REPORTS IN INFORMATICS

ISSN 0333-3590

Generalized Shuffle–Exchange Networks

Hans Munthe-Kaas [1]

REPORT NO 61 JUNE 1992

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

ISSN 0333-3590

Generalized Shuffle–Exchange Networks

Hans Munthe-Kaas [1]

REPORT NO 61 JUNE 1992

Abstract. A class of generalized Shuffle-Exchange (SE) nets is defined. As permutation networks these have the same functionality as the classical SE net, but some of them possess recursive structures lacking in the classical SE net. This make them very attractive from a hardware–designers point of view. We develop the theory of the topology of generalized SE nets and present general theorems showing how to construct networks built up recursively by using identical (or a small number of different) building blocks.

Index Terms interconnection networks, perfect shuffle, permutation networks, shuffle–exchange networks, SIMD computers.

1. Introduction. In massively parallel SIMD computers, or data parallel computers, the most common form of interprocessor communication is permutations of the data set. I.e. each (virtual) processor sends out one data item and receive one. A natural efficiency measure for SIMD interconnection networks is therefore the number of *routing steps* for a permutation, i.e. the number of parallel steps needed to do an arbitrary permutation. We consider here only static permutations, i.e. permutations that are known in advance, or that are to be performed many times, so that the cost of computing the optimal routing can be neglected. This is relevant for many algorithms and for e.g. measuring the ability of a given network to emulate arbitrary networks. We define the *cost* of a network as the product of the permuting power and the number of wires in the network, thus the cost is a price/performance measure. It is relatively easy to show that the cost of any interconnection network must satisfy:

$$c(n) = \Omega(n \cdot \log(n)) \ , \tag{1}$$

[1] Department of Informatics, University of Bergen, Høyteknologisenteret, N-5020 Bergen.

Network	Permuting power	No. of wires	Cost
Complete graph	1	$\Theta(n^2)$	$\Theta(n^2)$
Boolean Cube	$\Theta(\log(n))$	$\Theta(n\log(n))$	$\Theta(n\log^2(n))$
Butterfly	$\Theta(\log(n))$	$\Theta(n\log(n))$	$\Theta(n\log^2(n)$
2–D Mesh	$\Theta(\sqrt{n}))$	$\Theta(n)$	$\Theta(n\sqrt{n})$
Ring	$\Theta(n)$	$\Theta(n)$	$\Theta(n^2)$
Cube Connected Cycles	$\Theta(\log(n))$	$\Theta(n)$	$\Theta(n\log(n))$
(Generalized) SE	$\Theta(\log(n))$	$\Theta(n)$	$\Theta(n\log(n))$

TABLE 1
Cost complexity for various interconnection networks

where n is the number of nodes in the network. Figures for some networks are summarized in Table 1. The *Shuffle Exchange* (SE) network [18] is among the few networks with optimal cost complexity. Despite of this, the SE network has not been extensively used in hardware design. The reason for this is probably that the SE network is rather complicated to draw in a nice regular fashion; the graph lacks the recursive structure found in e.g. the Boolean cube and other popular networks. Layouts for the SE network has been studied in the context of VLSI design [7], but due to the lack of recursiveness, the layouts are very complicated. In his excellent new book [8] F.T. Leighton writes: "The structures of the shuffle–exchange and de Bruijn graphs are probably [among] the most intriguing and least understood".

In this paper we will introduce a family of *Generalized Shuffle Exchange* (GSE) networks. This family of graphs contains the 'classical' SE net as a special case. In terms of functionality as permutation networks, all the GSE networks are equivalent, but as graphs they are non-isomorphic. We show that the GSE family of networks contains two chains of 'maximally foldable graphs'. These are recursively defined, and allow very regular layouts. They represent exciting new alternatives to the hardware designers.

Our basis for searching for generalized versions of the SE net is the following question: *What is the most general form of a network with the same permuting functionality as the (classical) SE net?* To make this question more presize we must define the contents of this 'permuting functionality'. This is given in Assumption 1, motivated by the results below. Before plunging into the theory part of this paper, the reader is invited to look at the figures and read the Epilogue.

A famous theorem from telephone switching theory (in a more general setting known as the Slepian-Duguid [1] theorem) states that any permutation of $N = 2m$ objects can be performed in the following way:

1. Arrange the objects in a 2 by m array.
2. Permute objects within each column of the array.
3. Permute objects within each row of the array.
4. Permute objects within each column of the array.

The theorem is also valid for a general factorization $N = m_1 \cdot m_2$. We call the version above SD2.

In the case where $N = 2^n$, SD2 may be applied recursively, thus *any* permutation of 2^n objects can be performed in $2n - 1$ steps, where each step involves a rearrange-

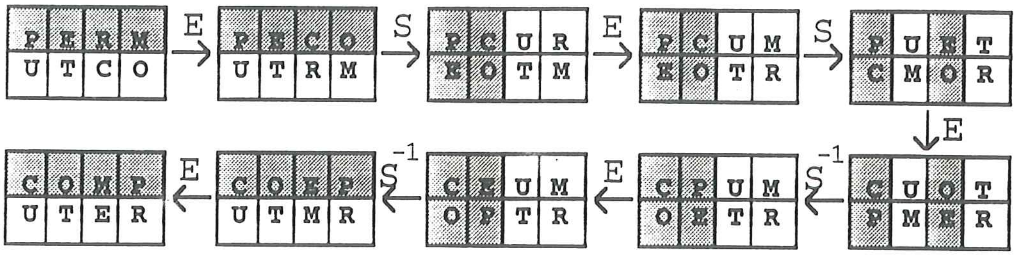

FIG. 1. *Recursive application of SD2 to perform arbitrary permutation on SE network*

ment of the objects in 2 by 2^k arrays, and (possibly) swapping pairs of objects. The algorithm runs forwards until all the elements are arranged in 2 by 1 arrays, and then returns back again. The Beneš network, based on this idea, is the most famous network capable of performing general permutations. It is well known that the Beneš network can be realized by a (recirculating) SE network. This is shown for $N = 8$ in Figure 1. The SE network consists of 2^n nodes, represented as n-digit binary numbers $g = (g_{n-1}, g_{n-2}, \ldots, g_0)$. The network can perform two different permutations; the *shuffle* and the *exchange* defined as:

$$S((g_{n-1}, g_{n-2}, \ldots, g_0)) = (g_{n-2}, g_{n-3}, \ldots, g_0, g_{n-1})$$

and

$$E((g_{n-1}, g_{n-2}, \ldots, g_0)) = (g_{n-1}, g_{n-2}, \ldots, g_0 \oplus 1) ,$$

where $\oplus$ denotes boolean addition (xor). We will use the shorthand notation $\overline{g_0} = g_0 \oplus 1$. The exchange is usually *conditional*, i.e. two objects differing only in the rightmost address bit may or may not be conjugated, depending on the value of the other bits.

In general any permutation P of $N = 2^n$ objects can be written as a product of shuffles, inverse shuffles and conditional exchanges as:

$$P = E_{2n-1}S^{-1}E_{2n-2}S^{-1}\cdots S^{-1}E_nSE_{n-1}S\cdots SE_2SE_1$$

The computation of conditions on each exchange E_i, to achieve a given permutation, is called the *routing problem* for the Beneš network, and is discussed in a series of papers [6, 9, 10, 11, 12, 20]. For general permutations the routing problem is hard to solve, but for many important classes of permutations it can be solved very efficiently.

2. Definition of Generalized SE networks. The (classical) SE-network is not the only recirculating network capable of performing the operations of SD2 recursively. It is our goal to characterize all recirculating networks with this capability, i.e. we seek all pairs of permutations $\{S, E\}$ such that $E^2 = I$ and $\{S^{-1}, S, E\}$ can realize a Beneš network. The resulting networks are closely related to the classical SE network, and we call them *Generalized Shuffle Exchange networks.*

The necessary and sufficient conditions for the recursive application of SD2 is that there exists a partitioning of the set of objects $\mathcal{G}$ in two sets that are mapped onto each other by E, and that successive use of S splits each part in two smaller parts mapped to each other by E. Formally, we require:

ASSUMPTION 1.

