SYNTHESIS OF SEQUENTIAL MACHINES A MATHEMATICAL APPROACH

JORGE AGUILÓ, ELENA VALDERRAMA, CARLOS SIMÓ

A new mathematical approach for encoding internal states in synchronous sequential machines is developed.

The relationship between transition matrices and the corresponding matrices of the isometric group is studied. An algorithm to determine a priori the null cost assignment is propossed.- The cycle lengths which appear for each n are analysed. The results obtained for the D - or SR flip-flop can be extended to JK and GL.

1. INTRODUCTION.

A mathematical approach for the synthesis of sequential machines is developed. In the synthesis, the state assignment problem remains still without solution. This topic is treated in this paper.

A great number of algorithms have been tested dealing with it from different points of view. Methods based on the search of SP partitions (1),(2) from the states set, or in the search of non-disjoint covers, represent attemps to solve this problem, presenting intermediate results close to ours. Nevertheless, for every sequential machine, they have to look for a convenient set of SPPs which do not let them an "a priori" definition over the existence of a null cost assignment. In addition, the general mathematical handling goes away from real conditions That causes that, some particular case excepted, nor the structures that can be realiced with a given number of memory elements, neither the number of flip flops you need to synthesise a given autonomous automaton have not been reach.

In other way, with SHR's method /3/, we have to test $(2^n-1)!/(2^n-R)!m!$ assignment schemes because of a previous condition about the existence of solutions does not exist. In some cases, when $SAN_j > MNS_j \ \forall_j$, SHR's method is better in the sense that it gets acceptable results in a relatively easy way.

Combinational techniques /4/ are very powerfull ones, but for the moment, have not reach relevant solutions to the problem, and their computing complexity is undoubtedly high.

In this paper, a first step towards a mathematical formulation in which there is no uncertainty in the final solution is given.The starting point is a geometrical representation of the Boolean algebra from a dinamic point of view. In spite of that the optimum cost of an automaton depends of the optimization criterium chose, independently of such criterium, the elements belonging to the same isomorphical class defined in the set of the automata behaviour graphes have all of them the same optimum cost. The study of this optimum is done by identifying the automata with the set of transformations of the Boolean hypercube. We get a first result dealing with the gateless sunchronous counters synthesised with D or SR flip flops (ffs).

- J. Aguiló, E. Valderrama, C. Simó - Departament d'Informàtica de la Universitat Autònoma de Barcelona. Bellaterra (Barcelona)

- Article rebut al Gener de 1982.

An algorithm that find when the synthesis -- with n memory elements ($2^{n-1} < R \leqslant 2^n$, R; -- number of states) and without gates is possible is given. In particular cases, analytical formulas valid from a certain value of n are developed, and, in general, an upper bound - of the number of memory elements needed in - the synthesis without gates is found.

1. ANALYSIS

The transition table of a counter (coded states) can be seen as a one to one mapping --- from B^n into B^n; where $B^n = \{1,-1\}^n$ is a n-- dimensional cube inside R^n .

With this representation, the usual method - of synthesis with D ffs. is reduced to an interpretation of the cube movement.

For example, let's take the following state table:

Present state (P.S.)			Next state (N.S.)		
-1	-1	-1	1	-1	-1
-1	-1	1	-1	-1	1
-1	1	-1	-1	-1	-1
-1	1	1	1	-1	1
1	-1	-1	1	1	1
1	-1	1	-1	1	-1
1	1	-1	-1	1	1
1	1	1	1	1	-1

This state table can be seen as a 3-cube movement (Fig. 1)

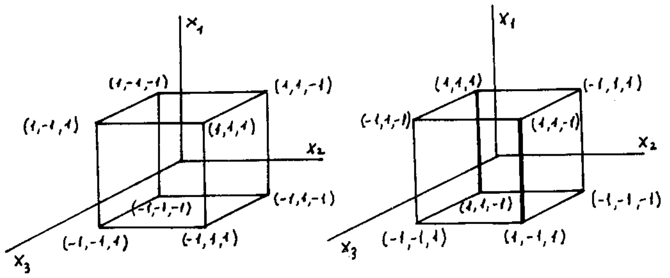


Fig 1

The face $x_1 = 1$ is transformed into the two edges ($x_2 = 1$, $x_3 = 1$) and ($x_2 = 0$, $x_3 = 0$). Then, the imput to the first D ff. has to be $D_1 = x_2x_3 + \bar{x}_2\bar{x}_3$, and, in the same way, $D_2 = x_1$ and $D_3 = \bar{x}_1x_3 + x_1\bar{x}_3$.

We infer that a given counter will be able - to be synthesised without gates if and only if that mapping changes (n-1)-cubes into --- (n-1)-cubes (5)(6).

