Implementation of Boolean Functions through Multiplexers with the Help of Shannon Expansion Theorem

Saurabh Rawat
Graphic Era University.
Dehradun

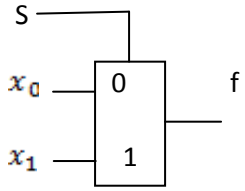
Anushree Sah
The University of Greenwich,
London, UK,

Sumit Pundir
Graphic Era University.
Dehradun

ABSTRACT
Implementation of Boolean function through multiplexer can be done by various multiplexers depending upon the select lines. Implementation of Boolean functions can be done by various methods, but in this particular paper stress is more on multiplexers. Through Shannon expansion theorem,it is easy for us to implement the Boolean functions in a simpler way. Upto three variables, can be handled by multiplexers, and above that we have taken aid of look out table, and how it uses multiplexers in their operations.

Keywords
Multiplexers, 2 x 1, 4 x 1, 8 x 1, multiplexers, Shannon Theorem..

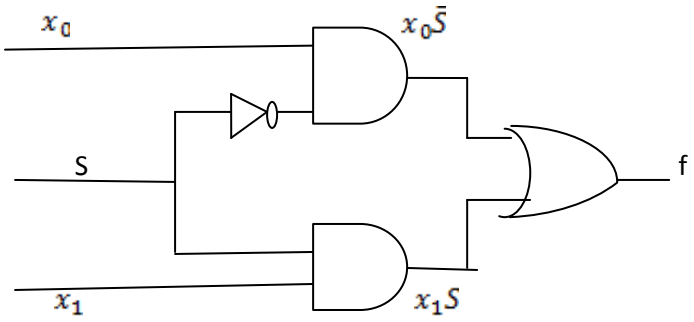
1. Multiplexer
A multiplexer circuit has a number of data inputs, one or more select inputs and one output . It passes the signal value on one of the data inputs to the output. Data input is selected by the values of the select inputs .Select input S, chooses as the output of the multiplexer either input x_0 or x_1. Multiplexer's functionality can be described in the form of a truth table.



1.1 Graphic symbol

s	f
0	x_0
1	x_1

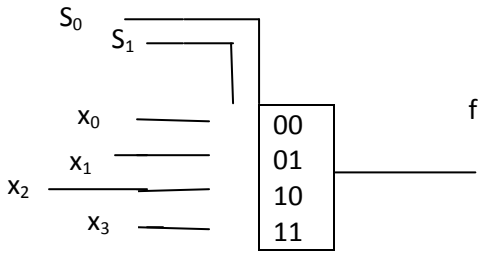
1.2 truth table



1.3 Sum of products circuit

A 2 to 1 Multiplexer (f= $x_0\bar{S} + x_1S$)

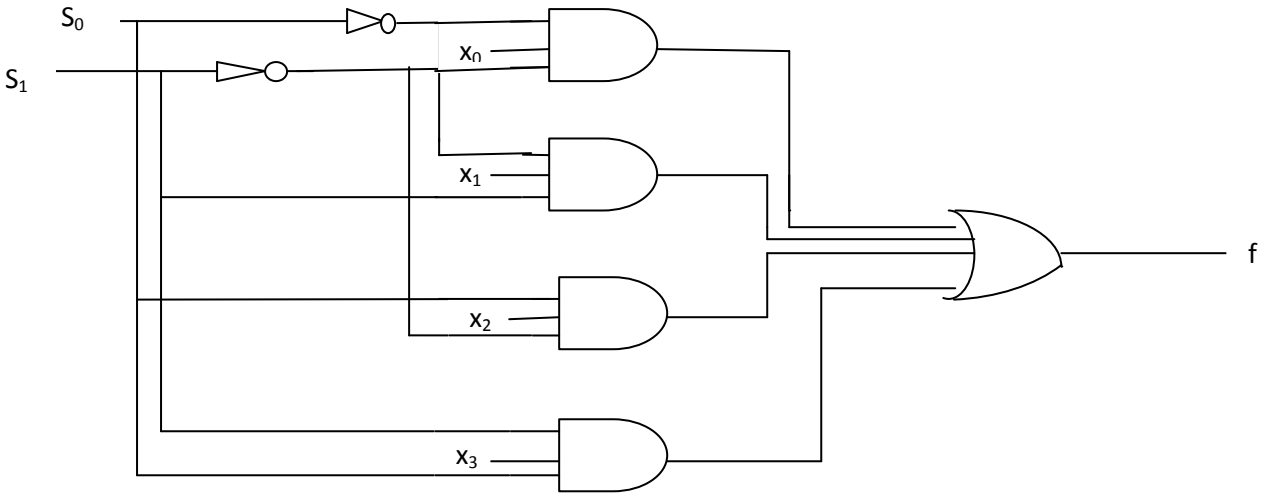
and 4 to 1 multiplexer have four data inputs x_0,x_1,x_2, & x_3 and two select inputs S1 and S0 . The two bit number represented by S1S0 select one of the data input as output of the multiplexer.



