for dynamic memory networks

Josep Fàbrega, Miquel A. Fiol, Oriol Serra and José L.A. Yebra
Universitat Politècnica de Catalunya
Barcelona

Abstract

The aim of this paper is to bring together the work done several years ago by M.A. Fiol and the other authors to formulate a quite general mathematical model for a kind of permutation networks known as dynamic memories. A dynamic memory is constituted by an array of cells, each storing one datum, and an interconnection network between the cells that allows the constant circulation of the stored data. The objective is to design the interconnection network in order to have short access time and a simple memory control. We review how most of the proposals of dynamic memories that have appeared in the literature fit in this general model, and how it can be used to design new structures with good access properties. Moreover, using the idea of projecting a digraph onto a de Bruijn digraph, we propose new structures for dynamic memories with vectorial capabilities. Some of these new proposals are based on iterated line digraphs, which have been widely and successfully used by M.A. Fiol and his coauthors to solve many different problems in graph theory.

1 Introduction

A *dynamic memory* is constituted by an array of cells, each storing one datum, and an interconnection network between the cells that allows the

movement of the data from a cell to a neighboring one at time-unit intervals. The result of all these data transfers during such an interval is called a *memory transformation* and it is a permutation of the contents of the memory. Because of the network topology (each cell is connected to a reduced number of neighboring ones), only a small number of these memory transformations are available at each time-unit interval.

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 (by a suitable sequence of memory transformations) to a specific cell, called the *read/write* (*r/w*) *cell*, where data can be read or written. Hence, the *control problem* in a dynamic memory consists of the following two steps:

(a) To find the physical address of the requested datum from its logical address and some additional information about the memory transformations that have been applied to the memory.

(b) To determine an optimal sequence of memory transformations that route the datum to the r/w cell.

Stone [20, 21] was the first author to propose a general model for dynamic memories. Since then, there have been many different proposals: Aho and Ullman [1], Iyer and Sinclair [11, 12], Kluge [13], Lenfant [15], Morris, Valiere III and Wisniewski [16], Wong and Tang [23], and the authors [5, 6, 7, 8, 9, 18, 19, 24].

In our formulation, the interconnection network of the dynamic memory is modeled by a δ-regular strongly connected digraph D in which the vertices represent the storing cells and the arcs the links between them. We refer to [3] for the standard concepts on digraphs. The memory transformations correspond to a decomposition into permutations of D in the following sense. A *decomposition into permutations* [5] of a δ-regular digraph D is a set $\{\gamma_i;\ 0 \leq i \leq \delta - 1\}$ of δ permutations of the vertices of D that satisfies:

$$\gamma_i(x) \text{ is a vertex adjacent from } x; \quad \gamma_i(x) \neq \gamma_j(x) \text{ for } i \neq j. \tag{1}$$

It is easily shown that any δ-regular digraph can be decomposed (usually in several ways) into permutations. Note that such a decomposition associates

a permutation to every arc of D: $(x, y) \mapsto \gamma_i$ if $\gamma_i(x) = y$, which is actually an (*arc-*)*coloring* of D, since different permutations (colors) are associated to the δ arcs to and from any vertex. Conversely, given a set V together with a set $\{\gamma_i;\ 0 \leq i \leq \delta - 1\}$ of permutations of V that satisfy the second condition in (1), we can consider the δ-regular digraph $(V, \{\gamma_i\})$ that has V as set of vertices and where, in view of the first condition in (1), each vertex x is adjacent to the vertices $\gamma_i(x)$, $0 \leq i \leq \delta - 1$. 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.

2 Group of permutations of a decomposition

We recall that given two groups G and H together with an homomorphism of H into the set of automorphisms of G, $\Pi\colon H \longrightarrow Aut\ G$, $\Pi(h) = \pi_h$, 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

$$(g_1, h_1) \star (g_2, h_2) = (g_1 \pi_{h_1}(g_2), h_1 h_2). \tag{2}$$

It coincides with the direct product $G \times H$ if and only if $\Pi = 0$, that is, when $\Pi(h)$ is always the identity. In this paper H will already be a subgroup of $Aut\ G$ so that, with Π the canonical embedding, (2) becomes $(g_1, \pi_1) \star (g_2, \pi_2) = (g_1 \pi_1(g_2), \pi_1 \pi_2)$. See for instance [17] for the standard concepts on group theory used in this work.

