $G = \mathbb{Z}_\delta \times \dots \times \mathbb{Z}_\delta$  with  $n$  terms. If we now choose for all automorphisms  $\pi_i$

the perfect shuffle permutation

$$\pi_1 = S, 0 \leq i \leq \delta-1, S(x_0 x_1 \dots x_{n-1}) = x_1 \dots x_{n-1} x_0 \quad (14)$$

and

$$g_1 = 00 \dots 0i, 0 \leq i \leq \delta-1,$$

so that (with additive notation)

$$\gamma_1(x) = \gamma_1(x_0 x_1 \dots x_{n-1}) = x_1 \dots x_{n-1} x_0 + i, 0 \leq i \leq \delta-1. \quad (15)$$

We obtain a decomposition into permutations of the digraph  $(V, \{\gamma_i\}) = B(\delta, n)$ . It follows that  $H = \langle S \rangle = C$

is the cyclic group with elements  $S, S^2, \dots, S^n = \epsilon$ , so that  $|G \rtimes H| = n\delta^n$ , and then  $|\Sigma| \leq n\delta^n$  because  $\Sigma \cong \Sigma' \subset G \rtimes H$ . Being a decomposition into permutations of  $B(\delta, n)$ , we necessarily have  $|\Sigma| = n\delta^n$  as stated. This corresponds to  $\Sigma \cong G \rtimes H$ .

The control of the memory can then be achieved with the knowledge of the logical address  $s$  of the datum in the  $r/w$  cell  $w$  and the number  $p \pmod n$  of shuffle permutations that have been applied to the memory. Since each permutation  $\gamma_1$  includes a shuffle permutation,  $p$  can be obtained from a cyclic register that counts modulo  $n$  the number of permutations applied to the memory. Then (11), with  $\pi = S^p$ , gives

$$y = \sigma(x) = w + S^p(x-s), \quad (16)$$

that is,  $y_0 y_1 \dots y_{n-1} = \sigma(x_0 x_1 \dots x_{n-1}) =$

$$= w_0 + x_p - s_p \quad w_1 + x_{p+1} - s_{p+1} \quad \dots \quad w_{n-1} + x_{p+n-1} - s_{p+n-1}$$

is the present position of the datum with logical address  $x$ . Then the following routing algorithm sends it to the  $r/w$  cell in at most  $n$  unit-time intervals:

While  $y_0 y_1 \dots y_{n-1} \neq 00 \dots 0$  do begin  
 Apply  $\gamma_1$  where  $i = -y_0 \pmod \delta$ ;  
 $y_0 y_1 \dots y_{n-1} := y_1 \dots y_{n-2} 0$

Note that, for  $\delta=2$ , (16) may be written as

$$\sigma(x) = w \oplus S^p(s \oplus x) \quad (17)$$

where  $\oplus$  stands for componentwise addition modulo 2. This is the structure proposed by Morris, Valiere III and Wisniewski in [11] for a memory of  $N=2^n$  cells, which is equivalent, except for the numeration of the cells, to the one proposed by Stone in [13,14]. More precisely, Stone makes use of the perfect shuffle ( $\gamma_0$ ) and the exchange shuffle ( $\gamma_1$ ) permutations, which correspond in our formulation to  $\pi_0 = \pi_1 = S$ ,  $g_0 = 00 \dots 0$  and  $g_1 = 00 \dots 010$ .

The general case ( $\delta \neq 2$ ) described above has been studied by the authors in [4].

In order to obtain sequential access and tours in memories based in such organizations see [16].

#### The memory of Aho and Ullman

The structure proposed by Aho and Ullman in [1] corresponds to  $G = \mathbb{Z}_N$ ,  $N = \delta^n - 1$ ,  $\delta \geq 2$ , and permutations  $\gamma_0(x) = \delta x$  ( $\pi_0(x) = \delta x$ ,  $g_0 = 0$ ) and  $\gamma_1(x) = x - 1$  ( $\pi_1 = \epsilon$ ,  $g_1 = -1$ ). Then  $H \cong C_n$  is the cyclic group with elements  $\pi_0, \pi_0^2, \pi_0^3, \dots, \pi_0^n = \epsilon$ , and  $\Sigma \cong G \rtimes H$ . Therefore, the control of the memory requires, besides the knowledge of the logical address of the datum in the  $r/w$  cell, the number  $p$

(mod  $n$ ) of permutations  $\pi_0$  applied to the memory.