1. *We are given* $N = 2^n$ *objects* $\mathcal{G}$ *and two permutations* S *and* E *on* $\mathcal{G}$ *such that* $E^2 = I$, *the identity permutation.*
2. *There exists a partitioning* $\mathcal{G} = \mathcal{P}_0 \cup \mathcal{P}_1$ *such that* $|\mathcal{P}_0| = |\mathcal{P}_1| = N/2$ *and* $E(\mathcal{P}_0) = \mathcal{P}_1$.
3. *Define recursively the sets* $\mathcal{P}_{i_0,i_1,\ldots,i_k}$ *by*

$$\mathcal{P}_{i_0,i_1,\ldots,i_k} = S(\mathcal{P}_{i_0,i_1,\ldots,i_{k-1}}) \cap \mathcal{P}_{i_k}$$

for all $k \in \{1, 2, \ldots, n-1\}$ *and* $i_j \in \{0, 1\}$. *Then:*

$$E(\mathcal{P}_{i_0,i_1,\ldots,i_k}) = \mathcal{P}_{i_0,i_1,\ldots,\overline{i_k}} .$$

The following theorem characterize the possible forms S and E can take:

THEOREM 2.1. *If* $\{S, E\}$ *is a pair of permutations satisfying the conditions in Assumption 1, then there exists a binary indexing of the elements* $g \in \mathcal{G}$:

$$g = (g_{n-1}, g_{n-2}, \ldots, g_1, g_0) \; ; \quad g_i \in \{0, 1\}$$

such that

$$E(g) = (g_{n-1}, g_{n-2}, \ldots, g_1, \overline{g_0}) \tag{2}$$

and

$$S(g) = (g_{n-2}, g_{n-3}, \ldots, g_0, f(g)) \tag{3}$$

where f *is a boolean function satisfying*

$$f(g) = f((g_{n-1}, g_{n-2}, \ldots, g_0)) = h((g_{n-2}, g_{n-3}, \ldots, g_0)) \oplus g_{n-1} , \tag{4}$$

for some boolean function h.

Conversely: let $\mathcal{G}$ *be the set of all* n *digit binary numbers. Given a boolean function* $f : \{0, 1\}^n \to \{0, 1\}$ *satisfying (4), then (2) and (3) define a pair of permutations satisfying the conditions in Assumption 1.*

Note: A map of the form given in (3) is called a *shift register*, and with the additional property in (4) it is called a *non-singular* (or invertible) shift register. A thorough account for the mathematical properties of shift registers is found in [4].

Proof: By condition 2 we find that

$$|\mathcal{P}_{i_0,i_1,\ldots,i_k}| = |\mathcal{P}_{i_0,i_1,\ldots,i_{k-1}}|/2 = N/2^k$$

thus $\mathcal{P}_{i_0,\ldots,i_{n-1}}$ consists of a uniquely defined single element for $\{i_j\}_{j=0}^{n-1}$ given. We make the identification

$$g = (i_0, i_1, \ldots, i_{n-1}) \equiv \mathcal{P}_{i_0,i_1,\ldots,i_{n-1}} .$$

By induction it is easy to show that

$$\mathcal{P}_{i_0,i_1,\ldots,i_{n-1}} = S^{n-1}(\mathcal{P}_{i_0}) \cap S^{n-2}(\mathcal{P}_{i_1}) \cap \cdots \cap S(\mathcal{P}_{i_{n-2}}) \cap \mathcal{P}_{i_{n-1}} .$$

Thus a re-indexing gives:

$$g = (g_{n-1}, g_{n-2}, \dots, g_0) = S^{n-1}\mathcal{P}_{g_{n-1}} \cap S^{n-2}\mathcal{P}_{g_{n-2}} \cap \dots \cap S\mathcal{P}_{g_1} \cap \mathcal{P}_{g_0} \ .$$

This shows that

$$S(g_{n-1}, \dots, g_0) = (g_{n-2}, g_{n-3}, \dots, g_0, g_{-1})$$

for some $g_{-1} \in \{0,1\}$, thus S is a shift register. In [4] it is shown that a shift register is invertible if and only if it satisfies (4).

The second part of the theorem is checked by letting

$$\mathcal{P}_0 = \{\, g = (g_{n-1}, \dots, g_0) \mid g_0 = 0 \,\} \ \ ; \ \mathcal{P}_1 = \mathcal{G} \backslash \mathcal{P}_0 \ .$$

□

DEFINITION 1. *A* Generalized Shuffle Exchange network, *GSE*(n,f), *is a graph consisting of* $N = 2^n$ *nodes* $\mathcal{G}$, *with edges* (g, Eg) *and* (g, Sg) *for all* $g \in \mathcal{G}$, *where* E *is defined in (2) and* S *is a non-singular shift register defined in (3) and (4).*

Note that the problem of computing a routing for a GSE network is equivalent to the routing problem for the classical SE network; if $\{E_i\}_{i=1}^{2n-1}$ is a routing for the classical network, then $\{\tilde{E}_i\}_{i=1}^{2n-1}$ is a routing for the same permutation on the GSE, where $\tilde{E}_i(g) = E_i(g) \oplus f(g)$. Thus the functionality of the different networks is identical.

The networks have, however, different topologies, and some of the generalized nets have recursive structures that make them more attractive from a hardware point of view than the classical SE network.

3. The de Bruijn graph.

DEFINITION 2. *The* de Bruijn *graph* $B(n)$ *is a graph with* $N = 2^n$ *nodes, each node* $(g_{n-1}, g_{n-2}, \dots, g_0)$ *is connected with an edge to the two nodes* $(g_{n-2}, g_{n-3}, \dots, g_0, 0)$ *and* $(g_{n-2}, g_{n-3}, \dots, g_0, 1)$.

Although the de Bruijn graph can be studied as an interconnection network in its own right [13] (it's functionality is very close to the SE network), we are mainly interested in it as a tool for understanding the topology of GSE graphs.

An *automorphism* of a graph is a permutation $\phi : \mathcal{G} \to \mathcal{G}$ s.t. $(\phi(g), \phi(h))$ is an edge in the graph if and only if (g,h) is an edge. Automorphisms of permutation networks are very useful from a hardware point of view, since they represents all the symmetries of the graph, and they allow the graph to be folded to a simpler and more compact form. The folded graph is called the *quotient* graph, and is defined as the graph obtained by merging pairs of nodes $\{g, \phi(g)\}$ into single 'supernodes'. In Section 5 we will develop more general folding theorems, where many points are collected to supernodes.

Parts 2 and 3 of the following theorem are also found in [3].

THEOREM 3.1. *Let* $g^* = (\overline{g_{n-1}}, \dots, \overline{g_0})$ *denote the* complement *of* g.

1. The $B(n)$ *graph has only one non-trivial automorphism, given by*

$$\phi(g) = g^* \ .$$

2. *The quotient graph of* $B(n)$ *w.r.t.* ϕ *is:*

$$B(n)/\phi = B(n-1)$$

3. *The elements in* $B(n)/\phi$ *are identified with elements in* $B(n-1)$ *by the function:*

$$\psi\left(\left\{\begin{array}{c} g \\ g^* \end{array}\right\}\right) = (g_{n-1} \oplus g_{n-2}, g_{n-2} \oplus g_{n-3}, \ldots, g_1 \oplus g_0)\ . \tag{5}$$

Before we prove the theorem, we need some lemmas:

LEMMA 3.2. *The de Bruijn graph is connected.*

Proof: Obvious.

□

LEMMA 3.3. *All nodes* $g \in B(n)$ *except* $(0,0,\ldots 0)$ *and* $(1,1,\ldots,1)$ *are connected to two other nodes. The two exeptional nodes are connected to themselves and to one other node.*

Proof: Each node $g = (g_{n-1},\ldots,g_0)$ is connected to the two nodes $(g_{n-2},\ldots,g_0,0)$ and $(g_{n-2},\ldots,g_0,1)$. These are different from g for every g except for $g = (0,\ldots,0)$ and $g = (1,\ldots,1)$.

