Syndrome-decoding of binary R=1/2 convolutional codes and a Fortran program for graphical display of the Trellis-decoding procedure

Harkink, H.

Award date:
1976

Syndrome-decoding of binary R=½ convolutional codes and a Fortran program for graphical display of the Trellis-decoding procedure.

Graduate report by
H. Harkink

Group Telecommunication E.C.A.
Prof.dr.ir. J.P.M. Schalkwijk

coached by ir. A.J. Vinck

period: april 1975- nov. 1976

Contents. page

SUMMARY.

Syndrome-decoding of convolutional codes has proved to be superior to Viterbi-decoding with respect to the amount of hardware [4]. In this report the binary R=½ convolutional codes in the class $\mathcal{T}_{\nu,1}$ are considered. The codes in the class $\mathcal{T}_{\nu,1}$ have connectionpolynomials $C_n(\alpha)=\sum_{i=0}^{\nu} c_{n,i}\alpha^i$, n=1,2 with $c_{n,0}=c_{n,\nu}=1$.

The structure of these codes and the syndrome-decoding procedure can at best be described in terms of states and transitions between states. An attractive and instructive way to demonstrate the decoding procedure is by means of the so-called "Trellis" of the code.

In section 1 the principle of syndrome-decoding is explained. The next sections deal with the structure of these codes in terms of the state-table, the state-diagram and the metricequations. In the decoding-algorithm one needs to determine all metriccombinations and the transitions between them. On the basis of a code of constraintlength $\nu=4$, the methods for determination of the characteristics mentioned above, are discussed (sections 2-5,7).

A reduction in metricequations and thus pathregisters, can be achieved [3]. Knowledge of the relationship between the connectionpattern in the polynomials $C_n(\alpha)$, n=1,2 , the free distance d_{free}, the reduction in pathregisters and the total number of metriccombinations is decisive in the choice of a particular code to be implemented as decoder (section 6).

In section 5 all codes of constraintlength $\nu=3$ and $\nu=4$ are analysed in terms of their free distance, the state-table, the state-diagram, the metricequations and the total number of metriccombinations. A comparison of these codes is made and some provisional conclusions are given.

For some codes a maximum of $\nu-1$ bits in the pathregister can be taken as fixed; this leads to an extra saving in hardware (section 8).

In section 9 a begin is made with the analysis of the 24 codes of constraintlength $\nu=6$ which have free distance $d_{free}=9$ or 10.

For demonstration of the Trellis syndrome-decoding procedure a graphical display program is constructed for calculation of the essential features and the decodingparameters for R=½ convolutional codes in the class $\mathcal{T}_{\nu,1}$ (sections 10,11). The decoding procedure can be visualized on the Terminal Display Unit Tektronix T-4014-1 by means of the Trellis of the code. At each time k, for each state S_i, the new metricvalue,

the survivor, the associated transition and the pathregisterbit are calculated and displayed in the Trellis.

The whole program (mainprogram + subroutines) is supplied with explanatory text and extensive comment with references to the theory in the report. Flowdiagrams are made and the subroutines are worked out for an example.

An example of the program is added in the form of a series of so-called "hard copies" of the screen picture.

A "Databook" is enclosed in which the essential features of all codes of constraintlength ν=2-4 and the 24 codes of constraintlength ν=6 are summarized.

1. INTRODUCTION.

In general the connectionpolynomials of a R=½ convolutional code can be expressed as

$$C_n(\alpha) = \sum_{i=0}^{\nu} c_{n,i}\,\alpha^i \quad , \ n=1,2 \ .$$

Here we use the expression $b(\alpha) = \sum_{-\infty}^{\infty} b_i\,\alpha^i$ for any binary sequence $\ldots,b_{-1},b_0,b_1,\ldots\ldots$, where the parameter α serves as a placeholder. We will consider the class $\mathcal{T}_{\nu,1}$ of R=½ convolutional codes of constraintlength ν with $c_{n,0} = c_{n,\nu} = 1$, n=1,2 .

As an example we take a code of constraintlength $\nu=4$ with connectionpolynomials :

$$C_1(\alpha) = \alpha^4 + \alpha^2 + \alpha^1 + \alpha^0$$

$$C_2(\alpha) = \alpha^4 + \alpha^1 + \alpha^0$$

The circuit which forms the so-called "syndrome" $z(\alpha) = C_2(\alpha)n_1(\alpha) + C_1(\alpha)n_2(\alpha)$ is depicted in Fig. 1.

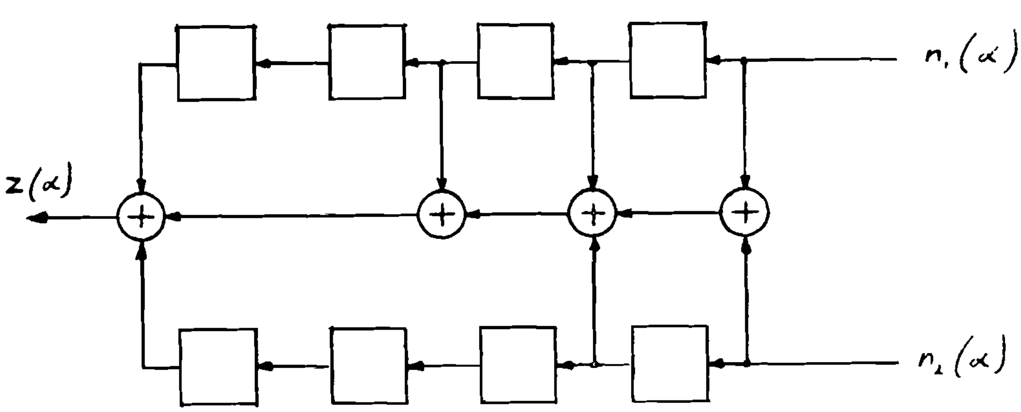


Fig. 1. Syndromeformer.

The additions are mod-2. According to Euclid's algorithm [1] there exist polynomials $D_n(\alpha) = \sum_{i=0}^{\mu} d_{n,i}\,\alpha^i$, n=1,2 , of degree $\mu \leqslant (\nu-1)$ such that $D_1(\alpha)C_1(\alpha) + D_2(\alpha)C_2(\alpha) = 1$. For our code these polynomials are

$$D_1(\alpha) = \alpha^3 + \alpha^2 + \alpha^1$$
$$D_2(\alpha) = \alpha^3 + \alpha^2 \qquad + \alpha^0$$

These polynomials are required for the reconstruction of the original datasequence $x(\alpha)$ out of the received sequences

$$y_1(\alpha) = C_1(\alpha)x(\alpha) + n_1(\alpha) \quad \text{and} \qquad (1)$$
$$y_2(\alpha) = C_2(\alpha)x(\alpha) + n_2(\alpha) \qquad (2)$$

Multiplication of (1) and (2) by $D_1(\alpha)$ and $D_2(\alpha)$ respectively yields

$$y_1(\alpha)D_1(\alpha) + y_2(\alpha)D_2(\alpha) = C_1(\alpha)D_1(\alpha)x(\alpha) + n_1(\alpha)D_1(\alpha) + C_2(\alpha)D_2(\alpha)x(\alpha)$$
$$+ n_2(\alpha)D_2(\alpha) = x(\alpha) + n_1(\alpha)D_1(\alpha) + n_2(\alpha)D_2(\alpha) .$$

Hence $\quad x(\alpha) = y_1(\alpha)D_1(\alpha) + y_2(\alpha)D_2(\alpha) + w(\alpha) \qquad (3)$

where $\quad w(\alpha) = n_1(\alpha)D_1(\alpha) + n_2(\alpha)D_2(\alpha) \qquad (4)$

The sequence $w(\alpha)$ can be thought of as generated by the circuit of Fig. 2.

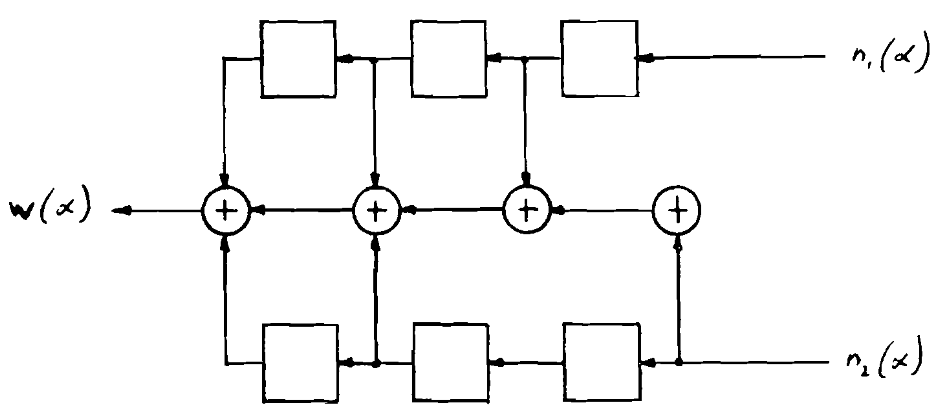


Fig. 2. The $w(\alpha)$-former.

With a recursive algorithm like Viterbi's [2] we can determine out of the syndrome $z(\alpha)$ the noisesequence-pair $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ of minimum Hammingweight that can be a possible cause of this syndrome. With the estimate $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]$ we obtain an estimate $\hat{w}(\alpha)$ by way of the w(α)-former:

$$\hat{w}(\alpha) = \hat{n}_1(\alpha)D_1(\alpha) + \hat{n}_2(\alpha)D_2(\alpha) \tag{5}$$

The estimate $\hat{x}(\alpha)$ of the original datasequence $x(\alpha)$ will then be:

$$\hat{x}(\alpha) = y_1(\alpha)D_1(\alpha) + y_2(\alpha)D_2(\alpha) + \hat{w}(\alpha) \tag{6}$$

The complete decoding scheme is represented in Fig. 2a.

2. STATE-TABLE.

The operation of convolutional codes can at best be described in terms of states and transitions between states. For instructive purposes we draw the syndromeformer in a different way.

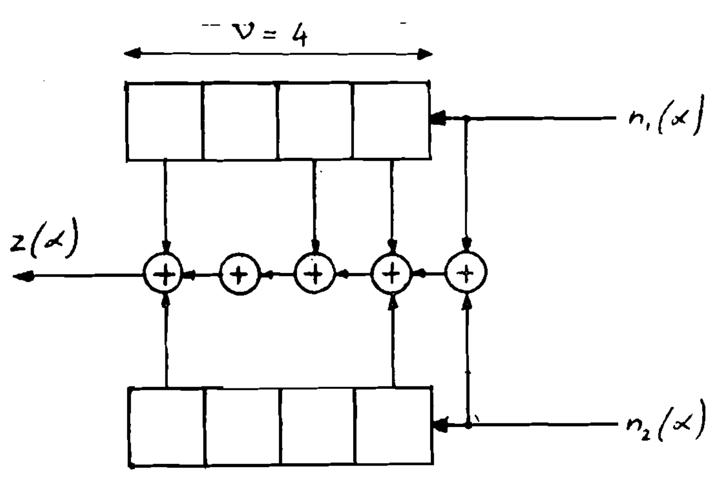


Fig. 3. Syndromeformer.

In general the state of a R=½ convolutional encoder of constraint-length ν is determined by the memorycontents of the ν stages of the shiftregister. The new state is determined by the old stae and the in-coming bit. There are 2^ν so-called "physical" states of the encoder. The corresponding syndromeformer has two shiftregisters and hence $2^{2\nu}$ physical states. The new physical state S(k+1) of the syndrome-

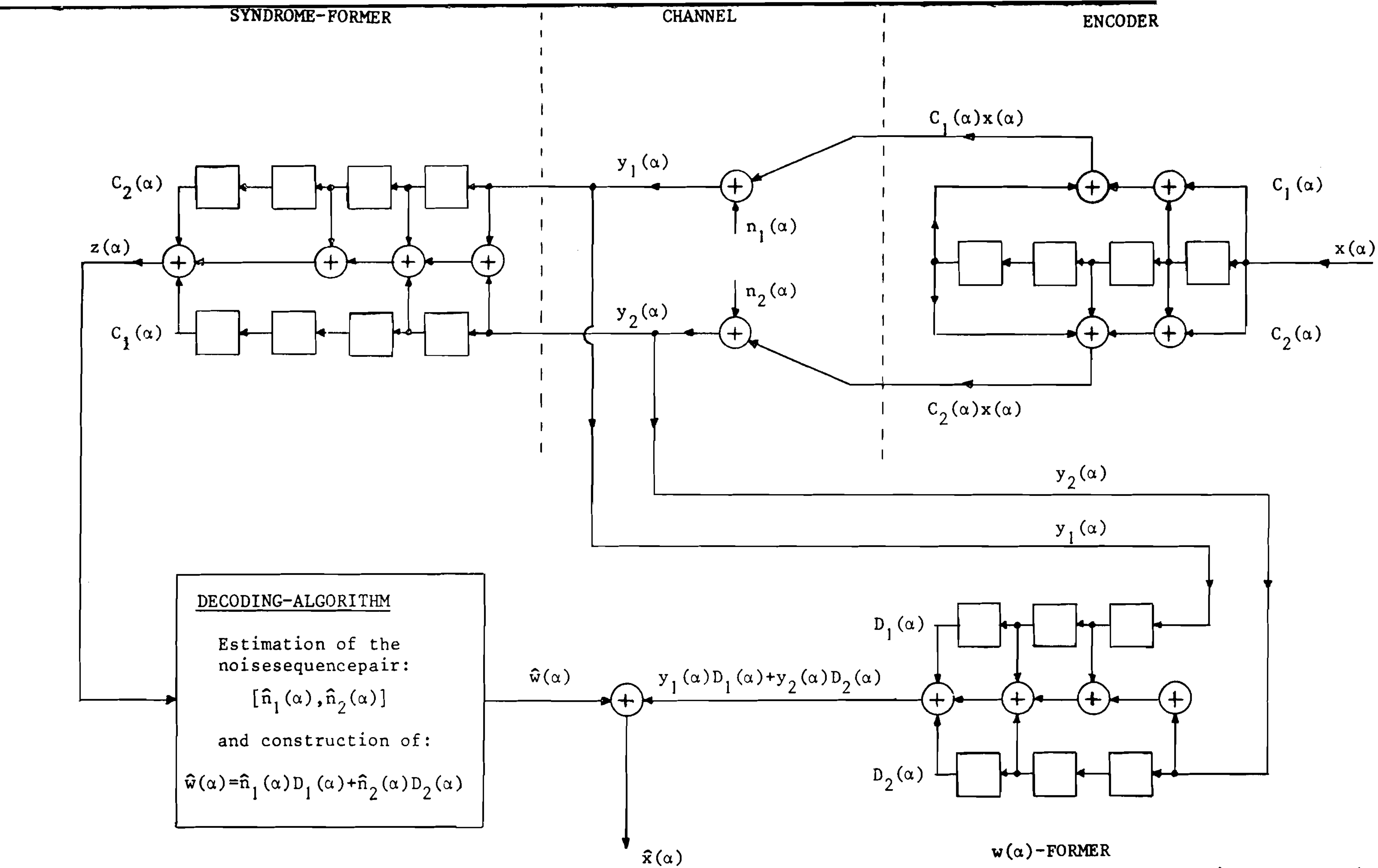


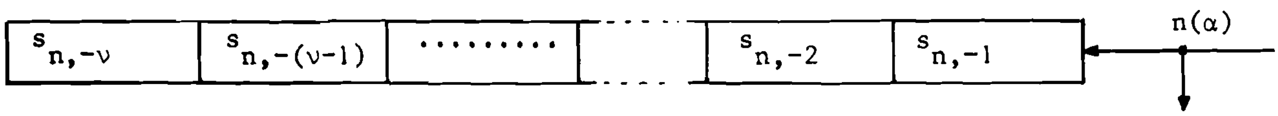
Fig. 2a. Decoding-scheme.

former at time k+1 is determined by the old state S(k) at time k and the bitpair $[n_1(\alpha),n_2(\alpha)]_k$ at time k. The physical state at time k is defined as the contents

$$S(k) = [s_j(\alpha),s_i(\alpha)]_k ,$$

where $s_n(\alpha) = s_{n,-\nu}\alpha^{-\nu} + s_{n,-(\nu-1)}\alpha^{-(\nu-1)} + \ldots + s_{n,-2}\alpha^{-2} + s_{n,-1}\alpha^{-1}$,

$$n = j,i .$$



In [3] is proved that the $2^{2\nu}$ physical states can be divided in 2^ν equivalence classes Kl(i), $i=0,1,\ldots,(2^\nu-1)$. Any physical state in the same equivalence class causes the same outputsequence $[z(\alpha)]_{k1}^{k2}$ in response to an inputsequence $[n_1(\alpha),n_2(\alpha)]_{k1}^{k2}$. Here $[b(\alpha)]_{k1}^{k2}$ indicates that part of the powerseries $b(\alpha)$ for which $k1 \leqslant \text{exponent} \leqslant k2$.

As a representative of each equivalence class Kl(i) we take the so-called "abstract" state S_i which is the unique member for which the topregister is all zero.

$$S_i = [\ 0\ ,\ s_i(\alpha)] \quad , \qquad i=0,1,\ldots,(2^\nu-1).$$

The index i is the decimal representation of the binary contents of the bottom register:

$$s_i(\alpha) = \sum_{k=-\nu}^{-1} s_{i,k}\alpha^k \ , \Longrightarrow \quad i = \sum_{k=-\nu}^{-1} s_{i,k}2^{-(k+1)} .$$

A <u>"zero-equivalent" state</u> is a physical state with the property that if the syndromeformer is in such a state at time k, an all zero input $[n_1(\alpha),n_2(\alpha)]_k^\infty = [0,0]_k^\infty$ gives rise to an all zero output $[z(\alpha)]_k^\infty = [0]_k^\infty$.

A <u>"base" state</u> $S_b = [\alpha^{-1},s_b(\alpha)]$ is a zero-equivalent state with a single "1" in the rightmost position of the topregister of the syndromeformer. The base state always exists and is unique [3].

In Fig. 5 we have again drawn the syndromeformer of our example. (this code will be used as a reference throughout the paper).

The base state of this code can be determined as follows:

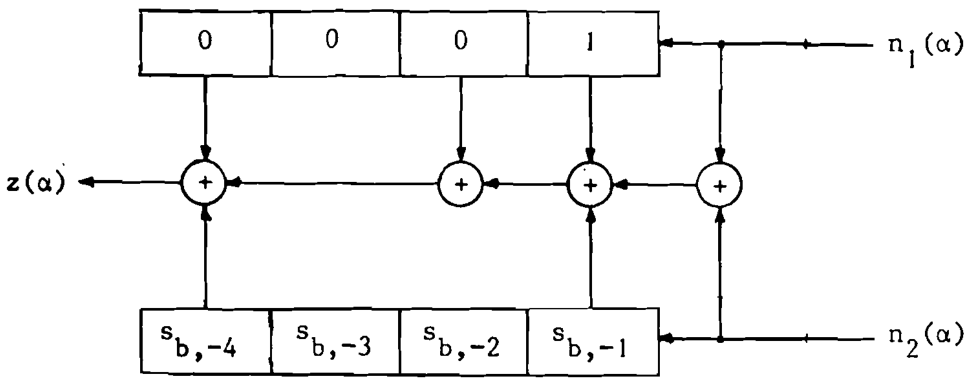


Fig. 5. Base state.

Suppose that the syndromeformer is in the base state $S_b(k) = [\alpha^{-1}, s_b(\alpha)]_k$ at time k. An all zero input $[n_1(\alpha), n_2(\alpha)]_k^{\infty} = [0,0]_k^{\infty}$ must give rise to an all zero output $[z(\alpha)]_k^{\infty} = [0]_k^{\infty}$. Hence

$$[z(\alpha)]_0 = 1 + s_{b,-4} + s_{b,-1} = 0 \longrightarrow s_{b,-4} = 1 + s_{b,-1} = 0$$

$$[z(\alpha)]_1 = 1 + s_{b,-3} = 0 \longrightarrow s_{b,-3} = 1$$

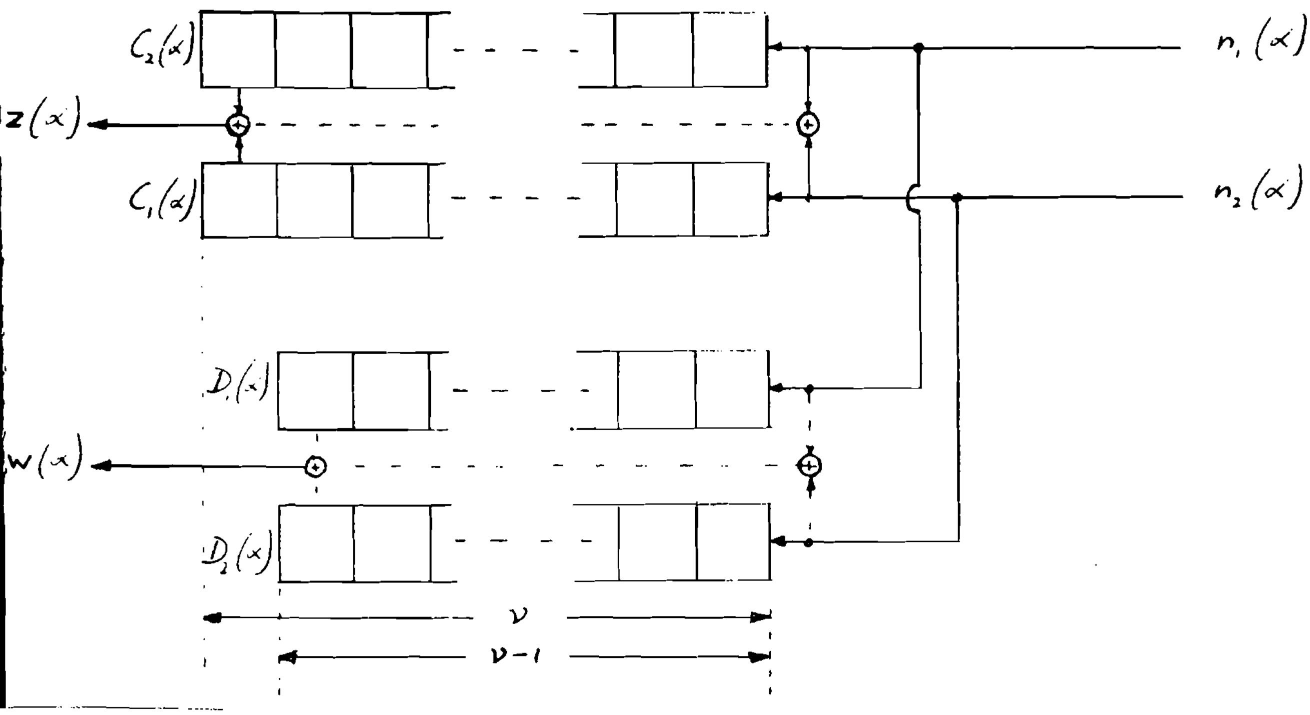
$$[z(\alpha)]_2 = s_{b,-2} = 0 \longrightarrow s_{b,-2} = 0$$

$$[z(\alpha)]_3 = 1 + s_{b,-1} = 0 \longrightarrow s_{b,-1} = 1$$

and consequently the base state is $S_b = [\alpha^{-1}, \alpha^{-3} + \alpha^{-1}] = [\ 0001\ ,\ 0101\ ]$. Since $[0101] \backsim [0.2^3 + 1.2^2 + 0.2^1 + 1.2^0] = [5]$, the index b in S_b is equal to 5. Thus we have the base state $S_{b=5}$.

With each state-transition of the syndromeformer at time k, we also want to determine the corresponding state-transition of the w(α)-former. In order to determine the state-transitions of the w(α)-former one has to determine the base-state of the w(α)-former. We will now prove that the base-state of the w(α)-former is the same as the base-state of the syndrome-former.

Consider the syndrome-former and the w(α)-former of a certain code of constraintlength ν in the class $\mathcal{T}_{\nu,1}$. The code's base-state is defined as $S_b = [\alpha^{-1}, s_b(\alpha)]$.



Suppose that both the syndromeformer and the w(α)-former are in the zero-state S_0. As input, both for the syndromeformer as the w(α)-former we take the sequence $[n_1(\alpha), n_2(\alpha)]_0^{\nu-1} = \alpha^{\nu}[\alpha^{-1}, s_b(\alpha)]$. The responses to this inputsequence are:

$$z(\alpha) = C_2(\alpha)n_1(\alpha) + C_1(\alpha)n_2(\alpha) = P_1(\alpha) \qquad \times D_1(\alpha)$$

$$w(\alpha) = D_1(\alpha)n_1(\alpha) + D_2(\alpha)n_2(\alpha) = P_2(\alpha) \qquad \times C_2(\alpha)$$

$$\overline{} +$$

$$\underbrace{n_2(\alpha)}_{\substack{\text{degree}\\ = \nu-1}} \underbrace{\{C_1(\alpha)D_1(\alpha) + C_2(\alpha)D_2(\alpha)\}}_{\equiv 1} = \underbrace{P_1(\alpha)D_1(\alpha)}_{\substack{\text{degree}\\ \leqslant \nu-1}} + \underbrace{P_2(\alpha)C_2(\alpha)}_{\substack{\text{degree}\\ = \nu}}$$

From this it follows that degree $P_2(\alpha) <$ degree $P_1(\alpha)$. From the **moment** that the base-state fits precisely in the syndromeformer, the output will be zero in response to a zero-input. Since degree $P_2(\alpha) <$ degree $P_1(\alpha)$, the output of the w(α)-former will also be zero from the moment mentioned above. As only a zero-equivalent state of the w(α)-former can produce a zero-output in response to a zero-input, we conclude that we may take the base-state of the syndromeformer as the base-state of the $w(\alpha)$-former.

Note that the $w(\alpha)$-former produces a zero-output one unit of time earlier than the syndromeformer does in the case mentioned above.

We are now ready to construct the state-table of our code. In the table of Fig. 6 in each row is indicated:

- the decimal index j of the state $S_j(k)$.
- the binary representation of the decimal statenumber j, i.e. the binary contents of the bottomregister of the syndromeformer.
- the indices i of the new states $S_i(k+1)$ with transitions $[n_1(\alpha),n_2(\alpha)]_k=$ $[0,0]$, $[0,1]$, $[1,1]$ and $[1,0]$ respectively.
- the outputvalues $[z(\alpha)]_k$ and $[w(\alpha)]_k$ corresponding to the state-transitions.

In general the transitions of a particular state in response to the inputpairs $[n_1(\alpha),n_2(\alpha)]_k = [0,0]$, $[0,1]$, $[1,1]$ and $[1,0]$ take place as indicated in the table of Fig. 7.

We distinguish between states S_j and S_j', where $S_j' = S_j$ except for the rightmost bit in the topregister which is a "1" for the state S_j'.

This table also contains the outputvalues $[z(\alpha)]_k$ for the transitions concerned. Assuming that the output for the transition $[0,0]$ is $[z(\alpha)]_k$, the other outputs can easily be determined.

Suppose that the syndromeformer is in the state $S_j(k)=[0,s_j(\alpha)]=$ $[0\ 0\ .\ .\ 0\ ,\ s_{j,-\nu}s_{j,-\nu+1}\ .\ .\ s_{j,-1}]$. The transitions $[1,1]$ and $[1,0]$ bring us in the physical states $S_{2j+1}'=[0\ 0\ .\ .\ 1\ ,\ s_{j,-\nu+1}s_{j,-\nu+2}\ .\ .s_{j,-1}\ 1]$ and $S_{2j}'=[0\ 0\ .\ .\ 1\ ,\ s_{j,-\nu+1}\ s_{j,-\nu+2}\ .\ .\ s_{j,-1}\ 0]$ respectively. To find the representative of these physical states within the same equivalence-class (i.e. the abstract states), we must add the base-state, mod-2. This addition is always permitted since the base-state is a zero-equivalent state. The response of a physical state $\oplus$ a zero-equivalent state (i.e. the abstract state) to a certain inputsequence, is the same as the response of the physical state alone to that inputsequence.

state $S_j(k)$	state contents	new state $S_i(k+1)$ $[n_1(\alpha),n_2(\alpha)]_k$				value $[z(\alpha)]_k$ $[n_1(\alpha),n_2(\alpha)]_k$				value $[w(\alpha)]_k$ $[n_1(\alpha),n_2(\alpha)]_k$			
j	$[s_j(\alpha)]$	00	01	11	10	00	01	11	10	00	01	11	10
0	0000	0	1	4	(5)	0	1	0	1	0	1	1	0
1	0001	2	3	6	7	1	0	1	0	0	1	1	0
2	0010	4	5	0	1	0	1	0	1	1	0	0	1
3	0011	6	7	2	3	1	0	1	0	1	0	0	1
4	0100	8	9	12	13	0	1	0	1	1	0	0	1
(5)	0101	10	11	14	15	1	0	1	0	1	0	0	1
6	0110	12	13	8	9	0	1	0	1	0	1	1	0
7	0111	14	15	10	11	1	0	1	0	0	1	1	0
8	1000	0	1	4	5	1	0	1	0	0	1	1	0
9	1001	2	3	6	7	0	1	0	1	0	1	1	0
10	1010	4	5	0	1	1	0	1	0	1	0	0	1
11	1011	6	7	2	3	0	1	0	1	1	0	0	1
12	1100	8	9	12	13	1	0	1	0	1	0	0	1
13	1101	10	11	14	15	0	1	0	1	1	0	0	1
14	1110	12	13	8	9	1	0	1	0	0	1	1	0
15	1111	14	15	10	11	0	1	0	1	0	1	1	0

Fig.6. State-table.

As an example we take the transitions of the state S_3. With the base-state $S_{b=5}$ we have :

$$S_3=[0000,0011]\xrightarrow{[0,0]}[0000,0110]=S_6$$

$$S_3=\quad " \quad \xrightarrow{[0,1]}[0000,0111]=S_7$$

$$S_3=\quad " \quad \xrightarrow{[1,1]}[0001,0111]\oplus[0001,0101]=[0000,0010]=S_2=S_7\oplus S_5$$

$$S_3=\quad " \quad \xrightarrow{[1,0]}[0001,0110]\oplus[0001,0101]=[0000,0011]=S_3=S_6\oplus S_5$$

Codes of constraintlength ν in the class $\Gamma_{\nu,1}$.   Base-state $S_b = [\,0\,0\,0\,\ldots\,0\,1\,,\ s_{b,-\nu}\ s_{b,-\nu+1}\ \ldots\ s_{b,-2}\ 1\,]$

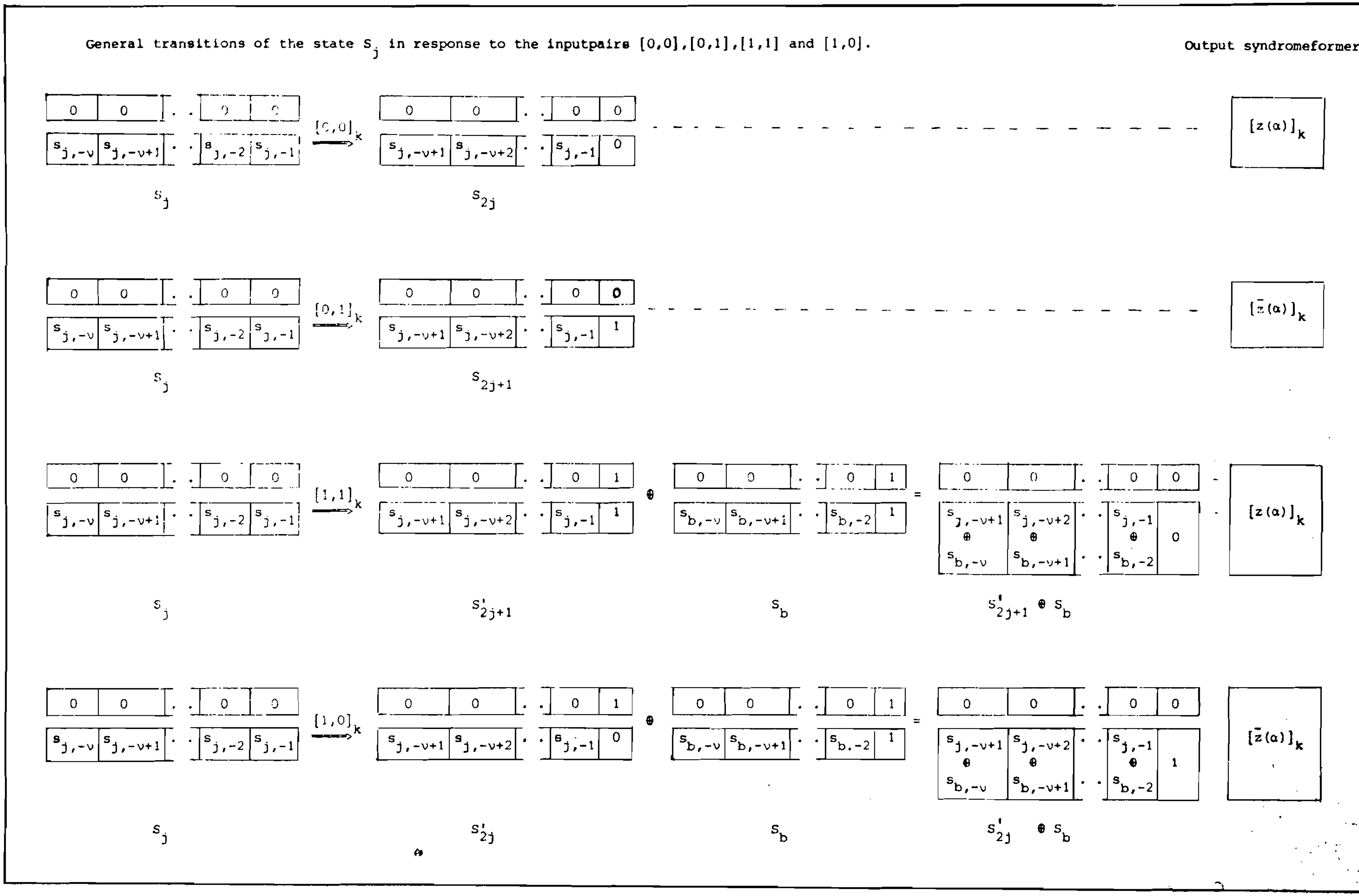


Fig. 7. Transition-table.

We have already proved that the base-state of the w(α)-former is the same as the base-state of the syndromrformer. Hence the state-transitions of the w(α)-former are the same as those of the syndromeformer.

In brief we can express the responses of a particular state $S_j(k)$ as denoted in the transition-table of Fig. 8.

			output	output w(α)	
old state	input	new state	z(α)	$d_{1,0} = 0$	$d_{1,0} = 1$
$S_j(k)$	$\xrightarrow{[0,0]}$	$S_{2j}(k+1)$	$[z(\alpha)]_k$	$[w(\alpha)]_k$	$[w(\alpha)]_k$
$S_j(k)$	$\xrightarrow{[0,1]}$	$S_{2j+1}(k+1)$	$[\bar{z}(\alpha)]_k$	$[\bar{w}(\alpha)]_k$	$[w(\alpha)]_k$
$S_j(k)$	$\xrightarrow{[1,1]}$	$S'_{2j+1}(k+1) \oplus S_b$	$[z(\alpha)]_k$	$[\bar{w}(\alpha)]_k$	$[\bar{w}(\alpha)]_k$
$S_j(k)$	$\xrightarrow{[1,0]}$	$S'_{2j}(k+1) \oplus S_b$	$[\bar{z}(\alpha)]_k$	$[w(\alpha)]_k$	$[\bar{w}(\alpha)]_k$

Fig. 8. Transition-table.

The pattern in the output-values $[w(\alpha)]_k$ can be determined when we know the polynomials $D_n(\alpha)$, n=1,2. In order to satisfy the equation $C_1(\alpha)D_1(\alpha) + C_2(\alpha)D_2(\alpha) = 1$, the polynomials $D_n(\alpha)$ must have complementary coefficients $d_{n,0}$, n=1,2. When we assume that the input $[0,0]_k$ causes an output $[w(\alpha)]_k$ then the output-pattern can be expressed as in Fig. 8. The consequence of say $D_1(\alpha)$ having $d_{1,0}=0$ is that both inputs [0,0] and [1,0] cause an output [w(α)] and both inputs [1,1] and [0,1] an output $[\bar{w}(\alpha)]$.

Filling up the state-table:

The states S_j and $S_{j+2^{\nu-1}}$, $j=0,1,\dots,2^{\nu-1}$, are equal except for the leftmost bit in the bottomregister, i.e.

$$[S_j]^0_{-(\nu-1)} = [S_{j+2^{\nu-1}}]^0_{-(\nu-1)} \quad \text{(the additions are mod-}2^\nu\text{)}.$$

Hence their state-transitions in response to an input $[n_1(\alpha),n_2(\alpha)]$ will be the same. The consequence is that the contents of the lower half of the third column of the state-table are equal to the contents of the upper half:

$$s_j(k) \xrightarrow{[n_1(\alpha),n_2(\alpha)]_k} [\,0\ 0\ 0\ n_1\ ,\ s_{j,-3}\ s_{j,-2}\ s_{j,-1}\ n_2]_{k+1} \qquad [z(\alpha)]_k$$
$$s_{j+2^{\nu-1}}(k) \xrightarrow{[n_1(\alpha),n_2(\alpha)]_k} [\,0\ 0\ 0\ n_1\ ,\ s_{j,-3}\ s_{j,-2}\ s_{j,-1}\ n_2]_{k+1} \qquad [\bar{z}(\alpha)]_k$$

Since $c_{1,0} = c_{2,0} = 1$, complementary inputs $[n_1(\alpha),n_2(\alpha)]_k$ and $[\bar{n}_1(\alpha),\bar{n}_2(\alpha)]_k$ give rise to the same output $[z(\alpha)]_k$ (look at the third column of the state-table).

With $c_{1,\nu} = c_{2,\nu} = 1$ and $[s_j]^0_{-(\nu-1)} = [s_{j+2^{\nu-1}}]^0_{-(\nu-1)}$, $j=0,1,..,2^{\nu-1}$, we conclude that the contents of the lower half of the fourth column of the state-table is the complement of the contents of the upper half.

As the base state of the w(α)-former is the same as the base state of the syndromeformer, the contents of the fifth column of the state-table can easily be determined. We notice that the lower half of the fifth column equals the upper half since degree $D_n(\alpha)$ < degree $C_n(\alpha)$, $n=1,2$ and $[s_j]^0_{-(\nu-1)} = [s_{j+2^{\nu-1}}]^0_{-(\nu-1)}$, $j=0,1,\ldots,2^{\nu-1}$.

The row-pattern in the outputvalues w(α) is:

$$\left.\begin{matrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{matrix}\right\}\ d_{1,0}=0 \qquad\qquad \left.\begin{matrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{matrix}\right\}\ d_{1,0}=1$$

Now that we have constructed the state-table, we can draw the state-diagram. The state-diagram is a more surveyable version of the state-table. All the essential information of a particular code in our class can easily be obtained from its state-diagram.

3. STATE-DIAGRAM.

In the state-diagram each state is represented as a node and the branches correspond to the transitions between the states. Each particular state has four incoming branches and four outgoiing branches. From the four incoming respectively outgoiing branches, one pair corresponds to an output z(α)=0 and the other pair to an output z(α)=1. In order to maintain sufficient visual insight we split the complete state-diagram up

into two diagrams; one for transitions $z(\alpha)=0$ and one for transitions $z(\alpha)=1$. The two state-diagrams of our example are drawn in the figures 9a and 9b. The structure of the state-diagrams for $z(\alpha)=0$ and $z(\alpha)=1$ is the same for all codes of a fixed constraintlength ν. A solid branch corresponds to a transition $w(\alpha)=0$ and a dashed branch to a transition $w(\alpha)=1$
The state-diagrams of two different codes only differ in the numbering of the nodes (the indices j of S_j) and the transitions $[n_1(\alpha),n_2(\alpha)]$ which are placed along the branches.

The state-diagram for $z(\alpha)=0$ can be constructed with the help of the state-table of Fig. 6 as follows:

- First find the two states that convert into themselves for $z(\alpha)=0$. These are the states S_0 and S_3 with transitions [0,0] and [1,0], and with outputs $w(\alpha)=0$ and $w(\alpha)=1$ respectively. Hence the lower node will be the "zero-state" S_0 and the upper node S_3.
- The lower left node will then be S_4 with transition [1,1] and the upper right node S_7 with transition [0,1].
- Find the state which converts into S_0 and S_4, and the state which converts into S_3 and S_7. These two states are S_2 and S_1 respectively. Hence the upper left node is S_1 and the lower right S_2.
- With the help of the $w(\alpha)$-column of the state-table all other nodes can easily be found.

The construction of the state-diagram for $z(\alpha)=1$ is carried out in a similar way. Now we have to find the two states which convert into themselves for $z(\alpha)=1$. These are S_{12} and S_{15} with $w(\alpha)=1$ and $w(\alpha)=0$ respectively.