Then, (11) and (12) give respectively

$$\begin{aligned} \sigma(x) &= w + \pi_0^p(x-s) = w + \delta^p(x-s) \\ \sigma^{-1}(y) &= s + \pi_0^{n-p}(y-w) = s + \delta^{n-p}(y-w). \end{aligned}$$

To transfer the datum at  $y$  to  $w$  without altering the ordering of the memory, write  $y-w$  in base  $\delta$  as  $y-w = r_{n-1}\delta^{n-1} + \dots + r_1\delta + r_0$  and notice that

$$r_k \delta^k = \pi_0^k(r_k) = \gamma_0^k(r_k). \quad \text{Therefore, the required permutation, given by (13) with } \pi = \epsilon, (w y^{-1}, \epsilon) = (w-y, \pi_0^n) = (-(y-w), \pi_0^n) \text{ can be obtained as}$$

$$\gamma_1^0 \gamma_0 \gamma_1^1 \gamma_0 \dots \gamma_0 \gamma_1^{n-2} \gamma_0 \gamma_1^{n-1} \gamma_0.$$

This structure has been slightly improved by Stone in [14], and Wong and Tang in [15] who replace  $\pi_0$  by  $\pi_0^{n-c}$  for different values of  $c$ .

#### Odd-sized memories

Another structure, proposed by Morris, Valiere III and Winiewski in [11] for any odd number of cells  $N=2m-1$  and for  $\delta=2$ , corresponds in our formulation to  $G = \mathbb{Z}_N$ ,  $\pi_0(x) = \pi_1(x) = 2x$ ,  $g_0 = 0$  and  $g_1 = 1$ . This leads to  $\gamma_0(x) = 2x$  (Out shuffle permutation) and  $\gamma_1(x) = 2x+1$  (In shuffle permutation). Then  $\Sigma \cong G \rtimes H$  and  $H \cong C_n$ , where  $n$  is the power to which 2 belongs modulo  $N$ , that is, the minimum  $n$  such that  $2^n \equiv 1 \pmod N$ . As before, if  $p$  is the number of permutations modulo  $n$  applied to the memory, (11) and (12) lead to

$$\begin{aligned} \sigma(x) &= w + \pi_0^p(x-s) = w + 2^p(x-s); \\ \sigma^{-1}(y) &= s + \pi_0^{n-p}(y-w) = s + 2^{n-p}(y-w). \end{aligned}$$

#### Memories of Lenfant

Let  $R_1, R_2, \dots, R_n, N, R_i > 1, N > 1$ , be integers such that  $R_1 R_2 \dots R_n \equiv 1 \pmod N$ . The structures proposed by Lenfant in [10] correspond in our formulation to  $G = \mathbb{Z}_N$ ,  $\pi_0 = \epsilon$ ,  $g_0 = -1$  and  $\pi_j(x) = R_j x$ ,  $g_j = 0$ ,  $1 \leq j \leq \delta-1$ , where  $\{R_1, R_2, \dots, R_{\delta-1}\}$  is the set of different integers in  $R_1, R_2, \dots, R_n$ . These organizations work as fast cyclic shift registers and are therefore suitable for sequential access. Some similar topologies with better access times are considered by Serra, Lladó and Fiol in [12].

We propose in what follows two new structures. The first one may be seen as a modification of the memory of Aho and Ulman for  $\delta=2$  that improves the worst-case access time for random access so that bound (1) is attained. The second construction is suitable for modelling a dynamic memory able to handle  $m$ -vectors.