1.4 Graphic symbol

S_1	S_2	f
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

1.5 Truth table

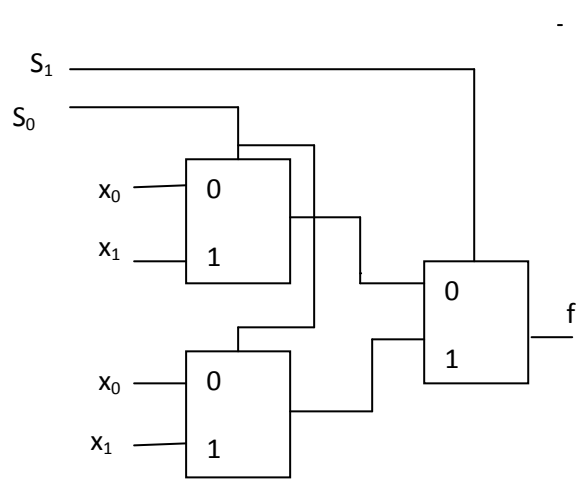


1.6 Sum of product circuit

A 4 to 1 multiplexer

$$f = \overline{S_1}\,\overline{S_0}x_0 + \overline{S_1}S_0x_1 + S_1\overline{S_0}x_2 + S_1S_0x_3$$

A multiplexer that has n data inputs, requires $\log_2 n$ select inputs. Larger multiplexer can be constructed from small multiplexers.



1.7 Using 2 to 1 multiplexer to build a 4 to 1 multiplexer

2. Boolean function implementation of multiplexer

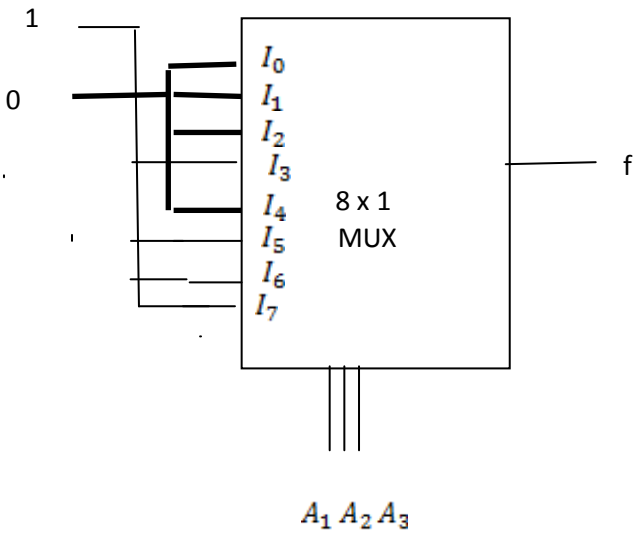
$n=2^m$

n – number of input variables

m- number of select inputs

2.1 I- **Using 8x1 multiplexer for implementation**

$f(A_1, A_2, A_3) = \Sigma\,(3,5,6,7)$



2.1.1 8x1 multiplexer implementation

2.2 II- **Using 4x1 Multiplexer for implementation**

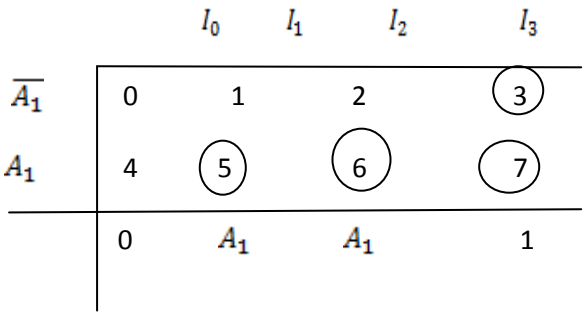
Connecting two variables with selection lines of multiplexer and remaining single variable of the function is used for the inputs of the multiplexer. If A is a single variable the inputs of multiplexer are chosen to be either A or $\overline{A}$ or 1 to 0