In order to derive some general properties of the state-diagram we determine the four states which all lead to a particular state S_i. These states are called the "parent-states" S_{p1}, S_{p2}, S_{p3} and S_{p4} of the state S_i.

We distinguish between even states S_i and odd states S_{i+1} where $i=2k$, $k=0,1,\ldots,2^{\nu-1}-1$.

In general the transitions of these parent-states take place as indicated in the transition-scheme of Fig. 10.

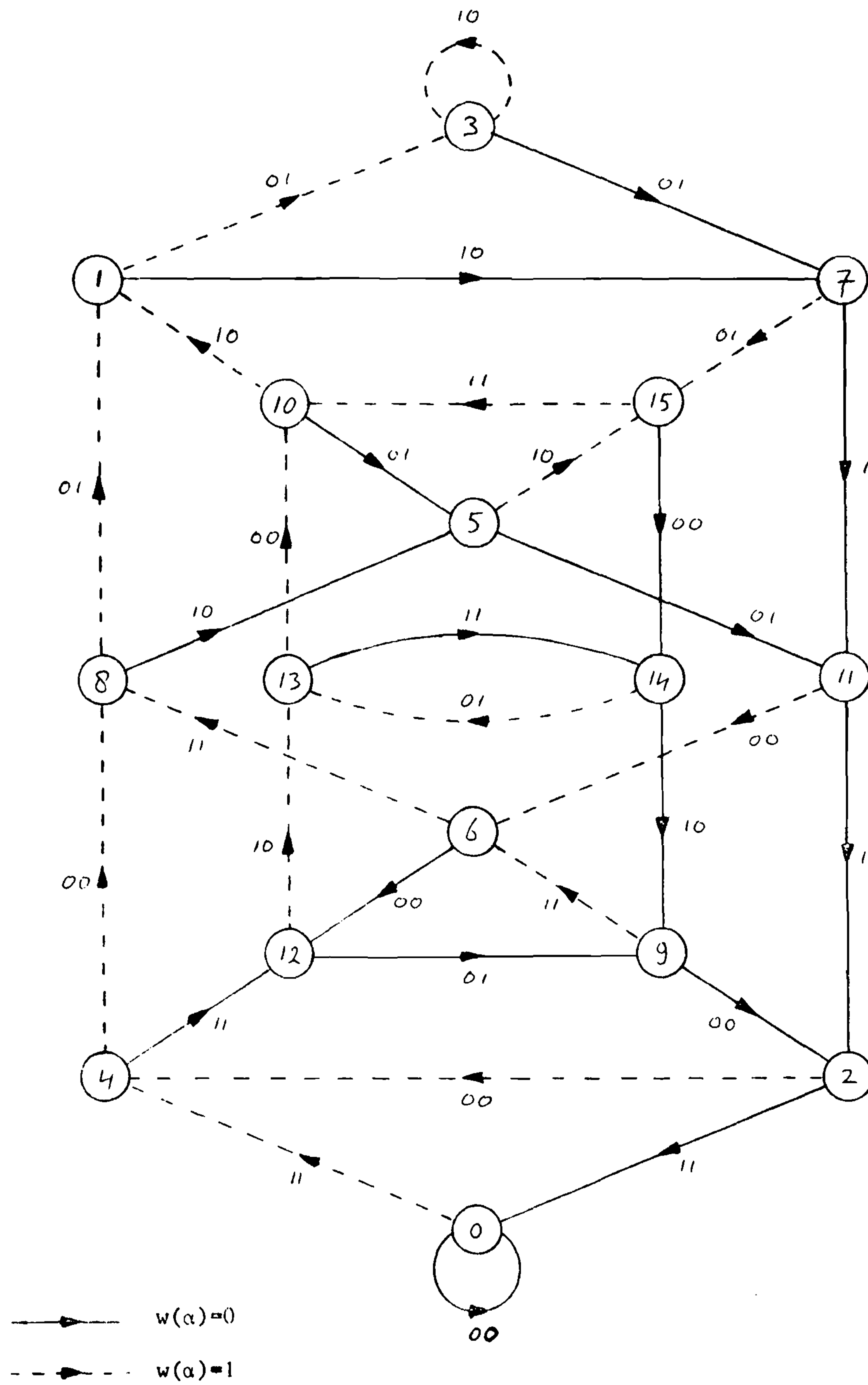


Fig. 9a. State-diagram z(α)=0.

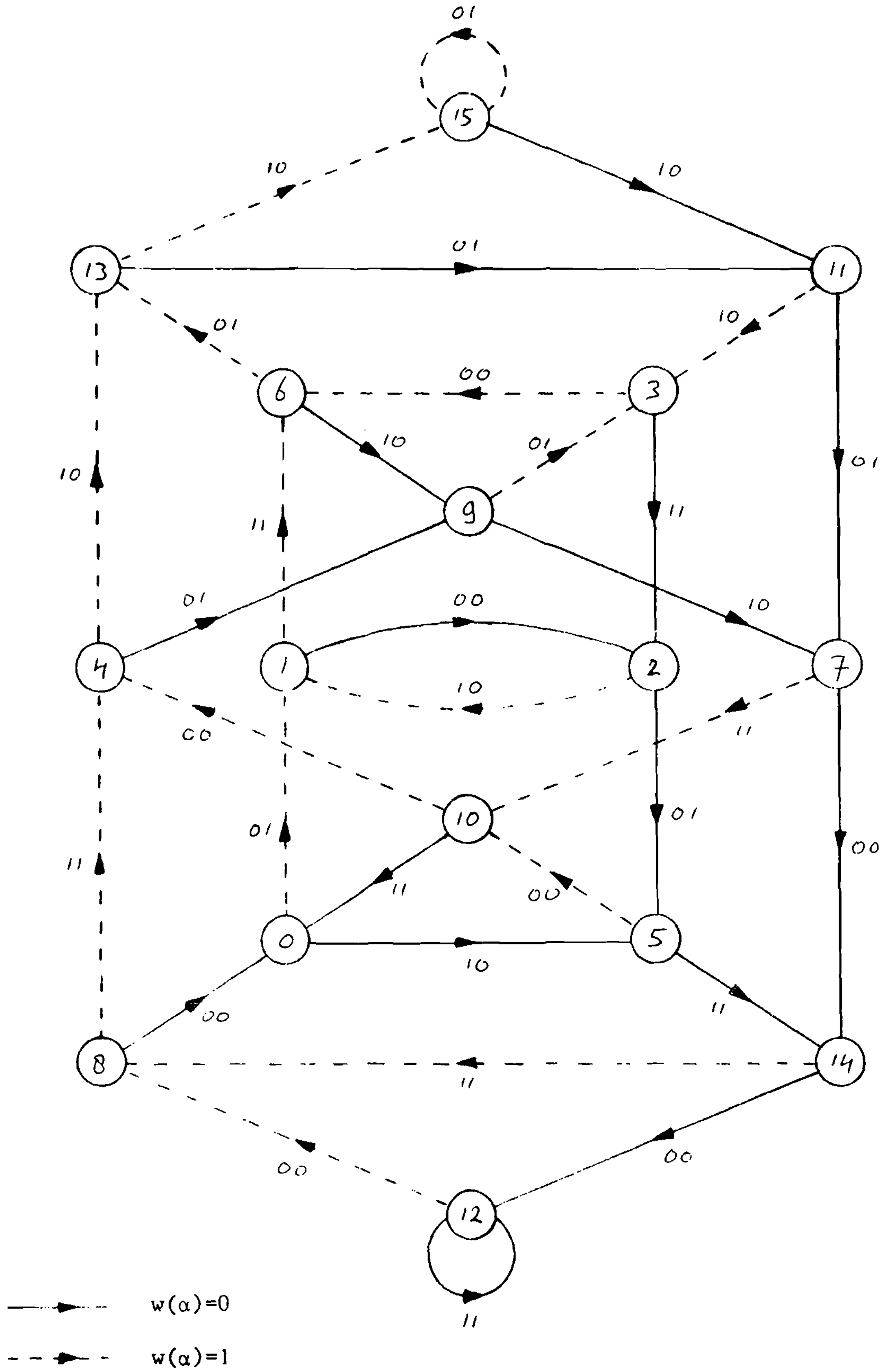


Fig. 9b. State-diagram z(α)=1.

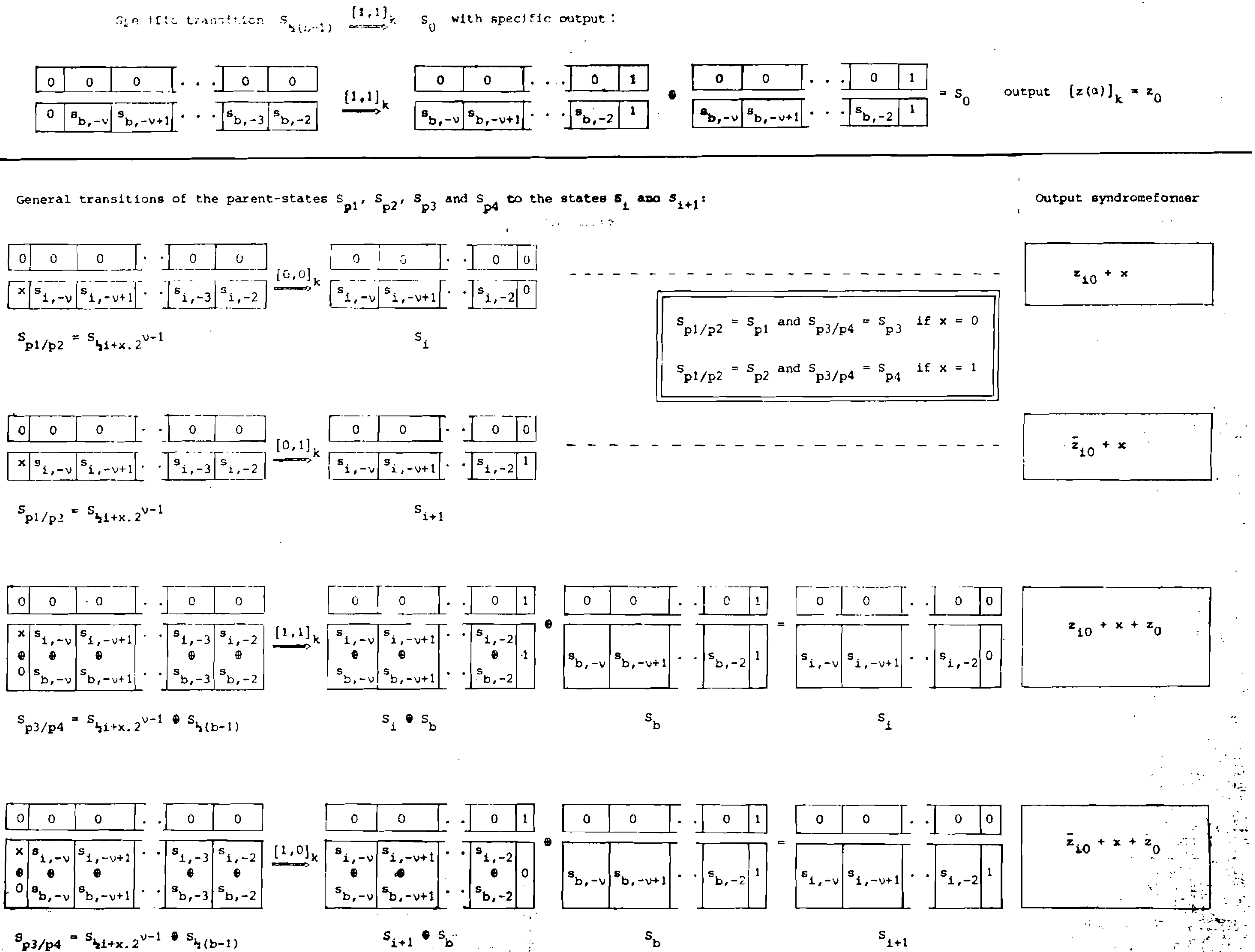


Fig. 10. Transition-table.

The outputvalue $[z(\alpha)]_k = z_{i0}$ corresponds to the basic transition $S_{\frac{1}{2}i}(k) \xrightarrow{[0,0]_k} S_i(k+1)$. With the knowledge of this value all outputvalues $[z(\alpha)]_k$ for the transitions [0,0] and [0,1] can be calculated.

For the transitions [1,1] and [1,0] we must add the base-state to get the new abstract state. In order to determine the syndromeoutputvalues for these transitions we need to know the specific outputvalue $[z(\alpha)]_k = z_0$ corresponding to the transition $S_{\frac{1}{2}(b-1)} \xrightarrow{[1,1]_k} S_0$.

Example i = 14 , b = 5 :

$S_{\frac{1}{2}(b-1)} = S_2 \xrightarrow{[1,1]} S_0$ with $z_0 = 0$.

			$z(\alpha)$	$w(\alpha)$
$S_{p1} = \quad = S_7 = [0000,0111]$	$\xrightarrow{[0,0]}$	$[0000,1110] = S_{14}$	1	0
$S_{p2} = S_{7+8} = S_{15} = [0000,1111]$	$\xrightarrow{[0,0]}$	$[0000,1110] = S_{14}$	0	0
$S_{p3} = S_7 \oplus S_2 = S_5 = [0000,0101]$	$\xrightarrow{[1,1]}$	$[0001,1011]+[0001,0101]=[0000,1110] = S_{14}$	1	0
$S_{p4} = S_{15} \oplus S_2 = S_{13} = [0000,1101]$	$\xrightarrow{[1,1]}$	$[0001,1011]+[0001,0101]=[0000,1110] = S_{14}$	0	0

			$z(\alpha)$	$w(\alpha)$
$S_{p1} = \quad = [0000,0111]$	$\xrightarrow{[0,1]}$	$[0000,1111] = S_{15}$	0	1
$S_{p2} = \quad = [0000,1111]$	$\xrightarrow{[0,1]}$	$[0000,1111] = S_{15}$	1	1
$S_{p3} = \quad = [0000,0101]$	$\xrightarrow{[1,0]}$	$[0001,1010]+[0001,0101]=[0000,1111] = S_{15}$	0	1
$S_{p4} = \quad = [0000,1101]$	$\xrightarrow{[1,0]}$	$[0001,1010]+[0001,0101]=[0000,1111] = S_{15}$	1	1

In this example the outputvalues of the w(α)-former are also given. We note that all four values of w(α) corresponding to the transitions of the four parent-states to a particular state S_i or S_{i+1} are the same. As is derived in the previous section the specific outputvalue $[w(\alpha)]_k = w_0$ corresponding to the transition $S_{\frac{1}{2}(b-1)} \xrightarrow{[1,1]_k} S_b = S_0$ is equal to zero. This is because the base-state in the w(α)-former causes an output w(α)=0 one unit of time earlier than the base-state in the syndromeformer.

Assuming that the outputvalue for the basic transition $S_{\frac{1}{2}i}(k) \xrightarrow{[0,0]_k} S_i(k+1)$ is $[w(\alpha)]_k = w_{i0}$, the pattern in the outputvalues corresponding to the other transitions can easily be determined.

If we replace the syndromeformer in the transition-table of Fig. 10 by the w(α)-former, we notice that there is no need to distinguish between the transitions [0,0] and [1,1] to the even states S_i, since degree $D_n(\alpha) <$ degree $C_n(\alpha)$ (hence the value of x does not count) and since the specific output $w_0=0$. The same applies to the transitions [0,1] and [1,0] to the odd states S_{i+1}.

In brief these results have been summarized in the table of Fig. 11.

Specific transition $S_{\frac{1}{2}(b-1)} \xrightarrow{[1,1]_k} S_0$ with output z_0.				output w(α)-former	
parent-state	input	new state	output syndromeformer	$d_{1,0}=0$	$d_{1,0}=1$
$S_{p1/p2}(k) = S_{\frac{1}{2}i+x.2^{\nu-1}}(k)$	$\xrightarrow{[0,0]_k}$	$S_i(k+1)$	$z_{i0} + x$	w_{i0}	w_{i0}
$S_{p3/p4}(k) = S_{\frac{1}{2}i+x.2^{\nu-1}}(k) \oplus S_{\frac{1}{2}(b-1)}$	$\xrightarrow{[1,1]_k}$	$S_i(k+1)$	$z_{i0} + x + z_0$	w_{i0}	w_{i0}
$S_{p1/p2}(k) = S_{\frac{1}{2}i+x.2^{\nu-1}}(k)$	$\xrightarrow{[0,1]_k}$	$S_{i+1}(k+1)$	$\bar{z}_{i0} + x$	$\bar{w}_{i0}$	w_{i0}
$S_{p3/p4}(k) = S_{\frac{1}{2}i+x.2^{\nu-1}}(k) \oplus S_{\frac{1}{2}(b-1)}$	$\xrightarrow{[1,0]_k}$	$S_{i+1}(k+1)$	$\bar{z}_{i0} + x + z_0$	$\bar{w}_{i0}$	w_{i0}

base state S_b

$i=2k$, $k=0,1,\ldots\ldots,2^{\nu-1}-1$.

$S_{p1/p2} = S_{p1}$ and $S_{p3/p4} = S_{p3}$ if $x = 0$.

$S_{p1/p2} = S_{p2}$ and $S_{p3/p4} = S_{p4}$ if $x = 1$.

Fig. 11. Transition-table.

From the transition-tables of Fig.8 and Fig.11 we can derive some general properties of the state-diagram:

- For fixed outputvalue $[z(\alpha)]_k$ the two outgoiing branches of a state have complementary inputpairs $[n_1(\alpha),n_2(\alpha)]_k$.
  The same applies to the two incoming branches.
- An even state can only be reached with the inputpairs $[n_1(\alpha),n_2(\alpha)]$ $=[0,0]$ and $[1,1]$.
- An odd state can only be reached with the inputpairs $[n_1(\alpha),n_2(\alpha)]$ $=[0,1]$ and $[1,0]$.
- For fixed outputvalue $[z(\alpha)]_k$ the two outgoiing branches have complementary outputvalues $[w(\alpha)]_k$ associated.
- The four branches that lead to a particular state all have the same outputvalue $[w(\alpha)]_k$ associated.

If we compare the diagrams for $z(\alpha)=0$ and $z(\alpha)=1$ we notice that they are each others complement as far as the transitions $[n_1(\alpha),n_2(\alpha)]$ along the branches is concerned. Only the numbering of the nodes is different.
It happens to be the case here, but this does not hold in general.

In the state-diagram two paths can be found which give us information about the free distance and the polynomials $D_n(\alpha)$, $n=1,2$ respectively.

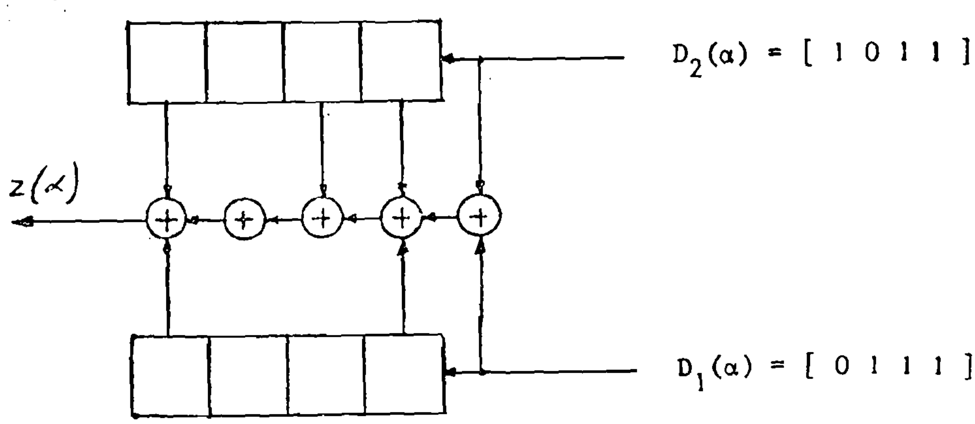
Free distance path: Since convolutional codes are group codes the set of distances (Hamming) of the zero-codeword to all other codewords is the same as the set of distances of any codeword to all others. The path with minimum distance to the so-called "zero-path" is called the free distance-path with distance d_{free}. The value of d_{free} can thus be evaluated by finding that path which is leaving the zero-state S_0 and leading back to it with minimum distance, irrespective the length of the path; that is, differing from the zero-path in as less places as possible. Hence the number of ones in that path equals d_{free}. For our example:

$$S_0 \xrightarrow{11} S_4 \xrightarrow{11} S_{12} \xrightarrow{01} S_9 \xrightarrow{00} S_2 \xrightarrow{11} S_0 \qquad d_{free} = 7$$

As will be explained in section 5 there are other codes of constraintlength $\nu=4$ which have $d_{free}=5$ or $d_{free}=6$. Apparently $d_{free}=7$ is the maximum of the minimum distances.

Path of the $D_n(\alpha)$-polynomials: The polynomials $D_1(\alpha)$ and $D_2(\alpha)$ can easily be derived from the state-diagram for $z(\alpha)=0$. The equation $C_1(\alpha)D_1(\alpha) + C_2(\alpha)D_2(\alpha) = 1$ must be satisfied. Now consider the polynomials $D_2(\alpha)$ and $D_1(\alpha)$ as an inputpair $[n_1(\alpha),n_2(\alpha)]$ to the syndrome-former and the $w(\alpha)$-former. The sequence $[n_1(\alpha),n_2(\alpha)]_k^{k+\ell}$ of length $\ell+1$ must be found which causes an output $[z(\alpha)]_k^{k+\ell}=[1,0,0,\ldots,0]_k^{k+\ell}$ and an output $[w(\alpha)]_k^{k+\ell}=[D_1(\alpha)D_2(\alpha) + D_2(\alpha)D_1(\alpha)]_k^{k+\ell}=[0,0,\ldots,0]_k^{k+\ell}$.

In terms of the state-diagram we must start in $S_0(k)$ and generate the sequence $[z(\alpha)]_k^{k+\ell}=[1,0,0,\ldots,0]_k^{k+\ell}$ in such a way that we return to $S_0(k+\ell)$ with outputs $[w(\alpha)]_k^{k+\ell}=[0,0,\ldots,0]_k^{k+\ell}$. The corresponding input-sequences $[n_1(\alpha)]_k^{k+\ell}$ and $[n_2(\alpha)]_k^{k+\ell}$ are then representative for the polynomials $D_2(\alpha)$ and $D_1(\alpha)$ respectively.



In the table below the path is indicated which leaves $S_0(k)$ with $[z(\alpha)]_k = 1$ and $[w(\alpha)]_k = 0$ and returns to $S_0(k+\ell)$ with $[z(\alpha)]_{k+1}^{k+\ell}=[w(\alpha)]_{k+1}^{k+\ell}=0$ in a total of $\ell+1$ steps.

time $k+\ell$ →	k	$k+1$	$k+2$	$k+3$
	(0) —10→ (5)	—01→ (11)	—11→ (2)	—11→ (0)
$[z(\alpha)]_{k+\ell}$	1	0	0	0
$[w(\alpha)]_{k+\ell}$	0	0	0	0

This path has a length $\ell+1 = 4$ and it follows:

$$[n_1(\alpha)]_k^{k+3} = [1\ 0\ 1\ 1] \leftrightarrow (1 \quad + \alpha^2 + \alpha^3) = D_2(\alpha)$$

$$[n_2(\alpha)]_k^{k+3} = [0\ 1\ 1\ 1] \leftrightarrow (\quad \alpha^1 + \alpha^2 + \alpha^3) = D_1(\alpha)$$

4. <u>METRICEQUATIONS and DECODING-ALGORITHM.</u>

As in the Viterbi algorithm, to find the path with minimum Hamming-weight, we associate with each possible state $S_i(k)$ a value $M_i(k)$, called "metric", which equals the minimum of the weights of the four paths $[n_1(\alpha),n_2(\alpha)]_{k-1}(p,i)$ leading from the parent states $S_p(k-1)$, $p\in\{p1,p2,p3,p4\}$, to $S_i(k)$. The metrics $M_i(k+1)$, $i=0,1,\dots,2^{\nu-1}$, can be determined recursively. From the transition-table of Fig.10 we know the four parent states $S_p(k)$, the associated inputs $[n_1(\alpha),n_2(\alpha)]_k(p,i)$ and the associated outputs $[z(\alpha)]_k(p,i)$ and $[w(\alpha)]_k(p,i)$. The outputvalues $[z(\alpha)]_k$ and the pattern in these outputs depend on the specific code.

In general the so-called "metricequations" have the following shape:

$$i = 2k\ ,\ k = 0,1,\dots,2^{\nu-1}-1.$$

$$M_i(k+1) = \bar{z}.\min[M_{px}, M_{py}+2] + z.\min[M_{px+2^{\nu-1}}, M_{py+2^{\nu-1}}+2]$$

$$M_{i+1}(k+1) = \bar{z}.\min[M_{px+2^{\nu-1}}+1, M_{py+2^{\nu-1}}+1] + z.\min[M_{px}+1, M_{py}+1]$$

$$z_0 = 0 \quad \begin{cases} (px,py) = (p1,p3) & \text{if } z_{i0} = 0 \\ (px,py) = (p2,p4) & \text{if } z_{i0} = 1 \end{cases}$$

$$z_0 = 1 \quad \begin{cases} (px,py) = (p1,p4) & \text{if } z_{i0} = 0 \\ (px,py) = (p2,p3) & \text{if } z_{i0} = 1 \end{cases}$$

$$p1 = p2 \pm 2^{\nu-1}$$

$$p3 = p4 \pm 2^{\nu-1}$$

$$S_{p1} = S_{\frac{1}{2}i}$$

$$S_{p2} = S_{\frac{1}{2}i} \oplus S_{\frac{1}{2}(b-1)}$$

With the knowledge of the index b of the code's base-state S_b, the specific outputvalue z_0 and the outputvalues z_{i0}, $i=0,1,\ldots,2^{\nu}-1$, all metrics can be calculated recursively with the equations above.

In practice we will not always calculate the metricequations with the help of these rather complex formula's. The metricequations can be directly derived from the state-table or the state-diagram. However in computer-calculations, where the use of an algorithm is more convenient, the general metricequations may be usefull.

In Fig. 12 the so-called metric-table of our example is shown. The excessive information in the general formulas above has been omitted.

metric $M_i(k+1)$	$[z(\alpha)]_k = 0$ minimumvalue of	$[z(\alpha)]_k = 1$ minimumvalue of
M_0	M_0 , M_2+2	M_8 $M_{10}+2$
M_1	M_8+1 , $M_{10}+1$	M_0+1 , M_2+1
M_2	M_9 , $M_{11}+2$	M_1 , M_3+2
M_3	M_1+1 , M_3+1	M_9+1 , $M_{11}+1$
M_4	M_2 , M_0+2	M_{10} , M_8+2
M_5	$M_{10}+1$, M_8+1	M_2+1 , M_0+1
M_6	M_{11} , M_9+2	M_3 , M_1+2
M_7	M_3+1 , M_1+1	$M_{11}+1$, M_9+1
M_8	M_4 , M_6+2	M_{12} , $M_{14}+2$
M_9	$M_{12}+1$, $M_{14}+1$	M_4+1 , M_6+1
M_{10}	M_{13} , $M_{15}+2$	M_5 , M_7+2
M_{11}	M_5+1 , M_7+1	$M_{13}+1$, $M_{15}+1$
M_{12}	M_6 , M_4+2	M_{14} , $M_{12}+2$
M_{13}	$M_{14}+1$, $M_{12}+1$	M_6+1 , M_4+1
M_{14}	M_{15} , $M_{13}+2$	M_7 , M_5+2
M_{15}	M_7+1 , M_5+1	$M_{15}+1$, $M_{13}+1$

$M_1 = M_5$
$M_3 = M_7$
$M_9 = M_{13}$
$M_{11} = M_{15}$
$M_3 = M_{11}$
$M_2 = M_{10}$
$M_6 = M_{14}$

Fig. 12. Metric-table.

Given the value of $[z(\alpha)]_k$, the new metric for each state can be determined with the help of the metric-table. To the metrics $M_p(k)$ of the two parent states $S_p(k)$, $p\in\{p1,p2,p3,p4\}$, of a particular state $S_i(k+1)$ the Hammingweight of their transitions $[\hat{n}_1(\alpha),\hat{n}_2(\alpha)]_k$ (p,i) is added. The minimum thus obtained is the new metric $M_i(k+1)$.

The transition associated with this minimumvalue is called the survivor and is specified by the index p of the parent state $S_p(k)$. In case of a tie, the survivor is choosen among the two candidates.

Goiing back from state $S_i(k)$, each time choosing the survivor we obtain the path $[n_1(\alpha),n_2(\alpha)]_{k-L}^{k-1}(p,i)$ of length L and of minimum weight, leading to that state $S_i(k)$. The outputvalues $[w(\alpha)]_{k-L}^{k-1}$ associated with each path $[n_1(\alpha),n_2(\alpha)]_{k-L}^{k-1}(p,i)$ are stored in the pathregister $PR[i]_{k-L}^{k-1}$ for state S_i. Each time k the new metrics $M_i(k+1)$, which arise by the given outputvalue $[z(\alpha)]_k$ of the syndromeformer, are stored in the metricregisters $MR[i]_k$.

In section 7 is explained how the new series of metrics $M_i(k+1)$, $i=0,1,\dots,2^{\nu-1}$, called "metriccombination", can be calculated from the metric-table for given outputvalues $[z(\alpha)]_k = 0 \vee 1$.

For any code in our class the organisation of the metriccombinations and the transitions between them, together with the indices p of the various survivors can be calculated. When we store these data in a read only memory (ROM) the use of metricregisters can be eliminated. It is this feature that makes syndrome-decoding so attractive.[4].

The pathregisterlength L is sometimes referred to as the coding delay. It will be clear that the bit-error probability decreases with increasing delay L.

5. CODES OF CONSTRAINTLENGTH $\nu=3$ AND $\nu=4$.

In order to obtain a better insight in the structure of convolutional codes we have analysed all non-catastrophic codes of constraintlength $\nu=3$ and $\nu=4$. In the tables of Fig. 13 and Fig. 14 all the relevant data of codes with $\nu=3$ and $\nu=4$ respectively, are summarized.

A complete survey of the characteristics of these codes is given in the "Databook" which is enclosed (see also section 10).

CODE	1	2	3
d_{free}	6	5	6
$C_1(\alpha)$	1011	1001	1011
$C_2(\alpha)$	1111	1101	1101
$D_1(\alpha)$	100	011	010
$D_2(\alpha)$	111	010	011
$S_{b=}$	3	3	7
z_{i0}	0101	0000	0101
w_{i0}	0110	0101	0101
py-px	5,3	1,7	7,5,3,1
pathregisters	6	6	6
s.s.m.c.	032244	032243	043332
total comb.	75	90	33

CODE	1-X	2-X	3-X
$C_1(\alpha)$	1111	1101	1101
$C_2(\alpha)$	1011	1001	1011
$D_1(\alpha)$	111	010	011
$D_2(\alpha)$	100	011	010
$S_{b=}$	7	7	3
z_{i0}	0110	0011	0011
w_{i0}	0011	0101	0101
py-px	3,1,7,5	7,5,3,1	1,7
s.s.m.c.	034422	033422	042333
total comb.	75	90	33

CODE	1-Y	2-Y
$C_1(\alpha)$	1101	1001
$C_2(\alpha)$	1111	1011
$D_1(\alpha)$	111	101
$D_2(\alpha)$	110	100
$S_{b=}$	5	5
z_{i0}	0011	0000
w_{i0}	0110	0011
py-px	6,2	2,6
s.s.m.c.	033424	033424
total comb.	74	99

CODE	1-X/Y	2-X/Y
$C_1(\alpha)$	1111	1011
$C_2(\alpha)$	1101	1001
$D_1(\alpha)$	110	100
$D_2(\alpha)$	111	101
$S_{b=}$	5	5
z_{i0}	0110	0101
w_{i0}	0110	0011
py-px	6,2	2,6
s.s.m.c.	033424	033424
total comb.	74	99

Fig. 13.

Codes of constraint-length $\nu=3$.

$C_1(\alpha)$	10011	10001	11011	10101	10111	10001	10011	10101	10011	[illegible]	10001	10011	10101
$C_2(\alpha)$	10111	10101	11111	11101	11111	11001	11011	11001	11111	[illegible]	11111	11001	11111
$D_1(\alpha)$	1110	0101	1000	1101	0100	0011	1000	0010	1101	[illegible]	1011	1011	0011
$D_2(\alpha)$	1101	0100	1011	1000	0111	0010	1111	0011	1010	[illegible]	1110	1100	0010
$S_{b=}$	5	5	13	11	11	3	3	15	7	15	15	11	3
z_{io}	0101 0101	0000 0000	0101 1010	0011 0011	0110 0110	0000 0000	0101 0101	0011 0011	0101 0101	[illegible]	0000 0000	0101 0101	0011 0011
w_{io}	0011 1100	0011 0011	0101 1010	0000 1111	0110 0110	0101 0101	0110 1001	0101 0101	0101 1010	[illegible]	0110 1001	0011 1100	0101 0101
$py-px$	2,14	2,14	14,10,6,2	5,3,13,11	13,11,5,3	1,15	9,7	15,13,11,9,7,5,3,1	11,9,7,5	[illegible]	7,5,3,1,15,13,11,9	13,11,5,3	1,15
pathregisters	9	9	9	12	12	12	12	12	12	[illegible]	12	12	12
s.s.m.c.	035424255444	033625253645	035445442524	033524342244	034425442244	032253244365	032244254455	044455535522	043532334423	[illegible]	065354345432	045553445243	042335353444
total comb.	2001	144	2805	1642	1969	1655	1784	270	195		>16000	659	459

CODE	1-X	2-X	3-X	4-X	5-X	6-X	7-X	8-X	9-X	[illegible]	11-X	12-X	13-X
$C_1(\alpha)$	10111	10101	11111	11101	11111	11001	11011	11001	11111	[illegible]	11111	11001	11111
$C_2(\alpha)$	10011	10001	11011	10101	10111	10001	10011	10101	10011	[illegible]	10001	10011	10101
$D_1(\alpha)$	1101	0100	1011	1000	0111	0010	1111	0011	1010	[illegible]	1110	1100	0010
$D_2(\alpha)$	1110	0101	1000	1101	0100	0011	0001	0010	1101	[illegible]	1011	1011	0011
$S_{b=}$	5	5	13	7	7	15	15	3	11	[illegible]	3	7	15
z_{io}	0110 0110	0011 0011	0110 1001	0011 1100	0110 1001	0000 1111	0101 1010	0000 1111	0110 1001	[illegible]	0110 1001	0000 1111	0110 1001
w_{io}	0110 1001	0011 0011	0000 1111	0011 1100	0011 0011	0101 0101	0000 1111	0101 0101	0011 1100	[illegible]	0101 1010	0101 1010	0101 0101
$py-px$	10,6	10,6	6,2,14,10	11,9,7,5	3,1,15,13	15,13,11,9,7,5,3,1	7,5,3,1,15,13,11,9	1,15	5,3,13,11	[illegible]	9,7	3,1,15,13	15,13,11,9,7,5,3,1
s.s.m.c.	035424255444	033625253645	035445442524	033442254423	034442244524	033536442522	034445452522	042244535555	043324353234	[illegible]	062335543544	044442553355	045344353432
total comb.	2001	144	2805	1642	1969	1655	1784	270	195		>16000	659	459

CODE	1-Y			4-Y	5-Y	6-Y	7-Y	8-Y	9-Y	[illegible]
$C_1(\alpha)$	11001			10101	11101	10001	11001	10101	11001	[illegible]
$C_2(\alpha)$	11101			10111	11111	10011	11011	10011	11111	[illegible]
$D_1(\alpha)$	0111			1011	1111	1001	1101	0111	0101	[illegible]
$D_2(\alpha)$	0110			1010	1110	1000	1100	0110	0100	[illegible]
$S_{b=}$	13			9	9	9	9	13	5	[illegible]
z_{io}	0000 1111			0011 0011	0011 1100	0000 0000	0000 1111	0011 0011	0000 1111	[illegible]
w_{io}	0110 0110			0101 1010	0110 1001	0000 1111	0011 1100	0110 0110	0011 0011	[illegible]
$py-px$	14,10,6,2			4,12	12,4	4,12	12,4	14,10,6,2	2,14	[illegible]
s.s.m.c.	033534552325			033444452455	033444452445	033436452545	033435452555	044433553324	044424533555	[illegible]
total comb.	1817			1959	2125	2582	1686	253	236	

CODE	1-X/Y			4-X/Y	5-X/Y	6-X/Y	7-X/Y	8-X/Y	9-X/Y	[illegible]
$C_1(\alpha)$	11101			10111	11111	10011	11011	10011	11111	[illegible]
$C_2(\alpha)$	11001			10101	11101	10001	11001	10101	11001	[illegible]
$D_1(\alpha)$	0110			1010	1110	1000	1100	0110	0100	[illegible]
$D_2(\alpha)$	0111			1011	1111	1001	1101	0111	0101	[illegible]
$S_{b=}$	13			9	9	9	9	13	5	7
z_{io}	0011 1100			0110 0110	0110 1001	0101 0101	0101 1010	0101 0101	0110 1001	[illegible]
w_{io}	0110 0110			0101 1010	0110 1001	0000 1111	0011 1100	0110 0110	0011 0011	[illegible]
$py-px$	6,2,14,10			4,12	12,4	4,12	12,4	6,2,14,10	10,6	[illegible]
s.s.m.c.	033534552325			033444452455	033444452445	033436452545	033435452555	043533443425	043525534454	[illegible]
total comb.	1817			1959	2125	2582	1686	253	236	

Fig. 14. Codes of constraintlength $\nu=4$.

Let us consider some characteristics of both tables.

In the connectionpolynomials $C_n(\alpha)$, $n=1,2$, we can interchange the roles of $n=1$ and $n=2$. We can also reflect both polynomials by first taking $\alpha=\alpha^{-1}$ and then multiplicating the new polynomials $C_n(\alpha^{-1})$ by the factor $\alpha^{-\nu}$. Hence each code has four different connectionpatterns as given in Fig. 15. The axes of symmetry are the X-axis and the Y-axis.

code C $\begin{cases} C_1(\alpha) \\ C_2(\alpha) \end{cases}$

code C-X $\begin{cases} C_2(\alpha) \\ C_1(\alpha) \end{cases}$

code C-Y $\begin{cases} C_1(\alpha^{-1})\alpha^{\nu} \\ C_2(\alpha^{-1})\alpha^{\nu} \end{cases}$

code C-X/Y $\begin{cases} C_2(\alpha^{-1})\alpha^{\nu} \\ C_1(\alpha^{-1})\alpha^{\nu} \end{cases}$

Y

$c_{1,\nu}\alpha^{\nu}+c_{1,(\nu-1)}{}^{(-1)}+\dots \mid \dots+c_{1,1}\alpha^{1}+c_{1,0}\alpha^{0}.$ X

$c_{2,\nu}\alpha^{\nu}+c_{2,(\nu-1)}\alpha^{(\nu-1)}+\dots \mid \dots+c_{2,1}\alpha^{1}+c_{2,0}\alpha^{0}.$

Fig. 15. Axes of symmetry.

It is obvious that with a reflection of the polynomials $C_n(\alpha)$, $n=1,2$, in the X-axis corresponds an identical reflection of the polynomials $D_n(\alpha)$, $n=1,2$.

The base states of the four members of code C need not to be the same as can be verified for code 10.

The series of values z_{i0} and w_{i0} which correspond to the transitions of the states $S_i(k)$, $i=0,1,\dots,2^{(\nu-1)}-1$, for an input $[0,0]_k$ are given in the rows represented by "z_{io}" and "w_{i0}" respectively.

The different values of the expression (py-px) are representative for the organisation in the metricequations (section 4).

The number of distinct metricequations is given in the row "pathregisters". In section 6 a formula is given for the particular relationship between the number of distinct metricequations and the connectionpattern of the polynomials $C_n(\alpha)$, $n=1,2$.

In section 7 will be dealt with the determination of the so-called "steady state metriccombination" (s.s.m.c.), and the total number of possible metriccombinations C+1.

Comparison and some provisional conclusions:

The codes C and C-X are equivalent in structure; only the roles of $n_1(\alpha)$ and $n_2(\alpha)$ are interchanged. A reflection in the Y-axis results in a fundamental difference in structure. Hence for constraintlength $\nu=3$ there are 5 distinct codes and for constraintlength $\nu=4$ there are 21 distinct codes.

A. Codes of constraintlength $\nu=3$ ($d_{free}=5,6$).

At first we remark that only codes of even constraintlength are of practical interest since codes with even free distance do not have error-correcting properties (note that the unique code of constraintlength $\nu=2$ has $d_{free}=5$).

The form of the various characteristics of codes of constraintlength $\nu=3$ is analogous to the form of codes of constraintlength $\nu=4$; the only difference is that they are smaller in dimension.

