Cellular bulk transfer system
by Kalyan Kumar Roy

A thesis submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in Electrical Engineering
Montana State University

Abstract:
In this thesis the results of investigations on a cellular bulk transfer system from the viewpoint of its logical capabilities have been presented. The model adopted for the bulk transfer system consists of an input array, a mapping device, an output array and an output logic. The influence of such factors as flexibility of the mapping device, flexibility of output logic and parallelism of operation has been determined. The main results obtained are: the bulk transfer system can be made logically universal with a proper combination of output logic and maps. In realizing arbitrary logic, a trade-off among the number of mapping operations, number of independent maps and amount of logical flexibility in the output logic is possible. A least upper bound on the number of necessary transposition maps is derived for an output logic consisting of a flexible cellular cascade. The possibility of a set of bulk transfer units operating in parallel has been studied and the functions realizable in this manner have been characterized. An algorithm for test-synthesis of realizable functions has been presented.

CELLULAR BULK TRANSFER SYSTEM

by

KALYAN KUMAR ROY

A thesis submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Approved:

Paul E. Schlinch
Head, Major Department

A. Mukhopadhyay
Chairman, Examining Committee

J. Boerng
Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

March, 1970

ACKNOWLEDGEMENT

The author wishes to offer grateful thanks to Dr. Amar Mukhopadhyay for many discussions and suggestions during the course of this work. He is also grateful to Professor R.C.Minnick for many helpful suggestions during this period.

The financial support for the graduate study through the award of a research assistantship by the Department of Electrical Engineering, Montana State University, a teaching assistantship by the Department of Computer Science, University of Iowa and through National Science Foundation Grant nos. GJ 158 and GJ 723 is thankfully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this thesis the results of investigations on a cellular bulk transfer system from the viewpoint of its logical capabilities have been presented. The model adopted for the bulk transfer system consists of an input array, a mapping device, an output array and an output logic. The influence of such factors as flexibility of the mapping device, flexibility of output logic and parallelism of operation has been determined. The main results obtained are: the bulk transfer system can be made logically universal with a proper combination of output logic and maps. In realizing arbitrary logic, a trade-off among the number of mapping operations, number of independent maps and amount of logical flexibility in the output logic is possible. A least upper bound on the number of necessary transposition maps is derived for an output logic consisting of a flexible cellular cascade. The possibility of a set of bulk transfer units operating in parallel has been studied and the functions realizable in this manner have been characterized. An algorithm for test-synthesis of realizable functions has been presented.

Chapter 1

INTRODUCTION

1.1 Introduction

A cellular array is some geometrical arrangement of cells in one, two or three dimensions. Each cell in a cellular array has some logical property and it may also have some storage capability. The cells of an array have a uniform interconnection structure. Because of this, the logic designer is faced with a new kind of constraint in realizing arbitrary logic functions. With existing techniques in cellular logic[11] it is necessary to use either a complicated interconnection pattern among cells or a very large number of cells to realize arbitrary logic.

The idea of a bulk transfer of data originated from the problems of realizing arbitrary logic functions with cellular arrays. With a view to avoiding the complicated interconnection pattern, it has been proposed to transfer the data from one cellular array to another by some kind of transferring device. The advantage in doing this is that necessary interconnections can be realized by suitably transferring the inputs in the second array. Pursuing this line, a cellular bulk transfer system can be conceived which consists of two cellular arrays which may be referred to as input and output arrays. Each of these arrays is capable of containing data and possibly performing logic on it. Between the two arrays there is a data-transfer device which may transfer data,

accompanied by some kind of transformation. An example of a simple type of transformation is a permutation of the variables on the array. If necessary, the device may have the capability for more complex transformations This device may be given the general name 'mapping device'. The mapping may be applied in reverse direction and can be iterated a limited number of times. Data may be logically processed using the built-in logic in the cellular arrays, maps in the mapping device and if required, by a separate logic device.

A study of the characteristics of this system reveals that logical universality can be achieved by a suitable combination of map and logic. This leads to many interesting questions regarding its efficiency, effect of variation in structure and construction, capability, and practicality as a computing system and so on. The model of the bulk transfer system is related to the perceptron system studied by Minsky and Papert[13] but the major interest in the system is based on a realization that it may have an important place in future digital systems because of a possibility that improvement in the computing power may be achieved through a mode of operation in which the advantages of increased parallelism of operations, greater homogeneity of the hardware structure, intermixture of storage,

arithmetic and control operations have been combined.

