Theory Comput. Systems **30**, 3–37 (1997)

Theory of Computing Systems

Practical Constructive Schemes for Deterministic Shared-Memory Access*

A. Pietracaprina[1] and F. P. Preparata[2]

[1]Dipartimento di Matematica Pura e Applicata, Università di Padova, I35131 Padova, Italy

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

Abstract. We present three explicit schemes for distributing M variables among N memory modules, where $M = \Theta(N^{1.5})$, $M = \Theta(N^2)$, and $M = \Theta(N^3)$, respectively. Each variable is replicated into a constant number of copies stored in distinct modules. We show that N processors, directly accessing the memories through a complete interconnection, can read/write any set of N variables in worst-case time $O(N^{1/3})$, $O(N^{1/2})$, and $O(N^{2/3})$, respectively for the three schemes. The access times for the last two schemes are optimal with respect to the particular redundancy values used by such schemes. The address computation can be carried out efficiently by each processor without recourse to a complete memory map and requiring only $O(1)$ internal storage.

1. Introduction

Consider a parallel system with N processors and N memory modules collectively storing $M \geq N$ *variables* that are available for access by the processors. A scheme is sought to distribute the variables among the modules so that any set of N variables can be efficiently accessed by the processors in parallel. This problem, originally referred to as the *granularity problem*, naturally arises in the design and implementation of parallel

* This paper was partially supported by NFS Grants CCR-91-96152 and CCR-94-00232, by ONR Contract N00014-91-J-4052, ARPA Order 8225, and by the ESPRIT III Basic Research Programme of the EC under Contract No. 9072 (Project GEPPCOM). Results reported here were presented in preliminary form at the 10th Symposium on Theoretical Aspects of Computer Science (Würzburg, Germany, 1993), and at the 5th ACM Symposium on Parallel Algorithms and Architectures (Velen, Germany, 1993).

systems (such as PRAMs and parallel databases) and has received considerable attention in the literature. An early survey [11] quotes 14 papers 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 more feasible machines.

Often, the problem is studied on a synchronous system where each processor is directly connected to all the memories, and each memory module is able to fulfill at most one access request (read/write) per time unit (*Module Parallel Computer* (MPC)) [13]. Thus, the time needed to access a set of variables is mainly determined by the maximum number of requests that a single module must fulfill. With this modeling, one can focus on reducing memory congestion without dealing with routing problems, which arise when processors and memories are connected through a *Bounded-Degree Network* (BDN).

A number of efficient randomized schemes have been developed for both the MPC and BDNs, based on the use of universal classes of hash functions to distribute the variables among the modules. It has been shown that N variables can be accessed in $O(\log N)$ time on a BDN [17], and in sublogarithmic time on an MPC [4], with high probability. On the other hand, the development of efficient deterministic schemes, which is the focus of this paper, appears to be much harder. The pioneering work of Mehlhorn and Vishkin [13] introduced the idea of representing each variable by several copies so that a read operation needs to 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 $M \in O(N^r)$, they present a memory organization scheme for the MPC that uses r copies per variable and allows a set of N read requests to be satisfied in time $O(rN^{1-1/r})$. However, this use of the copies penalizes the execution of write operations where all the copies of the variables must be accessed, thus requiring $O(rN)$ time in the worst case.

Later, Upfal and Widgerson [19] proposed a more balanced use of multiple copies exploiting the majority concept previously adopted for databases [5], [18]. Each variable is represented by r copies, where r is called the *redundancy* of the scheme. Each copy contains the value of the variable and a timestamp indicating the last time that particular copy has been accessed. A read/write operation needs to access only a majority $\lfloor r/2 \rfloor + 1$ 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 governed by a bipartite graph $G = (V, U; E)$, where V denotes the set of variables, U the set of modules, and r edges connect each variable to the modules that store its copies. For M polynomial in N and $r \in \Theta(\log N)$, Upfal and Widgerson show that there exists a graph G, with suitable expansion, which allows the MPC processors to access any N variables in $O(\log N(\log\log N)^2)$ worst-case time. They do not provide an explicit construction for G but show that a random graph exhibits the desired property, with high probability. The access time was later improved to $O(\log N)$ in [1].

Subsequently, several authors have adopted a similar framework to devise schemes for specific BDNs, trading the advantage of using more practical interconnections with a sublogarithmic increase in the access time [7], [12], [8], [9]. It has to be noted that all these schemes, as well as those for the MPC, aim at a fast access time and, for this purpose, need logarithmic redundancy. In [2] it is shown how to reduce the amount of global redundancy to a constant by using a suitable coding of the PRAM memory, at the expense of a more involved access protocol. In [16], instead, the redundancy is regarded

as a parameter and a class of schemes is devised, generalizing the one of [19]. The access time associated with each scheme is expressed as a function of the redundancy, showing a close relationship between these two quantities.

All the schemes presented in the aforementioned papers rely on expanding graphs for which no efficient implementation is known, other than resorting to a random graph. This represents the basic shortcoming (maybe fatal from the practical standpoint) of this class of approaches, for the following reasons. No efficient way is known of testing the expansion property of a random graph. As pointed out in [15], the only known technique, based on the second eigenvalue of a certain matrix related to the adjacency matrix, 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 nontrivial memory organizations. Furthermore, 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 due to memory blow-up. On the other hand, the approach proposed in [8], where the memory map is distributed among the processors with only polylogarithmic memory blow-up, has a rather involved implementation, which makes it less attractive for practical applications.

In this paper we present three schemes to distribute $\Theta(N^{1.5})$, $\Theta(N^2)$, and $\Theta(N^3)$ variables, respectively, among the N modules of the MPC. The schemes are presented in a framework similar to the one of [19]; however, the graphs used for the distribution of the variables are given explicitly and the redundancy, in all three cases, is a small constant. To read/write any given set of N variables, a simple access protocol is provided, and its worst-case performance is analyzed exploiting the expansion properties of the graphs. The results are summarized in Table 1, where for each scheme, identified by the number of variables M, we indicate the redundancy r, the time to satisfy N worst-case data requests, and the storage required in each processor to represent the memory map.

The important feature of these schemes is represented by the construction and implementation of the bipartite graphs governing the variable distribution, and the analysis of their expansion properties. The graphs are constructed by associating the two node sets, V and U, with certain quotients of $PGL_2(q^n)$ (the group of nonsingular 2×2 matrices over the field $\mathbb{F}_{q^n}$, modulo its center), or with suitably chosen subsets of them. The edges are then defined between cosets with nonempty intersection. This technique was introduced in [14] for the construction of bounded concentrators, which are bipartite graphs whose node sets have almost equal size. It has the advantage of providing the graphs with a rich algebraic structure and a remarkable "isotropy" which make them very attractive for a number of applications. We exploit such structures to determine their expansion properties and to devise an efficient implementation.

Table 1. List of results.

M	r	Time	Storage/Processor
$\Theta(N^{1.5})$	3	$O(N^{1/3})$	$O(1)$
$\Theta(N^2)$	3	$O(N^{1/2})$	$O(1)$
$\Theta(N^3)$	5	$O(N^{2/3})$	$O(1)$

The relevance of our schemes is twofold: (1) They are the first constructive approaches known to achieve *sublinear worst-case access time* for both read and write operations. (2) Their implementation is simple and involves only elementary algebra; in particular, a processor can efficiently determine the physical location of any copy with a limited use of resources. Although the time performance appears less attractive than that of the nonconstructive schemes cited before, it must be pointed out that this is essentially caused by the use of constant redundancy (which is desirable from a practical standpoint). In [16] it shown that, by using a *fixed* number r of copies per variables, the time complexity of any memory organization scheme is at least $\Omega(\min\{N/r, (M/N)^{1/(\lfloor r/2 \rfloor+1)}\})$, for $M \in \Omega(N^{1+\varepsilon})$. This result proves that our schemes for $M = \Theta(N^2)$ and $M = \Theta(N^3)$ are optimal, with respect to the specific values of the redundancy that they use.

The rest of the paper is organized as follows. In Section 2, we describe the basic structure of our schemes and present the access protocol used to satisfy a set of variable requests. The protocol's running time is given in terms of the expansion property of the graph governing the variable distribution. Section 3 introduces most of the notations and background facts concerning finite fields and the group $PGL_2(q^n)$, which are used throughout the rest of the paper. Sections 4 and 5 present the three schemes by defining the underlying graphs and studying their structural properties. In particular, for each graph we determine its expansion and provide a suitable representation that allows a processor to calculate the addresses of the copies of any given variable efficiently. For convenience, a number of technical facts, needed for the implementation of the first scheme, are reported in the Appendix.

2. Framework

As mentioned in the introduction, the parallel model used for our memory organization schemes is the MPC, consisting of N processors and N memory modules fully interconnected. (Equivalently, one may think of each module as being assigned to a distinct processor, and of each processor as being directly connected to every other processor.) In one MPC step, each processor can send one read/write request to any module, and each module satisfies one request arbitrarily selected among the incoming ones (if any). Thus, in order to guarantee efficient parallel access to a set of variables, we must ensure that the variables are well spread among the modules.

In order to distribute $M > N$ shared variables among the N modules, we adopt the standard approach originally proposed in [19], based on a *Memory Organization Scheme* (MOS) structured as follows. Each variable is replicated into r copies, r odd, stored in distinct modules, only a majority $\lfloor r/2 \rfloor + 1$ of which need to be accessed to perform a read/write operation. Each copy is provided with a timestamp which is updated every time the copy is written, so that a majority of the copies is always guaranteed to contain at least one most recently updated copy. The distribution of the copies of the variables among the modules is governed by a bipartite graph $G = (V, U; E)$, where V represents the set of variables, U the set of modules, and r edges connect each variable to the modules that store its copies.

Suppose each processor issues an access (read/write) request for a distinct variable. (The case of fewer processors issuing requests can be handled with minor modifications, obtaining the same access time, where N is replaced by the actual number of requests.

```
begin PROTOCOL
  for k := 1 to r do { Phase k }
    foreach 1 ≤ i ≤ N/r do in parallel
      P(i,k) broadcasts v(i,k) to the other processors in its cluster;
      foreach 1 ≤ j ≤ r do in parallel
        P(i,j) determines the address of the jth copy of v(i,k);
      set flag(i,k) 'alive';
      while flag(i,k) = 'alive' do
        foreach 1 ≤ j ≤ r do in parallel
          P(i,j) sends a request for the jth copy of v(i,k), if not yet accessed;
        { Each memory module accepts one request (if any) }
        count the number of copies of v(i,k) accessed so far;
        if count ≥ ⌊r/2⌋ + 1 then set flag(i,k) = 'dead'
end PROTOCOL.
```

Fig. 1. Access protocol.

The case of multiple requests for the same variable also requires minor changes, and introduces only an additive logarithmic factor in the access time.) In order to satisfy the requests, the following protocol is executed by the processors, in parallel. The N processors are subdivided into N/r *clusters*, with r processors per cluster. Let $P(i, j)$ denote the jth processor in cluster i, and let $v(i, j)$ denote the variable it wants to access, for $1 \leq i \leq N/r$ and $1 \leq j \leq r$. The protocol consists of r phases. In Phase k 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 every processor tries to access its assigned copy unless it previously succeeded, or other $\lfloor r/2 \rfloor + 1$ copies of the same variable have already been accessed. Since a memory module can satisfy at most one request per iteration, the number of copies accessed in one iteration is equal to the number of modules receiving requests in that iteration. At any point during the execution of a given phase, a copy is said to be *alive* if it has not been accessed yet; a variable is said to be *alive* if fewer than $\lfloor r/2 \rfloor + 1$ of its copies have been accessed, which implies, since r is odd, that at least $\lfloor r/2 \rfloor + 1$ of its copies are still alive. (This terminology is used only for variables and copies requested in the phase under consideration.) The code for the entire protocol is shown in Figure 1. For each variable, a flag is used to indicate whether the variable is alive or not.

