ON THE USE OF ASHENHURST DECOMPOSITION CHART AS AN ALTERNATIVE TO ALGORITHMIC TECHNIQUES IN THE SYNTHESIS OF MULTIPLEXER-BASED LOGIC CIRCUITS

by

C.C. Osuagwu
Department of Electronic Engineering
University of Nigeria, Nsukka

ABSTRACT

The non-uniqueness of a multiplexer implementation of a Boolean function results from the partition of the function variables into data select and data input variables. This partition affects critically the cost and structure of the final multiplexer realisation. Hence the central problem in the synthesis of logic circuits with multiplexers is the selection of that partition which results in the desired structure or the most economical realisation of the function. In this paper, the Ashenhurst decomposition chart is shown to be a mapping technique which solves this selection problem and enables the design of logic circuits with desirable attributes using multiplexers. The Ashenhurst decomposition chart also serves as a bridging technique between the map based and algorithmic based digital design methods in the synthesis of logic circuits with multiplexers.

1. INTRODUCTION

A multiplexer is an MSI device that has a set of q data select inputs and a set of p data inputs such that $p = 2^q$. Fig. 1 shows a p to 1 (p - 1) multiplexer. The data select inputs act as addresses that determine which of the p data inputs is to be connected to the output. Commercially available multiplexers include 2-1, 4-1, 8-1 and 16-1 multiplexers.

Multiplexers are extensively used as data selectors in digital design. They are also used in the design of digital controllers as well as being used as universal gates in the synthesis of Boolean functions.

The central problem in the synthesis of combinational logic circuits using multiplexers can be stated as follows:
given an n variable Boolean function, how do we partition these variables into q data select inputs and p data inputs so as to realise a design with one of the following desirable attributes:

i) a maximum number of logic 1 and logic 0 connected to the multiplexer inputs;

ii) a minimum number of multiplexers to implement the design;

iii) the direct realisation of the p residue functions without the use of additional gates.

A number of algorithms have been reported in the literature to obtain the above realisations. Whitehead (l) proposed an optimum partitioning of the n variables into q data select and (n-q) data input variables such that the multiplexer input will have a maximum number of logic 1 and logic 0 connected to it. The Whitehead's algorithm results in a multiplexer realisation that simplifies the input bus structure. The objective of Ektare's algorithm (2) was to provide a method of synthesising Boolean functions using the minimum number of multiplexers. If this cannot be achieved, the

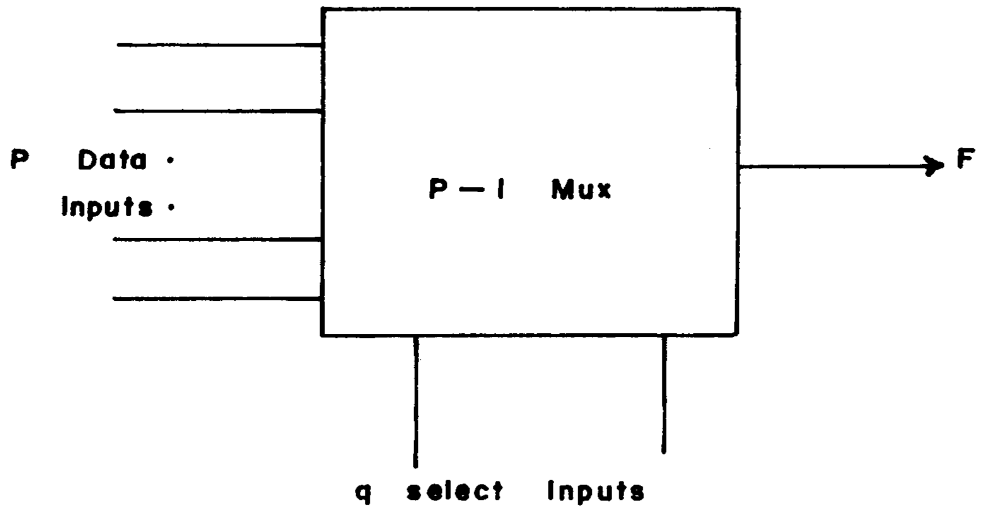


Fig. I : Block diagram of a multiplexer

algorithm reverts to the Whitehead design. Domindo and Canto (3) presented an algorithm which seeks to realise a logic function of n variables partitioned into q data select and (n-q) data input variables where n > q + 1 with a single multiplexer, such that the p residue functions can be synthesised directly without using additional gates.

The beauty of these algorithmic techniques is that the design can be carried out using computers. But these powerful computer aided design techniques are not routinely taught in undergraduate courses for a variety of reasons:

1. the original paper may be very difficult to understand because it was written for the benefit of experts (e.g. the Domindo and Canto paper).
2. a computer may not be available; and where available may not have been integrated in the teaching of the digital logic design course because of a lack of appropriate courseware.

Given the increasing importance of multiplexers in digital systems design, and the importance of understanding algorithmic techniques in the design of digital systems, it becomes necessary for students and practising engineers alike to become familiar with digital design techniques that provide a smooth transition from the familiar map based design techniques to computer based design methods.

The Ashenhurst decomposition chart, a mapping technique which exploits the properties of the familiar Karnaugh map to partition the function variables into two sets, is one such bridging technique. Langdon (4) used the technique to determine the data input conditions for a multiplexer realisation of a Boolean function.

The main objective of the paper is to show that using the Ashenhurst decomposition chart, logic functions can be implemented with multiplexers to achieve the same design aims as the algorithmic methods namely:

(i) a design that simplifies the input bus structure (Whitehead's algorithm).

(ii) a design that uses the minimum number of multiplexers (Ektare's algorithm).

(iii) a design in which the p residue functions are realised directly without the use of additional gates (Domindo and Canto algorithm).

2. FUNCTION FACTORISATION- BASIS OF MULTIPLEXER IMPLEMENTATION

Suppose we wish to implement a function which has the minterm list of Lofti (5) using an 8-1 multiplexer:

$F(A,B,C,D) = \sum m(6,7,9,11,13,14)$ (1)

The first step is to factor the cannonical sum of products form of the minterm list of equation 1 around the data select variables. An expansion of a Boolean function of n variables around any of its q variables where n > q is referred to as Shannon's Expansion Theorem. If in this example, the function variables (A,B,C) are used as data select variables, then expanding equation 1 around ABC gives:

Table 1: Data Input Condition Table.

