A Practical Constructive Scheme for Deterministic Shared-Memory Access

A. Pietracaprina and F.P. Preparata

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-93-14
April 1993

A Practical Constructive Scheme for Deterministic Shared-Memory Access*

A. Pietracaprina[†‡] and F.P. Preparata[‡]

Abstract

We present an explicit memory organization scheme for distributing M data items among N memory modules where $M \in \Theta(N^{1.5-O(1/\log N)})$. Each datum is replicated into a constant number of copies assigned to distinct modules. Assuming that N processors are connected to the memories through a complete graph, we provide an access protocol so that the processors can read/write any set of $N' \leq N$ distinct data in $O((N')^{1/3}\log^* N' + \log N)$ worst-case time. The address computation can be carried out efficiently without resort to a complete memory map and using $O(1)$ internal storage per processor.

1 Introduction

Consider a parallel system with N processors and N memory modules collectively storing $M \gg N$ *variables* that are available for access by the processors. A *memory organization scheme* is sought to distribute the data among the modules so that any set of $N' \leq N$ variables can be efficiently accessed by the processors in parallel. This problem, originally referred to as *granularity problem*, naturally arises in the design and implementation of parallel systems (such as PRAMs and parallel databases) and has received considerable attention in the literature. An early survey by [Kuc77] quotes fourteen works that deal with some special cases. More recently, it has become the main focus of the large body of work concerning the simulation of the PRAM on distributed memory machines [MV84, UW87, AHMP87, KU88, LPP88, Her89, LPP90, Her90, Ran91, Mey92, KLM92].

It is convenient to study this problem on a synchronous system where processors and memories are ideally thought of as being connected by a complete bipartite graph and each memory module is able to fulfill at most one access request (read/write) per time unit (*Module Parallel Computer* (MPC)) [MV84]. Thus, the time needed to access a set of variables is proportional to the maximum number of access requests that a single module must fulfill. This modeling enables us to separate the request routing problem - to be dealt with when the bipartite graph is simulated by a bounded-degree network - from the more difficult memory organization problem.

For the latter, a number of randomized schemes have been successfully developed based on the use of universal classes of hash functions to distribute the variables among the modules [MV84, KU88, LPP88, Ran91, Mey92, KLM92]. Instead, the development of efficient deterministic schemes appears to be much

*This paper was partially supported by NFS Grant CCR-91-96152 and ONR Contract N00014-91-J-4052, ARPA order 8225.

[†]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801

[‡]Department of Computer Science, Brown University, Providence, RI 02912

harder. The pioneering work of Mehlhorn and Vishkin [MV84] introduced the idea of representing each variable by several copies so that a read operation needs access only one (the most convenient) copy. This is necessary to avoid the worst case when all the requests are addressed to the same module. For $M \in O(N^c)$, they present a memory organization scheme that uses c copies per variable and allows a set of N read requests to be satisfied in time $O(cN^{1-1/c})$. However, this use of the copies penalizes the execution of write operations where all the copies of the variables must be accessed, so requiring $O(cN)$ time in the worst case.

Later, Upfal and Widgerson [UW87] proposed a more balanced use of the multiple copies exploiting the majority concept previously adopted for databases [Tho79]. Each variable is represented by $2c-1$ copies. Each copy contains the value of the variable and a time-stamp indicating the last time that particular copy has been accessed. Thus, a read/write operation must access only a majority c of the copies to assure that the most recent value of the variable is always retrieved. The assignment of the copies to the memory modules is represented by a bipartite graph $G = (V, U; E)$, where V represents the set of variables, U the set of modules, and $2c-1$ edges connect each variable to the modules storing its copies. For M polynomial in N and $c \in \Theta(\log N)$, [UW87] show that there exist graphs G with suitable expansion properties so that N variables can be accessed in $O(\log N(\log\log N)^2)$ worst-case time on the MPC. They do not provide an explicit construction for G but show that a random graph exhibits the desired properties, with high probability.

Several authors followed the ideas in [UW87] improving the time complexity and using bounded degree networks instead of the complete network of the MPC [AHMP87, HB88, Her89, LPP90, Her90, AS90]. Such schemes, however, are all based on the *existence* of similar expanding random graphs, which represents the basic shortcoming (maybe fatal from the practical standpoint) of this class of approaches, for the following reasons.

(1) No efficient way is known of testing a random graph for such expansion properties. As [PP93] have recently pointed out, the only known technique, based on the second eigenvalue, cannot be applied when the sizes of the two sets V and U differ by more than a constant factor, which is the case for memory organizations.

(2) The representation of the memory map poses substantial implementation problems. How can a processor determine, for any variable, the modules storing its copies and the physical address of each copy within its module? The hypothesis of a complete memory map stored internally in each processor appears eminently impractical and even the approach of Herley [Her90], where the memory map is distributed among the processors with only polylogarithmic storage per processor, has a very involved implementation.

Recently, we presented explicit deterministic memory organization schemes for $M \in \Theta(N^2)$ and $M \in \Theta(N^3)$ variables where any set of N variables can be accessed in $O(\sqrt{N})$ and $O(N^{2/3})$ time on the MPC, respectively, using constant redundancy [PP93]. In both schemes the implementation is simple and each processor can determine the physical address of any copy in $O(\log N)$ time using $O(1)$ storage. In this paper, we deal with a smaller number of variables; this is an interesting case, as [Her90] already pointed out, since, in a typical parallel application, the size of the data memory required is not enormously larger than the number of processors. Our main result is summarized in the following theorem.

Theorem 1 *For $M \in \Theta(N^{1.5-O(1/\log N)})$ there exists an explicit memory organization scheme that uses $O(1)$ copies per variable and such that any set of $N' \leq N$ variables can be accessed in $O((N')^{1/3}\log^* N' + \log N)$ worst-case time on the MPC. Moreover, each processor can determine the physical address of any copy in $O(\log N)$ time using $O(1)$ internal storage per processor.*

The advantage of our scheme is twofold: (1) The memory organization is *constructive*, i.e., not based on existential arguments. (2) Its implementation is simple and involves only elementary algebra, so that the

address computation can be carried out efficiently with a limited use of resources. The less attractive time complexity, however, is mitigated by the fact that with constant redundancy (desirable from a practical standpoint) a polylogarithmic complexity cannot be obtained, at least for $M \in \Omega(N^{1+\epsilon})$. In fact, [UW87] show that using a *fixed* number r of copies per variables on average, the time complexity of any memory organization scheme is at least $\Omega((M/N)^{1/2r})$. In the paper we show that when the copies per variable are exactly r the lower bound can be raised to $\Omega((M/N)^{1/r})$, thus showing that our time performance is not so far from being optimal.

Our memory access mechanism incorporates most of the ideas and notations pioneered by [UW87] and adopted since by several authors. Each variable is represented by $q+1$ copies (i.e., $q+1 = 2c-1$), where q is a power of 2, and a read/write operation needs access only $q/2+1$ copies. The assignment of the copies to the modules is described, as before, by a bipartite graph $G = (V, U; E)$.

The paramount contribution of this paper is the construction and implementation of the graph G and the analysis of its expansion capabilities. Indeed, the scheme presented here and that of [PP93] are the first constructive approaches known to achieve sublinear, and eminently practical, *worst-case* access time with constant redundancy. Inspired by the construction presented by [Mor91] for bounded concentrators, we associate the sets V and U with two quotients of $PGL_2(q^n)$ (the group of non-singular 2×2 matrices over the field $\mathbb{F}_{q^n}$, modulo its center), and define the edges as intersections of cosets. This kind of graphs exhibit a rich algebraic structure and a remarkable 'isotropy' (the graph appears the same from any of its vertices), which make them very attractive for many computer science applications. Here, in particular, we prove that for any set $S \subset V$, we have $|\Gamma(S)| \in \Omega(|S|^{2/3}q)$, where $\Gamma(S) \subset U$ represents the image of S over U in G. This is accomplished without resort to the (here ineffective) standard second-eigenvalue bound, and solves a major problem left open in [Mor91]. This result allows us to achieve the time complexity claimed in Theorem 1 using an access protocol similar to [UW87].

The rest of the paper describes the memory organization scheme. In Section 2 we define the graph G, study its structure and give a tight bound to its expansion capability. Section 3 presents the protocol for satisfying a set of memory requests on the MPC and analyzes its time complexity. Finally, in Section 4 we discuss how to represent the graph so that each processor can efficiently determine the physical address of any data item (copy of a variable).

2 The graph $G(V, U; E)$

Recall that V and U, with $|V| = M$ and $|U| = N$, represent the sets of variables and memory modules, respectively. Let $\mathbb{F}_k$ denote the finite field of order k, with k a prime power, and $\mathbb{F}_k^*$ its multiplicative subgroup. The *Projective Linear Group of degree* 2 *over* $\mathbb{F}_k$ ($PGL_2(k)$) is the group of 2×2 nonsingular matrices with entries in $\mathbb{F}_k$, modulo its center, the group of scalar matrices [Gor68]. For convenience, a matrix of $PGL_2(k)$ will be written either as $\begin{pmatrix} \alpha & \beta \\ \gamma & 1 \end{pmatrix}$ or $\begin{pmatrix} \alpha & \beta \\ 1 & 0 \end{pmatrix}$, with $\alpha, \beta, \gamma \in \mathbb{F}_k$, and the equality between matrices will be modulo scalar multiples. Let q be a power of 2 and $n \geq 3$. Consider the following two subgroups of $PGL_2(q^n)$:

$$H_{n-1} \triangleq \left\{ \begin{pmatrix} a & \alpha \\ 0 & 1 \end{pmatrix}, \ a \in \mathbb{F}_q^*, \text{ and } \alpha \in \mathbb{F}_{q^n} \right\}$$

$$H_0 \triangleq PGL_2(q)$$

The graph G is defined as follows:

$$V = PGL_2(q^n)/H_0$$

$$U \quad = \quad PGL_2(q^n)/H_{n-1}$$

Thus, the variables are associated with the left cosets of H_0 and the modules with the left cosets of H_{n-1}. The edge set is defined as follows:

$$E = \{(AH_0, BH_{n-1}) : A, B \in PGL_2(q^n) \text{ and } AH_0 \cap BH_{n-1} \neq \emptyset\},$$

and it can be easily seen that E is in a one-to-one correspondence with the cosets of $PGL_2(q^n)/(H_0 \cap H_{n-1})$. The following properties derive from well known facts in group theory (see [Gor68]).

Fact 1 *We have*

1. $|V| = \frac{(q^n+1)q^n(q^n-1)}{(q+1)q(q-1)}$
2. $|U| = (q^n+1)\frac{q^n-1}{q-1}$
3. *The degree of each node in* V *is* $q+1$
4. *The degree of each node in* U *is* q^{n-1}

Therefore, $N \in \Theta(q^{2n-1})$ and $M \in \Theta(q^{3n-3}) = \Theta(N^{\frac{3}{2}-\frac{3}{4n-2}})$. Each variable is represented by $q+1$ copies and each module is assigned q^{n-1} copies of distinct variables.

2.1 Notations and Technical Facts

Let γ be a primitive element of $\mathbb{F}_{q^n}$. We have

$$\mathbb{F}_{q^n} = \left\{ \sum_{i=0}^{n-1} a_i \gamma^i, \quad a_i \in \mathbb{F}_q, 0 \leq i < n \right\}$$

We need to distinguish a particular subset of $\mathbb{F}_{q^n}$, consisting of all those elements that can be written as polynomials in γ with constant term a_0 equal to 0. More specifically, we define

$$P_\gamma \quad \stackrel{\triangle}{=} \quad \left\{ \sum_{i=1}^{n-1} a_i \gamma^i, \quad a_i \in \mathbb{F}_q \right\}. \tag{1}$$

Note that $|P_\gamma| = q^{n-1}$.