In order to control a dynamic memory we need to know the structure of the set of its different states. This is the group generated by the memory transformations. A useful characterization of this group can be obtained by considering V as the set of elements of a group G, that is, each vertex of V stands for an element of G. Then, given $g_i \in G$ and $\pi_i \in Aut\ G$ for $0 \leq i \leq \delta - 1$, we can define the permutations

$$\gamma_i(x) = g_i \pi_i(x) \quad \forall x \in V, \quad 0 \leq i \leq \delta - 1, \tag{3}$$

and consider the digraph $D = (G, \{\gamma_i\})$ that has the decomposition into permutations $\{\gamma_i;\ 0 \leq i \leq \delta - 1\}$. (The elements g_i and the automorphisms π_i should be chosen in such a way that D is strongly connected.) Let $\Sigma = \langle \gamma_0, \gamma_1, \ldots, \gamma_{\delta-1} \rangle$ be the permutation group generated by $\gamma_0, \gamma_1, \ldots, \gamma_{\delta-1}$.

To characterize Σ, let $H = \langle \pi_0, \pi_1, \ldots, \pi_{\delta-1} \rangle$. We have the following result, see [8, 9].

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

We can identify Σ and its image in $G \rtimes H$, that is, we can identify every permutation $\sigma \in \Sigma$ with the pair $(g, \pi) \in G \rtimes H$ given by $\sigma(x) = g\pi(x)$. Since the digraph D is supposed to be strongly connected, the group Σ acts transitively. Hence, its order satisfies $|\Sigma| = N\ell$, where $N = |G|$ and ℓ is, for any $x \in G$, the order of the stabilizer $\mathrm{St}_\Sigma(x)$ of x in Σ. For $x = e$, the identity of G, the subgroup $\mathrm{St}_\Sigma(e)$ consists of the elements of Σ of the form (e, π), since $(g, \pi)(e) = e \iff g = e$. It follows that the number of right (or left) cosets of $\mathrm{St}_\Sigma(e)$ in Σ equals N.

3 Control of the memory

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. Since $\sigma(x) = g\pi(x)$, the memory address map σ is determined in turn by the pair $g \in G$, $\pi \in H$. To obtain π 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

$$\sigma(x) = g\pi(x) = w\pi(s^{-1})\pi(x) = w\pi(s^{-1}x). \tag{4}$$

Reciprocally, with $y = \sigma(x)$, the *memory configuration* σ^{-1} which gives the logical address of the datum in any cell y is

$$\sigma^{-1}(y) = s\pi^{-1}(w^{-1}y). \tag{5}$$

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 ℓ 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

$$(w\pi(y^{-1}), \pi) \in \Sigma \tag{6}$$

for ℓ appropriate choices of $\pi \in H$. In the digraph $(G, \{\gamma_i\})$ that models the dynamic memory, the required path from y to w corresponds to any factorization of one of these ℓ permutations as a product of the γ_i. Of course, in order to reduce the access time, the number of factors should be minimized both by an adequate choice of π in (6) and when factorizing.

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. When G is the cyclic group $\mathbb{Z}_N$, this can be accomplished whenever the choice $\pi = \iota$, the identity of H, is allowed in (6), for then the permutation applied to the memory is, using additive notation, $x \mapsto g + x$, with $g = w - y$.

In the next section we review how most of the proposals of dynamic memories that have appeared in the literature fit in our general model, and how this model can be used to design structures with optimal access time or with good properties for sequential access. In Sections 5 and 6, using the idea of projecting a digraph onto a de Bruijn one, we propose new structures for dynamic memories with vectorial capabilities. Some of these new proposals are based on iterated line digraphs.

4 Applications of the model

To illustrate the generality of the model, we now show how several known proposals of dynamic memories fit into it.