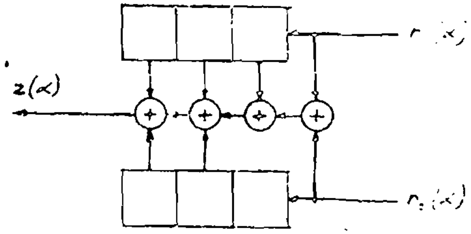
In this case we notice that the two state-diagrams for $z(\alpha)=0$ and $z(\alpha)=1$ are not each others complement as is the case for code 1 of constraintlength $\nu=4$.

The consequence of (py-px) being even is that each state can only be reached either from four odd parent states or either from four even parent states, as can be verified for code 1-Y. If (py-px) is odd then each state can be reached from two odd parent states and two even parent states.

In the table given below the above mentioned properties are summarized.

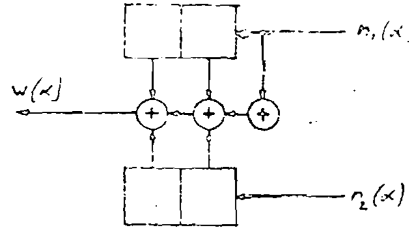
base state S_b	codes	(py-px)	metrics
b = 5	1-Y, 1-X/Y / 2-Y, 2-X/Y	6,2 / 2,6	1=5 , 3=7
b = 3	1 / 2,3-X	5,3 / 1,7	1=3 , 5=7
b = 7	2-X,3 / 1-X	7,5,3,1 / 3,1,7,5	1=7 , 3=5

Fig. 17. Characteristics of codes $\nu=3$.



Syndromeformer

$$C_1(\alpha) = \alpha^3 + \alpha^2 + \alpha^0$$
$$C_2(\alpha) = \alpha^3 + \alpha^2 + \alpha^1 + \alpha^0$$

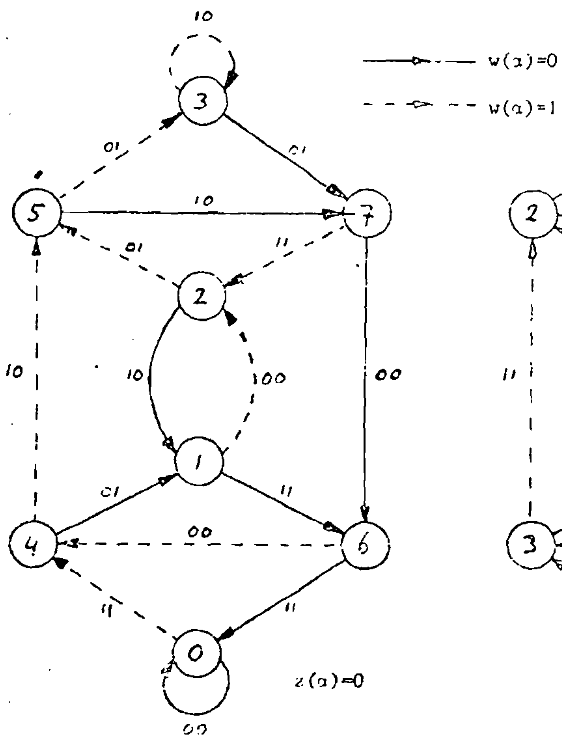


w(α)-former

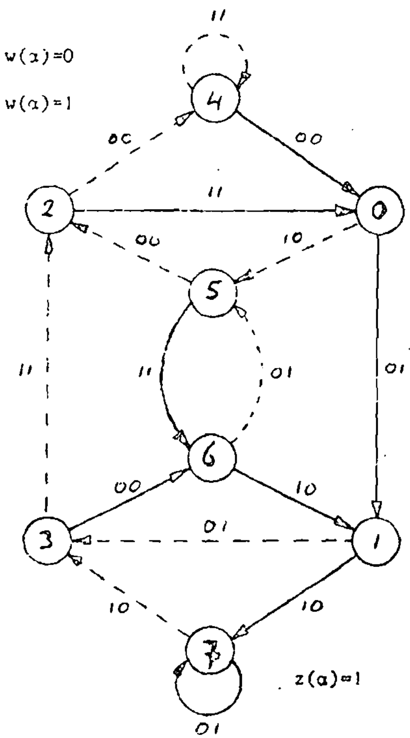
$$D_1(\alpha) = \alpha^2 + \alpha^1 + \alpha^0$$
$$D_2(\alpha) = \alpha^2 + \alpha^1$$

state $S_j(k)$	state contents	new state $S_i(k+1)$ $[n_1(\alpha),n_2(\alpha)]_k$				value $[z(\alpha)]_k$ $[n_1(\alpha),n_2(\alpha)]_k$				value $[w(\alpha)]_k$ $[n_1(\alpha),n_2(\alpha)]_k$			
j	$s_j(\alpha)$	00	01	11	10	00	01	11	10	00	01	11	10
0	000	0	1	4	(5)	0	1	0	1	0	0	1	1
1	001	2	3	6	7	0	1	0	1	1	1	0	0
2	010	4	5	0	1	1	0	1	0	1	1	0	0
3	011	6	7	2	3	1	0	1	0	0	0	1	1
4	100	0	1	4	5	1	0	1	0	0	0	1	1
(5)	101	2	3	6	7	1	0	1	0	1	1	0	0
6	110	4	5	0	1	0	1	0	1	1	1	0	0
7	111	6	7	2	3	0	1	0	1	0	0	1	1

State-table



state-diagram



state-diagram

metric $M_i(k+1)$	$[z(\alpha)]_k = 0$ minimumvalue of	$[z(\alpha)]_k = 1$ minimumvalue of
M_0	M_0 , $M_6 + 2$	M_4 , $M_2 + 2$
M_1	$M_4 + 1$, $M_2 + 1$	$M_0 + 1$, $M_6 + 1$
M_2	M_1 , $M_7 + 2$	M_5 , $M_3 + 2$
M_3	$M_5 + 1$, $M_3 + 1$	$M_1 + 1$, $M_7 + 1$
M_4	M_6 , $M_0 + 2$	M_2 , $M_4 + 2$
M_5	$M_2 + 1$, $M_4 + 1$	$M_6 + 1$, $M_0 + 1$
M_6	M_7 , $M_1 + 2$	M_3 , $M_5 + 2$
M_7	$M_3 + 1$, $M_5 + 1$	$M_7 + 1$, $M_1 + 1$

$M_1 = M_5$
$M_3 = M_7$

Metric-table

$d_{free} = 6$

$\underline{m}_0(k) = \{0\ 3\ 3\ 4\ 2\ 3\ 4\ 4\}$

free distance path:

(0) —11— (4) —10— (5) —10— (7) —00— (6) —11— (0)

$D_n(\alpha)$-polynomial path:

(0) —01— (1) —11— (5) —11— (0)

B. Codes of constraintlength $\nu=4$ (d_{free}= 5,6,7).

The distribution of the 21 codes concerning their free distances is as follows:

d_{free}	codes
5	2,6
6	3,4,5,8,9,11,12,13
7	1,7,10

Fig. 18. Free distance.

From a practical point of view the codes 1,7 and 10 with free distance d_{free}=7 are the most important. Up till now no special relationship between the structure of the polynomials $C_n(\alpha)$ and the code's free distance has been found.

The characteristics of codes with $\nu=4$ are given in the table of Fig. 19.

base state S_b	codes	(py-px)	metrics
b = 5	1,2,9Y / 1X,2X,9XY	2,14/10,6	1 = 5,3 = 7,9 =13,11=15
b = 9	4Y,4XY,6Y,6XY / 5Y,5XY,7Y,7XY	4,12/12,4	1 = 9,3 =11,5 =13,7 =15
b = 13	1Y,8Y,3 / 1XY,8XY,3X	14,10,6,2/6,2,14,10	1 =13,3 =15,5 = 9,7 =11
b = 3	7,11X / 6,13,8X,10X	9,7/1,15	1 = 3,5 = 7,9 =11,13=15
b = 7	5X,12X,10XY / 4X,9	3,15,13/11,9,7,5	1 = 7,3 = 5,9 =15,11=13
b = 11	4,9X / 5,10Y,12	5,3,13,11/13,11,5,3	1 =11,3 = 9,5 =15,7 =13
b = 15	7X,11 / 8,10,6X,13X	7,5,3,1,15,13,11,9/15,13,11,9,7,5,3,1	1 =15,3 =13,5 =11,7 = 9

Fig. 19. Characteristics of codes with $\nu=4$.

In the metricequations there are always $2^{\nu-2}$ pairs of equal metricequations.(see also section 6) It is striking that codes with the same base state always have the same pairs of equal metricequations.

A dicussion on the remaining characteristics is made in the sections 7 and 8.

6. REDUCTION IN PATHREGISTERS.

As we take a closer look at the metric-table of code 1 in Fig.12 we notice that $M_1=M_5$, $M_3=M_7$; $M_9=M_{13}$ and $M_{11}=M_{15}$. This is because there is no need to distinguish between the transitions [0,1] and [1,0], each having Hammingweight=1.

It is obvious from the statetransition-table of Fig.11 that there are two states S_α and S_β which have the same parent states S_{p1}, S_{p2}, S_{p3} and S_{p4}. If S_α and S_β are odd states we have the following transitions:

$S_{p1}=S_{\frac{1}{2}i}$	$\xrightarrow{[0,1]}$	$S_{i+1} = S_\alpha$					$\bar{z}_{i0}$
$S_{p2}=S_{\frac{1}{2}i+2^{\nu-1}}$	$\xrightarrow{[0,1]}$	$S_{i+1} = S_\alpha$					z_{i0}
$S_{p3}=S_{\frac{1}{2}i} \oplus S_{\frac{1}{2}(b-1)}$	$\xrightarrow{[1,0]}$	$S_{i+1} = S_\alpha$					$\bar{z}_{i0} + z_0$
$S_{p4}=S_{\frac{1}{2}i+2^{\nu-1}} \oplus S_{\frac{1}{2}(b-1)}$	$\xrightarrow{[1,0]}$	$S_{i+1} = S_\alpha$					$z_{i0} + z_0$
$_{\frac{1}{2}i}$	$\xrightarrow{[0,0]}$	S_i	z_{i0}	$S_{p1}=S_{\frac{1}{2}i} \oplus S_0$	$\xrightarrow{[1,0]}$	$S_i \oplus S_b = S_\beta$	$\bar{z}_{i0}$
$_0$	$\xrightarrow{[1,0]}$	S_b	$z = 1$				
$_{\frac{1}{2}i+2^{\nu-1}}$	$\xrightarrow{[0,0]}$	S_i	$\bar{z}_{i0}$	$S_{p2}=S_{\frac{1}{2}i+2^{\nu-1}} \oplus S_0$	$\xrightarrow{[1,0]}$	$S_i \oplus S_b = S_\beta$	z_{i0}
$_0$	$\xrightarrow{[1,0]}$	S_b	$z = 1$				
$_{\frac{1}{2}i}$	$\xrightarrow{[0,1]}$	S_{i+1}	$\bar{z}_{i0}$	$S_{p3}=S_{\frac{1}{2}i} \oplus S_{\frac{1}{2}(b-1)}$	$\xrightarrow{[0,1]}$	$S_i \oplus S_b = S_\beta$	$\bar{z}_{i0} + z_0$
$_{\frac{1}{2}(b-1)}$	$\xrightarrow{[0,0]}$	S_{b-1}	z_0				
$_{\frac{1}{2}i+2^{\nu-1}}$	$\xrightarrow{[0,1]}$	S_{i+1}	z_{i0}	$S_{p4}=S_{\frac{1}{2}i+2^{\nu-1}} \oplus S_{\frac{1}{2}(b-1)}$	$\xrightarrow{[0,1]}$	$S_i \oplus S_b = S_\beta$	$z_{i0} + z_0$
$_{\frac{1}{2}(b-1)}$	$\xrightarrow{[0,0]}$	S_{b-1}	z_0				

$S_\alpha = S_\beta \oplus S_{b-1}$ and S_α and S_β have identical metricequations:

if $z_0=0$ $\quad M_\alpha = M_\beta = \bar{z}.\min[M_{p1} + 1, M_{p3} + 1] + z.\min[M_{p2} + 1, M_{p4} + 1]$

if $z_0=1$ $\quad M_\alpha = M_\beta = \bar{z}.\min[M_{p1} + 1, M_{p4} + 1] + z.\min[M_{p2} + 1, M_{p3} + 1]$

metric $M_i(k+1)$	$[z(\alpha)]_k = 0$ minimumvalue of	$[z(\alpha)]_k = 1$ minimumvalue of
M_0	M_0 , $M_2 + 2$	M_8 $M_{10} + 2$
M_1	$M_8 + 1$, $M_{10} + 1$	$M_0 + 1$, $M_2 + 1$
M_2	M_9 , $M_{11} + 2$	M_1 , $M_3 + 2$
M_3	$M_1 + 1$, $M_3 + 1$	$M_9 + 1$, $M_{11} + 1$
M_4	M_2 , $M_0 + 2$	M_{10} , $M_8 + 2$
M_5	$M_{10} + 1$, $M_8 + 1$	$M_2 + 1$, $M_0 + 1$
M_6	M_{11} , $M_9 + 2$	M_3 , $M_1 + 2$
M_7	$M_3 + 1$, $M_1 + 1$	$M_{11} + 1$, $M_9 + 1$
M_8	M_4 , $M_6 + 2$	M_{12} , $M_{14} + 2$
M_9	$M_{12} + 1$, $M_{14} + 1$	$M_4 + 1$. $M_6 + 1$
M_{10}	M_{13} . $M_{15} + 2$	M_5 , $M_7 + 2$
M_{11}	$M_5 + 1$, $M_7 + 1$	$M_{13} + 1$, $M_{15} + 1$
M_{12}	M_6 , $M_4 + 2$	M_{14} , $M_{12} + 2$
M_{13}	$M_{14} + 1$, $M_{12} + 1$	$M_6 + 1$, $M_4 + 1$
M_{14}	M_{15} . $M_{13} + 2$	M_7 , $M_5 + 2$
M_{15}	$M_7 + 1$, $M_5 + 1$	$M_{15} + 1$, $M_{13} + 1$

$M_1 = M_5$
$M_3 = M_7$
$M_9 = M_{13}$
$M_{11} = M_{15}$
$M_3 = M_{11}$
$M_2 = M_{10}$
$M_6 = M_{14}$

Fig. 20. Metric-table.

M_0	M_0 , $M_2 + 2$	M_8 , $M_{10} + 2$
M_1	$M_8 + 1$, $M_{10} + 1$	$M_0 + 1$, $M_2 + 1$
M_2	M_9 , $M_{11} + 2$	M_1 , $M_3 + 2$
M_3	$M_1 + 1$, $M_3 + 1$	$M_9 + 1$, $M_{11} + 1$
M_4	M_2 , $M_0 + 2$	M_{10} , $M_8 + 2$
M_6	M_{11} , $M_9 + 2$	M_3 , $M_1 + 2$
M_8	M_4 , $M_6 + 2$	M_{12} , $M_{14} + 2$
M_9	$M_{12} + 1$, $M_{14} + 1$	$M_4 + 1$, $M_6 + 1$
M_{10}	M_9 , $M_{11} + 2$	M_1 , $M_3 + 2$
M_{11}	$M_1 + 1$, $M_3 + 1$	$M_9 + 1$, $M_{11} + 1$
M_{12}	M_6 , $M_4 + 2$	M_{14} , $M_{12} + 2$
M_{14}	M_{11} , $M_9 + 2$	M_3 , $M_1 + 2$

M_0	M_0 , $M_2 + 2$	M_8 , $M_2 + 2$
M_1	$M_8 + 1$, $M_2 + 1$	$M_0 + 1$, $M_2 + 1$
M_2	M_9 , $M_3 + 2$	M_1 , $M_3 + 2$
M_3	$M_1 + 1$, $M_3 + 1$	$M_9 + 1$, $M_3 + 1$
M_4	M_2 , $M_0 + 2$	M_2 , $M_8 + 2$
M_6	M_3 , $M_9 + 2$	M_3 , $M_1 + 2$
M_8	M_4 , $M_6 + 2$	M_{12} , $M_6 + 2$
M_9	$M_{12} + 1$, $M_6 + i$	$M_4 + 1$, $M_6 + 1$
M_{12}	M_6 , $M_4 + 2$	M_6 , $M_{12} + 2$

Fig. 21. Reduced metric-table.

If we select the identical survivor in case of a tie, the states S_α and S_β will have the same pathregistercontents. As far as metric- and pathregistercontents are concerned the states S_α and S_β are not distinct. In general for codes in the class $\mathcal{T}_{\nu,1}$ one can eliminate the metric- and pathregisters of half the odd states (i.e. $2^{\nu-2}$).

In the metric-table of Fig.20 we have eliminated the metrics M_5, M_7, M_{13} and M_{15}. The numbers 5,7,13 and 15 are replaced by the numbers 1,3,9 and 11. Note that now $M_3=M_{11}$, $M_2=M_{10}$ and $M_6=M_{14}$. We eliminate M_{11}, M_{10} and M_{14} and replace the numbers 11,10 and 14 by 3,2 and 6. Then we get the reduced metric-table of Fig.21 with 9 distinct metricequations and hence 9 distinct pathregisters.

In general there is a certain relationship between the connectionpattern of the polynomials $C_n(\alpha)$, n=1,2 , and the number of distinct metric-equations [3]. Consider the class $\mathcal{T}_{\nu,\ell}$ of $R=\frac{1}{2}$ convolutional codes of constraintlength ν with

$$\left.\begin{array}{l} c_{1,k} = c_{2,k} \\ c_{1,\nu-k} = c_{2,\nu-k} \end{array}\right\} \quad 0 \leqslant k \leqslant \ell-1 \ , \ 1 \leqslant \ell \text{ is a positive integer} \leqslant \nu/2$$

Then $\mathcal{T}_{\nu,1} \supset \mathcal{T}_{\nu,2} \supset \ldots\ldots \supset \mathcal{T}_{\nu,\nu/2}$ ($\mathcal{T}_{\nu,1}$ is our class of codes) and the total number of redundant states (i.e. metricequations) is equal to:

$$\sum_{j=0}^{\ell} \binom{\ell}{j} (2^{\nu-\ell}/2^j)(2^j-1) = 2^{\nu-2\ell}(4^\ell - 3^\ell) = 2^\nu\{1-(\tfrac{3}{4})^\ell\} \ , \quad 1 \leqslant \ell \leqslant \nu/2$$

In [3] is also stated that the complexity of the decoder for codes with even constraintlength $\nu=2L$, L=1,2,......., with $c_{1,-k} = c_{2,-k}$, $0 \leqslant k \leqslant 2L$, $k \neq L$, is:

$$2^\nu\{1-\tfrac{1}{4}[1+(\tfrac{3}{4}) + (\tfrac{3}{4})^2 + \ldots\ldots\ldots (\tfrac{3}{4})^{\ell-1}]\} = (\sqrt{3})^\nu .$$

This implies that an exponential saving in hardware can be obtained. It has been investigated that this class of codes has distance properties no worse than those of the systematic $R=\frac{1}{2}$ codes.

For codes with $\nu=2$, $\nu=4$ and $\nu=6$ we have the following possibilities:

$\nu=2$	$\ell=1$	$2^0(4-3)=1$	redundant	$2^2-1=3$	pathregisters
$\nu=4$	$\ell=1$	$2^2(4-3)=4$	"	$2^4-4=12$	"
	$\ell=2$	$2^0(4^2-3^2)=7$	"	$2^4-7=9$	"
$\nu=6$	$\ell=1$	$2^4(4-3)=16$	"	$2^6-16=48$	"
	$\ell=2$	$2^2(4^2-3^2)=28$	"	$2^6-28=36$	"
	$\ell=3$	$2^0(4^3-3^3)=37$	"	$2^6-37=27$	"

For $\nu=4$ the codes 1,2 and 3 have the property $c_{1,k} = c_{2,k}$, $0 \leqslant k \leqslant 4$ and $k \neq 2$, and hence 9 distinct pathregisters.

Note that the condition $s_{b,-2}=0$ is consistent for these three codes. Code 1 has the maximum free distance $d_{free} = 7$ and is thus the most interesting one.

As will be discussed in section 9, it is not always possible for even codes in the class $\mathcal{T}_{\nu=2L,1}$ to have $c_{1,k} = c_{2,k}$, $k \neq L$ and at the same time the maximum free distance.

7. THE STEADY STATE METRICCOMBINATION AND THE TOTAL NUMBER OF METRICCOMBINATIONS.

In the decodingalgorithm, each time k one has to calculate all metrics $M_i(k)$. These values arranged in a row are called the metriccombination $\underline{m}_c(k)$, $c\in\{0,1,\ldots,C\}$, at time k.

$$\underline{m}_c(k)=\{M_{0,c},M_{1,c},\ldots\ldots,M_{i,c},\ldots,M_{2^\nu-1,c}\}(k)$$

The total number of metriccombinations is C+1. For the unique code with $\nu=2$ there are C+1=12 metriccombinations [4]. All these metriccombinations can be evaluated by starting with the so-called "steady state metriccombination" $\underline{m}_0(k)$. The steady state metriccombination s.s.m.c. is that metriccombination which converts into itself for a syndromeoutputvalue $[z(a)]_k = 0$.

On the basis of our example (code 1) we will make clear how this s.s.m.c. $\underline{m}_0(k)$ can be determined.

Suppose that up to time k the zero codeword $[x(\alpha)]_{-\infty}^{k-1}=[0,0,\ldots,0]_{-\infty}^{k-1}$ has been sent and that the channel has not been disturbed by noise. It follows that the registers of the syndromeformer and the $w(\alpha)$-former contain only zero's; they both are in the zero state S_0. Now from time k we will not send the zero codeword any more, but we still assume that the channel is not disturbed by noise; hence the syndrome-output is $[z(\alpha)]_k^\infty=[000\ldots00]_k^\infty$. From the state-diagram for $z(\alpha)=0$ we deduce that only S_0 and S_4 can be reached from S_0. Suppose that $M_0(k)=0$, then with the metric-table of Fig.12 the new metrics of the states S_0 and S_4 at time k+1 can be calculated:

$$M_0(k+1)=\min[M_0\ ,M_2+2]=0.$$

$$M_4(k+1)=\min[M_2\ ,M_0+2]=2.$$

From the states S_0 and S_4 the new states S_0,S_4 and S_8,S_{12} can be reached and their metrics $M_0(k+2)$, $M_4(k+2)$, $M_8(k+2)$ and $M_{12}(k+2)$ can be calculated. In the reduced Trellis-diagram of Fig.22 the transitions from S_0 to all other states up to time k+5, are drawn. Next to each state $S_i(k)$ each time k, the value of $M_i(k)$ is indicated.

At the fourth step the metriccombination is $\underline{m}(k+4)=\{0354237425544574\}$. The state $S_6(k+4)$ has a metric $M_6(k+4)=7$. If we look one unit of time further we notice that $M_6(k+5)=4<M_6(k+4)=7$. In order to determine the s.s.m.c. $\underline{m}_0$ we must penetrate the Trellisdiagram so far until all metrics M_i are minimal. In this case a depth=5. It has been calculated that for code 7 this depth equals even 7 (Fig. 22a.).

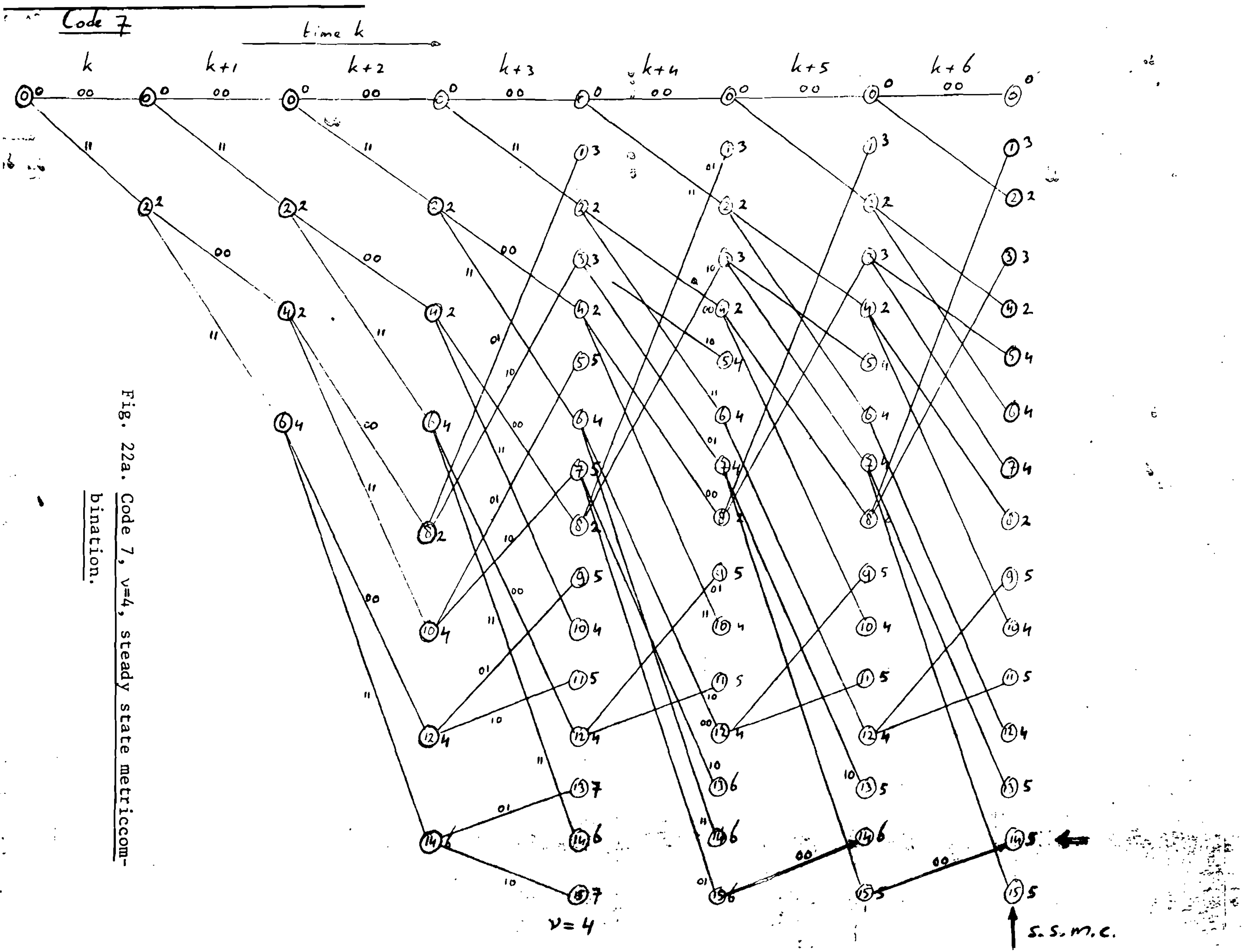


Fig. 22a. Code 7, ν=4, steady state metric combination.

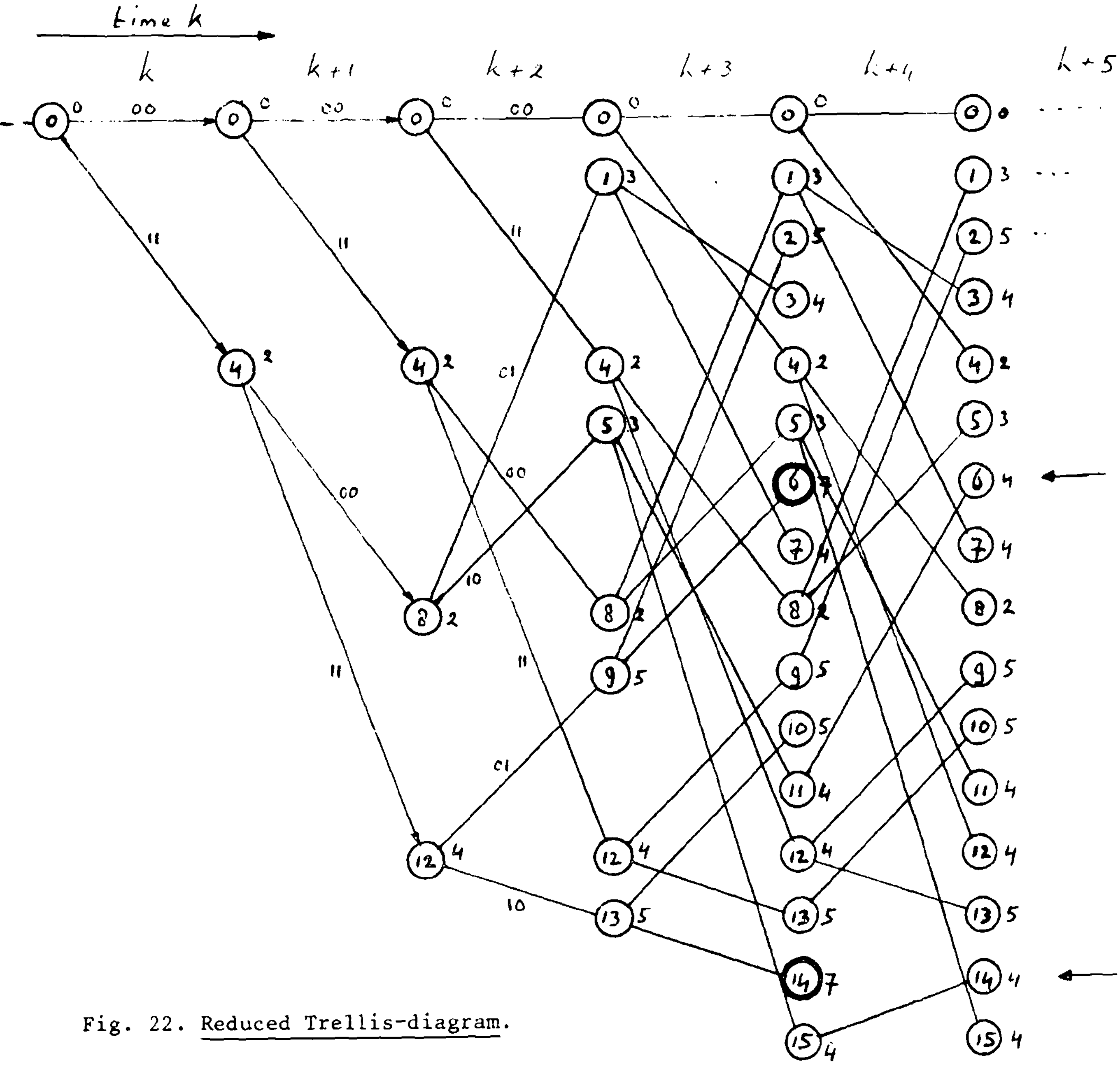


Fig. 22. Reduced Trellis-diagram.

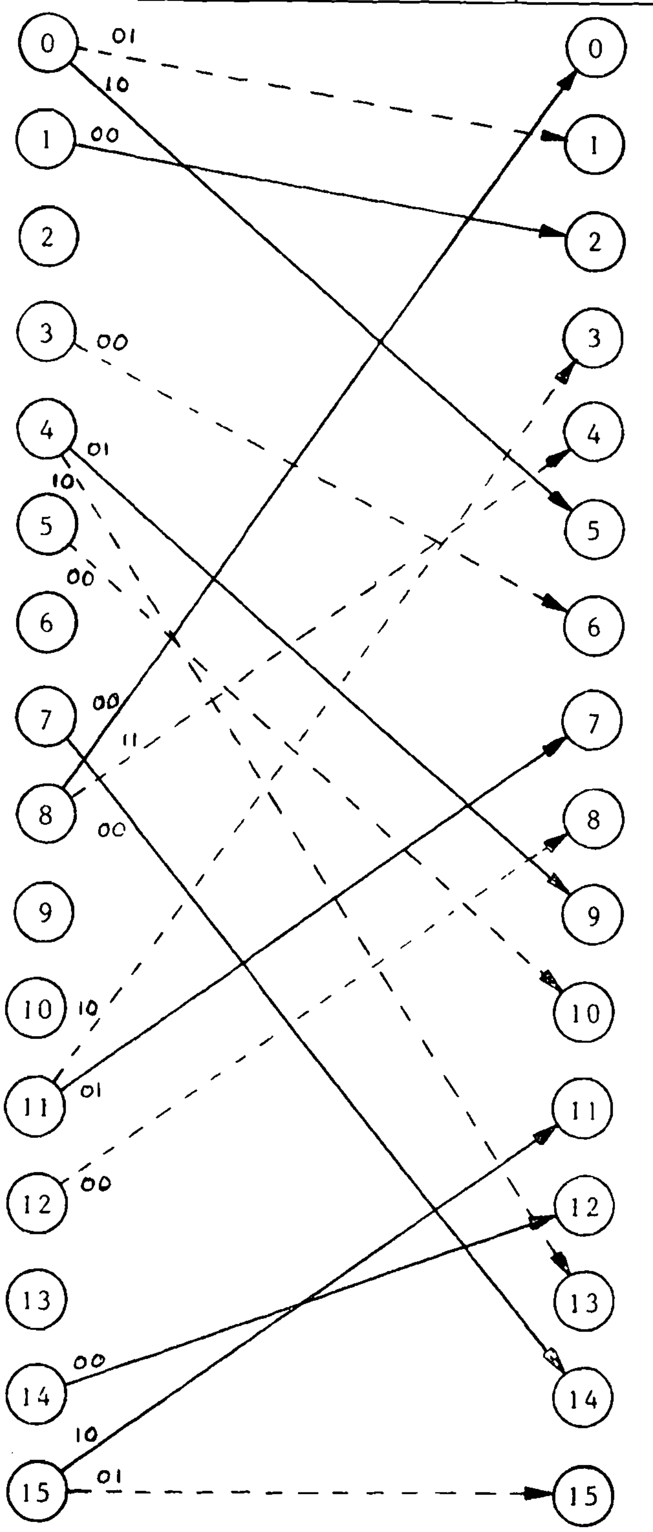
The s.s.m.c. for our example is $\underline{m}_0(k) = \{0354234425544544\}$. We leave it to the reader to verify that indeed:

$$\underline{m}_0(k) \xrightarrow{[z(\alpha)]_k = 0} \underline{m}_0(k+1)$$

With an output $[z(\alpha)]_k = 1$ we are able to evaluate the new combination $\underline{m}_1(k+1)$. In Fig.23 we have illustrated the transition between the metric-combinations $\underline{m}_0$ and $\underline{m}_1$.

Since only the difference between the metrics within a metriccombination matters, we may subtract from all members the minimumvalue in the combination.

$\underline{m}_0(k)$ $M_p(k)$	$S_p(k)$ index p	$[z(\alpha)]_k = 1$ $[n_1(\alpha),n_2(\alpha)]_k(p,i)$	$S_i(k+1)$ index i	$\underline{m}_1(k+1)$ $M_i(k+1)$	survivor-index p	index(cos) of minimum
0	0		0	2 := 1	8	
3	1		1	1 := 0	0	
5	2		2	3 := 2	1	
4	3		3	5 := 4	11	
2	4		4	4 := 3	8	
3	5		5	1 := 0	1	
4	6		6	4 := 3	3	
4	7		7	5 := 4	11	(1,5)
2	8		8	4 := 3	12	
5	9		9	3 := 2	4	
5	10		10	3 := 2	5	
4	11		11	5 := 4	15	
4	12		12	4 := 3	14	
5	13		13	3 := 2	4	
4	14		14	4 := 3	7	
4	15		15	5 := 4	15	



- - - - - - - -▶ $[w(\alpha)]_k = 1$

————————▶ $[w(\alpha)]_k = 0$

Fig. 23.
Transition between the metric-combinations $\underline{m}_0$ and $\underline{m}_1$.

Via the transitions $z(\alpha)=0$ and $z(\alpha)=1$ all possible combinations can be evaluated successively. As soon as a new combination is found, it is placed at the left in the transition-table with a serial-number c. This proces goes on until no new member can be found. For codes with constraintlength $\nu=2$ this algorithm can be executed by hand [4]. For codes with $\nu \geqslant 3$ a computer-program has been developed [5]. This program calculates the total number C+1 of metriccombinations together with the transitions between them for $z(\alpha)=0$ and $z(\alpha)=1$. Since each code has distinct metric-equations and a distinct s.s.m.c. $\underline{m}_0$, the computer-program must be modified for each code.

In order to get an impression about the necessary computing-time we will give some figures.

code	C+1	comp-time	≈
1	2001	218 sec.	4 min.
2	144	2 "	
6-Y	2582	307 "	5 "
10	13518	8435 "	2 uur 20 min.
11	16608	10800 "	3 "

Knowledge of the successive survivorindices for each transition between metriccombinations, together with the index(ces) of the minimum-value within the new combination is sufficient to determine the key-sequence $[w(\alpha)]_k^{\infty}$. In Fig.23 we have indicated the survivorindices p; the indices of the minimum within the new combination $\underline{m}_1$ are 1 and 5. The total amount of information about the transitions between the metriccombinations, the survivorindices and the indices of the minimum within the new combination, can be modified and stored in a ROM.

In section 10 (Fig. 37), all 75 metriccombinations together with their transitions are given for code 1 of constraintlength $\nu=3$.

More detailed information about the operation of this core-part of the syndromedecoder can be found in [4].

Comparison of the s.s.m.c. and the total number C+1 of metriccombinations for the various codes with $\nu=3$ and $\nu=4$.

A first effort in discovering any systematics in the s.s.m.c. and the total number of combinations of the various codes is little successful. As can be expected a reflection of the code in the X-axis does not change the total number of combinations. The two s.s.m.c. for this reflection are equivalent in that sense, that they contain the same metricvalues, only on different places. The codes C can be converted in the codes C-X by a renumbering of the states.

As an example we compare the state-diagrams of the codes 1 and 1-X. The four diagrams ($z(\alpha)=0$ and $z(\alpha)=1$) are drawn in Fig.24.

With a renumbering as:

$S_1 := S_5$	$S_6 := S_{14}$	$S_{12} := S_{12}$
$S_2 := S_{10}$	$S_7 := S_{11}$	$S_0 := S_0$
$S_3 := S_{15}$	$S_8 := S_8$	
$S_4 := S_4$	$S_9 := S_{13}$	

the metricequations will be equivalent since $d_H[0,1]=d_H[1,0]$. Hence the s.s.m.c. and the total number of metriccombinations will be equivalent.

When we consider the total number of metriccombinations in relation to the connectionpattern of the polynomials $C_n(\alpha)$, n=1,2 we get the table of Fig.25.

connection-pattern $C_n(\alpha)$		codes	C+1	pathregisters	
$c_{1,k} \neq c_{2,k}$	$k=1\wedge 2\wedge 3$	10,11	11000-16000	12	
$c_{1,k} = c_{2,k}$	$k=2$	12,13	459-659	12	
$c_{1,k} = c_{2,k}$	$k=1\vee 3$	8,9	195-270	12	$\nu=4$
$c_{1,k} = c_{2,k}$	$k=(1,2)\vee(2,3)$	4,5,6,7	1642-2582	12	
$c_{1,k} = c_{2,k}$	$k=1\wedge 3$	1,2,3	144-2805	9	
$c_{1,k} = c_{2,k}$	$k=1\wedge 2$	3	74-99	6	$\nu=3$
$c_{1,k} = c_{2,k}$	$k=1\vee 2$	1,2	33	6	

Fig. 25. Connection-pattern and total number C+1 of metriccombinations.

