Kybernetika

Čestmír Šimáně

De Bruijn cycles and their application for encoding of discrete positions

Kybernetika, Vol. 14 (1978), No. 6, (408)--420

Persistent URL: http://dml.cz/dmlcz/124271

KYBERNETIKA — VOLUME 14 (1978), NUMBER 6

De Bruijn Cycles and their Application for Encoding of Discrete Positions

ČESTMÍR ŠIMÁNĚ

By following the line of de Bruijn in studying the P_n cycles by the method of graphs, an algorithm leading directly from a known P_n cycle to a cycle P_{n+1} has been derived by the author. At least a part of all possible P_n cycles can be constructed by this procedure. The applicability of P_n cycles in position encoding systems using one path coding scale is demonstrated both for P_n in fundamental as in the transformed form.

In accordance with [1] a complete de Bruijn cycle P_n is a cycle of 2^n digits (elements) 0 or 1 ordered in such a way, that the 2^n possible sets (combinations) of n consecutive digits in the cycle are all different. All cycles derived from an original one by cyclic permutation are considered to be the same cycle.

Let us denote by a_j the elements of P_n. The n-tuple of consecutive digits $a_i, a_{i+1}, \ldots$ $\ldots, a_{i+n-1}$ be a combination C_i. The neighbouring combination C_{i+1} starting with a_{i+1}, has on the first $n - 1$ places the same digits as C_i on the last $n - 1$ places and on the end the element a_{i+n}, which can be either 0 or 1. We can assign to the n-tuple forming the combination C_i significance of a binary number A_i, the decimal value of which is given by

$$A_i = a_i 2^{n-1} + a_{i+1} 2^{n-2} + \ldots + a_{i+n-1} 2^0 . \tag{1}$$

The decimal value of A_{i+1} is evidently related to A_i by the relations

$$A_{i+1} = 2A_i \quad \text{or} \quad A_{i+1} = 2A_i + 1 \tag{2}$$

depending on whether the element a_{i+n} is 0 or 1. Note that if $A_{i+1} > 2^n - 1$, the modulus 2^n has to be subtracted.

The following formula was derived by de Bruijn [2] for the total number N_n of complete cycles of the length 2^n elements

$$N_n = 2^{2^{n-1}-n} . \tag{3}$$

In any sequence of two kinds of elements one can define groups of m elements as sets of m consecutive identical elements limited on both sides by at least one element of the other kind. These two limiting elements are thus unseparable from the group. A group of $m = n$ elements can occur in P_n once, because there exists but one combination of n identical elements of one kind. A group $m = n - 1$ must not occur at all, because combinations, in which such a group would occur, are formed already by four elements of the group $m = n$ with one of the limiting elements. The group $m = n - 2$ gives, with the two elements of other kind, just one combination of n elements. There are two groups of $m = n - 3$ elements, because $n - 1$ places in the combination are occupied by the group with the two limiting elements, so that one place is left free to be filled either by 0 or 1. By analogous reasoning we derive a formula for the number N_m of groups consisting of m elements of one kind, valid for $m \leqq n - 2$

$$N_m = 2^{n-m-2} \tag{4}$$

and for the total number of elements of one kind

$$N = n + \sum_{m=1}^{n-2} m \,.\, 2^{n-m-2} = 2^{n-1} \tag{5}$$

so that the total numbers of the elements 0 or 1 in a P_n cycle are equal and the total number of elements in the cycle equals to 2^n.

Cycles for $n = 1, 2$ and 3 are very simple

$n = 1$ P_1: 0 1 ,

$n = 2$ P_2: 0 0 1 1 ,

$n = 3$ P_3: 0 0 0 1 0 1 1 1 ,

1 1 1 0 1 0 0 0 .

The 16 P_4 cycles are presented in Table 1, some of the P_5 in Table 3 and one P_6 in Table 5. The heuristic method of constructing the de Bruijn cycles becomes more and more difficult with increasing n, although the knowledge of number of groups can be of significant help. Therefore some methods are needed, leading directly to a P_n cycle.

One of the methods of constructing the de Bruijn P_n cycles is based on finding solution of a recurrence formula of the form

$$a_{i+k} = \sum_{j=0}^{k-1} h_j a_{i+j} \tag{6}$$

which leads to a periodical sequence $a_0, a_1, a_2, \ldots$ of $m \leqq 2^n - 1$ elements of the field GF(2). By proper choice of the coefficients h_j, one can obtain the maximum possible periodicity (length) of the sequence equal to $2^n - 1$. The length cannot be 2^n, because obviously combination C_i consisting of all 0 must be in this method excluded. For detailed information on this method and its relation to so called shift register generators of maximum length sequences see [3], [4].