□

Let a *cycle* denote a set of edges connecting a node to itself.

LEMMA 3.4. *Given a node* $g \in \mathcal{G}$ *there is a unique cycle of minimal length passing through* g.

Proof: The classical shuffle ($f \equiv 0$) has a period of at most n, thus the length p of the minimal cycle through g satisfies $p \leq n$. Any cycle through g with length $p \leq n$ is the unique path:

$$(g_{n-1}, g_{n-2}, \ldots, g_0) \to (g_{n-2}, \ldots, g_0, g_{p-1}) \to (g_{n-3}, \ldots, g_0, g_{p-1}, g_{p-2}) \to \cdots \text{ etc.}$$

□

LEMMA 3.5. *If an automorphism has a fixed point* $\phi(g') = g'$ *for some* g', *then* $\phi(g) = g$ *for every* $g \in \mathcal{G}$, *i.e.* ϕ *is the trivial automorphism.*

Proof: We start with a function ϕ s.t. $\phi(g') = g'$, and show that there is only one way to extend ϕ to an automorphism on $\mathcal{G}$. g' connects to two points, g^1 and g^2. These are topologically distinct, since one of them lies in the minimal cycle through g'. Thus the only way of extending ϕ to g^1 and g^2 is $\phi(g^1) = g^1$ and $\phi(g^2) = g^2$. Since $B(n)$ is connected, ϕ is extended uniquely in the same fashion to the trivial map on all of $\mathcal{G}$.

□

Proof of Theorem 3.1: Part 1: Since the two nodes $(0,\ldots,0)$ and $(1,\ldots,1)$ are topological exceptions, any automorphism must be of one of the two types:

1. Mappings where $(0,\ldots,0)$ and $(1,\ldots,1)$ are fixed points.
2. Mappings swapping $(0,\ldots,0)$ and $(1,\ldots,1)$.

The only automorphism of type 1 is the trivial map. It is readily checked that the ϕ given in the theorem is an automorphism (of type 2). From the group property of automorphisms, it follows that there is no other automorphism of type 2. (Suppose

$\tilde{\phi}$ is another automorphism of type 2, then $\tilde{\phi} o \phi^{-1}$ is an automorphism of type 1, and must equal the identity map. Hence $\tilde{\phi} = \phi$.)

Part 2 and 3: The given $\psi : B(n)/\phi \to B(n-1)$ is bijective, its inverse being

$$\psi^{-1}((g_{n-2}, \ldots, g_0)) = \{ (g_{n-1}, g_{n-1} \oplus g_{n-2}, \ldots, \oplus_{i=1}^{n-1} g_i, \oplus_{i=0}^{n-1} g_i) \mid g_{n-1} \in \{0,1\} \} .$$

A straightforward computation shows that ψ maps edges in $B(n)/\phi$ onto edges in $B(n-1)$. Thus ψ is a graph isomorphism.

□

4. The Recursive Structure of GSE-networks. We continue the discussion by finding automorphisms and quotient graphs for the GSE-networks.

To make life easier, we assume that shuffle and exchange edges are of different 'color', i.e. we exclude the possibility of automorphisms mapping exchange edges to shuffle edges or vice versa. This assumption can be justified by the different functionality of the two edge types in computer hardware. The omission of this assumption leads to more complicated proofs, but not to essentially different results.

Parts of the Theorems 4.1,4.3 and 4.4 are given in an other context in [3].

THEOREM 4.1. *If the condition*

$$f(g) = f(g^*) \quad \text{for all } g \in \mathcal{G} \tag{6}$$

holds, then the function ϕ in Theorem 3.1 is the unique non-trivial automorphism of GSE(n,f). If the condition fails, then GSE(n,f) has no non-trivial automorphism.

Proof: Let ϕ be an automorphism of GSE(n,f), i.e. it must satisfy $\phi o E = E o \phi$ and $\phi o S = S o \phi$. We also find that

$$\phi o E o S = E o S o \phi$$

thus ϕ must also be an automorphism of $B(n)$. This proves that ϕ in (5) is the only possible automorphism of GSE(n,f). Now, if condition (6) fails, it is evident that $\phi o S \neq S o \phi$, thus ϕ is not an automorphism for GSE(n,f).

□

Functions satisfying (6) are called *self complementary.*

THEOREM 4.2. *Let $n \geq 2$. There are $2^{2^{n-1}}$ different functions f satisfying (4), but some of these produce isomorphic GSE(n,f) graphs. There are $\frac{1}{2}(2^{2^{n-1}} + 2^{2^{n-2}})$ different (i.e. non-isomorphic) GSE$(n,\cdot)$ graphs. There are $2^{2^{n-2}}$ different GSE$(n,\cdot)$ graphs with non-trivial automorphisms.*

For $n = 1$ there are 2 graphs, both with automorphisms.

Proof: First statement: The set $\{0,1\}^n$ contains 2^n different points and f can be freely specified on half of these, thus $2^{2^{n-1}}$ different functions. Third statement: When (6) holds, then f can be specified on $2^{n-1}/2 = 2^{n-2}$ different points, this yields the result. Second statement: Self complementary functions must represent a unique graph, because if there are two different self complementary functions representing the same graph, we would get more than one non-trivial automorphism for $B(n)$. By the same reason there are exactly two different non–self complementary functions representing the same graph, they are pairs f and $f o \phi$. In total, the number of different graphs must equal the number of self complementary f plus half the number of f that are not self complementary. This yields the result. Last statement: is checked directly.

□

THEOREM 4.3. *If f is self complementary, then*

$$GSE(n,f)/\phi = GSE(n-1,h) \ ,$$

where h is the function

$$h(g_{n-2},g_{n-3},\ldots,g_0) = f(\oplus_{i=0}^{n-2} g_i, \oplus_{i=0}^{n-3} g_i, \ldots, g_0, 0) \ . \tag{7}$$

Nodes in $GSE(n,f)/\phi$ are identified with nodes in $GSE(n-1,h)$ where the function ψ given in (5).

Proof: A simple calculation shows that ψ maps E-edges in GSE$(n,f)/\phi$ onto E-edges in GSE$(n-1,h)$. A staightforward, but somewhat longer, computation shows that if we let S_f denote the shuffle on GSE(n,f) and S_h the shuffle on GSE$(n-1,h)$, then $\psi o S_f = S_h o \psi$. This shows that shuffle-edges in GSE$(n,f)/\phi$ are mapped onto shuffle-edges in GSE$(n-1,h)$.

□

This may be called the 'lowering theorem' for GSE networks. An inverse to this theorem is the following 'lifting theorem', which shows how foldable graphs can be recursively constructed:

THEOREM 4.4. *A graph $GSE(n-1,h)$ can be written as:*

$$GSE(n-1,h) = GSE(n,f)/\phi$$

where f is the function:

$$f(g_{n-1},g_{n-2},\ldots,g_0) = h(g_{n-1}\oplus g_{n-2}, g_{n-2}\oplus g_{n-3},\ldots,g_1\oplus g_0)\oplus g_0 \ . \tag{8}$$

If h is self complementary, then $GSE(n,f)$ is the unique $GSE(n,\cdot)$ graph folding to $GSE(n-1,h)$. If h is not self complementary, then there are two non-isomorphic graphs folding to $GSE(n-1,h)$, one of them is given by (8), the other by:

$$\tilde{f}(g_{n-1},g_{n-2},\ldots,g_0) = h(\overline{g_{n-1}}\oplus g_{n-2}, \overline{g_{n-2}}\oplus g_{n-3},\ldots,\overline{g_1}\oplus g_0)\oplus g_0 \ . \tag{9}$$

Proof: From (7) we find that

$$h((g_{n-1}\oplus g_{n-2},\ldots,g_1\oplus g_0)) = f((g_{n-1}\oplus g_0, g_{n-2}\oplus g_0,\ldots,g_1\oplus g_0, 0)) \ .$$

Since f must be self complementary, we arrive at (8). Second part of the theorem: From Theorem 4.2 it follows by counting that there is a 1–1 correspondence between general functions h of order $n-1$ and self complementary functions f of order n. Each self complementary f represents a unique graph, so since self complementary h represent GSE$(n-1,h)$ uniquely, such functions are lifted to a unique graph GSE(n,f). Non self complementary h, on the other hand, represent the same graph as $ho\phi$. These functions h may either be lifted directly by (8) or by applying (8) to $ho\phi$, which yields the alternative lifting (9).