A remarkeble point is that the codes 10 and 11 have considerable more metriccombinations then the other codes. It could be that this

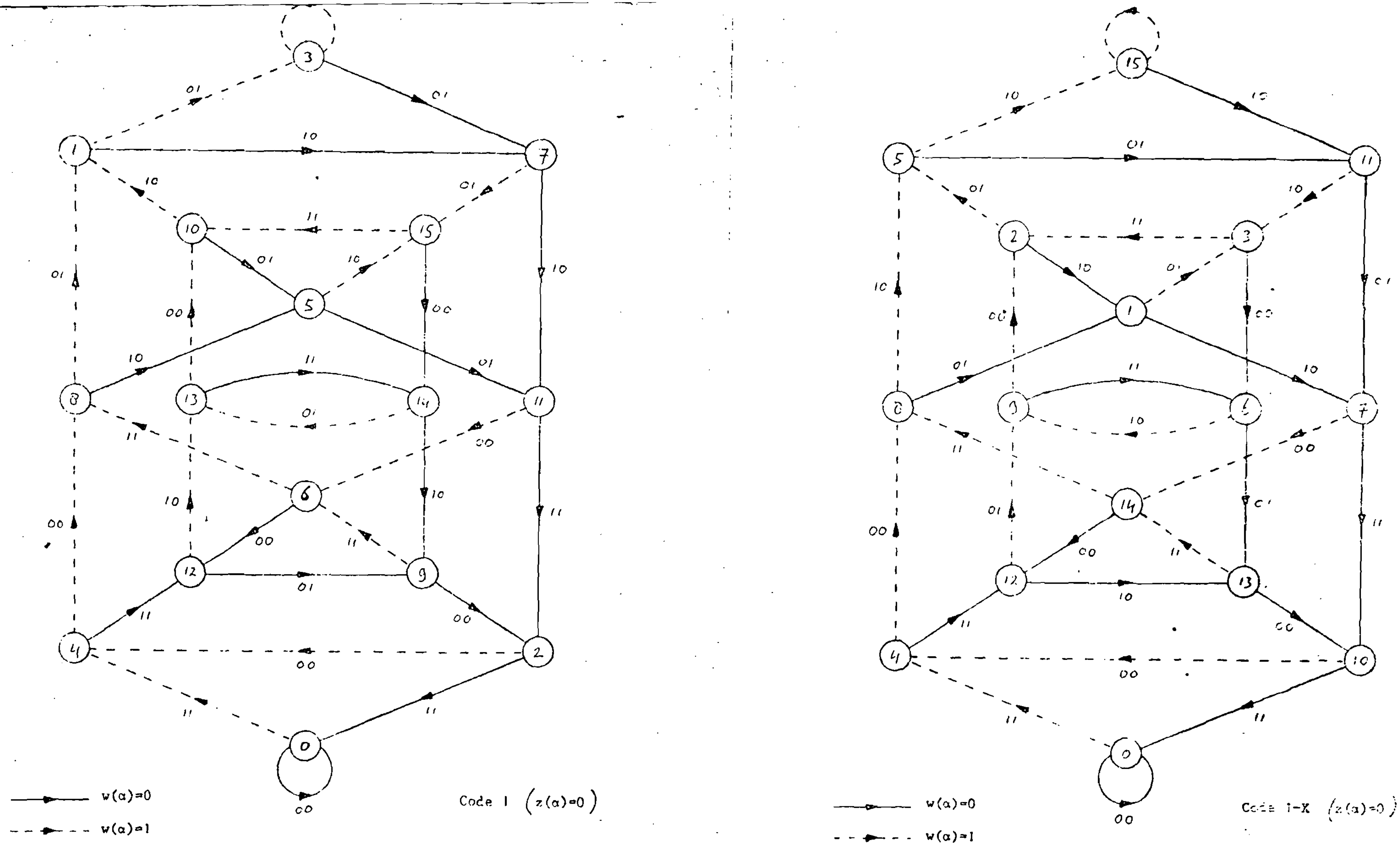


Fig. 24a. Complementary state-diagrams.

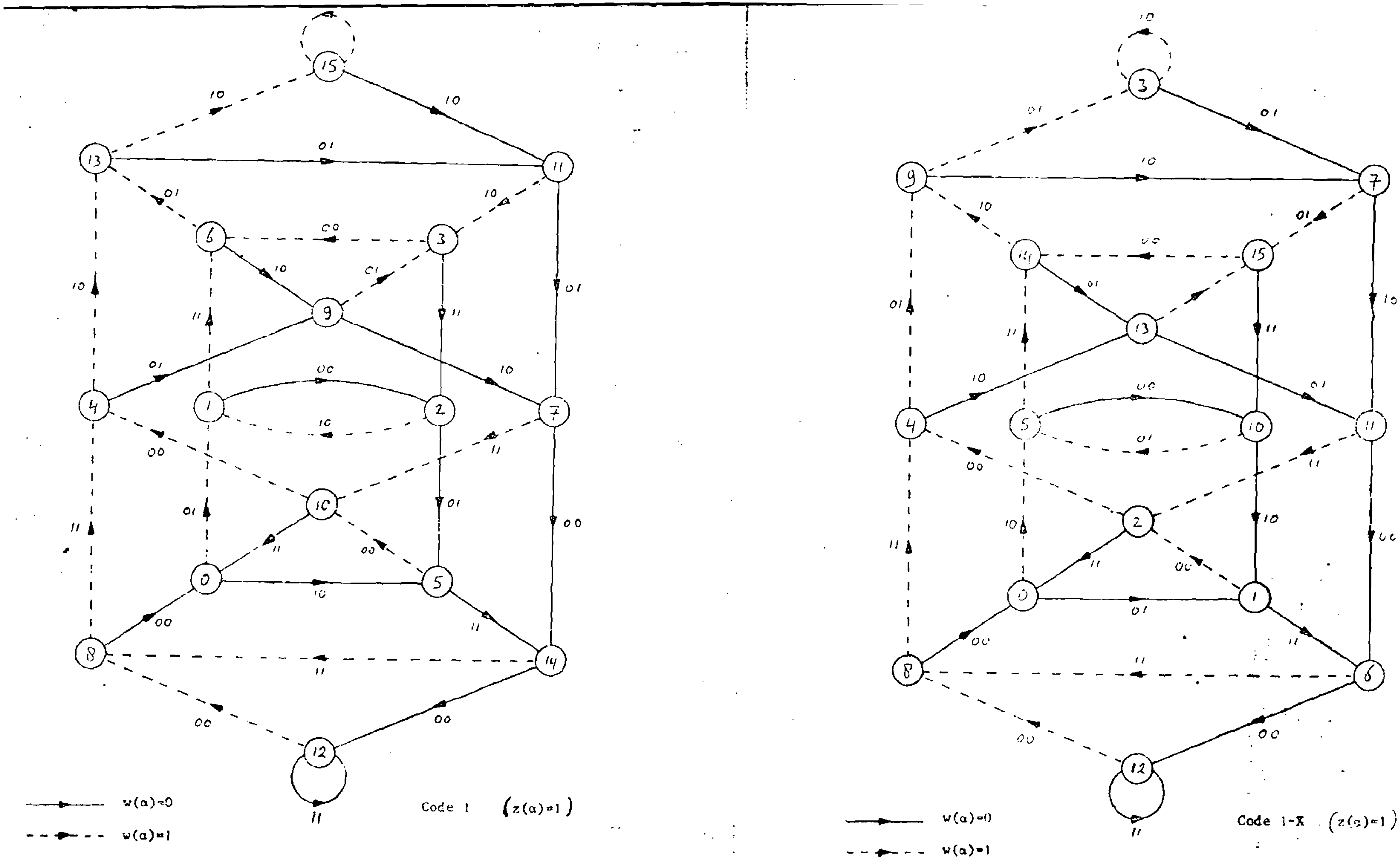


Fig. 24b. Complementary state-diagrams.

property is related to the typical connection-pattern $c_{1,k} \neq c_{2,k}$ for all $k \neq \{0,\nu\}$.

In Fig.26 we have plotted the value C+1 as a function of the constraintlength ν.

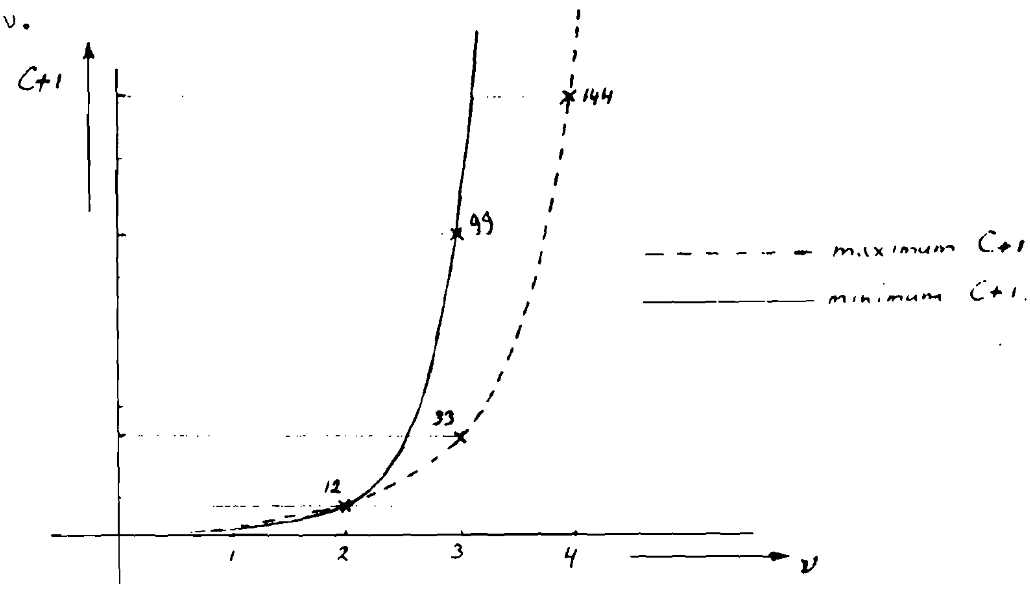


Fig. 26. Metriccombinations and constraintlength.

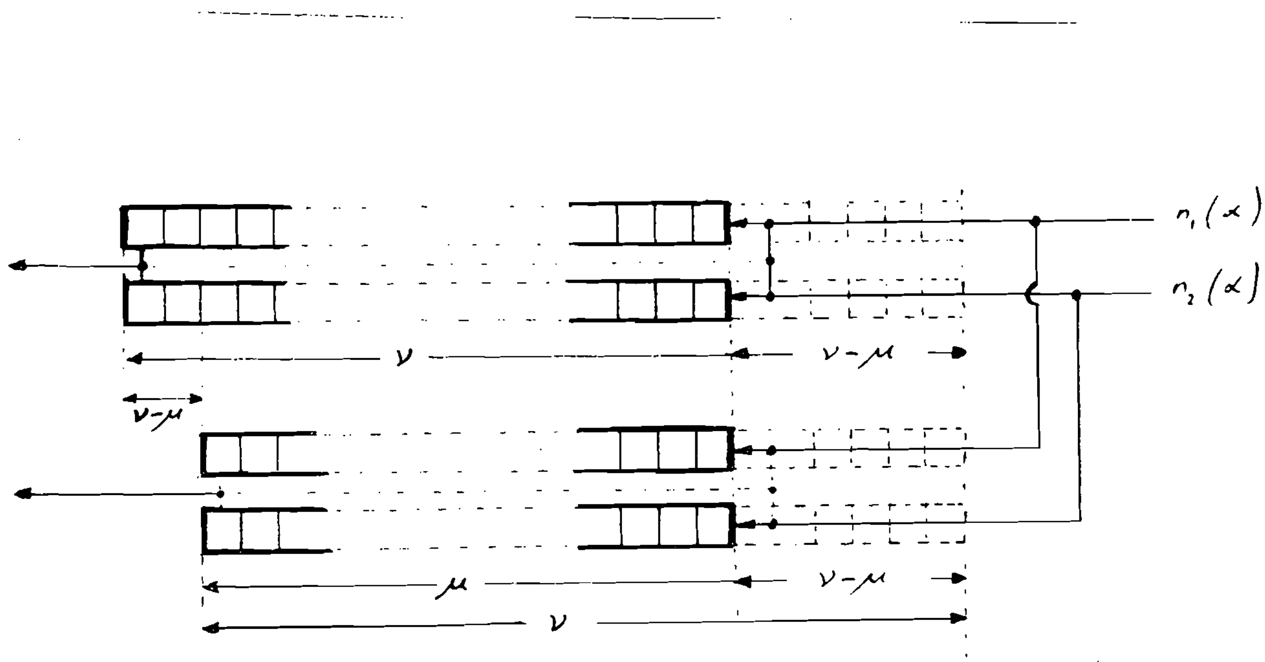
For the time being it will not be possible with the present time and means, to calculate the total number of metriccombinations for codes of constraintlength $\nu \geqslant 5$.

8. CALCULATION OF FIRST BITS IN THE PATHREGISTER.

There are codes for which the $\nu-\mu$ outputvalues $[w(\alpha)]_{k-\nu+\mu}^{k-1}(p,i)$, associated with any path $[n_1(\alpha),n_2(\alpha)]_{k-\nu+\mu}^{k-1}(p,i)$ leading to the state $S_i(k)$ are all the same. Knowledge of these pathregisterbits leads to an extra saving in hardware required for realisation of the decoder.

We will now show that for codes of constraintlength ν with polynomials $D_n(\alpha)$, n=1,2, of degree $\mu \leqslant \nu-1$, exactly $\nu-\mu$ pathregisterbits $[w(\alpha)]_{k-\nu+\mu}^{k-1}$ can be taken as fixed.

In the figure below we have drawn the syndromeformer and the w(α)-former of an arbitrary code of constraintlength ν with $D_n(\alpha)$-polynomials of degree μ:



Suppose that both circuits are in the physical state S_1 at time 0. As inputsequence we take the physical state $[n_1(\alpha),n_2(\alpha)]_0^{\nu-1} = S_2$. We will now describe the transition in ν steps from $S_1 \longrightarrow S_2$.

After μ steps the w(α)-former is completely filled-up with the prefix $[S_2]_0^{\mu-1}$ of the state S_2. From that moment the remaining ν-μ output-values $[w(\alpha)]_\mu^{\nu-1}$ are unambiguously determined by the last ν-μ bits of the state S_2. These last bits are fixed and hence the last ν-μ bits of the w(α)-former are fixed.

This applies to all 2^ν states. As there are only $2^{\nu-\mu}$ combinations of ν-μ bits, there must be 2^μ groups of states with the same fixed path-registerbits.

As an example we take code 10 with ν=4 . The state-table of this code is shown in Fig. 27.

state	state	new state $S_i(k+1)$				value $[z(\alpha)]_k$				value $[w(\alpha)]_k$			
$S_j(k)$	contents	$[n_1(\alpha),n_2(\alpha)]_k$				$[n_1(\alpha),n_2(\alpha)]_k$				$[n_1(\alpha),n_2(\alpha)]_k$			
j	$s_j(\alpha)$	00	01	11	10	00	01	11	10	00	01	11	10
0	0000	0	1	14	(15)	0	1	0	1	0	1	1	0
1	0001	2	3	12	13	1	0	1	0	1	0	0	1
2	0010	4	5	10	11	0	1	0	1	0	1	1	0
3	0011	6	7	8	9	1	0	1	0	1	0	0	1
4	0100	8	9	6	7	0	1	0	1	0	1	1	0
5	0101	10	11	4	5	1	0	1	0	1	0	0	1
6	0110	12	13	2	3	0	1	0	1	0	1	1	0
7	0111	14	15	0	1	1	0	1	0	1	0	0	1
8	1000	0	1	14	15	1	0	1	0	0	1	1	0
9	1001	2	3	12	13	0	1	0	1	1	0	0	1
10	1010	4	5	10	11	1	0	1	0	0	1	1	0
11	1011	6	7	8	9	0	1	0	1	1	0	0	1
12	1100	8	9	6	7	1	0	1	0	0	1	1	0
13	1101	10	11	4	5	0	1	0	1	1	0	0	1
14	1110	12	13	2	3	1	0	1	0	0	1	1	0
(15)	1111	14	15	0	1	0	1	0	1	1	0	0	1

Fig. 27. State-table of code 10.

A part of this state-table is drawn in Fig.28 where S_6 is taken as the central node. S_6 can be reached from the parent states S_3,S_4,S_{11} and S_{12} with $[w(\alpha)]_{k-1}=1$. Each of these parent-states can be reached at its turn from the groups (S_1,S_6,S_9,S_{14}) and (S_2,S_5,S_{10},S_{13}) with $[w(\alpha)]_{k-2}=0$. One step further all states $S_1,S_2,S_5,S_6,S_9,S_{10},S_{13}$ and S_{14} have a last pathregisterbit $[w(\alpha)]_{k-3}=1$. When we go back one more step we cannot predict the value of $[w(\alpha)]_{k-4}$ any more as it may be either 0 or 1(consider the paths $S_0-S_1-S_2-S_4-S_6$ and $S_0-S_0-S_1-S_3-S_6$).

Note that both S_6 and S_9 can be reached from the states S_3,S_4,S_{11} and S_{12}. Hence the last three bits in the pathregisters PR[6] and PR[9] can be recorded as:

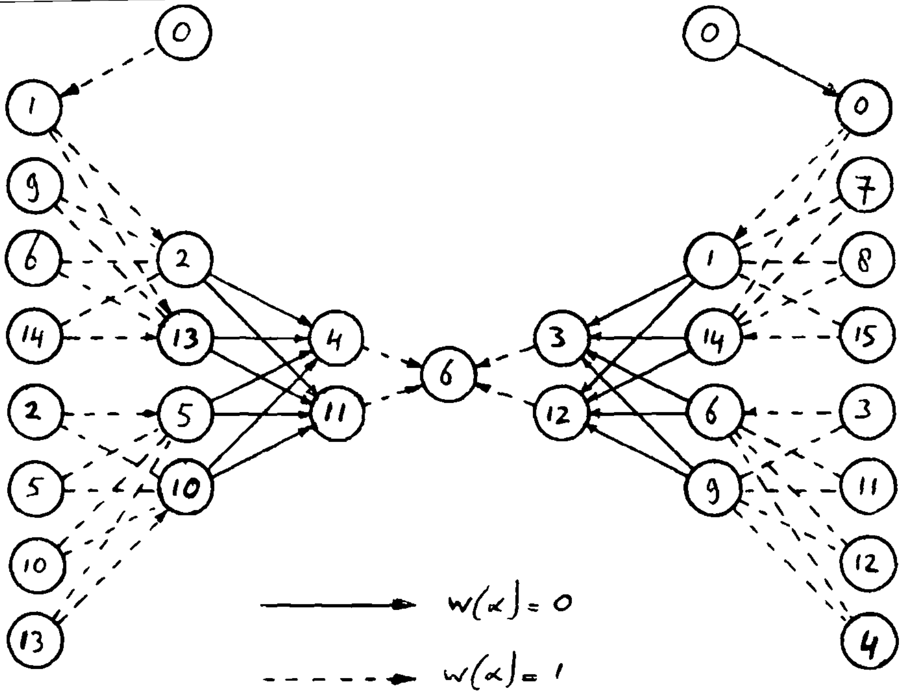


Fig. 28. Parent states of S_6.

$$PR[6]_{k-3}^{k-1}=PR[9]_{k-3}^{k-1}=[w(\alpha)]_{k-3}^{k-1}(p,6)=[w(\alpha)]_{k-3}^{k-1}(p,9)=[1\ 0\ 1]_{k-3}^{k-1}.$$

When we determine the contents of all pathregisters $PR[i]_{k-3}^{k-1}$, $i=0,1,\ldots,2^{\nu}-1$, we get the table of Fig.29.

state $S_i(k)$	pathregisterbits $PR[i]_{k-3}^{k-1}(p,i)$
0	0 0 0
1	0 0 1
2	0 1 1
3	0 1 0
4	0 1 0
5	0 1 1
6	0 0 1
7	0 0 0
8	1 0 0
9	1 0 1
10	1 1 1
11	1 1 0
12	1 1 0
13	1 1 1
14	1 0 1
15	1 0 0

Fig. 29. Fixed pathregisterbits.

Note that there are always $2^{\nu-1}$ pairs of pathregisters with identical last pathregisterbits $[w(\alpha)]_{k-1}$.
The last bit in the pathregister PR[i] is recorded as given in the first two columns of the head " $[w(\alpha)]_k$ " of the state-table. There is an easy and fast method for determination of the fixed pathregisterbits of any code.
From (8) and (9) we know that the even states S_i ($i=2k$, $k=0,1,\ldots,2^{\nu-2}-1$) and the odd states S_{i+1}, both have $S_{\frac{1}{2}i}$ as parent state. Thus each time $k-\ell-1$ ($\ell=0,1,\ldots$) we jump back from the states S_i and S_{i+1} to the state $S_{\frac{1}{2}i}$ and record their last pathregisterbits. The column $PR[i]_{k-2}$ can thus be derived from the column $PR[i]_{k-1}$ by multiplication by a factor 2. In essence, each bit in the column $PR[i]_{k-1}$ is written double in the column $PR[i]_{k-2}$. The column $PR[i]_{k-3}$

arises from the column $PR[i]_{k-2}$ again with a factor 2, that is each bit in the column $PR[i]_{k-1}$ is written four times, etc.

The fixed pathregisterbits of all codes of constraintlength $\nu=3$ and $\nu=4$ are recorded in the tables of Fig.30.

The codes with minimum degree $D_n(\alpha) = 1$, have the maximum $\nu-1$, of fixed pathregisterbits.

From the tables of the figures 13 and 14 we derive that these codes have either $S_b = S_3$ or $S_b = S_{2^\nu-1}$. Why there exists such a relationship is still a subject of research.

9. CODES OF CONSTRAINTLENGTH $\nu=6$.

A computer-search program has been developed [6] which computes all 24 codes of constraintlength $\nu=6$ which achieve the free distance $d_{free} = 9/10$ These codes have connectionpolynomials $C_n(\alpha)$, $n=1,2$, as shown in the table of Fig.31.

From a practical point of view we are interested in codes which require a minimum amount of hardware in implementation. According to [3] codes of constraintlength $\nu=6$ with a maximal reduction in pathregisters must have polynomials $C_n(\alpha)$, $n=1,2$, such that $c_{1,k} = c_{2,k}$, $k=0,1,2,4,5$ and 6. If we look at the table of Fig.31 we notice that there exists no such code with $d_{free}=9$. There are two codes which have $c_{1,k} = c_{2,k}$, $k=0,1,5,6$, namely code 1 and code 2. (both in $T_{6,\ell}$).

Let us consider code 2 with connectionpolynomials

$$C_1(\alpha) = \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha^0 \longrightarrow (\underline{1}\ \underline{1}\ 0\ 1\ 1\ \underline{0}\ \underline{1})$$

$$C_2(\alpha) = \alpha^6 + \alpha^5 + \alpha^4 + \alpha^0 \longrightarrow (\underline{1}\ \underline{1}\ 1\ 0\ 0\ \underline{0}\ \underline{1})$$

and polynomials

$$D_1(\alpha) = \alpha^2 \longrightarrow (0\ 0\ 0\ 1\ 0\ 0)$$

$$D_2(\alpha) = \alpha^2 + \alpha^0 \longrightarrow (0\ 0\ 0\ 1\ 0\ 1)$$

$[w(\alpha)]_{k-1}$	PR[i]	—	—	—	—	—	—	—
codes —	6-X/Y	4-Y,9	7-Y,12	5-X/Y,7	1-X,5-Y,11	1,4-X,9-X,7-X/Y	3,4-X/Y,11-X,12-X	3-X,4,6-Y,7-X
0	0,2,4,6,9 11,13,15	0,1,4,5,10 11,14,15	0,1,2,3,12 13,14,15	0,3,5,6,9 10,12,15	0,1,6,7,10 11,12,13	0,2,5,7,9 11,12,14	0,3,4,7,9 11,12,14	0,1,2,3,4 5,6,7
1	1,3,5,7,8 10,12,14	2,3,6,7,8 9,12,13	4,5,6,7,8 9,10,11	1,2,4,7,8 11,13,14	2,3,4,5,8 9,14,15	1,3,4,6,8 10,13,15	1,2,5,6,8 11,12,15	8,9,10,11 12,13,14,15

$[w(\alpha)]_{k-2}^{k-1}$	PR[i]	—	—	—
codes —	1-Y,8-Y	2-X,9-X/Y	2,5-X,9-Y,10-X/Y	1-X/Y,5,8-X/Y,10-Y
0 0	0,1,14,15	0,5,10,15	0,1,2,3	0,6,11,13
0 1	2,3,12,13	1,4,11,14	4,5,6,7	1,7,10,12
1 0	6,7,8,9	2,7,8,13	8,9,10,11	3,5,8,14
1 1	4,5,10,11	3,6,9,12	12,13,14,15	2,4,9,15

$[w(\alpha)]_{k-3}^{k-1}$	PR[i]	—
codes —	6,8-X,10-X,13	6-X,8,10,13-X
0 0 0	0,1	0,15
0 0 1	2,3	1,14
0 1 0	4,5	3,12
0 1 1	6,7	2,13
1 0 0	8,9	7,8
1 0 1	10,11	6,9
1 1 0	12,13	4,11
1 1 1	14,15	5,10

constraintlength ν=4.

$[w(\alpha)]_{k-1}$	PR[i]	—	—	—
codes —	1,1-X/Y	1-X,2-Y	1-Y	2-X/Y
0	0,3,5,6	0,1,2,3	0,1,6,7	0,2,5,7
1	1,2,4,7	4,5,6,7	2,3,4,5	1,3,4,6

$[w(\alpha)]_{k-3}^{k-1}$	PR[i]	—
codes —	2,3-X	2-X,3
0 0	0,1	0,7
0 1	2,3	1,6
1 0	4,5	3,4
1 1	6,7	2,5

constraintlength ν=3.

Fig. 30. <u>Fixed pathregisterbits of codes with ν=3 and ν=4.</u>

code	polynomials $C_1(\alpha)/C_2(\alpha)$	code	polynomials $C_1(\alpha)/C_2(\alpha)$	
1	1 0 1 1 0 1 1 1 0 1 1 1 1 1	13	1 0 0 0 1 0 1 1 1 1 0 1 1 1	
2	1 1 0 1 1 0 1 1 1 1 0 0 0 1	14	1 0 0 0 1 1 1 1 1 1 0 1 0 1	
3	1 0 0 1 0 1 1 1 1 0 1 0 1 1	15	1 0 0 1 1 0 1 1 1 0 1 0 1 1	
4	1 0 0 0 1 1 1 1 1 0 1 1 0 1	16	1 0 1 1 1 0 1 1 1 0 1 1 1 1	
5	1 0 1 0 1 1 1 1 1 0 1 1 1 1	17	1 0 1 1 0 1 1 1 1 1 1 1 0 1	
6	1 0 1 0 1 1 1 1 1 0 0 0 1 1	18	1 0 1 0 0 1 1 1 1 1 0 1 0 1	
7	1 0 0 1 1 0 1 1 1 1 1 0 0 1	19	1 0 0 1 1 1 1 1 1 0 1 1 0 1	– d_{free} = 10
8	1 0 1 0 1 1 1 1 1 1 1 0 1 1	20	1 0 1 1 1 0 1 1 1 0 0 1 1 1	– d_{free} = 10
9	1 0 0 1 0 1 1 1 1 0 0 1 1 1	21	1 0 0 1 1 0 1 1 1 0 0 1 1 1	
10	1 0 0 0 1 1 1 1 0 1 1 1 0 1	22	1 0 0 1 1 0 1 1 1 1 0 0 1 1	
11	1 0 0 1 1 0 1 1 0 1 0 1 1 1	23	1 0 1 0 0 1 1 1 1 0 1 1 0 1	
12	1 0 0 1 0 1 1 1 1 1 1 0 0 1	24	1 0 0 0 1 0 1 1 1 1 1 0 1 1	

Fig. 31. Codes with free distance d_{free} = 9/10.

of minimum degree which satisfy $C_1(\alpha)D_1(\alpha) + C_2(\alpha)D_2(\alpha) = 1$.
The syndromeformer and the $w(\alpha)$-former are drawn in Fig.32 and Fig.33 respectively.

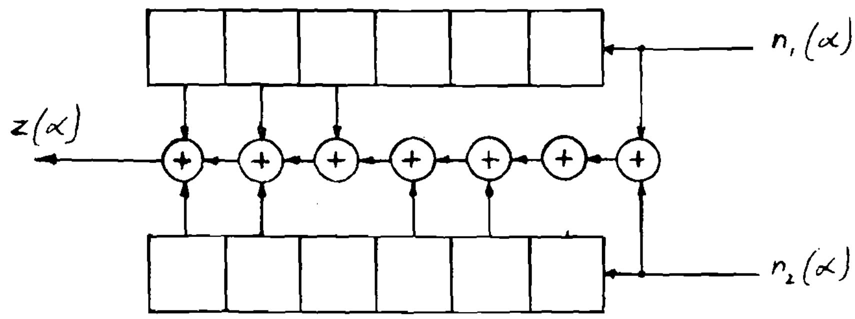


Fig. 32. Syndromeformer.

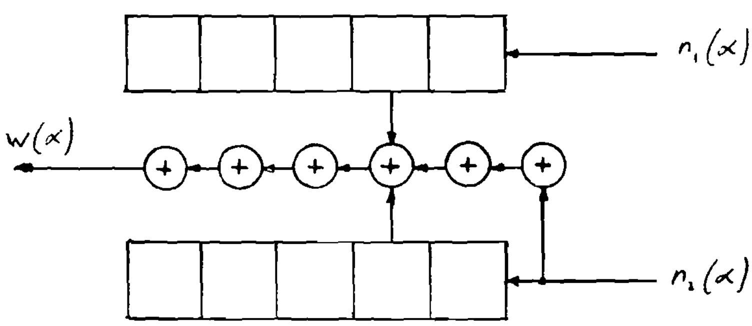


Fig. 33. $w(\alpha)$-former.

There are $2^6=64$ abstract states of the syndromeformer. The base state $\mathbf{s}_b$ is determined as follows:

$$s_{b,-2} + s_{b,-3} + s_{b,-5} + s_{b,-6} = 0 \qquad s_{b,-6} = 0$$

$$1 + s_{b,-2} + s_{b,-4} + s_{b,-5} = 0 \qquad s_{b,-5} = 1$$

$$1 + s_{b,-3} + s_{b,-4} = 0 \qquad s_{b,-4} = 0$$

$$1 + s_{b,-2} + s_{b,-3} = 0 \qquad s_{b,-3} = 1$$

$$1 + 1 + s_{b,-2} = 0 \qquad s_{b,-2} = 0$$

$$1 + s_{b,-1} = 0 \qquad s_{b,-1} = 0$$

$$s_{b=21} = [0\ 0\ 0\ 0\ 0\ 1\ ,\ 0\ 1\ 0\ 1\ 0\ 1]$$

The state-table of this code is given in Fig.34. It is clear that for such an amount of states, a surveyable representation of the transitions between states in the form of a state-diagram or Trellis-diagram is not possible.

From the state-table we derive the metricequations which are given in the metric-table of Fig.35.

Since this code is in the class $\Gamma_{6,2}$ there will be

$2^{\nu-2\ell}(4^{\ell}-3^{\ell}) = 28$ pairs of equal metricequations.

The number of pathregisterbits which can be taken fixed is

ν - {degree $D_n(\alpha)$, n=1,2} = 6 - 2 = 4.

The essential features of these 24 codes are to be found in the enclosed "Databook" at the pages 49-104.

$S_j(k)$	contents	$[n_1(\alpha),n_2(\alpha)]_k$				$[n_1(\alpha),n_2(\alpha)]_k$				$[n_1(\alpha),n_2(\alpha)]_k$			
j	$s_j^*(\alpha)$	00	01	11	10	00	01	11	10	00	01	11	10
0	000000	0	1	20	21 (circled)	0	1	0	1	0	1	1	0
1	000001	2	3	22	23	0	1	0	1	0	1	1	0
2	000010	4	5	16	17	1	0	1	0	1	0	0	1
3	000011	6	7	18	19	1	0	1	0	1	0	0	1
4	000100	8	9	28	29	1	0	1	0	0	1	1	0
5	000101	10	11	30	31	1	0	1	0	0	1	1	0
6	000110	12	13	24	25	0	1	0	1	1	0	0	1
7	000111	14	15	26	27	0	1	0	1	1	0	0	1
8	001000	16	17	4	5	0	1	0	1	0	1	1	0
9	001001	18	19	6	7	0	1	0	1	0	1	1	0
10	001010	20	21	0	1	1	0	1	0	1	0	0	1
11	001011	22	23	2	3	1	0	1	0	1	0	0	1
12	001100	24	25	12	13	1	0	1	0	0	1	1	0
13	001101	26	27	14	15	1	0	1	0	0	1	1	0
14	001110	28	29	8	9	0	1	0	1	1	0	0	1
15	001111	30	31	10	11	0	1	0	1	1	0	0	1
16	010000	32	33	52	53	1	0	1	0	0	1	1	0
17	010001	34	35	54	55	1	0	1	0	0	1	1	0
18	010010	36	37	48	49	0	1	0	1	1	0	0	1
19	010011	38	39	50	51	0	1	0	1	1	0	0	1
20	010100	40	41	60	61	0	1	0	1	0	1	1	0
21 (circled)	010101	42	43	62	63	0	1	0	1	0	1	1	0
22	010110	44	45	56	57	1	0	1	0	1	0	0	1
23	010111	46	47	58	59	1	0	1	0	1	0	0	1
24	011000	48	49	36	37	1	0	1	0	0	1	1	0
25	011001	50	51	38	39	1	0	1	0	0	1	1	0
26	011010	52	53	32	33	0	1	0	1	1	0	0	1
27	011011	54	55	34	35	0	1	0	1	1	0	0	1
28	011100	56	57	44	45	0	1	0	1	0	1	1	0
29	011101	58	59	46	47	0	1	0	1	0	1	1	0
30	011110	60	61	40	41	1	0	1	0	1	0	0	1
31	011111	62	63	42	43	1	0	1	0	1	0	0	1

$S_j(k)$	contents	$[n_1(\alpha),n_2(\alpha)]_k$				$[n_1(\alpha),n_2(\alpha)]_k$				$[n_1(\alpha),n_2(\alpha)]_k$			
j	$s_j^*(\alpha)$	00	01	11	10	00	01	11	10	00	01	11	10
32	100000	0	1	20	21	1	0	1	0	0	1	1	0
33	100001	2	3	22	23	1	0	1	0	0	1	1	0
34	100010	4	5	16	17	0	1	0	1	1	0	0	1
35	100011	6	7	18	19	0	1	0	1	1	0	0	1
36	100100	8	9	28	29	0	1	0	1	0	1	1	0
37	100101	10	11	30	31	0	1	0	1	0	1	1	0
38	100110	12	13	24	25	1	0	1	0	1	0	0	1
39	100111	14	15	26	27	1	0	1	0	1	0	0	1
40	101000	16	17	4	5	1	0	1	0	0	1	1	0
41	101001	18	19	6	7	1	0	1	0	0	1	1	0
42	101010	20	21	0	1	0	1	0	1	1	0	0	1
43	101011	22	23	2	3	0	1	0	1	1	0	0	1
44	101100	24	25	12	13	0	1	0	1	0	1	1	0
45	101101	26	27	14	15	0	1	0	1	0	1	1	0
46	101110	28	29	8	9	1	0	1	0	1	0	0	1
47	101111	30	31	10	11	1	0	1	0	1	0	0	1
48	110000	32	33	52	53	0	1	0	1	0	1	1	0
49	110001	34	35	54	55	0	1	0	1	0	1	1	0
50	110010	36	37	48	49	1	0	1	0	1	0	0	1
51	110011	38	39	50	51	1	0	1	0	1	0	0	1
52	110100	40	41	60	61	1	0	1	0	0	1	1	0
53	110101	42	43	62	63	1	0	1	0	0	1	1	0
54	110110	44	45	56	57	0	1	0	1	1	0	0	1
55	110111	46	47	58	59	0	1	0	1	1	0	0	1
56	111000	48	49	36	37	0	1	0	1	0	1	1	0
57	111001	50	51	38	39	0	1	0	1	0	1	1	0
58	111010	52	53	32	33	1	0	1	0	1	0	0	1
59	111011	54	55	34	35	1	0	1	0	1	0	0	1
60	111100	56	57	44	45	1	0	1	0	0	1	1	0
61	111101	58	59	46	47	1	0	1	0	0	1	1	0
62	111110	60	61	40	41	0	1	0	1	1	0	0	1
63	111111	62	63	42	43	0	1	0	1	1	0	0	1

Fig. 34. State-table code 2.

Fig. 35. Metric-table.

metric $M_i(k+1)$	$[z(\alpha)]_k = 0$ minimumvalue	$[z(\alpha)]_k = 1$ minimumvalue
M_0	$M_0 \quad , M_{42}+2$	$M_{32} \quad , M_{10}+2$
M_1	$M_{32}+1, M_{10}+1$	$M_0+1, M_{42}+1$
M_2	$M_1 \quad , M_{43}+2$	$M_{33} \quad , M_{11}+2$
M_3	$M_{33}+1, M_{11}+1$	$M_1+1, M_{43}+1$
M_4	$M_{34} \quad , M_8+2$	$M_2 \quad , M_{40}+2$
M_5	$M_2+1, M_{40}+1$	$M_{34}+1, M_8+1$
M_6	$M_{35} \quad , M_9+2$	$M_3 \quad , M_{41}+2$
M_7	$M_3+1, M_{41}+1$	$M_{35}+1, M_9+1$
M_8	$M_{36} \quad , M_{14}+2$	$M_4 \quad , M_{46}+2$
M_9	$M_4+1, M_{46}+1$	$M_{36}+1, M_{14}+1$
M_{10}	$M_{37} \quad , M_{15}+2$	$M_5 \quad , M_{47}+2$
M_{11}	$M_5+1, M_{47}+1$	$M_{37}+1, M_{15}+1$
M_{12}	$M_6 \quad , M_{44}+2$	$M_{38} \quad , M_{12}+2$
M_{13}	$M_{38}+1, M_{12}+1$	$M_6+1, M_{44}+1$
M_{14}	$M_7 \quad , M_{45}+2$	$M_{39} \quad , M_{13}+2$
M_{15}	$M_{39}+1, M_{13}+1$	$M_7+1, M_{45}+1$
M_{16}	$M_8 \quad , M_{34}+2$	$M_{40} \quad , M_2+2$
M_{17}	$M_{40}+1, M_2+1$	$M_8+1, M_{34}+1$
M_{18}	$M_9 \quad , M_{35}+2$	$M_{41} \quad , M_3+2$
M_{19}	$M_{41}+1, M_3+1$	$M_9+1, M_{35}+1$
M_{20}	$M_{42} \quad , M_0+2$	$M_{10} \quad , M_{32}+2$
M_{21}	$M_{10}+1, M_{32}+1$	$M_{42}+1, M_0+1$
M_{22}	$M_{43} \quad , M_1+2$	$M_{11} \quad , M_{33}+2$
M_{23}	$M_{11}+1, M_{33}+1$	$M_{43}+1, M_1+1$
M_{24}	$M_{44} \quad , M_6+2$	$M_{12} \quad , M_{38}+2$
M_{25}	$M_{12}+1, M_{38}+1$	$M_{44}+1, M_6+1$
M_{26}	$M_{45} \quad , M_7+2$	$M_{13} \quad , M_{39}+2$
M_{27}	$M_{13}+1, M_{39}+1$	$M_{45}+1, M_7+1$
M_{28}	$M_{14} \quad , M_{36}+2$	$M_{46} \quad , M_4+2$
M_{29}	$M_{46}+1, M_4+1$	$M_{14}+1, M_{36}+1$
M_{30}	$M_{15} \quad , M_{37}+2$	$M_{47} \quad , M_5+2$
M_{31}	$M_{47}+1, M_5+1$	$M_{15}+1, M_{37}+1$

metric $M_i(k+1)$	$[z(\alpha)]_k = 0$ minimumvalue	$[z(\alpha)]_k = 1$ minimumvalue
M_{32}	$M_{48} \quad , M_{26}+2$	$M_{16} \quad , M_{58}+2$
M_{33}	$M_{16}+1, M_{58}+1$	$M_{48}+1, M_{26}+1$
M_{34}	$M_{49} \quad , M_{27}+2$	$M_{17} \quad , M_{59}+2$
M_{35}	$M_{17}+1, M_{59}+1$	$M_{49}+1, M_{27}+1$
M_{36}	$M_{18} \quad , M_{56}+2$	$M_{50} \quad , M_{24}+2$
M_{37}	$M_{50}+1, M_{24}+1$	$M_{18}+1, M_{56}+1$
M_{38}	$M_{19} \quad , M_{57}+2$	$M_{51} \quad , M_{25}+2$
M_{39}	$M_{51}+1, M_{25}+1$	$M_{19}+1, M_{57}+1$
M_{40}	$M_{20} \quad , M_{62}+2$	$M_{52} \quad , M_{30}+2$
M_{41}	$M_{52}+1, M_{30}+1$	$M_{20}+1, M_{62}+1$
M_{42}	$M_{21} \quad , M_{63}+2$	$M_{53} \quad , M_{31}+2$
M_{43}	$M_{53}+1, M_{31}+1$	$M_{21}+1, M_{63}+1$
M_{44}	$M_{54} \quad , M_{28}+2$	$M_{22} \quad , M_{60}+2$
M_{45}	$M_{22}+1, M_{60}+1$	$M_{54}+1, M_{28}+1$
M_{46}	$M_{55} \quad , M_{29}+2$	$M_{23} \quad , M_{61}+2$
M_{47}	$M_{23}+1, M_{61}+1$	$M_{55}+1, M_{29}+1$
M_{48}	$M_{56} \quad , M_{18}+2$	$M_{24} \quad , M_{50}+2$
M_{49}	$M_{24}+1, M_{50}+1$	$M_{56}+1, M_{18}+1$
M_{50}	$M_{57} \quad , M_{19}+2$	$M_{25} \quad , M_{51}+2$
M_{51}	$M_{25}+1, M_{51}+1$	$M_{57}+1, M_{19}+1$
M_{52}	$M_{26} \quad , M_{48}+2$	$M_{58} \quad , M_{16}+2$
M_{53}	$M_{58}+1, M_{16}+1$	$M_{26}+1, M_{48}+1$
M_{54}	$M_{27} \quad , M_{49}+2$	$M_{59} \quad , M_{17}+2$
M_{55}	$M_{59}+1, M_{17}+1$	$M_{27}+1, M_{49}+1$
M_{56}	$M_{28} \quad , M_{54}+2$	$M_{60} \quad , M_{22}+2$
M_{57}	$M_{60}+1, M_{22}+1$	$M_{28}+1, M_{54}+1$
M_{58}	$M_{29} \quad , M_{55}+2$	$M_{61} \quad , M_{23}+2$
M_{59}	$M_{61}+1, M_{23}+1$	$M_{29}+1, M_{55}+1$
M_{60}	$M_{62} \quad , M_{20}+2$	$M_{30} \quad , M_{52}+2$
M_{61}	$M_{30}+1, M_{52}+1$	$M_{62}+1, M_{20}+1$
M_{62}	$M_{63} \quad , M_{21}+2$	$M_{31} \quad , M_{53}+2$
M_{63}	$M_{31}+1, M_{53}+1$	$M_{63}+1, M_{21}+1$

$M_1 = M_{21}$	$M_9 = M_{29}$	$M_{33} = M_{53}$	$M_{41} = M_{61}$	$M_{19} = M_{47}$	$M_2 = M_{42}$	$M_{18} = M_{58}$
$M_3 = M_{23}$	$M_{11} = M_{31}$	$M_{35} = M_{55}$	$M_{43} = M_{63}$	$M_{23} = M_{43}$	$M_6 = M_{46}$	$M_{22} = M_{62}$
$M_5 = M_{17}$	$M_{13} = M_{25}$	$M_{37} = M_{49}$	$M_{45} = M_{57}$	$M_{27} = M_{39}$	$M_{10} = M_{34}$	$M_{26} = M_{50}$
$M_7 = M_{19}$	$M_{15} = M_{27}$	$M_{39} = M_{51}$	$M_{47} = M_{59}$	$M_{31} = M_{35}$	$M_{14} = M_{38}$	$M_{30} = M_{54}$

10. FORTRAN-PROGRAM FOR CALCULATION OF THE ESSENTIAL FEATURES AND SYNDROME-DECODING PROCEDURE OF R=½ CONVOLUTIONAL CODES.

