NOR and NAND operators in Boolean algebra applied to switching circuit design

D. Zissos and F. G. Duncan*

Department of Mathematics, Statistics and Computing Science, University of Calgary, Calgary 44, Alberta, Canada

Operators in Boolean algebra for NOR and NAND are introduced and applied to problems in switching circuit design. These operators have properties which very closely parallel the properties of NOR and NAND gates. Their relationship to the problems of fan-in, fan-out and signal-delay are briefly discussed.

(Received November 1970)

In nearly all writings on logic design, the Boolean algebra is worked out in terms of the elementary operators AND, OR, and NOT. However, modern techniques call for the design of NOR and NAND circuits. It seems reasonable, therefore, to introduce operators for NOR and NAND, and to develop theorems and other results for use in the design of NOR/NAND circuits.

AND and OR are usually regarded as dyadic (binary) operators. Since they are associative this restriction is no inconvenience. However, dyadic NOR and NAND are not associative.

[If $\mathrm{NOR}(A, B) \equiv \bar{A}.\bar{B}$
then $\mathrm{NOR}(\mathrm{NOR}(A, B), C) \equiv \mathrm{NOR}(\bar{A}.\bar{B}, C) \equiv (A + B).\bar{C}$
while $\mathrm{NOR}(A, \mathrm{NOR}(B, C)) \equiv \mathrm{NOR}(A, \bar{B}\bar{C}) \equiv \bar{A}(B + C)$]

Therefore NOR and NAND will not be defined as dyadic operators. Instead, NOR and NAND will be defined for any number of operands, and relationships between them for different numbers will be worked at.

This treatment follows very closely the properties of electronic NOR and NAND gates with different fan-in restrictions. The question of fan-in will be referred to later.

It is well known that there is a duality between NOR and NAND; therefore the worked examples will be given in terms of NOR gates only.

The development of results with the proposed notation illustrates the disadvantages of the Sheffer stroke and Pierce arrow notation for this purpose. These latter are conceived as dyadic infix operators; they are extended for more than two operands as are arithmetic operators, while for single operands a very artificial convention is adopted ("$a|a$"). These extensions mean that there is not in general a one-to-one correspondence between operations and operators. In the proposed notation, however, there is a very close, one-to-one correspondence between operations and operators, which in terms of switching circuits means that there is a one-to-one correspondence between gates and operators and between input signals and operands. This point is illustrated in the circuit diagrams which accompany the following text.

In the worked examples, the procedure adopted is first to replace the conventional operators by NAND or NOR operators, and then to simplify the NAND or NOR expression as far as possible. The replacement of conventional operators follows the usual rules for evaluating expressions: inner brackets first, NOT before AND, and AND before OR. The simplification is rather unsystematic; a methodical approach to this problem is under consideration, although naturally it will depend to a large extent on practice obtained with the new notation.

The last example, which is the problem posed by Roth, Karp, McFarlin and Wilts (1961) and solved on a computer by Barnard and Holman (1968), has been worked rather differently. The expression has been minimised with the new operators first; the steps are justified in detail, in terms of the conventional operators, in another paper (Duncan and Zissos, 1970) by the present authors.

Definitions

∇ stands for the NOR operator.

1. $\nabla(A) \equiv \nabla A \equiv \bar{A}$
2. $\nabla(A, B) \equiv \bar{A}.\bar{B}$
3. $\nabla(A, B, C) \equiv \bar{A}.\bar{B}.\bar{C}$
4. $\nabla(A, B, C, \ldots, X) \equiv \bar{A}.\bar{B}.\bar{C} \ldots \bar{X}$

Theorems, etc.