When G is a finite group generated by $\Delta = \{a_0, a_1, \ldots, a_{\delta-1}\}$ and we choose $g_i = a_i$, $\pi_i = \iota$, $0 \le i \le \delta - 1$, so that $\gamma_i(x) = a_i x$, $0 \le i \le \delta - 1$, the corresponding digraph, $D = (G, \{\gamma_i\})$, which is δ-regular and strongly connected, is the *Cayley digraph* $D(G, \Delta)$. Since $H = \{\iota\}$, we obtain $\Sigma \simeq G$ 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 w, i.e., (4) and (5) give

$$\sigma(x) = ws^{-1}x \quad \text{and} \quad \sigma^{-1}(y) = sw^{-1}y. \tag{7}$$

A generalization of this simple model is the following. The well-known de Bruijn digraphs $B(\delta, n)$ are set up on the set of $N = \delta^n$ vertices $V = \{x = x_0x_1 \ldots x_{n-1};\ x_i \in S\}$, where S is an alphabet on δ symbols, and vertex x is adjacent to vertex y if and only if $y = x_1x_2 \ldots x_{n-1}x_n$, $x_n \in S$. Given a finite group Γ of order δ generated by $\Delta = \Gamma$, we can associate

to Γ a de Bruijn digraph $B(\delta, n)$, with a decomposition into permutations $\{\gamma_i\}$, defining $V(B(\delta, n)) = \{x = x_0 x_1 \dots x_{n-1},\ x_i \in \Gamma\}$ and each vertex x being adjacent to

$$\gamma_i(x) = x_1 \dots x_{n-1} x_n,\ x_n = a_i x_0, \tag{8}$$

through the arc $[x, \gamma_i(x)]$ for all $a_i \in \Gamma$. For $n = 1$, $B(\delta, 1)$ is just the Cayley digraph of Γ when generated by all its elements. Thus, $B(\delta, n)$ with the decomposition into permutations (8) may be called the *n-dimensional Cayley digraph* of Γ, see [7].

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 $00\dots0$, there always exists a permutation γ that fixes it: $\gamma(00\dots0) = 00\dots0$. Then, the n permutations γ, γ^2, $\dots$, γ^n also fix this vertex and are necessarily different since $\gamma^p(00\dots01) \neq \gamma^q(00\dots01)$ 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 π_i the *perfect shuffle* permutation $\pi_i = S$, $0 \leq i \leq \delta - 1$, $S(x_0 x_1 \dots x_{n-1}) = x_1 \dots x_{n-1} x_0$, and $g_i = 00\dots01$, $0 \leq i \leq \delta - 1$, we obtain the decomposition into permutations of the digraph $(V, \{\gamma_i\}) = B(\delta, n)$ given in (8). It follows that $H = \langle S \rangle = C_n$ is the cyclic group with elements S, S^2, $\dots$, $S_n = \iota$, 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, see also [6]. Since each permutation γ_i 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 (4), with $\pi = S^p$, gives $y = \sigma(x) = w + S^p(x - s)$, that is (with additive notation),

$$\begin{aligned} y_0 y_1 \dots y_{n-1} &= \sigma(x_0 x_1 \dots x_{n-1}) = \\ & w_0 + x_p - s_p\ w_1 + x_{p+1} - s_{p+1} \dots w_{n-1} + x_{p+n-1} - s_{p+n-1} \end{aligned} \tag{9}$$

is the present position of the datum with logical address x. Once we know

y, we can use the standard shortest path routing algorithm in $B(\delta, n)$ to send x to the r/w cell in at most n time-unit intervals.

For $\delta = 2$ we can write $\sigma(x) = w \oplus S^p(s \oplus x)$, where $\oplus$ stands for componentwise addition modulo 2. This is the structure proposed by Morris, Valiere III and Wisniewski in [16] 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 [20, 22]. More precisely, Stone makes use of the perfect shuffle (γ_0) and the *exchange shuffle* (γ_1) permutations, which correspond in our formulation to $\pi_0 = \pi_1 = S$, $g_0 = 00\ldots0$ and $g_1 = 00\ldots010$. The general case ($\delta \neq 2$) described above has been studied by the authors in [6, 8].

To achieve sequential access to a block of data, Aho and Ullman proposed in [1] an architecture for a dynamic memory of $N = \delta^n - 1$ cells, using a pair of transformations, that can bring any datum to the r/w cell in $\mathcal{O}(\log N)$ steps. Moreover, once two consecutive data have been accessed, the rest of the block can be accessed in one step per datum. This structure corresponds in our formulation 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 $\gamma_1(x) = x - 1$ ($\pi_1 = \iota, g_1 = -1$). Then $H = \mathbb{Z}_n$ is the cyclic group with elements $\pi_0, \pi_0^2, \ldots, \pi_0^n = \iota$, and $\Sigma \simeq G \rtimes H$. Therefore, the control of the memory requires, besides the knowledge of the logical address of the datum in the r/w cell, the number $p \pmod n$ of permutations π_0 applied to the memory. Then, (4) and (5) give respectively