Introduction:

All basic calculations are split up into 11 subroutines. In these subroutines the basic parameters of the syndrome-decoding procedure are calculated. In the Fortran language we have strived for a naming of identifiers and variables analogous to the naming used in the theory of the previous sections. A summary of this analogy is given in the Appendix at the pages 1-4.

The program is completely variable; that is, any code C of any constraintlength ν in the class $\mathcal{T}_{\nu,1}$ can be analysed. The user may choose the code with the aid of a table of connectionpolynomials which are factorized in irreducible polynomials. After this he may choose the input-data required for the decoding procedure that he wishes to see (in print or graphical displayed, dependent on the device used; see also section 11).

One restriction on the variability of the program is the memory capacity of the PRIME-200 computer. Another restriction is the number of states (nodes) in vertical sense and the number of sections in horizontal sense in the Trellis which is to be displayed. The last two demands are the strongest ones.

In vertical sense we restrict ourselves to codes up to constraintlength $\nu=4$ with $2^4=16$ states.

When we want to decode we need an encoder, a datasequence and a channel with noise. The datasequence $[x(\alpha)]_0^{L-1}$ of length L and the noise-sequence pair $[n_1(\alpha),n_2(\alpha)]_0^{L-1}$ may be choosen freely. In horizontal sense we restrict ourselves to a maximum decodinglength $\nu+L=16$ when we use the display and a maximum $\nu+L=12$ when we use the teletype.

Strictly, this program is combined with the graphical display program of section 11 into the complete syndrome-decoding program called *SDECO. At the beginning of this program the user is asked to give the value of the variable MODE, which must be 1 when using the teletype and 2 when using the graphical display.

Program:

The mainprogram MAINPR with output OUTPUT and the 11 subroutines are supplied with explanatorial text and extensive comment between the statements. Flow-diagrams have been made and each subroutine is worked out for a particular example. These data are to be found in the Appendix at the pages 5-59.

We will give here a brief summary of the contents of the mainprogram with output and the 11 subroutines:

MAINPR: As input the program asks the code's constraintlength ν, the connectionpolynomials $C_n(\alpha)$, n=1,2, the datasequencelength L, the datasequence $[x(\alpha)]_0^{L-1}$ and the noise-sequence pair $[n_1(\alpha),n_2(\alpha)]_0^{L-1}$. The productsequences $C_1(\alpha)x(\alpha)$ and $C_2(\alpha)x(\alpha)$ are formed and the two linesequences $y_1(\alpha)$ and $y_2(\alpha)$ are constructed. Then the syndromesequence $z(\alpha)=n_2(\alpha)C_1(\alpha)+n_1(\alpha)C_2(\alpha)$ is calculated. The subroutine DPOLYN which calculates the two polynomials $D_n(\alpha)$, n=1,2, is called and the sequence $w(\alpha)= y_1(\alpha)D_1(\alpha)+y_2(\alpha)D_2(\alpha)$ is constructed. The specific outputvalue z_0 is calculated and after this the remaining 10 subroutines are called.

OUTPUT: All the essential productsequences are printed. The polynomials $D_n(\alpha)$, n=1,2, the base state, the value z_0 and the state-table are given. The so-called "parent state matrix" is printed and the decoding scheme with the decoded path is given.

PRODCT: Multiplication of an arbitrary binary sequence (polynomial) with a binary sequence (polynomial) beginning with a "one".

DPOLYN: Euclid's algorithm for the calculation of two polynomials $D_n(\alpha)$, n=1,2 of minimum degree which satisfy $C_1(\alpha)D_1(\alpha)+C_2(\alpha)D_2(\alpha)=1$.

BSTATE: Calculation of the code's base state $S_b=[\alpha^{-1},s_b(\alpha)]$.

STATMX: Calculation of the binary representation of the decimal statenumbers $i=0,1,\ldots,2^{\nu}-1$, where $s_i(\alpha)=\sum_{k=-\nu}^{-1} s_{i,k}\alpha^k$.

ZWNUL: Calculation of the specific outputvalues z_{i0} and w_{i0} for the transitions $S_{\frac{1}{2}i} \xrightarrow{00} S_i$.

TRATBL: Construction of the state-table.

PSTMX: Calculation of the four parent-states of each state S_i and the associated transitions and outputvalues $[z(\alpha)]$ and $[w(\alpha)]$.

TREEMC: Calculation of the so-called "treemetricvalues" $M_i(\nu-1)$ at time $\nu-1$ (depth ν), starting with $M_0(0)=0$ at time 0.

SSMCMB: Determination of the steady state metriccombination $\underline{m}_0(\nu+2\nu-1)$ at time $\nu+2\nu-1$ (depth $\nu+2\nu$), starting with the treemetriccombination $M_i(\nu-1)$, $i=0,..,2^\nu-1$, at time $\nu-1$.

DECOD: Each time k, for each state S_i, the metricvalue $M_i(k)$, the survivor $S_p(k)$, the transition $[n_1(\alpha),n_2(\alpha)]_k(p,i)$ and the associated pathregisterbit $[w(\alpha)]_k(p,i)$ are calculated.

DCPATH: Starting with the state $S_i(\nu+L-1)$ at time $\nu+L-1$, which has the minimum metricvalue $M_i(\nu+L-1)=0$, the decoded path is calculated backwards.

<u>Examples:</u>

1. Code 1 of constraintlength $\nu=3$ with connectionpolynomials $[C_1(\alpha),C_2(\alpha)]=$ [1011,1111].
   The complete output-scheme of the program is shown in Fig. 36 (see also section 11 Fig. 43a/b.).

   With the computer program mentioned in section 7, all metriccombinations of a particular code can be calculated and printed with their mutual transitions for $[z(\alpha)]=0$ or 1. In Fig. 37 all 75 combinations of this code are shown. The decoded path is indicated by a sequence of successive metriccombinations which are connected with each other by arrows.

2. A code of constraintlength $\nu=5$ with connectionpolynomials $[C_1(\alpha),C_2(\alpha)]=$ [100011,100111].
   The output and the decoding scheme for this code are shown in Fig. 38a/b/c.

3. Code 2 of constraintlength $\nu=6$ with connectionpolynomials $[C_1(\alpha),C_2(\alpha)]=$ [1101101,1110001].
   The output and decoding scheme for this code are shown in Fig. 39a/b.

```
NØ=  1
NU=  3
C(1,K)=  1011
C(2,K)=  1111
XL=  8
X(K)=  00110101
NØISE(1,K)=  10000000
NØISE(2,K)=  00011000
D(1,K)=  100
D(2,K)=  111
SB(K)  =  011
ZNUL   =  1
```

INPUT (applies to the lines from NØ= 1 to NØISE(2,K)= 00011000)

```
XC1(K)      =  00101001001
NØISE(1,K)=  10000000000
---------------------------+
Y(1,K)      =  10101001001

XC2(K)      =  00100100011
NØISE(2,K)=  00011000000
---------------------------+
Y(2,K)      =  00111100011

N1C2(K)     =  11110000000
N2C1(K)     =  00010111000
---------------------------+
Z(K)        =  11100111000

Y1D1(K)     =  0010101001001
Y2D2(K)     =  0010110101001
---------------------------+
SØMYD(K)    =  00000111000
```

```
STATE-TABLE:
  0-   0  1  2  3    0  1  0  1    0  1  1  0
  1-   2  3  0  1    1  0  1  0    1  0  0  1
  2-   4  5  6  7    0  1  0  1    1  0  0  1
  3-   6  7  4  5    1  0  1  0    0  1  1  0
  4-   0  1  2  3    1  0  1  0    0  1  1  0
  5-   2  3  0  1    0  1  0  1    1  0  0  1
  6-   4  5  6  7    1  0  1  0    1  0  0  1
  7-   6  7  4  5    0  1  0  1    0  1  1  0

PARENT-STATE MATRIX:
  0-   0  4  1  5    0  1  1  0    0
  1-   0  4  1  5    1  0  0  1    1
  2-   1  5  0  4    1  0  0  1    1
  3-   1  5  0  4    0  1  1  0    0
  4-   2  6  3  7    0  1  1  0    1
  5-   2  6  3  7    1  0  0  1    0
  6-   3  7  2  6    1  0  0  1    0
  7-   3  7  2  6    0  1  1  0    1

TREEMETRIC(I)=  03232545
SSMC(I)       =  03232444
```

DECØDING SCHEME:

```
0    1 400 2 103 2 400 1 000 1 000 0 400 1 400 0 400 0 000 0 000 0 000
3    0 011 2 011 3 011 0 411 1 112 2 011 0 011 2 011 2 411 1 411 2 112
2    2 110 0 110 2 110 0 510 2 510 1 110 1 110 0 110 2 510 1 510 2 013
3    0 002 2 002 3 002 0 402 1 101 2 501 0 002 2 501 2 402 1 402 2 101
2    3 610 2 610 0 610 1 210 0 210 2 610 0 610 1 610 0 210 2 210 1 210
4    2 201 3 201 1 201 2 601 1 601 2 702 1 201 2 201 1 601 3 601 2 601
4    2 300 0 300 2 300 0 700 2 700 1 300 1 300 0 300 2 700 1 700 3 700
4    2 212 3 711 1 212 2 612 1 311 2 711 1 212 2 711 1 612 3 311 2 311
```

DECØDED PATH:

```
Z(K)         =  1    1    1    0    0    1    1    1    0    0    0

WESTIM(K)    =  0    0    1    1    0    0    1    0    0    0    0

STATE(K)     = 0--- 3--- 6--- 4--- 1--- 3--- 6--- 4--- 0--- 0--- 0--- 0---

NESTIM(1,K)  =  1    0    0    0    0    0    0    0    0    0    0
NESTIM(2,K)  =  0    0    0    1    1    0    0    0    0    0    0

SØMYD(K)     = 00000111000
WESTIM(K)    = 00110010000
---------------------------+
XESTIM(K)    = 00110101000

X(K)         = 00110101
```

Fig. 36. Output-scheme code 1, ν=3.

OLD COMB	NEW COMB Z=0	NEW COMB Z=1
03232444	03232444	10203222
00101010	01011101	11011101
01011101	02120111	11110111
23330121	10002202	02222232
02120111	01111212	01211222
12120111	00000101	02221222
02222232	03232323	10102212
01111212	02221222	00000101
00000101	01110111	01010101
02221222	02222323	00101212
22020101	21110111	02220121
10102212	00100010	22021202
02222222	03232323	10101212
01110111	01111212	01111212
00100111	01111111	01011202
00101212	01211121	11011202
02122212	03231222	10100111
02220121	01112313	01212222
10201222	00101010	12022303
22021202	22220121	13230121
01210121	01112212	01112212
01111111	02121212	00000101
01211121	02122212	00001101
03231222	02222323	00201222
10000101	11110111	02020101
01212222	02222222	10001202
10202222	00101010	22022303
22220121	10001202	02222222
12021202	12220121	12220121
12220121	00001202	02222222
00000000	01010101	01010101
02020101	01110111	01210121
13330121	00002202	02222232
11012303	12320121	22120111
00002202	01210121	21010101
12320121	00002202	02222222
21011202	22220121	13130111
13130111	00000101	02221232

OLD COMB	NEW COMB Z=0	NEW COMB Z=1
10203222	00101010	22022303
22022303	23330121	23230121
11011101	12120111	12120111
23230121	10001202	02222232
11110111	00000101	02121212
10002202	11210121	22020101
10001202	11210121	12020101
01211222	02222222	00001202
02121212	02221222	00100111
11210121	00001101	02122212
03232323	03232333	10201222
12020101	11110111	02220121
00001202	01210121	11010101
01010101	01110111	01110111
02222323	03232333	10101212
00001101	01110111	11010101
21110111	10000101	02121212
03232333	03232434	10202222
00100010	01011101	01011101
10101212	00100010	12021202
11010101	11110111	02120111
01011202	02220121	11110111
11011202	12220121	12120111
10100111	00000000	02021202
01112313	02221232	10000101
03232434	03232444	10202222
12022303	13330121	22220121
13230121	00001202	02222232
01112212	02221222	10000101
00201222	01212121	11012303
02021202	02220121	11210121
02221232	02222323	00102212
01212121	02122212	10001101
00102212	01211121	21011202
10001101	11110111	12020101
22120111	10000101	02221222
21010101	21110111	02120111

Fig. 37. Transitions between metriccombinations.

```
NØ= 1
NU= 5
C(1,K)= 100011
C(2,K)= 100111
XL= 6
X(K)       = 111111
NØISE(1,K)= 001000
NØISE(2,K)= 100010
D(1,K)= 11010
D(2,K)= 11001
SB(K) = 01001
ZNUL  = 0
```

Fig. 38a. <u>Outputscheme code ν=5.</u>

```
XC1(K)     = 10000101111
NØISE(1,K)= 00100000000
---------------------------+
Y(1,K)     = 10100101111

XC2(K)     = 10111010111
NØISE(2,K)= 10001000000
---------------------------+
Y(2,K)     = 00110010111

N1C2(K)    = 00111001000
N2C1(K)    = 11001000010
---------------------------+
Z(K)       = 11110001010

Y1D1(K)    = 010011000001001
Y2D2(K)    = 001101111001001
-------------------------------+
SØMYD(K)   = 01111011100
```

STATE-TABLE:

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

PARENT-STATE MATRIX:

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

TREEMETRIC(I)= 035456452374766525547845457498 65
SSMC(I) = 03545545236445552554664545645655

DECØDING SCHEME:

0	11600	31600	41600	31600	3 000	2 000	2 000	11600	1 000	01600	0 000
3	0 011	2 011	3 412	0 412	32012	02012	11611	3 011	31611	2 011	31611
5	2 100	0 100	2 100	2 100	21700	22103	21700	1 100	11700	3 100	21700
4	51711	31711	51711	32112	1 111	3 111	1 111	22112	4 111	31711	3 111
5	41800	21800	01800	11800	2 200	1 200	2 200	21800	1 200	11800	3 200
5	4 612	3 211	1 211	2 211	31811	21811	12212	3 211	21811	2 211	41811
4	3 300	5 300	3 300	4 300	31900	01900	31900	1 300	21900	4 300	31900
5	41911	52312	41911	42312	4 311	1 311	2 712	21911	3 311	42312	4 311
2	31613	52010	42010	12010	1 410	1 410	1 410	02010	2 410	12010	1 410
3	0 002	2 002	3 401	0 401	32001	02001	11602	3 401	31602	2 401	32001
6	4 510	2 113	3 510	0 510	32110	02110	32110	1 510	22110	2 510	32110
4	51702	31702	51702	32101	1 102	3 501	1 102	22101	4 501	32101	3 501
4	32210	32210	21813	22210	4 610	2 610	0 610	32210	1 610	22210	4 610
5	4 601	3 202	1 202	2 202	31802	21802	12201	3 202	22201	2 202	41802
5	4 710	4 710	5 710	3 710	42310	21913	12310	2 710	22310	3 710	42310
5	41902	52301	41902	42301	4 302	1 302	2 701	21902	3 701	42301	4 302
2	32410	42410	42410	32410	1 810	0 810	1 810	22410	0 810	22410	1 810
5	2 801	4 801	41202	21202	42401	22401	32401	11202	22802	21202	32401
5	2 910	0 910	2 910	2 910	22510	22913	22510	1 910	12510	3 910	22510
4	52501	32501	52501	32902	1 901	3 901	1 901	22902	4 901	32501	3 901
6	52610	42610	22610	22610	01010	21010	01010	32610	11010	22610	21010
6	51402	51001	31001	31001	12601	32601	12601	21402	22601	31001	32601
4	31110	51110	31110	41110	32710	02710	32710	11110	22710	41110	32710
5	42701	53102	42701	43102	41101	11101	21502	22701	31101	43102	41101
4	42800	42800	42800	32800	21200	2 803	21200	22800	2 803	22800	21200
5	2 812	41211	41211	21211	42412	22811	32412	11211	22811	21211	32811
6	41300	2 903	31300	01300	32900	02900	32900	11300	22900	21300	32900
4	52512	32512	52512	32911	1 912	31311	1 912	22911	41311	32911	31311
5	43000	43000	43000	43000	21003	31400	21400	13000	21400	23000	31400
6	51411	51411	31012	31012	12612	32612	12612	21411	22612	31411	32612
5	41500	41500	51500	31500	43100	22703	13100	21500	23100	31500	43100
5	42712	53111	42712	43111	41112	11112	21511	22712	31511	43111	41112

DECØDED PATH:

Z(K) = 1 1 1 1 0 0 0 1 0 1 0

WESTIM(K) = 1 0 0 0 0 1 1 1 1 0 0

STATE(K) = 0--- 1--- 2---13---26---21---10---20--- 8---16--- 0--- 0---

NESTIM(1,K) = 0 0 1 0 0 0 0 0 0 0 0
NESTIM(2,K) = 1 0 0 0 1 0 0 0 0 0 0

SØMYD(K) = 01111011100
WESTIM(K) = 10000111100
-------------------------+
XESTIM(K) = 11111100000

X(K) = 111111

Fig. 38b. Outputscheme code ν=5.

```
NØ= 2
NU= 6
C(1,K)= 1101101
C(2,K)= 1110001
XL= 5
X(K)       = 11111
NØISE(1,K)= 10001
NØISE(2,K)= 00100
D(1,K)= 100
D(2,K)= 101
SB(K) = 010101
ZNUL  = 1

XC1(K)     = 11011101001
NØISE(1,K)= 10001000000
---------------------------+
Y(1,K)     = 01010101001

XC2(K)     = 11110011101
NØISE(2,K)= 00100000000
---------------------------+
Y(2,K)     = 11010011101

N1C2(K)    = 10000110111
N2C1(K)    = 00101101100
---------------------------+
Z(K)       = 10101011011

Y1D1(K)    = 0001010101001
Y2D2(K)    = 1110011101001
---------------------------------+
SØMYD(K)  = 11110010000
```

Fig. 39a. Output-scheme code 2, ν=6.

STATE-TABLE:

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

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

PARENT-STATE MATRIX:

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

TREEMETRIC(I)= 0875$34695644879936628556859758477646679267595468659578475664655
SSMC(I)   = 07756346556445665356275565566564666456662675654666565664655 64655

DECØDED PATH:

Z(K)   = 1 0 1 0 1 0 1 1 0 1 1

WESTIM(K) = 0 0 0 0 1 0 1 0 0 0 0

STATE(K)   = 0---21---42---21---42--- 1--- 2--- 4--- 8---16---32--- 0---

NESTIM(1,K) = 1 0 0 0 1 0 0 0 0 0 0
NESTIM(2,K) = 0 0 1 0 0 0 0 0 0 0 0

SØMYD(K)   = 11110010000
WESTIM(K) = 00001010000
---------------------------+
XESTIM(K) = 11111000000

X(K)   = 11111

Fig. 39b. Output-scheme code 2, ν=6.

DECØDING SCHEME:

0	53200	5 000	53200	34203	33200	2 000	03200	23200	2 000	23200	03200
7	0 011	31012	14212	43211	14212	13211	14212	1 011	33211	24212	3 011
7	53300	0 100	23300	0 100	23300	0 100	23300	33300	1 100	33300	43300
5	54312	63311	4 111	43311	5 111	33311	2 111	2 111	33311	4 111	3 111
6	34013	23410	0 210	3 813	0 210	13410	0 210	2 210	2 813	1 210	3 210
3	5 802	54002	5 802	2 201	43401	2 201	3 802	2 802	4 201	53401	3 802
4	4 310	63510	6 310	43510	4 310	43510	3 310	2 310	43510	3 310	4 310
6	43501	34102	5 902	34102	1 902	24102	1 902	2 902	3 301	43501	33501
5	5 400	43600	2 400	53600	3 400	23600	1 400	0 400	43600	2 400	1 400
5	53611	4 411	51412	0 411	53611	0 411	11412	33611	3 411	33611	43611
6	2 500	53700	5 500	33700	2 500	33700	2 500	3 500	43700	4 500	5 500
4	63711	54712	53711	5 511	53711	14712	43711	41512	3 511	23711	43711
4	53810	4 610	43810	34413	43810	3 610	03810	21213	2 610	23810	33810
5	4 601	63801	54402	41202	34402	43801	34402	34402	31202	24402	34402
6	53910	4 710	53910	4 710	43910	0 710	33910	33910	2 710	33910	43910
6	54502	51302	4 701	43901	4 701	31302	3 701	2 701	33901	4 701	44502
5	14000	43403	2 203	1 800	2 203	2 800	24000	24000	0 800	24000	34000
3	5 811	54011	5 811	2 212	43412	2 212	3 811	2 811	4 212	5 811	3 811
5	54100	5 900	54100	4 900	34100	4 900	34100	24100	3 900	24100	34100
6	43512	34111	5 911	34111	1 911	24111	1 911	2 911	34111	4 911	33512
2	51010	54210	51010	14210	31010	14210	23213	21010	34210	41010	23213
7	0 002	31001	14201	43202	14201	13202	14201	1 002	33202	24201	3 002
5	31110	2 113	43313	2 113	43313	2 113	11110	41110	24310	31110	21110
5	54301	63302	4 102	43302	54301	33302	2 102	2 102	33302	44301	3 102
6	31200	44400	41200	14400	31200	14400	23803	01200	14400	21200	21200
5	4 612	61211	54411	41211	34411	41211	34411	34411	31211	24411	34411
5	41300	64500	61300	54500	41300	2 703	41300	31300	34500	31300	21300
6	54511	51311	4 712	43912	4 712	31311	3 712	2 712	33912	44511	44511
6	34610	51410	4 413	41410	44610	31410	3 413	2 413	31410	44610	34610
5	53602	4 402	51401	0 402	51401	0 402	11401	33602	34601	31401	41401
6	4 513	51510	34710	31510	34710	31510	24710	14710	21510	34710	44710
4	61501	54701	53702	54701	51501	14701	41501	41501	34701	23702	43702
6	41600	54800	41600	34800	11600	04800	21600	21600	24800	01600	21600
6	52612	21611	44811	21611	34811	21611	34811	24811	31611	44811	24811
6	21700	54900	51700	34900	21700	34900	21700	31700	44900	41700	51700
4	64911	55912	54911	51711	54911	15912	44911	42712	31711	24911	44911
5	45010	51810	65010	41810	32413	21810	25010	45010	21810	35010	35010
6	51801	42402	45602	42402	45602	32402	45602	41801	12402	35602	31801
6	55110	41910	55110	41910	45110	01910	35110	35110	21910	35110	45110
6	55702	52502	41901	45101	41901	32502	31901	21901	35101	41901	45702
2	45200	52000	45200	42000	55200	22000	25200	45200	22000	35200	25200
6	22011	35211	36212	33012	22011	35211	22011	26212	23012	36212	46212
7	55300	02100	25300	02100	25300	02100	25300	35300	12100	35300	45300
5	56312	65311	42111	45311	52111	35311	22111	22111	35311	42111	32111
6	42210	45410	22210	25410	22210	25410	22210	12210	15410	22210	32210
5	65401	42201	65401	42201	45401	36002	45401	35401	36002	35401	45401
4	42310	65510	62310	45510	42310	45510	32310	22310	45510	32310	42310
6	45501	36102	52902	36102	12902	26102	12902	22902	32301	45501	35501
6	52400	35600	42400	25600	12400	25600	12400	22400	35600	12400	22400
6	51812	42411	45611	42411	45611	32411	45611	45611	12411	35611	31812
5	42500	65700	62500	55700	42500	21903	42500	32500	35700	32500	22500
6	55711	52511	41912	45112	41912	32511	31912	21912	35112	45711	45711
5	45810	42610	55810	52610	31613	24813	45810	35810	32610	21613	25810
6	52601	21602	44802	21602	34802	21602	32601	24802	35801	42601	24802
6	41713	52710	35910	32710	35910	32710	25910	15910	22710	35910	45910
4	62701	55901	54902	55901	52701	15901	42701	42701	35901	24902	44902
6	36000	32800	36000	32800	36000	32800	36000	36000	22800	46000	26000
5	62811	42212	62811	42212	45412	36011	42811	35412	36011	35412	45412
5	56100	52900	56100	42900	36100	42900	36100	26100	32900	26100	36100
6	45512	36111	52911	36111	12911	26111	12911	22911	36111	42911	35512
4	53010	36210	53010	36210	33010	36210	33010	23010	46210	23010	33010
6	22002	53001	36201	33001	22002	33001	22002	26201	23001	36201	46201
5	33110	22113	45313	22113	45313	22113	13110	43110	26310	33110	23110
5	56301	65302	42102	45302	56301	35302	22102	22102	35302	46301	32102

Fig. 93c. Output-scheme code 2, v=6.

Errors made when decoding:

On the basis of code 1 of constraintlength $\nu=2$ with connectionpolynomials $[C_1(\alpha),C_2(\alpha)]=[111,101]$, the decoding procedure with errors will be demonstrated. This code has a free distance $d_{free}=5$. The free distance path has length 3. This path is represented by the sequence:

(0) - - 11 - - (2) —10— (3) —11— (0)

This code must be able to decode $\frac{1}{2}(d_{free}-1)=2$ errors on a linesequence of any length $\geqslant 2$, when the following noisebits are taken zero. When we send the zero codeword $[x(\alpha)]_0^3=[0000]_0^3$ of length L=4 and a noisesequence pair $[n_1(\alpha),n_2(\alpha)]_0^3=[01,10,00,00]_0^3$ with 2 errors in the first three pairs, the decoder must be able to decode correctly. In the decodingscheme of Fig. 40 we notice that no decoding errors have been made.

When we send the the zero codeword with a noise-sequence pair $[n_1(\alpha),n_2(\alpha)]_0^3=[01,10,01,00]_0^3$ with three errors in the first three pairs we notice that the wrong decoding path is choosen and that one decoding error has been made.

In Fig. 41. the decoding scheme in the form of a Trellis is shown. With the received linesequence pair $[01,10,01,00,00,00]_0^5$ corresponds a syndromesequence $[z(\alpha)]_0^5=[100010]_0^5$ and a sequence $[y_1(\alpha)D_1(\alpha)+y_2(\alpha)D_2(\alpha)]_0^5=[110100]_0^5$. At time k=6 the zero state S_0 has the minimum metric $M_6(0)=0$. Hence the algorithm starts calculating backwards from state S_0 till state S_3 at time k=3. Since the survivor of state S_3 at time k=3 is state S_2, the wrong path (0) —10— (3) - - 00 - - (2) —10— (3) - - 00 - - (2) —00— (0) —00— (0)

is decoded.

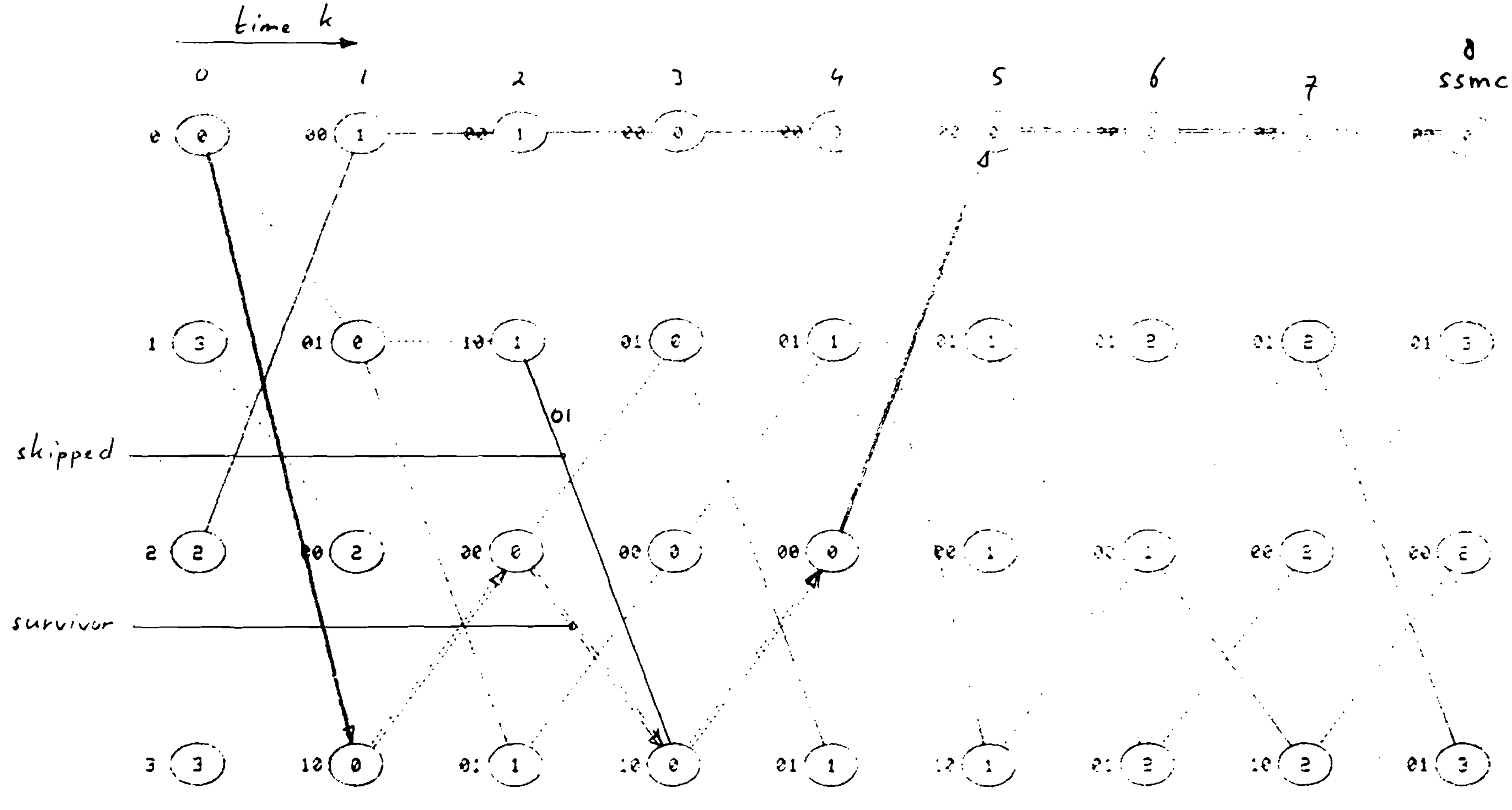
We notice that the first pathregisterbit $[w(\alpha)]_{k=0}$ is decoded erroneously, resulting in one error in the datasequence:

decoded sequence	$[\hat{w}(\alpha)]_0^5=[\ 0\ 1\ 0\ 1\ 0\ 0\ ]_0^5$	
sequence	$[y_1(\alpha)D_1(\alpha)+y_2(\alpha)D_2(\alpha)]_0^5=[\ 1\ 1\ 0\ 1\ 0\ 0\ ]_0^5$	$\oplus$
	$[\hat{x}(\alpha)]_0^3=[\ 1\ 0\ 0\ 0\ ]_0^3$	instead of
	$[x(\alpha)]=[\ 0\ 0\ 0\ 0\ ]_0^3$	

```
NØ= 1
NU= 2
C(1,K)= 111
C(2,K)= 101
XL= 4
X(K)= 0000
NØISE(1,K)= 0100
NØISE(2,K)= 1000
D(1,K)= 10
D(2,K)= 11
SB(K) = 11
ZNUL  = 1

XC1(K)     = 000000
NØISE(1,K)= 010000
---------------------------+
Y(1,K)     = 010000

XC2(K)     = 000000
NØISE(2,K)= 100000
---------------------------+
Y(2,K)     = 100000

N1C2(K)    = 010100
N2C1(K)    = 111000
---------------------------+
Z(K)       = 101100

Y1D1(K)    = 0010000
Y2D2(K)    = 1100000
---------------------------------+
SØMYD(K)   = 111000


DECØDING SCHEME:
0        1 200 1 000 0 200 0 200 0 000 0 000
3        0 011 1 112 2 011 0 011 1 112 1 211
2        2 110 0 310 1 110 1 110 0 310 1 310
3        0 002 1 101 2 301 0 002 1 101 1 202

DECØDED PATH:

Z(K)       =   1    0    1    1    0    0

WESTIM(K)  =   1    1    1    0    0    0

STATE(K)   = 0--- 1--- 1--- 2--- 0--- 0--- 0

NESTIM(1,K) =  0    1    0    0    0    0
NESTIM(2,K) =  1    0    0    0    0    0

SØMYD(K)   = 111000
WESTIM(K)  = 111000
--------------------------+
XESTIM(K)  = 000000

X(K)       = 0000
```

```
NØ= 1
NU= 2
C(1,K)= 111
C(2,K)= 101
XL= 4
X(K)= 0000
NØISE(1,K)= 0100
NØISE(2,K)= 1010
D(1,K)= 10
D(2,K)= 11
SB(K) = 11
ZNUL  = 1

XC1(K)     = 000000
NØISE(1,K)= 010000
---------------------------+
Y(1,K)     = 010000

XC2(K)     = 000000
NØISE(2,K)= 101000
---------------------------+
Y(2,K)     = 101000

N1C2(K)    = 010100
N2C1(K)    = 110110
---------------------------+
Z(K)       = 100010

Y1D1(K)    = 0010000
Y2D2(K)    = 1111000
---------------------------------+
SØMYD(K)   = 110100


DECØDING SCHEME:
0        1 200 1 000 0 000 0 000 0 200 0 000
3        0 011 1 112 0 211 1 211 1 011 2 211
2        2 110 0 310 0 310 0 310 1 110 1 310
3        0 002 1 101 0 202 1 101 1 002 2 101

DECØDED PATH:

Z(K)       =   1    0    0    0    1    0

WESTIM(K)  =   0    1    0    1    0    0

STATE(K)   = 0--- 3--- 2--- 3--- 2--- 0--- 0

NESTIM(1,K) =  1    0    1    0    0    0
NESTIM(2,K) =  0    0    0    0    0    0

SØMYD(K)   = 110100
WESTIM(K)  = 010100
--------------------------+
XESTIM(K)  = 100000

X(K)       = 0000
```

Fig. 40. Decoding-scheme.




```
Z(K)=        1        0        0        0        1        0        0        0

NESTIM(1,K) =  1 0 1 0 0 0 0 0
NESTIM(2,K) =  0 0 0 0 0 0 0 0

UESTIM(K)   =  0 1 0 1 0 0 0 0
SOMYD(K)    =  1 1 0 1 0 0 0 0
------------------------------------------+
XESTIM(K)   =  1 0 0 0 0 0 0 0

Y(K)        =  0 0 0 0 0 0
```

Fig. 41. Trellis decoding-scheme code 1, ν=2.

In the Trellis of Fig. 41 we can draw another path which correspondes to the syndromesequence $[z(\alpha)]_0^5=[100010]_0^5$ and which terminates at state S_0 :

(0) -01- (1) -10- (1) -01- (3) -00- (2) -00- (0) -00- (0)

Note that state S_1 is not the survivor of state S_3 at time k=3, since $M_1(2)+1=2 > M_2(2)+1=1$. Although this path would have been the correct one since the corresponding pathregisterbitsequence is $[1\ 1\ 0\ 1\ 0\ 0]_0^5$ and the estimated noisesequence pair is $[01,10,01,00,00,00]_0^5$ (identical to the channel noise), the decoding algorithm skips this possibility.

There are no other paths possible as can be seen when we penetrate deeper into the Trellis for syndromeoutputs $[z(\alpha)]=0$. Note that already at time k=8 the steady state metriccombination is reached.

---

11. FORTRAN PROGRAM FOR GRAPHICAL DISPLAY OF THE TRELLIS-DECODING PROCEDURE.

Graphical display:

The Terminal Control System TCS Tektronix T-4014-1 is a system capable of displaying both alphanumerical and graphical information. The enhanced graphics module provides for hardware dashed lines, point-plotting and incremental point-plotting. The beam (cursor) of the cathode-ray tube can be controlled to any of the 3120 by 4096 addressable tag-points on the screen in either bright or dark mode; that is visible or not visible.

The TCS is provided with the Fortran software package PLOT-10. This package contains all subroutines by which the computer and the TCS communicate.

Intention:

The aim of the program is didactical. The decoding procedure for convolutional codes is rather untransparent. The decoding procedure can at best be explained on the basis of the Trellis since the Trellis of a code comprises all the essential information for the decoding process. At each time k (section), for each state S_i (node), the survivor, the metric, the associated transition and the pathregisterbit are all known for a specified syndromeoutput. A solid branch from the survivor towards the state S_i corresponds to a pathregisterbit 0 and a dashed branch to a bit 1.

With all this information, at any time k a path in the Trellis can be reconstructed with the minimum distance to the noisesequence pair $[n_1(\alpha),n_2(\alpha)]_0^k$. At time k, we start with the state which has the minimum metric=0 and calculate backwards via the survivors.

The principle of the graphical display program is to visualize the Trellis section by section and to indicate the decoded path.

Program:

The program has been constructed in such a way that the decoding procedure is printed and visualized step by step from the beginning (datasequence at the decoder side) till the end (the decoded path and the estimated datasequence).

We will now give a survey of the course of the graphical display program *SDECO step by step (seperated by means of a carriage return):

- First the value of the variable MODE is asked. When MODE=1 the program discussed in section 10 is executed. The decoding procedure will appear in printed form on the teletype. When MODE=2 the graphical version will be displayed on the TCS.
- MODE=2
- Display of two pages of introductory text for instruction of the user about the course of the program and explanation of the meaning of the output.
- The table of all connectionpolynomials of degree ν=2-6 which are factorized in irreducible polynomials is printed. The user may select a non-catastrohic code with the help of this table.
- The input is asked and all the essential features of the code are printed (see MAINPR+OUTPUT in section 10).
- The complete Trellis is displayed step by step:
  - First all nodes are drawn with in the first section the steady state metriccombination at time 0. The syndromeoutputsequence $[z(\alpha)]_0^{\nu+L-1}$ is printed beneath the Trellis. The dimension of the circles (nodes) has been adapted to the total number of nodes that must be displayed.
  - The sections, representing a transition between two metriccombinations, are drawn step by step (seperated by a carriage return). Each state S_i is connected with its survivor by a solid branch for a pathregisterbit $[w(\alpha)]_k=0$ and a dashed branch for a bit $[w(\alpha)]_k=1$. The metricvalue $M_i(k)$ is printed in the centre of the node and the associated transitions 00,01,10 or 11 is placed at the left of the node (state) S_i.
  - The decoded path is reconstructed from the right to the left, starting at the node S_i with zero metric $M_i(\nu+L-1)=0$. The path is indicated by means of a doubling of the seperate branches.
  - The estimated datasequence $[\hat{x}(\alpha)]_0^{\nu+L-1}$ is evaluated and compared with the original datasequence $[x(\alpha)]_0^{L-1}$.

<u>Subroutines:</u>

The graphical elements of the Trellis are circles (nodes) and dashed or solid lines between the circles (nodes). As the PLOT-10 package does not supply a subroutine for drawing circles, the subroutine CIRCLE(XC,YC,R) has been added. The other subroutines which are needed in the program are TEXT1, FACPOL, TRELLI, EDGCIR and NUMBER.