As we will see later, all the schemes presented in this paper use constant redundancy, and their implementations allows each processor to determine the physical address of any copy in $O(\log N)$ time. Thus, letting Φ denote the maximum number of iterations of the **while** loop, executed in any of the r phases, it can easily be seen that the entire access protocol takes $O(\Phi + \log N)$ steps on the MPC. Next, we devise a general expression for Φ based on the expansion properties of the graph G.

Let $S \subset V$ be a set of variables. A *c-bundle* for S is defined as a subset of copies of variables in S containing at least c copies for each variable (similar terminology is used in [9]). For a c-bundle η of S, let $\Gamma_\eta(S)$ denote the set of modules storing the copies in η.

Definition 1. G has (ν, μ)-*expansion* if, for any $S \subset V$ and any $(\lfloor r/2 \rfloor + 1)$-bundle η,

$$|\Gamma_\eta(S)| \geq \mu |S|^{1-\nu}.$$

The following lemma is similar to Lemma 3.3 of [19].

Lemma 1. *Consider k consecutive iterations of the* **while** *loop in a phase and suppose that at the beginning of this set of iterations there are X live copies. Let R_k be the number of live copies remaining after the last iteration. If G has (ν, μ)-expansion, then*

$$R_k \leq X\left(1 - \frac{\alpha}{X^\nu}\right)^k,$$

with $\alpha = \mu/r^{1-\nu}$.

Proof. The proof is by induction on k. Let $k = 1$. At the beginning there are X live copies, which means that there are at least X/r live variables. By using the expansion property of G, we conclude that the live copies reside in at least $\mu(X/r)^{1-\nu}$ modules, and, therefore, after the first iteration

$$R_1 \leq X - \mu\left(\frac{X}{r}\right)^{1-\nu} = X\left(1 - \frac{\alpha}{X^\nu}\right)$$

copies are still alive. This establishes the basis. Assuming that the lemma holds for $k-1$, by a similar reasoning, we can show that

$$\begin{aligned} R_k &\leq R_{k-1} - \mu\left(\frac{R_{k-1}}{r}\right)^{1-\nu} = R_{k-1}\left(1 - \frac{\alpha}{R_{k-1}^\nu}\right) \\ &\leq X\left(1 - \frac{\alpha}{X^\nu}\right)^{k-1}\left(1 - \frac{\alpha}{R_{k-1}^\nu}\right). \end{aligned}$$

Since $R_{k-1} \leq X$, we have that $(1 - \alpha/R_{k-1}^\nu) \leq (1 - \alpha/X^\nu)$, which concludes the proof. □

Theorem 1. *Suppose that G has (ν, μ)-expansion, for some constant ν, $0 < \nu < 1$, and that $\alpha = \mu/r^{1-\nu} \in \Theta(1)$. Then $\Phi \in O(N^\nu)$ iterations of the* **while** *loop are sufficient to access the N/r variables requested in any phase. Therefore, the access protocol requires a total of $O(N^\nu)$ steps on the MPC.*

Proof. Consider a phase of the protocol where N/r variables are accessed, one per cluster (for a total of N copies), and let Φ be the number of iterations in this phase. Let k_i be the number of iterations used to reduce the number of live copies from R_i to $R_{i+1} \leq R_i/e^\alpha$, with $R_0 = N$. Then $\Phi = \sum_{i=0}^{J-1} k_i$, with $J = (1/\alpha)\log_e N = O(\log N)$. Choosing R_i^ν as an upper bound to k_i and applying Lemma 1, we have

$$R_{i+1} \leq R_i\left(1 - \frac{\alpha}{R_i^\nu}\right)^{k_i} \leq R_i\left(1 - \frac{\alpha}{R_i^\nu}\right)^{R_i^\nu} < R_i e^{-\alpha}.$$

Hence,

$$\Phi \leq \sum_{i=0}^{J-1} R_i^\nu \leq N^\nu \sum_{i=0}^{J-1} e^{-\alpha i\nu} = O(N^\nu).$$

□

The analysis of the MOSs presented in later sections is aimed at determining the expansion properties of the specific graphs used in such schemes, so that the above theorem can be used to obtain their time performance.

3. Definitions and Notations

Let q be a prime power and let n be an integer. Let $\mathbb{F}_{q^n}$ denote the finite field with q^n elements, let $\mathbb{F}_{q^n}^*$ be its multiplicative group, and let γ be a primitive element of $\mathbb{F}_{q^n}$. As is well known from Galois theory, $\langle\gamma\rangle = \mathbb{F}_{q^n}^*$, where $\langle\gamma\rangle$ is the cyclic group generated by γ, and, furthermore, the elements of $\mathbb{F}_{q^n}$ can be represented as polynomials in γ of degree less than n, with coefficients in $\mathbb{F}_q$. Throughout this paper, unless differently specified, lower-case roman letters are used to denote the elements of $\mathbb{F}_q$, and lower-case greek letters to denote those of $\mathbb{F}_{q^n}$.

The *Projective Linear Group of degree* 2 *over* $\mathbb{F}_{q^n}$ ($PGL_2(q^n)$) is the group (under matrix multiplication) of 2×2 nonsingular matrices with entries in $\mathbb{F}_{q^n}$, modulo its center, the group of scalar matrices (i.e., scalar multiples of the identity) [6]. In other words, matrices that differ by a scalar multiple represent the same group element. It is well known that

$$|PGL_2(q^n)| = (q^n + 1)q^n(q^n - 1). \tag{1}$$

A matrix of $PGL_2(q^n)$ will usually be written either as $\begin{bmatrix} \alpha & \beta \\ \delta & 1 \end{bmatrix}$ or $\begin{bmatrix} \alpha & \beta \\ 1 & 0 \end{bmatrix}$, with $\alpha, \beta, \delta \in \mathbb{F}_{q^n}$, except for few cases when a different notation is more convenient. The following sets, which can be easily seen to be subgroups of $PGL_2(q^n)$, play an important role in the definition of our MOSs. Define

$$H_0 = PGL_2(q),$$

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

$$H_n = \left\{ \begin{bmatrix} \alpha & \beta \\ 0 & 1 \end{bmatrix} : \ \alpha, \beta \in \mathbb{F}_{q^n} \text{ and } \alpha \neq 0 \right\}.$$

It is easy to see that

$$|H_0| = (q + 1)q(q - 1), \tag{2}$$

$$|H_{n-1}| = q^n(q - 1), \tag{3}$$

$$|H_n| = q^n(q^n - 1). \tag{4}$$

Finally, we need to distinguish two particular subsets of $\mathbb{F}_{q^n}$. One is the set P_γ of all the polynomials in γ with constant term equal to 0, that is, $P_\gamma = \{\sum_{i=1}^{n-1} c_i \gamma^i : \ c_i \in \mathbb{F}_q\}$. Note that $|P_\gamma| = q^{n-1}$, and we denote these polynomials as

$$\{p_i : \ 0 \leq i < q^{n-1}\}.$$

The other set is that of monic polynomials in γ of degree less than n. Note that there are $(q^n - 1)/(q - 1)$ such polynomials, which we denote as

$$\left\{\pi_i\colon\ 0 \le i < \frac{q^n - 1}{q - 1}\right\}.$$

4. Memory Organization Scheme for $M \in \Theta(N^{1.5})$ Variables

4.1. *The Graph*

Let q be a prime power and let $n \ge 3$ be an integer, such that either q is even, or both q and n are odd. The graph $G = (V, U; E)$ that specifies how the copies of the variables are distributed among the memory modules is defined as follows:

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

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

Thus, the variables are associated with the left cosets of H_0 and the modules with the left cosets of H_{n-1}. By (1), (2), and (3), it follows that

$$M = |V| = q^{n-1}\frac{q^{2n} - 1}{q^2 - 1},$$

$$N = |U| = \frac{q^{2n} - 1}{q - 1}.$$

Therefore, for fixed q, we have $M \in \Theta(N^{1.5})$. The edge set is defined as follows:

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

We now show that the edges are in one-to-one correspondence with the cosets of the subgroup $H_0 \cap H_{n-1}$. It is easily seen that

$$H_0 \cap H_{n-1} = \left\{\begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix}\colon\ a, b \in \mathbb{F}_q,\ a \ne 0\right\}, \tag{5}$$

thus, $|H_0 \cap H_{n-1}| = q(q - 1)$. By (2) and (3), we get

$$|H_0/(H_0 \cap H_{n-1})| = q + 1,$$

$$|H_{n-1}/(H_0 \cap H_{n-1})| = q^{n-1}.$$

In the following two lemmas, we determine two sets of matrices representative of the cosets of $H_0/(H_0 \cap H_{n-1})$ and $H_{n-1}/(H_0 \cap H_{n-1})$, respectively, which allow us to characterize the edges of the graph explicitly. We use $\Gamma(x)$ to denote the neighbors of a node x in the graph.

Lemma 2. *Let* $\mathbb{F}_q = \{a_0, \ldots, a_{q-1}\}$ *and define*

$$L_{-1}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$L_k^0 = \begin{bmatrix} a_k & 1 \\ 1 & 0 \end{bmatrix}, \qquad 0 \le k < q.$$

Then

$$H_0 = \bigcup_{k=-1}^{q-1} L_k^0 (H_0 \cap H_{n-1}),$$

and, for any $A \in PGL_2(q^n)$,

$$\Gamma(AH_0) = \{AL_k^0 H_{n-1} \colon \ -1 \le k < q\}.$$

Proof. We first prove that

$$H_0 = \bigcup_{k=-1}^{q-1} L_k^0 (H_0 \cap H_{n-1}).$$

Since $L_k^0 \in H_0$, for $-1 \le k < q$, we have that

$$H_0 \supseteq \bigcup_{k=-1}^{q-1} L_k^0 (H_0 \cap H_{n-1}).$$

The equality follows from recalling that $|H_0/(H_0 \cap H_{n-1})| = q + 1$ and observing that, for $k_1 \ne k_2$, $L_{k_1}^0 (H_0 \cap H_{n-1}) \cap L_{k_2}^0 (H_0 \cap H_{n-1}) = \emptyset$.