The generality of the system and its potential capabilities may be visualized by considering the basic schemes of the parallel processors suggested recently and attempting to mirror their functions into the bulk transfer system. Most of these machines have utilized the basic concepts of two distinct parallel processor organization proposed by Unger[17] and Holland[7]. These machines will be briefly discussed in order to bring out functional similarities between these and the bulk transfer system.

1.2 The Bulk Transfer System in Relation to Some Parallel Processors

Unger Machine

In 1958 Unger described a stored program computer oriented toward spatial problems. The particular problem illustrated with this machine was pattern detection. The computer consists of a master control and a rectangular array of logical modules each of which can communicate with its nearest neighbors (Figure 1.2.1). The master control contains a clock, decoding circuits and a random access memory for storing instructions. It reads out instructions from memory, decodes them and sends out appropriate commands which go simultaneously to all the modules. Each module contains a one-bit

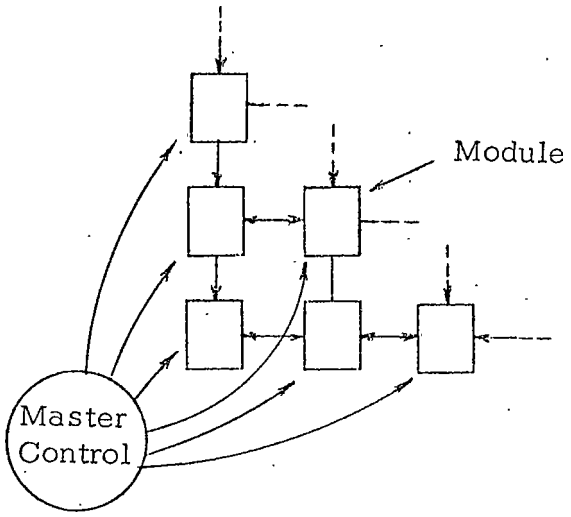

Figure 1.2.1 The Unger Machine

accumulator, some storage and associated logic and works in parallel with the remaining modules. With the use of an elementary instruction set Unger has described programs to detect certain local and global features of patterns on the two dimensional field.

To reflect these features of the Unger machine in the bulk transfer system, let us regard the field consisting of the modules as the input array and assume that the resulting configuration of the field on execution of one instruction will be on the output array. The instruction can be described by a proper combination of mapping with logic. Thus, an iterated procedure of mapping with logic may constitute a program for the determination of some particular feature of a pattern. It appears that the bulk transfer system can be potentially more powerful in some

respects than the Unger machine in view of the fact that suitable choice of maps in the mapping device may enable the bulk transfer system to process a larger part of the field than the group of immediate neighbors. Furthermore, some operations can be conveniently executed by a simple mapping where it may take several instructions in the Unger machine to do the same. On the other hand, there may be certain features in the Unger machine which may be difficult to implement by mapping in the bulk transfer system, but can be incorporated through the built-in logic of the input and output arrays.

Holland Machine

The Holland machine consists of a two-dimensional array of modules. Each module is a small general purpose computer and can be active or inactive at a given time. When active, a module treats the contents of its storage register as an instruction. After determining the location of the operand, a path is built to access the data. The instruction in the module is then executed and the active status is transferred to one of the four nearest neighboring modules in the array. Instructions can be arranged spatially throughout the array of modules with an arbitrary number of these executed at the same time.

Reflecting on the B.T. system, it appears that, though any data transformation possible in the Holland machine can be done by performing suitable mapping, a modification of the structure of the mapping device may be more suited to perform the kind of parallel computation that the Holland machine is able to do. Let the input array be the proper configuration containing data and instructions. The mapping device may be divided into many separate elements, each having the capability of scanning a limited area of any region in the input array. The execution of an instruction may consist of a transformation of data, first by mapping and then by the built-in logic of the arrays. If the operand of an instruction belongs to an area different from that scanned by a mapping element, a sequence of mapping operations may be undertaken to bring the data into the proper area, thus simulating the path-building and data-access procedure in the Holland machine. There could be other variations of this simulation procedure including a global mapping.

Two-dimensional Computing Technique

The bulk transfer system is structurally oriented toward parallel computation in the same manner as the two-dimensional iterative network computing technique of Hawkins and Munsey[7]. The computing machine devised by these authors consists of two data planes,

called the input plane and resultant plane (Figure 1.2.2). There is an intervening plane called the mapping mask. Data on the input plane can be processed through the use of the mask and projected onto the resultant plane. Using optical techniques and linear threshold logic to process spatially distributed data on the input plane, the authors have shown that the system is capable of recognizing certain tiny objects against the background of other large objects.