1. Since Boolean multiplication (AND) is commutative, the order of terms in the argument list of ∇ is irrelevant. In particular, $\nabla(A, B, C) \equiv \nabla(B, A, C)$
2. $\nabla(A, A) \equiv \nabla A$
   $\nabla(A, A, B) \equiv \nabla(A, B)$
   In general, a repeated term is equivalent to the single term.
3. $\nabla(0) \equiv 1 \equiv \nabla 0$
   $\nabla(A, 0) \equiv \nabla A$
   In general, a zero term can be omitted.
4. $\nabla(1) \equiv 0 \equiv \nabla 1$
   $\nabla(A, 1) \equiv 0.$
   If any term is 1, the whole is 0
5. $\nabla A.\nabla B \equiv \bar{A}.\bar{B} \equiv \nabla(A, B)$
   $\nabla A.\nabla B.\nabla C \equiv \bar{A}.\bar{B}.\bar{C} \equiv \nabla(A, B, C)$ etc.
6. $A + B \equiv \overline{\bar{A}.\bar{B}} \equiv \nabla(\bar{A}.\bar{B}) \equiv \nabla\nabla(A, B).$
7. $A.B \equiv \nabla(\bar{A}, \bar{B}) \equiv \nabla(\nabla A, \nabla B)$
8. $A + B + C + \ldots + X \equiv \nabla\nabla(A, B, C, \ldots, X)$
9. $A.B.C \ldots X \equiv \nabla(\nabla A, \nabla B, \nabla C, \ldots, \nabla X)$
10. $\nabla\nabla A \equiv \nabla\bar{A} \equiv A.$
11. $\nabla\nabla\nabla(A, B) \equiv \nabla\nabla(\bar{A}\bar{B}) \equiv \nabla(A + B) \equiv \overline{A + B} \equiv \bar{A}.\bar{B} \equiv \nabla(A, B)$
12. $\nabla\nabla\nabla(A, B, \ldots, X) \equiv \nabla(A, B, \ldots, X)$ similarly.
13. Analogue to the distributive law:
    $\nabla\nabla(A, \nabla(B, C)) \equiv \nabla(\nabla(A, \nabla B), \nabla(A, \nabla C))$
    Proof: $\mathrm{RHS} \equiv \nabla(\bar{A}B, \bar{A}C)$
    $\equiv \nabla\nabla\nabla(\bar{A}B, \bar{A}C)$
    $\equiv \nabla(\bar{A}B + \bar{A}C)$
    $\equiv \nabla(A.(B + C))$
    $\equiv \nabla\nabla(A, \nabla(B + C))$
    $\equiv \nabla\nabla(A, \nabla\nabla\nabla(B, C))$
    $\equiv \nabla\nabla(A, \nabla(B, C)) \equiv \mathrm{LHS}.$
14. $\nabla(A, B) \equiv \nabla(A + B)$
15. $\nabla(A, B, \ldots, X) \equiv \nabla(A + B + \ldots + X)$
16. $\nabla(A, \nabla A) \equiv 0$

*Present address: Department of Mathematics, University of Bristol.

Definitions

$\triangle$ stands for the NAND operator.

1. $\triangle(A) \equiv \triangle A \equiv \bar{A}$.
2. $\triangle(A, B) \equiv \bar{A} + \bar{B}$
3. $\triangle(A, B, C) \equiv \bar{A} + \bar{B} + \bar{C}$
4. $\triangle(A, B, C, \ldots, X) \equiv \bar{A} + \bar{B} + \bar{C} + \ldots + \bar{X}$

Theorems, etc.

1. Since Boolean addition (OR) is commutative, the order of terms in the argument list of $\triangle$ is irrelevant. In particular, $\triangle(A, B, C) \equiv \triangle(B, A, C)$.
2. $\triangle(A, A) \equiv \triangle A$
   $\triangle(A, A, B) \equiv \triangle(A, B)$
   In general, a repeated term is equivalent to the single term.
3. $\triangle(1) \equiv 0 \equiv \triangle 1$
   $\triangle(A, 1) \equiv \triangle A$
   In general, if a term is 1 it can be omitted.
4. $\triangle(0) \equiv 1 \equiv \triangle 0$
   $\triangle(A, 0) \equiv 1$
   If any term is 0, the whole is 1.
5. $\triangle A + \triangle B \equiv \bar{A} + \bar{B} \equiv \triangle(A, B)$.
   $\triangle A + \triangle B + \triangle C \equiv \bar{A} + \bar{B} + \bar{C} \equiv \triangle(A, B, C)$ etc.
6. $A.B \equiv \overline{\bar{A} + \bar{B}} \equiv \triangle(\bar{A} + \bar{B}) \equiv \triangle\triangle(A, B)$
7. $A + B \equiv \triangle(\bar{A}, \bar{B}) \equiv \triangle(\triangle A, \triangle B)$
8. $A.B.C \ldots X \equiv \triangle\triangle(A, B, C, \ldots, X)$.
9. $A + B + C + \ldots + X \equiv \triangle(\triangle A, \triangle B, \triangle C, \ldots, \triangle X)$
10. $\triangle\triangle A \equiv \triangle\bar{A} \equiv A$
11. $\triangle\triangle\triangle(A, B) \equiv \triangle\triangle(\bar{A} + \bar{B}) \equiv \triangle(AB) \equiv \bar{A} + \bar{B}$
    $\equiv (A, B)$
12. $\triangle\triangle\triangle(A, B, \ldots, X) \equiv \triangle(A, B, \ldots, X)$, similarly.
13. Analogue to distributive law:

$$\triangle\triangle(A, \triangle(B, C)) \equiv \triangle(\triangle(A, \triangle B), \triangle(A, \triangle C))$$