For $v \in V$ let $\Gamma(v) \subset U$ denote the set of modules storing the copies of v (i.e., $\Gamma(v) = \{u \in U : (v,u) \in E\}$). Correspondingly, for $u \in U$ let $\Gamma(u)$ denote the set of variables storing copies in u (i.e., $\Gamma(u) = \{v \in V : (v,u) \in E\}$). Thus, Γ gives the set of neighbors of a node in the graph. Its definition is naturally extended to subsets. Also define

$$\Gamma^2(u) \quad \stackrel{\triangle}{=} \quad \Gamma(\Gamma(u)) - u \quad \forall\, u \in U. \tag{2}$$

2.2 Structure of G

We first give a convenient representation for the nodes in U. More specifically, we determine a set of matrices representatives of the cosets of $PGL_2(q^n)/H_{n-1} = U$.

Lemma 1

$$U = \left\{ \begin{pmatrix} \gamma^i & 0 \\ 0 & 1 \end{pmatrix} H_{n-1} : \ 0 \le i < \frac{q^n-1}{q-1} \right\} \bigcup \left\{ \begin{pmatrix} \alpha & \gamma^i \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \text{ and } 0 \le i < \frac{q^n-1}{q-1} \right\}$$

Proof: It is not difficult to see that all of the above cosets are distinct and, therefore, (since their number is $(q^n+1)\frac{q^n-1}{q-1}$) they form a partition of U. □

Next, we list a number of facts concerning the functions Γ and Γ^2. Let $A \in PGL_2(q^n)$. We have

Lemma 2 $\Gamma(AH_0) = \{AH_{n-1}\} \cup \left\{ A \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$.

Proof: We first prove that

$$\Gamma(H_0) = \{H_{n-1}\} \cup \left\{ \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}.$$

Note that $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in H_0 \cap H_{n-1}$ and $\begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} \in H_0$, for $a \in \mathbb{F}_q$, so that $\begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} \in H_0 \cap \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1}$

Therefore, $H_0 \cap H_{n-1} \neq \emptyset$ and $H_0 \cap \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} \neq \emptyset$, for $a \in \mathbb{F}_q$. This implies

$$\Gamma(H_0) \supseteq \{H_{n-1}\} \cup \left\{ \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}.$$

However, since $|\Gamma(H_0)| = q+1$ (by Fact 1), and the set $\{H_{n-1}\} \cup \left\{ \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$ consists of $q+1$ distinct cosets, the above containment relation holds with equality.

Now, it is easy to see that for any $A, B \in PGL_2(q^n)$,

$$H_0 \cap BH_{n-1} \neq \emptyset \iff AH_0 \cap ABH_{n-1} \neq \emptyset$$

This implies

$$\Gamma(AH_0) = \{AH_{n-1}\} \cup \left\{ A \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}.$$

□

Lemma 3 $\Gamma(AH_{n-1}) = \left\{ A \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}$.

Proof: Reasoning as before, we only need to prove

$$\Gamma(H_{n-1}) = \left\{ \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}.$$

Since $\begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} \in H_{n-1}$, we have $H_{n-1} \cap \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 \neq \emptyset$ and, therefore,

$$\Gamma(H_{n-1}) \supseteq \left\{ \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}.$$

By Fact 1, $|\Gamma(H_{n-1})| = q^{n-1} = |P_\gamma|$. So, all we have to show is that for any $p_i, p_j \in P_\gamma$, $p_i \neq p_j$, $\begin{pmatrix} 1 & p_i \\ 0 & 1 \end{pmatrix} H_0 \neq \begin{pmatrix} 1 & p_j \\ 0 & 1 \end{pmatrix} H_0$. Suppose, for a contradiction, that there exists $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in H_0$ such that $\begin{pmatrix} 1 & p_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & p_j \\ 0 & 1 \end{pmatrix}$, that is, $\begin{pmatrix} cp_i + a & dp_i + b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & p_j \\ 0 & 1 \end{pmatrix}$. This implies $c = 0$,$d = 1$,$a = 1$ and $p_i + b = p_j$, which violates the definition of P_γ. □

Combining the two lemmas we can prove that, given any pair of variables, their copies share at most one memory module. This is formally stated in the following theorem.

Theorem 2 *Let $A, B \in PGL_2(q^n)$ with $AH_0 \neq BH_0$. Then $|\Gamma(AH_0) \cap \Gamma(BH_0)| \leq 1$.*

Proof: For a contradiction suppose that $|\Gamma(AH_0) \cap \Gamma(BH_0)| \geq 2$. Then, there exist $C, D \in PGL_2(q^n)$ with $CH_{n-1} \neq DH_{n-1}$, such that $CH_{n-1}, DH_{n-1} \in \Gamma(AH_0) \cap \Gamma(BH_0)$ (see Fig. 1). Without loss of generality, we assume that $CH_{n-1} = H_{n-1}$, otherwise we could premultiply A, B, C and D by C^{-1} and obtain a similar situation. With this choice of C, using Lemma 3 we can choose $A = \begin{pmatrix} 1 & p_1 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & p_2 \\ 0 & 1 \end{pmatrix}$, with $p_1, p_2 \in P_\gamma$, $p_1 \neq p_2$. Note that $AH_{n-1} = BH_{n-1} = H_{n-1}$, since, for $p \in P_\gamma$, $\begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_{n-1} = H_{n-1}$. Then by Lemma 2

$$\begin{aligned} \Gamma(AH_0) - H_{n-1} &= \Gamma(AH_0) - AH_{n-1} \\ &= \left\{ \begin{pmatrix} 1 & p_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\} \\ &= \left\{ \begin{pmatrix} p_1 + a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\} \end{aligned}$$

and, similarly,

$$\Gamma(BH_0) - H_{n-1} = \left\{ \begin{pmatrix} p_2 + a & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ a \in \mathbb{F}_q \right\}$$

This implies $|(\Gamma(AH_0) - H_{n-1}) \cap (\Gamma(BH_0) - H_{n-1})| = |\Gamma(AH_0) \cap \Gamma(BH_0) - H_{n-1}| = 0$, which contradicts the assumption that $\Gamma(AH_0)$ and $\Gamma(BH_0)$ share more than one element. □

Next we consider the function Γ^2.

Lemma 4 $\Gamma^2(AH_{n-1}) = \left\{ A \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix} H_{n-1} : \ \alpha \in \mathbb{F}_{q^n} \right\}$.

Proof: By definition (2),

$$\Gamma^2(AH_{n-1}) = \Gamma(\Gamma(AH_{n-1})) - AH_{n-1}$$

By Lemma 3,

$$\Gamma(AH_{n-1}) = \left\{ A \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix} H_0 : \ p \in P_\gamma \right\}.$$

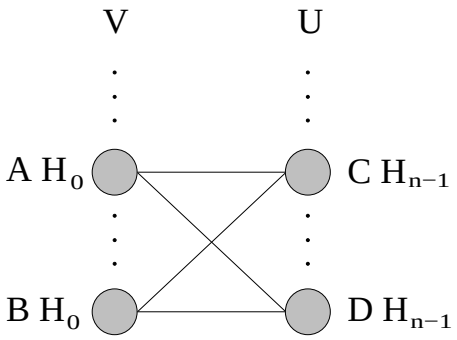


Figure 1: AH_0, BH_0, CH_{n-1}, and DH_{n-1}

By Lemma 2,

$$\begin{aligned}\Gamma\left(A\begin{pmatrix}1 & p\\ 0 & 1\end{pmatrix}H_0\right) &= \left\{A\begin{pmatrix}1 & p\\ 0 & 1\end{pmatrix}H_{n-1}\right\}\cup\left\{A\begin{pmatrix}1 & p\\ 0 & 1\end{pmatrix}\begin{pmatrix}a & 1\\ 1 & 0\end{pmatrix}H_{n-1}:\ \ a\in\mathbb{F}_q\right\}\\ &= \{AH_{n-1}\}\cup\left\{A\begin{pmatrix}p+a & 1\\ 1 & 0\end{pmatrix}H_{n-1}:\ \ a\in\mathbb{F}_q\right\}\end{aligned}$$

since $\begin{pmatrix}1 & p\\ 0 & 1\end{pmatrix}H_{n-1}=H_{n-1}$. Hence,

$$\Gamma^2(AH_{n-1})=\left\{A\begin{pmatrix}p+a & 1\\ 1 & 0\end{pmatrix}H_{n-1}:\ \ a\in\mathbb{F}_q \text{ and } p\in P_\gamma\right\}$$

The proof is completed by observing that $\{p+a:\ \ p\in P_\gamma \text{ and } a\in\mathbb{F}_q\}=\mathbb{F}_{q^n}$. □

Theorem 3 *Let $A,B\in PGL_2(q^n)$ with $AH_{n-1}\neq BH_{n-1}$. Then $|\Gamma^2(AH_{n-1})\cap\Gamma^2(BH_{n-1})|\leq q-1$.*

Proof: Without loss of generality, assume $AH_{n-1}=H_{n-1}$ (same argument as in the proof of Theorem 2). By Lemma 4, $\Gamma^2(H_{n-1})=\left\{\begin{pmatrix}\delta & 1\\ 1 & 0\end{pmatrix}H_{n-1}:\ \ \delta\in\mathbb{F}_{q^n}\right\}$. Then, we have three cases, depending upon the form of BH_{n-1}.

CASE 1: $BH_{n-1}=\begin{pmatrix}\gamma^i & 0\\ 0 & 1\end{pmatrix}H_{n-1}$, for some i, $1\leq i<(q^n-1)/(q-1)$. Applying Lemma 3 we get

$$\begin{aligned}\Gamma^2(BH_{n-1}) &= \left\{\begin{pmatrix}\gamma^i & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\alpha & 1\\ 1 & 0\end{pmatrix}H_{n-1}:\ \ \alpha\in\mathbb{F}_{q^n}\right\}\\ &= \left\{\begin{pmatrix}\gamma^i\alpha & \gamma^i\\ 1 & 0\end{pmatrix}H_{n-1}:\ \ \alpha\in\mathbb{F}_{q^n}\right\}\end{aligned}$$

Since $\gamma^i\neq 1$, we have $|\Gamma^2(H_{n-1})\cap\Gamma^2(BH_{n-1})|=0$.

CASE 2: $BH_{n-1}=\begin{pmatrix}\beta & 1\\ 1 & 0\end{pmatrix}H_{n-1}$, for some $\beta\in\mathbb{F}_{q^n}$. Applying Lemma 3 we get

$$\begin{aligned}\Gamma^2(BH_{n-1}) &= \left\{\begin{pmatrix}\beta & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\alpha & 1\\ 1 & 0\end{pmatrix}H_{n-1}:\ \ \alpha\in\mathbb{F}_{q^n}\right\}\\ &= \left\{\begin{pmatrix}\beta\alpha+1 & \beta\\ \alpha & 1\end{pmatrix}H_{n-1}:\ \ \alpha\in\mathbb{F}_{q^n}\right\}\end{aligned}$$

Now, if $\alpha = 0$ then $\begin{pmatrix} \beta\alpha+1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1} = H_{n-1} \notin \Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1})$. Otherwise, let $-\alpha^{-2} = b\gamma^k$, for some k, $0 \leq k < (q^n-1)/(q-1)$, and $b \in \mathbb{F}_q^*$. Simple calculations show that

$$\begin{pmatrix} \beta+\alpha^{-1} & \gamma^k \\ 1 & 0 \end{pmatrix} \begin{pmatrix} b^{-1} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \alpha^{-1} \\ 0 & 1 \end{pmatrix} H_{n-1} = b^{-1}\alpha^{-1} \begin{pmatrix} \beta\alpha+1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1}$$
$$= \begin{pmatrix} \beta\alpha+1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1}.$$

Since both $\begin{pmatrix} b^{-1} & 0 \\ 0 & 1 \end{pmatrix} \in H_{n-1}$ and $\begin{pmatrix} 1 & \alpha^{-1} \\ 0 & 1 \end{pmatrix} \in H_{n-1}$, we have

$$\begin{pmatrix} \beta\alpha+1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1} = \begin{pmatrix} \beta+\alpha^{-1} & \gamma^k \\ 1 & 0 \end{pmatrix} H_{n-1}.$$

Notice that $\{-\alpha^{-2} : \alpha \in \mathbb{F}_{q^n}^*\} = \{\gamma^{2i} : 0 \leq i \leq q^n-2\} = \mathbb{F}_{q^n}^*$, since q is even. Therefore, there are exactly $q-1$ values of α such that $-\alpha^{-2} \in \mathbb{F}_q^*$, i.e., $k = 0$. These are the values of α for which $\begin{pmatrix} \beta\alpha+1 & \beta \\ \alpha & 1 \end{pmatrix} H_{n-1} \in \Gamma^2(H_{n-1})$. Thus $|\Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1})| = q-1$.