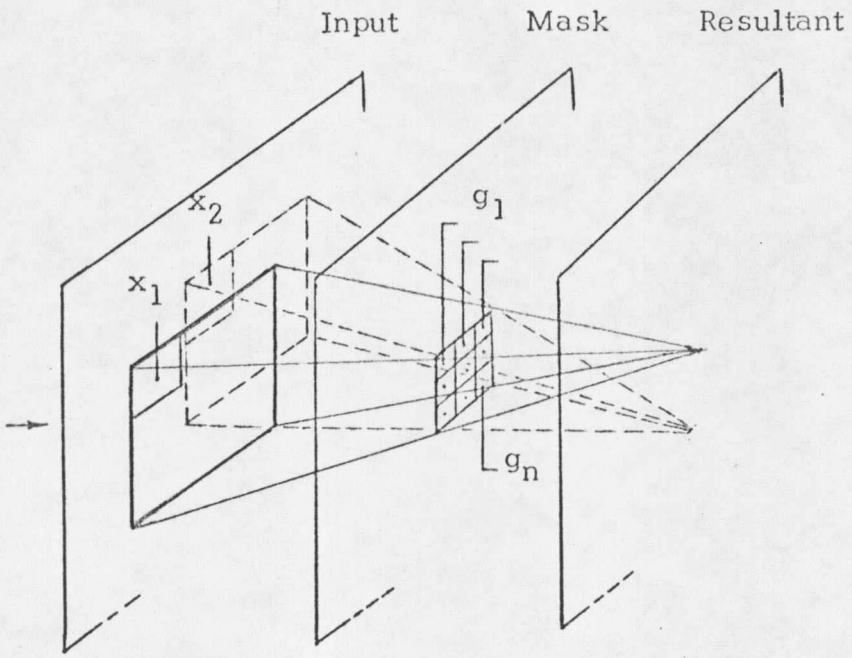

Figure 1.2.2 Optical Summation System
(x's are input variables, g's are transmittances)

From the above discussions it should be apparent that given the proper technological background, the bulk transfer system has the potentialities of becoming an improved, economic and versatile data-processing device. The technological advance in the field of batch fabrication technique has made it possible to produce reliable and economical arrays of logic cells on a mass scale [11]. Current research [1] on the use of optical techniques for data transfer on a large scale points to a direction in which the practical realization of a bulk transfer system may seem feasible.

With these aspects in view, a study on the logical capabilities of the bulk transfer system has been proposed in this thesis. The contents and organization of the thesis is given in the following section.

1.3 Organization of the Remaining Chapters

Chapter 2 discusses the logical capability of a simple bulk transfer system. The results obtained in this chapter form the basis of further development on this topic in Chapter 3. In this chapter the effect of limited flexibility in the output logic is studied and results on bulk transfer with cellular cascades are reported.

Chapter 4 deals with the characterization of realizable functions using parallel bulk transfer technique. It contains an algorithm for the test-synthesis of functions using parallel bulk transfer on the assumption that the partioning of the input array is fixed. In Chapter 5 a similar algorithm on the assumption of flexibility in the paritioning of the input array has been developed.

Chapter 6 gives a summary of the foregoing chapters followed by a discussion about the scope of further research work in the area.

Chapter 2

LOGICAL CAPABILITY OF A BULK TRANSFER SYSTEM

2.1 A Bulk Transfer System

An idea of the possible function of a bulk transfer system has been given in the previous chapter. Such a system will be described in this chapter and then investigations into its logical capabilities will be made. A block diagram of the system is shown in Figure 2.1.1. The block called Input Array may be conceived as a rectangular plane divided into many rectangular cells. Each cell is a storage device which can take on one of the two input values, 1 or 0. Corresponding to each input cell there is a binary variable and an assignment of input values to all the cells represents one of 2^n possible input configurations, where n is the number of cells in the input array.