#### An optimal size memory

According to (1) for  $\delta=2$  and  $k=n-1$ , there are at most  $2^{n-1}$  cells at distance  $\leq n-1$  of a given one. This bound can be attained with the choice  $G = \mathbb{Z}_n$ ,  $N = 2^n - 1$  and permutations defined by

$$\begin{aligned} \gamma_0(x) &= 2x & (\pi_0(x) &= 2x, g_0 = 0) \\ \gamma_1(x) &= 2x-1 & (\pi_1 = \pi_0, g_1 = -1). \end{aligned} \quad (18)$$

Fig. 2 shows the decomposition into permutations when  $n=3$ .

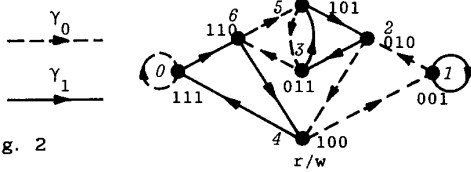


Fig. 2

As in the case of the memory of Aho and Ullman  $H = \langle \pi_0 \rangle \cong C_n$ ,  $\Sigma \cong G \rtimes H$  and

$$y = \sigma(x) = w + \pi^p(x-s) = w + 2^p(x-s) \quad (19)$$

but  $p \pmod n$  is now the number of all permutations applied to the memory, since as well  $\gamma_0$  as  $\gamma_1$  involve a  $\pi_0$  permutation.

To control the memory, both to calculate  $y = \sigma(x)$  and to route the datum at  $y$  to the  $r/w$  cell, it is convenient to represent the different addresses in base 2 (exceptionally 11...1, that should correspond to  $2^n - 1$ , stands for 0) and to use one's complement arithmetic as Stone does in [14]. We thus consider the group of order  $2^n - 1$  that has as elements the  $n$ -length sequences of 0's and 1's different from 00...0, with one's complement addition which we represent by  $\oplus$ . The identity element is 11...1 and the inverse for any other element  $x_0 x_1 \dots x_{n-1}$  is  $\bar{x}_0 \bar{x}_1 \dots \bar{x}_{n-1}$ . The permutations  $\gamma_0$  and  $\gamma_1$  of (18) are now

$$\begin{aligned} \gamma_0(x_0 x_1 \dots x_{n-1}) &= S(x_0 x_1 \dots x_{n-1}) = x_1 \dots x_{n-1} x_0 \\ \gamma_1(x_0 x_1 \dots x_{n-1}) &= x_1 \dots x_{n-1} x_0 \oplus 11 \dots 10 \end{aligned} \quad (20)$$

since 11...10 is  $-1$  in one's complement addition, and (19) becomes

$$\begin{aligned} y &= y_0 y_1 \dots y_{n-1} = \sigma(x_0 x_1 \dots x_{n-1}) = \\ &= w_0 w_1 \dots w_{n-1} \oplus S^p(x_0 x_1 \dots x_{n-1} \oplus \bar{s}_0 \bar{s}_1 \dots \bar{s}_{n-1}). \end{aligned}$$

If we place the  $r/w$  cell at  $w = 100 \dots 0$  any datum can easily be routed from  $y = y_0 y_1 \dots y_{n-1}$  to  $w$  in at most  $n-1$  steps. The idea is to shift the last 1 in the  $y$  sequence (there will always exist one) to the first position using either  $\gamma_0$  or  $\gamma_1$  so as to make it be repeatedly followed by 0's.

#### Vectorial dynamic memories

We just sketch a second construction of a dynamic memory which is able to handle  $m$ -vectors.

Given a strongly connected  $\delta$ -regular digraph  $D = (V, A)$  with a decomposition into permutations  $\{\gamma_i; 0 \leq i \leq \delta - 1\}$  that generate a group  $\Sigma$ , consider the digraph  $D^* = (V^*, A^*)$  where  $V^* = \mathbb{Z}_m \times V$  and each vertex  $(a, x)$  is adjacent to the  $\delta$  vertices  $(a+1, y)$  for  $[x, y] \in A$ . The decomposition into permutations of  $D$  leads naturally to the decomposition into permutations of  $D^*$  defined by

$$\gamma_i^*((a, x)) = (a+1, \gamma_i(x)), \quad 0 \leq i \leq \delta - 1. \quad (21)$$

A consequence of this definition is that the projection

$$\Phi: V^* \longrightarrow V, \quad \Phi((a, x)) = x \quad (22)$$

is a digraph homomorphism that transforms the permutation  $\gamma_i^*$  of  $D^*$  into the permutation  $\gamma_i$  of  $D$ .