CASE 3: $BH_{n-1} = \begin{pmatrix} \beta & \gamma^i \\ 1 & 0 \end{pmatrix} H_{n-1}$, for some $\beta \in \mathbb{F}_{q^n}$ and i, $1 \leq i < (q^n-1)/(q-1)$. We have $|\Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1})| = q-1$, and the proof is analogous to that for CASE 2.

By recalling the set of representatives for U given by Lemma 1, one concludes that the above three cases cover all possible cosets of U. □

Theorems 2 and 3 have two important consequences, stated below as corollaries, which will play a key role in proving the expansion capability of G.

Corollary 1 *Let $u \in U$ and consider a set of t distinct variables $v_1, \ldots, v_t$ such that $u \in \Gamma(v_i)$, for $1 \leq i \leq t$. Then*

$$|\Gamma(\{v_1, \ldots, v_t\}) - u| = qt.$$

Proof: Immediate from Theorem 2. □

Corollary 2 *Let $S \subset V$ be a set of variables and consider a set of k distinct modules $u_1, \ldots, u_k$. Let $t_i = |\Gamma(u_i) \cap S|$. Then*

$$|\Gamma(S)| \geq \sum_{i=1}^{k} qt_i - \binom{k}{2}(q-1).$$

Proof: Let $\mathcal{A}_i$ denote the set $\Gamma(\Gamma(u_i) \cap S) - u_i$. Since $\Gamma(\bigcup_{i=1}^{k} \mathcal{A}_i) \subset \Gamma(S)$, by the inclusion-exclusion principle we have

$$\begin{aligned} |\Gamma(S)| &\geq \sum_{i=1}^{k} |\mathcal{A}_i| - \sum_{1 \leq i < j \leq k} |\mathcal{A}_i \cap \mathcal{A}_j| \\ &\overset{\text{by Cor. 1}}{\geq} \sum_{i=1}^{k} qt_i - \sum_{1 \leq i < j \leq k} |\Gamma^2(u_i) \cap \Gamma^2(u_j)| \\ &\overset{\text{by Th. 3}}{\geq} \sum_{i=1}^{k} qt_i - \binom{k}{2}(q-1) \end{aligned}$$

□

We are now ready to analyze the expansion properties of graph G. In particular, given a set of variables $S \subset V$ we want to bound from below the size of the set of modules $\Gamma(S)$ containing the copies of the variables in S. This problem was left open in [Mor91], where the node sets V and U were two stages of a multistage diagram, each stage consisting of a quotient of $PGL_2(q^n)$ with respect to a different subgroup.

Theorem 4 *Let $S \subset V$. We have*

$$|\Gamma(S)| \geq \frac{1}{2^{1/3}}|S|^{2/3}q$$

Proof: We first prove that $|\Gamma(S)| > \sqrt{|S|}q$. There are $|S|(q+1)$ edges incident to members of S. Suppose $|\Gamma(S)| \leq \sqrt{|S|}q$. Then there exists a node in U receiving at least $\sqrt{|S|}(q+1)/q$ edges from S. By Corollary 1 this implies $|\Gamma(S)| > q\sqrt{|S|}(q+1)/q > \sqrt{|S|}q$, which contradicts the assumption $|\Gamma(S)| \leq \sqrt{|S|}q$.

Now, let $|\Gamma(S)| = \sqrt{|S|}\epsilon$, with $\epsilon > q$. We claim that each node of $\Gamma(S)$ receives fewer than $\sqrt{|S|}\epsilon/q$ edges from S. Indeed, suppose there is some $u \in \Gamma(S)$ receiving $s \geq \sqrt{|S|}\epsilon/q$ edges from S; clearly, $\Gamma(\Gamma(u) \cap S) \subset \Gamma(S)$ and, by Corollary 1, $|\Gamma(\Gamma(u) \cap S)| = sq+1$, yielding $|\Gamma(S)| \geq |\Gamma(\Gamma(u) \cap S)| \geq \sqrt{|S|}\epsilon + 1$, a contradiction. Let t be the number of nodes in $\Gamma(S)$ receiving at least $\sqrt{|S|}q/\epsilon$ edges from S. We majorize the degrees of these t nodes (strictly) to $\sqrt{|S|}\epsilon/q$, and the degrees of the remaining $(\sqrt{|S|}\epsilon - t)$ nodes to $\sqrt{|S|}q/\epsilon$, and obtain the following inequality:

$$t\sqrt{|S|}\epsilon/q + (\sqrt{|S|}\epsilon - t)\sqrt{|S|}q/\epsilon > |S|(q+1)$$

This implies $t \geq \sqrt{|S|}\frac{q\epsilon}{\epsilon^2 - q^2}$, and the fact $\epsilon > q$ implies $t \geq \sqrt{|S|}q/\epsilon$. Now, let u_i, for $1 \leq i \leq \sqrt{|S|}q/\epsilon$, be distinct nodes of $\Gamma(S)$ receiving each at least $\sqrt{|S|}q/\epsilon$ edges from S. From Corollary 2 we get

$$\begin{aligned}
|\Gamma(S)| &\geq \sum_{i=1}^{\sqrt{|S|}q/\epsilon} q|\Gamma(u_i) \cap S| - \binom{\sqrt{|S|}q/\epsilon}{2}(q-1) \\
&\geq q(\sqrt{|S|}q/\epsilon)^2 - \frac{1}{2}(\sqrt{|S|}q/\epsilon)^2(q-1) \\
&\geq |S|\frac{q^3}{2\epsilon^2}
\end{aligned}$$

Combining the hypothesis $|\Gamma(S)| = \sqrt{|S|}\epsilon$ with the above inequality we obtain

$$|\Gamma(S)| \geq \frac{1}{2^{1/3}}|S|^{2/3}q$$

□

The next theorem proves that when n is composite the lower bound given by Theorem 4 is tight.

Theorem 5 *For any integer m, $0 < m \leq n$ and $m \mid n$, there exists $S \subset V$ such that*

$$\begin{aligned}
|S| &= \frac{(q^m+1)q^m(q^m-1)}{(q+1)q(q-1)} \\
|\Gamma(S)| &= (q^m+1)\frac{q^m-1}{q-1}
\end{aligned}$$

Hence, $|S| \in \Theta(q^{3m-3})$ and $|\Gamma(S)| \in \Theta(q^{2m-1}) = \Theta(|S|^{2/3}q)$.

Proof: Let $S = PGL_2(q^m)/H_0$. Since $m \mid n$, $\mathbb{F}_{q^m}$ is a subfield of $\mathbb{F}_{q^n}$ and $PGL_2(q^m)$ is a subgroup of $PGL_2(q^n)$. Therefore, $S \subset V$ and, by Fact 1, $|S| = \frac{(q^m+1)q^m(q^m-1)}{(q+1)q(q-1)}$, as desired. Since S consists of all the matrices in $PGL_2(q^m)$, and only of those, it is clear that $\Gamma(S) = \{AH_{n-1} : \ A \in PGL_2(q^m)\}$. Let

$$H_{m-1} = \left\{ \begin{pmatrix} a & \alpha \\ 0 & 1 \end{pmatrix}, \ a \in \mathbb{F}_q^*, \text{ and } \alpha \in \mathbb{F}_{q^m} \right\}$$

It is easy to see that $H_{m-1} = H_{n-1} \cap PGL_2(q^m)$, therefore

$$\begin{aligned} |\Gamma(S)| &= |PGL_2(q^m)|/|H_{m-1}| \\ &= |PGL_2(q^m)/H_{m-1}| \\ &= (q^m+1)\frac{q^m-1}{q-1} \end{aligned}$$

□

Corollary 3 *Suppose a set of M variables, each represented by $q+1$ copies, is distributed among the N modules of the MPC according to the graph G. For any integer m, $0 < m \leq n$ and $m \mid n$, there exists a set of variables S, with $|S| = \frac{(q^m+1)q^m(q^m-1)}{(q+1)q(q-1)}$, such that $\Omega(|S|^{1/3}/q)$ parallel steps are needed to access at least one copy per variable.*

Proof: Consider a set S as in Theorem 5. Obviously, the processors cannot access more than $|\Gamma(S)|$ copies per step. Since $|\Gamma(S)| \in \Theta(|S|^{2/3}q)$ and at least S copies have to be accessed, we need $\Omega(|S|/|\Gamma(S)|) = \Omega(|S|^{1/3}/q)$ parallel steps. □

It is not known whether for n prime the result in Theorem 4 is tight.

3 Access Protocol and Analysis

Consider an MPC with N processors, N memory modules and M variables with their copies distributed among the modules according to the graph G defined in the previous section. Suppose each processor issues a read/write request for a distinct variable. Recall that, because of the majority rule, for each variable it is sufficient to access only $q/2+1$ copies.

The access protocol we propose is similar to the ones of [UW87, PP93]. The processors are subdivided into $N/(q+1)$ clusters, with $q+1$ processors per cluster. Let $P(i,j)$ denote the jth processor in cluster i, for $1 \leq i \leq N/(q+1)$ and $1 \leq j \leq q+1$. Let also $v(i,j)$ denote the variable requested by processor $P(i,j)$. The protocol consists of $q+1$ phases. In the kth phase the processors of each cluster cooperate to access the variable requested by their kth companion. More specifically, processor $P(i,j)$ is in charge of the jth copy of $v(i,k)$, for any i and j. A number of iterations are executed. In each iteration each processor tries to access its assigned variable unless it previously succeeded or other $q/2+1$ copies of the same variable have already been accessed. A memory module can satisfy at most one request per iteration, so the number of copies accessed in an iteration is equal to the number of modules receiving requests in that iteration. The code for Phase j is shown in Figure 2. For each variable, a flag is used to indicate whether the variable is alive or not.

Let Φ be the maximum number of iterations executed in any phase. As we will see in the next section, each processor is able to determine the physical address of any copy in $O(\log N)$ time. Therefore, it can be seen that the entire access protocol is completed in $O(q(\Phi \log q + \log N))$ steps on the MPC. We will

Phase j

```
begin
  for i := 1 to N/(q+1) do in parallel
    P(i,j) broadcasts v(i,j) to the processors in cluster i;
    for l := 1 to q+1 do in parallel
      P(i,l) determines the address of the lth copy of v(i,j)
    endfor;
    set flag(i,j) 'alive';
    while flag(i,j) = 'alive' do
      for l := 1 to q+1 do in parallel
        P(i,l) tries to access the lth copy of v(i,j), if not yet accessed
      endfor;
      \* Each memory module accepts one request (if any) *\
      count the number of copies of v(i,j) accessed so far;
      if count ≥ c then set flag(i,j) 'dead' endif
    endwhile
  endfor
end.
```

Figure 2: Code for Phase j

now give an upper bound to the value of Φ. Consider a phase. At some point during the execution of the phase, a copy is said to be *alive* if it has not been accessed yet; a variable is said to be *alive* if at least $q/2+1$ of its copies are still alive. (This terminology is used only for variables and copies requested in the phase under consideration.) We need to slightly modify the result in Theorem 4. For a set of live variables $S \subset V$, let $\Gamma'(S)$ denote the set of modules storing the live copies of S.