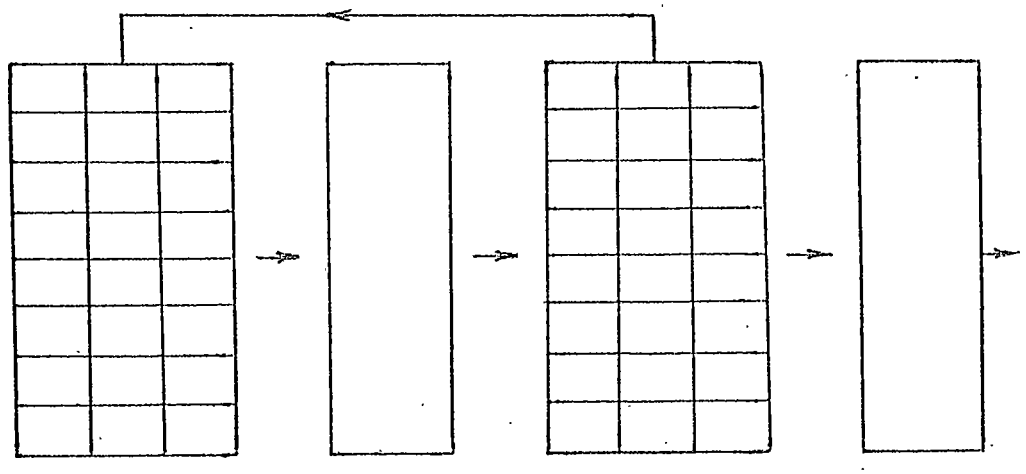

Figure 2.1.1 A Bulk Transfer System

A second component of the bulk transfer system is the block called the Output Array: it is a plane identical to the input array. The contents of the input cells are mapped (transferred) onto the cells of the output array. For example, a stored bit in a cell (i,j) in the input array may be mapped into a cell (m,k) in the output array.

The third component is called the Mapping Device: it implements the mapping operation between the input and output arrays. The mapping may include logic (Figure 2.1.2) or may not. A mapping is called logic-free if the value of an input bit is not changed during mapping. For the present discussion, such a mapping is always one-to-one; that is, each output cell gets a signal from at most one

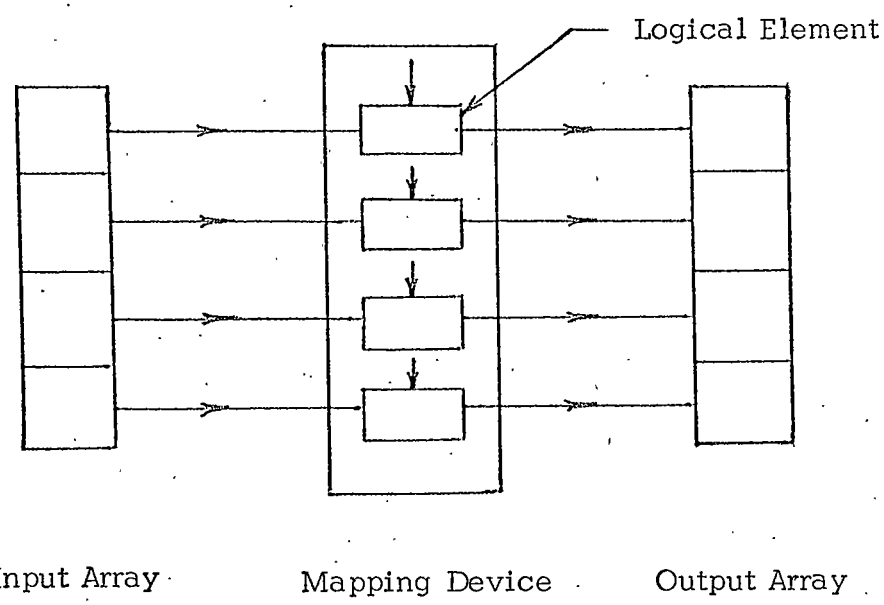

Figure 2.1.2 Mapping Device with Logic

input cell. (Note that all mappings have been assumed to be onto, that is, an output cell gets a signal from at least one input cell.) A logical mapping is one that involves transformation of the input bits during mapping (for example, complementation). Besides being one-to-one, a logical mapping may be many-to-one, where a number of input bits combine with some logic to produce a binary signal which is then mapped into an output cell.

After one mapping, the output array can be identically mapped back into the input array for a second mapping from input array to output array. Assume that the previous assignment of switching values to the cells in an array is automatically cleared when a new mapping to the array is performed. Also, assume that a mapping from the input array to the output array can be repeated any number of times by identically mapping back from the output array to the input array after each mapping from the input array to the output array. Henceforth, by "mapping" the mapping from input array to output array will be meant, unless otherwise specified, because the reverse mapping is assumed always to be the identity mapping.