As for $\Gamma(AH_0)$, by definition we have

$$\Gamma(AH_0) = \{BH_{n-1} \colon \ AH_0 \cap BH_{n-1} \ne \emptyset\}.$$

Suppose $AH_0 \cap BH_{n-1} \ne \emptyset$ and let $C \in AH_0 \cap BH_{n-1}$, that is, $C \in AH_0$ and $C \in BH_{n-1}$. However, $C \in AH_0$ implies $AH_0 = CH_0$, and, analogously, we have $BH_{n-1} = CH_{n-1}$. This yields $AH_0 \cap BH_{n-1} = C(H_0 \cap H_{n-1})$. This implies that the edges incident to AH_0 are associated with the cosets of $AH_0/(H_0 \cap H_{n-1})$, and the lemma follows. □

Lemma 3. *Recall that* $P_\gamma = \{p_i \colon \ 0 \le i < q^{n-1}\}$, *and define*

$$L_h^{n-1} = \begin{bmatrix} 1 & p_h \\ 0 & 1 \end{bmatrix}, \qquad 0 \le h < q^{n-1}.$$

Then

$$H_{n-1} = \bigcup_{h=0}^{q^{n-1}-1} L_h^{n-1} (H_0 \cap H_{n-1}),$$

and, for any $B \in PGL_2(q^n)$,

$$\Gamma(BH_{n-1}) = \{BL_h^{n-1} H_0 \colon \ 0 \le h < q^{n-1}\}.$$

Proof. The proof is similar to that of Lemma 2. For the first part, since $L_h^{n-1} \in H_{n-1}$, for $0 \leq h < q^{n-1}$, we have that $H_{n-1} \supseteq \bigcup_{h=0}^{q^{n-1}-1} L_h^{n-1}(H_0 \cap H_{n-1})$. The equality follows by noting that $|H_{n-1}/(H_0 \cap H_{n-1})| = q^{n-1}$ and that, for $h_1 \neq h_2$, $L_{h_1}^{n-1}(H_0 \cap H_{n-1}) \cap L_{h_2}^{n-1}(H_0 \cap H_{n-1}) = \emptyset$.

As for the second part, by definition we have that

$$\Gamma(BH_{n-1}) = \{AH_0\colon\ AH_0 \cap BH_{n-1} \neq \emptyset\}.$$

If $AH_0 \cap BH_{n-1} \neq \emptyset$, then there exists a C such that $AH_0 \cap BH_{n-1} = C(H_0 \cap H_{n-1})$. Therefore, distinct edges incident to BH_{n-1} correspond to distinct cosets in $BH_{n-1}/(H_0 \cap H_{n-1})$. □

Thus, $|\Gamma(AH_0)| = q + 1$ and $|\Gamma(BH_{n-1})| = q^{n-1}$, which means that there are $r = q + 1$ copies of each variable, stored in distinct modules, and that every module stores q^{n-1} copies of distinct variables.

Before proceeding with the analysis of the structure of the graph, we need a convenient representation for the nodes of U, given in the following lemma.

Lemma 4.

$$U = \left\{ \begin{bmatrix} \gamma^i & 0 \\ 0 & 1 \end{bmatrix} H_{n-1}\colon\ 0 \leq i < \frac{q^n - 1}{q - 1} \right\} \cup \left\{ \begin{bmatrix} \alpha & \gamma^i \\ 1 & 0 \end{bmatrix} H_{n-1}\colon\ \alpha \in \mathbb{F}_{q^n},\ 0 \leq i < \frac{q^n - 1}{q - 1} \right\}.$$

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

The theorem below shows that the copies of any two variables share at most one memory module.

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 Figure 2). Without loss of generality, we assume that $CH_{n-1} = H_{n-1}$, otherwise an analogous situation could be obtained by premultiplying A, B, C, and D by C^{-1}. Thus, Lemma 3 allows us to choose

$$A = \begin{bmatrix} 1 & p_i \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & p_j \\ 0 & 1 \end{bmatrix}, \qquad \text{for some} \quad p_i, p_j \in P_\gamma, \quad i \neq j.$$

It is easy to see that $AL_{-1}^0 H_{n-1} = BL_{-1}^0 H_{n-1} = H_{n-1}$ and, by Lemma 2, we have

$$\begin{aligned} \Gamma(AH_0) - H_{n-1} &= \Gamma(AH_0) - AL_{-1}^0 H_{n-1} \\ &= \left\{ \begin{bmatrix} 1 & p_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_k & 1 \\ 1 & 0 \end{bmatrix} H_{n-1}\colon\ a_k \in \mathbb{F}_q \right\} \\ &= \left\{ \begin{bmatrix} p_i + a_k & 1 \\ 1 & 0 \end{bmatrix} H_{n-1}\colon\ a_k \in \mathbb{F}_q \right\} \end{aligned}$$

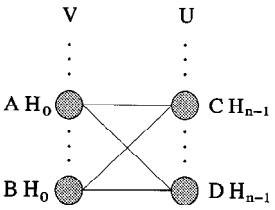


Fig. 2. AH_0, BH_0, CH_{n-1}, and DH_{n-1}.

and, similarly,

$$\Gamma(BH_0) - H_{n-1} = \left\{ \begin{bmatrix} p_j + a_k & 1 \\ 1 & 0 \end{bmatrix} H_{n-1} \colon\ a_k \in \mathbb{F}_q \right\}.$$

Since $i \neq j$, the definition of P_γ implies that $p_i + a_k \neq p_j + a_h$, for any $0 \leq k, h < q$, and, by Lemma 4 we conclude that $(\Gamma(AH_0) - H_{n-1}) \cap (\Gamma(BH_0) - H_{n-1}) = \emptyset$. Therefore, $|(\Gamma(AH_0) - H_{n-1}) \cap (\Gamma(BH_0) - H_{n-1})| = |\Gamma(AH_0) \cap \Gamma(BH_0) - H_{n-1}| = 0$, which contradicts the assumption that $\Gamma(AH_0)$ and $\Gamma(BH_0)$ share more than one element. □

In order to determine the expansion property of the graph, we need to introduce the function Γ^2, defined as

$$\Gamma^2(u) = \Gamma(\Gamma(u)) - u, \qquad \forall u \in U. \tag{6}$$

We have

Lemma 5.

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

Proof. By definition,

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

By Lemma 3,

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

By Lemma 2,

$$\begin{aligned} \Gamma\left(A \begin{bmatrix} 1 & p_h \\ 0 & 1 \end{bmatrix} H_0 \right) &= \left\{ A \begin{bmatrix} 1 & p_h \\ 0 & 1 \end{bmatrix} H_{n-1} \right\} \cup \left\{ A \begin{bmatrix} p_h + a_k & 1 \\ 1 & 0 \end{bmatrix} H_{n-1} \colon\ a_k \in \mathbb{F}_q \right\} \\ &= \{AH_{n-1}\} \cup \left\{ A \begin{bmatrix} p_h + a_k & 1 \\ 1 & 0 \end{bmatrix} H_{n-1} \colon\ a_k \in \mathbb{F}_q \right\}. \end{aligned}$$

Hence,

$$\Gamma^2(AH_{n-1}) = \left\{ A \begin{bmatrix} p_h + a_k & 1 \\ 1 & 0 \end{bmatrix} H_{n-1} \colon\ p_h \in P_\gamma \text{ and } a_k \in \mathbb{F}_q \right\}.$$

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

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 $AH_{n-1} = H_{n-1}$ (the same argument as in the proof of Theorem 2). By Lemma 5,

$$\Gamma^2(H_{n-1}) = \left\{ \begin{bmatrix} \delta & 1 \\ 1 & 0 \end{bmatrix} H_{n-1} \colon \delta \in \mathbb{F}_{q^n} \right\}.$$

Then we have three exhaustive cases, depending upon the form of B (refer to Lemma 4).

Case 1: $B = \begin{bmatrix} \gamma^i & 0 \\ 0 & 1 \end{bmatrix}$, *for some* i, $1 \leq i < (q^n - 1)/(q - 1)$. Applying Lemma 5 we get

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

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

Case 2: $B = \begin{bmatrix} \beta & 1 \\ 1 & 0 \end{bmatrix}$, *for some* $\beta \in \mathbb{F}_{q^n}$. Applying Lemma 5 again, we get

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

Now, if $\alpha = 0$, then

$$\begin{bmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{bmatrix} H_{n-1} = H_{n-1} \notin \Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1}).$$

Otherwise, let $-\alpha^{-2} = b\gamma^k$, for some k, $0 \leq k < (q^n - 1)/(q - 1)$, and $b \in \mathbb{F}_q^*$. (It is easy to see that any element of $\mathbb{F}_{q^n}^*$ can be written as $b\gamma^k$, for suitable k, $0 \leq k < (q^n - 1)/(q - 1)$, and $b \in \mathbb{F}_q^*$.) Simple calculations show that

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

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

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

Note that $\{-\alpha^{-2}\colon\ \alpha \in \mathbb{F}_{q^n}^*\} = \{\gamma^{2i}\colon\ 0 \le i \le q^n - 2\}$. If q is even, or both q and n are odd, 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

$$\begin{bmatrix} \beta\alpha + 1 & \beta \\ \alpha & 1 \end{bmatrix} H_{n-1} \in \Gamma^2(H_{n-1}).$$

Thus $|\Gamma^2(H_{n-1}) \cap \Gamma^2(BH_{n-1})| = q - 1$.

Case 3: $BH_{n-1} = \begin{bmatrix} \beta & \gamma^i \\ 1 & 0 \end{bmatrix} H_{n-1}$, *for some* $\beta \in \mathbb{F}_{q^n}$ *and* i, $1 \le i < (q^n - 1)/(q-1)$. 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. □

Theorems 2 and 3 have two important consequences, stated below as corollaries. Consider a set of variables $S \subseteq V$, and let η be a $(\lfloor (q+1)/2 \rfloor + 1)$-bundle for S. The copies included in η are referred to as *bundle copies*.

Corollary 1. *Let $u \in U$ and let $v_1, \ldots, v_t$ be t distinct variables in S, such that each v_i has a bundle copy stored in u. Then*

$$|\Gamma_\eta(\{v_1, \ldots, v_t\}) - \{u\}| \ge \left\lfloor \frac{q+1}{2} \right\rfloor t.$$

Proof. Immediate from Theorem 2. □

Corollary 2. *Consider a set of k distinct modules $u_1, \ldots, u_k$. Let t_i be the number of variables of S that have a bundle copy stored in module u_i. Then*

$$|\Gamma_\eta(S)| \ge \sum_{i=1}^{k} \left\lfloor \frac{q+1}{2} \right\rfloor t_i - \binom{k}{2}(q-1).$$

Proof. For $1 \le i \le k$, let $\mathcal{A}_i \subset S$ denote the set of variables that store a bundle copy in u_i. By hypothesis, $|\mathcal{A}_i| = t_i$. Let also $\mathcal{B}_i = \Gamma_\eta(\mathcal{A}_i) - \{u_i\}$, for $1 \le i \le k$. Clearly, $\bigcup_{i=1}^{k} \mathcal{B}_i \subseteq \Gamma_\eta(S)$, and, by the inclusion–exclusion principle, we have

$$\begin{aligned} |\Gamma_\eta(S)| &\ge \sum_{i=1}^{k} |\Gamma_\eta(\mathcal{B}_i)| - \sum_{1 \le i < j \le k} |\Gamma_\eta(\mathcal{B}_i) \cap \Gamma_\eta(\mathcal{B}_j)| \\ &\overset{\text{by Cor. 1}}{\ge} \sum_{i=1}^{k} \left\lfloor \frac{q+1}{2} \right\rfloor t_i - \sum_{1 \le i < j \le k} |\Gamma^2(u_i) \cap \Gamma^2(u_j)| \\ &\overset{\text{by Th. 3}}{\ge} \sum_{i=1}^{k} \left\lfloor \frac{q+1}{2} \right\rfloor t_i - \binom{k}{2}(q-1). \end{aligned}$$

□

We are now ready to determine the expansion property of G. In particular, we want to find the values ν and μ for which G has (ν, μ)-expansion (see Definition 1).