2. ISOMETRIC TRANSFORMATIONS ON A CUBE

We shall call Δ_n the set of all B^n-isometric transformations. Being clear that Δ_n is a -- subgroup of the symmetric group S_{2^n} and ---- $\sigma \in \Delta_n$ if and only if it changes r-subcubes into r-subcubes. Specifically, it changes - (n-1)-cubes into (n-1)-cubes. Then, it represents a counter which can be synthesised --- without gates.

If d is the Euclidean distance restricted to an n-cube;

$$\sharp(d_x^{-1}(2\sqrt{k})) = \binom{n}{k}$$

where,

$$d_x : \{x\} \times B^n \longrightarrow R^n$$
$$(x\ ,\ y) \longrightarrow d_x(x,y) = d(x.y)$$

In such a way $C \subset B^n$ is a k-subcube of B^n if and only if

$$d_x(C \times C) = \{0,\ 2\sqrt{1}, \dots,\ 2\sqrt{k}\}$$

and

$$\sharp(d_x^{-1}(2\sqrt{j})) = \binom{k}{j};\ \forall x \in C,\ j = 0, 1, \dots, k$$

and we infer that Δ_n is a group isomorphic to the group of all the linear mapping σ_s -- such that,

$$\sigma_s(e_1) = \pm e_{s(i)},\ \text{with } s \in S_{2^n}$$

As a consecuence, any element $\sigma_s \in \Delta_n \subset S_{2^n}$ can be represented by a nxn square matrix of 0,1 and -1, with a sole non null element by row and column.

For example, let's consider the following n-cube isometry σ: (see Fig. 2)

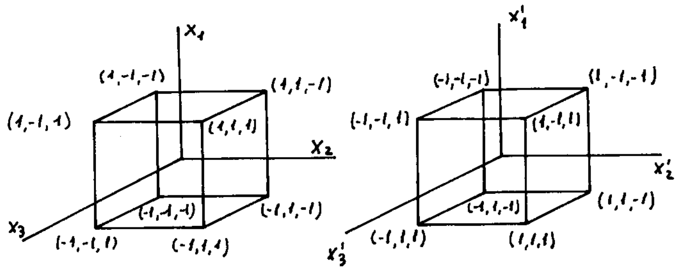


Fig. 2

$\sigma(e_1) = e_2$; $\sigma(e_2) = -e_1$; $\sigma(e_3) = e_3$

where $\{e_1, e_2, e_3\}$ is the canonical base. -- This isometry can be biunivocally represen-- ted by the matrix

$$\sigma = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Accordingly with the above considerations, - the counter whose state table is represented by the 3-cube movement (see Fig. 3) will - be able to be synthesised with 3 D-ffs. and without gates. In addition, the portion of - the non zero elements in the matrix gives us the input function for every ff. That is

$D_1 = B$, $D_2 = \bar{A}$, $D_3 = C$

or, (see Fig. 4).

P.S.			N.S.		
-1	-1	-1	-1	1	-1
-1	-1	1	-1	1	1
-1	1	-1	1	1	-1
-1	1	1	1	1	1
1	-1	-1	-1	-1	-1
1	-1	1	-1	-1	1
1	1	-1	1	-1	-1
1	1	1	1	-1	1

Fig. 3

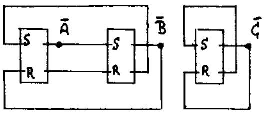


Fig. 4

$S_1 = \bar{R}_1 = B$, $S_2 = \bar{R}_2 = \bar{A}$, $S_3 = \bar{R}_3 = C$

3. SYNTHESIS

We will call two autonomous automata m and -- m' equivalents if and only if their behaviour graphes are isomorphes.

In other words, if we call M and M' to the -- matrices associated to behaviour graphes of - m and m', $(M, M' \in S_{2^n})$,

$$m \sim m' \Leftrightarrow \exists\, U \in S_{2^n} \mid U.M.U^{-1} = M'$$

Matrix U^{-1} gives the assignment done over the states of m. In other words, m and m' are --- equivalents when we can obtained one of them from the other through an assignment.

The conjugation classes defined by the last - relation represent sets of automata having - the same optimum, independently of the optimization criterium chosen.

Given a state table (no coded states),we will see that there exist a state assignment which allows us to synthesise the counter without - gates if and only if its conjugation class -- has at least one element of Δ_n(5) (6) (7).