(a) $f(A_1, A_2, A_3) = \Sigma\,(3,5,6,7)$

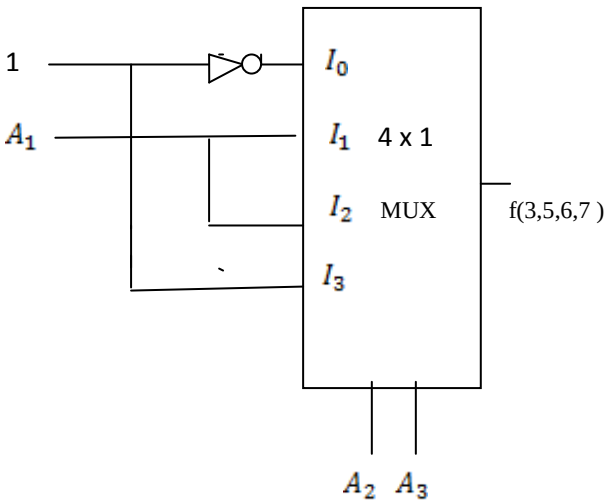
if MSB i.e. A_1 is used as single variable, and A_2, A_3 as select inputs.

Minterms	A_1	A_2	A_3	**f**
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

2.2.1 Truth table



2.2.2 Multiplexer Implementation



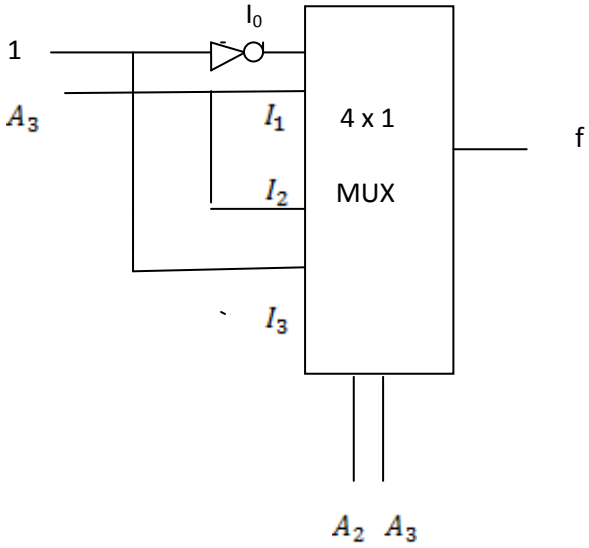
2.2.3 4x1 Multiplexer implementation

(b) $f(A_1, A_2, A_3) = \Sigma (3,5,6,7)$

If LSB i.e. A_3 is used as single variable and A_1, A_2 as select inputs

Minterm	A_1	A_2	A_3	f	
0	0	0	0	0	
1	0	0	1	0 f= 0	I_0
2	0	1	0	0	
3	0	1	1	1 f= A_3	I_1
4	1	0	0	0	
5	1	0	1	1 f= A_3	I_2
6	1	1	0	1	
7	1	1	1	1 f= 1	I_3

2.2.4 Truth table



2.2.5 4x1 Multiplexer Implementation

Besides using such inputs, it is possible to connect more complex circuit as inputs to a multiplexer allowing function to be synthesized using a combination of multiplexers & other logic gates.

3. Shannon's Expansion theorem

Shannon's expansion or the Shannon decomposition is a method by which a Boolean function can be represented by the sum of two sub function of the original. Shannon expansion develops the idea that boolean function can be reduced by means of the identity.

$$f = xf_x + \bar{x} f_{\bar{x}}$$

where f is any function and f_x and $f_{\bar{x}}$ are positive & negative shannon cofactors of f respectively . A positive shannon cofactor of function f with respect to variable x is defend as that function with all instances of x replaced by 1 A negative shannon co factor is the same, but replaces all instances of x by 0.