Proof: RHS $\equiv \triangle(\bar{A} + B, \bar{A} + C)$
$\equiv \triangle\triangle\triangle(\bar{A} + B, \bar{A} + C)$
$\equiv \triangle((\bar{A} + B)(\bar{A} + C))$
$\equiv \triangle(\bar{A} + BC)$
$\equiv \triangle\triangle(A, \triangle(BC))$
$\equiv \triangle\triangle(A, \triangle\triangle\triangle(B, C))$
$\equiv \triangle\triangle(A, \triangle(B, C)) \equiv$ LHS

14. $\triangle(A, B) \equiv \triangle(A.B)$
15. $\triangle(A, B, \ldots, X) \equiv \triangle(A.B. \ldots . X)$
16. $\triangle(A, \triangle A) \equiv 1$

Examples

1. $P = A + BC$
   $= A + \nabla(\nabla B, \nabla C)$ (T.7)
   $= \nabla\nabla(A, \nabla(\nabla B, \nabla C))$ (T.6) – (i)
   $= \nabla(\nabla(A, B), \nabla(A, C))$ (T.13) –(ii)

Here (i) corresponds to the five-gate circuit of **figure 1** while (ii) corresponds to the three-gate circuit of **figure 2**

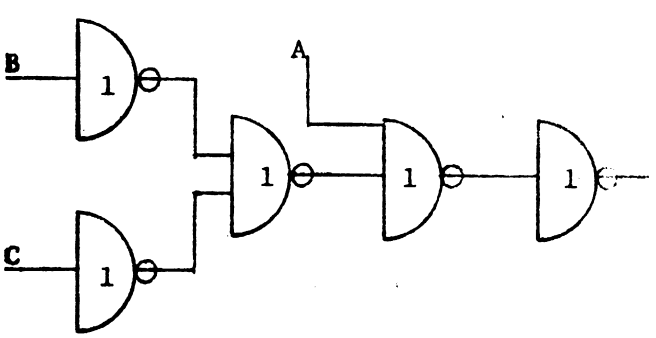

Fig. 1

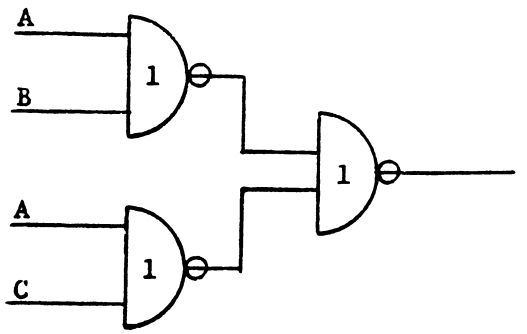

Fig. 2

2. $P = (A + B)(A + C)$
   $= \nabla\nabla(A, B).\nabla\nabla(A, C)$ (T.6)
   $= \nabla(\nabla\nabla\nabla(A, B), \nabla\nabla\nabla(A, C))$ (T.7)
   $= \nabla(\nabla(A, B), \nabla(A, C))$ (T.11)

as in Example 1.

3. $P = A + \bar{B}\bar{C}$
   $= A + \nabla(B, C)$
   $= \nabla\nabla(A, \nabla(B, C))$ **(Figure 4)**
4. $P = (A + \bar{B})(A + \bar{C})$
   $= \nabla\nabla(A, \nabla B).\nabla\nabla(A, \nabla C)$
   $= \nabla(\nabla\nabla\nabla(A, \nabla B), \nabla\nabla\nabla(A, \nabla C))$
   $= \nabla(\nabla(A, \nabla B), \nabla(A, \nabla C))$ – (i)
   $= \nabla\nabla(A, \nabla(B, C))$ (T.13) –(ii)

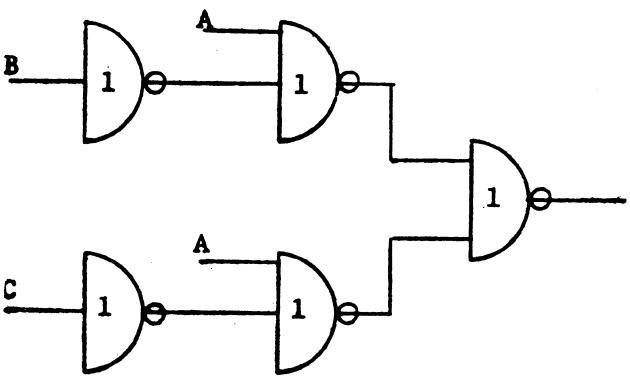

Fig. 3

Here (i) corresponds to the five-gate circuit of **figure 3** while (ii) (and the result of example 3) corresponds to the three-gate circuit of figure 4.