Theorem 4. *G has $(\frac{1}{3}, q/2^{5/3})$-expansion.*

Proof. Let $\rho = \lfloor (q+1)/2 \rfloor$. We must prove that, for any set $S \subset V$ and for any $(\rho+1)$-bundle η for S,

$$|\Gamma_\eta(S)| > \frac{q}{2^{5/3}}|S|^{2/3}.$$

We first prove that $|\Gamma_\eta(S)| > \sqrt{|S|}\rho$. Note that there is a total of at least $|S|(\rho+1)$ bundle copies. Suppose, for a contradiction, that $|\Gamma_\eta(S)| \leq \sqrt{|S|}\rho$. Then there exists a node in U storing at least $|S|(\rho+1)/\sqrt{|S|}\rho > \sqrt{|S|}$ copies in η. Since a module stores copies of distinct variables, by Corollary 1 this implies $|\Gamma_\eta(S)| > \sqrt{|S|}\rho$, which contradicts the assumption $|\Gamma_\eta(S)| \leq \sqrt{|S|}\rho$.

Now, let $|\Gamma_\eta(S)| = \sqrt{|S|}\varepsilon$, with $\varepsilon > \rho$. We claim that each module in $\Gamma_\eta(S)$ stores fewer than $\sqrt{|S|}\varepsilon/\rho$ bundle copies. Indeed, suppose there is some $u \in \Gamma_\eta(S)$ storing $x \geq \sqrt{|S|}\varepsilon/\rho$ bundle copies. Using again Corollary 1, we conclude that $|\Gamma_\eta(S) - u| \geq \sqrt{|S|}\varepsilon$, that is $|\Gamma_\eta(S)| \geq \sqrt{|S|}\varepsilon + 1$, a contradiction. Let t be the number of modules in $\Gamma_\eta(S)$ storing at least $\sqrt{|S|}\rho/\varepsilon$ bundle copies. We majorize the number of bundle copies in these t modules by $\sqrt{|S|}\varepsilon/\rho$, and the number of bundle copies in the remaining $(\sqrt{|S|}\varepsilon - t)$ modules by $\sqrt{|S|}\rho/\varepsilon$, and obtain the following inequality:

$$t\sqrt{|S|}\frac{\varepsilon}{\rho} + (\sqrt{|S|}\varepsilon - t)\sqrt{|S|}\frac{\rho}{\varepsilon} \geq |S|(\rho+1).$$

This implies $t > \sqrt{|S|}(\rho\varepsilon/(\varepsilon^2 - \rho^2)) > \sqrt{|S|}\rho/\varepsilon$, since $\varepsilon > \rho$.

Let $k = \frac{1}{2}\sqrt{|S|}\rho/\varepsilon < t$. Consider k nodes of $\Gamma_\eta(S)$ storing at least $\sqrt{|S|}\rho/\varepsilon$ bundle copies. By Corollary 2 we have

$$\begin{aligned}
|\Gamma_\eta(S)| &\geq k\rho\sqrt{|S|}\frac{\rho}{\varepsilon} - \binom{k}{2}(q-1) \\
&> k\rho\sqrt{|S|}\frac{\rho}{\varepsilon} - \frac{q-1}{2}k^2 \\
&= \frac{\rho}{2}|S|\left(\frac{\rho}{\varepsilon}\right)^2 - \frac{q-1}{8}|S|\left(\frac{\rho}{\varepsilon}\right)^2 \\
&= \left(\frac{\rho}{2} - \frac{q-1}{8}\right)|S|\left(\frac{\rho}{\varepsilon}\right)^2 \\
&> \frac{q^3}{32\varepsilon^2}|S|.
\end{aligned}$$

Combining the hypothesis $|\Gamma_\eta(S)| = \sqrt{|S|}\varepsilon$ with the above inequality we obtain

$$|\Gamma_\eta(S)| > \frac{1}{2^{5/3}}|S|^{2/3}q. \qquad \square$$

Note that the above theorem holds for any prime power q. In particular we can choose $q = 2$. Assuming that the graph G can be implemented in such a way that a processor is able to compute the physical address of any copy in $O(\log N)$ time using constant internal storage, which we prove in the next section, we can combine the results of Theorems 4 and 1 and get

Theorem 5. *$M \in O(N^{1.5})$ variables can be distributed among N processors of an MPC, with redundancy 3, so that any set of N distinct variables can be accessed, using the access protocol of Section 2, in time $O(N^{1/3})$. Moreover, each processor can determine the physical address of any copy in $O(\log N)$ time using $O(1)$ internal storage.*

4.2. *Implementation*

A crucial aspect of the design of an MOS concerns its implementation, an issue that has often been ignored in the past. In particular, a processor that wants to access a specific copy of a variable must be able to determine efficiently the module storing that copy and the physical address of the copy within the module. This subsection explains how this can be accomplished when the variables are distributed among the modules according to the graph G presented in the preceding subsection.

Let $v_0, \ldots, v_{M-1}$ denote the variables and $u_0, \ldots, u_{N-1}$ the memory modules. Recall that $M = q^{n-1}((q^{2n}-1)/(q^2-1))$ and $N = (q^{2n}-1)/(q-1)$. We first need to associate variables and modules with the appropriate cosets. That is, we need to establish the following bijections:

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

The definition of the A_i's involves a number of technical details, which, for convenience of presentation, are dealt with in the Appendix. As for the B_j's, Lemma 4 already suggests a set of possible candidates, which, however, we need to modify slightly as follows. For nonnegative integers $s < (q^n-1)/(q-1)$ and $t < q^n + 1$, we define the integer

$$J(s,t) \triangleq s(q^n+1) + t,$$

and with integer $j \in [0, N-1]$ we associate the pair (s,t) if and only if $j = J(s,t)$. Recall that $\{\pi_s\colon\ 0 \le s < (q^n-1)/(q-1)\}$ denotes the set of monic polynomials in γ of degree less than n, and let $\mathbb{F}_{q^n} = \{\alpha_0, \ldots, \alpha_{q^n-1}\}$. For $0 \le s < (q^n-1)/(q-1)$ and $0 \le t < q^n + 1$, define

$$B_{J(s,t)} = \begin{cases} \begin{bmatrix} \pi_s & 0 \\ 0 & 1 \end{bmatrix} & \text{if} \quad t = 0, \\ \begin{bmatrix} \alpha_{t-1} & \pi_s \\ 1 & 0 \end{bmatrix} & \text{otherwise.} \end{cases} \tag{7}$$

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

Proof. Since both $\{\pi_s\colon\ 0 \le s < (q^n-1)/(q-1)\}$ and $\{\gamma^{s'}\colon\ 0 \le s' < (q^n-1)/(q-1)\}$ are sets of representatives for $\mathbb{F}_{q^n}^*/\mathbb{F}_q^*$, there is a bijection between indices s and s' such that

$$\pi_s = b_s \gamma^{s'},$$

for some $b_s \in \mathbb{F}_q^*$. Since any diagonal matrix with entries in $\mathbb{F}_q^*$ belongs to H_{n-1}, it is straightforward that

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

and

$$\begin{bmatrix} \alpha_{t-1} & \pi_s \\ 1 & 0 \end{bmatrix} H_{n-1} = \begin{bmatrix} \alpha_{t-1} & \gamma^{s'} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & b_s \end{bmatrix} H_{n-1} = \begin{bmatrix} \alpha_{t-1} & \gamma^{s'} \\ 1 & 0 \end{bmatrix} H_{n-1},$$

so that, by Lemma 4, the claim holds □

Each variable has $q+1$ copies stored in distinct modules, and each module contains copies of q^{n-1} distinct variables. By Lemma 2, the copies of $v_i = A_i H_0$ are stored in modules

$$A_i L_k^0 H_{n-1}, \qquad -1 \le k < q.$$

Moreover, by Lemma 3, module $u_{J(s,t)}$ contains the copies of

$$B_{J(s,t)} L_h^{n-1} H_0, \qquad 0 \le h < q^{n-1}.$$

Also, recall that edge $(A_i H_0, A_i L_k^0 H_{n-1})$ is associated with coset $A_i L_k^0 (H_0 \cap H_{n-1})$, and, analogously, edge $(B_{J(s,t)} H_{n-1}, B_{J(s,t)} L_h^{n-1} H_0)$ with coset $B_{J(s,t)} L_h^{n-1} (H_0 \cap H_{n-1})$. We adopt the following convention: *the* kth *copy of* v_i *is stored at the address* h *in module* $u_{J(s,t)}$ *if and only if* $A_i L_k^0 (H_0 \cap H_{n-1}) = B_{J(s,t)} L_h^{n-1} (H_0 \cap H_{n-1})$, *that is, if and only if*

$$A_i L_k^0 \in B_{J(s,t)} L_h^{n-1} (H_0 \cap H_{n-1}).$$

Therefore, we seek a method to compute (s, t, h) from (i, k). To this purpose, a processor proceeds as follows:

Step 1. From i and k compute A_i and L_k^0.

Step 2. Find s, t, and h such that $A_i L_k^0 \in B_{J(s,t)} L_h^{n-1} (H_0 \cap H_{n-1})$.

The following lemma identifies the matrices in each coset $B_{J(s,t)} L_h^{n-1} (H_0 \cap H_{n-1})$.

Lemma 6. *If* $t = 0$, *then*

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

Otherwise $(t > 0)$,

$$B_{J(s,t)} L_h^{n-1} (H_0 \cap H_{n-1}) = \left\{ \begin{bmatrix} a\alpha_{t-1} & (p_h + b)\alpha_{t-1} + \pi_s \\ a & p_h + b \end{bmatrix} : \; a, b \in \mathbb{F}_q, \; a \neq 0 \right\}.$$

Proof. The lemma follows from (5), definition (7) of the $B_{J(s,t)}$'s, and the definition of the L_h^{n-1}'s given in Lemma 3. □

We are now ready to describe in more detail the two-step procedure that computes the physical address of a copy. We assume that each processor has a local work space consisting of a constant number of registers, and that, in addition to the ordinary 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. Representing the elements of $\mathbb{F}_{q^n}$ as polynomials, the operations in $\mathbb{F}_{q^n}$ can be implemented using feedback shift registers, each operation taking $O(n) = O(\log N)$

```
begin STEP 2
  if z = 0
    then
      t := 0;
      s := INDEX3(x);
      h := INDEX2(y ⊗ INV(MONIC(x)));
    else      { z ≠ 0 }
      t := INDEX1(x ⊗ INV(z))+1;
      calculate α_{t-1};
      if v = 0
        then
          s := INDEX3(y);
          h := 0
        else      { v = 1 }
          s := INDEX3((y ⊖ α_{t-1}) ⊗ INV (z));
          calculate π_s;
          h := INDEX2(INV(y ⊖ α_{t  1}) ⊗ π_s)
end STEP 2.
```

Fig. 3. Code for Step 2.

time and space equivalent to the storage of $O(1)$ field elements (see Chapter 2 of [3] for details). Each polynomial of P_γ 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 $\mathbb{F}_q$. A similar lexicographic indexing can also be established for the set of monic polynomials $\{\pi_s\colon\ 0 \le s < (q^n-1)/(q-1)\}$.