Any Boolean function f (A_1, A_2, A_3, ---An) can be written in the form

f (A_1, A_2 ---An) =$\bar{A}_1$ .f (0, A_2 , A_3---An) + A_1. f (1, A_2 , A_3---An)

This expansion can be done in terms of any of the n variables.

as f (A_1, A_2, A_3)= Σ (3,5,6,7)

Function can be expressed as sum of products form

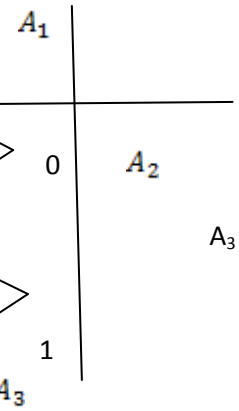
f = $\bar{A}_1A_2A_3$ + A_1 $\bar{A}_2A_3$ + A_1 A_2 $\bar{A}_3$ + A_1A_2 A_3

It can be manipulated into

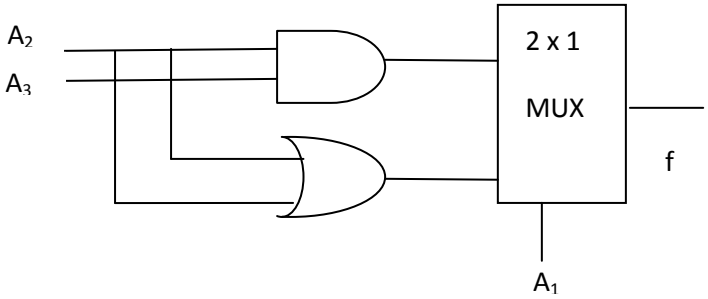
f = $\bar{A}_1(A_2A_3)$ + $A_1(\bar{A}_2A_3$ +A_2 $\bar{A}_3$ + A_2 A_3)

= $\bar{A}_1$ (A_2A_3) + A_1 (A_2+A_3)

A_1	A_2	A_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



3.1 Truth table



3.2 Truth table Circuit

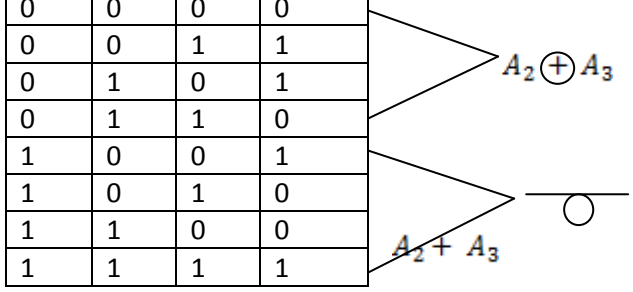
Three input majority function implemented using a 2 to 1 multiplexer

For three input XOR function

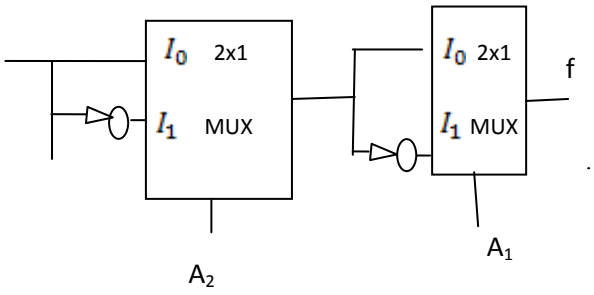
$$f = A_1 \oplus A_2 \oplus A_3$$

$$= \bar{A}\ (A_2 \oplus A_3) + A_1(\overline{A_2 \oplus A_3})$$

A_1	A_2	A_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



3.3 Truth table



3.4 Three input XOR implemented with 2 to 1 Multiplexer

In Shannon's expansion the term f (0, A_2,----An) is called the co-factor of f with respect to $\bar{A}_1$, denoted as $f_{\bar{A}_1}$. Similarly the term f (1, A_2 ----------An) is called the co-factor of f with respect to A_1, written as f_{A_1} , hence

$$f = \overline{A_1}\, f_{\overline{A_1}} + A_1\, f_{A_1}$$

In general if the expansion is done with respect to variable A_i , then f_{A_i} denotes

$f(A_1,-----,A_{i-1},1,A_{i+1},-----,A_n)$ and

$$f(A_1,----,A_n) = \overline{A_i} f_{\overline{A_i}} + A_i f_{A_i}$$

Complexity of the logic expression may vary, depending on which variable A_i , is used.