The mapping device is capable of producing a limited number of independent mappings, none of which can be produced by any composition of other mappings that the device implements.

The final component of the bulk transfer system is called the Output Logic: it is a device that performs a logical function on the output array to produce an output value. Assume that during composite mapping this device does not operate until the end of composition. The logic of the device may be flexible so that different functions of its inputs can be obtained.

Note that an assignment of values to the input array of n cells represents a minterm of n variables which is converted into another minterm in the output array after mapping. Consider the set M of all minterms of n variables. Clearly, the mapping device can produce a mapping from M into M. Let a minterm m_1 after a mapping α be changed to another minterm represented by $m_1\alpha$. The same minterm when subjected to the map α twice, is changed to a minterm represented as $m_1\alpha^2$. If $m_1\alpha^2$ is different from $m_1\alpha$ and the values of the output function for the two minterms are different, then evidently, a new function (i.e., a function different from that obtainable after single mapping) is generated at the output by one repetition of the α -map.

The bulk transfer system may be looked upon as a device for producing different logical functions of a set of input variables,

using a set of maps ($\alpha_1, \alpha_2, \ldots \alpha_k$) and an output logic device. We shall study the logical capabilities of the system under different types of mapping and output logic. In this connection, some of the basic ideas and terminology of Group Theory[3] will often be used.

2.2 Logical Capability of the Bulk Transfer System

Permutation Maps

Let the set of 2^n minterms be arranged in some order as $(m_1, m_2, \ldots m_{2^n})$. If it were possible to produce all possible permutations of the set of minterms with the help of the limited number of independent maps of the bulk transfer (B.T.) system, then, with an output logic device of very limited flexibility, an extremely large number of functions could be generated, because, by permuting the set of minterms a different function could be obtained at the output without changing the output function of the output device. So the attempt in the following is directed toward this goal. Theorem 2.2.1 is a result of the foregoing discussion, and so is given without proof.

Theorem 2.2.1: A one-to-one map of the input array in a bulk transfer system produces a permutation of the set of minterms.

A logic-free mapping of the input array is effectively a geometrical repositioning of the input bits. This type of mapping has limitations

in the matter of producing arbitrary permutations of the set of minterms as the following theorem will show:

Theorem 2.2.2: With all possible logic-free one-to-one mappings of the input array, it is not possible to produce all permutations of the set of minterms.

Proof: A logic-free mapping cannot map a minterm of index number i (number of 1's in the minterm) into a minterm of index number j, where $i \neq j$. Therefore, such a mapping cannot produce all possible permutations. Q.E.D.

The types and the total number of permutations that all logic-free one-to-one maps produce will be given under Theorem 2.2.5. Suppose the set of minterms is identically mapped and the function produced by the output logic (fixed) is f. Can a different function be produced at the output by resorting to a different permutation of the set of minterms? Consider a permutation of the set of minterms where two minterms m_i and m_j are mapped one into the other while the rest are mapped identically. If the output logic is such that both m_i and m_j are true or both false, then, with this permutation, no new function can be produced. The condition for producing a new function is given in the following theorem.

Theorem 2.2.3: Let f be the function produced with identity mapping by a bulk transfer system having a fixed output logic. Then with any mapping of the input array which produces a permutation of the set of minterms, it is possible to produce a new function at the output if and only if the corresponding permutation of the set of minterms maps at least one true (false) minterm into a false (true) minterm.

Proof: The proof follows easily from earlier discussions and so is omitted.

Theorem 2.2.4 gives the maximum number of functions realizable by a bulk transfer system with a fixed map and a fixed output logic.

Theorem 2.2.4: A bulk transfer system with a single one-to-one map and with fixed output logic can realize at most k functions where k is the order of the corresponding permutation of the set of minterms.

Proof: Since with repeated applications of the map, only k different permutations of the set of minterms are produced, hence, if every permutation generated a new function, at most k functions could be produced. Q.E.D.