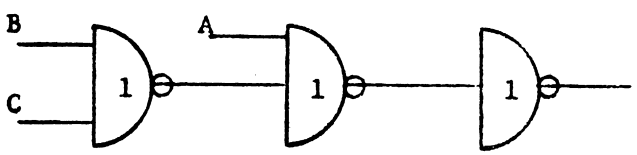

Fig. 4

5. $P = \bar{A}(B + C)$
   $= \nabla(A, \nabla(B + C))$
   $= \nabla(A, \nabla\nabla\nabla(B, C))$
   $= \nabla(A, \nabla(B, C))$ **(Figure 5)**

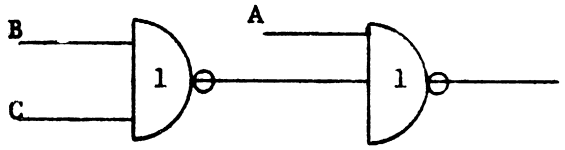

Fig. 5

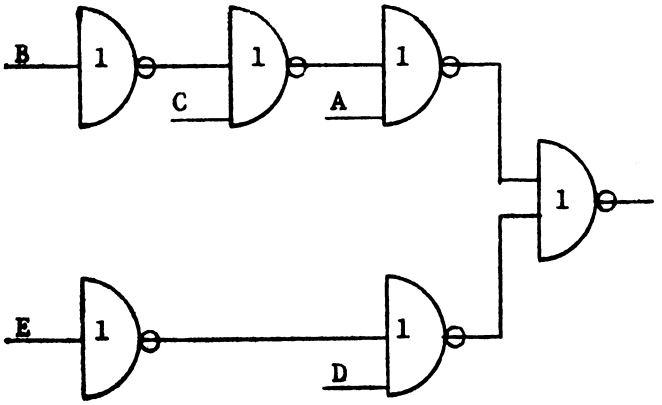

Fig. 6

6. $P = (A + B\bar{C})(D + \bar{E})$
$= (A + \nabla(\nabla B, C)).(D + \bar{E})$
$= \nabla\nabla(A, \nabla(\nabla B, C)).\nabla\nabla(D, \nabla E)$
$= \nabla(\nabla\nabla\nabla(A, \nabla(\nabla B, C)), \nabla\nabla\nabla(D, \nabla E))$
$= \nabla(\nabla(A, \nabla(\nabla B, C)), \nabla(D, \nabla E))$ **(Figure 6)**

7. $P = (\bar{A}\bar{B} + \bar{C}\bar{D}).(\bar{E} + F)$
$= (\nabla(A, B) + \nabla(C, D)).(\nabla E + F)$
$= \nabla\nabla(\nabla(A, B), \nabla(C, D)).\nabla\nabla(\nabla E, F)$
$= \nabla(\nabla\nabla\nabla(\nabla(A, B), \nabla(C, D)), \nabla\nabla\nabla(\nabla E, F))$
$= \nabla(\nabla(\nabla(A, B), \nabla(C, D)), \nabla(\nabla E, F))$ **(Figure 7)**