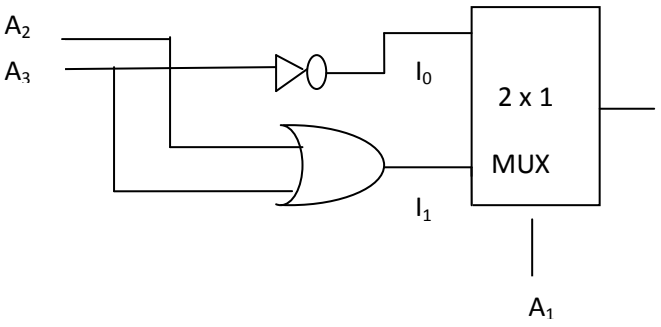
Taking another example, to implement the function

f = $\bar{A}_1\bar{A}_3 + A_1A_2 + A_1A_3$

a) Using 2 to 1 multiplexer, Shannon's expansion using A_1 gives

$$f = \overline{A_1} f_{\overline{A_1}} + A_1 f_{A_1}$$

= $\bar{A}1\ (\bar{A}_3) + A_1\ (A_2 + A_3)$

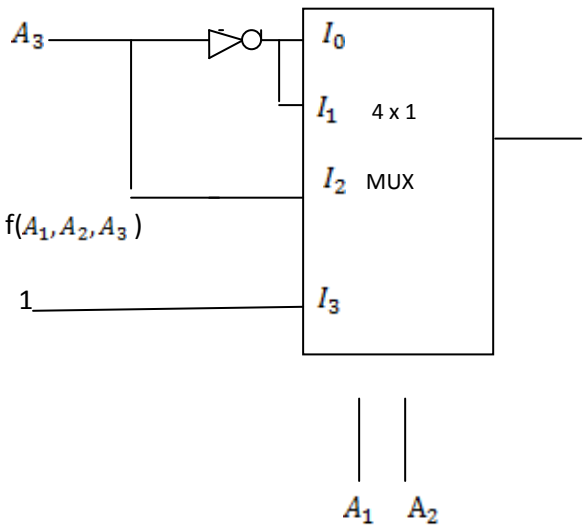


3.5 Using 2 to 1 multiplexer

b) Using 4 to 1 multiplexer, further .....using A_2 gives

f= $\bar{A}_1\bar{A}_2\ f_{\bar{A}1\bar{A}2} + \bar{A}_1A_2\ f_{\bar{A}1A2} + A_1\bar{A}_2\ f_{A1\ \bar{A}2} + A_1A_2\ f_{A1\ A2}$

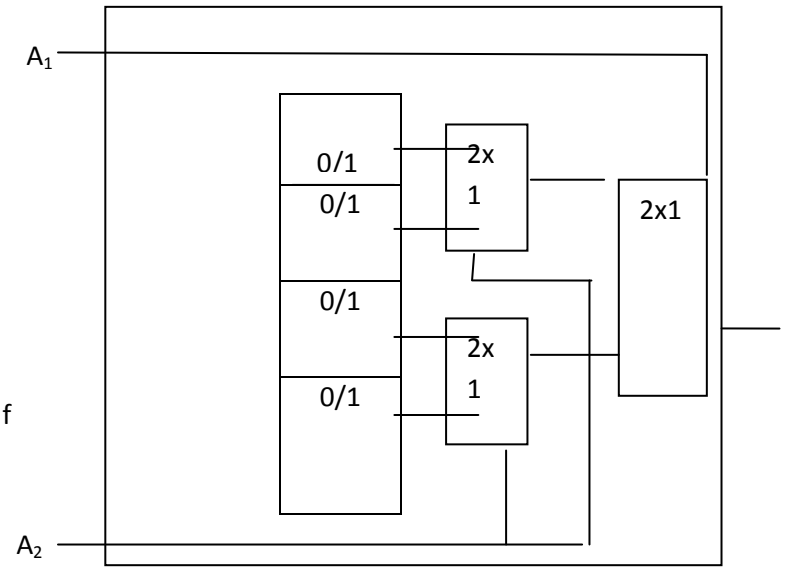
= $\bar{A}_1\bar{A}_2\ (\bar{A}_3) + \bar{A}_1A_2\ (\bar{A}_3) + A_1\bar{A}_2(\bar{A}_3) + A_1A_2(1)$