$$\sigma(x) = w + \pi_0^p(x - s) = w + \delta^p(x - s); \tag{10}$$

$$\sigma^{-1}(y) = s + \pi_0^{n-p}(y - w) = s + \delta^{n-p}(y - w). \tag{11}$$

To transfer the datum at y to w without altering the ordering of the memory, write $y - w$ in base δ as $y - w = \sum_{k=0}^{n-1} r_k \delta^k = \sum_{k=0}^{n-1} \gamma_0^k(r_k)$. Therefore, the required permutation given by (6) with $\pi = \iota$, $(\omega - y, \iota) = (-(y - w), \pi_0^n)$, can be obtained as $\gamma_1^{r_0}\gamma_0\gamma_1^{r_1}\gamma_0 \cdots \gamma_1^{r_{n-1}}\gamma_0$. As pointed out before, if only cyclic permutations of the form (g, ι) are performed on the memory, the memory address map is given by $\sigma(x) = g + x$, and the cyclic structure is always preserved.

This structure has been slightly improved by Stone in [22] and Wong and Tang in [23] who replace π_0 by π_0^{n-c} for different values of c.

In [8, 9] it is shown how the mathematical model can also be used to describe and study other dynamic memories based on de Bruijn digraphs [5], [6], [16], [20], [22], the variants of the memory of Aho and Ullman

[22], [23], the memory of Lenfant [15], together with its improved version studied in [18], the Odd-sized memories proposed by Morris, Valiere III and Wisniewski in [16], an optimal size memory proposed by the authors, and other proposals of the authors suitable for sequential access.

5 Dynamic memories based on iterated line digraphs

As it is well known, de Bruijn digraphs can be obtained by line digraph iterations. In the iterated line digraph $L^n(D)$ of a digraph D, each vertex x represents a sequence $x_0x_1\dots x_n$ of vertices of D such that each x_j is adjacent to x_{j+1} in D, and each vertex x is adjacent to the vertices of the form $x_1\dots x_nx_{n+1}$. It is shown in [10] that, if D is a δ-regular digraph on N vertices with diameter k, its iterated line digraph $L^n(D)$ is also δ-regular on $\delta^n N$ vertices and has diameter $k+n$. The de Bruijn digraph $B(\delta,n)$ is just $L^{n-1}(F_\delta)$, where F_δ is the complete symmetric digraph on δ vertices. Another well-know family of iterated line digraphs is the family of Kautz digraphs $K(\delta,n)=L^{n-1}(F^*_\delta)$, where F^*_δ is now the complete symmetric digraph on $\delta+1$ vertices without loops. The order of $K(\delta,n)$ is $\delta^n+\delta^{n-1}$. In [7] a proposal for a dynamic memory based on $K(\delta,n)$ is presented. See also [2] for a mathematical description of the group of permutations generated by a decomposition into permutations of a Kautz digraph.

Any iterated line digraph $L^n(D)$ of a δ-regular digraph D decomposed into permutations can be adequately mapped onto the de Bruijn digraph $B(\delta,n)$. Notice first that, if $\{\tau_i;\ 0\le i\le\delta-1\}$ is a decomposition into permutations of $F_\delta=B(\delta,1)$, the permutations

$$\phi_i(j_0j_1\dots j_{n-1})=j_1\dots j_{n-1}\tau_i(j_0),\ 0\le i\le\delta-1, \tag{12}$$

where $0\le j_k\le\delta-1$, form a decomposition of $B(\delta,n)=L^{n-1}(F_\delta)$. Let D be a strongly connected δ-regular digraph G' with a decomposition into permutations $\{\beta_i;\ 0\le i\le d-1\}$, and consider its iterated line digraph $L^n(D)$. Every vertex x of $L^n(D)$ corresponds to a sequence of successively adjacent vertices of D, $x_0x_1\dots x_n$. For $0\le i\le n-1$ let β_{j_i} be the permutation belonging to $\{\beta_i\}$ such that $\beta_{j_i}(x_i)=x_{i+1}$. Thus, an alternative way of representing x is $x_0;j_0j_1\dots j_{n-1}$ since $\beta_{j_0}(x_0)=x_1$, $\beta_{j_1}(x_1)=x_2$ and so on. Now vertex x is adjacent to the vertices $y=\beta_{j_0}(x_0);j_1\dots j_{n-1}j_n$,