Theorem 6 *Let $S \subset V$ be a set of live variables. Then*

$$|\Gamma'(S)| \geq \frac{1}{4}|S|^{2/3}q$$

Proof: We first adapt the results of Corollaries 1 and 2. Consider t live variables $v_1, \ldots, v_t$, each storing a live copy in a fixed module u. Since each live variable has at least $q/2+1$ live copies, by Theorem 2 we get

$$|\Gamma'(\{v_1, \ldots, v_t\}) - \{u\}| \geq \frac{q}{2}t \tag{3}$$

Now consider a set of live variables $S \subset V$ and k memory modules, $u_1, \ldots, u_k$. Let t_i be the number of live copies of variables in S stored in u_i, $1 \leq i \leq k$. Reasoning as in the proof of Corollary 2 and applying (3) we can prove

$$|\Gamma'(S)| \geq \sum_{i=1}^{k} \frac{q}{2}t_i - \binom{k}{2}(q-1) \tag{4}$$

Once these two facts are established, the proof proceeds similarly to that of Theorem 4. We first show that $|\Gamma'(S)| > \sqrt{|S|}(q/2)$. Suppose $|\Gamma'(S)| \leq \sqrt{|S|}(q/2)$. Since there are at least $|S|(q/2+1)$ live copies, there must be a node in U storing at least $\sqrt{|S|}(q/2+1)/(q/2) > \sqrt{|S|}$ live copies. By (3) this implies $|\Gamma'(S)| > \sqrt{|S|}(q/2)$, which contradicts the assumption $|\Gamma'(S)| \leq \sqrt{|S|}(q/2)$.

Now, let $|\Gamma'(S)| = \sqrt{|S|}\epsilon$, with $\epsilon > (q/2)$. Again using (3), we can easily see that each node in $\Gamma'(S)$ must store fewer than $\sqrt{|S|}\epsilon/(q/2)$ live copies. Let t be the number of nodes in $\Gamma'(S)$ storing at least

$\sqrt{|S|}(q/2)/\epsilon$ live copies. We have

$$t\sqrt{|S|}\frac{\epsilon}{(q/2)} + (\sqrt{|S|}\epsilon - t)\sqrt{|S|}\frac{(q/2)}{\epsilon} \geq |S|(q/2+1)$$

This implies $t \geq \sqrt{|S|}\frac{(q/2)\epsilon}{\epsilon^2-(q/2)^2} \geq \sqrt{|S|}(q/2)/\epsilon$. Let $k = \frac{1}{2}\sqrt{|S|}\frac{(q/2)}{\epsilon}$. Note that $t > k$. Consider k nodes of $\Gamma'(S)$ storing at least $\sqrt{|S|}(q/2)/\epsilon$ live copies. Applying (4) we get

$$\begin{aligned}
|\Gamma'(S)| &\geq k\sqrt{|S|}\frac{(q/2)}{\epsilon}\frac{q}{2} - \binom{k}{2}(q-1) \\
&> \frac{1}{2}|S|\left(\frac{(q/2)}{\epsilon}\right)^2\frac{q}{2} - \frac{1}{4}|S|\left(\frac{(q/2)}{\epsilon}\right)^2\frac{q-1}{2} \\
&> \frac{1}{4}|S|\left(\frac{q}{2}\right)^3\left(\frac{1}{\epsilon}\right)^2
\end{aligned}$$

Combining the hypothesis $|\Gamma'(S)| = \sqrt{|S|}\epsilon$ with the above inequality we obtain

$$|\Gamma'(S)| \geq \frac{1}{2\cdot 2^{1/3}}|S|^{2/3}q > \frac{1}{4}|S|^{2/3}q$$

□

For a given phase, let R_k denote the number of live copies remaining after the end of the kth iteration, where $R_0 = N$. The following lemma is similar to Lemma 3.3 of [UW87].

Lemma 5 *For any $k \geq 0$,*

$$R_{k+1} \leq R_k\left(1 - c\left(\frac{q}{R_k}\right)^{1/3}\right)$$

with $0.19 \leq c \leq 0.25$.

Proof: Let L_k be the number of live variables remaining after the kth iteration. Note that $L_k \geq R_k/(q+1)$. By Theorem 6 we have

$$\begin{aligned}
R_{k+1} &\leq R_k - \frac{1}{4}L_k^{2/3}q \\
&\leq R_k - \frac{1}{4}\left(\frac{R_k}{q+1}\right)^{2/3}q \\
&\leq R_k\left(1 - \frac{1}{4}\left(\frac{q}{q+1}\right)^{2/3}\left(\frac{q}{R_k}\right)^{1/3}\right)
\end{aligned}$$

Note that for $q \geq 2$,

$$0.19 \leq c = \frac{1}{4}\left(\frac{q}{q+1}\right)^{2/3} \leq 0.25$$

□

Theorem 7 *If q is a constant, $\Phi \in O(N^{1/3}\log^* N)$.*

Proof: Let $\{k_i\}_{i\geq 0}$ be a sequence of indices such that $0 = k_0 < k_1 < k_2 < \ldots < k_i < \ldots$, and define $\epsilon_i = k_i - k_{i-1}$, for $i \geq 1$. By applying Lemma 5 it follows immediately that

$$R_{k_i} \leq R_{k_{i-1}} \left(1 - c\left(\frac{q}{R_{k_i-1}}\right)^{1/3}\right)^{\epsilon_i} \tag{5}$$

for any $i \geq 1$. Let $b > 1$ be a suitable constant and define the sequence $\{\delta_i\}_{i\geq 0}$ as follows

$$\begin{cases} \delta_0 & = & 1 \\ \delta_{i+1} & = & b^{\delta_i^{1/3}} \quad \text{for } i \geq 0 \end{cases}$$

It can be easily seen that for $i \geq 1$,

$$\delta_i = b^{w^{\cdot^{\cdot^{w}}}}$$

where the exponent of b is a tower of $i-1$ w's and $w = b^{1/3}$. We choose the integers k_i's so that

$$R_{k_i} \leq \frac{N(q+1)}{\delta_i} < R_{k_{i+1}}$$

Thus, for some smallest $j \in O(\log^* N)$, $R_{k_j} \leq 1$.

Claim *For any $i \geq 1$, $\epsilon_i \in O(N^{1/3})$.*
Proof: We need to estimate $\epsilon_i = k_i - k_{i-1}$ knowing that $R_{k_i} \leq \frac{N(q+1)}{\delta_i}$ and $R_{k_i-1} \leq \frac{N(q+1)}{\delta_{i-1}}$. Since the RHS of (5) is an increasing function of R_{k_i-1}, we can substitute $\frac{N(q+1)}{\delta_{i-1}}$ for R_{k_i-1} and obtain

$$R_{k_i} \leq \frac{N(q+1)}{\delta_{i-1}} \left(1 - c\left(\frac{q}{q+1}\frac{\delta_{i-1}}{N}\right)^{1/3}\right)^{\epsilon_i}$$

This inequality certainly holds if

$$\frac{N(q+1)}{\delta_{i-1}} \left(1 - c\left(\frac{q}{q+1}\frac{\delta_{i-1}}{N}\right)^{1/3}\right)^{\epsilon_i} = \frac{N(q+1)}{\delta_i}.$$

Solving for ϵ_i, we get

$$\epsilon_i = \frac{\log_b \delta_i - \log_b \delta_{i-1}}{-\log_b \left(1 - c\left(\frac{q}{q+1}\frac{\delta_{i-1}}{N}\right)^{1/3}\right)}$$

Using the well known fact $\log_b(1+x) \leq x \log_b e$, for $x > -1$, we get

$$\begin{aligned} \epsilon_i &\leq N^{1/3} \frac{1}{c\left(\frac{q}{q+1}\right)^{1/3} \log_b e} \frac{\log_b \delta_i - \log_b \delta_{i-1}}{\delta_i^{1/3}} \\ &\leq N^{1/3} \frac{1}{c\left(\frac{q}{q+1}\right)^{1/3} \log_b e} \frac{\log_b \delta_i}{\delta_{i-1}^{1/3}} \end{aligned}$$

By definition of δ_i, $\log_b \delta_i = \delta_{i-1}^{1/3}$, therefore,

$$\epsilon_i \leq N^{1/3} \frac{1}{c\left(\frac{q}{q+1}\right)^{1/3} \log_b e} \in O(N^{1/3})$$

that proves the claim. □

Combining the result in the claim with the fact that for some $j \in O(\log^* N)$, $R_{k_j} \leq 1$, we conclude that after $\Phi \in O(N^{1/3} \log^* N)$ iterations there are no live copies, i.e., $R_\Phi \leq 1$. □

Therefore, when $q \in O(1)$, the entire protocol requires $O(N^{1/3}\log^* N)$ parallel steps to satisfy a set of N requests.

In general, if $N' < N$ requests are issued by the processors a similar protocol is executed with $\Phi \in O((N')^{1/3}\log^* N')$ iterations per phase. Then the total time complexity becomes $O((N')^{1/3}\log^* N' + \log N)$, as claimed in Theorem 1.

Before closing this section, we want to give the reader an idea on how far from optimal the performance of our scheme is. The following theorem, which is patterned after [UW87, Theorem 4.1] with appropriate modifications, provides a lower bound to the performance of any memory organization scheme.

Theorem 8 *Consider any memory organization scheme where M variables, each represented by exactly r copies, are distributed among N memory modules (with copies of the same variable assigned to distinct modules), and N processors are connected to the memories. If the number of variables that can be stored in the processors' internal registers is negligible with respect to M, then the worst-case time needed to access a set of N variables is*

$$\Omega(\min(N/r, (M/N)^{1/r}))$$

Proof: Let $t-1$ be the number of parallel steps needed to access a set of N variables, in the worst case. If $t > N/(cr)$, for some constant $c > 2$ then we are done. Therefore, assume $t \leq N/(cr)$. Since at least one copy per variable has to be accessed, for any set of N variables their copies must reside in more than N/t modules, otherwise at least $N/(N/t) = t$ steps would be required. So any set of N/t modules contain all of the copies of at most $N-1$ variables. Note that the r copies of any variable occur in exactly

$$\left(\begin{array}{c} N-r \\ N/t-r \end{array} \right)$$

of the $\left(\begin{array}{c} N \\ N/t \end{array} \right)$ qsets of N/t modules. Counting these variables by rows and columns in a (variables) $\times$ (sets of modules) array, we obtain

$$M \leq \frac{(N-1)\left(\begin{array}{c} N \\ N/t \end{array} \right)}{\left(\begin{array}{c} N-r \\ N/t-r \end{array} \right)}. \tag{6}$$

Using the well known fact

$$\left(\begin{array}{c} a \\ b \end{array} \right) \in \Theta\left(\left(\frac{a}{b}\right)^b \left(\frac{a}{a-b}\right)^{a-b} \sqrt{\frac{a}{b(a-b)}} \right)$$

we get

$$\begin{aligned}
\frac{(N-1)\left(\begin{array}{c} N \\ N/t \end{array} \right)}{\left(\begin{array}{c} N-r \\ N/t-r \end{array} \right)} &\in \Theta\left(Nt^{N/t} \left(\frac{N}{N-r}\right)^{N-N/t} \left(\frac{N/t-r}{N-r}\right)^{N/t-r} \right) \\
&= \Theta\left(Nt^r \left(1 - \frac{tr-r}{N-r}\right)^{N/t-r} \left(1 + \frac{r}{N-r}\right)^{N-N/t} \right) \\
&= \Theta\left(Nt^r \exp\left(r\frac{N-N/t}{N-r} - r\frac{(N/t-r)(t-1)}{N-r} \right) \right) \quad \text{(for large } N\text{)} \\
&= \Theta\left(Nt^r\right)
\end{aligned}$$

since $t \leq N/(cr)$. By (6), we obtain

$$t \in \Omega((M/N)^{1/r})$$

□

Note that the result is independent of any strategy adopted to access the variables (e.g., majority rule) and of any interconnection pattern between processors and memories. In our case, choosing $q = 2$ (i.e., $r = 3$), the lower bound becomes $\Omega(N^{((1/2)-O(1/\log N))(1/3)}) = \Omega(N^{(1/6)-O(1/\log N)})$.