Here another method is presented based on the theory of graphs (networks), which has been used by de Bruijn for derivation of his formula (3) and which is in details described either in [1] or [2]. We shall briefly review this method in a bit different

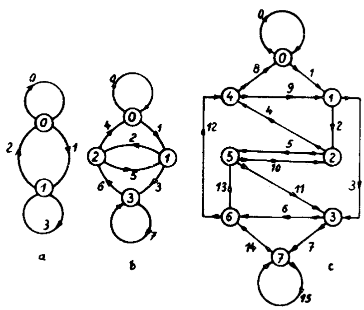

Fig. 1a—c

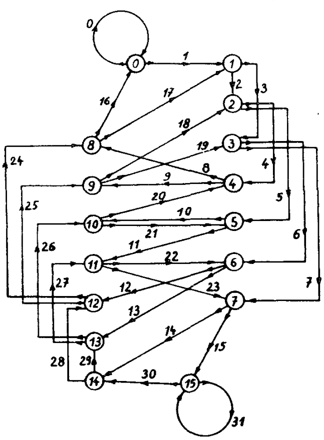

Fig. 1d

form more suitable for our purposes. The essence of the method consists in the construction of an oriented graph with 2^{n-1} junctions p_j and 2^n roads A. This graph will be called G_n. The p_0 junction is denoted by 0, the p_1 by the number 1 and so on, until the last junction with the number $2^{n-1} - 1$, so that under p_j we can simply understand the number of the junction. Each junction has two inputs and two outputs

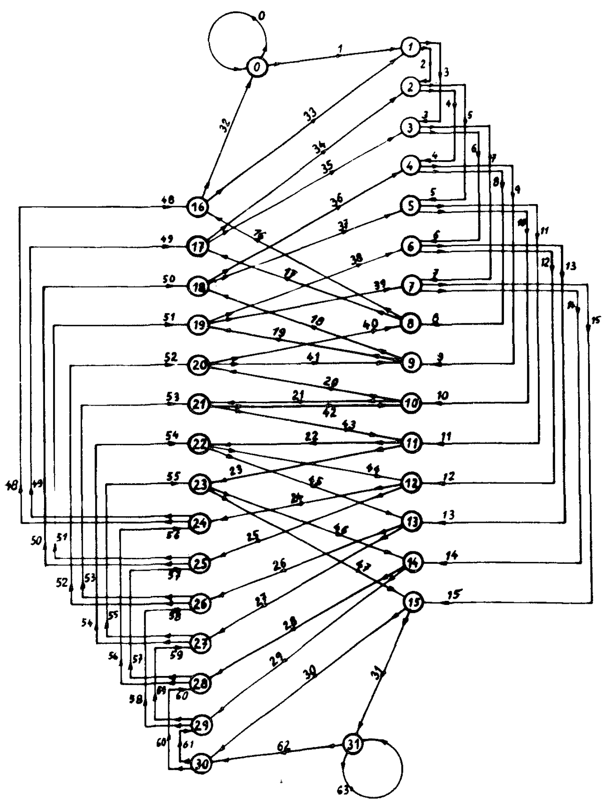

Fig. 1e

for roads A, connecting all the junctions. Let one of the road start in junction p_r and end in junction p_s. The rule for connecting the junctions is given by a relation between the numbers p_r and p_s

$$p_s = 2p_r \quad \text{or} \quad p_s = 2p_r + 1 \,. \tag{7}$$

We see, that each of the junctions is a starting point of two roads, one leading to an even and one to an odd junction. Note that if p_s exceeds $2^{n-1} - 1$, then 2^{n-1} must be subtracted.