□

By starting with the trivial one-node GSE graph, and lift n times, Theorem 4.4 yields:

THEOREM 4.5. *For each n there are exactly two different GSE graphs that can be folded n times.*

These are called *the maximally foldable* GSE graphs. We return to an algebraic description of them in the next section.

5. Linear GSE networks. Whereas the previous section showed that there are a tremendous number of different GSE graphs for each n, we will in this section restrict our attention to a much smaller class of networks, which can be studied in terms of linear recursion theory.

A *linear homogenous* boolean function is a function

$$f((g_{n-1},\dots,g_0)) = \sum_{i=0}^{n-1} c_i \cdot g_{n-1-i} \ ,$$

where additions and multiplications are modulo 2. A linear inhomogenous boolean function is defined as

$$\bar{f}(g) = f(g) \oplus 1 \ ,$$

where f is homogenous. A *linear* shift register is a shift register where f is linear (homogenous or inhomogenous), and we define a *linear GSE network* similarly. The mathematical theory for linear shift registers is rich, and a lot is known about their structure (see [4, 16]). The dynamics of a linear shift register is most easily studied by introducing the *characteristic polynomial* defined for a homogenous f as:

$$r(x) = \sum_{i=0}^{n} c_i \cdot x^i \ , \tag{10}$$

where $c_n = 1$. (All polynomials are over the binary field $GF[2]$). For the results in this paper, we do not need to define the characteristic polynomial for the inhomogenous case, and we will use a bar over the characteristic polynomial to indicate that it corresponds to an inhomogenous recursion. The networks are henceforth written in terms of the characteristic polynomial as GSE($r(x)$) or GSE($\overline{r(x)}$).

For homogenous f the following results are easily derived from Section 4:

- f is regular $\Leftrightarrow$ $c_0 = 1$.
- f is self complementary $\Leftrightarrow$ an even number of coefficients c_i equals 1 $\Leftrightarrow$ $r_f(1) = 0 \Leftrightarrow (x+1)$ divides $r_f(x)$.

The same holds for $f = \bar{f} \oplus 1$ if $\bar{f}$ is inhomogenous. Lifting (and lowering) takes a particularly simple form (verified by a direct computation):

LEMMA 5.1. *Let f be derived from h by a lift, as in (8). Then $r_f(x) = r_h(x)(x+1)$.*

COROLLARY 5.2. *The maximally foldable GSE graphs are given as GSE($\sigma_n(x)$) and GSE($\overline{\sigma_n(x)}$), where the characteristic polynomial $\sigma_n(x)$ is given as*

$$\sigma_n(x) = (1+x)^n = \sum c_i^n \cdot x^i \ \textit{where} \ c_i^n = \binom{n}{i} \bmod 2 \ .$$

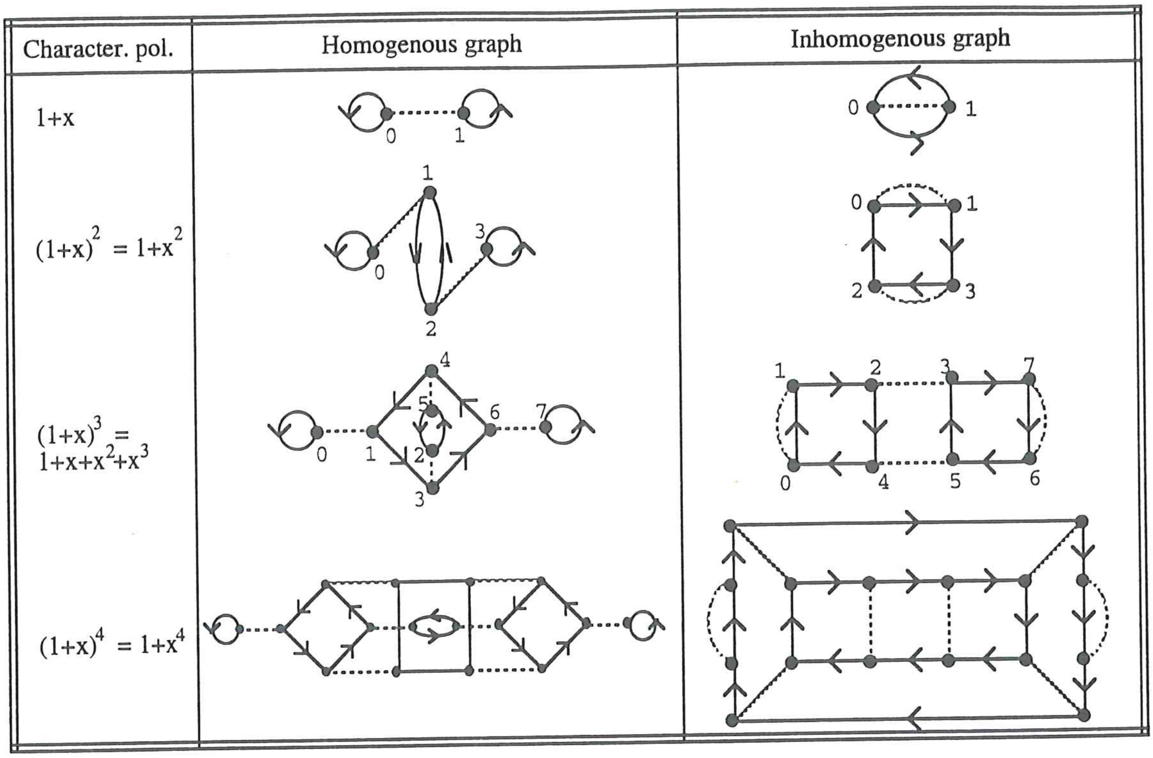

FIG. 2. *The two families of maximally fodable GSE graphs.*

The first of these graphs is shown in Figure 2. If $n = 2^t$ it is known that $\binom{n}{i} = 0 \bmod 2$ for $i \notin \{1, n\}$, hence

$$(1 + x)^{2^t} = 1 + x^{2^t}$$

showing that GSE(σ_{2^t}) $= SE(2^t)$, the classical SE net. The functions $\overline{\sigma_n}$ for $n = 2^t$ represent shuffles

$$S_{\overline{\sigma_n}}(g_{n-1}, \dots, g_0) = (g_{n-2}, \dots, g_0, \overline{g_{n-1}}) \ .$$

It can be verified that the shuffle–orbits for $\overline{\sigma_n}$, $n = 2^t$ are all of equal length $2n = 2^{t+1}$. (The possible periods must be $p = 2^i$, but if we assume $i < t$ we will get contradictions $g_j = \overline{g_j}$ for the components of g). It is further evident that n shuffles maps g to g^*. This shows that if GSE($\overline{\sigma_{2^t}}$) is folded one time, the periods of each orbit must be halved. If we continue folding, the periods cannot be halved before we reach $n = 2^{t-1}$, since we know that this level has periods $p = 2^t$. Thus, we have proved the following:

THEOREM 5.3. *For all $n \in \{1, 2, 3, \dots\}$ the shuffle orbits of GSE($\overline{\sigma_n}$) have the the same length:*

$$p = 2^{\lfloor \log_2(n) \rfloor + 1} \ .$$

This theorem is found in a different context in [3], and can be derived from more general results given in [16, 19].

Lemma 5.1 indicates that factorizations of the characteristic polynomial is important for folding the GSE graphs to 'nice' layouts. In the remaining part of this section we elaborate upon this, showing that *any* factorization of $r(x) = s(x)\cdot t(x)$ can be used to fold the graph GSE($r(x)$) onto GSE($s(x)$). The proof of this requires some additional formalism.