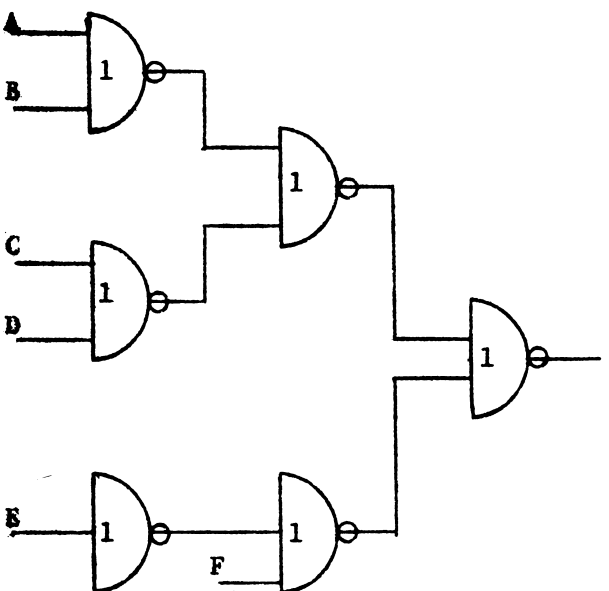

Fig. 7

8. $P = \bar{A}\bar{B}\bar{E} + \bar{E}\bar{C}\bar{D} + \bar{A}\bar{B}F + \bar{C}\bar{D}F$
$= \nabla(A, B, E) + \nabla(E, C, D) + \nabla(A, B, \nabla F) + \nabla(C, D, \nabla F)$
$= \nabla(E, \nabla\nabla(A, B)) + \nabla(E, \nabla\nabla(C, D)) + \nabla(\nabla F, \nabla\nabla(A, B)) + \nabla(\nabla F, \nabla\nabla(C, D))$
$= \nabla\nabla(\nabla(E, \nabla\nabla(A, B)), \nabla(\nabla F, \nabla\nabla(A, B))) + \nabla\nabla(\nabla(E, \nabla\nabla(C, D)), \nabla(\nabla F, \nabla\nabla(C, D)))$
$= \nabla\nabla\nabla(\nabla\nabla(A, B)\nabla(\nabla E, F)) + \nabla\nabla\nabla(\nabla\nabla(C, D), \nabla(\nabla E, F))$
$= \nabla\nabla(\nabla(\nabla\nabla(A, B), \nabla(\nabla E, F)), \nabla(\nabla\nabla(C, D), \nabla(\nabla E, F)))$
$= \nabla\nabla\nabla(\nabla(\nabla E, F), \nabla(\nabla(A, B), \nabla(C, D)))$
$= \nabla(\nabla E, F), \nabla(\nabla(A, B), \nabla(C, D)))$ **(Figure 8)**

9. $P = \bar{A}\bar{B}\bar{E} + \bar{E}\bar{C}\bar{D} + \bar{A}\bar{B}F + \bar{C}\bar{D}F$
$= (\bar{A}\bar{B} + \bar{C}\bar{D})(\bar{E} + F)$
$= [(A, B) + \nabla(C, D)].(\nabla\nabla(\nabla E, F))$
$= (\nabla\nabla(\nabla(A, B), \nabla(C, D))).(\nabla\nabla(\nabla E, F))$
$= \nabla(\nabla\nabla\nabla(\nabla(A, B), \nabla(C, D)), \nabla\nabla\nabla(\nabla E, F))$
$= \nabla(\nabla(\nabla(A, B), \nabla(C, D)), \nabla(\nabla E, F))$

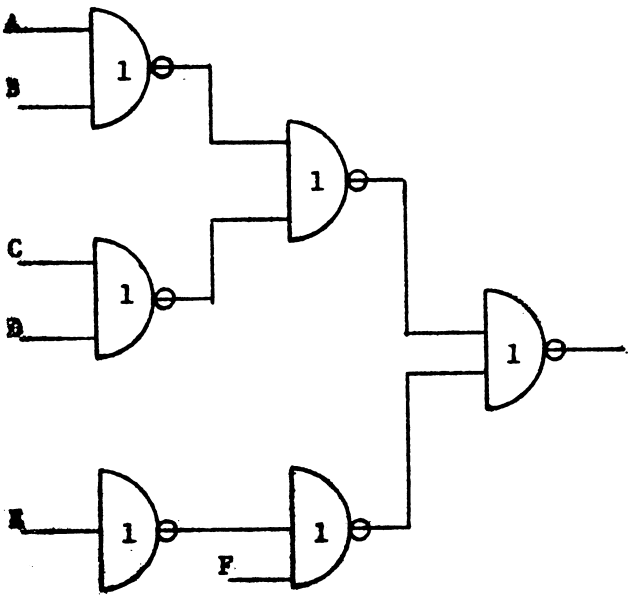

Fig. 8

as in example 8.