Each road starting in p_r and ending in p_s is denoted by a number A given by

$$A = 2p_r \quad \text{for even} \quad p_s \tag{8}$$

or

$$A = 2p_r + 1 \quad \text{for odd} \quad p_s \,.$$

The graphs G_2 to G_6 are shown on the Figures 1a to 1e. The close relation of the graphs G_n to the cycles P_n can be most easily seen, if instead of decimal numbering of junctions and roads we use binary numbers. Then the passage (road) from p_r to p_s is denoted by the n-digital number A of the road, where the first digit is the first digit (standing to the left) of p_r, followed by the $n - 2$ overlapping digits of p_r and p_s, and the last digit is identical with the last digit of p_s, which can be either 0 or 1. The passage from the road A_i to the road A_{i+1} through junction p_s is described by two successive passages, namely from junction p_r to p_s and from p_s to p_t. Respecting relations (7) between the numbers of the junctions and using (8) for numbering the roads we obtain relations between numbers of two successive roads

$$A_{i+1} = 2A_i \quad \text{or} \quad A_{i+1} = 2A_i + 1 \,. \tag{9}$$

Again, if A_{i+1} becomes larger than $2^n - 1$, the modulus 2^n is to be subtracted. All A numbers are n-digital numbers and we can identify them with combinations of n digits. We see, that relations between successive A_i's are the same as between successive C_i's (relations (2) and (9)). By accomplishing a so called complete walk in graph G_n, which crosses all the junctions twice and uses all the roads just once in prescribed direction, we do the same as if we would pass through all the combinations C_i in the cycle P_n. The problem of constructing a cycle P_n is in this way reduced to design a complete walk in G_n, it is to find the corresponding sequence A_i of roads. To go back from A_i to P_n, we have simply to write 0 instead of an even A and 1 instead of an odd A.

The aim of this contribution is now to demonstrate a method, which leads directly to the construction of at least a part of all possible cycles. We shall start with the construction of so called doubled graph G_n^d with respect to G_n. The doubled graph [2] consists of 2^n junctions, the numbers of which are identical with the road numbers A_i^n

in G_n. The index n at the road number A indicates, that we deal with roads in G_n. If two successive roads A_i and A_{i+1} cross a junction in G_n, then in G_n^d the junctions numbered A_i^n and A_{i+1}^n must be connected by a road A^{n+1} starting in A_i and ending in A_{i+1}. According to (8), which is valid for any n, the road numbers in G_n^d will be either $2A_i$ or $2A_i + 1$, depending on whether A_{i+1} is even or odd. Because the relation (9) for roads in G_n is in G_n^d valid for junctions and because (9) has the same form as (7), the doubled graph G_n^d is identical with the graph G_{n+1}.

Let us assume, that we know already a complete walk in G_n, described by a road sequence A_i^n. We can reproduce this complete, closed walk in G_{n+1}, going successively through junctions in G_{n+1} denoted by the numbers A_i of the sequence in G_n. We obtain a walk in G_{n+1}, which is closed but not complete. All junctions have been crossed only once except the junctions 0 and $2^n - 1$, which have been crossed twice. In such a way, $2^n - 2$ junctions, each with one input and one output not used in previous walk, are disponible for performing another closed walk in G_{n+1}. We shall denote the first walk by Q_{n+1}, the second by R_{n+1}. All junctions connected by R_{n+1} are common to both Q_{n+1} and R_{n+1}. The simplest way to combine these two walks in a complete one is to start the walk in Q_{n+1}, then in any of the $2^n - 2$ common junctions pass to R_{n+1}, accomplish the walk in R_{n+1}, return to Q_{n+1} and finish the walk. From one known P_n we can construct by the above procedure $N'_{n+1} = 2^n - 2$ cycles P_{n+1}, so that

(10) $$N'_{n+1}/N_n = 2^n - 2\,.$$

According to (1), the fraction

(11) $$N_{n+1}/N_n = 2^{2^{n-1}-1}$$

and we see, that

(12) $$N'_{n+1}/N_{n+1} = (2^n - 2)/(2^{2^{n-1}-1})\,.$$

The last fraction is equal for $n = 3, 4$ and 5 to 3/4, 7/64, and 15/16384 respectively. Its value decreases rapidly with increasing n, obviously, because the role of more than one interconnection between Q_{n+1} and R_{n+1} becomes more and more important.

The above procedure can be performed by a simple algorithm, by the use of which the necessity of designing the graphs is fully eliminated. It is deduced from relations (9), valid for road numbers in G_n and G_{n+1}. We start with a sequence A_i in G_n. Before using (9) we have to write a new sequence A'_i, differing from the previous one by writing twice the road numbers 0 and $2^n - 1$, because there are loops in these junctions in G_{n+1}. Now we write the sequence Q_{n+1} by multiplying the road numbers in A'_i by two and adding 0 or 1, depending on wheter the next road number in A'_i is even or odd. The results are written below the road numbers A'_i. The sequence R_{n+1} must contain all numbers from the interval 0 to $2^{n+1} - 1$, which didn't appear in Q_{n+1}

in the order given by (9). The whole procedure is illustrated by the following example, where starting from a P_3, one P_4 and one P_5 have been constructed:

P_3:	0	0	0	1	0	1	1	1	
A_3:	0	1	2	5	3	7	6	4	
A'_3:	0	0	1	2	5	3	7	7	6 4
Q_4:	0	1	2	5	11	7	15	14	12 8
R_4:	3	6	13	10	4	9			

if Q_4 and R_4 are combined in 2 and 4, then

A_4:	0	1	2	4	9	3	6	13	10	5	11	7	15	14	12	8		
P_4:	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0		
A'_4:	0	0	1	2	4	9	3	6	13	10	5	11	7	15	15	14	12	8
Q_5:	0	1	2	4	9	19	6	13	26	21	11	23	15	31	30	28	24	16
R_5:	3	7	14	29	27	22	12	25	18	5	10	20	8	17				

if Q_5 and R_5 are combined in 4 and 8, then

A_5:	0	1	2	4	8	17	3	7	14	29	27	22	12	25	18	5	10	20	9	19	6	13	26	21	11	23
P_5:	0	1	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1

15	31	30	28	24	16
1	1	0	0	0	0

In such a way we can obtain a P_n for any n.

POSITION ENCODING SYSTEMS USING DE BRUIJN CYCLES

An interesting field of application of this type of cycles is the encoding of discrete positions i for example of a rotating wheel. One can materialize the cycle P_n either on the periphery or on some circle of diameter r on the wheel in form of two kinds of elements, one corresponding to 0, the other to 1, and use some type of sensors looking at n neighbouring elements and reproducing their values at their outputs. Their output of the sensors interpreted as A_i is in encoded form the position i of the wheel. To get i from A_i, some type of decoder $i(A_i)$ has to be incorporated into the system.

The position code P_n on the wheel needs but one coding path in the coding scale. This is perhaps the most important feature as compared with the standard systems using one path per every bit of the binary output. Fig. 2a and 2b illustrate in schematic way the difference between these two systems for encoding of 16 positions. It is clear, teresting, because after the transformation it has the same form as P_4^1. The sequence that one path scale using de Bruijn cycle is much more compact, but necessitates an additional decoder.

Sometimes it may be advantageous to have the sensors separated by more than the distance between two neighbouring scale elements. In this case we can transform the cycle P_n into another cycle P_n^m, which is felt by sensors separated by m distances between the code elements. The length of the cycle P_n must not be divisible by m.

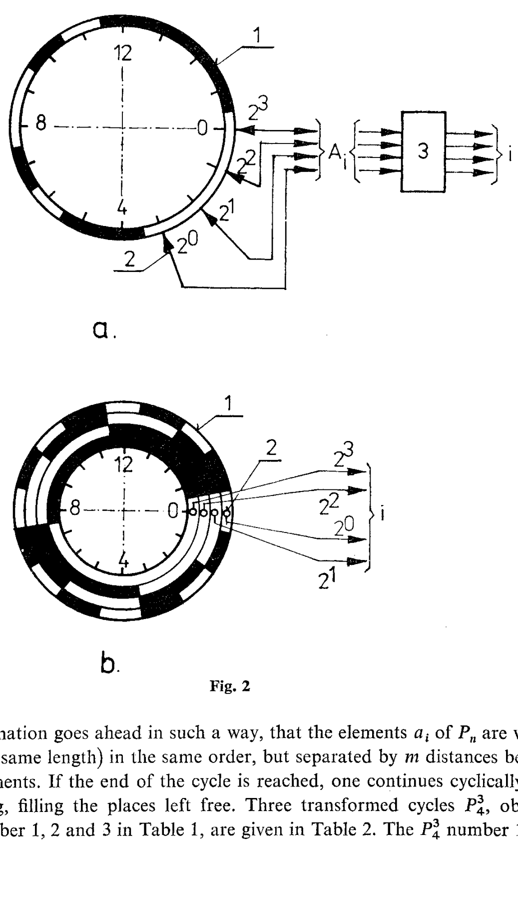

Fig. 2

The transformation goes ahead in such a way, that the elements a_i of P_n are written in P_n^m (of the same length) in the same order, but separated by m distances between the code elements. If the end of the cycle is reached, one continues cyclically from the beginning, filling the places left free. Three transformed cycles P_4^3, obtained from P_4^1 number 1, 2 and 3 in Table 1, are given in Table 2. The P_4^3 number 1 is in-of A_i as seen by the sensors, is of course different. The P_4^3 numbers 2 and 3 are cycles with a minimum number of groups. There are only one group of 5, one of 2 and one of 1 elements in the code scale. A transformed P_5^5 is presented in Table 4. Two schemes of systems using transformed P_4^3 cycles for encoding of 16 positions are shown on Fig. 3a and 3b.