Data select input	Data input
$\bar{A}\,\bar{B}\,\bar{C}$	0
$\bar{A}\,\bar{B}\,C$	0
$\bar{A}\,B\,\bar{C}$	0
$\bar{A}\,B\,C$	$1 = (\bar{D} + D)$
$A\,\bar{B}\,\bar{C}$	D
$A\,\bar{B}\,C$	D
$AB\,\bar{C}$	D
$A\,B\,C$	$\bar{D}$

$$F = \bar{A}BC\,(\bar{D} + D) + A\bar{B}\,\bar{C}\,(D) + ABC\,(D) + AB\bar{C}\,(D) + ABC\,(D) \qquad (2)$$

Equation 2 is now in a form where it can be implemented directly with a multiplexer since the eight data input conditions can be read directly as shown in table 1. The multiplexer implementation of this function is shown in Fig. 2. Note that the eight data input conditions, called the residues of the expansion of equation 1, take values from the set (0,1,$\bar{D}$ and D). In general, an 8-1 multiplexer can realise any function of four variables by eliminating the three variables used as data select inputs and using the remaining variable as the data input variable.

If only a 4-1 multiplexer is available and AB is used as the data select input, we obtain the expansion of equation 3:

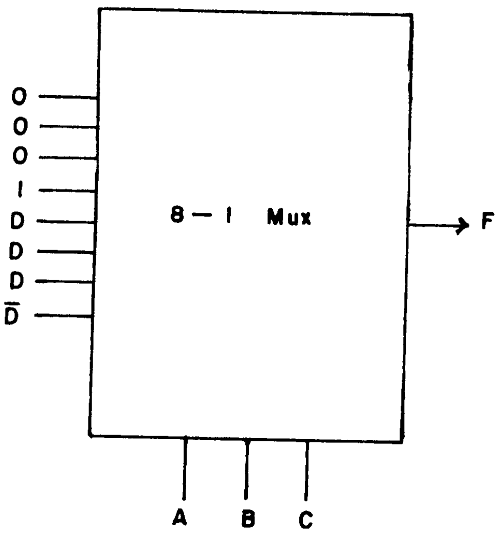


Fig. 2 : 8 — 1 Multiplexer Realisation of Eqn (1)

$$F = \bar{A}B\,(C) + A\,\bar{B}\,(D) + A\,B\,(C \oplus D) \qquad (3)$$

The implementation of this function is shown in Fig. 3, where it can also be seen that an Exclusive - OR discrete gate is required to implement one of the residues.

If we compare Fig. 2 with Fig. 3, we find that for the partition where the total number of variables n is given by n = q + 1 (Fig. 2), no gates are required to realise the residue functions. But for the case where n > q + 1, additional gates may be required to realise the residue functions. As we shall see in section 4.3, for more complex residue functions, it may be cheaper to realise the residue function using multiplexers rather than discrete gates. This realisation will be shown in section 4.4 to correspond to a successive application of Shannon's expansion theorem.

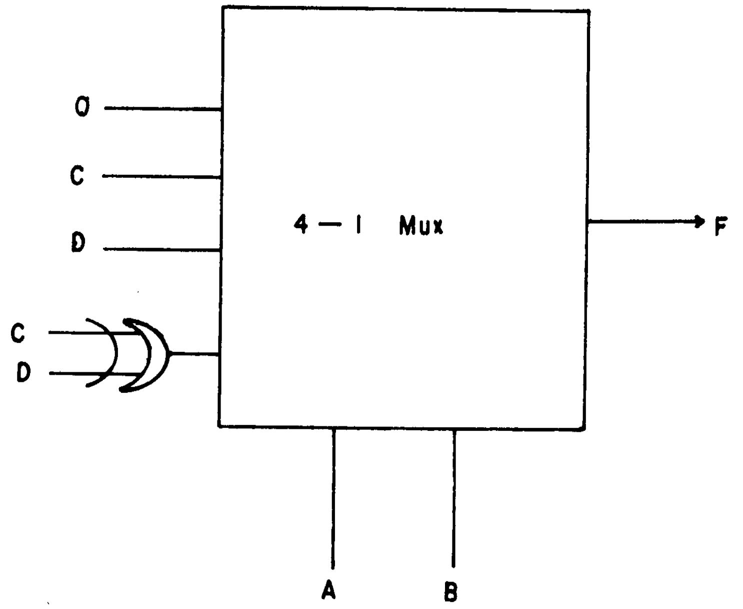


Fig. 3 : 4 — 1 Multiplexer Realisation of Equation 1.

It is clear from the results of this section that the structure and hence the cost of the multiplexer realisation of a Boolean function depends on which of the variables are selected as the data select and data input variables. There is, therefore, a need to examine all the possible partitions of n variables into q data select and (n - q) data input variables in order to make the optimum selection. The Ashenhurst decomposition chart is a simple and systematic way of doing this.

3. GENERATING THE ASHENHURST DECOMPOSITION CHART

Fig. 4 shows the k-map of the minterm list of equation 1.

Notice that the k-map partitions the function variables into two sets, A B and CD. Suppose we use AB, the variables running vertically down on the k-map, as the data select variables; and CD, the variables across the top of the k-map, as the data input variables. If we read the resulting k-map of Fig. 4 horizontally, we find that there are no minterms in the first row $\bar{A}\bar{B}$. Hence that row has a data input of logic 0 denoted as $\bar{A}\bar{B}$ (0). Similarly the second row, AB has value of C; the third row, AB, has value of $CD +$ CD or $(C \oplus D)$ and the last row A B has a value of D. The value of the k-map read horizontally is

$$F = \bar{A}\bar{B}(0) + \bar{A}B(C) + A\bar{B}(D) + AB(C \oplus D) \qquad (4)$$

Hence, the k-map of Fig. 4 read horizontally gives the same data input condition as the