3.6 Using 4 to 1 multiplexer

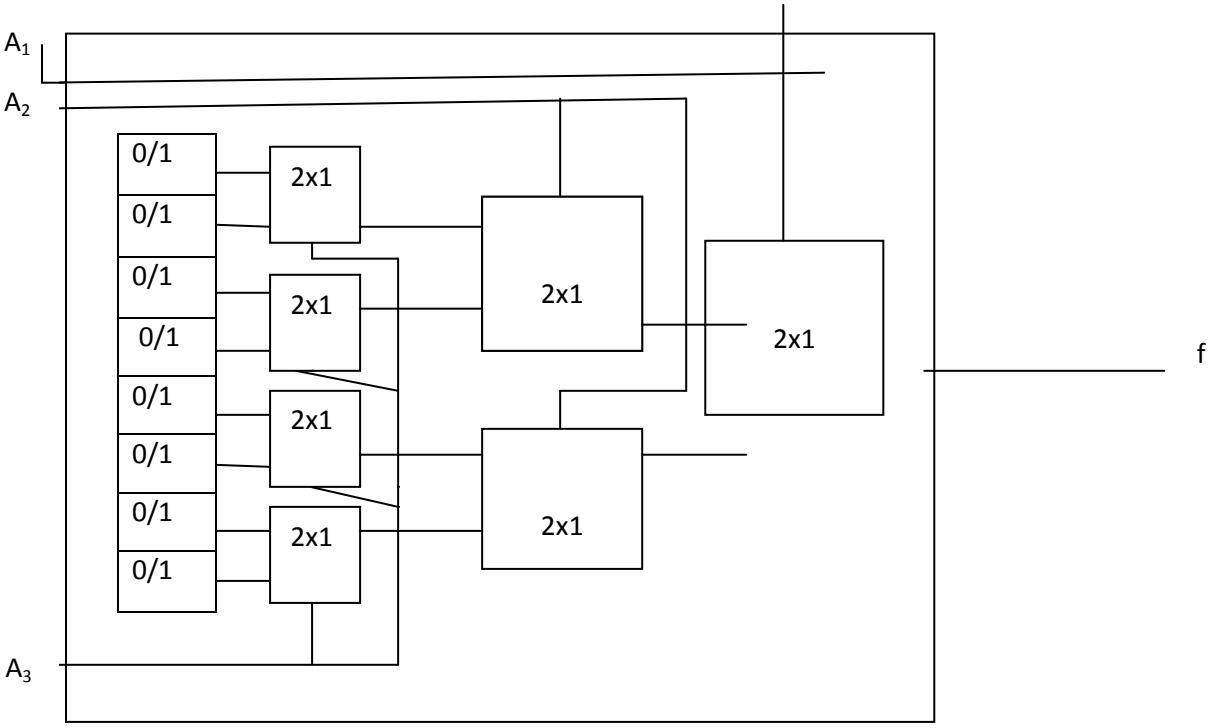
The most commonly used logic block is a lookout table (LUT), which contains storage cells that are used to implement a small logic function. Each cell is capable of holding a single logic value either 0 or 1. LUTs of various sizes may be created, where the size is defined by number of inputs.

inputs .



3.7 Circuit for a two input LUT

)



3.8 A three input LUT

Using shannon's expansion any four variable function can be realized with at most three 3- LUTs (look up tables).
Considering the function