We will give a brief description of these subroutines together with the subroutines out of the PLOT-10 package which are used in the program.

TEXT1: The introductory text is printed on two pages.

FACPOL: The table of factorized connectionpolynomials with explanatory text is given.

CIRCLE(XC,YC,R): A circle of radius R is drawn with the centre at the coordinates (XC,YC).

TRELLI(TWONU,NUXL,R): The statenumbers 0,1,...,2^{ν}-1, are printed in a column at the left of the screen and the 2^{ν} by ν+L circles with radius R are drawn column by column.

EDGCIR(R,TWONU,NUXL,XD,YD): The begin- and end points of the branches at the edges of the nodes (circles) are calculated and entered in the arrays XD(I) and YD(I). XD(I) and YD(I) are the relative coordinates of the intersection of a branch with the edge of a circle.

NUMBER(TWONU,PLACE): The decimal statenumbers 0,1,...,2^{ν}-1 are printed in the column at the absolute position XC=PLACE.

INITT(N): All parameters are initialized, the screen is erased, the cursor (beam) is directed to the home position (upper left), the mode is switched to alphanumerical and the baudrate is set to N characters/second.

FINITT(IX,IY): Analogue to INITT, but now the cursor is directed to the coordinates (IX,IY).

ANMODE: Placing into alphanumerical mode.

ERASE: Erasure of the screen without change of mode and beamposition.

DWINDO(XMIN,XMAX,YMIN,YMAX): The users rectangle (picture) is defined.

XMIN= the minimum horizontal user coordinate.
XMAX= " maximum " " " .
YMIN= " minimum vertical " " .
YMAX= " maximum " " " .

TWINDO(MINX,MAXX,MINY,MAXY): A rectangular portion of the screen is reserved.

MINX= the minimum horizontal screen coordinate.
MAXX= " maximum " " " .
MINY= " minimum vertical " " .
MAXY= " maximum " " " .

MOVABS(IX,IY): The cursor is directed to the coordinates (IX,IY).

MOVREL(A,B): The cursor is moved a number of (A,B) units relatively to the original coordinates (X,Y).

DASHA(X,Y,L): A L-type line is drawn from the original coordinates to the coordinates (X,Y).
L=0 solid line.
L=1 a dashed line of 5 raster units.

ANCHO(ICHAR): One alphanumerical character is printed at the current position. ICHAR is an integer which represents a 7-bit ASCII character right adjusted.

ANSTR(NCHAR,NADE): The output is a string of NCHAR characters. The ASCII decimal equivalents of the NCHAR characters are entered in the array NADE.

LINWDT(NUMCHR): The width in raster units of NUMCHR characters in the current character size.

LINHGT(NUMLIN): The height in raster units of NUMLIN lines in the current character size.

Listings:

The listing of the complete program called DISPLY together with the subroutines TEXT1, FACPOL, TRELLI, EDGCIR and NUMBER is given in the Appendix at the pages 60-72. The programs are supplied with explanatory text and extensive comment between the statements.

Examples:

1- A complete scheme of the output of the graphical display program is shown in Fig. 42. for the example code 1 of constraintlength $\nu=2$ with connectionpolynomials $[C_1(\alpha),C_2(\alpha)]=[111,101]$.

Fig. 42a Introductory text.

Fig. 42b Table of factorized connectionpolynomials.

Fig. 42c Input + essential features.

Fig. 42d Trellis with the steady state metriccombination and the syndromesequence $[z(\alpha)]_0^{\nu+L-1}$.

Fig. 42e First section for $[z(\alpha)]_0=1$.

Fig. 42f Complete Trellis with the decoded path and the estimated datasequence $[\hat{x}(\alpha)]_0^{\nu+L-1}$.

FACTORIZED CONNECTIONPOLYNOMIALS FOR BINARY R=1/2 CONVO-
LUTIONAL CODES OF CONSTRAINTLENGTH NU=2,3,4,5 AND 6 IN
THE CLASS T(NU,1).
THE NOTATION FOR THESE CONNECTIONPOLYNOMIALS IS:

C(N,K), N=1,2 AND K=1,NU+1   WHERE C(N,1)=C(N,NU+1)=1.

THE POLYNOMIALS ARE OF THE FORM C(N,K)*X**(NU+1-K),K=1,NU+1,
WHERE X IS A FORMAL PARAMETER WHICH SERVES AS A PLACEHOLDER.
****

NU	POLYNOMIAL	FACTORS
2	101	(11)*(11)
	111	IRREDUCIBLE
3	1001	(11)*(111)
	1011	IRREDUCIBLE
	1101	IRREDUCIBLE
	1111	(11)*(11)*(11)
4	10001	(11)*(11)*(11)*(11)
	10011	IRREDUCIBLE
	10101	(111)*(111)
	10111	(11)*(1101)
	11001	IRREDUCIBLE
	11011	(11)*(11)*(111)
	11101	(11)*(1011)
	11111	IRREDUCIBLE
5	100001	(11)*(11111)
	100011	(111)*(1101)
	100101	IRREDUCIBLE
	100111	(11)*(11)*(1011)
	101001	IRREDUCIBLE
	101011	(11)*(11001)
	101101	(11)*(11)*(11)*(111)
	101111	IRREDUCIBLE
	110001	(111)*(1011)
	110011	(11)*(11)*(11)*(11)*(11)
	110101	(11*)*(10011)
	110111	IRREDUCIBLE
	111001	(11)*(11)*(1101)
	111011	IRREDUCIBLE
	111101	IRREDUCIBLE
	111111	(11)*(111)*(111)
6	1000001	(11)*(11)*(111)*(111)
	1000011	IRREDUCIBLE
	1000101	(1011)*(1011)
	1000111	(11)*(111101)
	1001001	IRREDUCIBLE
	1001011	(11)*(11)*(11)*(1101)
	1001101	(11)*(111011)
	1001111	(111)*(11001)
	1010001	(1101)*(1101)
	1010011	(11)*(111)*(1011)
	1010101	(11)*(11)*(11)*(11)*(11)*(11)
	1010111	IRREDUCIBLE
	1011001	(11)*(1101111)
	1011011	IRREDUCIBLE
	1011101	(111)*(11111)
	1011111	(11)*(11)*(10011)
	1100001	IRREDUCIBLE
	1100011	(11)*(11)*(11111)
	1100101	(11)*(111)*(1101)
	1100111	IRREDUCIBLE
	1101001	(11)*(11)*(11)*(1011)
	1101011	(111)*(111)*(111)
	1101101	IRREDUCIBLE
	1101111	(11)*(100101)
	1110001	(11)*(101111)
	1110011	IRREDUCIBLE
	1110101	IRREDUCIBLE
	1110111	(11)*(11)*(11)*(11)*(111)
	1111001	(111)*(10011)
	1111011	(11)*(101001)
	1111101	(11)*(11)*(11001)
	1111111	(1011)*(1101)

****

WHEN YOU HAVE CHOOSEN A PARTICULAR CODE, YOU MAY GIVE
A CARRIAGE RETURN IN ORDER TO PROCEED WITH INPUT

Fig. 42b. Table of factorized connection-polynomials.

```
   NOW GIVE AS INPUT:
 - THE CODES CONSTRAINTLENGTH NU
 - THE CONNECTIONPOLYNOMIALS C(N,K), K=1,2 AND K=1,NU+1
 - THE DATASEQUENCELENGTH XL, WHERE XL >1
 - THE DATASEQUENCE X(K), K=1,XL
 - THE NOISESEQUENCE-PAIR NOISE(N,K), N=1,2 AND K=1,XL

   REMEMBER THAT NU+XL < 16 !!


NU= 2
C(1,K)= 111
C(2,K)= 101
XL= 4
X(K)       = 0000
NOISE(1,K)= 0100
NOISE(2,K)= 1000
D(1,K)= 10
D(2,K)= 11
SB(K) = 11
ZNUL  = 1

XC1(K)     = 000000
NOISE(1,K)= 010000
---------------------------+
Y(1,K)     = 010000

XC2(K)     = 000000
NOISE(2,K)= 100000
---------------------------+
Y(2,K)     = 100000

N1C2(K)    = 010100
N2C1(K)    = 111000
---------------------------+
Z(K)       = 101100

Y1D1(K)    = 0010000
Y2D2(K)    = 1100000
-------------------------------+
SOMYD(K)   = 111000
```

```
STATE-TABLE:
  0-  0  1  2  3   0  1  0  1   0  1  1  0
  1-  2  3  0  1   1  0  1  0   1  0  0  1
  2-  0  1  2  3   1  0  1  0   0  1  1  0
  3-  2  3  0  1   0  1  0  1   1  0  0  1

PARENT-STATE MATRIX:
  0-  0  2  1  3   0  1  1  2   0
  1-  0  2  1  3   1  0  0  1   1
  2-  1  3  0  2   1  0  0  1   1
  3-  1  3  0  2   0  1  1  0   0

TREEMETRIC(I)= 0323
SSMC(I)      = 0323


*******************************************************


   BY GIVING A CARRIAGE RETURN, THE TRELLIS IS DRAWN
WITH IN THE FIRST ROW THE STEADY STATE METRICCOMBINA-
TION.

   NOW,BY GIVING A CARRIAGE RETURN AT EACH TIME K, THE
K TH SECTION WITH ITS DECODING DATA WILL BE SHOWN.
```

Fig. 42c. Input + essential features.

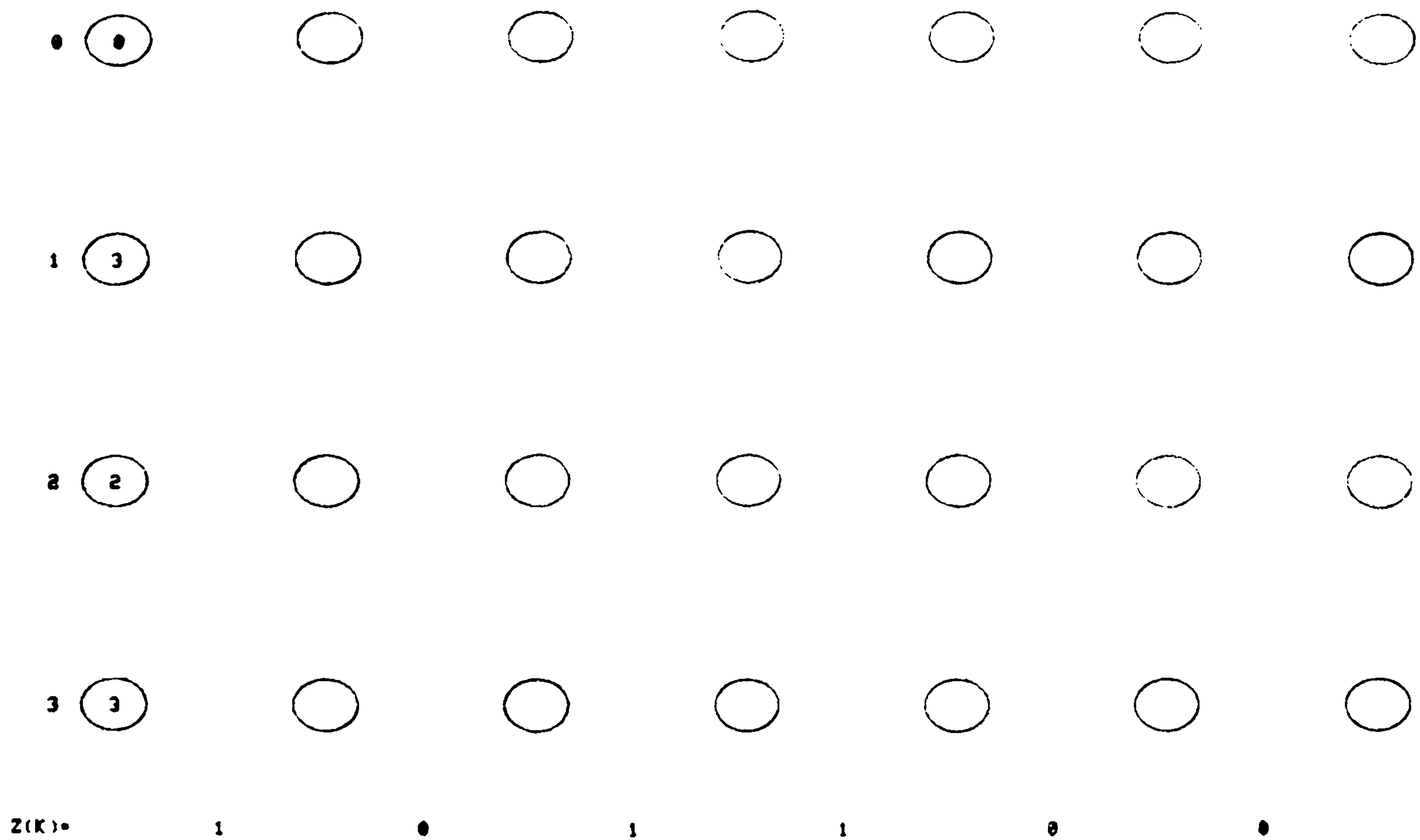


Fig. 42d. Trellis with the steady state metric-combination and syndromesequence.

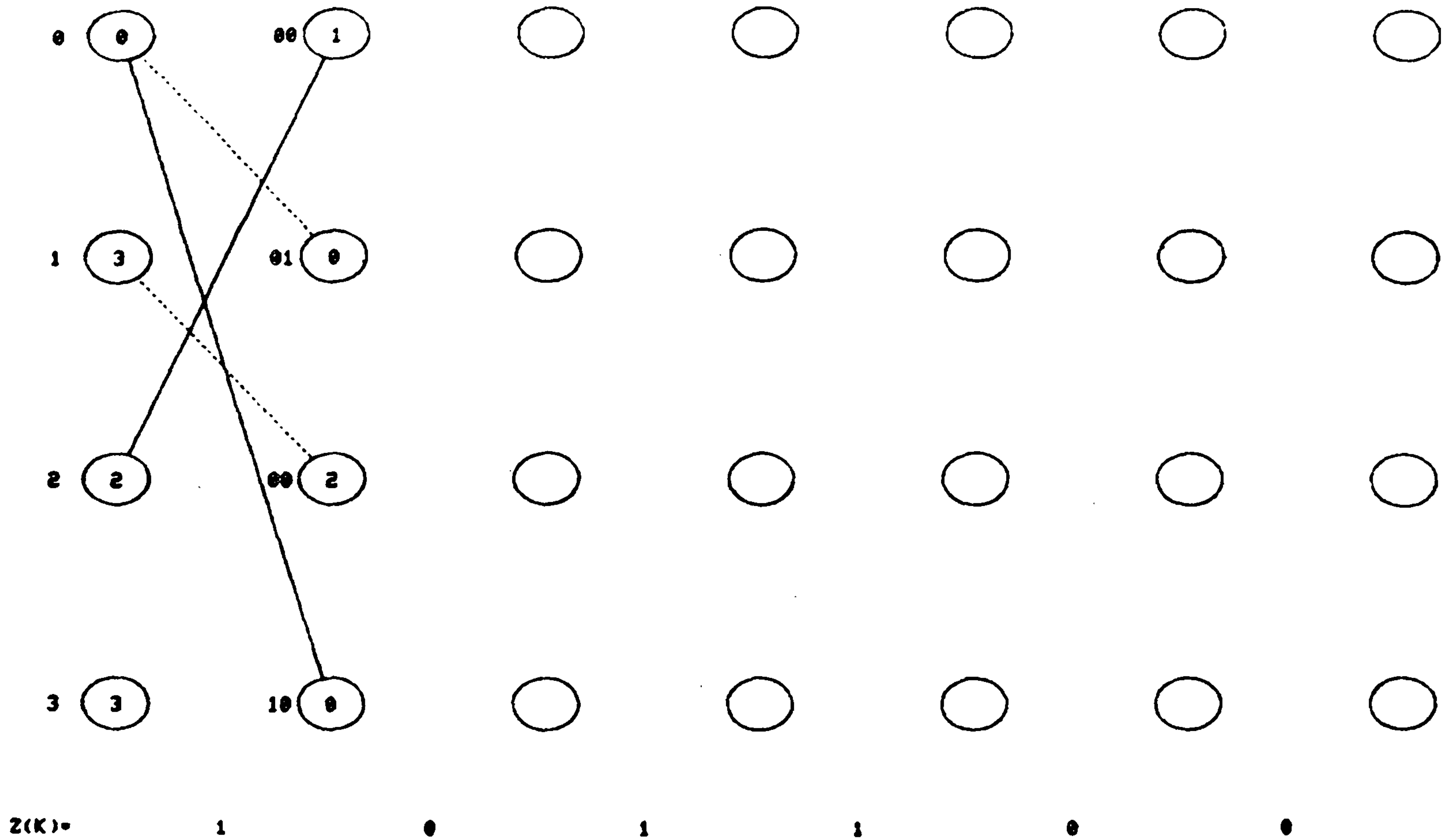
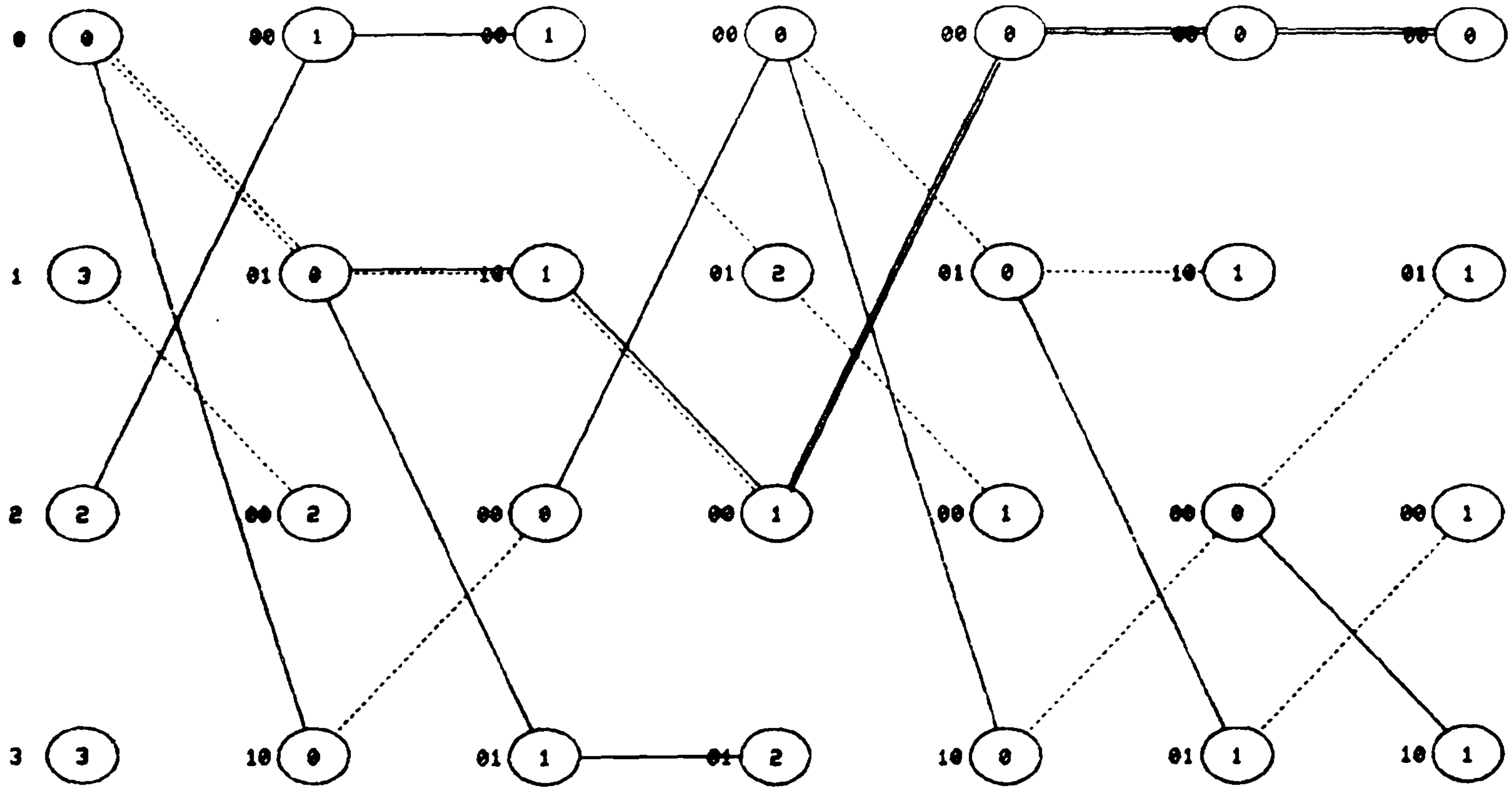


Fig. 42e. First section for $[z(\alpha)]_0=1$.




```
Z(K)=           1           0           1           1           0           0

NESTIM(1,K) =  0 1 0 0 0 0
NESTIM(2,K) =  1 0 0 0 0 0

WESTIM(K)   =  1 1 1 0 0 0
SOMYD(K)    =  1 1 1 0 0 0
-----------------------------------------------+
XESTIM(K)   =  0 0 0 0 0 0

Y(K)        =  0 0 0 0
```

Fig. 42f. Complete Trellis with the decoded path and estimated datasequence.

2- Code 1 of constraintlength $\nu=3$ with connectionpolynomials $[C_1(\alpha),C_2(\alpha)]=$ [1011,1111]. The essential features and the Trellis are shown in the Figure 43a/b.

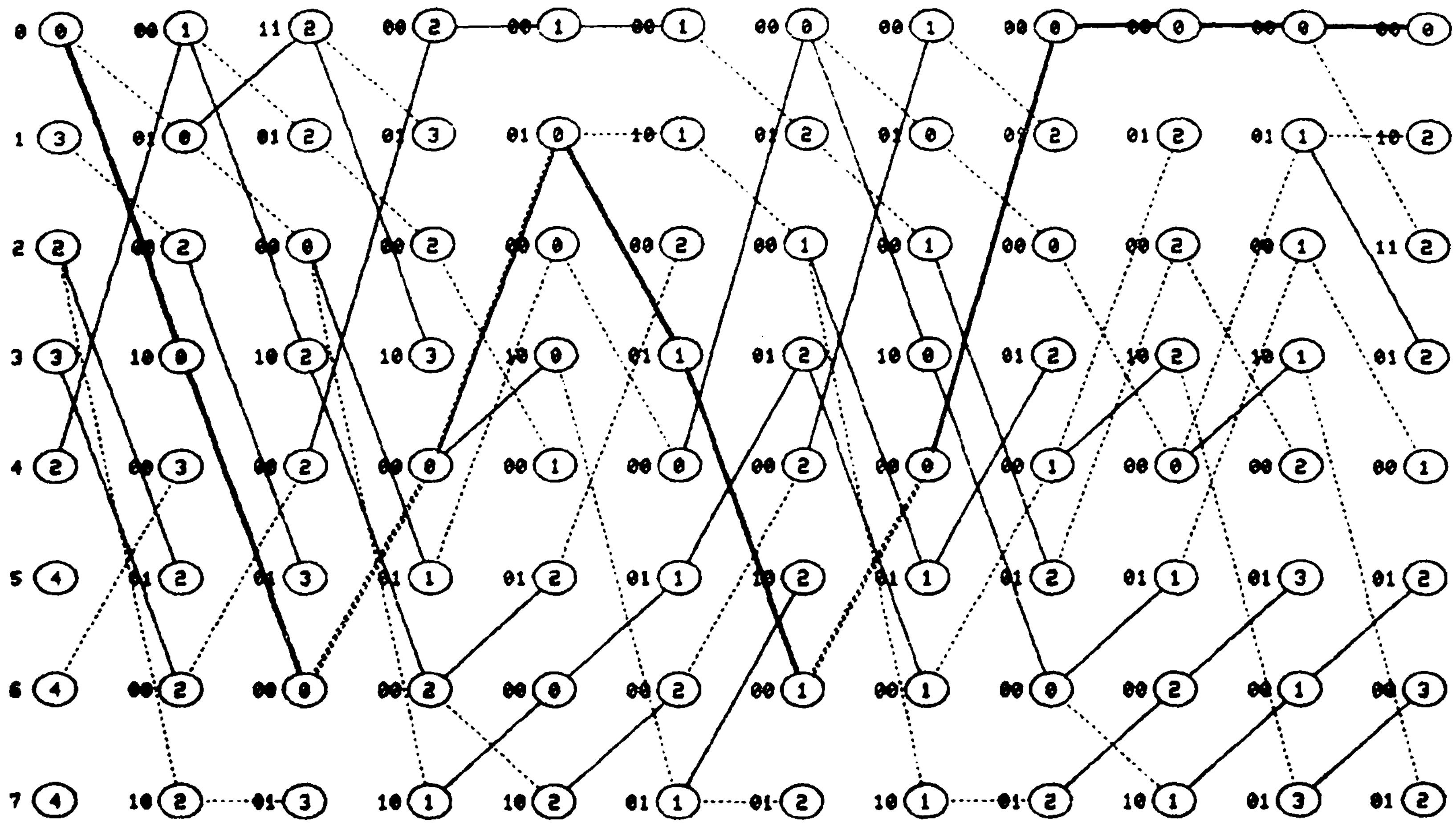
The minimum number of nodes in the Trellis is represented by code 1 of constraintlength $\nu=2$ with a datasequence of length L=2. This example has $2^2=4$ by $\nu+L+1=5$ nodes (Fig. 44a/b).

The maximum number of nodes is represented by code 1 of constraintlength $\nu=4$ with a datasequence of length L=10. This example has $2^4=16$ by $\nu+L+1=15$ nodes (Fig.45a/b).

```
   NOW GIVE AS INPUT:
 - THE CODES CONSTRAINTLENGTH NU
 - THE CONNECTIONPOLYNOMIALS C(N,K), K=1,2 AND K=1,NU+1
 - THE DATASEQUENCELENGTH XL, WHERE XL >1
 - THE DATASEQUENCE X(K), K=1,XL
 - THE NOISESEQUENCE-PAIR NOISE(N,K), N=1,2 AND K=1,XL

   REMEMBER THAT NU+XL < 16 !!


NU= 3
C(1,K)= 1011
C(2,K)= 1111
XL= 8
X(K)       = 00110101
NOISE(1,K)= 10000000
NOISE(2,K)= 00011000
D(1,K)= 100
D(2,K)= 111
SB(K) = 011
ZNUL  = 1

XC1(K)     = 00101001001
NOISE(1,K)= 10000000000
---------------------------+
Y(1,K)     = 10101001001

XC2(K)     = 00100100011
NOISE(2,K)= 00011000000
---------------------------+
Y(2,K)     = 00111100011

N1C2(K)    = 11110000000
N2C1(K)    = 00010111000
---------------------------+
Z(K)       = 11100111000

Y1D1(K)    = 0010101001001
Y2D2(K)    = 0010110101001
---------------------------------+
SOMVD(K)   = 00000111000
```

```
STATE-TABLE:
  0-  0  1  2  3    0  1  0  1    0  1  1  0
  1-  2  3  0  1    1  0  1  0    1  0  0  1
  2-  4  5  6  7    0  1  0  1    1  0  0  1
  3-  6  7  4  5    1  0  1  0    0  1  1  0
  4-  0  1  2  3    1  0  1  0    0  1  1  0
  5-  2  3  0  1    0  1  0  1    1  0  0  1
  6-  4  5  6  7    1  0  1  0    1  0  0  1
  7-  6  7  4  5    0  1  0  1    0  1  1  0

PARENT-STATE MATRIX:
  0-  0  4  1  5    0  1  1  0    0
  1-  0  4  1  5    1  0  0  1    1
  2-  1  5  0  4    1  0  0  1    1
  3-  1  5  0  4    0  1  1  0    0
  4-  2  6  3  7    0  1  1  0    1
  5-  2  6  3  7    1  0  0  1    0
  6-  3  7  2  6    1  0  0  1    0
  7-  3  7  2  6    0  1  1  0    1

TREEMETRIC(I)= 03232545
SSMC(I)       = 03232444


***************************************************


   BY GIVING A CARRIAGE RETURN, THE TRELLIS IS DRAWN
WITH IN THE FIRST ROW THE STEADY STATE METRICCOMBINA-
TION.

   NOW,BY GIVING A CARRIAGE RETURN AT EACH TIME K, THE
K TH SECTION WITH ITS DECODING DATA WILL BE SHOWN.
```

Fig. 43a. Input + essential features code 1, ν=3.



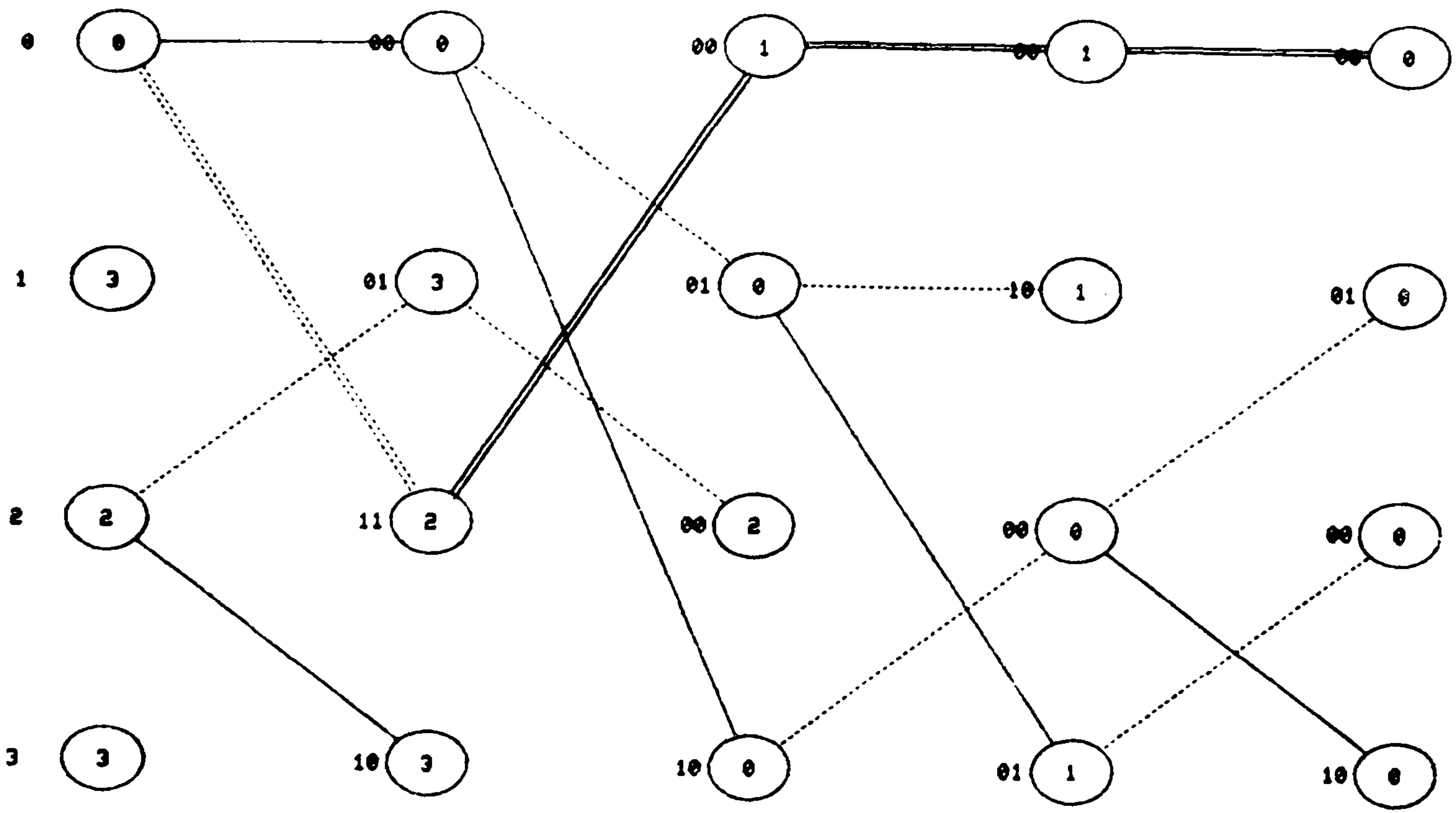
```
Z(K)=    1     1     1     0     0     1     1     1     0     0     0

NESTIM(1,K) =  1 0 0 0 0 0 0 0 0 0 0
NESTIM(2,K) =  0 0 0 1 1 0 0 0 0 0 0

WESTIM(K)   =  0 0 1 1 0 0 1 0 0 0 0
SOMYD(K)    =  0 0 0 0 0 1 1 1 0 0 0
-----------------------------------------+
XESTIM(K)   =  0 0 1 1 0 1 0 1 0 0 0

X(K)        =  0 0 1 1 0 1 0 1
```

Fig. 43b. Trellis of code 1, ν=3.

```
   NOW GIVE AS INPUT:
- THE CODES CONSTRAINTLENGTH NU
- THE CONNECTIONPOLYNOMIALS C(N,K), K=1,2 AND K=1,NU+1
- THE DATASEQUENCELENGTH XL, WHERE XL >1
- THE DATASEQUENCE X(K), K=1,XL
- THE NOISESEQUENCE-PAIR NOISE(N,K), N=1,2 AND K=1,XL

   REMEMBER THAT NU+XL < 16 !!


NU= 2
C(1,K)= 111
C(2,K)= 101
XL= 2
X(K)         = 11
NOISE(1,K)= 10
NOISE(2,K)= 10
D(1,K)= 10
D(2,K)= 11
SB(K) = 11
ZNUL  = 1

XC1(K)       = 1001
NOISE(1,K)= 1000
----------------------------+
Y(1,K)       = 0001

XC2(K)       = 1111
NOISE(2,K)= 1000
----------------------------+
Y(2,K)       = 0111

N1C2(K)      = 1010
N2C1(K)      = 1110
----------------------------+
Z(K)         = 0100

Y1D1(K)      = 00001
Y2D2(K)      = 01001
-----------------------------------+
SOMYD(K)     = 0100
```

```
STATE-TABLE:
  0-  0  1  2  3   0  1  0  :   0  1  1  0
  1-  2  3  0  1   1  0  1  0   1  0  0  1
  2-  0  1  2  3   1  0  1  0   0  1  1  0
  3-  2  3  0  1   0  1  0  1   1  0  0  1

PARENT-STATE MATRIX:
  0-  0  2  1  3   0  1  1  0   0
  1-  0  2  1  3   1  0  0  1   1
  2-  1  3  0  2   1  0  0  1   1
  3-  1  3  0  2   0  1  1  0   0

TREEMETRIC(I)= 0323
SSMC(I)      = 0323


*****************************************************


   BY GIVING A CARRIAGE RETURN, THE TRELLIS IS DRAWN
WITH IN THE FIRST ROW THE STEADY STATE METRICCOMBINA-
TION.

   NOW,BY GIVING A CARRIAGE RETURN AT EACH TIME K, THE
K TH SECTION WITH ITS DECODING DATA WILL BE SHOWN.
```

Fig. 44a. Code 1, ν=2.




```
Z(K)=          0          1          0          0

NESTIM(1,K) =  1 0 0 0
NESTIM(2,K) =  1 0 0 0

WESTIM(K)   =  1 0 0 0
SOMYD(K)    =  0 1 0 0
-----------------------------------------------+
XESTIM(K)   =  1 1 0 0

X(K)        =  1 1
```

Fig. 44b. Code 1, ν=2.

```
   NOW GIVE AS INPUT:
 - THE CODES CONSTRAINTLENGTH NU
 - THE CONNECTIONPOLYNOMIALS C(N,K), K=1,2 AND K=1,NU+1
 - THE DATASEQUENCELENGTH XL, WHERE XL >1
 - THE DATASEQUENCE X(K), K=1,XL
 - THE NOISESEQUENCE-PAIR NOISE(N,K), N=1,2 AND K=1,XL

   REMEMBER THAT NU+XL < 16 !!
```

```
NU= 4
C(1,K)= 10011
C(2,K)= 10111
XL= 10
X(K)       = 1111111111
NOISE(1,K)= 0000100001
NOISE(2,K)= 1000001000
D(1,K)= 1110
D(2,K)= 1101
SB(K) = 0101
ZNUL  = 0

XC1(K)     = 10001111110111
NOISE(1,K)= 00001000010000
--------------------------+
Y(1,K)     = 10000111100111

XC2(K)     = 10110000001011
NOISE(2,K)= 10000010000000
--------------------------+
Y(2,K)     = 00110010001011

N1C2(K)    = 00001110111101
N2C1(K)    = 11001011001000
--------------------------+
Z(K)       = 11000101110101

Y1D1(K)    = 01110010110110101
Y2D2(K)    = 00111000111000101
---------------------------------+
SOMYD(K)   = 01001010001110
```

Fig. 45a. Code 1, $\nu=4$.

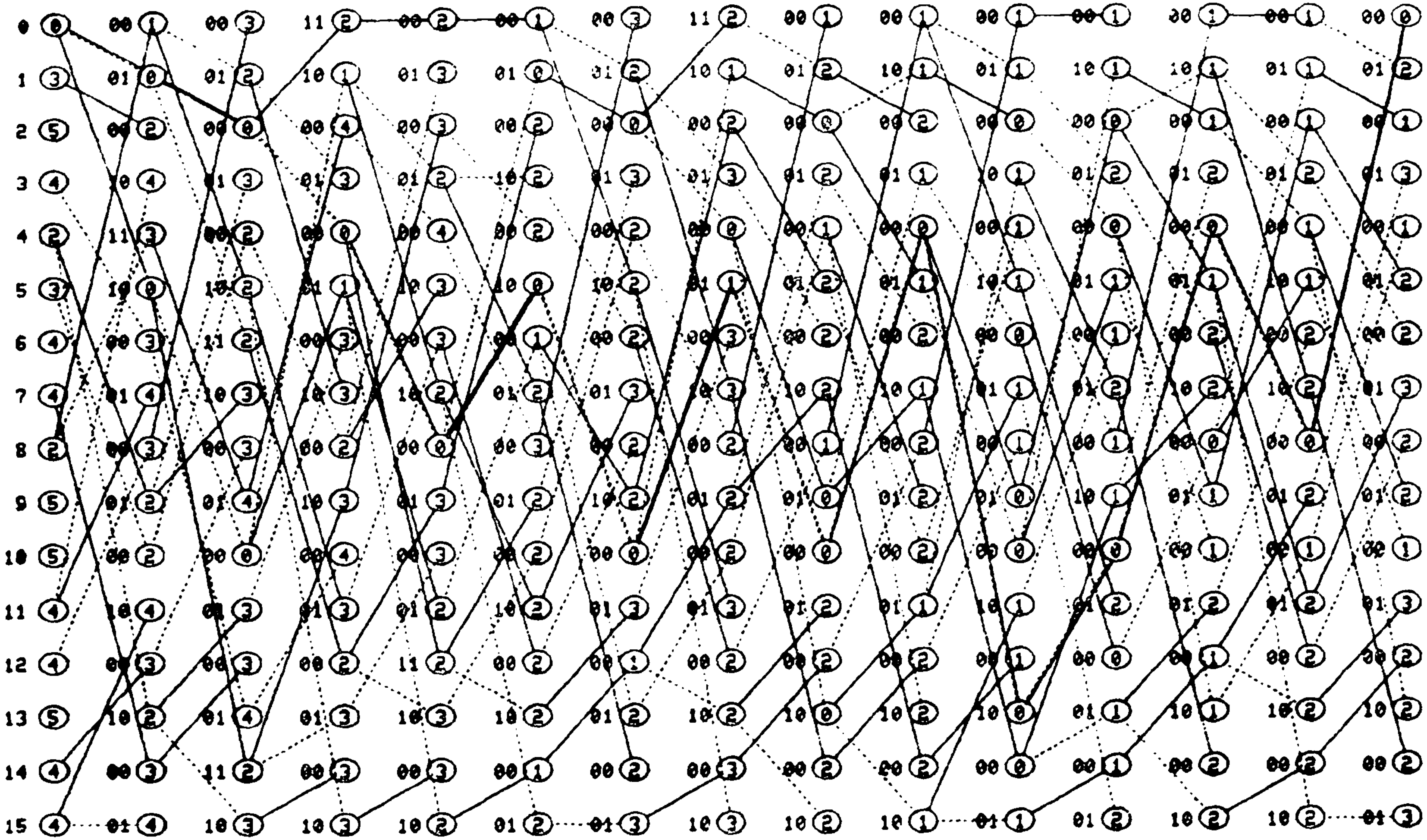
STATE-TABLE:

0-	0	1	4	5	0	1	0	1	0	1	1	0
1-	2	3	6	7	1	0	1	0	0	1	1	0
2-	4	5	0	1	0	1	0	1	1	0	0	1
3-	6	7	2	3	1	0	1	0	1	0	0	1
4-	8	9	12	13	0	1	0	1	1	0	0	1
5-	10	11	14	15	1	0	1	0	1	0	0	1
6-	12	13	8	9	0	1	0	1	0	1	1	0
7-	14	15	10	11	1	0	1	0	0	1	1	0
8-	0	1	4	5	1	0	1	0	0	1	1	0
9-	2	3	6	7	0	1	0	1	0	1	1	0
10-	4	5	0	1	1	0	1	0	1	0	0	1
11-	6	7	2	3	0	1	0	1	1	0	0	1
12-	8	9	12	13	1	0	1	0	1	0	0	1
13-	10	11	14	15	0	1	0	1	1	0	0	1
14-	12	13	8	9	1	0	1	0	0	1	1	0
15-	14	15	10	11	0	1	0	1	0	1	1	0

PARENT-STATE MATRIX:

0-	0	8	2	10	0	1	0	1	0
1-	0	8	2	10	1	0	1	0	1
2-	1	9	3	11	1	0	1	0	0
3-	1	9	3	11	0	1	0	1	1
4-	2	10	0	8	0	1	0	1	1
5-	2	10	0	8	1	0	1	0	0
6-	3	11	1	9	1	0	1	0	1
7-	3	11	1	9	0	1	0	1	0
8-	4	12	6	14	0	1	0	1	1
9-	4	12	6	14	1	0	1	0	0
10-	5	13	7	15	1	0	1	0	1
11-	5	13	7	15	0	1	0	1	0
12-	6	14	4	12	0	1	0	1	0
13-	6	14	4	12	1	0	1	0	1
14-	7	15	5	13	1	0	1	0	0
15-	7	15	5	13	0	1	0	1	1

```
TREEMETRIC(I)= 0354237425544574
SSMC(I)      = 0354234425544544


**********************************************************


   BY GIVING A CARRIAGE RETURN, THE TRELLIS IS DRAWN
WITH IN THE FIRST ROW THE STEADY STATE METRICCOMBINA-
TION.

   NOW,BY GIVING A CARRIAGE RETURN AT EACH TIME K, THE
K TH SECTION WITH ITS DECODING DATA WILL BE SHOWN.
```



```
Z(K)=  1   1   0   0   0   1   0   1   1   1   0   1   0   1

NESTIM(1,K) =  0 0 0 0 1 0 0 0 0 1 0 0 0 0
NESTIM(2,K) =  1 0 0 0 0 0 1 0 0 0 0 0 0 0

WESTIM(K)   =  1 0 1 1 0 1 0 1 1 1 1 1 1 0
SOMYD(K)    =  0 1 0 0 1 0 1 0 0 0 1 1 1 0
------------------------------------------->
XESTIM(K)   =  1 1 1 1 1 1 1 1 1 1 0 0 0 0

(K)         =  1 1 1 1 1 1 1 1 1 1
```

Fig. 45b. Code 1, ν=4.