In the Appendix we show that, given i, a processor can compute A_i in time $O(\log N)$ using $O(1)$ storage (Theorem 15). Such a result is assumed in what follows. Clearly, in additional $O(1)$ time we obtain L_k^0, which completes Step 1. Consider now Step 2. Let

$$A_i L_k^0 = \begin{bmatrix} x & y \\ z & v \end{bmatrix},$$

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 Figure 3, which is entirely based on Lemma 6, computes the indices s, t, and h such that $A_i L_k^0 \in B_{J(s,t)} L_h^{n-1}(H_0 \cap H_{n-1})$, that is, $A_i L_k^0(H_0 \cap H_{n-1}) = B_{J(s,t)} L_h^{n-1}(H_0 \cap H_{n-1})$. In the code given in Figure 3, symbols $\oplus$, $\ominus$, $\otimes$, and INV denote, respectively, addition, subtraction, multiplication, and inverse in $\mathbb{F}_{q^n}$. We also use the following four macros, all executable in $O(\log N)$ time. Let x denote a generic element of $\mathbb{F}_{q^n}$.

- 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 $x = a\pi_s$.

We conclude with the following theorem, whose proof follows immediately from Lemma 6 and the above discussion.

Theorem 6. *The code in Figure* 3 *correctly executes Step* 2 *in* $O(\log N)$ *time. Therefore, a processor is able to compute the physical address of any copy of any variable in* $O(\log N)$ *time using* $O(1)$ *internal storage.*

5. Memory Organization Schemes for $M \in O(N^2)$ and $M \in O(N^3)$ Variables

We jointly present the MOS for $M \in O(N^2)$ and $M \in O(N^3)$ variables, because their underlying graphs are derived from the same graph $G = (V, U; E)$ defined below. Let

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

$$U = PGL_2(q^n)/H_n.$$

By (1), (2), and (4), we have

$$|V| = q^{n-1}\frac{q^{2n}-1}{q^2-1},$$

$$|U| = q^n + 1.$$

The edge set is

$$E = \{(AH_0, BH_n)\colon\ A, B \in PGL_2(q^n) \text{ and } AH_0 \cap BH_n \neq \emptyset\}.$$

Note that the graph is similar to the one studied in the previous section, with the difference that the subgroup H_n replaces H_{n-1} in the definition of U. The edges are in one-to-one correspondence with the cosets of $H_0 \cap H_n$, and can be characterized as follows. It is easily seen that

$$H_0 \cap H_n = H_0 \cap H_{n-1} = \left\{\begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix}\colon\ a, b \in \mathbb{F}_q,\ a \neq 0\right\}. \tag{8}$$

Therefore, we have $|H_n/(H_0 \cap H_n)| = q^{n-1}((q^n-1)/(q-1))$, and

$$H_0 = \bigcup_{k=-1}^{q-1} L_k^0(H_0 \cap H_n). \tag{9}$$

Lemma 7. *We define*

$$L_{s,t}^n = \begin{bmatrix} \pi_s & \pi_s p_t \\ 0 & 1 \end{bmatrix}, \qquad 0 \le s < \frac{q^n-1}{q-1}, \quad \textit{and} \quad 0 \le t < q^{n-1}.$$

We have

$$H_n = \bigcup_{s=0}^{(q^n-1)/(q-1)-1} \ \bigcup_{t=0}^{q^{n-1}-1} L_{s,t}^n(H_0 \cap H_n),$$

and, for any $B \in PGL_2(q^n)$,

$$\Gamma(BH_n) = \left\{ BL^n_{s,t} H_0 \colon\ 0 \le s < \frac{q^n - 1}{q - 1} \ \text{ and } \ 0 \le t < q^{n-1} \right\}.$$

Proof. Notice that, for any π_s and any p_t,

$$\begin{bmatrix} \pi_s & \pi_s p_t \\ 0 & 1 \end{bmatrix} \in H_n,$$

therefore

$$H_n \supseteq \bigcup_{s=0}^{(q^n-1)/(q-1)-1} \ \bigcup_{t=0}^{q^{n-1}-1} L^n_{s,t}(H_0 \cap H_n).$$

To prove equality (on the basis of cardinalities), we must prove that the cosets of the right-hand set are distinct. Suppose

$$L^n_{s_1,t_1}(H_0 \cap H_n) = L^n_{s_2,t_2}(H_0 \cap H_n).$$

Then, there is a matrix

$$\begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix} \in H_0 \cap H_n$$

such that

$$L^n_{s_2,t_2} = \begin{bmatrix} \pi_{s_2} & \pi_{s_2} p_{t_2} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \pi_{s_1} & \pi_{s_1} p_{t_1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} a\pi_{s_1} & b\pi_{s_1} + \pi_{s_1} p_{t_1} \\ 0 & 1 \end{bmatrix},$$

which implies $\pi_{s_2} = a\pi_{s_1}$ and $\pi_{s_2} p_{t_2} = \pi_{s_1}(p_{t_1} + b)$. Since π_{s_1} and π_{s_2} are monic polynomials and $p_{t_1}, p_{t_2} \in P_\gamma$, we conclude that $s_1 = s_2$ and $t_1 = t_2$. □

Thus, the degree of each node of V is $q + 1$ and the degree of each node of U is $q^{n-1}((q^n - 1)/(q - 1))$.

With an argument similar to the one used to prove Lemma 4, it can easily be shown that

$$U = \{H_n\} \cup \left\{ \begin{bmatrix} \alpha & 1 \\ 1 & 0 \end{bmatrix} H_n \colon\ \alpha \in \mathbb{F}_{q^n} \right\}. \tag{10}$$

In Theorem 4.10 of [10] it is shown that G is a 3-$(q^n+1, q+1, 1)$-*design*, which has the property that for any distinct $u_1, u_2, u_3 \in U$ there exists *exactly one* $v \in V$ adjacent to all three of them. (The parameters $q^n + 1$ and $q + 1$ indicate the output size and the input degree of the graph, respectively, whereas the parameters 3 and 1 indicate that for any three outputs there is one input adjacent to them.)

5.1. *Memory Organization Scheme for* $M \in O(N^2)$ *Variables*

5.1.1. *The Graph.* A *Balanced Incomplete Block Design* with parameters q^n, q, and 1 ($(q^n, q, 1)$-BIBD) is a bipartite graph with q^n outputs, input degree q and such that for any pair of outputs there exists *exactly one* input adjacent to both. This immediately implies

that there are $q^{n-1}((q^n-1)/(q-1))$ inputs and that the output degree is $(q^n-1)/(q-1)$. Let $N = q^n$ and $M = q^{n-1}((q^n-1)/(q-1))$ and note that, if $q \in O(1)$, $M \in \Theta(N^2)$. We use a $(q^n, q, 1)$-BIBD to distribute M variables among N memory modules, with each variable represented by q copies and each module storing $(q^n-1)/(q-1)$ copies of distinct variables. We call such a graph $G_1 = (V_1, U_1; E_1)$, where the set V_1 denotes the variables and U_1 the modules. We will see in the next subsection how G_1 is obtained, using a standard technique, as a subgraph of the 3-$(q^n+1, q+1, 1)$-design G defined earlier.

The following theorem establishes the expansion properties of G_1. As usual, given a set of variables S, and a c-bundle η for S, let $\Gamma_\eta(S)$ denote the set of modules storing the bundle copies.

Theorem 7. *G_1 has $(\frac{1}{2}, (q-1)/2)$-expansion.*

Proof. Let $\rho = \lfloor q/2 \rfloor$. We must prove that, for any set S and any $(\rho+1)$-bundle η,

$$|\Gamma_\eta(S)| \geq \frac{q-1}{2}|S|^{1/2}.$$

Without loss of generality, suppose $|\Gamma_\eta(S)| = |S|^{1/2}\mu$ for some $\mu > 0$. Since η contains at least $|S|(\rho+1)$ copies, there must be a module $u \in \Gamma_\eta(S)$ storing at least

$$\frac{|S|(\rho+1)}{|S|^{1/2}\mu} = \frac{\rho+1}{\mu}|S|^{1/2}$$

such copies. Clearly, these copies belong to distinct variables, which cannot share any module other than u because, otherwise, we would have more than one variable connected to the same pair of modules, violating the BIBD property. Since each such variable accounts for at least other ρ bundle copies, beside the one stored in u, we conclude that

$$|\Gamma_\eta(S)| > |S|^{1/2}\frac{(\rho+1)}{\mu}\rho,$$

which, combined with the hypothesis $|\Gamma_\eta(S)| = |S|^{1/2}\mu$, yields

$$\mu \geq \sqrt{\rho(\rho+1)} \geq \frac{q-1}{2}.$$ □

In the next section we show how G_1 can be implemented for any prime power q and integer n, so that a processor is able to compute the physical address of any copy of any variable in $O(\log N)$ time using constant internal storage (Theorem 9). Fixing $q = 3$ and combining the results of Theorems 7 and 1, we get

Theorem 8. *$M \in O(N^2)$ variables can be distributed among N processors of an MPC, with redundancy 3, so that any set of N distinct variables can be accessed, using the access protocol of Section 2, in time $O(N^{1/2})$. Moreover, each processor can determine the physical address of any copy in $O(\log N)$ time using $O(1)$ internal storage.*

5.1.2. *Implementation.* A $(q^n, q, 1)$-BIBD $G_1 = (V_1, U_1; E_1)$ can be obtained as a subgraph of the 3-$(q^n+1, q+1, 1)$-design $G = (V, U; E)$, using the following standard technique. Let u be an arbitrary node of U and define

$$V_1 = \{v \in V\colon\ (v, u) \in E\},$$

$$U_1 = U - \{u\},$$

$$E_1 = E_{/(V_1, U_1)},$$

where $E_{/(V_1, U_1)}$ denotes the edges of E between V_1 and U_1. As shown in Theorem 1.14 of [10], the graph $G_1 = (V_1, U_1; E_1)$ is a $(q^n, q, 1)$-BIBD, with

$$|V_1| = q^{n-1}\frac{q^n - 1}{q - 1} \ = \ M,$$

$$|U_1| = q^n \ = \ N,$$

where each node in V_1 has degree q, and each node in U_1 has degree $(q^n - 1)/(q - 1)$.

For convenience, we choose u to be the node corresponding to coset H_n. The following lemma identifies the cosets associated with the nodes in V_1 and U_1, according to the above construction.

Lemma 8. *We have*

$$V_1 = \left\{ \begin{bmatrix} \pi_s & \pi_s p_t \\ 0 & 1 \end{bmatrix} H_0\colon\ 0 \le s < \frac{q^n - 1}{q - 1} \text{ and } p_t \in P_\gamma \right\},$$

$$U_1 = \left\{ \begin{bmatrix} \alpha & 1 \\ 1 & 0 \end{bmatrix} H_n\colon\ \alpha \in \mathbb{F}_{q^n} \right\}.$$

Proof. Immediate from the definition of U_1 and V_1, Lemma 7, and equality (10). □

The edge set E_1 is, by definition, a subset of E consisting of all those edges incident to both V_1 and U_1. For each node x in G_1, denote the set of its neighbors by $\Gamma(x)$.