4 Implementation

A crucial aspect of the development of a memory organization scheme concerns its implementation, an issue that has been often ignored in the past. In particular, a processor that wants to access a specific copy of a variable must be able to efficiently determine the module storing that copy and the physical address of the copy within the module. This section explains how this can be done in our memory organization.

Let $v_0, \ldots, v_{M-1}$ denote the variables, and $u_0, \ldots, u_{N-1}$ the memory modules. We first need to associate variables and modules with the appropriate cosets. That is, we need to establish the following bijections.

1. For $0 \leq i < M$, $v_i \longleftrightarrow A_i H_0$, for some $A_i \in PGL_2(q^n)$.
2. For $0 \leq j < N$, $u_j \longleftrightarrow B_j H_{n-1}$, for some $B_j \in PGL_2(q^n)$.

Recall that each variable is represented by $q+1$ copies assigned to distinct modules, and that each module is assigned exactly $q^{n-1} = M(q+1)/N$ copies. In other words, in graph G each node of V (U, resp.) has degree $q+1$ (q^{n-1}, resp.). Also recall that the edges of the graph are in one-to-one correspondence with the cosets of $H_0 \cap H_{n-1}$. Let

$$\begin{aligned} A_i H_0 &= \bigcup_{k=0}^{q} A_i^k (H_0 \cap H_{n-1}) \\ B_j H_{n-1} &= \bigcup_{h=0}^{q^{n-1}-1} B_j^h (H_0 \cap H_{n-1}) \end{aligned}$$

for $0 \leq i < M$ and $0 \leq j < N$. We adopt the convention that the kth copy of v_i is the hth item stored in module u_j if and only if $A_i^k(H_0 \cap H_{n-1}) = B_j^h(H_0 \cap H_{n-1})$. Therefore, to determine the address of the kth copy of v_i, a processor proceeds as follows:

STEP 1: Compute A_i^k

STEP 2: Find j and h such that $A_i^k(H_0 \cap H_{n-1}) = B_j^h(H_0 \cap H_{n-1})$

The goal now is to define the A_i's, B_j's, A_i^k's and B_h^j's such that STEP 1 and 2 can be carried out efficiently. The only delicate part regards the definition of the A_i's, which we will deal with later in a separate section. For the time being assume that, given any index i, a processor is able to compute A_i in $O(\log N)$ time using $O(1)$ internal storage (Theorem 11).

We first need to introduce some notations. The elements of the field $\mathbb{F}_{q^n}$ can be denoted by polynomials in γ of degree less than n, with coefficients in $\mathbb{F}_q$. Let $\mathbb{F}_{q^n} = \{\alpha_0, \ldots, \alpha_{q^n-1}\}$, where the indexing is given by the lexicographic order of the n-tuples of coefficients. Note that among the $q^n - 1$ nonzero polynomials

there are $\frac{q^n-1}{q-1}$ monic ones. We will denote these polynomials by π_i, for $0 \leq i < \frac{q^n-1}{q-1}$. Finally, let $\mathbb{F}_q = \{a_0, \ldots, a_{q-1}\}$.

For $0 \leq i < M$, set

$$A_i^k = \begin{cases} A_i \begin{pmatrix} a_k & 1 \\ 1 & 0 \end{pmatrix} & \text{if} \quad 0 \leq k \leq q-1 \\ \\ A_i & \text{if} \quad k = q \end{cases}$$

Lemma 2 implies that for $k_1 \neq k_2$, $A_i^{k_1}(H_0 \cap H_{n-1}) \neq A_i^{k_2}(H_0 \cap H_{n-1})$. For the definition of the B_j's, it is convenient to associate the integers $[0, \ldots, N-1]$ with pairs of indices as follows. For $0 \leq s < \frac{q^n-1}{q-1}$ and $-1 \leq t < q^n$ let

$$J(s,t) \stackrel{\triangle}{=} s(q^n+1) + t + 1.$$

Clearly, $J(s,t)$ takes all the values between 0 and $N-1$. Define

$$B_{J(s,t)} = \begin{cases} \begin{pmatrix} \pi_s & 0 \\ 0 & 1 \end{pmatrix} & \text{if } t = -1 \\ \\ \begin{pmatrix} \alpha_t & \pi_s \\ 1 & 0 \end{pmatrix} & \text{otherwise} \end{cases}$$

Claim 1 *The $B_{J(s,t)}$'s belong to distinct cosets of $PGL_2(q^n)/H_{n-1}$.*

Proof: Note that $\{\pi_s : \quad 0 \leq s < \frac{q^n-1}{q-1}\}$ forms a set of representatives for the quotient $\mathbb{F}_{q^n}^*/\mathbb{F}_q^*$. Since $\{\gamma^s : \quad 0 \leq s < \frac{q^n-1}{q-1}\}$ is also a set of representatives for the same quotient, we have that for each s, $0 \leq s < \frac{q^n-1}{q-1}\}$, there exist $a_s \in \mathbb{F}_q^*$ and s', $0 \leq s' < \frac{q^n-1}{q-1}\}$, such that

$$\pi_s = a_s \gamma^{s'}$$

and if $s_1 \neq s_2$ then $s_1' \neq s_2'$. Now, it is easy to see that

$$\begin{pmatrix} \pi_s & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a_s\gamma^{s'} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \gamma^{s'} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_s & 0 \\ 0 & 1 \end{pmatrix}$$

and

$$\begin{pmatrix} \alpha_t & \pi_s \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \alpha_t & a_s\gamma^{s'} \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \alpha_t & \gamma^{s'} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & a_s \end{pmatrix}.$$

Since both $\begin{pmatrix} a_s & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 0 & a_s \end{pmatrix}$ belong to H_{n-1}, we conclude that

$$\begin{pmatrix} \pi_s & 0 \\ 0 & 1 \end{pmatrix} H_{n-1} = \begin{pmatrix} \gamma^{s'} & 0 \\ 0 & 1 \end{pmatrix} H_{n-1}$$

and

$$\begin{pmatrix} \alpha_t & \pi_s \\ 1 & 0 \end{pmatrix} H_{n-1} = \begin{pmatrix} \alpha_t & \gamma^{s'} \\ 1 & 0 \end{pmatrix} H_{n-1}.$$

Thus, by Lemma 1, the $B_{J(s,t)}$'s belong to distinct cosets of $PGL_2(q^n)/H_{n-1}$. □

Let $P_\gamma = \{p_0, \ldots, p_{q^{n-1}-1}\}$. (Recall that P_γ represents the set of polynomials of degree at most $n-1$ in γ with constant term equal to 0.) For $0 \leq s < \frac{q^n-1}{q-1}$, $-1 \leq t < q^n$ and $0 \leq h < q^{n-1}$ let

$$B^h_{J(s,t)} = B_{J(s,t)} \begin{pmatrix} 1 & p_h \\ 0 & 1 \end{pmatrix}$$

Lemma 3 implies that for $h_1 \neq h_2$, $B^{h_1}_{J(s,t)}(H_0 \cap H_{n-1}) \neq B^{h_2}_{J(s,t)}(H_0 \cap H_{n-1})$. The following lemma identifies the matrices in each coset $B^h_{J(s,t)}(H_0 \cap H_{n-1})$.

Lemma 6 *If $t = -1$ then*

$$B^h_{J(s,t)}(H_0 \cap H_{n-1}) = \left\{ \begin{pmatrix} a\pi_s & (p_h + b)\pi_s \\ 0 & 1 \end{pmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\}$$

Otherwise ($t \geq 0$),

$$B^h_{J(s,t)}(H_0 \cap H_{n-1}) = \left\{ \begin{pmatrix} a\alpha_t & (p_h + b)\alpha_t + \pi_s \\ a & p_h + b \end{pmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\}$$

Proof: It immediately derives for the definition of the $B^h_{J(s,t)}$'s and the fact

$$H_0 \cap H_{n-1} = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\}.$$

□

We are now ready to describe how the two-step procedure that computes the physical address of a copy can be effectively implemented. We assume that each processor has a local work space consisting of a constant number of registers, and that, in addition to the usual arithmetic operations, it is able to perform addition, multiplication, and inverse in the fields $\mathbb{F}_q$ and $\mathbb{F}_{q^n}$. We also assume that arithmetic operations and operations in $\mathbb{F}_q$ take constant time. The operations in $\mathbb{F}_{q^n}$ can be implemented using feedback shift registers, each operation taking $O(n) = O(\log N)$ time and space equivalent to the storage of $O(1)$ field elements (see [Ber68, Ch. 2] for details).

We used $p_0, \ldots, p_{q^{n-1}-1}$ to denote the polynomials of P_γ. Each polynomial can be uniquely associated with its $(n-1)$ coefficients. Thus, the indexing is given by the lexicographic order of the q^{n-1} distinct $(n-1)$-tuples of elements of $\mathbb{F}_q$.

Suppose the address of the kth copy of v_i is sought and consider the two steps presented before. For STEP 1, we assume that A_i has been already determined in $O(\log N)$ time. It is clear that, given A_i, A_i^k can be determined with only a constant number of operations; therefore, STEP 1 requires $O(\log N)$ time, in total.

Let us consider now STEP 2. Let $A_i^k = \begin{pmatrix} x & y \\ z & v \end{pmatrix}$, where x, y, z and v are elements of $\mathbb{F}_{q^n}$ and the usual notation for matrices of $PGL_2(q^n)$ is adopted, i.e., $v = 1$ or $z\,v = 1\,0$. The procedure in Fig. 3, which exploits the result of Lemma 6, computes the indices s, t and h such that $A_i^k \in B^h_{J(s,t)}(H_0 \cap H_{n-1})$, that is, $A_i^k(H_0 \cap H_{n-1}) = B^h_{J(s,t)}(H_0 \cap H_{n-1})$. In the code, given in Fig. 2, we denote the operations in $\mathbb{F}_{q^n}$ as follows

- *(Addition)* $x \oplus y, \ \forall x, y \in \mathbb{F}_{q^n}$

- *(Subtraction)* $x \ominus y, \ \forall x, y \in \mathbb{F}_{q^n}$
- *(Multiplication)* $x \otimes y, \ \forall x, y \in \mathbb{F}_{q^n}$
- *(Inverse)* $\mathrm{INV}(x) = x^{-1}, \ \forall x \in \mathbb{F}_{q^n}^*$

We also use four macros INDEX1, INDEX2, INDEX3 and MONIC, defined as follows. Given $x \in \mathbb{F}_{q^n}$,

- INDEX1$(x) = t$, where $x = \alpha_t$
- INDEX2$(x) = h$, where $x = p_h + b$, for some $b \in \mathbb{F}_q$
- INDEX3$(x) = s$, where $x = a\pi_s$, for some $a \in \mathbb{F}_q^*$
- MONIC$(x) = \pi_s$, where $x = a\pi_s$.