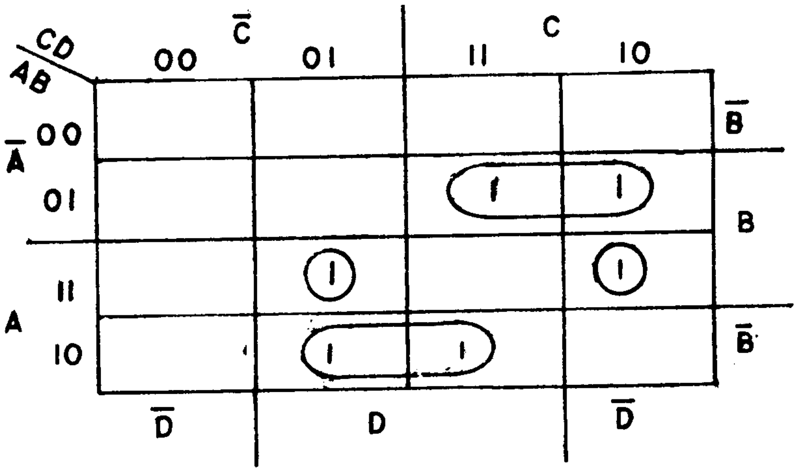


Fig. 4 : Karnaugh map of Equation 1 .

multiplexer realisation of Fig. 3. It is clear that a single k-map can be used to obtain two different multiplexer realisations. For if we read the k-map of Fig. 4 vertically (this corresponds to using C D as the data select and AB as the data input variables), the resulting multiplexer realisation becomes:

$F = \bar{C}\bar{D}(O) + \bar{C}D(A) + C\bar{D}(B) + CD(A \oplus B)$ (5)

The procedure for obtaining the Ashenhurst decomposition chart from the familiar k-map for a four variable function can be summarized as follows:

1. Label the four variable k-map such that the data select variables run vertically down the side of the map, and the data input variables run across the top.
2. Label each cell of the k-map keeping the original order of the minterms of the function. Then obtain the decimal equivalent of the cannonical minterms of each cell. The resulting k-map is an Ashenhurst decomposition chart for the chosen partition. (Using the above procedure, the Ashenhurst decomposition charts can be generated for all possible partitions of function variables into data select and data input variables) .

To use the chart to determine the multiplexer input conditions proceed as follows:

1. Circle the given minterm numbers of the function in the chart.
2. Read the map horizontally for the multiplexer data input conditions (or residues of the function).

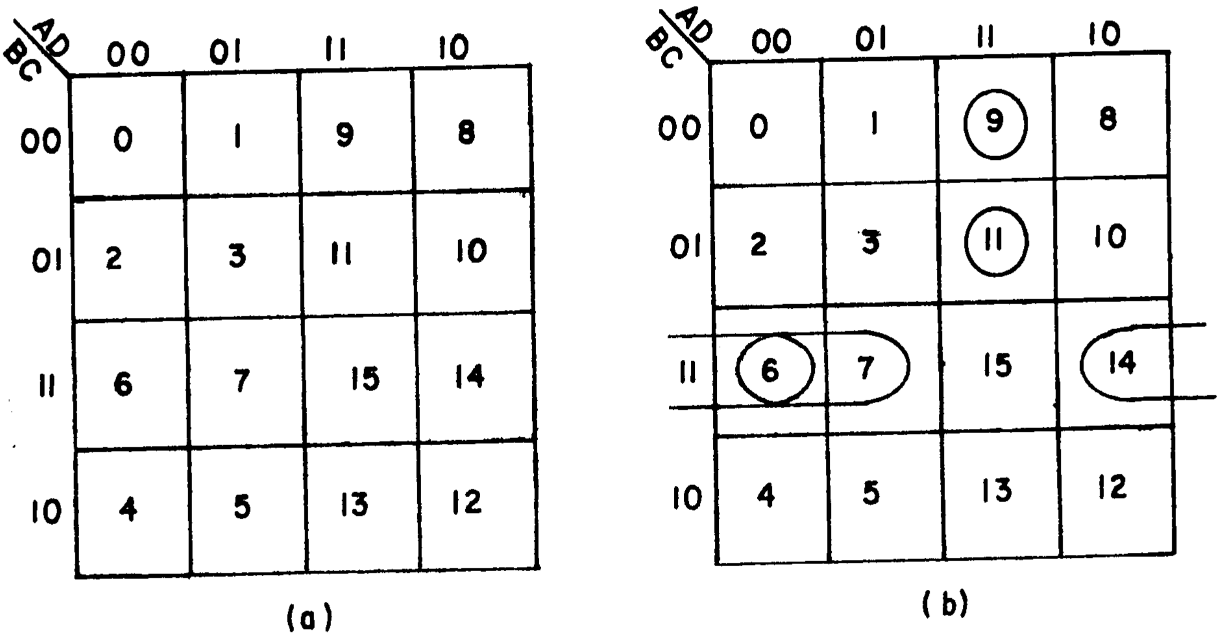
Suppose we wish to implement the minterm list of equation 1:

$$F(A, B, C, D) = \sum m(6,7,9,11,13,14)$$

with a multiplexer using BC as the data select variables. The Ashenhurst decomposition chart obtained using the procedure outlined above is shown in Fig. 5(a) while the data input conditions are read out horizontally from Fig. 5(b) as

$F = \bar{B}\bar{C}(AD) + \bar{B}C(AD) + B\bar{C}(AD) + BC(\bar{A} + \bar{D})$ (6)

The multiplexer which realises equation 6 is shown in Fig. 6. Note that common residues can be implemented from the output of a single gate. If we compare Fig. 3 (one residue gate) with Fig. 6 (two residue gates), we see clearly that the cost of a multiplexer realisation of a Boolean function depends on which of the variables are used as the data select variables



BC \ AD	00	01	11	10
00	0	1	9	8
01	2	3	11	10
11	6	7	15	14
10	4	5	13	12

(a)

BC \ AD	00	01	11	10
00	0	1	9	8
01	2	3	11	10
11	6	7	15	14
10	4	5	13	12

(b)

Fig. 5(a) Ashenhurst decomposition chart with BC as data select input.