According to this theorem, out of 256 functions of three variables, at most 15 functions may be realized by a bulk transfer system with a single map and output logic. If there is no restriction about the number of logic-free maps that the mapping device can implement, then the maximum number of functions that a bulk transfer system with fixed output logic can realize, is given by the following theorem:

Theorem 2.2.5: A bulk transfer system with a fixed output logic and capable of producing all logic-free one-to-one maps can realize at most

$$\prod_{i=0}^{n} \binom{\binom{n}{i}}{a_i}$$ functions

where n = number of input cells,

i = index number of a minterm,

a_i = number of true minterms in the output logic function that have index number i, where $0 \leq a_i \leq \binom{n}{i}$

Proof: Let the set of 2^n minterms be divided into subsets such that a subset G_i contains all minterms of index number i. There will be (n+1) such subsets, i ranging from 0 to n. A subset G_i will contain $\binom{n}{i}$ members. If the output logic is such that only a_i minterms of

these $\binom{n}{i}$ minterms are true, then the number of possible combinations of these a_i minterms is $\binom{\binom{n}{i}}{a_i}$. Now, in logic-free mapping, the permutation of the set of minterms is restricted so that a minterm of index number i is mapped to another minterm of same index number. If we assume all possible permutations within the subset of minterms of same index number then the number of different functions that can be generated by permutations of the subset G_i is $\binom{\binom{n}{i}}{a_i}$. Now each of these ways of generating a function can be linked with each way of generating a function by the members of a different subset G_j. So the maximum number of functions possible is $\prod_{i=0}^{n} \binom{\binom{n}{i}}{a_i}$. Q.E.D.

It has been seen that the set of all logic-free one-to-one maps cannot produce all permutations of the set of minterms. With a set of logical maps, however, it is possible to produce all permutations. Given all possible permutations of M, it is still not possible to realize all functions of n variables unless the output logic is sufficiently flexible. The flexibility of the output logic is an essential factor in

the realization of arbitrary logic with a bulk transfer system capable of producing only permutation maps. The following theorem states this:

Theorem 2.2.6: In order to produce arbitrary functions of input variables, a bulk transfer system having the means to produce all permutations of the set of minterms must have an output logic of sufficient flexibility so that at least $(2^n + 1)$ different functions can be produced by the output logic device.

Proof: Let f_r be the function produced by the fixed output logic device of a B. T. system under identity map where r is number of true minterms. With all possible permutations of M, the total number of functions realized is $\binom{2^n}{r}$. With the fixed output logic and a set of maps to produce all permutations of M, it is not possible to change r, the number of true minterms.

So in order to realize arbitrary functions, the output logic must be flexible and capable of producing each function f_i, $i = 0,1,\ldots 2^n$. These functions are listed below:

(1) f_0 = false for all minterms

(2) f_1 = true for only one minterm

(3) f_2 = true for only two minterms

. . . . . .

$(2^r + 1)$ f_2r = true for only 2^r minterms

. . . . . .

$(2^n + 1)$ f_2n = true for all minterms

The total number of distinct functions that can be realized by the B.T. system in this case is $\binom{2^n}{0} + \binom{2^n}{1} + .. + \binom{2^n}{r} + .. + \binom{2^n}{n} = 2^{2^n}$ Q.E.D.

According to this theorem, the number of functions required of the output logic has been reduced to $(2^n + 1)$ from 2^{2^n} for systems without using bulk transfer, a value proportional to the logarithm of the original number. We can further improve upon this and drastically reduce the number of required functions of the output logic with the use of a special map to be discussed now.

Many-to-one Map

The preceding discussion was concerned with permutations of the set of minterms and, therefore, involved only those mappings of the input array that produced permutations. Consider the many-to-one logical maps that map the set of minterms onto one of its proper subsets. In a special case, a sufficient number of repetitions of such

a map may result in a situation in which all the minterms will be mapped onto a single minterm. Let $\emptyset$ be a mapping such that a minterm m_i is mapped into m_{i+1} except the last minterm m_{2^n} which is mapped into itself. The mapping $\emptyset^k$, $k \leq (2^n-1)$ (i.e., $\emptyset$ is repeated k-times) maps (k+1) minterms onto m_{2^n}.

If the mapping device of the bulk transfer system is capable of producing $\emptyset$ along with arbitrary permutation, and the output logic is fixed and produces only the function f_1 where f_1 = true, only when m_{2^n} is true, then every function listed in the proof of Theorem 2.2.6 except the zero function can be produced with the use of a suitable number of repetitions of the $\emptyset$-map. For example, $\emptyset$ applied thrice will generate the function f_4 (the function is true for four minterms). Therefore, the required flexibility of the output logic comes down to two functions:

1) f_0 (zero function)

2) f_1 (function is true for only one minterm, m_{2^n}).

It has been seen that to produce all possible permutations of the set of minterms, logical maps must be used. The following lemma gives the minimum number of independent logical maps necessary to achieve this.