12. CONCLUSION.

The essential features of binary $R=\frac{1}{2}$ convolutional codes in the class $\mathcal{T}_{\nu,1}$ have been calculated. A comparison between the characteristics of the codes has been made.

From a practical point of view, the most important result following from the analysis of these codes is the formula for the reduction in pathregisters. A code can be choosen which requires a minimum of hardware when implemented as decoder. The remaining results might be of interest in the further development of the theory of convolutional codes.

The graphical display program makes the rather complex decoding procedure of convolutional codes much more transparent. Even for the interested reader the decoding procedure becomes accessible.

The program is completely variable in that sense that binary $R=\frac{1}{2}$ convolutional codes of any constraintlength ν in the class $\mathcal{T}_{\nu,1}$ can be analysed. With some modifications, which would not be all to radical, the program could be changed in such a way that other binary $R=\frac{1}{2}$ convolutional codes can be analysed, even convolutional codes of rates $R\neq\frac{1}{2}$.

The program also opens the possibility of complex theoretical calculations on convolutional codes on the computer.

References.

1 - Berlekamp, E.R., Algebraic Coding Theory, chapter 2, pp 36-44, Mc. Grawhill, N.Y., 1968.

2 - Viterbi, A.J., Convolutional codes and their performance in communication systems, IEEE Trans. on Communication Technology, vol. COM-19, pp 751-772, oct. 1971.

3 - Schalkwijk, J.P.M., Symmetries of the state diagram of the syndromeformer of a binary rate-½ convolutional code, Advanced School on "New trends and open problems in Information Theory", C.I.S.M., 33100 Udine, Italy, sept. 1975.

4 - Schalkwijk, J.P.M./Vinck, A.J., Syndrome decoding of convolutional codes, IEEE Trans. on Communications, vol. COM-23, pp 789-792, july 1975.

5 - Vinck, A.J., Computer program for the calculation of metriccombinations, unpublished results, Technical University Eindhoven.

6 - Wijk, J.G. van, Computer-search program for best rate-½ convolutional codes, unpublished results, Technical University Eindhoven.

7 - Carnahan, B./Wilkes, J.O., Digital Computing and Numerical Methods, J. Wiley & Sons Inc., N.Y., 1973.

8 - PLOT-10, Fortran software package for the Terminal Control System T-4014-1, Tektronix Inc., P.O. Box 500, Beaverton, Oregon 97005.

9 - Computer Display Terminal T-4014-1, Instruction Manual, Tektronix Inc., P.O. Box 500, Beaverton, Oregon 97005.

10 - Schalkwijk, J.P.M./Vinck, A.J./Rust, L.J.A.E., Analysis and simulation of a syndrome decoder for a constraintlength k=5, rate-½ binary convolutional code, T-H Report 75-E-58, Technical University Eindhoven, april 1975.

Appendix.

Contents: page

Fortran- notation of the variables used in the theory:

Variable	Fortran version
class $\mathcal{T}_{\nu,1}$	class T(NU,1)
constraintlength ν	NU
connectionpolynomials $C_n(\alpha) = \sum_{i=0}^{\nu} c_{n,i}\, \alpha^i$, n=1,2	C(N,K), N=1,2 K=1,NU+1
degree of the $D_n(\alpha)$-polynomials μ	MU
polynomials $D_n(\alpha) = \sum_{i=0}^{\nu} d_{n,i}\, \alpha^i$,n=1,2	D(N,K), N=1,2 K=1,MU+1
2^{ν}	TWONU
$2^{(\nu-1)}$	TWONU1
length of the datasequence $[x(\alpha)]_{k=0}^{k=L-1}$	XL
datasequence $[x(\alpha)]_{k=0}^{k=L-1}$	X1(1,K), K=1,XL
som $\nu + L$	NUXL=NU+XL
som $\nu + \mu + L$	NUMUXL=NU+MU+XL
noisesequence pair $[n_1(\alpha),n_2(\alpha)]_{k=0}^{k=L-1}$	NOISE(N,K), N=1,2 K=1,XL
product $[C_1(\alpha)x(\alpha)]_{k=0}^{k=NU+XL-1}$	XC1(K), K=1,NUXL
product $[C_2(\alpha)x(\alpha)]_{k=0}^{k=NU+XL-1}$	XC2(K), K=1,NUXL
linesequence $[y_1(\alpha)=C_1(\alpha)x(\alpha)+n_1(\alpha)]_{k=0}^{k=NUXL-1}$	Y(1,K), K=1,NUXL
linesequence $[y_2(\alpha)=C_2(\alpha)x(\alpha)+n_2(\alpha)]_{k=0}^{k=NUXL-1}$	Y(2,K), K=1,NUXL
product $[y_1(\alpha)D_1(\alpha)]_{k=0}^{k=NUMUXL-1}$	Y1D1(K), K=1,NUMUXL
product $[y_2(\alpha)D_2(\alpha)]_{k=0}^{k=NUMUXL-1}$	Y2D2(K), K=1,NUMUXL
som $[y_1(\alpha)D_1(\alpha)+y_2(\alpha)D_2(\alpha)]_{k=0}^{k=NUMUXL-1}$	SOMYD(K), K=1,NUMUXL

product $[n_1(\alpha)C_2(\alpha)]_{k=0}^{k=NUXL-1}$	N1C2(K), K=1,NUXL
product $[n_2(\alpha)C_1(\alpha)]_{k=0}^{k=NUXL-1}$	N2C1(K), K=1,NUXL
syndromesequence $[z(\alpha)]_{k=0}^{k=NUXL-1}$	Z(K), K=1,NUXL
state $S_i = [\ 0\ ,\ s_i(\alpha)\ ]$ where	S(I,J), J=1,NU
$s_i(\alpha) = \sum_{k=-\nu}^{-1} s_{i,k}\ \alpha^k$	
statenumber $i = \sum_{k=-\nu}^{-1} s_{i,k}\ 2^{-(k+1)}$, $i=0,2^\nu-1$	I=1,TWONU
base-state $S_b = [\ 0\ ,\ s_b(\alpha)\ ]$	SB(K), K=1,NU
base-state number b	NRB
specific outputvalue z_0	ZNUL
outputvalues z_{i0} and w_{i0} for the transition	ZNUL2I(I) and WNUL2I(I)
$S_{\frac{1}{2}i} \xrightarrow{[00]} S_i$ of the syndrome- and $w(\alpha)$-former	for $S((I+1)/2) \xrightarrow{[00]} S(I)$
with $i=2k$, $k=0,2^{\nu-1}-1$	with I=2K-1 , K=1,TWONU1

<u>state transitions:</u> $i=0,2^\nu-1$	I=1,TWONU
$S_i \xrightarrow{[0,0]} S_{2i}$	$TT(I,1) \xrightarrow{[0,0]} TT(I,2)$
$S_i \xrightarrow{[0,1]} S_{2i+1}$	$TT(I,1) \xrightarrow{[0,1]} TT(I,3)$
$S_i \xrightarrow{[1,1]} S'_{2i+1} \oplus S_b$	$TT(I,1) \xrightarrow{[1,1]} TT(I,4)$
$S_i \xrightarrow{[1,0]} S'_{2i} \oplus S_b$	$TT(I,1) \xrightarrow{[1,0]} TT(I,5)$

<u>parent-states with their transitions and outputs $[z(\alpha)]_k$ and $[w(\alpha)]_k$:</u>

<u>parent-states:</u> $i=2k,\ k=0,2^{\nu-1}-1$	I=2K-1, K=1,TWONU1
$S_{p1} = S_{\frac{1}{2}i}$	TRANMX(I,1) = TRANMX(I+1,1)
$S_{p2} = S_{\frac{1}{2}i+2^{\nu-1}}$	TRANMX(I,2) = TRANMX(I+1,2)
$S_{p3} = S_{\frac{1}{2}i} \oplus S_{\frac{1}{2}(b-1)}$	TRANMX(I,3) = TRANMX(I+1,3)
$S_{p4} = S_{\frac{1}{2}i+2^{\nu-1}} \oplus S_{\frac{1}{2}(b-1)}$	TRANMX(I,4) = TRANMX(I+1,4)

<u>transitions:</u>

$S_{p1} \xrightarrow{0,0} S_i$	$S_{p1} \xrightarrow{0,1} S_{i+1}$	TRANMX(I,1) $\underline{0,0}$ S(I,J)	TRANMX(I+1,1) $\underline{0,1}$ S(I+1,J)
$S_{p2} \xrightarrow{0,0} S_i$	$S_{p2} \xrightarrow{0,1} S_{i+1}$	TRANMX(I,2) $\underline{0,0}$ S(I,J)	TRANMX(I+1,2) $\underline{0,1}$ S(I+1,J)
$S_{p3} \xrightarrow{1,1} S_i$	$S_{p3} \xrightarrow{1,0} S_{i+1}$	TRANMX(I,3) $\underline{1,1}$ S(I,J)	TRANMX(I+1,J) $\underline{1,0}$ S(I+1,J)
$S_{p4} \xrightarrow{1,1} S_i$	$S_{p4} \xrightarrow{1,0} S_{i+1}$	TRANMX(I,4) $\underline{1,1}$ S(I,J)	TRANMX(I+1,J) $\underline{1,0}$ S(I+1,J)

<u>outputvalues:</u>	$d_{1,0}=0$	$d_{1,0}=1$		
$S_{p1} \xrightarrow{0,0} S_i \quad z_{i0}$	w_{i0}	w_{i0}	TRANMX(I,5)=ZNUL2I(I)	TRANMX(I,9)
$S_{p2} \xrightarrow{0,0} S_i \quad z_{i0}\oplus 1$	w_{i0}	w_{i0}	TRANMX(I,6)=1-TRANMX(I,5)	"
$S_{p3} \xrightarrow{1,1} S_i \quad z_{i0}\oplus z_0$	w_{i0}	w_{i0}	TRANMX(I,7)=ZNUL2I(I)⊕ZNUL	"
$S_{p4} \xrightarrow{1,1} S_i \quad z_{i0}\oplus 1\oplus z_0$	w_{i0}	w_{i0}	TRANMX(I,8)=1-TRANMX(I,7)	"
$S_{p1} \xrightarrow{0,1} S_{i+1} \quad \bar{z}_{i0}$	$\bar{w}_{i0}$	w_{i0}	TRANMX(I+1,5)=1-TRANMX(I,5)	TRANMX(I+1,9)
$S_{p2} \xrightarrow{0,1} S_{i+1} \quad \bar{z}_{i0}\oplus 1$	$\bar{w}_{i0}$	w_{i0}	TRANMX(I+1,6)=1-TRANMX(I,6)	"
$S_{p3} \xrightarrow{1,0} S_{i+1} \quad \bar{z}_{i0}\oplus z_0$	$\bar{w}_{i0}$	w_{i0}	TRANMX(I+1,7)=1-TRANMX(I,7)	"
$S_{p4} \xrightarrow{1,0} S_{i+1} \quad \bar{z}_{i0}\oplus 1\oplus z_0$	$\bar{w}_{i0}$	w_{i0}	TRANMX(I+1,8)=1-TRANMX(I,8)	"

metricvalue of state S_i at time k: $M_i(k)$ MTCOMB(I,K)

metric-combination at time k:

$\underline{m}_c(k)=\{M_{0,c},M_{1,c},\ldots,M_{i,c},\ldots,M_{2^{\nu-1},c}\}(k)$ MTCOMB(I,K), I=1,TWONU

steady state metriccombination, $i=0,2^{\nu}-1$: I=1,TWONU

$\underline{m}_0=\{M_{0,0},M_{1,0},\ldots,M_{i,0},\ldots,M_{2^{\nu-1},0}\}$ SSMC(I), I=1,TWONU

Survivor $S_p(k)$, at time k, of the two parent-states of the state S_i. ($p \in \{p1,p2,p3,p4\}$)	SURV(I,K)
specific transition $[n_1(\alpha),n_2(\alpha)]_k(p,i)$ at time k of the survivor S_p towards S_i. transitions 00,01,10 and 11:	TRANS(I,K) TRANS(I,K)=0,1,2 and 3
outputvalue $[w(\alpha)]_k$ at time k, for the transition of the survivor S_p towards S_i	PATHRG(I,K)
the decoded path in terms of the states at time k=0,NUXL	STATE(K), K=1,NUXL+1
the transitions $[n_1(\alpha),n_2(\alpha)]_{k=0}^{k=NUXL-1}$ corresponding to the decoded path. transitionpairs 00,01,10 and 11.	NESTIM(1,K)/NESTIM(2,K), K=1,NUXL
the outputvalues $[w(\alpha)]_{k=0}^{k=NUXL}$ corresponding to the decoded path.	WESTIM(K), K=1,NUXL

```
ØK, ED MAINPR
GØ
EDIT
P1000
.NULL.
C      THIS IS THE MAIN PRØGRAM WHERE ALL NECESSARY CALCULATIØNS TAKE
C      PLACE. ALL SUBRØUTINES ARE CALLED AND THE CØMPLETE DECØDING-
C      PRØCEDURE IS PRINTED ØUT.
C      AS INPUT THE PRØGRAM ASKS LINE BY LINE:
C      NØ (THE NUMBER ØF THE CØDE WHEN USING THE TELETYPE)
C      NU
C      C(1,K)
C      C(2,K)
C      XL
C      X(K)
C      NØISE(1,K)
C      NØISE(2,K)
C      ************************************************************
C
C
       INTEGER XL,XL1,XLK,ZNUL,TWØNU,TWØNU1,X1(2,15)
       INTEGER C(2,7),D(2,6),X(15),NØISE(2,15),Z(21),Y(2,15),N2C1(21),
      1 N1C2(21),Y1D1(21),Y2D2(21),XC1(21),XC2(21),SB(6),S(16,6),
      1 ZNUL2I(16),WNUL2I(16),TT(16,13),TRANMX(16,9),M(16),TRMC(16),
      1 SSMC(16),MTCØMB(16,15),SURV(16,15),PATHRG(16,15),TRANS(16,15),
      1 BITS(15),STATE(15),WESTIM(15),NESTIM(2,15),SØMYD(15),XESTIM(15),
      1 DECØUT(16,50)
       REAL XD(16),YD(16)
       CALL TNØUA('NØ= ',4)
C READ THE NUMBER ØF THE CØDE.
       READ(1,4) NØ
       CALL TNØUA('NU= ',4)
C READ THE CØDE'S CØNSTRAINTLENGTH NU.
       READ(1,5) NU
       NU1=NU+1
       NU2=NU+2
       NUMIN1=NU-1
       TWØNU=2**NU
       TWØNU1=2**(NU-1)
       CALL TNØUA('C(1,K)= ',8)
C READ THE PØLYNØMIALS C(N,K),N=1,2 AND K=1,NU1.
       READ(1,6) (C(1,K),K=1,NU1)
       CALL TNØUA('C(2,K)= ',8)
       READ(1,6) (C(2,K),K=1,NU1)
       CALL TNØUA('XL= ',4)
C READ THE LENGTH XL ØF THE DATASEQUENCE X(K).(NU+XL NØT GREATER
C THEN 11).
       READ(1,5) XL
       XL1=XL+1
       NUXL=NU+XL
       NUXL1=NUXL+1
       CALL TNØUA('X(K)= ',6)
C READ THE DATASEQUENCE X(K),K=1,XL.
       READ(1,6) (X(K),K=1,XL)
       CALL TNØUA('NØISE(1,K)= ',12)
C READ THE NØISE-SEQUENCES NØISE(N,K),N=1,2 AND K=1,XL.
       READ(1,6) (NØISE(1,K),K=1,XL)
       CALL TNØUA('NØISE(2,K)= ',12)
       READ(1,6) (NØISE(2,K),K=1,XL)
4      FØRMAT(3A2)
5      FØRMAT(I2)
6      FØRMAT(15I1)
```

```
****************************************************************
   THIS FORTRAN VIDEO-DISPLAY PROGRAM SHOWS YOU THE COM-
PLETE SYNDROME-DECODING PROCEDURE OF BINARY R=1/2 CONVO-
LUTIONAL CODES OF CONSTRAINTLENGTH NU=2,3 AND 4 IN THE
CLASS T(NU,1).
   THIS CLASS CONSISTS OF CODES FOR WHICH BOTH THE FIRST
AND LAST STAGES OF THE ENCODERS SHIFTREGISTER ARE CONNEC-
TED WITH THE TWO MOD-2 ADDERS.
****************************************************************


   AS INPUT THE PROGRAM ASKS THE CODES CONSTRAINTLENGTH,
ITS CONNECTIONPOLYNOMIALS, THE DATASEQUENCE AT THE ENCO-
DER SIDE AND THE NOISESEQUENCE-PAIR ON THE CHANNEL.

   THE CODE, IN TERMS OF THE TWO CONNECTIONPOLYNOMIALS
C(N,K), N=1,2 AND K=1,NU+1 OF DEGREE NU, MAY BE CHOOSEN
FROM A TABLE. IN THIS TABLE ALL POSSIBLE POLYNOMIALS
HAVE BEEN FACTORIZED IN IRREDUCIBLE POLYNOMIALS IN ORDER
TO HELP THE USER TO CONSTRUCT A NON-CATASTROPHIC CODE.

   THE DATASEQUENCE X(K), K=1,XL OF LENGTH XL, AT THE
ENCODER INPUT MAY BE CHOOSEN FREELY WITH THE RESRICTION
THAT XL > 1 AND THAT SUM NUXL=NU+XL < 16 .

   THE NOISE-SEQUENCEPAIR NOISE(N,K), N=1,2 AND K=1,XL ,
MAY ALSO BE CHOOSEN FREELY.


 AT FIRST ALL THE ESSENTIAL DATA REQUIRED FOR THE DECO-
DING PROCEDURE, IS PRINTED OUT.

- THE POLYNOMIALS D(N,K), N=1,2 AND K=1,NU+1, OF DEGREE
  NU(NU WHICH SATISFY C(1,K)*D(1,K) + C(2,K)*D(2,K) = 1.
- THE CODES BASE STATE SB(K), K=1,NU.
- THE SPECIFIC OUTPUTVALUE ZNUL.
- THE CONSTRUCTION OF THE CHANNEL-SEQUENCES Y(N,K), N=1,2
  AND K=1,NUXL AT THE DECODER SIDE.
- THE OUTPUTSEQUENCE Z(K), K=1,NUXL OF THE W(ALPHA)-
  FORMER.
- THE STATE-TABLE WITH THE 2**NU STATES, THE FOUR STATE-
  TRANSITIONS AND THE FOUR SYNDROME-OUTPUTS Z FOR THE
  TRANSITIONS 00,01,11 AND 10 RESPECTIVELY.
- THE PARENT-STATE MATRIX WITH THE 2**NU STATES, THE
  FOUR PARENT-STATES, THE FOUR SYNDROME-OUTPUTS Z AND
  THE OUTPUTVALUE W.
  FOR THE EVEN-NUMBERED STATES WE HAVE THE FOUR PARENT-
  STATES FOR THE TRANSITIONS 00,00,11 AND 11, AND FOR
  THE ODD-NUMBERDED STATES WE HAVE THE FOUR PARENT-
  STATES FOR THE TRANSITIONS 01,01,10 AND 10 RESPEC-
  TIVELY.
- THE METRICVALUE TREEMETRIC(I), I=1,2**NU AT DEPTH NU.
- THE STEADY-STATE-METRICCOMBINATION SSMC(I), I=1,2**NU.

HEREAFTER THE COMPLETE TRELLIS-DECODING PROCEDURE IS
SHOWN, STARTING WITH THE STEADY-STATE-METRIC-
COMBINATION AND THE SYNDROMESEQUENCE Z(K).

EACH TIME K A COMPLETE SECTION OF THE TRELLIS IS SHOWN.
IN EACH SECTION FOR EACH STATE I, THE FOLLOWING DATA
ARE INDICATED:

- THE NEW METRICVALUE, PRINTED IN THE SPECIFIC NODE.
- THE SPECIFIC NODE (STATE), CONNECTED WITH ITS SURVI-
  VOR BY A SOLID BRANCH FOR A PATHREGISTERBIT 0, AND
  BY A DASHED BRANCH FOR A PATHREGISTERBIT 1.
- THE SPECIFIC TRANSITION (00,01,11 OR 10), PLACED AT
  THE LEFT SIDE OF THE NODE.


EACH SECTION IN THE TRELLIS CAN BE GENERATED BY THE
USER BY A CARRIAGE RETURN.

AT THE END, THE DECODED PATH IN THE TRELLIS IS SHOWN
BY MEANS OF A DOUBLING OF THE SEPERATE BRANCHES

IN CONCLUSION THE ESTIMATED DATASEQUENCE XESTIM(K),
K=1,XL, IS EVALUATED.
FOR THAT PURPOSE THE FOLLOWING OUTCOMES ARE PRINTED
OUT BENEATH THE TRELLIS:
- NESTIM(1,K), K=1,NUXL
- NESTIM(2,K), K=1,NUXL
- WESTIM(K) = NESTIM(1,K)*D(1,K) + NESTIM(2,K)*D(2,K)
- SOMYD(K)  = Y(1,K)*D(1,K) + Y(2,K)*D(2,K)
- XESTIM(K) = SOMYD(K) + WESTIM(K)

****************************************************************


BY GIVING A CARRIAGE RETURN THE TABLE OF FACTORIZED
POLYNOMIALS IS PRINTED.
```

Fig. 42a. Introductory text.

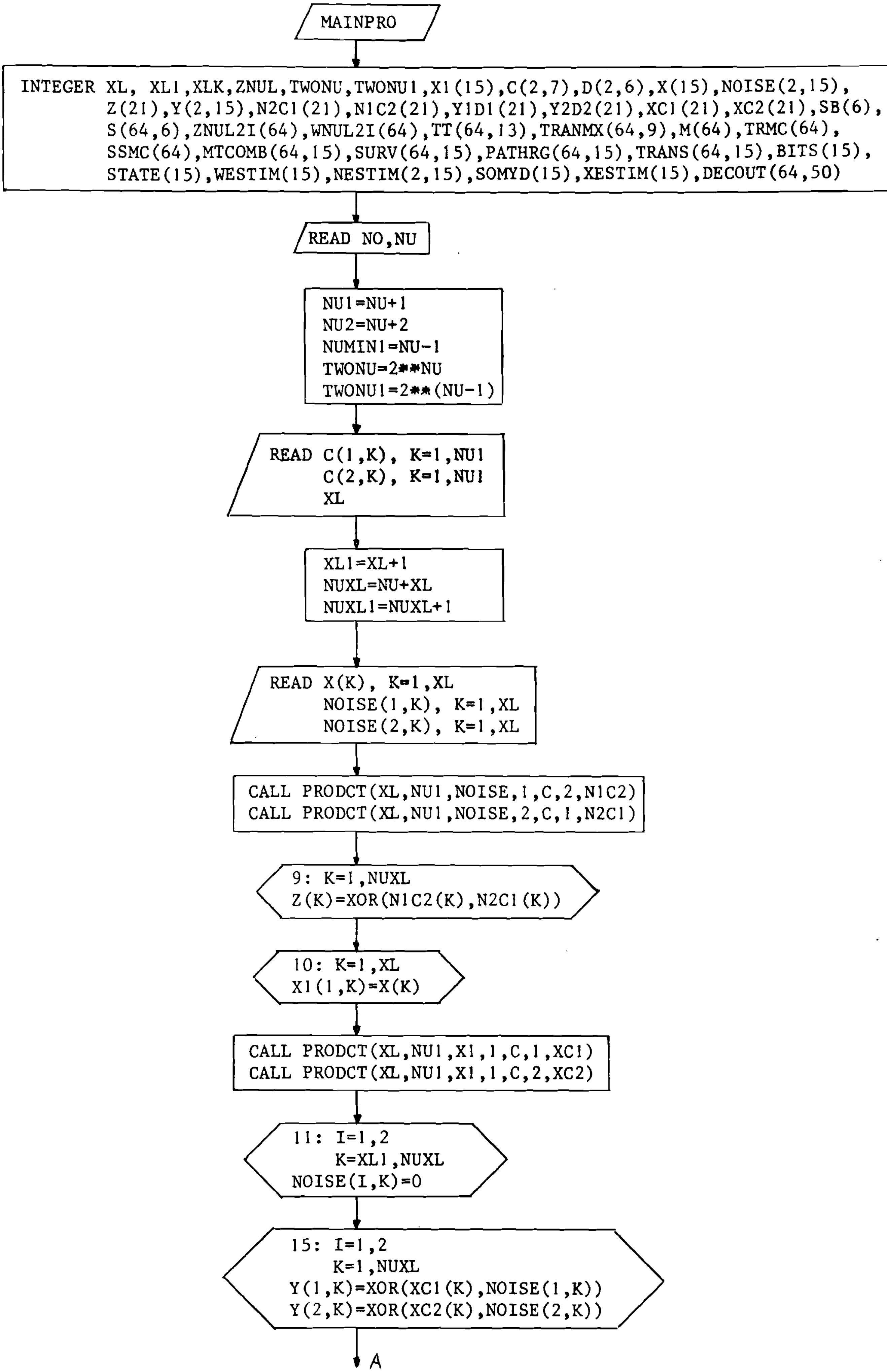
```
C CALCULATE THE PRØDUCTS NØISE(1,K)*C(2,K) AND NØISE(2,K)*C(1,K)
C ØF LENGTH NUXL.
      CALL PRØDCT(XL,NU1,NØISE,1,C,2,N1C2)
      CALL PRØDCT(XL,NU1,NØISE,2,C,1,N2C1)
C LØØP 9: THE MØD-2 SUM ØF THE PRØDUCTS ABØVE FØRM THE SYNDRØMEØUT-
C PUTSEQUENCE Z(K),K=1,NUXL.
      DØ 9 K=1,NUXL
9     Z(K)=XØR(N1C2(K),N2C1(K))
C LØØP 10: MAKE A TWØ-DIMENSIØNAL ARRAY ØUT ØF THE ØNE-DIMENSIØNAL
C ARRAY X(K), IN ØRDER TØ SATISFY THE FØRMAT ØF THE SUBRØUTINE
C PRØDCT.
      DØ 10 K=1,XL
10    X1(1,K)=X(K)
C CALCULATE THE ØUTPUTSEQUENCES X(K)*C(1,K) AND X(K)*C(2,K) ØF THE
C ENCØDER.
      CALL PRØDCT(XL,NU1,X1,1,C,1,XC1)
      CALL PRØDCT(XL,NU1,X1,1,C,2,XC2)
C LØØP 11: MAKE THE LAST NU BITS ØF THE NØISE-SEQUENCES EQUAL ZERØ.
      DØ 11 I=1,2
      DØ 11 K=XL1,NUXL
11    NØISE(I,K)=0
C LØØP 15: CALCULATE THE RECEIVED LINE-SEQUENCES Y(1,K)=X(K)*C(1,K)
C + NØISE(1,K) AND Y(2,K)=X(K)*C(2,K) + NØISE(2,K) ØF LENGTH NUXL.
      DØ 15 I=1,2
      DØ 15 K=1,NUXL
      Y(1,K)=XØR(XC1(K),NØISE(1,K))
15    Y(2,K)=XØR(XC2(K),NØISE(2,K))
C EVALUATE THE PØLYNØMIALS D(N,K),N=1,2 ØF DEGREE MU.
      CALL DPØLYN(1,NU,C,MU,D)
      CALL DPØLYN(2,NU,C,MU,D)
      MU1=MU+1
      NUMUXL=NU+MU+XL
C CALCULATE THE PRØDUCTS Y(1,K)*D(1,K) AND Y(2,K)*D(2,K) ØF
C LENGTH NUMUXL.
      CALL PRØDCT(NUXL,MU1,Y,1,D,1,Y1D1)
      CALL PRØDCT(NUXL,MU1,Y,2,D,2,Y2D2)
C LØØP 16: CALCULATE THEIR SUM MØD-2.
      DØ 16 K=1,NUMUXL
16    SØMYD(K)=XØR(Y1D1(K),Y2D2(K))
C CALCULATE THE BASE-STATE SB(K),K=1,NU.
      CALL BSTATE(C,NU,SB)
C CALCULATE THE SPECIFIC ØUTPUTVALUE ZNUL ØF THE TRANSITIØN ØF STATE
C S(B/2) TØWARDS THE ZERØ STATE S(0) WITH INPUTBITS 11.
      ZNUL=0
      DØ 20 K=2,NU
20    ZNUL=ZNUL+C(1,K)*SB(K-1)
C IF ZNUL IS EVEN THEN ZNUL=0 ELSE ZNUL=1.
      IF (ZNUL.EQ.((ZNUL/2)*2)) GØTØ 25
      ZNUL=1
      GØTØ 30
25    ZNUL=0
30    CØNTINUE
C EVALUATE THE STATE-MATRIX S(I,J),I=1,TWØNU AND J=1,NU. (IN FØRTRAN
C A ZERØ INDEX IN AN ARRAY IS NØT ALLØWED, HENCE S(I,J),I=1,J=1,NU
C HØLDS THE ZERØ STATE 00000...0 ØF LENGTH NU).
      CALL STATMX(NU,S)
```

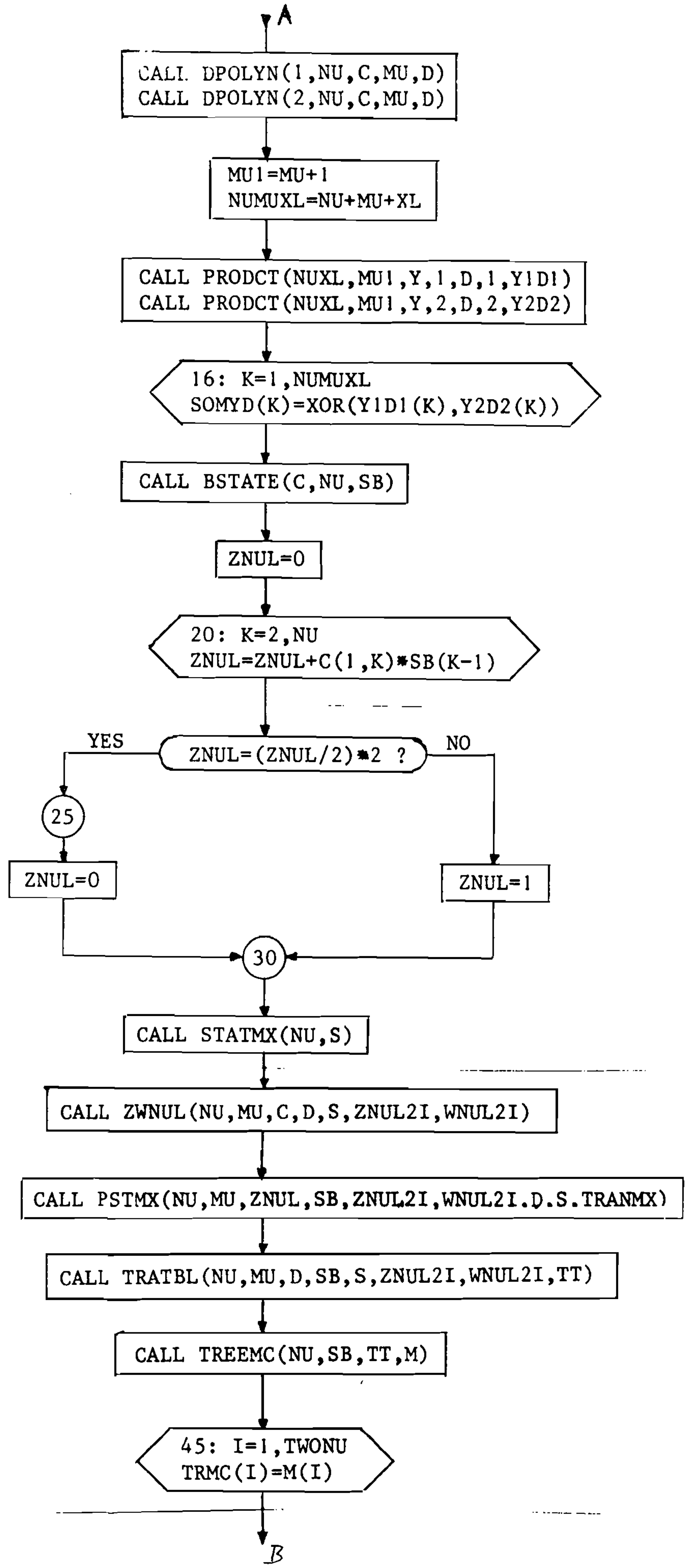
```
C CALCULATE THE PRØDUCTS NØISE(1,K)*C(2,K) AND NØISE(2,K)*C(1,K)
C ØF LENGTH NUXL.
      CALL PRØDCT(XL,NU1,NØISE,1,C,2,N1C2)
      CALL PRØDCT(XL,NU1,NØISE,2,C,1,N2C1)
C LØØP 9: THE MØD-2 SUM ØF THE PRØDUCTS ABØVE FØRM THE SYNDRØMEØUT-
C PUTSEQUENCE Z(K),K=1,NUXL.
      DØ 9 K=1,NUXL
9     Z(K)=XØR(N1C2(K),N2C1(K))
C LØØP 10: MAKE A TWØ-DIMENSIØNAL ARRAY ØUT ØF THE ØNE-DIMENSIØNAL
C ARRAY X(K), IN ØRDER TØ SATISFY THE FØRMAT ØF THE SUBRØUTINE
C PRØDCT.
      DØ 10 K=1,XL
10    X1(1,K)=X(K)
C CALCULATE THE ØUTPUTSEQUENCES X(K)*C(1,K) AND X(K)*C(2,K) ØF THE
C ENCØDER.
      CALL PRØDCT(XL,NU1,X1,1,C,1,XC1)
      CALL PRØDCT(XL,NU1,X1,1,C,2,XC2)
C LØØP 11: MAKE THE LAST NU BITS ØF THE NØISE-SEQUENCES EQUAL ZERØ.
      DØ 11 I=1,2
      DØ 11 K=XL1,NUXL
11    NØISE(I,K)=0
C LØØP 15: CALCULATE THE RECEIVED LINE-SEQUENCES Y(1,K)=X(K)*C(1,K)
C + NØISE(1,K) AND Y(2,K)=X(K)*C(2,K) + NØISE(2,K) ØF LENGTH NUXL.
      DØ 15 I=1,2
      DØ 15 K=1,NUXL
      Y(1,K)=XØR(XC1(K),NØISE(1,K))
15    Y(2,K)=XØR(XC2(K),NØISE(2,K))
C EVALUATE THE PØLYNØMIALS D(N,K),N=1,2 ØF DEGREE MU.
      CALL DPØLYN(1,NU,C,MU,D)
      CALL DPØLYN(2,NU,C,MU,D)
      MU1=MU+1
      NUMUXL=NU+MU+XL
C CALCULATE THE PRØDUCTS Y(1,K)*D(1,K) AND Y(2,K)*D(2,K) ØF
C LENGTH NUMUXL.
      CALL PRØDCT(NUXL,MU1,Y,1,D,1,Y1D1)
      CALL PRØDCT(NUXL,MU1,Y,2,D,2,Y2D2)
C LØØP 16: CALCULATE THEIR SUM MØD-2.
      DØ 16 K=1,NUMUXL
16    SUMYD(K)=XØR(Y1D1(K),Y2D2(K))
C CALCULATE THE BASE-STATE SB(K),K=1,NU.
      CALL BSTATE(C,NU,SB)
C CALCULATE THE SPECIFIC ØUTPUTVALUE ZNUL ØF THE TRANSITIØN ØF STATE
C S(B/2) TØWARDS THE ZERØ STATE S(0) WITH INPUTBITS 11.
      ZNUL=0
      DØ 20 K=2,NU
20    ZNUL=ZNUL+C(1,K)*SB(K-1)
C IF ZNUL IS EVEN THEN ZNUL=0 ELSE ZNUL=1.
      IF (ZNUL.EQ.((ZNUL/2)*2)) GØTØ 25
      ZNUL=1
      GØTØ 30
25    ZNUL=0
30    CØNTINUE
C EVALUATE THE STATE-MATRIX S(I,J),I=1,TWØNU AND J=1,NU. (IN FØRTRAN
C A ZERØ INDEX IN AN ARRAY IS NØT ALLØWED, HENCE S(I,J),I=1,J=1,NU
C HØLDS THE ZERØ STATE 00000...0 ØF LENGTH NU).
      CALL STATMX(NU,S)
C EVALUATE THE ØUTPUTVALUES ZNUL2I(I) AND WNUL2I(I) ,I=1,TWØNU/2.
      CALL ZWNUL(NU,MU,C,D,S,ZNUL2I,WNUL2I)
C CØNSTRUCT THE TRANSITIØN TABLE.
      CALL TRATBL(NU,MU,D,SB,S,ZNUL2I,WNUL2I,TT)
C EVALUATE THE FØUR PARENT STATES ØF EACH STATE S(I) WITH THEIR
C PARTICULAR TRANSITIØNS AND ØUTPUTVALUES Z(K) AND W(K).
      CALL PSTMX(NU,MU,ZNUL,SB,ZNUL2I,WNUL2I,D,S,TRANMX)
```