Fig. 5(b) Residue function

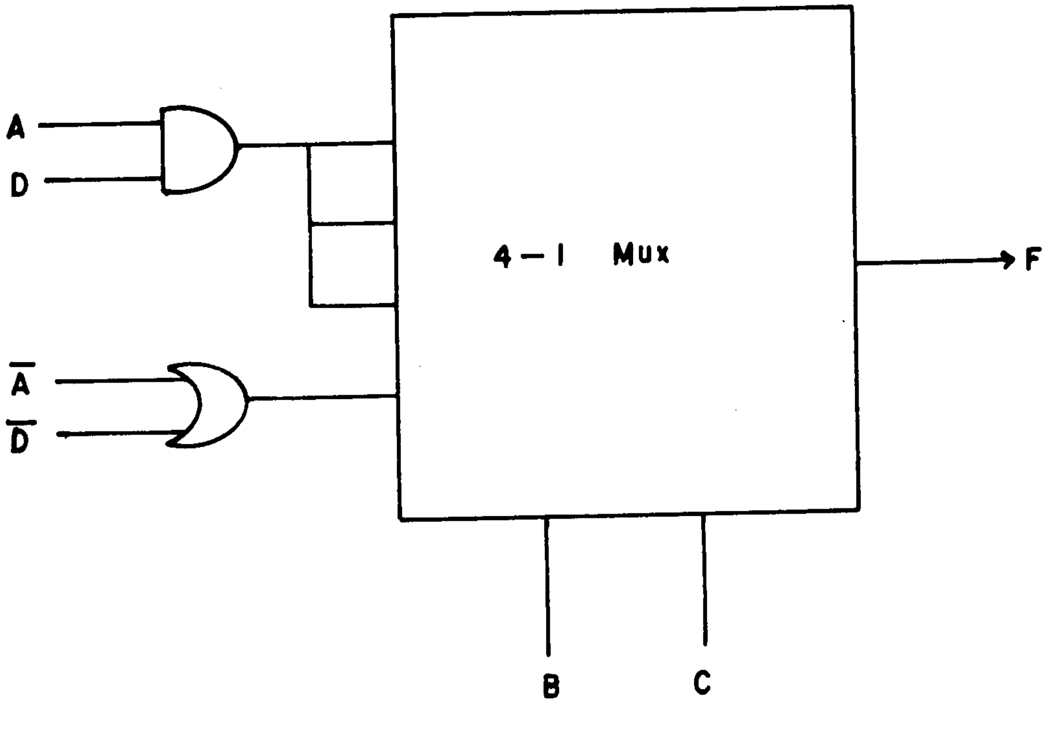


Fig. 6: 4 - 1 multiplexer realisation of equation 1 with BC as data select inputs.

4. EQUIVALENCE OF ASHENHURST DECOMPOSITION CHART WITH ALGORITHMIC TECHNIQUES

We wish to use the Ashenhurst Decomposition Chart in a multiplexer implementation of the minterm list of Lloyd (6), also used by Ektare (2) and listed below:

$$F(A,B,C,D,E) = \sum m(1\ ,2,6,9,10,12,13,14,17,21,25,\ 26,27,29,3\ 0,31) \qquad (7)$$

The objective is to use this mapping technique as an alternative to the algorithmic techniques in obtaining the following:

i) the Whitehead's structure (i.e a multiplexer design that simplifies the input bus structure)

ii) the Ektare structure (a minimum tree structure realisation).

iii) the Domindo and Canto structure (a multiplexer realisation without additional gates for cases were n > q + 1).

It is not necessary to generate the Ashenhurst decomposition charts since these charts are available in some text books (e.g. (7)). In the following sections, only partitions that result in the stated design objectives are discussed.

4.1 The Whitehead structure: A design that simplifies the input bus structure

The Whitehead structure applies in multiplexer design where n = q + 1. Table 2 shows the statistics of the data input conditions obtained by examining all the five possible Ashenhurst decomposition charts obtained by partitioning the five function variables into four data select variables and one data input variable.

As can be seen from table 2, the data input conditions (or residues) can only take values from the set 0, 1, data input variable and complement of data input variable. The multiplexer realisation that simplifies the input bus structure is the one that has the maximum number of logic 0 and logic 1 connected to its inputs. From table 2 the partition that yields this structure (the Whitehead's structure) is ABDE as data select variables and C as the data input variable. Fig. 7(a) shows the Ashenhurst decomposition chart for this partition and Fig. 7(b) shows the resulting Whitehead's structure. (The data input conditions are obtained by reading the chart vertically).

4.3 The Domindo and Canto structure

The Domindo and Canto structure is sought in multiplexer implementations where n > q + 1. The design objective for partitions where n > q + 1 is to select the partition which results in the cheapest possible realization of the (n - q) variable residue functions. The cheapest realisation of the function is obtained from that partition which results in the residue functions being implemented without additional gates (the Domindo structure), or where such a partition does not exist, to use one which results in the residue functions being synthesised with the least number of multiplexers (the Ektare structure) or the least number of gates.

Table 2: Statistics of all possible data input conditions for n = q + 1.

Data Select Variables	BCDE	ACDE	ABDE	ABCE	ABCD
Data Input Variables	A	B	C	D	E
Number of logic 1	5	5	7	3	3
Number of logic 0	5	5	7	3	3
Number of Variable and Variable	6	6	2	10	10

	AB	00				01				11				10			
	DE	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
C	0	0	①	3	②	8	⑨	11	⑩	24	㉕	㉗	㉖	16	⑰	19	18
	1	4	5	7	⑥	⑫	⑬	15	⑭	28	㉙	㉛	㉚	20	㉑	23	22

Fig. 7 (a): Ashenhurst dicomposition chart with ABDE as data select and C as data input variable.