Instead of working with the 'state' vector $g = (g_{n-1}, \dots, g_0)$ as we have done up to now, we will use *shift register sequences* defined as the (periodic) infinite sequences of bits the shift register can produce:

$$\{a_r\} = \{\dots, a_i, a_{i-1}, \dots, a_1, a_0\} ,$$

thus n consecutive bits from $\{a_i\}$ define the node in the graph. The characteristic polynomial $r(x)$ in (10) defines the following recurrence relation on $\{a_i\}$:

$$\sum_{i=0}^{n} c_i \cdot a_{r-i} = 0 .$$

A 1–1 correspondence between shift register sequences and nodes in the graph is obtained by associating the node $g = (g_{n-1}, g_{n-2}, \dots, g_0)$ with the sequence $\{a_r\}$ with *initial conditions*:

$$\{a_{-1} = g_{n-1}, a_{-2} = g_{n-2}, \dots, a_{-n} = g_0\} .$$

The sequence associated with $g = (0, 0, \dots, 1)$ is called the *pulse response.* We let $\Omega(r(x))$ denote the set of shift register sequences generated by $r(x)$. The following facts are proved in [16]:

- $\Omega(r(x))$ forms a 2^n dimensional vector space over the binary field $GF[2]$ (i.e. $(a_i), (b_i) \in \Omega(r(x)) \Rightarrow (a_i \oplus b_i) \in \Omega(r(x))$).
- The pulse response and its first $n-1$ shifts spans $\Omega(r)$.
- $\Omega(s(x)) \subset \Omega(s(x)\cdot t(x))$.
- If $s(x)$ and $t(x)$ are relative prime then $\Omega(s(x)\cdot t(x)) = \Omega(s(x)) + \Omega(t(x))$.

For each $\{a_i\} \in \Omega(r(x))$ define the *generating function* as:

(11) $$G(x) = \sum_{i=0}^{\infty} a_i \cdot x^i .$$

It is a fundamental fact that if the initial state $\{a_{-1}, a_{-2}, \dots, a_{-n}\} = g$ is given, then [4]:

(12) $$G(x) = \frac{p_g(x)}{r(x)} ,$$

where $p_g(x)$ is a polynomial of degree $< n$, depending on the initial state. If $g(x) = g_{n-1}x^{n-1} + \dots + g_0 x^0$ it can be shown that:

(13) $$p_g(x) = \text{polynomial\_part\_of}\ (g(x)\cdot r(x)/x^n) .$$

Equations (12) and (13) define an isomorphism between GSE($r(x)$) and rational functions $p(x)/r(x)$, where $\deg(p(x)) < \deg(r(x))$. If $\deg(p(x)) \geq \deg(r(x))$, the coefficients of the power series of $p(x)/r(x)$ and $(p(x) \bmod r(x))/r(x)$ are eventually equal, and we define the equivalence

$$\frac{p(x)}{r(x)} = \frac{q(x)}{r(x)} \text{ if } p(x) = q(x) \bmod r(x) .$$

From (11) we see that the shuffle is given as

$$S(p(x)/r(x)) = \frac{x\cdot p(x) \bmod r(x)}{r(x)} \ .$$

The pulse response is represented by $1/r(x)$, hence the exchange must be given as

$$E(p(x)/r(x)) = (p(x)+1)/r(x) \ .$$

Theorems 5.4, 5.6 and 5.7 are stated for the homogenous case. For the inhomogenous case, see comments after Theorem 5.7.

THEOREM 5.4. *Let $r(x) = s(x)\cdot t(x)$. Define the function*

$$\phi_t\left(\frac{p(x)}{r(x)}\right) = \frac{p(x) \bmod t(x)}{t(x)} \ .$$

Then ϕ_t defines a homomorphism of GSE$(r(x))$ onto GSE$(t(x))$. The kernel of the homomorphism is given as

$$Ker(\phi_t) = \Omega(s(x)) \ .$$

If $\gcd(s(x),t(x)) = 1$ *then* $\phi_t|_{\Omega(t)}$ *is 1–1.*

Proof: ϕ_t is clearly onto. Furthermore:

$$\begin{aligned} \phi_t\left(\frac{x\cdot p(x) \bmod r(x)}{r(x)}\right) &= \frac{(x\cdot p(x) \bmod s(x)\cdot t(x)) \bmod t(x)}{t(x)} \\ &= \frac{x\cdot p(x) \bmod t(x)}{t(x)} \Rightarrow \phi_t \circ S_r = S_t \circ \phi_t \end{aligned}$$

and

$$\phi_t\left(\frac{p(x)+1}{r(x)}\right) = \frac{1+p(x) \bmod t(x)}{t(x)} \Rightarrow \phi_t \circ E_r = E_t \circ \phi_t \ .$$

Since

$$\phi_t\left(\frac{p(x)}{s(x)}\right) = \phi_t\left(\frac{p(x)\cdot t(x)}{r(x)}\right) = 0$$

we find that $\mathrm{Ker}(\phi_t) = \Omega(s(x))$. If $r(x)$ and $s(x)$ are relative prime, then $\Omega(r) = \Omega(s) + \Omega(t)$ thus counting dimensions yields that ϕ_t must be 1–1 on $\Omega(t)$.

□

It is easy to express ϕ_t in terms of the state vector representation. The following lemma can be shown by inspecting the generating function $G(x)$:

LEMMA 5.5. *Let r, s, t and ϕ_t be as in Theorem 5.4, where*

$$s(x) = \sum_{j=0}^{k} s_j\cdot x^j \ .$$

If $\phi_t((g_{n-1},\ldots,g_0)) = (h_{n-k-1},h_{n-k-2},\ldots,h_0)$ *then*

$$h_{i-k} = \sum_{j=0}^{k} g_{i-j}\cdot s_j \ \textit{for}\ i \in \{k,k+1,\ldots,n-1\} \ .$$

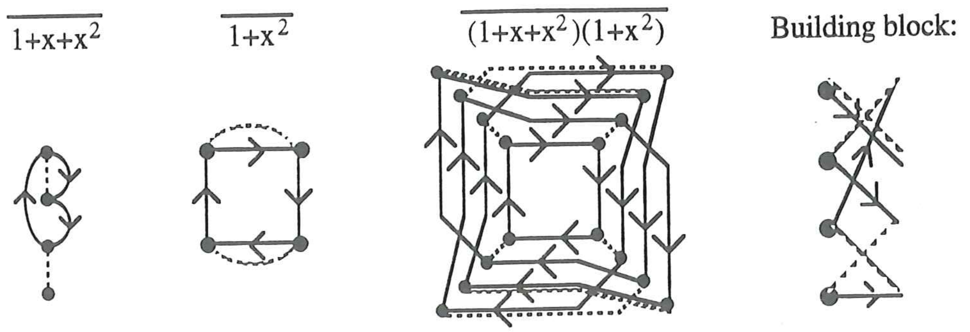

FIG. 3. *Graph represented as the direct product of relatice prime polynomials, as in Theorem 5.6.*

Example: $s(x) = (1 + x)$ yields $h_{i-1} = g_i \oplus g_{i-1}$, as we found in (5).
Example: $s(x) = (1 + x + x^3)$ yields $h_{i-3} = g_i \oplus g_{i-1} \oplus g_{i-3}$.

