

Scan A6845

Fiedler
Correspondence
1991

4 pages

894

6845

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

TELEPHONE: (404) 894-2951

June 21, 1991

Dr. N. J. A. Sloane
Room 2C-376
ATT Bell Labs
Murray Hill, NJ 07974

Dear Neil,

In addition to the material I sent you on the 17th, I am sending some other documentation which possibly may be of interest to you.

The formula for the number of distinct state assignments N for a completely specified state machine having n states and k state variables is given as

$$N = \frac{(2^k - 1)!}{(2^k - n)k!},$$

where $k = \lceil \log_2 n \rceil$. The class usually derives this formula in my switching theory classes. A table of the first few values is given on page 308 of [1] and a partial derivation is given on 307 of [1]. The complete derivation is asked for as Problem 12-1 on page 437 of [2]. Between the two references the derivation is obvious. I have included a tabulation through $n = 32$. I added the 1 state value because a one state "synchronous" machine is a combinatorial circuit and has no change in state and hence no state variables. I calculated the values for the tabulation using Derive and saved them in an ASCII file. The ASCII file was later imported into my math word processor EXP, version 2, and edited therein.

I have observed two other sources of tables with which you are probably already familiar, but I'll mention them anyway. Reference [3], pages 233-242, includes an assortment of the usual tables plus some of the author's own. Reference [4], pages 290-300, has block design tables and tables of Hadamard matrices of the Williamson type. I will include xerox copies of pages from [3] and [4] for your convenience just in case the references are hard to find.

If I can be of further help, please call on me.

Sincerely,

Dan Fielder

Dan Fielder, Professor Emeritus

P.S. Note some extra sheets

6/27



TELEPHONE: (404) 894-2451

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

A7041
→ A6845

Dec. 9, 1991

Dr. N. J. A. Sloane 2c376
AT&T Bell Laboratories
600 Mountain Avenue
PO Box 636
Murray Hill, NJ 07974-0636

Dear Neil:

I ran across another count of state assignments in an out of print text by Woods. I've enclosed the pertinent pages. This count evidently has slightly different restrictions from the one I sent you in my letter of June 21, 1991. For what it's worth, here it is.

I didn't realize until I talked with some of my friends a while back that you were also the Sloane who is the coding theory "guru." Good luck again on your revision.

Sincerely,

Dan Fielder
Professor Emeritus

Lincoln Laboratory Publications

SWITCHING THEORY

Paul E. Wood, Jr.

Lincoln Laboratory

Massachusetts Institute of Technology

1968

Wood 68

Athans and Falb *Optimal Control*

Bartee, Lebow, and Reed *Theory and Design of Digital Machines*

Davenport and Root *Random Signals and Noise*

Green *Digital Computers in Research*

Harman and Honig *Thermoelectric and Thermomagnetic Effects and Applications*

Kleinrock *Communication Nets: Stochastic Message Flow and Delay*

Lax and Button *Microwave Ferrites and Ferrimagnetics*

Shapiro *Prediction of Ballistic Missile Trajectories from Radar Observations*

Wood *Switching Theory*

McGraw-Hill Book Company

New York, St. Louis, San Francisco

Toronto, London, Sydney

Number of Distinct State Assignments It is common practice to produce both delay and amplification with the same device so that in effect an amplifier is inseparably associated with each unit delay. Also, it is often convenient to produce, in addition to the delayed output, the negation of the delayed output. The resulting delay element is illustrated in Fig. 6-24. There are abundant practical reasons for using the complex delay element of Fig. 6-24. First, amplification is very often accompanied by both delay and inversion (e.g., vacuum tube and transistor amplifiers). Second, if passive gates (e.g., diode-resistor gates) are employed in the g network, then the gate elements will have less than unity gain. Now, the use of an amplifier with every unit delay assures that there will be no more than two levels of gates per amplifier (on the assumption of two-level AND-OR- or OR-AND-gate networks), and hence, greater than unity loop gain can be achieved. Finally, memory

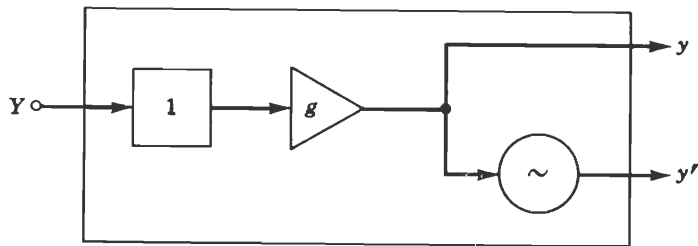


Fig. 6-24 Delay element consisting of a unit delay with associated amplifier and negated output.

elements with characteristics very similar to the delay element of Fig. 6-24 will be introduced in Chap. 7 in order to achieve proper sequential-network operation in the face of input and delay asynchronism.

The delay element of Fig. 6-24 is usually far more complex than a single gate. Therefore, it is reasonable to employ the minimum number of such elements in a sequential-network realization. Hence, it will be assumed that one such delay element will be employed for each feedback connection of Fig. 5-23, where $\lceil \log_2 \#(S) \rceil$ is the number of feedback connections. Thus, it is assumed that $\lceil \log_2 \#(S) \rceil$ state variables will be used to realize a sequential machine with $\#(S)$ nonequivalent states. Now, if, in addition to negated state variables, negated input variables are provided to the sequential network, then NOT gates are not required in the f or g networks.

Given that a minimum number of state variables are used and that negated state and input variables are available without NOT gates, then the number of state assignments N which may result in different f - and g -network complexities can be computed as follows.¹¹ Let the number of nonequivalent states be denoted $s = \#(S)$ and the number of state

variables be denoted $r = \lceil \log_2 s \rceil$. Now, the state assignment involves the assignment of 2^r valuations to s states, which can be done in $2^r! / (2^r - s)!s!$ ways, not including reordering. Since s objects can be reordered in $s!$ different ways, there are a total of $2^r!s! / (2^r - s)!s! = 2^r! / (2^r - s)!$ ways of assigning 2^r valuations to s states. However, not all these assignments produce different f - and g -network complexities. If the i th and j th positions of every assigned valuation are interchanged to generate new assigned valuations, then the new excitation expressions can be obtained from the old by simply substituting y_i for y_j (y'_i for y'_j), and vice versa. But this can be accomplished without altering f - and g -network complexity by simply interchanging y_i and y_j (y'_i and y'_j) inputs. Note that negation of one or more positions may lead to f and g networks of different complexities.¹² Therefore, since there are $r!$ permutation transformations for each assignment, then the value of N is given as

$$N = \frac{2^r!}{(2^r - s)!r!}$$

Values of N for $0 \leq s \leq 9$ are given in Table 6-1.

Table 6-1 Number of Distinct State Assignments N for a Sequential Machine with $\#(S)$ Nonequivalent States

$\#(S)$	$\lceil \log_2 \#(S) \rceil$	N
2	1	2
3	2	12
4	2	12
5	3	1,120
6	3	3,360
7	3	6,720
8	3	6,720
9	4	172,972,800

Reduction of f - and g -network Complexities Using Adjacent State Assignments The results of Table 6-1 make it very clear that, even for sequential machines with a modest number of states, the minimum-complexity realization cannot be obtained by enumerating every state assignment and calculating the network complexity for each. Instead, the state assignment must be generated from the structure of the sequential machine to be realized. The previous two state-assignment procedures (i.e., shift-register realizations and reduction of state-variable dependency) are fruitful only if the machine in question has an SRP or a partition with SP. A machine chosen at random probably will have neither of these partitions. Further, the resulting state assignments are

7041