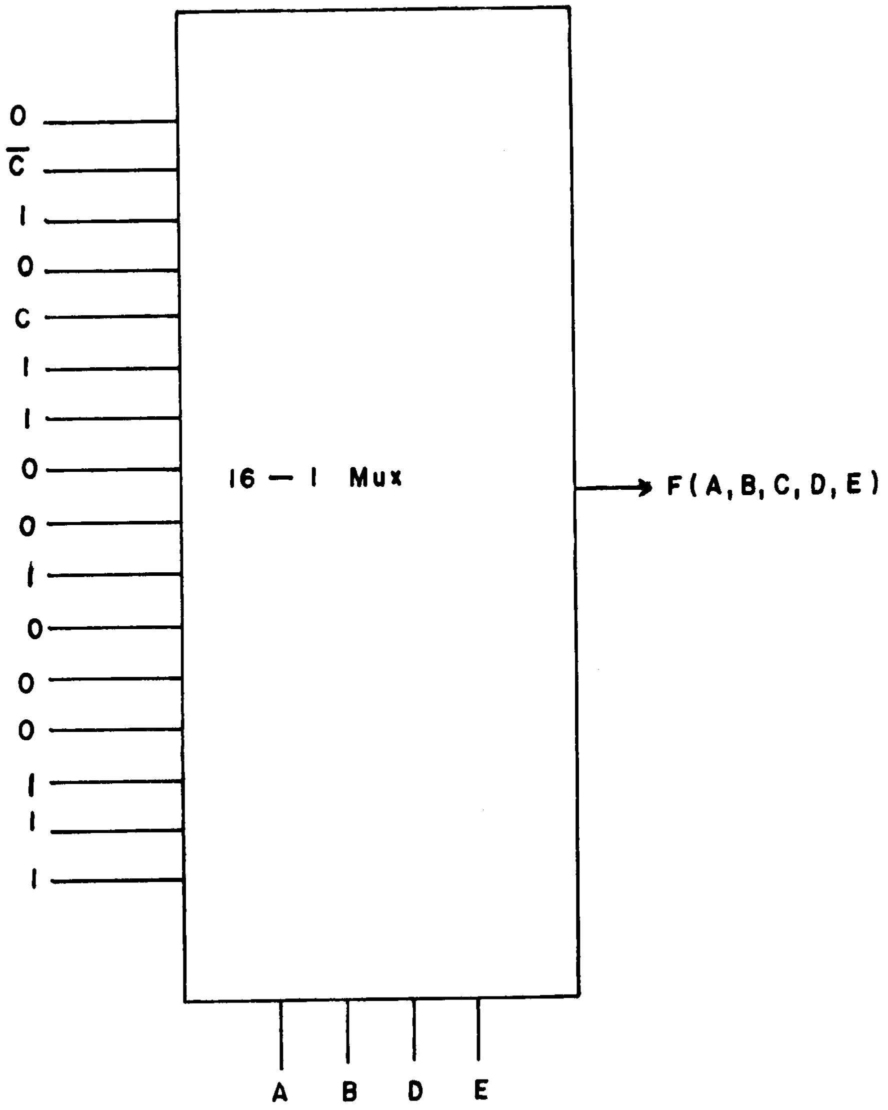


Fig. 7 (b): Whitehead's structure.

Table 3: All Possible Partitions of fie variables F(A,B,C,D,E) into 3 data select and 2 data input variables.

Select variables	CDE	BDE	BCE	BCD	ADE	ACE	ACD	ABE	ABD	ABC
Data input variables	AB	AC	AD	AE	BC	BD	BE	CD	CE	DE

Table 3 shows the ten possible partitions of five variables into 3 data select and 2 data input variables. If we use ABC as the data input variable, we obtain the multiplexer structure shown in Fig. 8. Note that the residue functions are realised using five additional gates.

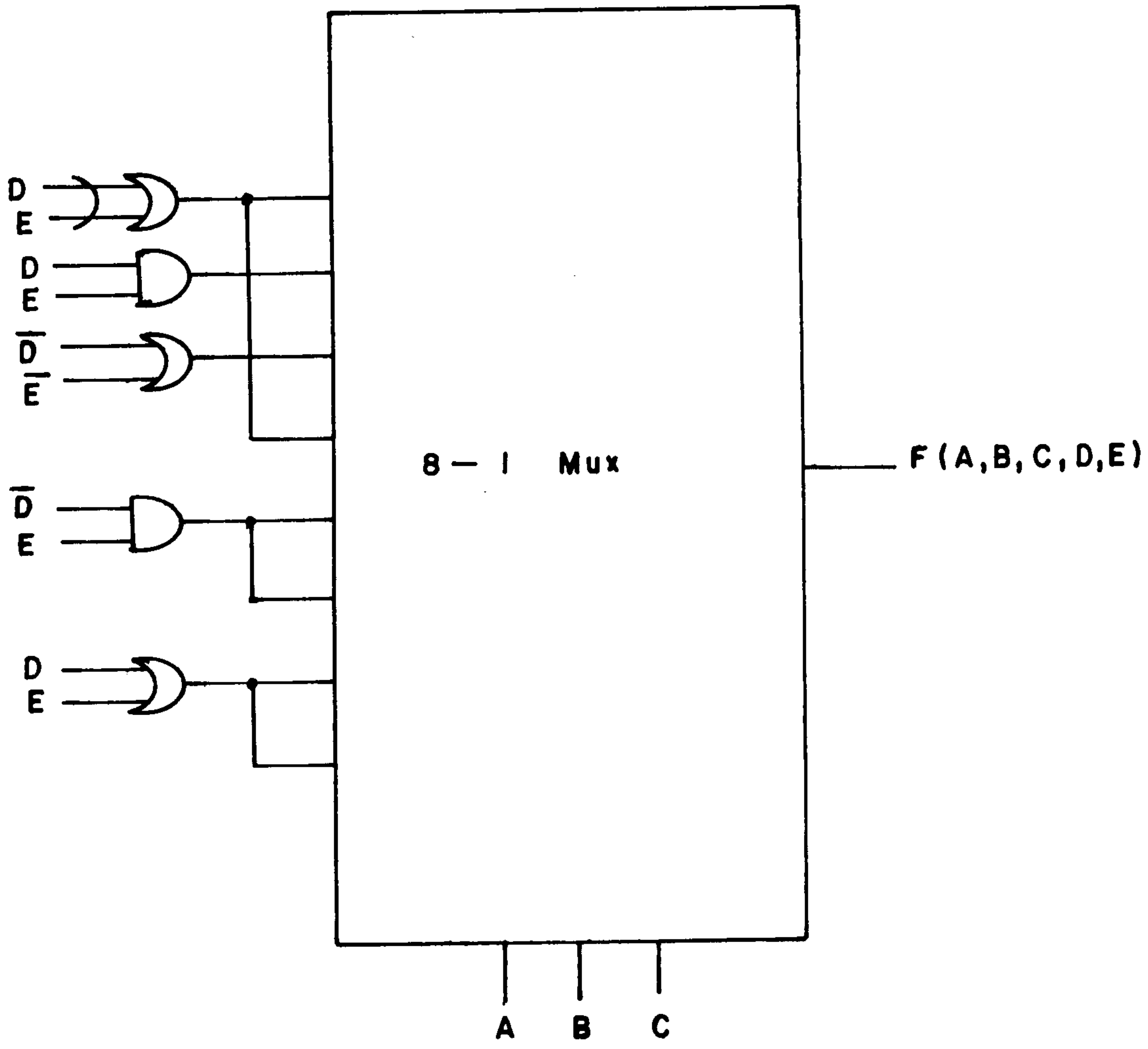


Fig. 8 : Multiplexer implementation for the ABC partition.