$0 \leq j_n \leq \delta - 1$, and it is easily verified that the permutations

$$\psi_i(x_0; j_0 j_1 \dots j_{n-1}) = \beta_{j_0}(x_0); j_1 \dots j_{n-1} \tau_i(j_0) \tag{13}$$

for $0 \leq i \leq \delta - 1$, form a decomposition into permutations of $L^n(D)$. Moreover, we obtain as a consequence the following result:

Proposition 2 *Let Φ be the mapping of the vertex set of $L^n(D)$ onto the vertex set of $B(\delta, n)$ defined by $\Phi(x_0; j_0 j_1 \dots j_{n-1}) = j_0 j_1 \dots j_{n-1}$. Then, Φ is a digraph homomorphism such that $\Phi(\psi_i) = \phi_i$ for all i.*

Proof: If vertex x of $L^n(D)$ is adjacent to vertex $y = \psi_i(x)$, then vertex $\Phi(x)$ is adjacent to vertex $\Phi(y) = \Phi(\psi_i(x)) = \phi_i(\Phi(x))$. □

The case $D = B(\delta, 1)$, that is when the mapping Φ is an homomorphism from $B(\delta, n+1)$ onto $B(\delta, n)$, has been studied in the context of the design of feedback shift registers by Lempel [14].

To use $L^n(D)$ as a model for a dynamic memory with vectorial capabilities, select a permutation τ_i of the decomposition of F_δ that fixes an element, that is, such that $\tau_i(j) = j$ for some j, $0 \leq j \leq \delta - 1$, and consider a digraph D of order N that has an N-length cycle. In the decomposition into permutations of D we choose the permutation β_j as the one associated to this cycle. Then, the permutation ψ_i of (13) is such that

$$\psi_i(x_0; jj \dots j) = \beta_j(x_0); j \dots j\tau_i(j) = x_1; jj \dots j. \tag{14}$$

Let the memory be organized in such a way that it consists of δ^n blocks or vectors of N words, each stored at the N cells that are mapped by the homomorphism Φ onto a common vertex of $B(\delta, n)$. If the r/w cell w is at $x_0; jj \dots j$, with the routing algorithm of $B(\delta, n)$ it is possible to send one of the data of any block to w and then, by (14), the application of ψ_i allows the rest of the block to visit the r/w cell at one time-unit interval per word. Notice also that, according to (13), the application of any permutation ψ alters the ordering of the N words of each block as it does the permutation β_{j_0}. Therefore, if D has a multiple ring structure with a permutation β associated to each ring, the initial cyclic ordering of the blocks will be preserved. Alternatively, if N read/write cells are available and they are placed at the N cells $*; jj \dots j$, the N words of each block can be accessed simultaneously.

6 Vectorial dynamic memories

In this last section we will make better use of the idea of projecting a digraph onto a de Bruijn one in order to obtain a dynamic memory with vectorial capabilities. Let $D = (V, A)$ be a strongly connected δ-regular digraph with a decomposition into permutations $\{\gamma_i;\ 0 \leq i \leq \delta - 1\}$ that generate a group Σ, and consider the digraph $D^* = (V^*, A^*)$ where $V^* = \mathbb{Z}_m \times V$ and each vertex (a, x) is adjacent to the vertices $(a + 1, y)$ for $[x, y] \in A$. Then, the decomposition into permutations of D leads naturally to the decomposition into permutations of D^* defined by

$$\gamma_i^*((a, x)) = (a + 1, \gamma_i(x)),\ 0 \leq i \leq \delta - 1. \tag{15}$$

A consequence of this definition is that the projection

$$\begin{array}{c} \Phi :\ D^* \longrightarrow D \\ \Phi((a, x)) = x \end{array} \tag{16}$$

is a digraph homomorphism that preserves the coloring, that is, it transforms the permutation γ_i^* of D^* into the permutation γ_i of D.

Now, let $\Sigma^* = \langle \gamma_0^*, \gamma_1^*, \ldots, \gamma_{d-1}^* \rangle$ and suppose that D^* is strongly connected. Then, $|\Sigma^*| = |V^*|\ell^* = mN\ell^*$ where $\ell^* = |\mathrm{St}_{\Sigma^*}((a, x))|$ and $N = |V|$. The following result restricts the possible values of ℓ^*.