Lemma 9. *Let $AH_0 \in V_1$ and $BH_n \in U_1$, for some $A, B \in PGL_2(q^n)$ of the kind given in Lemma* 8. *Then*

$$\Gamma(AH_0) = \{AL_k^0 H_n\colon\ 0 \le k < q\}, \tag{11}$$

$$\Gamma(BH_n) = \left\{ BL_{s,0}^n H_0\colon\ 0 \le s < \frac{q^n - 1}{q - 1} \right\}. \tag{12}$$

Proof. Equality (11) follows immediately from (9), due to the exclusion of $H_n = AL_{-1}^0 H_n$ from U_1. To prove (12) let

$$B = \begin{bmatrix} \alpha & 1 \\ 1 & 0 \end{bmatrix}.$$

By Lemma 7, in G the $(q^n-1)/(q-1)$ nodes

$$\left\{BL^n_{s,0}H_0\colon\ 0\le s<\frac{q^n-1}{q-1}\right\}$$

are all adjacent to BH_n. Since the degree of a node of U_1 in G_1 is $(q^n-1)/(q-1)$, all we have to prove is that indeed

$$BL^n_{s,0}H_0=\begin{bmatrix}\alpha & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\pi_s & 0\\ 0 & 1\end{bmatrix}H_0\in V_1,\qquad 0\le s<\frac{q^n-1}{q-1}.$$

In fact,

$$\begin{bmatrix}\alpha & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\pi_s & 0\\ 0 & 1\end{bmatrix}=\begin{bmatrix}\alpha & \pi_s^{-1}\\ 1 & 0\end{bmatrix}=\begin{bmatrix}\pi_{s'} & \pi_{s'}p_t\\ 0 & 1\end{bmatrix}\begin{bmatrix}a & b\\ 1 & 0\end{bmatrix},$$

where the equation $b\pi_{s'}=\pi_s^{-1}$ determines b and $\pi_{s'}$, and the equation $\pi_{s'}(p_t+a)=\alpha$ determines p_t and a. Since $a,b\in\mathbb{F}_q$, the matrix

$$\begin{bmatrix}a & b\\ 1 & 0\end{bmatrix}\in H_0;$$

therefore, $BL^n_{s,0}H_0$ belongs to V_1. □

We now show how the copies of the variables are effectively distributed among the modules and how a processor determines the physical address of any copy. The approach is similar to the one used in Section 4.2. Again, the elements of $\mathbb{F}_{q^n}$ are represented as polynomials in γ of degree less than n, and the operations in $\mathbb{F}_{q^n}$ take $O(\log N)$ time, whereas all the other operations are performed in constant time. Let $v_0,\ldots,v_{M-1}$ denote the variables and $u_0,\ldots,u_{N-1}$ the memory modules. We first associate variables and modules with the appropriate cosets, as specified by Lemma 8. It is easy to give an ordering of the matrices

$$\left\{\begin{bmatrix}\pi_s & \pi_s p_t\\ 0 & 1\end{bmatrix}H_0\colon\ 0\le s<\frac{q^n-1}{q-1}\text{ and }p_t\in P_\gamma\right\}$$

so that given an index i, $0\le i<M$, a processor can compute the i-indexed such matrix, which we denote as A_i, in $O(\log N)$ time. Similarly, we establish an ordering for the matrices

$$\left\{B_j=\begin{bmatrix}\alpha_j & 1\\ 1 & 0\end{bmatrix}\colon\ \alpha_j\in\mathbb{F}_{q^n}\right\}.$$

Thus, for $0\le i<M$, variable v_i is associated with A_iH_0, and, for $0\le j<N$, module u_j is associated with B_jH_n. Let $\mathbb{F}_{q^n}=\{\alpha_0,\ldots,\alpha_{q^n-1}\}$, where the indexing of the α_i's is given by the lexicographic order of the n-tuples of coefficients.

By virtue of Lemma 9, we can establish that the kth copy of a variable v_i is stored in the module associated with coset $A_iL^0_kH_n$, $0\le k<q$. The item stored at the address h of module j is a copy of the variable associated with coset $B_jL^n_{h,0}H_0$. Since the edges in the graph are cosets of $H_0\cap H_n$ we adopt, as before, the following convention: *the* kth *copy of*

v_i is stored at the address h in module u_j if and only if $A_i L_k^0(H_0 \cap H_n) = B_j L_{h,0}^n(H_0 \cap H_n)$, that is, if and only if

$$A_i L_k^0 \in B_j L_{h,0}^n(H_0 \cap H_n).$$

Suppose a processor wants to compute the address of the kth copy of v_i. A two-step procedure, similar to the one used in Section 4.2, is executed:

Step 1. From i and k compute A_i and L_k^0.

Step 2. Find j and h such that $A_i L_k^0 \in B_j L_{h,0}^n(H_0 \cap H_n)$.

It is easy to see that Step 1 takes $O(\log N)$ time. Consider Step 2. We have that

$$\begin{aligned} B_j L_{h,0}^n(H_0 \cap H_n) &= \left\{ \begin{bmatrix} \alpha_j & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \pi_h & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\} \\ &= \left\{ \begin{bmatrix} a\alpha_j\pi_h & b\alpha_j\pi_h + 1 \\ a\pi_h & b\pi_h \end{bmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\} \\ &= \left\{ \begin{bmatrix} \alpha_j & a^{-1}b\alpha_j + (a\pi_h)^{-1} \\ 1 & a^{-1}b \end{bmatrix} : \ a, b \in \mathbb{F}_q, \ a \neq 0 \right\}. \end{aligned}$$

Note that $A_i L_k^0$ is of the form $\begin{bmatrix} x & y \\ 1 & 0 \end{bmatrix}$, where x and y are elements of $\mathbb{F}_{q^n}$. The processor has to determine the indices j and h such that

$$\begin{bmatrix} x & y \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_j & a^{-1}b\alpha_j + (a\pi_h)^{-1} \\ 1 & a^{-1}b \end{bmatrix},$$

for some $a, b \in \mathbb{F}_q$ and $a \neq 0$. This implies $b = 0$, and, therefore, $\alpha_j = x$ and $(a\pi_h)^{-1} = y$, which allows a processor to determine j and h in $O(\log N)$ time. We have

Theorem 9. *A processor computes the physical address of any copy in $O(\log N)$ time using $O(1)$ internal storage.*

5.2. *Memory Organization Scheme for $M \in O(N^3)$ Variables*

5.2.1. *The Graph.* The graph that we use in this case is a 3-$(q^n + 1, q, q - 2)$-*design* with $q > 3$, which is a bipartite graph $G_2 = (V_2, U_2; E_2)$ such that $|U_2| = q^n + 1$, each node in V_2 has degree q, and for any three distinct nodes $u_1, u_2, u_3 \in U_2$ there exist *exactly* $q - 2$ nodes of V_2 adjacent to all three of them. The properties of such a graph are similar to those of the graph G defined at the beginning of Section 5, which is a 3-$(q^n + 1, q + 1, 1)$-design, and would yield a comparable time performance for the MOS. However, G_2 allows us to devise a simpler implementation. From its definition, it immediately follows that $|V_2| = q^{n-1}((q^{2n}-1)/(q-1))$, and that the degree of each node of U_2 is $q^n((q^n-1)/(q-1))$. Thus, we use G_2 to distribute $M = q^{n-1}((q^{2n}-1)/(q-1))$ variables among $N = q^n + 1$ modules, with each variable replicated in q copies, and each module storing $q^n((q^n - 1)/(q - 1))$ copies of distinct variables.

Observe that for $q = 3$ the graph is trivial because $M = \binom{N}{3}$ and the nodes of V_2 are in one-to-one correspondence with the triplets of nodes of U_2. In this case the expansion property of the graph, proved below, does not hold, and this is why we require $q > 3$.

Theorem 10. *For $q > 3$, G_2 has $(\frac{2}{3}, q^{2/3}/3)$-expansion.*

Proof. Let $\rho = \lfloor q/2 \rfloor$. We must prove that, for any set $S \in V$ and any $(\rho+1)$-bundle η for S,

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

Without loss of generality, suppose $|\Gamma_\eta(S)| = |S|^{1/3}\mu$ for some $\mu > 0$. Note that if $q > 3$, then $\rho \geq 2$. Since η contains at least $|S|(\rho+1)$ copies, there must be a module $u \in \Gamma_\eta(S)$ storing at least

$$\frac{|S|(\rho+1)}{|S|^{1/3}\mu} = \frac{\rho+1}{\mu}|S|^{2/3}$$

bundle copies, which clearly belong to distinct variables. Let $S' \subseteq S$ be the set of these variables. Each such variable accounts for at least other ρ bundle copies, beside the one stored in u, which, in turn, account for at least $\binom{\rho}{2}$ pairs of modules, not including u. By adding u to each pair, we obtain $\binom{\rho}{2}$ triplets. Let

$$t = |\Gamma_\eta(S') - \{u\}|.$$

There are $\binom{t}{2}$ triplets formed by two modules in $\Gamma_\eta(S') - \{u\}$ and u. By the definition of a 3-$(q^n+1, q, q-2)$-design, each triplet occurs exactly $q-2$ times, and, therefore, we conclude that

$$|S'|\binom{\rho}{2} \leq \binom{t}{2}(q-2).$$

Recalling that $|S'| \geq ((\rho+1)/\mu)|S|^{2/3}$, the above relation implies

$$t \geq |S|^{1/3}\left(\frac{\rho(\rho^2-1)}{\mu(q-2)}\right)^{1/2}.$$

Since $|\Gamma_\eta(S)| \geq |\Gamma_\eta(S')| > t$, combining the inequality for t with the hypothesis $|\Gamma_\eta(S)| = |S|^{1/3}\mu$, we obtain

$$\mu \geq \left(\frac{\rho(\rho^2-1)}{q-2}\right)^{1/3} \geq \frac{q^{2/3}}{3}.$$

□

The next section shows how to construct G_2 for any prime power $q > 3$ and integer n, and how the copies of the variables can be organized among the modules according to G_2, so that a processor is able to compute the physical address of any copy in $O(\log N)$ time using constant internal storage (Theorem 13). Choosing $q = 5$ and combining the results of Theorems 10 and 1 we obtain

Theorem 11. *$M \in O(N^3)$ variables can be distributed among N processors of an MPC, with redundancy 5, so that any set of N distinct variables can be accessed, using the access protocol of Section 2, in time $O(N^{2/3})$. Moreover, each processor can determine the physical address of any copy in $O(\log N)$ time using $O(1)$ internal storage.*

5.2.2. *Implementation.* We construct G_2 by combining $q^n + 1$ copies of the graph G_1, studied in Section 5.1, as explained below. Recall that G_1 was derived from $G = (V, U; E)$ by choosing an arbitrary node $u \in U$ and considering set $U - \{u\}$ as the output set, set $\{v \in V\colon (v, u) \in E\}$ as the input set, and retaining only the edges incident to both these sets. Since $|U| = q^n + 1$, we can construct $q^n + 1$ such graphs, by choosing different u's. For convenience, let $U = \{u_{-1}, u_0, \ldots u_{q^n-1}\}$, where, consistently with (10), we set

$$u_{-1} = H_n,$$

$$u_j = \begin{bmatrix} \alpha_j & 1 \\ 1 & 0 \end{bmatrix} H_n, \qquad \alpha_j \in \mathbb{F}_{q^n}, \quad 0 \le j < q^n.$$

For $-1 \le j < q^n$, define the graph $G_{u_j} = (V_{u_j}, U_{u_j}; E_{u_j})$ as

$$V_{u_j} = \{v \in V\colon (v, u_j) \in E\},$$

$$U_{u_j} = U - \{u_j\},$$

$$E_{u_j} = E_{/(V_{u_j}, U_{u_j})}.$$

The following lemma generalizes the result of Lemma 8.