The objective of the design is to see if we can realise the function without additional gates. An examination of the ten Ashenhurst decomposition charts shows that the partition ACD as data select variables and BE as data input variables achieves the objective. The Ashenhurst decomposition chart for this partition is shown in Fig. 9(a) and the resulting Domindo and Canto structure is shown in Fig. 9(b). Compared to Fig. 8, the cost advantage of Fig. 9(b) is obvious.

4.4 THE EKTARE STRUCTURE

The Ektare algorithm seeks to realise a Boolean function with a minimum number of multiplexers for partitions where n > q + 1. If in table 3, we use the data input variables as data select and the data select variables as the data input variables, we obtain ten new partitions. However we do not need to draw additional Ashenhurst decomposition charts as the ten charts of section 4.3 can be used to obtain the desired residue functions by reading the chart horizontally rather than vertically. Examination of the Ashenhurst charts shows that the partition AD as the data select variables and BCE as the data input variables gives the cheapest multiplexer realisation of the function as it requires the minimum number of additional gates to implement the residue functions. Fig. l0(a) shows the Ashenhurst decomposition chart for this partition while Fig. l0(b) shows the resulting multiplexer implementation.

The residue functions of Fig. l0(b) can be implemented alternatively using multiplexers. This implementation was said in section 2 to be equivalent to a recursive application of Shannon's expansion theorem.
The residue function of Fig. 10(b) is

$$\mathrm{f}_{00} = \bar{\mathrm{C}}\mathrm{E} + \mathrm{BC} \qquad (8)$$

The cannonical form of this residue function is

$$f_{00} = \bar{B}\bar{C}E + B\bar{C}E + BCE + BC\bar{E} \quad (9)$$

If equation 9 is expanded about BC we obtain:

$$\mathrm{f}_{00} = \bar{\mathrm{B}}\bar{\mathrm{C}}\mathrm{E}) + \mathrm{B}\bar{\mathrm{C}}\mathrm{E}) + \mathrm{BC}(\bar{\mathrm{E}} + \mathrm{E})$$

(10)

Equation 10 is now in a form where it can be realised directly using a multiplexer as shown in Fig. 11. Fig. 11 shows the Ektare minimum multiplexer tree structure realisation.

4.5 MULTIPLEXER TREE STRUCTURE REALI SA TION

The Domindo and Canto structure gives the cheapest multiplexer realisation for n > q + 1. Compared to the Whitehead's structure (which applies for n = q + 1) it gives the maximum loading of the input but structure. It is of interest to find a partition for n > q + 1 which gives the same input but structure as the Whitehead's structure. Fig. 12(a) shows the Ashenhurst decomposition chart for this partition which uses AB as the data select variables and CDE as the data input variables while Fig. 12(b) shows the resulting multiplexer realisation. To obtain the tree structure realisation, we need to expand the residue functions in turn about DE. The result is as follows:

$$\mathrm{f}_{00} = \bar{\mathrm{D}}\mathrm{E}(\mathrm{C}) + \mathrm{D}\bar{\mathrm{E}}(\bar{\mathrm{C}} + \mathrm{C}) \qquad (11)$$

$$f_{01} = \bar{D}\bar{E}(C) + \bar{D}E(\bar{C} + C) + DE(\bar{C} + C) \quad (12)$$

$$f_{10} = \bar{D}\bar{E}(C) + \bar{D}E(\bar{C} + C) + D\bar{E}(0)DE(0) \quad (13)$$

$$f_{11} = \bar{D}E(\bar{C} + C) + DE(\bar{C} + C) + DE(\bar{C} + C)$$

(14)

Equations (11) to (14) can be implemented using multiplexers resulting in the tree structure realisation of Fig. 13.

Comparing the tree structure realisations of Figs. 11 and 13 we find that the realisation of Fig. 11 which uses a minimum number of multiplexers has an unsimplified input bus structure; whereas Fig. 13 which realises the function with a maximum number of multiplexers has a simplified input bus structure. There is, therefore, a trade off between the

	A	0				1			
	CD	00	01	11	10	00	01	11	10
BE	00	0	②	⑥	4	16	18	22	20
	01	①	3	7	5	⑰	19	23	㉑
	11	⑨	11	15	⑬	㉕	㉗	㉛	㉙
	10	8	⑩	⑭	⑫	24	㉖	㉚	28

Fig. 9 (a): Ashenhurst decomposition chart for the ACD data select partition.