We will now use this theorem to find GSE graphs with regular layouts. This is done by letting the cosets of $\mathrm{Ker}(\phi_t)$ define supernodes in the graph. Theorem 5.4 guarantees that the supernode graph is isomorphic to GSE($t(x)$). We also want to compute the local dynamics within the supernodes. The most symmetric case is when $r(x)$ can be factored in two relative prime factors:

THEOREM 5.6. *Let* $r(x) = s(x) \cdot t(x)$ *where* $\gcd(s(x), t(x)) = 1$. *Then every node* $g_r \in \Omega(r(x))$ *can be uniquely identified with a pair of nodes*

$$g_r = (g_s, g_t) \text{ where } g_s \in \Omega(s(x)) \text{ and } g_t \in \Omega(t(x)),$$

such that shuffle and exchange act according to:

$$E_r(g_r) = (E_s(g_s), E_t(g_t))$$

and

$$S_r(g_r) = (S_s(g_s), S_t(g_t)) .$$

In other words: GSE($r(x)$*) is the direct product of GSE(*$s(x)$*) and GSE(*$t(x)$*).*
Proof: Since $\Omega(r) = \Omega(s) + \Omega(t)$, each $g_r \in \Omega(r)$ can be uniquely written as $g_r = \tilde{g}_s + \tilde{g}_t$. Let ϕ_s and ϕ_t be defined as in Theorem 5.4. Let $g_s = \phi_s(g_r) = \phi_s(\tilde{g}_s)$, and $g_t = \phi_t(g_t) = \phi_t(\tilde{g}_t)$. Then this theorem follows immediately from Theorem 5.4.

□

We may think of g_t as a global coordinate for g_r (supernode address) and g_s as the local coordinate within each supernode. The application of this theorem is shown in Figure 3. In the general case it is not possible to achieve symmetry between global and local coordinates. We may, however, choose global coordinates as above, and local coordinates to achieve a high degree of regularity also here.

THEOREM 5.7. *Let* $r(x) = s(x) \cdot t(x)$. *Then every node* $g_r \in \Omega(r(x))$ *can be uniquely identified with a pair of nodes*

$$g_r = (g_s, g_t) \text{ where } g_s \in \Omega(s(x)) \text{ and } g_t \in \Omega(t(x)),$$

such that shuffle and exchange act according to:

$$E_r(g_r) = (g_s, E_t(g_t)) \tag{14}$$

and

$$S_r(g_r) = \begin{cases} (S_s(g_s), S_t(g_t)) & \textit{if leftmost bit in } g_t = 0 \\ (S_{\overline{s}}(g_s), S_t(g_t)) & \textit{else} \end{cases} \tag{15}$$

Proof: Let $g_r = p(x)/s(x) \cdot t(x)$. We split $p(x)$ as

$$p(x) = u(x) + t(x) \cdot v(x) \text{ where } \deg(u(x)) < \deg(t(x)) ,$$

and identify g with the pair $g = (g_s, g_t)$ where $g_s = v(x)/s(x)$ and $g_t = u(x)/t(x) = \phi_t(g_r)$. This identificaiton is clearly 1–1. Since $p+1$ splits as $p(x)+1 = (u(x)+1) + t(x) \cdot v(x)$ we get (14). Note that if $\deg(p(x)) \geq \deg(s(x) \cdot t(x))$, reduction modulo $r(x)$ is done as:

$$p(x) \bmod (s(x) \cdot t(x)) = u(x) + t(x) \cdot (v(x) \bmod s(x)) ,$$

so if $\deg(u(x)) < \deg(t(x)) - 1$ we find

$$(x \cdot p(x)) \bmod s(x) \cdot t(x) = x \cdot u(x) + t(x) \cdot (x \cdot v(x) \bmod s(x)) ,$$

yielding the upper part of (15). On the other hand, if $\deg(u(x)) = \deg(t(x)) - 1$ we get

$$x \cdot u(x) = (x \cdot u(x) \bmod t(x)) + t(x) = \tilde{u}(x) + t(x) ,$$

where $\deg(\tilde{u}(x)) < \deg(t(x))$, thus:

$$x \cdot p(x) \bmod s(x) \cdot t(x) = \tilde{u}(x) + t(x) \cdot ((v(x) + 1) \bmod s(x)) ,$$

yielding the lower part of (15).

□

Note: The local structure comes in only two different variants, and there are no local exchanges. Thus a network can be built up from only two basic building blocks, as shown in Figure 4.

Now to the inhomogenous versions of the above theorems. Since an inhomogenous shuffle can be written as $\bar{S} = E \circ S$, where S is homogenous, we arrive at the following modifications:

- Theorem 5.4: ϕ_t defines an homomorphism of GSE($\overline{r(x)}$) onto GSE($\overline{t(x)}$).
- Theorem 5.6 GSE($\overline{r(x)}$) is the direct product of GSE($\overline{s(x)}$) and GSE($\overline{t(x)}$).
- Theorem 5.7 GSE($\overline{r(x)}$) factores into the global structure of GSE($\overline{t(x)}$), and local structure given by S_s and $S_{\overline{s}}$ as in (15).

The theorems in this section extends readily to the case where $r(x)$ can be factorized into three or more factors.

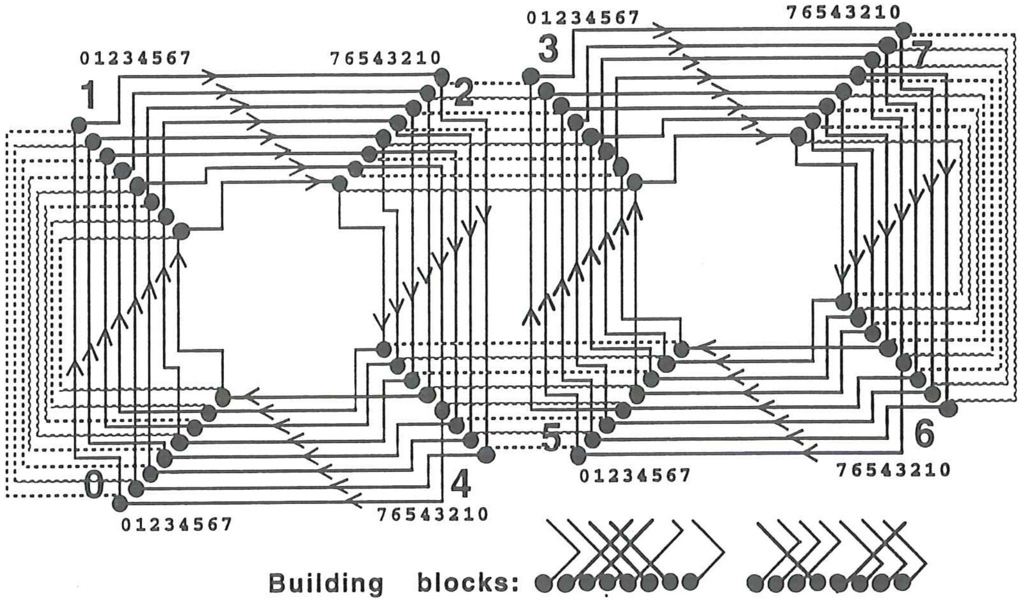

FIG. 4. *$GSE(\overline{(1+x)^6})$ represented as in Theorem 5.7, where $\overline{(1+x)^6} = \overline{(1+x)^3} \cdot \overline{(1+x)^3}$.*

6. Epilogue. We have a strong belief in the future for Shuffle-Exchange type networks in massively parallel computing. This is partly based on the optimal bound given in (1), partly on the fact that many important algorithms map elegantly onto SE nets (more easily than e.g. onto cube connected cycles) and partly also because of a close resemblance between the perfect shuffle and so called *mixing flows* or Axiom A flows in dynamical systems [5]. These are flows over infinite sets that are mixing the inputs at an optimal (i.e. exponential) rate. There are some generic properties found in almost any mixing flow: One of these is the existence of the *Small's horseshoe map* in the flow, which is a continuous analogue to the perfect shuffle. The other is the existence of *Markov partitions*, which are the continuous analogue of the sets $P_{i_1,\ldots,i_k}$ in Assumption 1. *Thus Shuffle-Exchange type networks are the natural discrete analogue of continuous mixing flows.* This gives the SE type networks a philosophical appeal, as the 'correct way' of mixing data.

A major purpose of this paper has been to show that SE-type networks are also very attractive from a hardware designers point of view, and to tie the bonds between parallel computing and the shift register art. Since the theory of shift registers appears to be fairly unknown in the field of parallel computing, it has been a goal to keep the discussion at a self contained level.