For  $D = B(\delta, n)$  with the decomposition into

permutations (15), the above construction results in the digraph  $D^*$  that is able to model a dynamic memory to store  $\delta^n m$ -vectors. Using the projection  $\Phi$  we can consider that each vector is stored in a cell of a memory modelled by the digraph  $B(\delta, n)$ . Therefore, the control of  $D^*$  for vectors is equivalent to the control of  $B(\delta, n)$ . Besides, the use of the permutations  $\gamma_i$  does not modify the cyclic order of the components of each vector.

#### REFERENCES

- [1] A.V. Aho and J.D. Ullman, "Dynamic memories with rapid random and sequential access", *IEEE Trans. Comput.*, vol. C-23, pp. 272-276, Mar. 1974.
- [2] K.E. Batcher, "Aparatus for shifting data in a long register", U.S. Patent 3605024, Sep. 1971.
- [3] J. Fàbrega, M.A. Fiol and J.L.A. Yebra, "Permutation decomposition of dynamic memory networks", *Proc. ISMM Int. Conf. MIMI'84*, Bari, Italy, pp. 186-189, Jun. 1984.
- [4] J. Fàbrega, M.A. Fiol and J.L.A. Yebra, "Dynamic memory networks for random and block access", *Proc. ISMM Int. Conf. MIMI'85*, Sant Feliu de Guixols, Spain, pp. 204-207, Jun. 1985.
- [5] Honeywell Inf., "Addressable dynamic memory system", U.K. Patent 1406117, Nov. 1972.
- [6] IBM Corp., "Shift register memory with multidimensional dynamic ordering", U.K. Patent 1401098, Sep. 1973.
- [7] B.R. Iyer and J.B. Sinclair, "Dynamic memory interconnection for rapid access", *IEEE Trans. Comput.*, vol. C-33, pp. 923-927, Oct. 1984.
- [8] B.R. Iyer and J.B. Sinclair, "Comparison of dynamic memory organizations: a tutorial", *J. VLSI Comput. Systems*, vol.1, no. 3, pp. 217-241, 1985.
- [9] W. Kluge, "A dynamic memory with fast random access and page transfer properties", *Digital Processes*, vol. 1, no. 4, pp. 279-293, 1975.
- [10] J. Lenfant, "Fast random and sequential access to dynamic memories of any size", *IEEE Trans. Comput.*, vol. C-26, pp. 847-855, Sep. 1977.
- [11] S.B. Morris, A. Valiere III and R.A. Wisniewski, "Processes for random and sequential accessing in dynamic memories", *IEEE Trans. Comput.*, vol. C-28, pp. 225-237, Mar. 1979.
- [12] O. Serra, A.S. Lladó and M.A. Fiol, "Fast cyclic shift register for dynamic memories of any size", *Proc. V IAESTED Int. Symp. AI'87*, Grindelwald, Switzerland, pp. 43-46, Feb. 1987.
- [13] H.S. Stone, "Dynamic memories with enhanced data access", *IEEE Trans. Comput.*, vol. C-21, pp. 359-366, Apr. 1972.
- [14] H.S. Stone, "Dynamic memories with fast random and sequential access", *IEEE Trans. Comput.*, vol. C-24, pp. 1167-1174, Dec. 1975.
- [15] C.K. Wong and D.T. Tang, "Dynamic memories with faster random and sequential access", *IBM J. Res. Devel.*, vol. 21, no. 3, pp. 281-288, May 1977.
- [16] J.L.A. Yebra, J. Fàbrega and M.A. Fiol, "Interconnection networks for dynamic memories: Tours in De Bruijn digraphs", *3ème Colloque Int. Theorie des Graphes et Combinatoire*, Marseille, France, Jun. 1986.