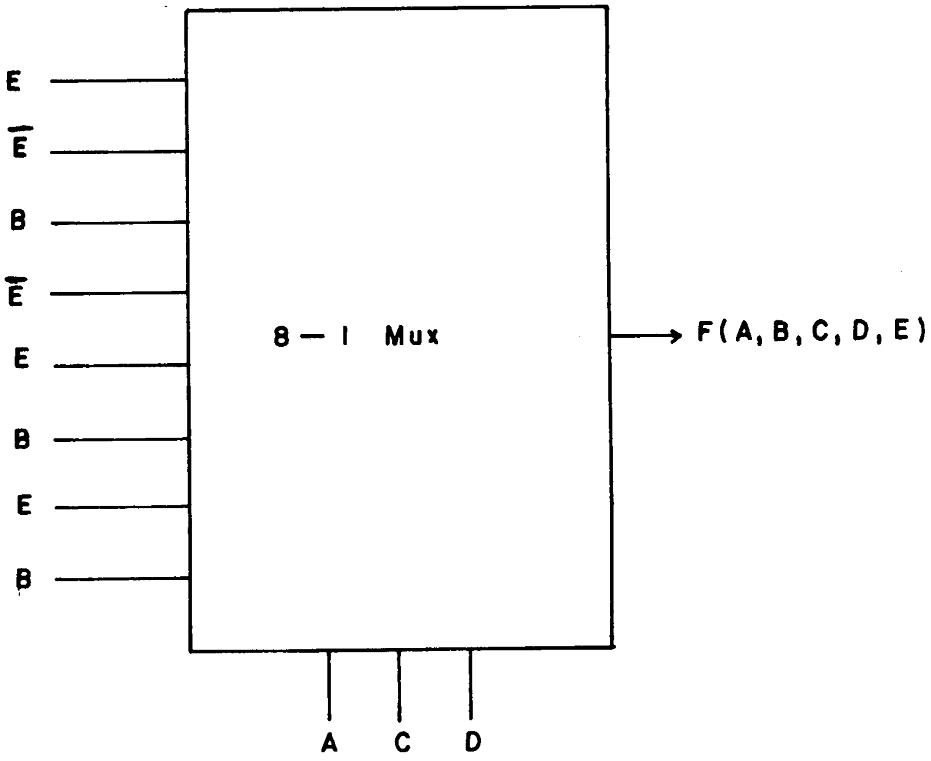


Fig. 9 (b): The Domindo and Canto Structure .

	B	0				1			
	CE	00	01	11	10	00	01	11	10
,D	00	0	1	5	4	8	9	13	12
	01	2	3	7	6	10	11	15	14
	11	18	19	23	22	26	27	31	30
	10	16	17	21	20	24	25	29	28

Fig. 10(a): Ashenhurst Decomposition Chart for the AD data select partition.

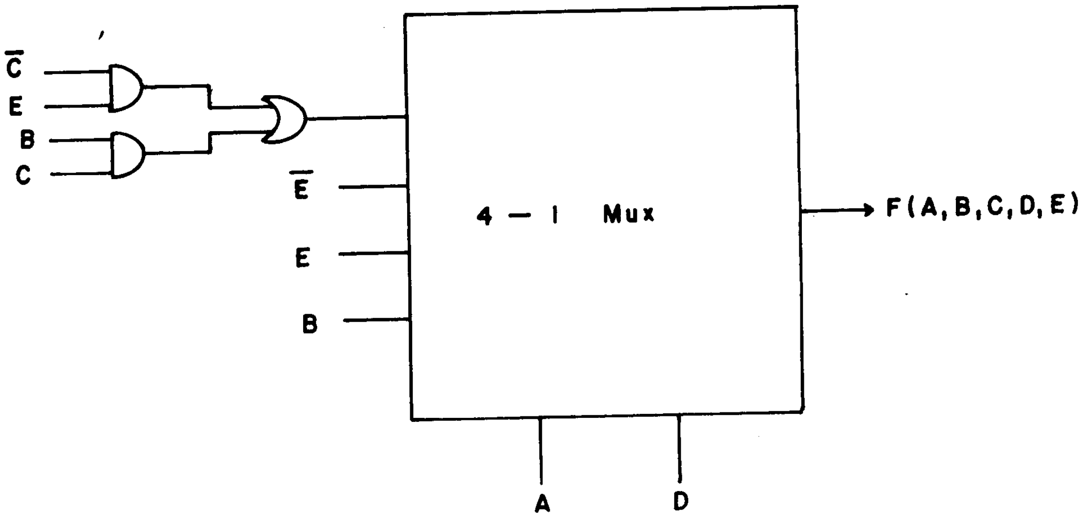


Fig. 10(b): Multiplexer implementation.

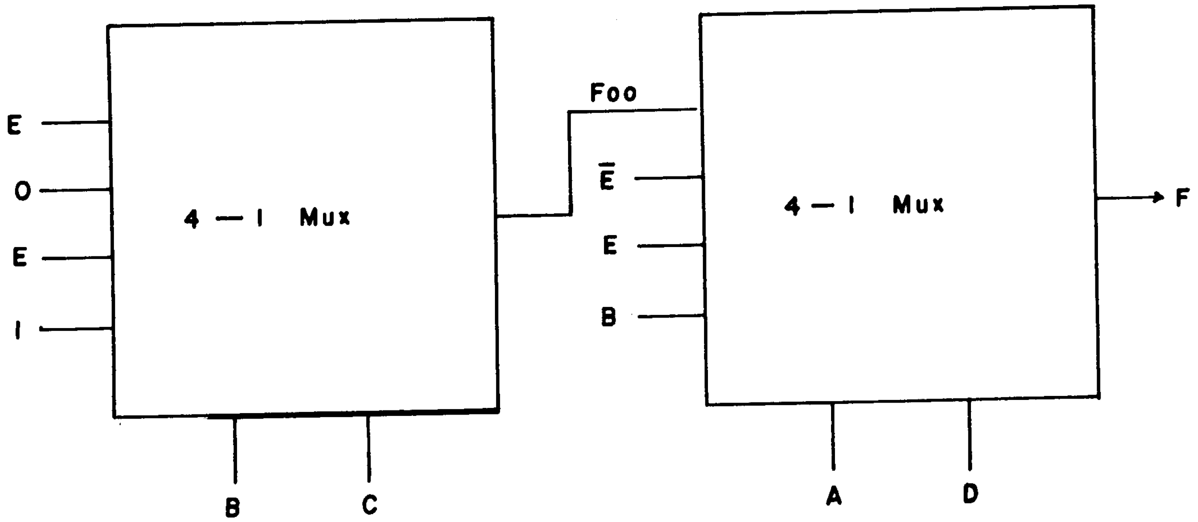


Fig. 11: Ektare's Structure.

	C	0				1			
	DE	00	01	11	10	00	10	11	10
AB	00	0	①	3	②	4	5	7	⑥
	01	8	⑨	11	⑩	⑫	⑬	15	⑭
	11	24	㉕	㉗	㉖	28	㉙	㉛	㉚
	10	16	⑰	19	18	20	㉑	23	22

Fig. 12(a): Ashenhurst Decomposition Chart for the AB data select partition.