The theory in this paper has several interesting applications that has not been addressed. One is the derivation of routing and mapping algorithms for SE networks. Another is the construction of optimal layouts (in terms of area usage) for VLSI design of SE graphs. Still another is the partitioning of SE networks. It is known that the

SE graph cannot be partitioned in smaller SE graphs [17]. Theorems 5.6 and 5.7 do, however, show that the generalized graphs can be made partitionable by adding a small number of additional edges. These issues will be addressed in forthcomming papers.

7. Acknowledgements. The author arrived at the results in this paper motivated by problems in parallel computing, and was surprised to find a strong connection to the Coding Theory group in his immediate neighborhood. He would like to thank Prof. Tor Helleseth for pointing to the relevant references in the shift register litterature.

REFERENCES

[1] Beneš V.E, *Mathematical Theory of Connecting Networks and Telephone Traffic*, New York: Academic Press 1965.
[2] Bermond J-C., Peyrat C.: *The deBruijn and Kautz networks: a competiton for the hypercube?*, CNRS, Paris, 199?.
[3] Fredricksen H.: *Lechtures on Recursion*, unpublished lechture notes, University of Bergen, 1970.
[4] Golomb S.W.: *Shift Register Sequences*, Holden-Day, San Francisco 1967 (224 pp.).
[5] Guckenheimer J.: *A Brief Introduction to Dynamical Systems*, Lectures in Appl. Math. Vol. 17, pp.187-253, 1979.
[6] Huang S-T, Tripathi S.K.: *Self-Routing Technique in Perfect-Shuffle Networks Using Control Tags*, IEEE Trans. Comput., vol. 37, pp. 251-256, no. 2 Feb. 1988.
[7] Kleitman A., Leighton F.T., Lepley M., Miller G.L.: *An Asymptotically Optimal Layout for the Shuffle–Exchange Graph*, J. of Computer and Systems Sciences 26, 339-361 (1983).
[8] Leighton F.T.: *Introduction to Parallel Algorithms and Architectures*, Morgan Kaufman Publ., San Mateo, Calif. 1992. (831 pp).
[9] Lenfant J.: *Parallel Permutations of Data: A Benes Network Control Algorithm for Frequently Used Permutations*, IEEE Trans. Comput., vol. C-27,pp. 637-647, no. 7 July 1978.
[10] Munthe-Kaas H.: *Practical Parallel Permutation Procedures*, to appear.
[11] Nassimi D., Sahni S.: *A Self-Routing Benes Network and Parallel Permutation Algorithms*, IEEE Trans. Comput., vol. C-30,pp. 332-340, no. 5 May 1981.
[12] Parker S.D.: *Notes on Shuffle/Exchange-Type Switching Networks*, IEEE Trans. Comput., vol. C-29,pp. 213-222, no. 3 March 1980.
[13] Pradhan D.K., Samtham M.R.: The deBruijn multiprocessor network: A versatile parallel processing and sorting network for VLSI, IEEE trans. on comput. 38 (1989).
[14] Pradhan D.K., Kodandapani K.L: *A Uniform Representation of Single and Multistage Interconnection Networks Used in SIMD Machines*, IEEE trans. on Comp., vol. C-29, no. 9, 1980.
[15] Preparata F.P., Vuillemin J.: *The Cube-Connected Cycles: A versatile network for parallel computation*, Comm. ACM vol. 24 no. 5, 1981.
[16] Selmer E.S.:*Linear Rercurrence Relations over Finite Fields*, (mimeograph) Dept. of Mathematics University of Bergen, Norway, 1966, (212 pp.).
[17] Siegel H.J.: *The Theory Underlying the Partitioning of Permutation Networks*, IEEE Trans. on Computers, vol. C-29, No. 9, 1980.
[18] Stone H.S.: *Parallel Processing with the Perfect Shuffle*, IEEE trans. on comput. C-20, 2, (Feb 1971).
[19] Ward M.: *The Arithmetical Theory of Linear Recurring Series* Trans. Amer. Math. Soc. 33 (1931), pp. 153-165.
[20] Wu C., Feng T.: *The Universality of the Shuffle-Exchange Network*, IEEE Trans. Comput., vol. C-30, pp. 324-331, May 1981.

REPORTS IN INFORMATICS

"Reports in Informatics" is a series of publications from the University of Bergen, Norway, containing material from computer science, numerical mathematics, software engineering and other parts of informatics.

The purposes of the series are several:

1. to be used as study material in seminars, graduate education etc.
2. to provide a faster way of publication than through journals
3. to allow a more detailed expositon than in normal publications
4. to document results of technical or development work, such as software systems

Most of the texts will be in English.

The reports will be exchanged with similar publications from other institutions having corresponding interests.

Order for copies and other correspondance should be sent to:

Department of Informatics
University of Bergen
Høyteknologisenteret
N-5020 Bergen
Norway

REPORTS IN INFORMATICS

1. **KHALID AZIM MUGHAL:**
INTERACTIVE CONSTRUCTION, SYNTAX ANALYSIS AND EDITING OF PROGRAMS.
OCTOBER 1981.

2. **PER OFSTAD:**
PROGRAM ENTRY AND INSPECTION SYSTEM (PEIS). A PROJECT PROPOSAL.
NOVEMBER 1981.

3. **STEIN W. WALLACE:**
ENUMERATION ALGORITHM IN LINEAR PROGRAMMING.
FEBRUARY 1982.

4. **JARLE BERNTSEN:**
NUMERICAL COMPUTATION OF A THREE-DIMENSIONAL INTEGRAL.
NOVEMBER 1982.

5. **TORLEIV KLØVE;**
L CODES ON ALPHABETS WITH TWO SYMBOLS.
APRIL 1983.

6. **JARLE BERNTSEN AND TERJE O. ESPELID:**
ON THE USE OF GAUSS QUADRATURE IN ADAPTIVE AUTOMATIC INTEGRATION SCHEMES.
APRIL 1983.

7. **JARLE BERNTSEN AND TERJE O. ESPELID:**
ON THE CONSTRUCTION OF MINIMUM POINT SYMMETRIC THREE DIMENSIONAL EMBEDDED INTEGRATION RULES FOR ADAPTIVE INTEGRATION SCHEMES.
JULY 1983.

8. **TORLEIV KLØVE:**
GENERALIZATION OF THE KORZHIK BOUND.
JANUARY 1984.

9. **TORLEIV KLØVE:**
CODES FOR ERROR CORRECTION AND DETECTION.
JANUARY 1984.

10. **MAGNE MYRTVEIT:**
HANDLING SEVERAL WINDOWS ON ALPHANUMERIC DISPLAYS.
MARCH 1984.

11. **JARLE BERNTSEN:**
ON THE SUBDIVISION STRATEGY IN NUMERICAL ADAPTIVE INTEGRATION OVER THE CUBE.
MAY 1984.

12. **SVERRE STORØY:**
ON THE RELATIVE RANKING OF COMPUTER SYSTEMS.
JULY 1984.

13. **TERJE O. ESPELID:**
ON THE CONSTRUCTION OF GOOD FULLY SYMMETRIC INTEGRATION RULES.
OCTOBER 1984.

14. **MAGNE HAVERAAEN:**
UNIFICATION OF THE LOOP-STRUCTURES.
FEBRUARY 1985.

15. **JARLE BERNTSEN:**
A TEST OF THE NAG SOFTWARE FOR AUTOMATIC NUMERICAL INTEGRATION OVER THE 3–CUBE.
MARCH 1985.

16. **JARLE BERNTSEN AND TERJE O. ESPELID:**
ON THE CONSTRUCTION OF HIGHER DEGREE THRE DIMENSIONAL EMBEDDED INTEGRATIONS RULES.
MAY 1985.

17. **JARLE BERNTSEN:**
CAUTIOUS ADAPTIVE NUMERICAL INTEGRATION OVER THE 3–CUBE.
JULY 1985.

18. **SVERRE STORØY:**
PARALLEL VERTEX ENUMERATION — A HOLISTIC APPROACH.
FEBRUARY 1986.