It is easy to see that if x is represented as polynomial in γ of degree less than n, INDEX1, INDEX2, INDEX3 and MONIC are all executed in $O(\log N)$ time.

STEP 2

```
begin
  if z = 0
  then
    begin
      t := −1;
      π_s := MONIC (x);
      s := INDEX3 (x);
      h := INDEX2 (y ⊗ INV (π_s));
    end
  else      \ * z ≠ 0 * \
    begin
      α_t := x ⊗ INV (z);
      t := INDEX1 (α_t);
      if v = 0
      then
        begin
          s := INDEX3 (y);
          h := 0
        end
      else      \ * v = 1 * \
        begin
          π_s := MONIC ((y ⊖ α_t) ⊗ INV (z));
          s := INDEX3 ((y ⊖ α_t) ⊗ INV (z));
          h := INDEX2 ( INV (y ⊖ α_t) ⊗ π_s)
        end
    end
end.
```

Figure 3: Code for STEP 2

Theorem 9 *The code for STEP 2 is correct and is executed in $O(\log N)$ time.*

Proof: Refer to Lemma 6 and distinguish three cases.

CASE 1. ($z = 0$): Clearly $t = -1$ and we must have

$$\begin{pmatrix} x & y \\ z & v \end{pmatrix} = \begin{pmatrix} a\pi_s & (p_h + b)\pi_s \\ 0 & 1 \end{pmatrix}$$

for some $0 \leq s < \frac{q^n-1}{q-1}$, $0 \leq h < q^{n-1}$ and $a, b \in \mathbb{F}_q$ with $a \neq 0$. Therefore, $\pi_s = \text{MONIC}(x)$ and $s = \text{INDEX3}(x)$. Once π_s and s have been determined, we compute $p_h + b = y(\pi_s)^{-1}$, from which the index h is derived using INDEX2.

CASE 2. ($z = 1 \wedge v = 0$): We must have $t \geq 0$ and

$$\begin{pmatrix} x & y \\ z & v \end{pmatrix} = \begin{pmatrix} \alpha_t & a^{-1}\pi_s \\ 1 & 0 \end{pmatrix}$$

for some $0 \leq t < q^n$, $0 \leq s < \frac{q^n-1}{q-1}$ and $a \in \mathbb{F}_q^*$. Note that $h = 0$ since $p_h = 0$. Note also that $\alpha_t = x = xz^{-1}$. The computation of s and t is obvious.

CASE 3. ($z \neq 0 \wedge v = 1$): Again, we must have $t \geq 0$ and

$$\begin{pmatrix} x & y \\ z & v \end{pmatrix} = \begin{pmatrix} a\alpha_t(p_h + b)^{-1} & \alpha_t + (p_h + b)^{-1}\pi_s \\ a(p_h + b)^{-1} & 1 \end{pmatrix}$$

for some $0 \leq t < q^n$, $0 \leq s < \frac{q^n-1}{q-1}$, $0 \leq h < q^{n-1}$ and $a, b \in \mathbb{F}_q$ with $a \neq 0$. As before, $\alpha_t = xz^{-1}$. Once α_t has been computed, we can derive s from $(y - \alpha_t)z^{-1} = a^{-1}\pi_s$. And, given s, we compute $p_h + b$ as $(y - \alpha_t)^{-1}\pi_s$, which enables us to determine h. □

4.1 Representatives for $PGL_2(q^n)/H_0$

The purpose of this section is to find a suitable set of matrices representatives of $PGL_2(q^n)/H_0$, such that, given an integer i, $0 \leq i < M$, the ith matrix is easily retrieved. For simplicity we will only consider the case of $q = 2$ and n odd[1]. The other cases can be handled using similar techniques, but require more involved computations.

Let $\mathbb{F}_2 = \{0, 1\}$. The group $H_0 = PGL_2(2)$ consists of six matrices:

$$H_0(1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad H_0(4) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$H_0(2) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \qquad H_0(5) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

$$H_0(3) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad H_0(6) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

[1]Note that when $q = 2$ there are three copies per variable, which, as [Mey92] pointed out, appears to be one of the interesting cases for practical PRAM simulations.

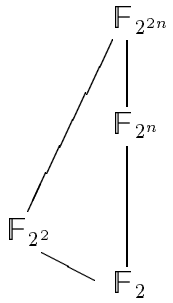
Since $|PGL_2(2^n)| = (2^{2n}-1)2^n$, we have

$$M = |PGL_2(2^n)/H_0| = 2^{n-1}\frac{2^{2n}-1}{3}$$

The main idea that allows us to choose the representatives of $PGL_2(2^n)/H_0$ is to focus on the set of top (bottom) rows of each coset. We first observe that each row of a matrix of $PGL_2(2^n)$ can be uniquely associated with a non-zero element of the extension field $\mathbb{F}_{2^{2n}}$, as follows. Let λ be a generator of the multiplicative group $\mathbb{F}^*_{2^{2n}}$, and let

$$\rho \triangleq \frac{2^{2n}-1}{3}$$

We have $\mathbb{F}^*_{2^2} = \{\lambda^{i\rho} \ : \ 0 \le i < 3\}$; in other words, λ^ρ generates $\mathbb{F}^*_{2^2}$. Let us denote $w \triangleq \lambda^\rho$. Note that $\mathbb{F}_{2^2} \subset \mathbb{F}_{2^{2n}}$ but, since n is odd, $\mathbb{F}_{2^2} \not\subset \mathbb{F}_{2^n}$, as pictured in the diagram below.



Therefore, $w \in \mathbb{F}_{2^{2n}} - \mathbb{F}_{2^n}$ and $(w, 1)$ forms a basis for $\mathbb{F}_{2^{2n}}$ over $\mathbb{F}_{2^n}$; that is, each element of $\mathbb{F}_{2^{2n}}$ can be uniquely written as $\alpha w + \beta$, for some $\alpha, \beta \in \mathbb{F}_{2^n}$. Define the function $\Phi : \mathbb{F}_{2^n} \times \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^{2n}}$ as

$$\Phi(\alpha, \beta) = \alpha w + \beta \quad \forall \alpha, \beta \in \mathbb{F}_{q^n}$$

Now consider a matrix $\begin{pmatrix} x & y \\ z & v \end{pmatrix} \in PGL_2(2^n)$. Such matrix can be uniquely represented by the pair

$$\langle \Phi(x, y), \Phi(z, v) \rangle$$

where, since the matrix is non-singular, both $\Phi(x, y)$ and $\Phi(z, v)$ are nonzero. From now on, both the usual matrix notation and the pair notation will be used interchangeably.

Partition the elements of $\mathbb{F}^*_{2^{2n}}$ (i.e., λ^i, for $0 \le i < 2^{2n}-1$) into cosets of $\mathbb{F}^*_{2^{2n}}/\mathbb{F}^*_{2^2}$. For $0 \le i < \rho$ let

$$[\lambda^i] = \{\lambda^i w^k \ : \ 0 \le k < 3\} = \{\lambda^{i+k\rho} \ : \ 0 \le k < 3\}$$

We need to identify the cosets that contain the elements of $\mathbb{F}^*_{2^n}$. Define

$$\sigma \triangleq 2^n + 1$$
$$\tau \triangleq \frac{2^n+1}{3}$$

and note that $\mathbb{F}^*_{2^n} = \{\lambda^{i\sigma} \ : \ 0 \le i < 2^n - 1\}$. It is easy to prove that

$$\{[\lambda^{i\sigma}] : \ 0 \le i < 2^n - 1\} = \{[\lambda^{i\tau}] \ : \ 0 \le i < 2^n - 1\} \tag{7}$$

and that each such coset contains exactly one element of $\mathbb{F}_{2^n}^*$. We subdivide the cosets of $\mathbb{F}_{2^{2n}}^*/\mathbb{F}_{2^2}^*$ in two sets

$$\begin{aligned} S_1 &= \{[\lambda^{i\tau}] \,:\, 0 \le i < 2^n - 1\} \\ S_2 &= \{[\lambda^i] \,:\, 0 \le i < \rho \text{ such that } \tau \not| \; i\}. \end{aligned}$$

Now, given a row vector (x, y) with $x, y \in \mathbb{F}_{2^n}^*$ and $\Phi(x, y) \neq 0$, consider the six vectors $(x, y)H_0(i)$, for $i = 1, \ldots, 6$. Using the function Φ, view these vectors as elements of $\mathbb{F}_{2^{2n}}^*$, and let

$$\alpha_i = \Phi((x, y)H_0(i)) \quad i = 1, \ldots, 6$$

The following lemma shows that the partition of $\mathbb{F}_{2^{2n}}^*$ into the cosets of $\mathbb{F}_{2^{2n}}^*/\mathbb{F}_{2^2}^*$ is, in a certain sense, preserved among the α_i's. Suppose $\alpha_1 \in [\lambda^k]$, for some k, $0 \le k < \rho$.

Lemma 7

1. $\{\alpha_i \,:\, 1 \le i \le 6\} = [\lambda^k] \cup [\lambda^{(k2^n) \bmod \rho}]$.
2. $[\lambda^k] = [\lambda^{(k2^n) \bmod \rho}]$ *if and only if* τ *divides* k *(i.e.,* $[\lambda^k] \in S_1$*).*

Proof: Observe that

$$\begin{aligned} \alpha_1 &= \Phi(x, y) & \alpha_4 &= \Phi(y, x) \\ \alpha_2 &= \Phi(x + y, x) & \alpha_5 &= \Phi(x, x + y) \\ \alpha_3 &= \Phi(y, x + y) & \alpha_6 &= \Phi(x + y, y) \end{aligned}$$

Note that $\mathbb{F}_{2^2}^* = \{1, w, w + 1\}$ and, since w generates $\mathbb{F}_{2^2}^*$, we must have $w^2 = w + 1$. Easy calculations show that $\Phi(x + y, x) = \Phi(x, y)w$ and $\Phi(y, x + y) = \Phi(x, y)w^2$, therefore

$$[\Phi(x, y)] = \{\Phi(x, y), \Phi(x + y, x), \Phi(y, x + y)\} = \{\alpha_1, \alpha_2, \alpha_3\}$$

Similarly,

$$[\Phi(y, x)] = \{\Phi(y, x), \Phi(x, x + y), \Phi(x + y, y)\} = \{\alpha_4, \alpha_5, \alpha_6\}$$

By induction one can easily prove that, when n is odd, $w^{2^n} = w + 1$. Thus,

$$(\Phi(x, y))^{2^n} = (xw + y)^{2^n} = x^{2^n} w^{2^n} + y^{2^n} = xw^{2^n} + y = xw + (x + y) = \Phi(x, x + y)$$

This implies that

$$\{\alpha_i \,:\, 1 \le i \le 6\} = [\lambda^k] \cup [\lambda^{k2^n \bmod \rho}]$$

Now,

$$\begin{aligned} & (k2^n) \bmod \rho &=& \; k \\ \iff & k2^n &=& \; k + h\rho \quad \text{(for some } h\text{)} \\ \iff & k(2^n - 1) &=& \; h\rho \\ \iff & k &=& \; h\tau \end{aligned}$$

Hence, $[\lambda^k] = [\lambda^{(k2^n) \bmod \rho}]$ if and only if $[\lambda^k] \in S_1$. □

In other words, the set of α_i's consists of exactly two cosets of $\mathbb{F}_{2^n}^*/\mathbb{F}_{2^2}^*$ if $[\lambda^k] \in S_2$, and is equal to $[\gamma^k]$, with each element occurring twice, if $[\gamma^k] \in S_1$.