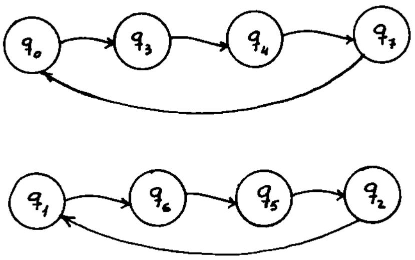
There exist a one to one correspondence between the matrices associated to $\sigma \in S_{2^n}$ and the transition matrices of the graph corresponding to a complete autonomous automation. The previous condition can be expressed in the following way: If M_σ is the transition matrix associated to the autonomous automaton m_σ, $\sigma \in S_{2^n}$ and M_{σ_s} is the transition matrix corresponding to an isometry $\sigma_s \in \Delta_n$, the automaton m_σ can be synthesised without gates through an appopriate assignment if and only if there exist $U \in S_{2^n}$ and σ_s such that:

$$U.M_\sigma.U^{-1} = M_{\sigma_s}$$

As an example, let the automaton m be:

P.S.	N.S.
q_0	q_3
q_1	q_6
q_2	q_1
q_3	q_4
q_4	q_7
q_5	q_2
q_6	q_5
q_7	q_0

Fig. 5



Fiq. 5

Then,

$$M_\sigma = \begin{pmatrix} 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0 \end{pmatrix}$$

and let's take the matrix U,

$$U = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \end{pmatrix}$$

and

$$M_{\sigma_s} = U.M_\sigma.U^{-1} = \begin{pmatrix} 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \end{pmatrix}$$

M_{σ_s} corresponds to the isometry σ of the last example, and, in consecuence, the automaton m_σ can be synthesised without gates as Fig. 4 shows. The optimum assignment is determined by U^{-1}:

uncoded state	0	1	2	3	4	5	6	7
coded state	0	1	5	2	6	7	3	4

Each state assignment which allows the synthesis of the automaton with 0-cost take us the state table determined by σ_s.Generally, for a given M_σ, M_{σ_s} will not be unique.

Let us note, in addition, that being m and m' autonomous automata, their behaviour graphes can be easily represented by their cycle decompositions. Then, m and m' will be equivalent machines if and only if they have the same cycle lengths and, in consecuence, our aim will be to obtain the cycle decomposition of the n-cube isometries. To do so, and if we define the above equivalence relation in Δ_n, we only have to test the cycle lengths of the representative isometries.

4. PROCEDURE

From the considerations above mentioned, the conjugation classes containing at least one isometry can be obtained; and then the optimum assignment for a given state table can be reached from the representative isometry.

The algorithm to obtain these representative isometries has been programmed in a UNIVAC 1108 computer. The time required for the classification is of the order of

$$t(n) = 10^{-2} .\ 1.5^{n}\quad n! \quad \text{msec.}$$

and the memory used is of the order of 1.8^{n} K-words.

5. RESULTS

The table in the next page gives the lengths of the cycles that can be synthesised with a given number of memory elements and without gates.

To make the table shorter, the lengths for every n which are twice the ones of n-1 are omitted.

A set of theoretical results has been obtained which in some sense, are interesting by themselves, and, in the other way, represent hints that make easier the searching.

It has been proved that:

- For a given n, the length of the longest cycle is a multiple of the lengths of the remaining cycles.

- The lowest common multiple of the lengths which appear for each n is such that,

$$\text{l.c.m.}(n) = 2.\prod_{p} p.\log_{p} n \qquad p \text{ prime} \leqslant n$$

in such a way that for a given n, we find all lengths of type l.c.m.$(l_k,\ l_{n-k})$ where l_k is any length obtained in a previous case.

This result includes other author's one (1) (8) that determines that the minimum n in which appears a cycle of length p,

$$p = 2^{r_1} . p_2^{r_2} \ldots p_m^{r_m}$$

is

$$N = \sum_{i=1}^{m} p_i^{r_i} - \left\lceil 2/2^{r_i} \right\rceil$$

From this two results we can infer, for instance, that a sequential machine having one cycle of length 7 and another one of length 5, will be implemented by a number n of memory elements such that the longest length for this n will be a multiple of 35, and in accordance with the second result, this will be only possible for $n \leqslant 12$.

- There exist only one class with a length 2^{2n-2} cycle. Such a class has n.(n-1) elements; that is, there are n.(n-1) equally optimal assignments.

- For $n \geqslant 4$, there exist classes $2^{3.2^{n-3}}$ and 3^{2n-2} with $12\binom{n}{3}$ and $18\binom{n}{3}$optimal assignments each one respectively.

- If $n > 5$, there appear classes $4^{3.2^{n-4}} . 2^{2^{n-4}}$; $6^{n-4} . 3^{2^{n-3}} . 2^{2^{n-4}}$ and $2^{7.2^{n-4}}$.

We can generalize the results for other types of ffs.. For the JK ff., the necessary and sufficient condition so that a counter might be able to be synthesised with only JKs is that the mapping $B^{n} \longrightarrow B^{n}$ transforms every (n-1)-cube in two (n-2)-cubes on the first one or on her complementary. In the case of the GL ffs., the transformation is every (n-1)-cube in two (n-2)-cubes, one on the first one, and the other on the (n-1)-cube complementary to the cube corresponding to the input variable G.