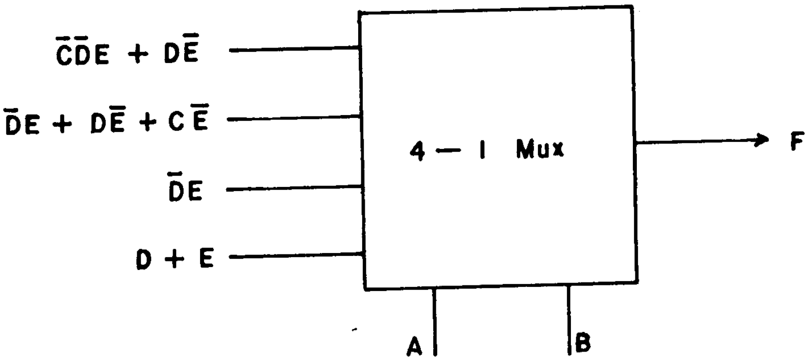

$\bar{C}\bar{D}E + D\bar{E}$
$\bar{D}E + D\bar{E} + C\bar{E}$
$\bar{D}E$
$D + E$
4 — 1 Mux
F
A
B

Fig. 12(b): Multiplexer Implementation.

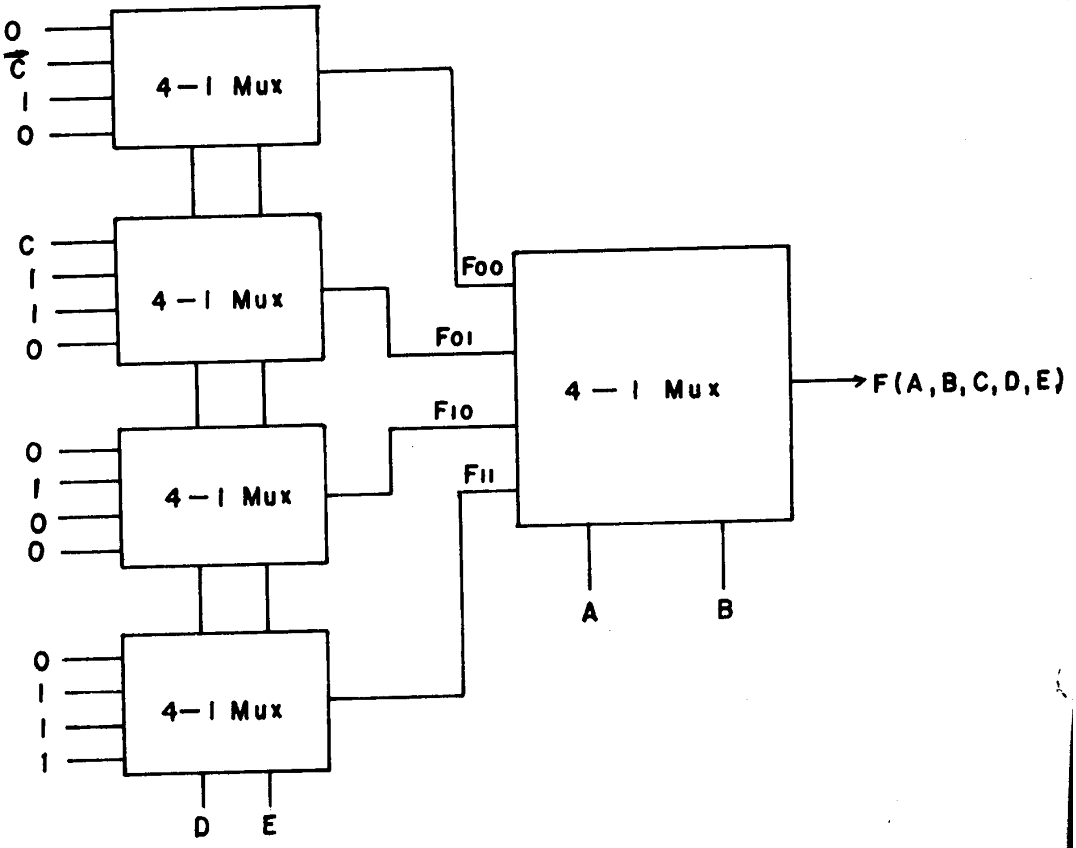


Fig. 13: A tree structure equivalent of the Whitehead's 16 — 1 Multiplexer realisation.

number of multiplexers needed to realise a given Boolean function and the complexity (with respect to loading) of the input bus structure.

5. CONCLUSION

The Ashenhurst decomposition chart is an easy to use mapping technique in the design of logic circuits with multiplexers. Using this technique, suitable partitions can be selected which meet different design objectives. Also the mapping technique is a good alternative to the algorithmic techniques thereby making the transition to computer based design much easier. The draw back of the Ashenhurst decomposition chart is the large number of charts to be examined. For functions of more than six variables, the number of charts to be examined in order to select the optimum partition becomes excessive.

Each algorithmic technique, on the other hand, realises a specific multiplexer structure without the need to examine all the possible Ashenhurst decomposition charts. Because it is based on the familiar Karnaugh map technique, the Ashenhurst decomposition chart offers a ready vehicle for teaching the synthesis of logic circuits with multiplexers.

REFERENCES

1. Whitehead, D.G., "Algorithm for logic circuit synthesis using multiplexers". *Electronic lett.,* Vol. 13, pp. 355-356, 1978.

2. Ektare, A.B., and Mital, D.P. "An Algorithm for designing multiplexer logic circuits", *Int.* J. *Electronics,* Vol. 49, pp. 103-114, 1980.

3. Domindo, B., and Canto, D. "Systematic Synthesis of Combinational Circuits using multiplexers". *Electronic Lett.,* Vol. 14, pp. 588-590, 1978.

4. Langdon, G.G., "A Decomposition Chart Technique to Aid in Realisations with multiplexers". *IEEE Transactions on Computers,* Vol. 27, pp. 157-159, 1978.

5. Lofti, Z.M., and Tosser, A.J., "Using Multiplexers to Realize Switching Functions Having particular properties "Int. J. Electronics, Vol. 47, pp. 253-266, 1979.

6. Lloyd, A.M. "Design of multiplexer ULM Networks using spectral Techniques." Proc. inst. Elect. Engrs., Nov. 127, pp. 31-36, 1980.

7. Hill, F.J. and Peterson, G.R., Introduction to Switching Theory and Logical Design. Wiley, New York, 1968.