10. $P = (B + F + H)(B + D + E)(A + \bar{B} + H)$
$(B + \bar{D} + H)(A + \bar{B} + G)(B + \bar{D} + G)$
$(B + F + G)(A + \bar{B} + D)(A + C)$
$= \nabla[\nabla(B, F, H), \nabla(B, D, E), \nabla(A, \nabla B, H), \nabla(B, \nabla D, H), \nabla(A, \nabla B, G), \nabla(B, \nabla D, G), \nabla(B, F, G), \nabla(A, \nabla B, D), \nabla(A, C)]$
$= \nabla[\nabla(A, C), \nabla(B, D, E), \nabla(B, F, \nabla(\nabla G, \nabla H)), \nabla(B, \nabla D, \nabla(\nabla G, \nabla H)), \nabla(A, \nabla B, \nabla(\nabla D, \nabla G, \nabla H))]$
$= \nabla[\nabla(A, C), \nabla(B, D, E), \nabla(B, \nabla D, \nabla F), \nabla(\nabla G, \nabla H)), \nabla(A, \nabla B, \nabla(\nabla D, \nabla G, \nabla H))]$
$= \nabla[\nabla(A, C), \nabla\{B, \nabla(\nabla(D, E), \nabla(\nabla(D, \nabla F), \nabla(\nabla G, \nabla H)))\}, \nabla(A, \nabla B, \nabla(\nabla D, \nabla G, \nabla H))]$
$= \nabla[\nabla(A, C), \nabla\{\nabla(\nabla B, \nabla(A, \nabla(\nabla D, \nabla G, \nabla H))), \nabla(B, \nabla(D, E), \nabla(\nabla(D, \nabla F), \nabla(\nabla G, \nabla H)))\}]$
$= \nabla[\nabla(A, C), \nabla\{\nabla(\nabla A, \nabla B), \nabla(\nabla B, \nabla D, \nabla G, \nabla H), \nabla(B, \nabla D, \nabla G, \nabla H), \nabla(B, D, \nabla E, \nabla F), \nabla(B, \nabla E, \nabla G, \nabla H)\}]$
$= \nabla[\nabla(A, C), \nabla\{\nabla(\nabla A, \nabla B), \nabla(\nabla D, \nabla G, \nabla H), \nabla(B, \nabla E, \nabla(\nabla(D, \nabla F), \nabla(\nabla G, \nabla H)))\}]$
$= \nabla[\nabla(A, C), \nabla\{\nabla(\nabla A, \nabla B), \nabla[\nabla(D, \nabla(B, \nabla E)), \nabla(\nabla(D, \nabla F), \nabla(\nabla G, \nabla H))]\}]$ **(Figure 9)**

Further considerations

1. *Fan-in*

The number of arguments in a list (e.g. n in $\triangle(A_1, A_2, \ldots, A_n)$) is the fan-in of the corresponding gate.

If there is a fan-in restriction, it may be necessary to manipulate the expression to reduce the length of argument lists to the maximum permitted fan-in. This can be achieved with results such as the following:

$$\triangle(A, B, C) \equiv \triangle(A, \triangle\triangle(B, C))$$
$$\nabla(A, B, C) \equiv \nabla(A, \nabla\nabla(B, C))$$

[Proof: $\triangle\triangle(B, C) = B.C$
$\therefore \quad \triangle(A, \triangle\triangle(B, C)) = \triangle(A, BC)$
$= \bar{A} + (\bar{B} + \bar{C}) = \bar{A} + \bar{B} + \bar{C} = \triangle(A, B, C)$
and similarly for ∇].

This asserts that the two circuits of **figure 10** are equivalent.

However, if $\triangle(A, B, C)$ is replaced by $\triangle(A, \triangle\triangle(B, C))$, further algebraic simplification may well be possible, and a simpler circuit produced.

2. *Fan-out*

Fan-out means the number of gates to which the output signal from a given gate is taken as input.