Before proceeding with the choice of representatives of $PGL_2(2^n)/H_0$ we need a technical fact. For $1 \le i \le \frac{2^{n-1}-1}{3}$ and $0 \le j < 2^n - 1$ define

$$k(i, j) \;=\; (i + j\sigma) \bmod \rho \tag{8}$$

Fact 2 *For any* $1 \leq i \leq \frac{2^{n-1}-1}{3}$ *and* $0 \leq j < 2^n - 1$ *we have*

1. $[\lambda^{k(i,j)}] \in S_2$
2. *For any* $1 \leq i' \leq \frac{2^{n-1}-1}{3}$ *and* $0 \leq j' < 2^n - 1$,
   (a) $k(i',j') = k(i,j) \Longleftrightarrow i' = i$ *and* $j' = j$
   (b) $k(i',j') \neq (2^n k(i,j)) \bmod \rho$

Proof:

1. Since $\tau \mid \sigma$ and $\tau \mid \rho$, then
$$\tau \mid k(i,j) \Longleftrightarrow \tau \mid i.$$
But $1 \leq i \leq \frac{2^{n-1}-1}{3} < \tau$, therefore $\tau \not\mid i$ and $[\lambda^{k(i,j)}] \in S_2$.

2. (a) The 'if' part is trivial. Suppose $k(i',j') = k(i,j)$. Then
$$(i' + j'\sigma) \bmod \rho \;=\; (i + j\sigma) \bmod \rho$$
Since $i \leq \rho$, the above equation can be rewritten as
$$\begin{aligned} i' &= (i + (j - j')\sigma) \bmod \rho \\ &= (i + (j - j')\sigma \bmod \rho) \bmod \rho \end{aligned}$$
Note that $\lambda^{(j'-j)\sigma} \in \mathbb{F}_{2^n}$, hence, by (7), $(j - j')\sigma \bmod \rho = h\tau$ for some h, $0 \leq h < 2^n - 1$. Since $i < \tau$, $i + h\tau < (2^n - 1)\tau = \rho$. Thus we have
$$i' = i + h\tau$$
The condition $i' < \tau$ implies $h = 0$ and, hence, $i = i'$. Also, $h = 0$ implies $(j - j')\sigma = d\rho$, for some $0 \leq d < 2^n - 1$. Now, since n is odd, lcm $(\sigma, \rho) = 2^{2n} - 1$ and, thus, $(j' - j)\sigma = d\rho$ implies that $(j' - j)$ is a multiple of lcm $(\sigma, \rho)/\sigma = 2^n - 1$. But by definition, $(j' - j) < 2^n - 1$, so we must have $(j' - j) = 0$, that is, $j' = j$.

   (b) Suppose for a contradiction that there exist i' and j' such that $k(i',j') = (2^n k(i,j)) \bmod \rho$. Then,
$$(i' + j'\sigma) \bmod \rho = (2^n i + 2^n j\sigma) \bmod \rho,$$
that is,
$$2^n i \bmod \rho = (i' + (j' - 2^n j)\sigma) \bmod \rho$$
Noting that $2^n i < \rho$ and reasoning as before, we conclude that
$$2^n i = i' + h\tau$$
for some h, $0 \leq h < 2^n - 1$. In other words, $i' = 2^n i \bmod \tau$. Since $2^n = 3\tau - 1$, then
$$\begin{aligned} i' &= 2^n i \bmod \tau \\ &= (3\tau - 1)i \bmod \tau \\ &= -i \bmod \tau, \end{aligned}$$
which is impossible since $\tau = (2^n + 1)/3$ and both i and i' are at most $(2^{n-1} - 1)/3 < \lfloor \tau/2 \rfloor$.

□

We are now ready to define the matrices representatives of $PGL_2(2^n)/H_0$. For convenience, we partition these matrices into four groups, L_1, L_2, L_3 and L_4.

Definition 1

$$\begin{aligned}
L_1 &\stackrel{\triangle}{=} \{\langle 1, \lambda^{h\sigma} w\rangle \ : \ 0 \le h < 2^n - 1\} \\
L_2 &\stackrel{\triangle}{=} \{\langle 1, \lambda^{k(i,j)} w^s\rangle\} \\
L_3 &\stackrel{\triangle}{=} \{\langle \lambda^{k(i,j)} w^s, 1\rangle\} \\
L_4 &\stackrel{\triangle}{=} \{\langle \lambda^{k(i,0)}, \lambda^t w^s\rangle \ : \ 1 \le t < \rho\,,\ \tau \not| \ t \quad \textit{and} \\
&\qquad \lambda^{k(i,0)}(\lambda^t w^s)^{-1} \notin \mathbb{F}_{2^n}^*\}
\end{aligned}$$

with $1 \le i \le \frac{2^{n-1}-1}{3}$, $0 \le j < 2^n - 1$, *and* $0 \le s < 3$, *wherever they occur.*

Lemma 8

$$|L_1| + |L_2| + |L_3| + |L_4| = M$$

Proof: It is immediate that

$$\begin{aligned}
|L_1| &= 2^n - 1 \\
|L_2| = |L_3| &= (2^n - 1)(2^{n-1} - 1)
\end{aligned}$$

As for the cardinality of L_4, note that there are $\frac{(2^n-1)(2^n-2)}{3}$ values of t between 1 and ρ not multiples of τ. It is not difficult to show that, for each i, there are exactly $2^n - 1$ pairs of indices t and s, with $1 \le t < \rho$, $\tau \not| \ t$ and $0 \le s < 3$, such that $\lambda^{k(i,0)}(\lambda^t w^s)^{-1} \in \mathbb{F}_{2^n}^*$. Therefore,

$$\begin{aligned}
|L_4| &= \frac{2^{n-1}-1}{3}[3\frac{(2^n-1)(2^n-2)}{3} - (2^n - 1)] \\
&= \frac{2^{n-1}-1}{3}[(2^n-1)(2^n-2) - (2^n-1)] \\
&= (2^n-1)\frac{2^{n-1}-1}{3}(2^n-3)
\end{aligned}$$

Simple algebra shows that $|L_1| + |L_2| + |L_3| + |L_4| = 2^{n-1}\frac{2^{2n}-1}{3} = M$. □

Thus, the A_i's defining the bijection between the variables and the cosets of $PGL_2(2^n)/H_0$, can be identified with the matrices in the above four sets. Let

$$\mathcal{L} \stackrel{\triangle}{=} L_1 \cup L_2 \cup L_3 \cup L_4$$

Theorem 10 *The matrices in* $\mathcal{L}$ *belong to distinct cosets of* $PGL_2(2^n)/H_0$, *thus forming a complete set of representatives.*

Proof: It is sufficient to prove the following two facts. (1) $\mathcal{L} \subset PGL_2(2^n)$. (2) For each $A, B \in \mathcal{L}$, $A \neq B$, and $\delta \in \mathbb{F}_{2^n}$, $A \notin \delta B H_0$.

For the first fact we only have to prove that each $A \in \mathcal{L}$ is non-singular. Let $A = \langle x, y\rangle \in \mathcal{L}$. Note that both x and y are non-zero; therefore A is non-singular if and only if $x^{-1}y \notin \mathbb{F}_{2^n}$ (or equivalently, $xy^{-1} \notin \mathbb{F}_{2^n}$). We distinguish among four cases, according to the form of A.

- $A = \langle 1, \lambda^{h\sigma} w \rangle \in L_1$. Since $w \notin \mathbb{F}_{2^n}$ and $\lambda^{h\sigma} \in \mathbb{F}_{2^n}$, $\lambda^{h\sigma} w \notin \mathbb{F}_{2^n}$.
- $A = \langle 1, \lambda^{k(i,j)} w^s \rangle \in L_2$. By Fact 2, $[\lambda^{k(i,j)} w^s] = [\lambda^{k(i,j)}] \in S_2$ and, by (7) and the definition of S_2, $\lambda^{k(i,j)} w^s \notin \mathbb{F}_{2^n}$.
- $A = \langle \lambda^{k(i,j)} w^s, 1 \rangle \in L_3$. Similar to the previous case.
- $A = \langle \lambda^{k(i,0)}, \lambda^t w^s \rangle \in L_4$. The condition $\lambda^{k(i,0)} (\lambda^t w^s)^{-1} \notin \mathbb{F}_{2^n}$ in the definition of L_4 assures that A is non-singular.

We now prove the second fact. Let $A, B \in \mathcal{L}$, with $A \neq B$, and let $\delta = \lambda^{a\sigma}$, for some a, $0 \leq a < 2^n - 1$. As before, we need to distinguish among various cases according to the form of A and B.

- *A and B belong to distinct L_i's.* We show only the case of $A \in L_1$ and $B \in L_2$. The other cases can be dealt with in a similar fashion. Let $A = \langle 1, \lambda^{h\sigma} w \rangle$ and $\delta B = \langle \lambda^{a\sigma}, \lambda^{k(i,j)+a\sigma} w^s \rangle$. Note that $[\lambda^{h\sigma} w] \in S_1$ and $[\lambda^{k(i,j)+a\sigma} w^s] \in S_2$. From Lemma 7, one can argue that for any $\langle x, y \rangle \in \delta B H_0$, $[y] \in S_2$. Therefore, $A \notin \delta B H_0$.
- *$A, B \in L_1$, with $A \neq B$.* Let $A = \langle 1, \lambda^{h\sigma} w \rangle$ and $B = \langle 1, \lambda^{h'\sigma} w \rangle$ with $h' \neq h$. Thus $\delta B = \langle \lambda^{a\sigma}, \lambda^{(h'+a)\sigma} w \rangle$. Note that $[1]$, $[\lambda^{h\sigma} w]$, $[\lambda^{a\sigma}]$ and $[\lambda^{(h'+a)\sigma} w]$ are all in S_1. By Lemma 7, for any $\langle x, y \rangle \in \delta B H_0$, $[x] = [\lambda^{a\sigma}]$, so if $A \in \delta B H_0$ we would have $[\lambda^{a\sigma}] = [1]$ (i.e., $a = 0$). For the same reason, $[y] = [\lambda^{(h'+a)\sigma} w]$; so if $a = 0$, $A \in \delta B H_0$ would imply $[\lambda^{h'\sigma}] = [\lambda^{h\sigma}]$, that is, $h = h'$, a contradiction.
- *$A, B \in L_2$, with $A \neq B$.* Let $A = \langle 1, \lambda^{k(i,j)} w^s \rangle$ and $B = \langle 1, \lambda^{k(i',j')} w^{s'} \rangle$, where it cannot be $(i = i') \wedge (j = j') \wedge (s = s')$. Thus, $\delta B = \langle \lambda^{a\sigma}, \lambda^{k(i',j')+a\sigma} w^{s'} \rangle$ As in the previous case, we can show that $A \in \delta B H_0$ implies $a = 0$. Since $[\lambda^{k(i',j')} w^{s'}] = [\lambda^{k(i',j')}] \in S_2$, by Lemma 7 for any $\langle x, y \rangle \in \delta B H_0$, $[y] = [\lambda^{k(i',j')}]$, or $[y] = [\lambda^{(k(i',j')2^n) \bmod \rho}]$. Therefore, in order to have $A \in \delta B H_0$, we need $[\lambda^{k(i,j)}] = [\lambda^{k(i',j')}]$ or $[\lambda^{k(i,j)}] = [\lambda^{(k(i',j')2^n) \bmod \rho}]$, which, by Fact 2, implies $i = i' \wedge j = j'$. Thus we have $A = \langle 1, \lambda^{k(i,j)} w^s \rangle$ and $\delta B = \langle 1, \lambda^{k(i,j)} w^{s'} \rangle$ with $s \neq s'$. Suppose $i = i'$, $j = j'$ and $\delta = 1$. From the proof of Lemma 7 one can easily see that the only matrix in BH_0 other than B itself of the form $\langle 1, y \rangle$ must have $[y] = [\lambda^{(k(i,j)2^n) \bmod \rho}]$. Therefore, A cannot belong to BH_0.
- *$A, B \in L_3$, with $A \neq B$.* Similar to the previous case.
- *$A, B \in L_4$, with $A \neq B$.* Let $A = \langle \lambda^{k(i,0)}, \lambda^t w^s \rangle = \langle \lambda^i, \lambda^t w^s \rangle$ and $B = \langle \lambda^{k(i',0)}, \lambda^{t'} w^{s'} \rangle = \langle \lambda^{i'}, \lambda^{t'} w^{s'} \rangle$ where it cannot be $(i = i') \wedge (t = t') \wedge (s = s')$. Thus, $\delta B = \langle \lambda^{i'+a\sigma}, \lambda^{t'+a\sigma} w^{s'} \rangle$. Note that $[\lambda^{i'+a\sigma}] = [\lambda^{k(i',a)}]$, and by Lemma 7 $A \in \delta B H_0$ requires $k(i,0) = k(i',a)$, which, by Fact 2, implies $i = i'$ and $a = 0$. Thus we must have $A = \langle \lambda^i, \lambda^t w^s \rangle$ and $\delta B = \langle \lambda^i, \lambda^{t'} w^{s'} \rangle$ Now, since $[\lambda^i] \in S_2$, Lemma 7 implies that δB is actually the only matrix in $\delta B H_0$ whose first component is λ^i. Therefore, in order to have $A \in \delta B H_0$ it must be $A = \delta B$, that is $t = t'$ and $s = s'$, a contradiction.