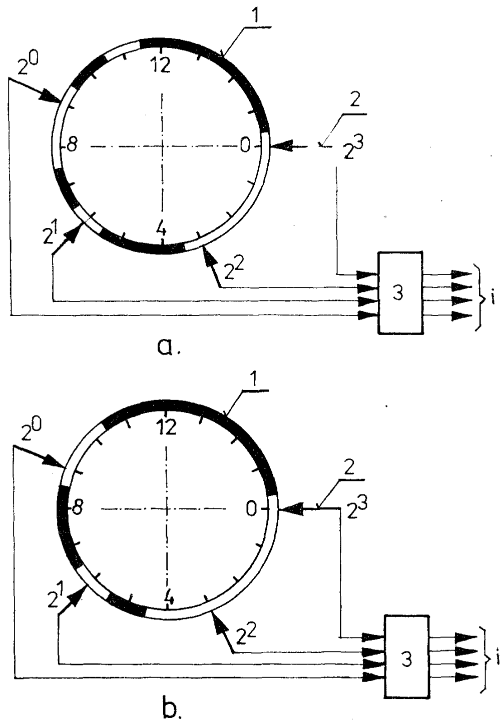

Fig. 3

SHORTENED (INCOMPLETE) CYCLES

The method of graphs offers the possibility to find easily walks, which are closed but not complete. These walks represent then shortened, incomplete cycles, suitable for encoding of number of positions smaller than 2^n. We can perform for example

Table 1. P_4 Cycles

Cycle number	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	P_4	0	0	0	0	1	1	0	1	0	0	1	0	1	1	1	1
	A_i	0	1	3	6	13	10	4	9	2	5	11	7	15	14	12	8
2	P_4	0	0	0	0	1	1	0	1	1	1	1	0	0	1	0	1
	A_i	0	1	3	6	13	11	7	15	14	12	9	2	5	10	4	8
3	P_4	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	1
	A_i	0	1	3	6	13	10	5	11	7	15	14	12	9	2	4	8
4	P_4	0	0	0	0	1	0	0	1	1	1	1	0	1	0	1	1
	A_i	0	1	2	4	9	3	7	15	14	13	10	5	11	6	12	8
5	P_4	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
	A_i	0	1	2	4	9	3	6	13	10	5	11	7	15	14	12	8
6	P_4	0	0	0	0	1	0	1	0	0	1	1	0	1	1	1	1
	A_i	0	1	2	5	10	4	9	3	6	13	11	7	15	14	12	8
7	P_4	0	0	0	0	1	0	1	0	0	1	1	1	1	0	1	1
	A_i	0	1	2	5	10	4	9	3	7	15	14	13	11	6	12	8
8	P_4	0	0	0	0	1	0	1	1	0	1	0	0	1	1	1	1
	A_i	0	1	2	5	11	6	13	10	4	9	3	7	15	14	12	8
9	P_4	0	0	0	0	1	0	1	1	0	0	1	1	1	1	0	1
	A_i	0	1	2	5	11	6	12	9	3	7	15	14	13	10	4	8
10	P_4	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	1
	A_i	0	1	2	5	11	7	15	14	13	10	4	9	3	6	12	8
11	P_4	0	0	0	0	1	0	1	1	1	1	0	0	1	1	0	1
	A_i	0	1	2	5	11	7	15	14	12	9	3	6	13	10	4	8
12	P_4	0	0	0	0	1	1	1	1	0	1	1	0	0	1	0	1
	A_i	0	1	3	7	15	14	13	11	6	12	9	2	5	10	4	8
13	P_4	0	0	0	0	1	1	1	1	0	1	0	0	1	0	1	1
	A_i	0	1	3	7	15	14	13	10	4	9	2	5	11	6	12	8
14	P_4	0	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1
	A_i	0	1	3	7	15	14	13	10	5	11	6	12	9	2	4	8
15	P_4	0	0	0	0	1	1	0	0	1	0	1	1	1	1	0	1
	A_i	0	1	3	6	12	9	2	5	11	7	15	14	13	10	4	8
16	P_4	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1
	A_i	0	1	3	7	15	14	12	9	2	5	11	6	13	10	4	8