19. **ADI BEN-ISRAEL AND SVERRE STORØY:**
ON THE COMPUTATION OF BASIC FEASIBLE SOLUTIONS IN LINEAR PROGRAMMING.
JUNE 1986.

20. **JARLE BERNTSEN:**
A TEST OF SOME WELL KNOWN DIMENSIONAL GENERAL PURPOSE AUTOMATIC QUADRATURE ROUTINES.
JUNE 1986.

21. **TOR HELLESETH AND ØYVIND YTREHUS:**
NEW BOUNDS ON THE MINIMUM LENGTH OF BINARY BLOCK CODES OF DIMENSION 8.
SEPTEMBER 1986.

22. **SJUR D. FLÅM:**
STABILITY OF CONVEX PROGRAMS UNDER A DISTRIBUTED CONSTRAINT QUALIFICATION.
NOVEMBER 1986.

23. **TORLEIV KLØVE:**
BOUNDS ON THE SIZE OGF OPTIMAL DIFFERENCE TRIANGLE SETS.
DECEMBER 1986.

24. **JARLE BERNTSEN AND MORTEN SKOGEN:**
PROGRAMS FOR TESTING AUTOMATIC QUADRATURE ROUTINES.
FEBRUARY 1987.

25. **BENGT ASPVALL:**
LINEAR INEQUALITIES.
MARCH 1987.

26. **ØYVIND YTREHUS:**
CODE-BUSTER: A SOFTWARE TOOL FOR CHARACTERIZING ABSTRACT CODES.
MARCH 1987.

27. **JARLE BERNTSEN AND TERJE O. ESPELID:**
A PARALLEL GLOBAL ADAPTIVE QUADRATURE ALGORITHM FOR HYPERCUBES.
MARCH 1987.

28. **TOR SØREVIK AND TERJE O. ESPELID:**
FULLY SYMMETRIC INTEGRATION RULES FOR THE 4–CUBE.
AUGUST 1987.

29. **JARLE BERNTSEN:**
ADAPTIVE MULTIDIMENSIONAL QUADRATURE ROUTINES ON SHADED MEMORY PARALLEL COMPUTERS.
OCTOBER 1987.

30. **JARLE BERNTSEN:**
PRACTICAL ERROR ESTIMATION IN ADAPTIVE MULTIDIMENSIONAL QUADRATURE ROUTINES.
APRIL 1988.

31. **JARLE BERNTSEN, TERJE O. ESPELID AND ALAN GENZ:**
A TEST OF ADMINT.
SEPTEMBER 1988.

32. **PETTER MØLLER:**
BRUK AV STRUKTURELL INFORMASJON I FORM AV N-GRAM OG STATISTISK INFORMASJON I FORM AV EN MODIFISERT VITERBI-ALGORITME I KARAKTERGJENKJENNING.
OKTOBER 1988.

33. **TERJE O. ESPELID:**
INTEGRATION RULES, NULL RULES AND ERROR ESTIMATION.
DECEMBER 1988.

34. **JARLE BERNTSEN:**
TRITST: A SUBROUTINE FOR EVALUATING THE PERFORMANCE OF SUBROUTINES FOR AUTOMATIC INTEGRATION OVER TRIANGLES.
JANUARY 1989.

35. **BJØRNAR TESSEM:**
TRUTH MAINTENANCE IN INFINITELY-VALUED LOGIC.
MARCH 1989.

36. **KJELL J. OVERHOLT:**
THE P–ALGORITHMS FOR EXTRAPOLATION.
JUNE 1989.

37. **SIGURD MELDAL:**
EXTENDING CSP WITH TERMINATION INDEPENDENT PROCESS ACTIVATION.
AUGUST 1989.

38. **ØYVIND YTREHUS:**
A RATE 3/8 (1,3) CONSTRAINED CODE WITH FREE HAMMING DISTANCE 3.
SEPTEMBER 1989.

39. **BJØRNAR TESSEM:**
INTERVAL PROBABILITY PROPAGATION.
OCTOBER 1989.

40. **JARLE BERNTSEN AND TERJE O. ESPELID:**
A TEST OF DCUTRI AND TWODQD.
NOVEMBER 1989.

41. **TOR HELLESETH AND ØYVIND YTREHUS:**
HOW TO FIND A [33,8,14] CODE.
NOVEMBER 1989.

42. **BJØRNAR TESSEM:**
EXTENDING THE A/R ALGORITHM FOR INTERVAL PROBABILITY PROPAGATION.
DECEMBER 1989.

43. **BJØRNAR TESSEM:**
GRAPH PROPAGATION OF INTERVAL AND FUZZY TRUTH CONSTRAINTS.
FEBRUARY 1990.

44. **JARLE BERNTSEN AND TERJE O. ESPELID:**
DEGREE 13 SYMMETRIC QUADRATURE RULES FOR THE TRIANGLE.
MARCH 1990.

45. **MAGNE HAVERAAEN:**
DATA DEPENDENCIES AND SPACE-TIME ALGEBRAS IN PARALLEL PROGRAMMING.
(ALSO PUBLISHED AS YALE UNIVERSITY REPORT TR-784.)
JUNE 1990.

46. **JARLE BERNTSEN, RONALD COOLS AND TERJE O. ESPELID:**
A TEST OF DECUTET.
JUNE 1990.

47. **TORLEIV KLØVE:**
K-SEQUENCES, A GENERALIZATION OF SONAR SEQUENCES AND B-SEQUENCES.
JUNE 1990.

48. **MARIAN VAJTERSIC:**
MATRIX MULTIPLICATION ALGORITHMS FOR MATRICES OF THE SIZE $n \leq 128$ ON THE MASPAR PARAL-LEL COMPUTER.
AUGUST 1990.

49. **SVERRE STORØY:**
WEIGHTS IMPROVEMENT IN COLUMN AGGREGATION.
JANUARY 1991.

50. **TOR SØREVIK:**
TRIANGULAR SOLVERS FOR MASSIVELY PARALLEL SIMD COMPUTERS.
JUNE 1991.

51. **FREDRIK MANNE:**
REDUCING THE HEIGHT OF AN ELIMINATION TREE THROUGH LOCAL RECORDINGS.
JUNE 1991.

52. **HANS MUNTHE-KAAS:**
SUPER PARALLEL FFT's.
MAY 1991.

53. **P.BJØRSTAD, F.MANNE, T.SØREVIK AND M.VAJTERSIC:**
EFFICIENT MATRIX MULTIPLICATION ON SIMD COMPUTERS.
MARCH 1991.

54. **PETTER BJØRSTAD AND MORTEN D. SKOGEN:**
DOMAIN DECOMPOSITION ALGORITHMS OF SCHWARZ TYPE, DESIGNED FOR MASSIVELY PARALLEL COMPUTERS.
MARCH 1991.

55. **PETTER BJØRSTAD AND ERIK BOMAN:**
A NEW ALGORITHM FOR THE *SLALOM* BENCHMARK.
MAY 1991.

56. **PETTER BJØRSTAD, JON BRÆKHUS AND ANDERS HVIDSTEN:**
PARALLEL SUBSTRUCTURING ALGORITHMS IN STRUCTURAL ANALYSIS, DIRECT AND ITERATIVE METHODS.
JANUARY 1991.

57. **PETTER BJØRSTAD, JEREMY COOK, HANS MUNTHE-KAAS AND TOR SØREVIK:**
IMPLEMENTATION OF A SAR PROCESSING ALGORITHM ON MASPAR MP-1208.
OCTOBER 1991.

58. **KJELL JØRGEN HOLE:**
RUNLENGTH LIMITED ERROR CONTROL CODES OF HIGH RATES
FEBRUARY 1992.

59. **FREDRIK MANNE:**
AN ALGORITHM FOR COMPUTING A MINIMUM HEIGHT ELIMINATION TREE FOR A TREE.
FEBRUARY 1992.

60. **ØYVIND YTREHUS:**
CONSTRUCTIONS AND PROOFS OF NONEXISTENCE OF SOME CONVOLUTIONAL CODES.
APRIL 1992.