□

4.2 Numbering of the representatives of $PGL_2(2^n)/H_0$

In this section we show how the integers $0, 1, \ldots, M-1$ can be put in one-to-one correspondence with the matrices chosen in the previous section as representatives of $PGL_2(2^n)/H_0$ (i.e., the set $\mathcal{L}$). We first explain how, given and integer r, $0 \leq r < M$ the rth matrix of $\mathcal{L}$, say A_r, can be computed in the pair notation. Assume that a primitive element $\lambda \in \mathbb{F}_{2^{2n}}$ is known.

Recall that $\mathcal{L} = L_1 \cup L_2 \cup L_3 \cup L_4$ and, as shown in the proof of Lemma 8

$$\begin{aligned} |L_1| &= 2^n - 1 \\ |L_2| = |L_3| &= (2^n - 1)(2^{n-1} - 1) \\ |L_4| &= (2^n - 1)\frac{2^{n-1} - 1}{3}(2^n - 3) \end{aligned}$$

where $|L_1| + |L_2| + |L_3| + |L_4| = M$. Let $0 \leq r < M$. We number the matrices so that

$$\begin{array}{lrcl} A_r \in L_1 \quad \text{if} & 0 & \leq r < & 2^n - 1 \\ A_r \in L_2 \quad \text{if} & 2^n - 1 & \leq r < & (2^n - 1) + (2^n - 1)(2^{n-1} - 1) = (2^n - 1)2^{n-1} \\ A_r \in L_3 \quad \text{if} & (2^n - 1)2^{n-1} & \leq r < & (2^n - 1)2^{n-1} + (2^n - 1)(2^{n-1} - 1) = (2^n - 1)^2 \\ A_r \in L_4 \quad \text{if} & (2^n - 1)^2 & \leq r < & (2^n - 1)^2 + (2^n - 1)\frac{2^{n-1}-1}{3}(2^n - 3) = M \end{array}$$

Thus we must find a one-to-one mapping between the indices in each range and the matrices of the appropriate set. For the first three ranges, the mapping can be easily established based on the definition L_1, L_2 and L_3, and involves only a constant number of operations. The case of L_4 is slightly more complicated. Suppose $(2^n - 1)^2 \leq r < M$ and set $r' = r - (2^n - 1)^2$, so that $0 \leq r' < (2^n - 1)\frac{2^{n-1}-1}{3}(2^n - 3)$. A_r will be of the form

$$A_r \quad = \quad \langle \lambda^{k(i,0)}, \lambda^t w^s \rangle$$

for some i, t and s such that $1 \leq i \leq \frac{2^{n-1}-1}{3}$, $1 \leq t < \rho$, $\tau \not| \; t$, $0 \leq s < 3$ and $\lambda^{k(i,0)}(\lambda^t w^s)^{-1} \notin \mathbb{F}_{2^n}$. So we must associate r' with a triplet (i, t, s). The index i can be chosen as

$$i \quad = \quad \left\lfloor \frac{r'}{(2^n - 1)(2^n - 3)} \right\rfloor + 1,$$

which is clearly a value between 1 and $\frac{2^{n-1}-1}{3}$. Next we determine t and s. Fixed i, there are $(2^n - 1)(2^n - 3)$ pairs t, s to choose from and we want to find the lth, where $l = r' \bmod (2^n - 1)(2^n - 3)$. Note that, since $w = \lambda^\rho$, then $\lambda^t w^s = \lambda^{t+s\rho}$. It is not difficult to see that

$$\{t + s\rho : \;\; 1 \leq t < \rho, \; \tau \not| \; t \;\; \text{and} \;\; 0 \leq s < 3\} \quad = \quad \{t' : \;\; 0 \leq t' < 2^{2n} - 1, \; \tau \not| \; t'\}$$

So we have to find the lth index t', which is not multiple of τ and such that $\lambda^{k(i,0)-t'} = \lambda^{i-t'} \notin \mathbb{F}_{2^n}$; that is $t' \neq i + j\sigma$ for any $0 \leq j < 2^n - 1$. Among the integers $0 \ldots 2^{2n} - 2$, those that are multiples of τ or are equal to $i + j\sigma$ (i.e., the 'forbidden' indices), occupy fixed and easily calculable positions, as shown in Fig. 4 where the integers $0 \ldots 2^{2n} - 2$ are arranged consecutively into $2^n - 1$ rows and σ columns. Note that the multiples of τ are those in columns $0, \tau$ and 2τ, and the values $i + j\sigma$, with $0 \leq j < 2^n - 1$, are those in column i. It is not hard to see that t' can be computed with a constant number of operations.

Finally, from t' we determine t and s as follows. Since $t' = t + s\rho$, we have

$$\begin{aligned} t &= t' \bmod \rho \\ s &= \lfloor t'/\rho \rfloor \end{aligned}$$

Once the matrix A_r is known in pair notation, we need to transform it into the usual form. More specifically, let $A_r = \langle \lambda^i, \lambda^j \rangle$, for some i and j. We want to find $\alpha_i, \beta_i, \alpha_j, \beta_j \in \mathbb{F}_{2^n}$ such that

$$\begin{aligned} \alpha_i w + \beta_i &= \lambda_i \\ \alpha_j w + \beta_j &= \lambda_j \end{aligned}$$

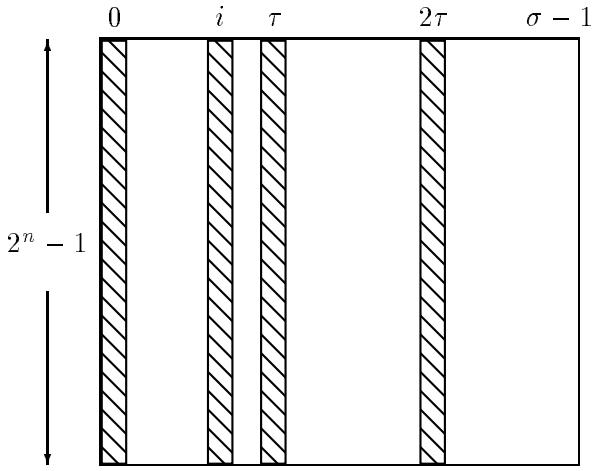


Figure 4: Table of the indices $0, \ldots, 2^{2n} - 2$

Suppose we know that $\lambda = \alpha_1 w + \beta_1$ (each processor has to store the two values α_1 and β_1). Then given i and j and using the fact $w^2 = w + 1$, $\alpha_i, \beta_i, \alpha_j$ and β_j can be easily computed with $O(n) = O(\log N)$ operations over $\mathbb{F}_{2^n}$.

In summary, we have proved

Theorem 11 *Given an integer r, $0 \leq r < M$, a processor is able to compute the rth matrix of $\mathcal{L}$ in $O(\log N)$ time using $O(1)$ internal storage.*

References

[AHMP87] H. Alt, T. Hagerup, K. Mehlhorn, and F.P. Preparata. Deterministic simulation of idealized parallel computers on more realistic ones. *SIAM J. on Computing*, 16(5):808–835, 1987.

[AS90] Y. Aumann and A. Schuster. Improved memory utilization in deterministic PRAM simulations. Manuscript, 1990.

[Ber68] E.R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill, 1968.

[Gor68] D. Gorenstein. *Finite Groups.* Harper and Row, New York NY, 1968.

[HB88] K.T. Herley and G. Bilardi. Deterministic simulations of PRAMs on bounded degree networks. *Proc. of the 26th Annual Allerton Conference on Communication, Control and Computation*, pages 1084–1093, 1988.

[Her89] K.T. Herley. Efficient simulations of small shared memories on bounded degree networks. *Proc. of the 30th IEEE Symp. on Foundations of Comp. Sc.*, pages 390–395, 1989.

[Her90] K.T. Herley. Space-efficient representations of shared data for parallel computers. *Proc. of the 2nd ACM Symp. on Parallel Algorithms and Architectures*, pages 407–416, 1990.

[KLM92] R. Karp, M. Luby, and F. Meyer auf der Heide. Efficient PRAM simulation on distributed machines. *Proc. of the 24th ACM Symp. on Theory of Comp.*, pages 318–326, 1992.

[KU88] A.R. Karlin and E. Upfal. Parallel hashing: An efficient implementation of shared memory. *J. ACM*, 35(4):876–892, 1988.

[Kuc77] D.J. Kuck. A survey of parallel machine organization and programming. *ACM Computing Surveys*, 21:339–374, 1977.

[LPP88] F. Luccio, A. Pietracaprina, and G. Pucci. A probabilistic simulation of PRAMs in VLSI. *Information Processing Lett.*, 28(3):141–147, 1988.

[LPP90] F. Luccio, A. Pietracaprina, and G. Pucci. A new scheme for the deterministic simulation of PRAMs in VLSI. *Algorithmica*, 5:529–544, 1990.

[Mey92] F. Meyer auf der Heide. Hashing strategies for simulating shared memory on distributed memory machines. *Proc. 1st Heinz Nixdorf Symp. on Parallel Architectures and their Efficient Use*, 1992. to appear in LNCS.

[Mor91] M. Morgenstern. Explicit construction of natural bounded concentrators. *Proc. of the 32th IEEE Symp. on Foundations of Comp. Sc.*, pages 392–397, 1991.

[MV84] K. Mehlhorn and U. Vishkin. Randomized and deterministic simulations of PRAMs by parallel machines with restricted granularity of parallel memories. *Acta Informatica*, 9(1):29–59, 1984.

[PP93] A. Pietracaprina and F.P. Preparata. An $O(\sqrt{n})$-worst-case-time solution to the granularity problem. *Proc. of the 10th Symp. on Theoretical Aspects of Comp. Sc.*, LNCS 665:110–119, 1993.

[Ran91] A.G. Ranade. How to emulate shared memory. *J. on Computers and System Sci.*, 42:307–326, 1991.

[Tho79] R.H. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Databases Systems*, 4(2):180–209, 1979.

[UW87] E. Upfal and A. Widgerson. How to share memory in a distributed system. *J. ACM*, 34(1):116–127, 1987.