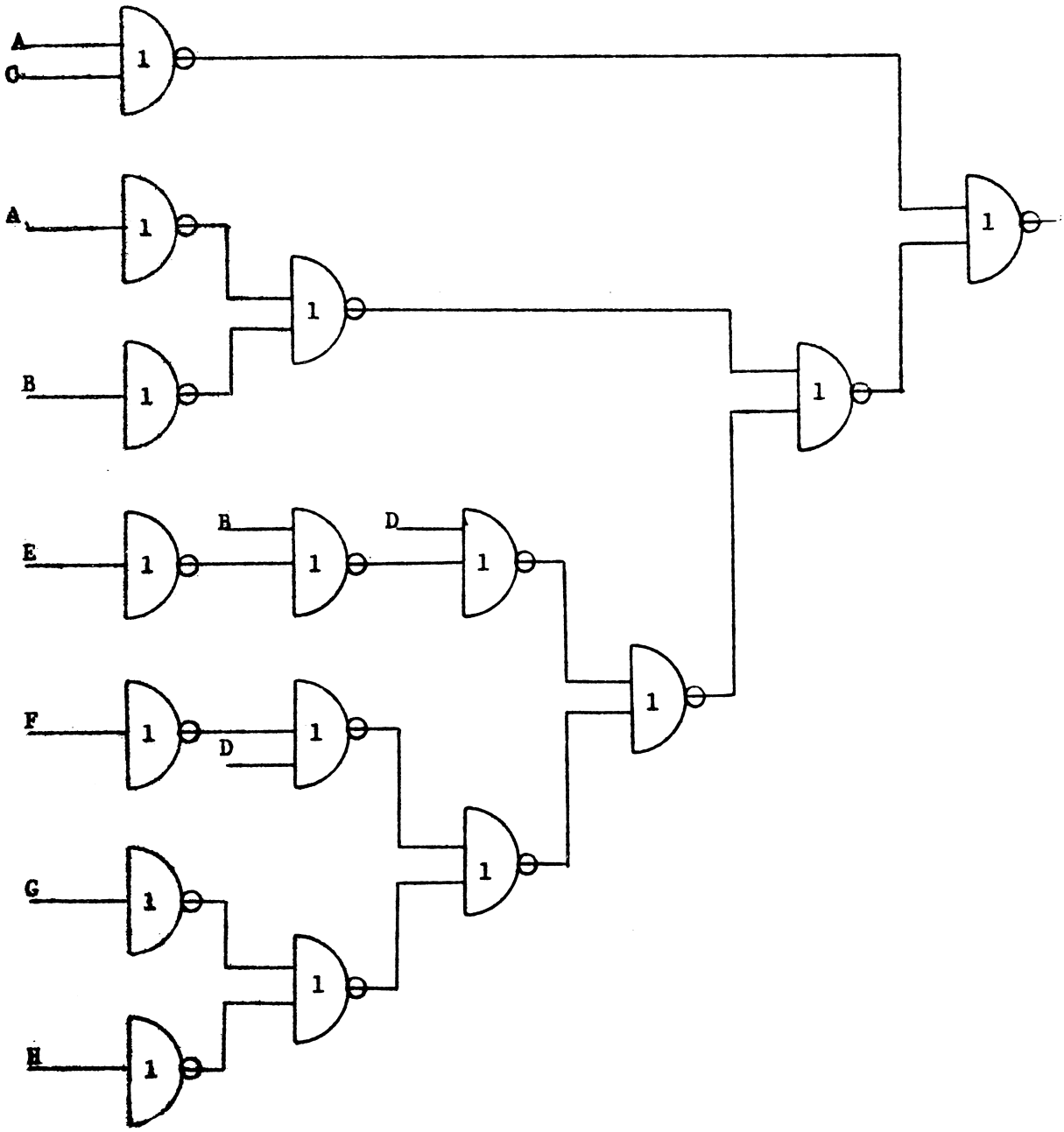

Fig. 9

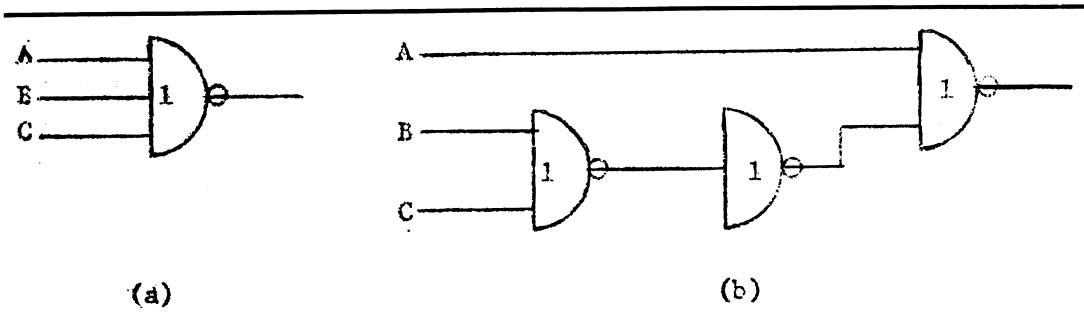

Fig. 10

In the ∇, $\triangle$ expressions, taken as they stand, a fan-out of one is implied, since each operator is an operand only in the list in which it is written.

However, there may well be cases in which a sub-expression is repeated.

e.g. in $P = (A + \bar{B}\bar{C})(D + \bar{B}\bar{C})$
$= \nabla(\nabla(A, \nabla(B, C)), \nabla(D, \nabla(B, C)))$

we have $\nabla(B, C)$ occurring twice—that is, we apparently need two gates each to receive the same pair of signals. If we write $\nabla(B, C) = I$ we have

$$P = \nabla(\nabla(A, I), \nabla(D, I)); I = \nabla(B, C).$$

In the expression for P, I appears twice. This corresponds to a fan-out of two from the gate generating I in the subsidiary expression. Taking the two expressions together, we produce first the two circuits of **figure 11** and therefore finally the circuit of **figure 12**.