Proposition 3 *ℓ^* divides $\ell = |St_\Sigma(x)|$.*

Proof: The mapping $\psi : \Sigma \to \Sigma^*$ defined by $\alpha = \gamma_{i_1}\gamma_{i_2}\cdots\gamma_{i_n} \mapsto \alpha^* = \gamma_{i_1}^*\gamma_{i_2}^*\cdots\gamma_{i_n}^*$ is an isomorphism between the subgroup of $\mathrm{St}_\Sigma(x)$ formed by the permutations that can be expressed as product of n (multiple of m) permutations γ_i and $\mathrm{St}_{\Sigma^*}((a, x))$. Therefore, $\mathrm{St}_{\Sigma^*}((a, x))$ can be seen as a subgroup of $\mathrm{St}_\Sigma(x)$ and its order ℓ^* must divide the order ℓ of $\mathrm{St}_\Sigma(x)$. □

For $D = B(\delta, n)$ with the decomposition into permutations given by $\gamma_i(x_0x_1 \ldots x_{n-1}) = x_1 \ldots x_{n-1}\ x_0 + i$, the above construction results in the digraph D^* with vertex set $V^* = \mathbb{Z}_m \times (\mathbb{Z}_\delta)^n$ and the decomposition into permutations $\{\gamma_i^*,\ 0 \leq i \leq \delta - 1\}$ defined by:

$$\gamma_i^*(a, x_0x_1 \ldots x_{n-1}) = (a + 1, x_1 \ldots x_{n-1}\ x_0 + i). \tag{17}$$

We call these digraphs $D_s(m, \delta, n)$ since they are the directed version of the graph $C_s(m, \delta, n)$ considered by Delorme and Fahri in [4]. The projection Φ is now

$$\begin{aligned} \Phi :\ & D_s(m, \delta, n) \longrightarrow B(\delta, n) \\ & \Phi((a, x_0 x_1 \dots x_{n-1})) = x_0 x_1 \dots x_{n-1}. \end{aligned} \tag{18}$$

The digraphs $D_s(m, \delta, n)$ can model a dynamic memory to store δ^n vectors of size m. Just assign to the component v_x^j, $0 \le j \le m-1$, of vector v_x, $0 \le x \le \delta^m - 1$, the logical address $(j, x_0 x_1 \dots x_{n-1})$ if x equals $x_0 x_1 \dots x_{n-1}$ in base δ. Using the projection Φ we can consider that each vector v_x is stored in a cell of a n-dimensional Cayley digraph $B(\delta, n)$. Therefore, the control of $D_s(m, \delta, n)$ for vectors is equivalent to the control of $B(\delta, n)$ described in Section 4. The use of the permutations γ_i^* do not modify the cyclic order of the components of each vector.

Notice now that when $x_0 = x_1 = \cdots = x_{n-1}$, all the arcs of the m-cycle $(0, x) \to (1, x) \to \cdots \to (m-1, x) \to (0, x)$ correspond to γ_0^*. As a consequence, if the r/w cell is at one of these vertices, once we have attained a component of the vector, the permutation γ_0^* brings at that cell the m components of the vector in cyclic order. The maximum access time is $n + m - 1$, that is, n time-unit intervals to access a component of the vector, and $m - 1$ time-unit intervals to access the requested component in the worst case.

To control the memory, suppose that $(b, s_0 s_1 \dots s_{n-1})$ is the logical address of the datum at the r/w cell, at say $w = (a, 00 \dots 0)$, after q permutations have been applied. Then:

(a) The knowledge of $s_0 s_1 \dots s_{n-1}$ and p $(\equiv q \bmod n)$ gives the position of each vector—control in $B(\delta, n)$;

(b) $a - b$ $(\equiv q \bmod m)$ gives the position of its components.

Therefore, the amount of information required to control the memory is $[m, n]\, \delta^n$ where $[m, n] = \operatorname{lcm}(m, n)$. Since $[m, n]\delta^n = m\delta^n(n/(m, n))$ (where $(m, n) = \gcd(m, n)$), besides the knowledge of the logical address of the datum in the r/w cell, we need to count modulo $\ell^* = n/(m, n)$ the number of permutations applied to the memory. Two interesting particular cases are:

(1) m divides n. The memory has only $|\Sigma^*| = m\delta^n$ states and it can be controlled with just the knowledge of the logical address of the datum in the r/w cell.