$f = \bar{A}_2A_3+\bar{A}_1A_2\bar{A}_3+A_2\bar{A}_3A_4+A_1\bar{A}_2A_4$

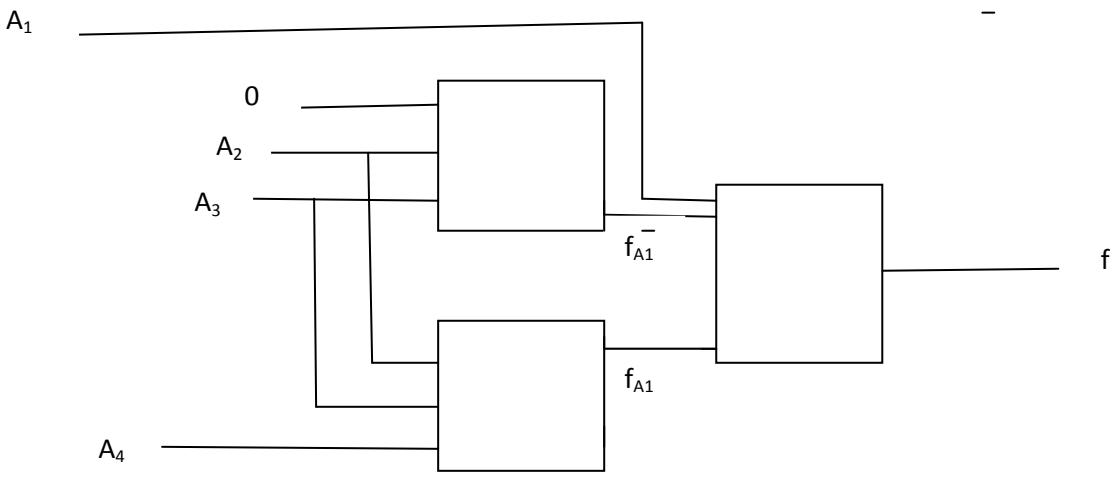
Expansion in terms of A_1 produces

$f = \bar{A}_1 f_{\bar{A}1} + A_1 f_{A1}$

$= \bar{A}_1 (\bar{A}_2A_3+A_2\bar{A}_3+A_2\bar{A}_3A_4)+A_1 (\bar{A}_2A_3+A_2\bar{A}_3A_4+\bar{A}_2\bar{A}_4)$

$= \bar{A}_1(\bar{A}_2A_3+A_2\bar{A}_3)+A_1(\bar{A}_2A_3+A_2\bar{A}_3A_4+\bar{A}_2\bar{A}_4)$

A circuit with there 3 LUTs that implements this expression



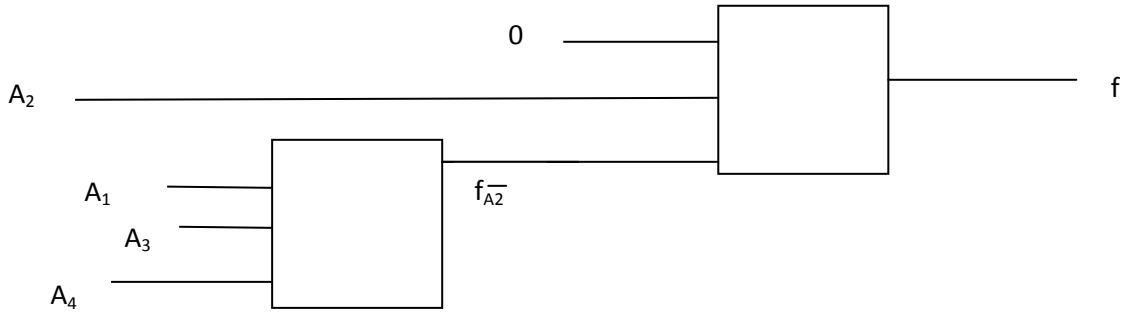
3.9 Using there 3 LUTs

Decomposition of the function using A_2, instead of A_1, gives

$f = \bar{A}_2 f_{A2} + A_2 f_{A2}$

$= \bar{A}_2(A_3 + A_1\bar{A}_4) + A_2(\bar{A}_1\bar{A}_3 + \bar{A}_3 A_4)$

as $f_{\bar{A}2} = f_{A2}$, hence only two 3 LUTs are needed .



3.10 Using two 3 LUTs**.**

4. CONCLUSION

In this paper we have seen that Boolean functions can be implemented using different multiplexers, 2x1, 4x1 or 8x1 . With the help of Shannon expansion theorem ,complicated Boolean functions can be made easy ,in implementing through multiplexers and LUTs (look up table),again formed with different combination of multiplexers. This study will be very helpful for researchers and intellectuals to easy understanding and practicing of implementation of Boolean functions through multiplexers in the field of computer science and technology.

5. REFERENCES

[1] en.wikipedia.org/wiki/Multiplexer

[2] en.wikipedia.org/wiki/Shannon's_expansion

[3] Arturo Hern´andez Aguirre, Bill P. Buckles, and Carlos Coello Coello. Evolutionary synthesis of logic functions using multiplexers. In C. Dagli, A.L. Buczak, and et al., editors, Proceedings of the 10th Conference Smart Engineering System Design, pages 311–315, New York, 2000. ASME Press.

[4] R. L. Ashenhurst, "The decomposition of switching functions," in Proc.Int. Symp. Theory of Switching Functions, Apr. 1957, pp. 74–116.

[5] Astola, J.T., Stankovi´c, R.S., Fundamentals of Switching Theory and Logic Design, Springer, 2006.

[6] M. MORRIS MANO "Digital Logic and Computer Design" 2nd edition

[7] D. Nasib S. Gill, J.B. Dixit "Digital Design and Computer Organisation"