Lemma 10. *We have*

$$V_{u_{-1}} = \left\{ \begin{bmatrix} \pi_s & \pi_s p_t \\ 0 & 1 \end{bmatrix} H_0\colon\ 0 \le s < \frac{q^n - 1}{q - 1} \text{ and } p_t \in P_\gamma \right\}, \tag{13}$$

$$U_{u_{-1}} = \left\{ \begin{bmatrix} \beta & 1 \\ 1 & 0 \end{bmatrix} H_n\colon\ \beta \in \mathbb{F}_{q^n} \right\}, \tag{14}$$

and, for $0 \le j < q^n$,

$$V_{u_j} = \left\{ \begin{bmatrix} \alpha_j & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \pi_s & \pi_s p_t \\ 0 & 1 \end{bmatrix} H_0\colon\ 0 \le s < \frac{q^n - 1}{q - 1} \text{ and } p_t \in P_\gamma \right\}, \tag{15}$$

$$U_{u_j} = \left\{ \begin{bmatrix} \alpha_j & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \beta & 1 \\ 1 & 0 \end{bmatrix} H_n\colon\ \beta \in \mathbb{F}_{q^n} \right\}. \tag{16}$$

Proof. Equalities (13) and (14) follow from Lemma 8, since $V_{u_{-1}} = V_1$ and $U_{u_{-1}} = U_1$. Equality (15) follows from Lemma 7 and the definition of V_{u_j}. As for U_{u_j}, by using (10), it is easy to see that

$$\left\{ \begin{bmatrix} \alpha_j & 1 \\ 1 & 0 \end{bmatrix} H_n \right\} \cup \left\{ \begin{bmatrix} \alpha_j & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \beta & 1 \\ 1 & 0 \end{bmatrix} H_n\colon\ \beta \in \mathbb{F}_{q^n} \right\} = U,$$

therefore,

$$\left\{ \begin{bmatrix} \alpha_j & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \beta & 1 \\ 1 & 0 \end{bmatrix} H_n\colon\ \beta \in \mathbb{F}_{q^n} \right\} = U - \{u_j\}. \qquad \square$$

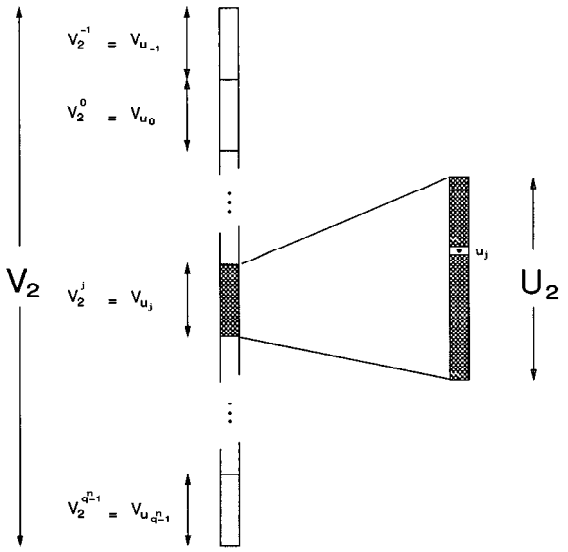


Fig. 4. $G_2 = (V_2, U_2; E_2)$.

For any $j \geq 0$ the graph G_{u_j} is isomorphic to $G_{u_{-1}}$, since the cosets associated with the nodes of G_{u_j} are obtained by multiplying those of $G_{u_{-1}}$ by the same matrix, and the adjacencies are preserved because the edges, which are associated with the cosets of $H_0 \cap H_n$, are also multiplied by the same matrix. Furthermore, since $G_{u_{-1}} = G_1$, we conclude that each G_{u_j} is a $(q^n, q, 1)$-BIBD.

We are now ready to construct the 3-$(q^n + 1, q, q - 2)$-design $G_2 = (V_2, U_2; E_2)$. Recall that $|U_2| = q^n + 1$ and $|V_2| = q^{n-1}((q^{2n} - 1)/(q - 1))$. Set $U_2 = U$, and partition the set V_2 into $q^n + 1$ *disjoint* subsets of $q^{n-1}((q^n - 1)/(q - 1))$ nodes each, namely $V_2^j = V_{u_j}$, for $-1 \leq j < q^n$. Note that the V_{u_j}'s are not disjoint, therefore distinct V_2^j's may include the same cosets which, however, will be reckoned as distinct nodes. Each V_2^j is connected to U by the same edges that connect V_{u_j} to U_{u_j} in G_{u_j} (see Figure 4).

It is easy to see that, in G_2, each node of V_2 has degree q and each node of U_2 has degree $q^n((q^n - 1)/(q - 1))$.

Theorem 12. *G_2 is a 3-$(q^n + 1, q, q - 2)$-design.*

Proof. We must prove that for any three distinct nodes $x, y, z \in U_2$, there are exactly $q - 2$ nodes in V_2 adjacent to all three of them. Fix a triplet (x, y, z) and consider these nodes in the graph G. Since G is a 3-$(q^n + 1, q + 1, 1)$-design, there is exactly one node $v \in V$ adjacent to x, y, and z. Since the degree of v is $q + 1$, there are other $q - 2$ nodes of U adjacent to v in G. Call these nodes v_i, for $1 \leq i \leq q - 2$. Thus, for any i we must have that $v \in V_{v_i}$, $x, y, z \in U_{v_i}$ and, clearly, $(v, x), (v, y), (v, z) \in E_{v_i}$. The same edges occur in G_2 and, therefore, we have found $q - 2$ nodes of V_2 adjacent to x, y, and z in

G_2. Since each node of V_2 accounts for $\binom{q}{3}$ triplets, we must have

$$|V_2|\binom{q}{3} \geq \binom{|U_2|}{3}(q-2).$$

The theorem follows by observing that the above is, in fact, an equality. □

The implementation of G_2, for the purposes of address computation, is easy, once we regard this graph as decomposed into the G_{u_j}'s. As observed before, each G_{u_j} is isomorphic to G_1 and can be implemented as explained in Section 5.1. Note that module $u_j \in U_2$ is included in every U_{u_i} with $i \neq j$. Therefore, u_j contains $(q^n-1)/(q-1)$ copies of the variables of $V_2^i = V_{u_i}$, for any $i \neq j$. Since there are q^n such indices i, we subdivide u_j into q^n *blocks* of size $(q^n-1)/(q-1)$, each block reserved for the copies of a distinct V_2^i. The blocks occupy consecutive position in the module, according to the ordering of the indices i. The organization of the copies in block i is identical to that of u_j in the graph G_{u_i}.

The variables are subdivided in groups according to the partition of V_2 into the subsets V_2^i, $-1 \leq i < q^n$. Suppose a processor wants to compute the physical address of the kth copy of a variable v, and suppose v belongs to V_2^i. We first compute the address of the copy in G_{u_i}, with the same procedure described before for G_1. Suppose the copy resides in module u_j. Then, once we have the address within u_j with respect to the graph G_{u_i}, we just add the appropriate multiple of $(q^n-1)/(q-1)$ to account for the number of blocks in u_j preceding the one where the copies of the variables of V_2^i are stored. The following theorem easily follows.

Theorem 13. *A processor computes the physical address of any copy in $O(\log N)$ time using $O(1)$ internal storage.*

Appendix

The goal of this appendix is to find a suitable set of matrices representatives of $PGL_2(q^n)/H_0$, such that, given an integer i, $0 \leq i < |PGL_2(q^n)/H_0|$, the ith matrix is easily retrieved. This is necessary for the implementation of the MOS presented in Section 4. For convenience, we only consider the case of $q = 2$ and n odd, which is simpler to explain and, yet, general enough for our purposes.

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

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

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

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

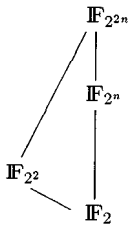
By (1), $|PGL_2(2^n)| = (2^{2n}-1)2^n$, thus

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

The choice of the representatives for $PGL_2(2^n)/H_0$ is made much simpler if we regard the rows of the matrices of $PGL_2(2^n)$ as elements of extension field $\mathbb{F}_{2^{2n}}$, as explained below. Let λ be a generator of the multiplicative group $\mathbb{F}^*_{2^{2n}}$, and define

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

We have $\mathbb{F}^*_{2^2} = \{\lambda^{i\rho}\colon\ 0 \le i < 3\}$, so $w = \lambda^\rho$ is a generator for $\mathbb{F}^*_{2^2}$. 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 $\mathbb{F}_{2^n}$; thus, 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 $\Phi\colon \mathbb{F}_{2^n} \times \mathbb{F}_{2^n} \to \mathbb{F}_{2^{2n}}$ as

$$\Phi(\alpha, \beta) = \alpha w + \beta, \qquad \forall \alpha, \beta \in \mathbb{F}_{2^n}.$$

We represent a matrix

$$\begin{bmatrix} x & y \\ z & v \end{bmatrix} \in PGL_2(2^n)$$

by the pair

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

where, since the matrix is nonsingular, both $\Phi(x, y)$ and $\Phi(z, v)$ are nonzero. From now on, either the usual matrix notation or the above pair notation is used wherever appropriate, 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

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

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

$$\sigma = 2^n + 1,$$

$$\tau = \frac{2^n + 1}{3},$$

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

$$\{[\lambda^{i\sigma}]\colon\ 0 \le i < 2^n - 1\} = \{[\lambda^{i\tau}]\colon\ 0 \le i < 2^n - 1\} \tag{17}$$

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

$$S_1 = \{[\lambda^{i\tau}]\colon\ 0 \le i < 2^n - 1\},$$

$$S_2 = \{[\lambda^i]\colon\ 0 \le i < \rho \text{ such that } \tau \not| \ i\}.$$

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 $(x, y) \cdot H_0(i)$, for $i = 1, \ldots, 6$. Using the function Φ, view these vectors as elements of $\mathbb{F}^*_{2^{2n}}$, and let

$$\alpha_i = \Phi((x, y) \cdot H_0(i)), \qquad i = 1, \ldots, 6.$$

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 some sense, preserved among the α_i's. Suppose $\alpha_1 \in [\lambda^k]$, for some k, $0 \le k < \rho$.

Lemma 11.

1. $\{\alpha_i\colon\ 1 \le i \le 6\} = [\lambda^k] \cup [\lambda^{(k2^n) \bmod \rho}]$.
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 it can easily be proved that, when n is odd, $w^{2^n} = w + 1$, therefore

$$\begin{aligned} (\Phi(x, y))^{2^n} &= (xw + y)^{2^n} = x^{2^n} w^{2^n} + y^{2^n} = xw^{2^n} + y = xw + (x + y) \\ &= \Phi(x, x + y). \end{aligned}$$

This shows that $\alpha_5 = \alpha_1^{2^n}$ and, since we assumed $\alpha_1 \in [\lambda^k]$, we have

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

Now we wish to identify the values of k for which $[\lambda^k] = [\lambda^{(k2^n) \bmod \rho}]$. We observe:

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

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

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 coincides with $[\lambda^k]$, with each element occurring twice, if $[\lambda^k] \in S_1$.