Table 2. P_4^3 Cycles

Cycle number	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	P_4	0	0	0	0	1	1	0	1	0	0	1	0	1	1	1	1
	A_i	0	7	4	1	15	9	3	14	2	6	12	5	13	8	11	10
2	P_4	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	1
	A_i	0	2	7	1	5	15	3	10	14	6	4	12	13	8	9	11
3	P_4	0	1	0	0	1	1	0	0	0	0	0	1	1	1	1	1
	A_i	0	12	5	1	9	11	3	2	7	6	4	15	13	8	14	10

Table 3. P_5 Cycles

1	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	P_5	0	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1
	A_i	0	1	2	4	8	17	3	6	13	26	21	11	22	12	25	18
	i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	P_5	0	0	1	0	1	0	0	1	1	1	0	1	1	1	1	1
	A_i	5	10	20	9	19	7	14	29	27	23	15	31	30	28	24	16
2	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	P_5	1	1	0	1	1	1	0	1	0	1	1	0	0	0	0	0
	i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	P_5	1	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0
3	P_5	1	0	0	1	1	1	1	1	0	1	1	1	0	0	1	0
		1	0	1	1	0	0	0	1	0	0	0	0	0	1	1	0
4	P_5	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	1
		0	1	1	1	1	1	0	0	1	1	1	0	1	1	0	1
5	P_5	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	1
		0	0	1	1	1	1	1	0	1	1	0	1	0	1	1	1
6	P_5	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0
		1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0
7	P_5	1	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0
		1	0	0	0	0	0	1	1	0	1	1	1	0	0	1	0
8	P_5	0	0	1	0	1	0	0	0	0	0	1	1	0	1	1	1
		0	0	0	1	0	0	1	1	1	1	1	0	1	0	1	1

Table 4. P_5^5 Cycle (obtained from cycle 1 in Table 3.)

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P_5	0	0	0	0	1	0	1	1	0	0	0	1	1	1	0	0
A_i	0	12	15	6	19	1	25	31	13	7	2	18	30	26	14	4
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P_5	0	1	1	1	0	0	1	0	1	1	1	1	1	1	0	0
A_i	5	28	21	29	8	10	24	11	27	17	20	16	22	23	3	9

Table 5. P_6 Cycle

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P_6	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0
A_i	0	1	2	4	8	16	33	3	6	12	24	49	34	5	10	20
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P_6	1	0	1	0	0	0	1	1	1	0	1	0	0	1	0	1
A_i	40	17	35	7	14	29	58	52	41	18	37	11	22	44	25	50
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P_6	1	0	0	1	0	0	1	1	0	1	1	0	1	0	1	0
A_i	36	9	19	38	13	27	54	45	26	53	42	21	43	23	46	28
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P_6	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1
A_i	57	51	39	15	30	61	59	55	47	31	63	62	60	56	48	32

in the graph G_4 (Fig. 1d) a closed walk 1, 2, 4, 9, 3, 7, 15, 14, 12, 8, which gives a cycle of ten elements 1 0 0 1 1 1 1 0 0 0, applicable for encoding of 10 positions, the problem which often occurs in decimal mechanical counters.

CONCLUSION

The method of construction of P_n cycles described above was derived by the use of the theory of graphs and differs from the method based on shift registers, used as codes generators. It offers the possibility of construction of a relatively large number of P_n cycles, which are either in their fundamental or transformed form suitable for one path coding scales in position encoders.

The author would like to express his gratitude to professor A. Apfelbeck and to Dr. M. Driml for encouraging discussions and interest in his work.

(Received January 25, 1978.)

REFERENCES

[1] Marshall Hall Jr.: Combinatorial Theory. Blaisdall Publishing Company, Toronto—London 1967, chapter 9. (Russian translation.)

[2] N. G. de Bruijn: A Combinatorial Problem. Nederl. Akad. Wetensch. Proc. *49* (1946), 758—764.

[3] W. Wesley Peterson, E. J. Weldon, Jr.: Error-Correcting Codes. The MIT Press, Cambridge, Massachusetts and London, England 1972 (second edition). (Russian translation Mir, Moscow 1976.)

[4] A. Culek et al.: On a Method of Pseudo-Random Numbers Generation. Kybernetika *2* (1966), 1, 215—225.

Prof. Ing. Čestmír Šimáně, fakulta jaderná a fyzikálně inženýrská ČVUT (Faculty of Nuclear Science and Physical Engineering — Czech Technical University), Břehová 7, 110 00 Praha 1. Czechoslovakia.