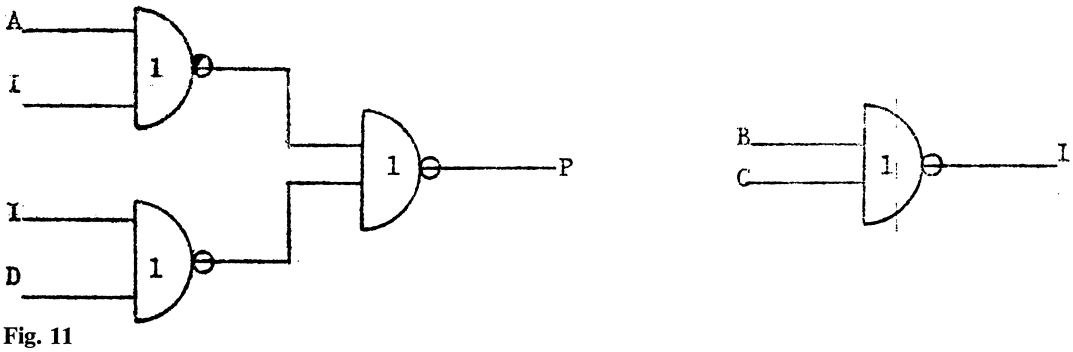

Fig. 11

Whenever a repeated sub-expression is seen in the expression for a circuit, it can be replaced by a new name (true literal) and the subsidiary expression written down separately. The number of occurrences of the new name in the expression is the fan-out required; clearly if there is a fan-out restriction which is less than the number of occurrences, these occurrences will have to be grouped in sets of not more than the fan-out restriction and each set provided with a (physical) gate.

3. *Signal delays*

The signal delay of any signal at any point in the circuit can be seen directly in the expression; starting at the point in the expression corresponding to the signal we simply count the number of operators out to the beginning of the expression. This is the number of gate delays, and so can be multiplied by the gate delay time to give the required signal delay.

E.g. in $P = \underset{1}{\nabla}(\underset{1}{\nabla}(\underset{2}{A}, \underset{2}{I}), \underset{3}{\nabla}(\underset{3}{D}, \underset{3}{I}); I = \underset{4}{\nabla}(\underset{4}{B}, \underset{4}{C})$

we have A delayed by two gates (2 and 1), and D delayed by two gates (3 and 1). B, C are each delayed by gate 4 and then by either 2 and 1 or 3 and 1; in both cases by 3 gates.

Conclusion

We have attempted to justify the use of specially defined NOR and NAND operators in the design of switching circuits. We believe that previous attempts to use similar operators have failed because of an imperfect correspondence between the proposed operators and the actual gates, and we have therefore taken care to ensure as exact a correspondence as possible. In the present paper we have discussed only combinational circuits. We have also used the notations presented here, with modifications to accommodate gate delay times, in the analysis and design of sequential circuits. In particular we have been able to predict, by algebraic means only, the behaviour of sequential circuits under varying race conditions. However this belongs to another paper.

Acknowledgements

The work described in this paper was supported by Grant Nos. A.7636 and A.7651 of the National Research Council of Canada.

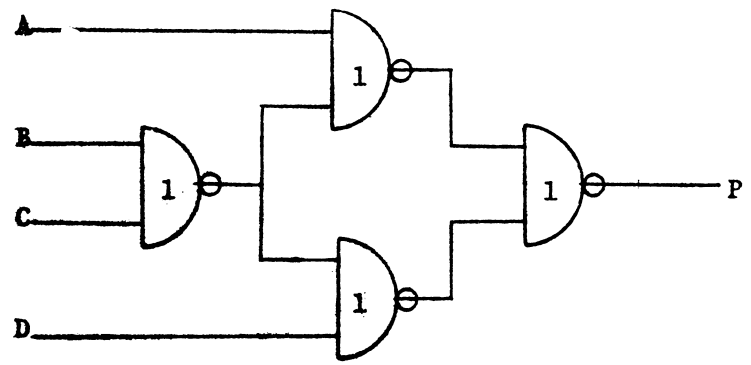

Fig. 12

References

BARNARD, D. F., and HOLMAN, D. F. (1968). The use of Roth's decomposition algorithm in multi-level design of circuits. *Computer Journal*, Vol. 11, pp. 269-276.

DUNCAN, F. G.. and ZISSOS, D. (1971). Fan-in restrictions in logic circuits. *Proceedings of the Institute of Electrical Engineers*, Vol. 118, No. 2, pp. 321-327.

ROTH, J. P., KARP, R. M., MCFARLIN, F. E., and WILTS, J. R. (1961). A Computer program for the Synthesis of Combinatorial Switching Circuits. Proceedings of the 2nd Annual Symposium on Switching Circuit Theory and Logical Design. (AIEE).