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

$$k(i, j) = (i + j\sigma) \bmod \rho. \tag{18}$$

Fact 1. *For any* $1 \le i \le (2^{n-1}-1)/3$ *and* $0 \le j < 2^n - 1$ *we have*:

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

Proof. 1. Since $\tau \mid \sigma$ and $\tau \mid \rho$, then

$$\tau \mid k(i, j) \quad \Leftrightarrow \quad \tau \mid i.$$

However, $1 \le i \le (2^{n-1}-1)/3 < \tau$, therefore $\tau \nmid i$ and $[\lambda^{k(i,j)}] \in S_2$.

2. (a) The "$\Leftarrow$" part is trivial. Suppose $k(i', j') = k(i, j)$. Then

$$(i' + j'\sigma) \bmod \rho = (i + j\sigma) \bmod \rho.$$

Since $i' \le \rho$, 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 (17), $(j - j')\sigma \bmod \rho = h\tau$ for some h, $0 \le h < 2^n - 1$. 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 $0 \le d < 2^n - 1$. Now, since n is odd, $\operatorname{lcm}(\sigma, \rho) = 2^{2n} - 1$ and, thus, $(j' - j)\sigma = d\rho$ implies that $(j' - j)$ is a multiple of $\operatorname{lcm}(\sigma, \rho)/\sigma = 2^n - 1$. However, 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, $0 \le h < 2^n - 1$. 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 $(2^{n-1} - 1)/3 \leq \lfloor \tau/2 \rfloor$. □

We are now ready to define the matrices representatives of $PGL_2(2^n)/H_0$. The matrices are given in the pair notation and, for convenience, are partitioned into four sets, L_1, L_2, L_3, and L_4.

Definition 2.

$$L_1 = \{\langle 1, \lambda^{h\sigma} w\rangle\colon\ 0 \leq h < 2^n - 1\},$$
$$L_2 = \{\langle 1, \lambda^{k(i,j)} w^s\rangle\},$$
$$L_3 = \{\langle \lambda^{k(i,j)} w^s, 1\rangle\},$$
$$L_4 = \{\langle \lambda^{k(i,0)}, \lambda^t w^s\rangle\colon\ 1 \leq t < \rho\ ,\ \tau \not| \ t \text{ and } \lambda^{k(i,0)}(\lambda^t w^s)^{-1} \notin \mathbb{F}^*_{2^n}\},$$

with $1 \leq i \leq (2^{n-1} - 1)/3$, $0 \leq j < 2^n - 1$, and $0 \leq s < 3$, wherever they occur.

Lemma 12.

$$|L_1| + |L_2| + |L_3| + |L_4| = |PGL_2(2^n)/H_0|.$$

Proof. It is immediate that

$$|L_1| = 2^n - 1,$$

$$|L_2| = |L_3| = (2^n - 1)(2^{n-1} - 1).$$

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

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

Simple algebra shows that

$$|L_1| + |L_2| + |L_3| + |L_4| = 2^{n-1}((2^{2n} - 1)/3) = |PGL_2(2^n)/H_0|.$$ □

The next theorem shows that the matrices in the above four sets are indeed a set of representatives for $PGL_2(2^n)/H_0$. Let

$$\mathcal{L} \triangleq L_1 \cup L_2 \cup L_3 \cup L_4.$$

Theorem 14. *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 any distinct $A, B \in \mathcal{L}$, and for any $\delta \in \mathbb{F}_{2^n}$, $A \notin \delta B H_0$.

For the first fact we only have to prove that each $A \in \mathcal{L}$ is nonsingular. Let $A = \langle x, y\rangle \in \mathcal{L}$. Note that both x and y are nonzero; therefore A is nonsingular 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 1, $[\lambda^{k(i,j)} w^s] = [\lambda^{k(i,j)}] \in S_2$ and, by (17) and the definition of S_2, $\lambda^{k(i,j)} w^s \notin \mathbb{F}_{2^n}$.
- $A = \langle \lambda^{k(i,j)} w^s, 1\rangle \in L_3$. Identical 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 nonsingular.

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 11, it can be argued 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 11, 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 that $(i = i') \wedge (j = j') \wedge (s = s')$. 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 11 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 1, 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 $s \neq s'$. From the proof of Lemma 11 it can easily be seen that the only matrix in $\delta B H_0$ 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 $B H_0$.
- *$A, B \in L_3$, with $A \neq B$.* Identical 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 that $(i = i') \wedge (t = t') \wedge (s = s')$. 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 11 and Fact 1, $A \in \delta B H_0$ implies $k(i, 0) = k(i', a)$, and therefore $i = i'$ 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 11 implies that δB is actually the only matrix in $\delta B H_0$ whose first com-

ponent is λ^i. Therefore, in order to have $A \in \delta B H_0$ it must be that $A = \delta B$, that is $t = t'$ and $s = s'$, a contradiction. □

Let $M = |PGL_2(2^n)/H_0|$. We show how to associate the integers $0, 1, \ldots, M-1$ with the matrices of $\mathcal{L}$ bijectively. We first explain how, given an integer r, $0 \le r < M$, 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 12,

$$|L_1| = 2^n - 1,$$

$$|L_2| = |L_3| = (2^n - 1)(2^{n-1} - 1),$$

$$|L_4| = (2^n - 1)\frac{2^{n-1} - 1}{3}(2^n - 3),$$

where $|L_1| + |L_2| + |L_3| + |L_4| = M$. We number the matrices so that

$$A_r \in L_1 \quad \text{if} \quad 0 \le r < 2^n - 1,$$

$$A_r \in L_2 \quad \text{if} \quad 2^n - 1 \le r < (2^n - 1) + (2^n - 1)(2^{n-1} - 1) = (2^n - 1)2^{n-1},$$

$$A_r \in L_3 \quad \text{if} \quad (2^n - 1)2^{n-1} \le r < (2^n - 1)2^{n-1} + (2^n - 1)(2^{n-1} - 1) = (2^n - 1)^2,$$

$$A_r \in L_4 \quad \text{if} \quad (2^n - 1)^2 \le r < (2^n - 1)^2 + (2^n - 1)\frac{2^{n-1} - 1}{3}(2^n - 3) = M.$$

Thus we must find a bijection 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 $(2^n - 1)^2 \le r < M$ and set $r' = r - (2^n - 1)^2$, so that $0 \le r' < (2^n - 1)((2^{n-1} - 1)/3)(2^n - 3)$. A_r will be of the form

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

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

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

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

$$\{t + s\rho \colon 1 \le t < \rho,\ \tau \not| t, \text{ and } 0 \le s < 3\} = \{t' \colon 0 \le t' < 2^{2n} - 1,\ \tau \not| t'\}.$$

Therefore, finding the lth pair (t, s) such that $\lambda^{k(i,0)}(\lambda^t w^s)^{-1} \notin \mathbb{F}_{2^n}$ is equivalent to finding the lth index t', which is not a multiple of τ and such that $\lambda^{k(i,0)-t'} = \lambda^{i-t'} \notin \mathbb{F}_{2^n}$;

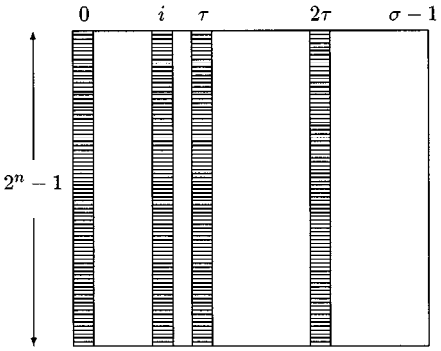


Fig. 5. Table of the indices $0, \ldots, 2^{2n} - 2$.

that is, $t' \neq i + j\sigma$ for any $0 \leq j < 2^n - 1$. Among the integers $0 \cdots 2^{2n} - 2$, those that are multiples of τ or are equal to $i + j\sigma$ (i.e., the "forbidden" indices), occupy fixed positions, as shown in Figure 5 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, τ, 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.

Once the matrix A_r is known in pair notation, we need to transform it into the usual form. 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

$$\alpha_i w + \beta_i = \lambda^i,$$
$$\alpha_j w + \beta_j = \lambda^j.$$

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 $O(n)$ operations over $\mathbb{F}_{2^n}$. Recalling that $n \in O(\log N)$, where N is the number of processors in the MOS, we have proved

Theorem 15. *Given an index 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

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

[2] Y. Aumann and A. Schuster. Improved memory utilization in deterministic PRAM simulations. *Journal of Parallel and Distributed Computing*, 12:146–151, 1990.

[3] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, New York, 1968.

[4] M. Dietzfelbinger and F. Meyer auf der Heide. Simple, efficient shared memory simulations. *Proc. 5th ACM Symp. on Parallel Algorithms and Architectures*, pages 110–119, 1993.

[5] D. K. Gifford. Weighted voting for replicated data. *Proc. 7th ACM Symp. on Operating System Principles*, pages 150–159, 1979.

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

[7] K. T. Herley. Efficient simulations of small shared memories on bounded degree networks. *Proc*. 30*th IEEE Symp*. *on Foundations of Computer Science*, pages 390–395, 1989.
[8] K. T. Herley. Space-efficient representations of shared data for parallel computers. *Proc*. 2*nd ACM Symp*. *on Parallel Algorithms and Architectures*, pages 407–416, 1990.
[9] K. Herley and G. Bilardi. Deterministic simulations of PRAMs on bounded-degree networks. *SIAM Journal on Computing*, 23(2):276–292, April 1994.
[10] D. R. Hughes and F. C. Piper. *Design Theory*. Cambridge University Press, Cambridge, MA, 1985.
[11] D. J. Kuck. A survey of parallel machine organization and programming. *ACM Computing Surveys*, 9(1):29–59, March 1977.
[12] F. Luccio, A. Pietracaprina, and G. Pucci. A new scheme for the deterministic simulation of PRAMs in VLSI. *Algorithmica*, 5(4):529–544, 1990.
[13] K. Mehlhorn and U. Vishkin. Randomized and deterministic simulations of PRAMs by parallel machines with restricted granularity of parallel memories. *Acta Informatica*, 21:339–374, 1984.
[14] M. Morgenstern. Natural bounded concentrators. *Combinatorica*, 15(1):111–122, 1995.
[15] A. Pietracaprina and F. P. Preparata. An $O(\sqrt{n})$-worst-case-time solution to the granularity problem. In K. W. Wagner P. Enjalbert, A. Finkel, editor, *Proc*. 10*th Symp*. *on Theoretical Aspects of Computer Science*, pages 110–119, Würzburg, February 1993. LNCS 665, Springer-Verlag, Berlin.
[16] A. Pietracaprina and G. Pucci. Tight bounds on deterministic PRAM emulations with constant redundancy. In J. V. Leeuwen, editor, *Proc*. 2*nd European Symp*. *on Algorithms*, pages 319–400, Utrecht, September 1994. LNCS 855, Springer-Verlag, Berlin.
[17] A. G. Ranade. How to emulate shared memory. *Journal of Computer and System Sciences*, 42:307–326, 1991. See also *Proc*. 28*th IEEE Symp*. *on Foundations of Computer Science*, pages 185–194, 1987.
[18] R. H. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Databases Systems*, 4(2):180–209, 1979.
[19] E. Upfal and A. Widgerson. How to share memory in a distributed system. *Journal of the ACM*, 34(1):116–127, 1987.

Received January 1995, *and in final form April* 1996.