It is clear that for the JK and GL ffs., the conditions are less restrictive than for D ffs. In such a way that the former are included in the later. The number of state tables of zero cost will be greater and as a consequence, the number of null cost conjugation classes is also greater. For two memory elements and for JK ffs. we can synthesise without gates whatever automaton. For three memory elements there exist 712 possible transition tables with 0 cost, a total of 12 conjugation classes and whatever cycle length.

n	LENGTHS
2	1^4, 2^1, 2^1, 4^1
3	3^2, $6^1 2^1$
4	$4^3 2^1$, 8^2, 2^6
5	$4^6 2^4$, 5^6, $10^3 2^1$, $6^2 3^4 2^2$, $12^2 4^2$
6	$6^9 3^2 2^1$, $12^5 4^1$, $4^{12} 2^6$, 3^{20}, $6^{10} 2^2$, 2^{28}
7	7^{18}, $14^9 2^1$, $10^6 5^{12} 2^2$, $20^6 4^2$, $12^6 6^2 4^6 3^4 2^2$, $24^4 8^4$, $12^6 6^4 4^6 2^4$, $6^{12} 3^8 2^{12}$
8	$8^{30} 4^3 2^1$, 16^{16}, $6^{38} 3^4 2^6$, $15^{12} 5^{12} 3^4$, $30^6 10^6 6^2 2^2$, $4^{60} 2^6$, $4^{48} 2^{28}$, $6^{20} 3^{40} 2^4$, 2^{120}
9	$9^{56} 3^2$, $8^{60} 4^6 2^4$, $14^{18} 7^{36} 2^2$, $28^{18} 4^2$, $6^{74} 3^{20} 2^2$. $6^{84} 2^4$, $12^{42} 4^2$, $20^{18} 10^6 5^{12} 4^{12} 2^2$, $40^{12} 8^4$, $20^{18} 10^{12} 4^6 2^4$, $4^{120} 2^{16}$, $10^{36} 5^{24} 2^{12}$, $12^{24} 6^{12} 4^{24} 3^8$, 3^{168}, $6^{56} 3^{16} 2^{56}$, $18^{28} 6^1 2^1$
10	$10^{99} 5^6 2^1$, $8^{120} 4^{12} 2^6$, $8^{120} 4^{16}$, $21^{36} 7^{36} 3^4$, $42^{18} 14^{18} 6^2 2^2$, $24^{40} 8^8$, $12^{60} 6^{38} 4^{12} 3^4 2^6$, 5^{204}, $10^{102} 2^2$, $6^{156} 3^8 2^{28}$, $30^{12} 15^{24} 10^{12} 6^4 5^{24} 3^8 2^4$, $60^{12} 20^{12} 12^4 4^4$, $4^{240} 2^{28}$, $12^{60} 6^{20} 4^{12} 3^{40} 2^4$, $12^{60} 6^{40} 4^{12} 2^8$, $4^{192} 2^{120}$, $6^{120} 2^{120}$, $6^{120} 3^{80} 2^{24}$, 2^{496}

6. REFERENCES

/1/ HALATSIS C, SIGALA M.: "Polylinear de-- composition of synchronous sequential - machines". IEEE Trans. on Computers, -- vol. C-27, nº 12, pp. 144-152

/2/ HARTMANIS J., STEARN R.E.: "Algebraic - structure, theory of sequential machi-- nes". Prentice Hall Inc., New Jersey -- 1966, pp 38-45, 238-246.

/3/ STORY J.R., HARRISON H.J., REINHART E.A "Optimum state assignment for synchro-- nous sequential circuits". IEEE Trans. on Computers, vol. C-21, nº 12, Dec.1972.

---:"Optimum state assignment for syn-- chronous sequential machines with flip flops' memory'. IBM internal Report --- 73 Z-00443.

/4/ HARRISON M.A.: "Counting theorems and - their application to swithing functions" Muckhopadhyay, Recent developments in -- Switching theory. Academic Press 1971, Ch.4.

/5/ AGUILO J., SIMO C.: "Synthese de circuits sequentiels a cout cul". First Meeting AFCET-SMF on Applied Mathematics. ------ Palaiseau, France, Sep. 1973.

/6/ VAQUERO AR., AGUILO J.: "Gateless syn-- chronous counters with D flip flops". - Electronic Letters, vol. 14, Aug.78, -- pp 496-498.

/7/ AGUILO J., VALDERRAMA E., VILLANUEVA JJ. " On the state assignment problem" IEE - Journal on Computer and Digital techni- ques. (To be published).

/8/ HWANG K.: " Periodic realizations of -- synchronous sequential machines." IEEE Trans.on Computers,vol C-22, Oct. 73, pp.923-927.