(2) $m = \delta = 2$. This case corresponds to a memory of 2^{n+1} cells and worst-case access time $n + 1$ as in $B(2, n + 1)$, but with a smaller average distance and requiring less amount of information to be controlled ($[2, n]2^n$ instead of $(n + 1)2^{n+1}$).

Acknowledgement

This work has been supported by the Spanish Research Council under project MTM2008-06620-C03-01 and by the Catalan Research Council under project 2009SGR01387.

References

[1] A.V. Aho and J.D. Ullman. Dynamic memories with rapid random and sequential access. *IEEE Trans. Comput.*, C-23:272–276, 1974.

[2] J.M. Brunat. Kautz digraphs as group action graphs and Cayley digraphs. *Proc. IWONT 2010*, Barcelona, Spain, June 2010.

[3] G. Chartrand and L. Lesniak. *Graphs and Digraphs*. Wadsworth and Brooks, 1986.

[4] C. Delorme and G. Fahri. Large graphs with given degree and diameter —Part I. *IEEE Trans. Comput.*, C-33:857–860, 1984.

[5] J. Fàbrega, M.A. Fiol and J.L.A. Yebra. Permutation decomposition of dynamic memory networks. *Proc. ISMM Int. Conf. MIMI'84*, Bari, Italy, June 1984, 186–189.

[6] 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, June 1985, 204–207.

[7] M.A. Fiol, J. Fàbrega and J.L.A. Yebra. Cayley and Kautz digraphs as models for dynamic memory networks. *Proc. IV IAESTED Int. Symp. AI'86*, Insbruck, Austria, Feb. 1986, 62–65.

[8] M.A. Fiol, J. Fàbrega and J.L.A. Yebra. The design and control of dynamic memory networks. *Proc. 1988 IEEE Int. Symp. on Circuits and Systems*, Espoo, Finland, June 1988, 175–179.

[9] M.A. Fiol, J. Fàbrega, O. Serra and J.L.A. Yebra. A unified approach to the design and control of dynamic memory networks. *Parallel Process. Lett.*, 3:445–446, 1993.

[10] M.A. Fiol, J.L.A. Yebra and I. Alegre. Line digraph iterations and the (d, k) digraph problem. *IEEE Trans. Comput.*, C-33:400–403, 1984.

[11] B.R. Iyer and J.B. Sinclair. Dynamic memory interconnection for rapid access. *IEEE Trans. Comput.*, C-33:923–927, 1984.

[12] B.R. Iyer and J.B. Sinclair. Comparison of dynamic memory organizations: a tutorial. *J. VLSI Comput. Systems*, 1:217–241, 1985.

[13] W. Kluge. A dynamic memory with fast random access and page transfer properties. *Digital Processes* 1:279–293, 1975..

[14] A. Lempel. On a homomorphism of the de Bruijn graph and its application to the design of feedback shift registers. *IEEE Trans. Comput.*, C-19:1204–1209, 1970.

[15] J. Lenfant. Fast random and sequential access to dynamic memories of any size. *IEEE Trans. Comput.*, C-26:847–855, 1977.

[16] S.B. Morris, A. Valiere III and R.A. Wisniewski. Processes for random and sequential accessing in dynamic memories. *IEEE Trans. Comput.*, C-28:225–237, 1979.

[17] D.J. Robinson. *A Course in the Theory of Groups*. Springer, 1980.

[18] 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, Feb. 1987, 43–46.

[19] O. Serra, M.A. Fiol. An optimal size network for dynamic memories. *Proc. ISMM Int. Conf. MIMI'88*, Sant Feliu de Guixols, Spain, June 1988, 287–291.

[20] H.S. Stone. Dynamic memories with enhanced data access. *IEEE Trans. Comput.*, C-21:359–366, 1972.

[21] H.S. Stone. *Discrete Mathematical Structures and their Applications*. Science Research Associates, 1973.

[22] H.S. Stone. Dynamic memories with fast random and sequential access. *IEEE Trans. Comput.*, C-24:1167–1174, 1975.

[23] C.K. Wong and D.T. Tang. Dynamic memories with faster random and sequential access. *IBM J. Res. Devel.*, 21: 281–288, 1977.

[24] J.L.A. Yebra, J. Fàbrega and M.A. Fiol. Interconnection networks for dynamic memories: Tours in de Bruijn digraphs. *3rme Colloque Int. Theorie des Graphes et Combinatoire*, Marseille, France, Jun. 1986.