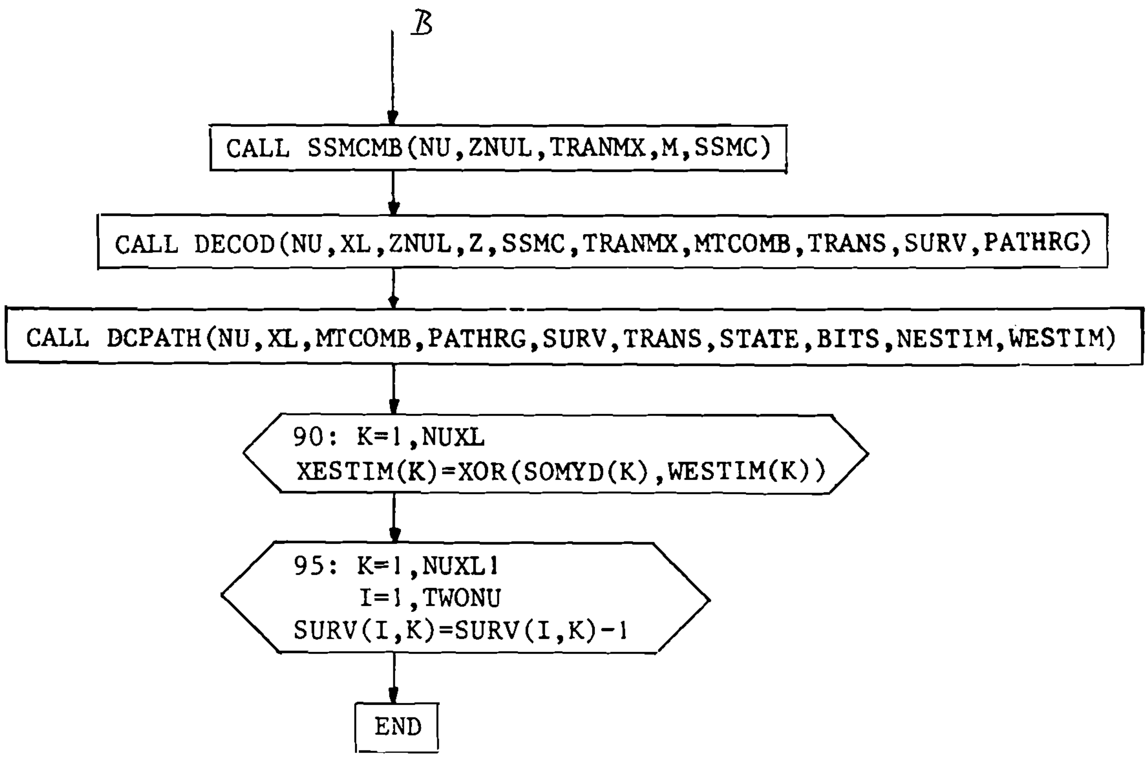
```
C CALCULATE THE METRIC-VALUES OF THE STATES S(I) AT DEPTH NUXL1
C STARTING WITH M(1)=0.
      CALL TREEMC(NU,SB,TT,M)
C GIVE THIS METRICCØMBINATIØN THE NAME TRMC.
      DØ 45 I=1,TWØNU
45    TRMC(I)=M(I)
C CALCULATE THE STEADY STATE METRIC CØMBINATIØN SSMC(I),I=1,TWØNU.
      CALL SSMCMB(NU,ZNUL,TRANMX,M,SSMC)
C CALCULATE EACH TIME K: THE NEW METRICCØMBINATIØN MTCØMB(I,K),
C THE SURVIVØRS SURV(I,K), THEIR TRANSITIØNS TRANS(I,K) AND THE
C CØRRESPØNDING ØUTPUTVALUES FØR W(K) IN PATHRG(I,K).
      CALL DECØD(NU,XL,ZNUL,Z,SSMC,TRANMX,MTCØMB,TRANS,SURV,PATHRG)
C EVALUATE THE DECØDED PATH IN TERMS ØF THE NEW STATE(K), THE
C TRANSITIØN TØ THAT STATE, NESTIM(1,K)/NESTIM(2,K) AND THE
C CØRRESPØNDING ØUTPUTVALUE WESTIM(K).
      CALL DCPATH(NU,XL,MTCØMB,PATHRG,SURV,TRANS,STATE,BITS,
     1 NESTIM,WESTIM)
C LØØP 90: EVALUATE THE ESTIMATED DATASEQUENCE.
      DØ 90 K=1,NUXL
90    XESTIM(K)=XØR(SØMYD(K),WESTIM(K))
C LØØP 95: AS THE SURVIVØRS ARE THE DECIMAL VALUE ØF THE BINARY
C REPRESENTATIØN ØF THE STATE, WE MUST SUBTRACT THE VALUE ØNE.
      DØ 95 K=1,NUXL1
      DØ 95 I=1,TWØNU
95    SURV(I,K)=SURV(I,K)-1
C     ************************************************************
C     ØUTPUT PRØGRAM
C     THE CØMPLETE DECØDINGPRØCEDURE IS PRINTED ØUT.
C     THE CØMPLETE TRELLIS UP TØ TIME K=NUXL IS GIVEN IN THE FØRM
C     ØF CØLUMNS AT EACH TIME K, THAT CØNTAIN THE METRICCØMBINATIØN
C     MTCØMB(I,K), THE SURVIVØRS SURV(I,K), THE ØUTPUTS W IN
C     PATHRG(I,K) AND THE TRANSITIØNS TRANS(I,K).
C     THE SPECIFIC DECØDED PATH IS INDICATED WITH AT EACH TIME K
C     THE ØUTPUTS Z(K) AND WESTIM(K), THE NEW STATE TIME K+1 AND
C     THE TRANSITIØNS TØ THAT NEW STATE.
C     AT LAST THE ESTIMATED DATASEQUENCE XESTIM(K) IS CØMPARED
C     WITH THE ØRIGINAL DATASEQUENCE X(K).
C     ***************************************************************
C
C
      WRITE(1,102) (D(1,K),K=1,MU1)
102   FØRMAT('D(1,K)= '6I1)
      WRITE(1,103) (D(2,K),K=1,MU1)
103   FØRMAT('D(2,K)= '6I1)
      WRITE(1,104) (SB(K),K=1,NU)
104   FØRMAT('SB(K) = '6I1)
      WRITE(1,105) ZNUL
105   FØRMAT('ZNUL  = 'I1/)
      WRITE(1,106) (XC1(K),K=1,NUXL)
106   FØRMAT(/'XC1(K)',T11,'= '20I1)
      WRITE(1,107) (NØISE(1,K),K=1,NUXL)
107   FØRMAT('NØISE(1,K)= '20I1)
      WRITE(1,108) (Y(1,K),K=1,NUXL)
108   FØRMAT('----------------------------+'/
     1'Y(1,K)',T11,'= '20I1/)
      WRITE(1,109) (XC2(K),K=1,NUXL)
109   FØRMAT(/'XC2(K)',T11,'= '20I1)
      WRITE(1,110) (NØISE(2,K),K=1,NUXL)
110   FØRMAT('NØISE(2,K)= '20I1)
      WRITE(1,111) (Y(2,K),K=1,NUXL)
111   FØRMAT('----------------------------+'/
     1'Y(2,K)',T11,'= '20I1/)
```

```
      WRITE(1,112) (N1C2(K),K=1,NUXL)
112   FORMAT(/'N1C2(K)',T11,'= '20I1)
      WRITE(1,113) (N2C1(K),K=1,NUXL)
113   FORMAT('N2C1(K)',T11,'= '20I1)
      WRITE(1,114) (Z(K),K=1,NUXL)
114   FORMAT('---------------------------+'/
     1'Z(K)',T11,'= '20I1/)
      WRITE(1,115) (Y1D1(K),K=1,NUMUXL)
115   FORMAT(/'Y1D1(K)',T11,'= '25I1)
      WRITE(1,116) (Y2D2(K),K=1,NUMUXL)
116   FORMAT('Y2D2(K)',T11,'= '25I1)
      WRITE(1,117) (SOMYD(K),K=1,NUXL)
117   FORMAT('---------------------------------+'/
     1'SOMYD(K)',T11,'= '20I1)
      WRITE(1,118)
118   FORMAT(/'STATE-TABLE:'/)
      DO 120 I=1,TWONU
120   WRITE(1,130) (TT(I,J),J=1,13)
130   FORMAT(I3,'-',4I3,1X,4I3,1X,4I3)
      WRITE(1,131)
131   FORMAT(/'PARENT-STATE MATRIX:'/)
      DO 135 I=1,TWONU
135   WRITE(1,140) TT(I,1),(TRANMX(I,J),J=1,9)
140   FORMAT(I3,'-',4I3,1X,4I3,3X,I1)
      WRITE(1,141) (TRMC(I),I=1,TWONU)
141   FORMAT(/'TREEMETRIC(I)= '64I1)
      WRITE(1,142) (SSMC(I),I=1,TWONU)
142   FORMAT('SSMC(I)       = '64I1)
      WRITE(1,143)
143   FORMAT(/'DECODING SCHEME:')
C LOOP 150: IN ORDER TO FACILITATE THE CONSTRUCTION OF THE LAYOUT
C OF THE DECODING SCHEME, THE ARRAY DECOUT IS INTRODUCED.
      DO 150 I=1,TWONU
      DECOUT(I,1)=MTCOMB(I,1)
      DO 150 K=1,NUXL
      DECOUT(I,4*K-2)=MTCOMB(I,K+1)
      DECOUT(I,4*K-1)=SURV(I,K+1)
      DECOUT(I,4*K  )=PATHRG(I,K+1)
      DECOUT(I,4*K+1)=TRANS(I,K+1)
150   CONTINUE
C THE NUMBER OF COLUMNS (SECTIONS) EQUALS NUXL1.
      JMAX=4*NUXL1-3
      DO 151 I=1,TWONU
151   WRITE(1,152) (DECOUT(I,J),J=1,JMAX)
152   FORMAT(I1,4X,39(2I2,2I1))
      WRITE(1,156) (Z(K),K=1,NUXL)
156   FORMAT(/'DECODED PATH:'//'Z(K)          ='14,19I5)
      WRITE(1,157) (WESTIM(K),K=1,NUXL)
157   FORMAT(/'WESTIM(K)    ='I4,19I5)
      WRITE(1,158) (STATE(K),K=1,NUXL1)
158   FORMAT(/'STATE(K)     =',20(I2,'---'))
      WRITE(1,160) (NESTIM(1,K),K=1,NUXL)
160   FORMAT(/'NESTIM(1,K) ='I4,19I5)
      WRITE(1,161) (NESTIM(2,K),K=1,NUXL)
161   FORMAT('NESTIM(2,K) ='I4,19I5)
```

```
      WRITE(1,162) (SØMYD(K),K=1,NUXL)
162   FØRMAT(/'SØMYD(K)     = '20I1)
      WRITE(1,164) (WESTIM(K),K=1,NUXL)
164   FØRMAT('WESTIM(K)    = '20I1)
      WRITE(1,166) (XESTIM(K),K=1,NUXL)
166   FØRMAT('-----------------------------+'/
     1'XESTIM(K)    = '20I1)
      WRITE(1,168) (X(K),K=1,XL)
168   FØRMAT(/,'X(K)          = '20I1)
      CALL EXIT
      END
C     END ØF ØUTPUT PRØGRAM
C     *************************************************************
BØTTØM
```


MAINPRO
INTEGER XL, XL1,XLK,ZNUL,TWONU,TWONU1,X1(15),C(2,7),D(2,6),X(15),NOISE(2,15),
Z(21),Y(2,15),N2C1(21),N1C2(21),Y1D1(21),Y2D2(21),XC1(21),XC2(21),SB(6),
S(64,6),ZNUL2I(64),WNUL2I(64),TT(64,13),TRANMX(64,9),M(64),TRMC(64),
SSMC(64),MTCOMB(64,15),SURV(64,15),PATHRG(64,15),TRANS(64,15),BITS(15),
STATE(15),WESTIM(15),NESTIM(2,15),SOMYD(15),XESTIM(15),DECOUT(64,50)
READ NO,NU
NU1=NU+1
NU2=NU+2
NUMIN1=NU-1
TWONU=2**NU
TWONU1=2**(NU-1)
READ C(1,K), K=1,NU1
C(2,K), K=1,NU1
XL
XL1=XL+1
NUXL=NU+XL
NUXL1=NUXL+1
READ X(K), K=1,XL
NOISE(1,K), K=1,XL
NOISE(2,K), K=1,XL
CALL PRODCT(XL,NU1,NOISE,1,C,2,N1C2)
CALL PRODCT(XL,NU1,NOISE,2,C,1,N2C1)
9: K=1,NUXL
Z(K)=XOR(N1C2(K),N2C1(K))
10: K=1,XL
X1(1,K)=X(K)
CALL PRODCT(XL,NU1,X1,1,C,1,XC1)
CALL PRODCT(XL,NU1,X1,1,C,2,XC2)
11: I=1,2
K=XL1,NUXL
NOISE(I,K)=0
15: I=1,2
K=1,NUXL
Y(1,K)=XOR(XC1(K),NOISE(1,K))
Y(2,K)=XOR(XC2(K),NOISE(2,K))
A


A
CALL DPOLYN(1,NU,C,MU,D)
CALL DPOLYN(2,NU,C,MU,D)
MU1=MU+1
NUMUXL=NU+MU+XL
CALL PRODCT(NUXL,MU1,Y,1,D,1,Y1D1)
CALL PRODCT(NUXL,MU1,Y,2,D,2,Y2D2)
16: K=1,NUMUXL
SOMYD(K)=XOR(Y1D1(K),Y2D2(K))
CALL BSTATE(C,NU,SB)
ZNUL=0
20: K=2,NU
ZNUL=ZNUL+C(1,K)*SB(K-1)
YES
ZNUL=(ZNUL/2)*2 ?
NO
25
ZNUL=0
ZNUL=1
30
CALL STATMX(NU,S)
CALL ZWNUL(NU,MU,C,D,S,ZNUL2I,WNUL2I)
CALL PSTMX(NU,MU,ZNUL,SB,ZNUL2I,WNUL2I.D.S.TRANMX)
CALL TRATBL(NU,MU,D,SB,S,ZNUL2I,WNUL2I,TT)
CALL TREEMC(NU,SB,TT,M)
45: I=1,TWONU
TRMC(I)=M(I)
B



B
CALL SSMCMB(NU,ZNUL,TRANMX,M,SSMC)
CALL DECOD(NU,XL,ZNUL,Z,SSMC,TRANMX,MTCOMB,TRANS,SURV,PATHRG)
CALL DCPATH(NU,XL,MTCOMB,PATHRG,SURV,TRANS,STATE,BITS,NESTIM,WESTIM)
90: K=1,NUXL
XESTIM(K)=XOR(SOMYD(K),WESTIM(K))
95: K=1,NUXL1
I=1,TWONU
SURV(I,K)=SURV(I,K)-1
END

```
ER! ED SUB1
GØ
EDIT
PP1000
.NULL.
C ***********************************************
C     SUBRØUTINE PRØDCT
C     THE PRØDUCT ØF TWØ BINARY PØLYNØMIALS A(IA,M), IA=1,2 AND
C     B(IB,N), IB=1,2 ØF DEGREE M AND N (M AND N NØT EQUAL ØNE),
C     IS CALCULATED AND ENTERED IN THE ARRAY PRØD(K),K=1,M+N-1.
C     A(IA,M) HAS THE CØEFFICIENT CØRRESPØNDING TØ THE HIGHEST DEGREE
C     AT THE LAST PØSITIØN.
C     B(IB,N) HAS THE CØEFFICIENT CØRRESPØNDING TØ THE HIGHEST DEGREE
C     AT THE FIRST PØSITIØN (B(IB,1)=1).
C     THE DEGREE ØF A(M) CØRRESPØNDS TØ THE TØTAL LENGTH ØF THE SERIES
C     ØF CØEFFICIENTS, EVEN IF A NUMBER ØF FIRST CØEFFICIENTS IS ZERØ.
C     (1) INTRØDUCTIØN , PP(1-3)
C     ****************************************************************
C
C
      SUBRØUTINE PRØDCT(M,N,A,IA,B,IB,PRØD)
C IA AND IB ARE INTEGER RØW-NUMBERS: IA=1,2 AND IB=1,2.
      INTEGER A(2,15),B(2,7),PRØD(21),NPRØD(21)
C LØØP 20: SINCE B(IB,1) IS ALWAYS EQUAL TØ ØNE, WE WILL FIRST
C MAKE THE PRØDUCT EQUAL TØ PØLYNØMIAL A(IA,K),K=1,M.
      DØ 20 K=1,M
20    PRØD(K)=A(IA,K)
      MN1=M+N-1
C START NUMBER ØF INTERMEDIATE ZERØ'S: NRZ=1.
50    NRZ=1
C LØØP 60: THE SUCCESSIVE SHIFTS AND MØD-2 ADDITIØNS ARE CARRIED ØUT.
      DØ 60 I=2,N
C IF B(IB,I)=0 GØTØ 55.
      IF (B(IB,I).EQ.0) GØTØ 55
C LØØP 51: IF B(IB,I)=1 WE TAKE ØVER THE FIRST NRZ PØSITIØNS ØF
C A(IA,K) IN THE NEW PRØDUCT.
      DØ 51 K=1,NRZ
51    NPRØD(K)=A(IA,K)
      KBEGIN=NRZ+1
C LØØP 52: ADD MØD-2 THE ØVERLAPPING PARTS ØF THE ØLD PRØDUCT AND
C A(IA,K).
      DØ 52 K=KBEGIN,M
      KN=K-NRZ
52    NPRØD(K)=XØR(A(IA,K),PRØD(KN))
      KBEGIN=M+1
      KEND= M+I-1
C LØØP 53: TAKE ØVER THE TAIL ØF THE ØLD PRØDUCT.
      DØ 53 K=KBEGIN,KEND
      KN=K-NRZ
53    NPRØD(K)=PRØD(KN)
C LØØP 54: THE ØLD PRØDUCT BECØMES THE NEW PRØDUCT.
      DØ 54 K=1,KEND
54    PRØD(K)=NPRØD(K)
```

```
C INITIAL VALUE NRZ=1 AND PRØCEED.
      NRZ=1
      GØTØ 60
C IF THE LAST BIT B(IB,N)=0 GØTØ 56 FØR SPECIAL CASES.
55    IF ((I.EQ.N).AND.(B(IB,N).EQ.0)) GØTØ 56
C IF B(IB,I)=0 AND I IS NØT EQUAL TO N, INCREASE NRZ WITH ØNE AND
C PRØCEED.
      NRZ=NRZ+1
      GØTØ 60
56    CØNTINUE
C LØØP57: MAKE THE FIRST NRZ BITS ØF THE NEW PRØDUCT EQUAL ZERØ.
      DØ 57 K=1,NRZ
57    NPRØD(K)=0
      NRZ1=NRZ+1
C LØØP 58: SHIFT THE ØLD PRØDUCT NRZ PLACES.
      DØ 58 K=NRZ1,MN1
      KN=K-NRZ
58    NPRØD(K)=PRØD(KN)
C LØØP 59: THE ØLD PRØDUCT BECØMES THE NEW PRØDUCT.
      DØ 59 K=1,MN1
59    PRØD(K)=NPRØD(K)
60    CØNTINUE
      RETURN
      END
C     END ØF PRØDCT
C     ****************************************************************
C     SUBRØUTINE DPØLYN
C     GIVEN NU AND THE PØLYNØMIALS C(N,K), N=1,2, K=1,NU+1, ØF DE-
C     GREE NU, THE PØLYNØMIALS D(N,K), N=1,2, K=1,MU+1, ØF MINIMUM
C     DEGREE MU, WHICH SATISFY C(1)D(1) + C(2)D(2) = 1, ARE
C     CØNSTRUCTED.
C     THE CØEFFICIENT ØF HIGHEST DEGREE IS BØTH FØR THE PØLYNØ-
C     MIALS C(N,K) AND D(N,K) LØCATED AT THE LEFTMØST PØSITIØN.
C     N=1 ----- D(1,K)
C     N=2 ----- D(2,K)
C     (1) INTRØDUCTIØN, PP(2)
C     ***********************************************************
C
C
      SUBRØUTINE DPØLYN(N,NU,C,MU,D)
      INTEGER P,SIGN,SØM(2),A(8),B(8),C(2,7),D(2,6)
      NU1=NU+1
      NU2=NU+2
      NUMIN1=NU-1
      NUMIN2=NU-2
C LØØP 5: CALCULATIØN IS CARRIED ØUT WITH HELP ØF THE REGISTERS
C A(K), K=1,NU2 AND B(K), K=1,NU2. WE START WITH A(K)=C(1,K), K=1,NU1
C AND B(K)=C(2,K), K=1,NU1.
      DØ 5 K=1,NU1
      A(K)=C(1,K)
5     B(K)=C(2,K)
C IF WE WANT TØ CALCULATE PØLYNØMIAL D(1,K) GØING WITH C(1,K), THE
C VALUE ØF THE PARAMTER P(-1) (SEE BERLEKAMP, CØDING THEØRY, PP(36-44))
C SHØULD BE EQUAL TØ ZERØ AND IF WE WANT TØ CALCULATE D(2,K) GØING
C WITH C(2,K) THIS VALUE SHØULD BE EQUAL TØ ØNE.
      P=N-1
C THE VALUE ØF THE PARAMETER P IS PLACED AFTER A(K) AND THE CØMPLE-
C MENT ØF THIS VALUE AFTER B(K).
      A(NU2)=P
      B(NU2)=1-P
```

```
C THE TWØ CØMMA'S KØMMA AND KØMMB, NECESSARY FØR THE SHIFTING ØPE-
C RATIØN ARE PLACED AT THE LAST BUT ØNE PØSITIØN.
      KØMMA=NU1
      KØMMB=NU1
C THE STARTING VALUE ØF THE ØPERATING VARIABLE SIGN IS SET TØ
C 0 (EVEN), IF N=1 AND SET TØ 1 (ØDD), IF N=2.
      SIGN=P
C IF A NUMBER ØF FIRST BITS IN A(K) EQUALS ZERØ THEN A(K) MUST BE
C SHIFTED THE LEFT UNTIL THE FIRST BIT EQUAL TØ ØNE IS ENCØUNTERED.
C THE CØMMA MUST BE SHIFTED AN EQUAL NUMBER ØF PLACES.
C THE LAST PØSITIØNS IN A(K) MUST BE FILLED UP WITH ZERØ'S.
10    IF (A(1).EQ.1) GØTØ 20
      KØMMA=KØMMA-1
C IF THE CØMMA HAS REACHED THE FIRST PØSITIØN THEN WE ARE READY
C AND THE PØLYNØMIAL A(K) AFTER THE CØMMA KØMMA IS THE DESIRED PØLY-
C NØMIAL D(N,K) WITH THE CØEFFICIENT ØF LØWEST DEGREE 0 AT THE LEFT-
C MØST PØSITIØN.
      IF (KØMMA.EQ.0) GØTØ 45
      DØ 15 I=1,NU1
15    A(I)=A(I+1)
      A(NU2)=0
      GØTØ 10
C IF A NUMBER ØF FIRST BITS IN B(K) EQUALS ZERØ THEN B(K) MUST BE
C SHIFTED TØ THE LEFT IN THE SAME MANNER AS A(K).
20    IF (B(1).EQ.1) GØTØ 30
      KØMMB=KØMMB-1
C IF THE CØMMA HAS REACHED THE FIRST PØSITIØN THEN WE ARE READY
C AND THE PØLYNØMIAL B(K) AFTER THE CØMMA KØMMB IS THE DESIRED
C PØLYNØMIAL D(N,K) WITH THE CØEFFICIENT ØF LØWEST DEGREE 0 AT
C THE LEFTMØST PØSITIØN.
      IF (KØMMB.EQ.0) GØTØ 50
      DØ 25 I=1,NU1
25    B(I)=B(I+1)
      B(NU2)=0
      GØTØ 20
C THE VALUE ØF THE VARIABLE SIGN IS DETERMINED BY THE PLACES ØF
C THE TWØ CØMMA'S KØMMA AND KØMMB.
30    IF ((KØMMA.EQ.KØMMB).AND.(SIGN.EQ.0)) GØTØ 35
      IF ((KØMMA.EQ.KØMMB).AND.(SIGN.EQ.1)) GØTØ 31
      IF (KØMMA.GT.KØMMB) GØTØ 35
      IF (KØMMA.LT.KØMMB) GØTØ 31
C LABEL 31-33: ØPERATIØN FØR SIGN=1.
31    SIGN=1
C LØØP 32: ADD THE CØNTENTS ØF REGISTER A(K) FRØM THE FIRST PØ-
C SITIØN UP TØ AND INCLUDING THE PØSITIØN KØMMA, MØD-2 TØ THE
C CØNTENTS ØF REGISTER B(K) IN THE SAME RANGE.
      DØ 32 K=1,KØMMA
32    B(K)=XØR(A(K),B(K))
C LØØP 33: ADD THE CØNTENTS ØF REGISTER B(K) FRØM PØSITIØN KØMMB+1
C UP TØ AND INCLUDING THE LAST PØSITIØN, MØD-2 TØ THE CØNTENTS ØF
C REGISTER A(K) IN THE SAME RANGE.
      KØMMB1=KØMMB+1
      DØ 33 K=KØMMB1,NU2
33    A(K)=XØR(A(K),B(K))
```

```
C AFTER THIS ØPERATIØN WE MUST FIRST CHECK UP ØN THE NUMBER ØF
C FIRST ZERØ'S.
      GØTØ 10
C LABEL 35-37: ØPERATIØN FØR SIGN=0.
35    SIGN=0
C LØØP 36: ADD THE CØNTENTS ØF REGISTER B(K) FRØM THE FIRST PØ-
C SITIØN UP TØ AND INCLUDING THE PØSITIØN KØMMB, MØD-2 TØ THE
C CØNTENTS ØF REGISTER A(K) IN THE SAME RANGE.
      DØ 36 K=1,KØMMB
36    A(K)=XØR(A(K),B(K))
C LØØP 37: ADD THE CØNTENTS ØF REGISTER A(K) FRØM PØSITIØN KØMMA+1
C UP TØ AND INCLUDING THE LAST PØSITIØN, MØD-2 TØ THE CØNTENTS ØF
C REGISTER B(K) IN THE SAME RANGE.
      KØMMA1=KØMMA+1
      DØ 37 K=KØMMA1,NU2
37    B(K)=XØR(A(K),B(K))
C AFTER THIS ØPERATIØN WE MUST FIRST CHECK UP ØN THE NUMBER ØF
C FIRST ZERØ'S.
      GØTØ 10
C LØØP 46: IN ØRDER TØ HAVE THE CØEFFICIENT ØF HIGHEST DEGREE AT
C THE LEFTMØST PØSITIØN, THE PØLYNØMIAL A(K) AFTER THE CØMMA KØMMA
C MUST BE REVERSED.
45    DØ 46 K=1,NU
      NU2K=NU2-K
46    D(N,K)=A(NU2K)
      GØTØ 52
C LØØP 51: IN ØRDER TØ HAVE THE CØEFFICIENT ØF HIGHEST DEGREE AT
C THE LEFTMØST PØSITIØN, THE PØLYNØMIAL B(K) AFTER THE CØMMA KØMMB
C MUST BE REVERSED.
50    DØ 51 K=1,NU
      NU2K=NU2-K
51    D(N,K)=B(NU2K)
52    CØNTINUE
C LØØP 54: IN ØRDER TØ DETERMINE THE DEGREE MU ØF THE PØLYNØMIAL
C D(N,K), WE MUST FIRST CALCULATE THE NUMBER ØF FIRST ZERØ'S.
C AS THE MAXIMUM VALUE ØF DEGREE MU IS EQUAL TØ NUMIN1, THE INITIAL
C VALUE ØF MU IS EQUAL TØ NUMIN1.
      MU=NUMIN1
      DØ 54 K=1,NUMIN2
      IF (D(N,K).EQ.1) GØTØ 55
54    MU=MU-1
C LØØP 56: THE FIRST ZERØ'S ARE DELETED AND THE PØLYNØ'001AL D(N,K)
C BECØMES THE TAIL ØF THE PØLYNØMIAL CALCULATED BEFØRE.
55    MU1=MU+1
      DØ 56 K=1,MU1
      NMK=NU-MU1+K
56    D(N,K)=D(N,NMK)
      RETURN
      END
C     END ØF DPØLYN
```

```
C      ************************************************************
C      SUBRØUTINE BSTATE
C      THE BINARY REPRESENTATIØN ØF THE CØDES BASE-STATE IS ENTERED
C      IN THE ARRAY SB(I), I=1,NU.
C      (2) STATE TABLE, PP(4,5)
C      ************************************************************
C
C
C
C
       SUBRØUTINE BSTATE(C,NU,SB)
       INTEGER SØM,SB(6),C(2,7)
C THE LAST SHIFT ØF THE BASE STATE IN THE SYNDRØME-FØRMER HØLDS:
C C(2,1)+C(1,1)*SB(NU)=0. SINCE C(1,1)=C(2,1)=1, WE HAVE
C SB(NU)=-C(2,1).
       SB(NU)=-C(2,1)
       NUMIN1=NU-1
C LØØP 15: WE SHIFT THE BASE STATE FRØM LEFT TØ RIGHT INTØ THE
C SYNDRØME-FØRMER AND CALCULATE AT EACH STEP K THE CØNTRIBUTIØN
C TØ THE ØUTPUT Z(K), WHICH MUST BE ZERØ.
       DØ 15 K=1,NUMIN1
       SØM=0
C LØØP 10: WE CALCULATE THE PRØDUCT ØF THE SHIFTED BASE STATE AND
C AND THE LØWER PØLYNØMIAL C(1,K).
       DØ 10 I=1,K
       NUKI=NU-K+I
10     SØM=SØM+SB(NUKI)*C(1,I+1)
C THE NEXT CØEFFICIENT SB(NU-K) IS CALCULATED.
       NUK=NU-K
       SB(NUK)=-C(2,K+1)-SØM
15     CØNTINUE
C LØØP 20: IF SB(K) IS EVEN THEN SB(K)=0 ELSE SB(K)=1.
       DØ 20 K=1,NU
       IF (SB(K).EQ.((SB(K)/2)*2)) GØTØ17
       SB(K)=1
       GØTØ 20
17     SB(K)=0
20     CØNTINUE
       RETURN
       END
C      END ØF BSTATE
C      ************************************************************
C      SUBRØUTINE STATMX
C      THE STATEMATRIX S(2**NU,NU) IS CØNSTRUCTED. THE MATRIX CØNTAINS
C      THE BINARY REPRESENTATIØN ØF THE DECIMAL VALUES ØF THE 2**NU
C      ABSTRACT STATES.
C      (2) STATE TABLE, PP(7,8)
C      ************************************************************
C
C
       SUBRØUTINE STATMX(NU,S)
       INTEGER TWØNU,S(16,6)
       TWØNU=2**NU
C LØØP 10: FILL UP THE STATE MATRIX WITH ZERØ'S.
       DØ 10 I=1,TWØNU
       DØ 10 J=1,NU
       S(I,J)=0
10     CØNTINUE
```

```
C THE ALGØRITHM IS BUILT UP FRØM THE TØP TØ THE BØTTØM, AND FRØM
C THE RIGHT TØ THE LEFT ØF THE STATE MATRIX. THE LAST CØLUMN
C CØNSISTS ØF THE SEQUENCE 010101010101... ØF LENGTH 2**NU.
C THE LAST BUT ØNE CØLUMN IS 0011001100110011...
C THE LAST BUT TWØ CØLUMN IS 0000111100001111000... AND SØ FØRTH.
      DØ 20 J=1,NU
      KMAX=2**(J-1)
      DØ 20 K=1,KMAX
      LMAX=2**(NU-J)
      DØ 20 L=1,LMAX
      I=2**(NU-J+1)*K -L+1
      S(I,J)=1
20    CØNTINUE
      RETURN
      END
C     END ØF STATMX
C     SUBRØUTINE ZWNUL
C     THE ØUTPUTVALUES Z AND W FØR THE TRANSITIØNS ØF THE STATES
C     S(I) TØWARDS THE STATES S(2I) FØR THE BITPAIR 00, ARE CAL-
C     CULATED AND ENTERED IN THE ARRAYS ZNUL2I(I) AND WNUL2I(I),
C     I=1,TWØNU,2.
C     (3) STATE-DIAGRAM, PP(11-13)
C     ***************************************************************
C
C
      SUBRØUTINE ZWNUL(NU,MU,C,D,S,ZNUL2I,WNUL2I)
      INTEGER TWØNU,C(2,7),D(2,6),S(16,6),ZNUL2I(16),WNUL2I(16)
      TWØNU=2**NU
      DØ 40 I=1,TWØNU,2
      ZNUL2I(I)=0
      WNUL2I(I)=0
C LØØP 31: CALCULATIØN ØF THE CØNTRIBUTIØNS ØF THE CØNTENTS ØF THE
C SEPERATE REGISTERPLACES ØF THE LØWER PØLYNØMIAL C(1,K).
      DØ 31 K=1,NU
      IHALF=(I+1)/2
31    ZNUL2I(I)=ZNUL2I(I)+C(1,K)*S(IHALF,K)
C IF ZNUL2I(I) IS EVEN THEN ZNUL2I(I)=0 ELSE ZNUL2I(I)=1.
      IF (ZNUL2I(I).EQ.((ZNUL2I(I)/2)*2)) GØTØ 32
      ZNUL2I(I)=1
      GØTØ 33
32    ZNUL2I(I)=0
33    CØNTINUE
C LØØP 34: CALCULATIØN ØF THE CØNTRIBUTIØNS ØF THE CØNTENTS ØF THE
C SEPERATE REGISTERPLACES ØF THE LØWER PØLYNØMIAL D(2,K).
      DØ 34 K=1,MU
      IHALF=(I+1)/2
      NUMUK=NU-MU+K
34    WNUL2I(I)=WNUL2I(I)+D(2,K)*S(IHALF,NUMUK)
C IF WNUL2I(I) IS EVEN THEN WNUL2I(I)=0 ELSE WNUL2I(I)=1.
      IF (WNUL2I(I).EQ.((WNUL2I(I)/2)*2)) GØTØ 35
      WNUL2I(I)=1
      GØTØ 40
35    WNUL2I(I)=0
40    CØNTINUE
      RETURN
      END
C     END ØF ZWNUL
C     *******************************************************************
```

```
C      ***************************************************************
C      SUBRØUTINE TRATBL
C      GIVEN THE STATE-MATRIX S(I,J), THE BASE-STATE AND THE ØUTPUT-
C SERIES ZNUL2I(I), I=1,TWØNU,2 AND WNUL2I(I), I=1,TWØNU,2, THE
C TRANSITIØNTABLE IS CØNSTRUCTED. THE FIRST CØLUMN CØNTAINS THE DE-
C CIMAL STATENUMBER. IN THE NEXT FØUR CØLUMNS THE NEW STATES FØR THE
C TRANSITIØNS 00,01,11 AND 10 RESPECTIVELY ARE INDICATED. THE CØR-
C RESPØNDING ØUTPUTVALUES Z AND W ARE TØ BE FØUND IN THE LAST EIGHT
C CØLUMNS.
C      (2) STATE TABLE, PP(7,8)
C      ***************************************************************
C
C
       SUBRØUTINE TRATBL(NU,MU,D,SB,S,ZNUL2I,WNUL2I,TT)
       INTEGER TWØNU1,SØM,SB(6),SSB(6),ZNUL2I(16),WNUL2I(16),S(16,6),
      1 TT(16,13),D(2,6)
       TWØNU1=2**(NU-1)
C LØØP 20: THE UPPER HALF ØF THE TRANSITIØN TABLE IS CØNSTRUCTED.
       DØ 20 I=1,TWØNU1
C IN THE FIRST CØLUMN THE DECIMAL STATENUMBER IS PLACED.
       TT(I,1)=I-1
C THE SECØND CØLUMN CØRRESPØNDS TØ THE TRANSITIØN S(I) TØWARDS
C S(2I) FØR AN INPUTPAIR 00.
       TT(I,2)=2*I-2
C THE THIRD CØLUMN CØRRESPØNDS TØ THE TRANSITIØN S(I) TØWARDS
C S(2I+1) FØR AN INPUTPAIR 01.
       TT(I,3)=TT(I,2)+1
C LØØP 10: THE STATE S(I) CØNVERTS INTØ THE STATE S(2I+1)+S(B)
C FØR AN INPUTPAIR 11. THE DECIMAL VALUE ØF THIS MØD-2 SUM
C IS ENTERED IN THE VARIABLE SØM.
       SØM=0
       DØ 10 K=1,NU
       SSB(K)=XØR(SB(K),S(2*I,K))
10     SØM=SØM+SSB(K)*(2**(NU-K))
       TT(I,4)=SØM
C THE STATE S(I) CØNVERTS INTØ THE STATE S(2I)+S(B) FØR AN INPUT-
C PAIR 10. THE TRANSITIØNS 11 AND 10 ØNLY DIFFER IN THE RIGHT-
C MØST BIT, HENCE THE CØNTENTS ØF THE FIFTH CØLUMN ARE THE
C CØNTENTS ØF THE FØURTH CØLUMN PLUS ØNE.
       TT(I,5)=TT(I,4)+1
C THE ØUTPUTVALUE ZNUL2I(I) CØRRESPØNDING TØ THE TRANSITIØN ØF
C STATE S(I) TØWARDS THE STATE S(2I), HAS ALREADY BEEN CALCU-
C LATED. THE CØNTENTS ØF CØLUMNS 7,8 AND 9 CAN DIRECTLY BE DERIVED
C FRØM THIS VALUE ZNUL2I(I).
       TT(I,6)=ZNUL2I(2*I-1)
       TT(I,8)=TT(I,6)
       TT(I,7)=1-TT(I,6)
       TT(I,9)=TT(I,7)
C THE ØUTPUTVALUE WNUL2I(I) CØRRESPØNDING TØ THE TRANSITIØN ØF
C STATE S(I) TØWARDS THE STATE S(2I), HAS ALREADY BEEN CALCU-
C LATED. THE CØNTENTS ØF CØLUMNS 11,12 AND 13 CAN BE DERIVED FRØM
C THIS VALUE WHEN WE KNØW THE VALUE D(1,MU+1) ØF THE PØLYNØMIAL
C D(1,K), K=1,MU1.
       TT(I,10)=WNUL2I(2*I-1)
       TT(I,11)=TT(I,10)
       IF (D(1,MU+1).EQ.0) TT(I,11)=1-TT(I,10)
       TT(I,12)=1-TT(I,10)
       TT(I,13)=1-TT(I,11)
20     CONTINUE
```

```
C LØØP 45: THE CØNTENTS ØF THE LØWER HALF ØF THE TRANSITIØN TABLE
C CAN EASILY BE DERIVED FRØM THE CØNTENTS ØF THE UPPER HALF.
      DØ 45 I=1,TWØNU1
      ITWØNU=I+TWØNU1
      TT(ITWØNU,1)=TT(I,1)+TWØNU1
      DØ 30 J=2,5
30    TT(ITWØNU,J)=TT(I,J)
C LØØP 35: THE LØWER HALF IS THE CØMPLEMENT ØF THE UPPER HALF.
      DØ 35 J=6,9
35    TT(ITWØNU,J)=1-TT(I,J)
C LØØP 40: THE LØWER HALF IS EQUAL TØ THE UPPER HALF.
      DØ 40 J=10,13
40    TT(ITWØNU,J)=TT(I,J)
45    CØNTINUE
      RETURN
      END
C     END ØF TRATBL
C     ***************************************************************
C     SUBRØUTINE PSTMX
C     THE PARENT-STATEMATRIX IS CØNSTRUCTED. IN EACH RØW ØNE CAN
C     FIND THE STATE I AND ITS FØUR PARENT-STATES, THE ØUTPUTS
C     Z FØR THE TRANSITIØNS 01,01,10,10 AND 00,00,11,11 RESPEC-
C     TIVELY. THE LAST CØLUMN CØRRESPØNDS WITH THE W-ØUTPUT.
C     (3) STATE DIAGRAM, PP(11-12)
C     ***************************************************************
C
C
      SUBRØUTINE PSTMX(NU,MU,ZNUL,SB,ZNUL2I,WNUL2I,D,S,TRANMX)
      INTEGER ZNUL,P3(6),SB(6),ZNUL2I(16),WNUL2I(16),TRANMX(16,9)
      INTEGER TWØNU,D(2,6),S(16,6)
      TWØNU=(2**NU)
      MU1=MU+1
C LØØP10: FILL UP THE CØLUMNS 5,6,7,8 AND 9 ØF THE ØDD RØWS.
      DØ 10 I=1,TWØNU,2
      TRANMX(I,5)=ZNUL2I(I)
      TRANMX(I,6)=1-ZNUL2I(I)
      TRANMX(I,7)=ZNUL2I(I) + ZNUL
      IF (TRANMX(I,7).EQ.2) TRANMX(I,7)=0
      TRANMX(I,8)=1-TRANMX(I,7)
      TRANMX(I,9)=WNUL2I(I)
10    CØNTINUE
C LØØP 20: FILL UP THE CØLUMNS 1,2,3 AND 4 ØF THE ØDD RØWS.
      DØ 20 I=1,TWØNU,2
C S(P1)=S(I/2).
      TRANMX(I,1)=(I-1)/2
C S(P2)=S(P1+2**(NU-1)).
      TRANMX(I,2)=TRANMX(I,1)+2**(NU-1)
      IHALF=(I+1)/2
C LØØP 15: S(P3)=S(I/2) + S((B-1)/2). (MØD-2 SUM)
      DØ 15 K=2,NU
      KMIN1=K-1
15    P3(K)=XØR(S(IHALF,K),SB(KMIN1))
C LØØP 16: THE DECIMAL VALUE ØF THE BINARY REPRESENTATIØN ØF
C THE PARENT STATE P3(K) IS CALCULATED.
      TRANMX(I,3)=0
      DØ 16 K=2,NU
16    TRANMX(I,3)=TRANMX(I,3)+P3(K)*(2**(NU-K))
C S(P4)=S(P3+2**(NU-1)).
      TRANMX(I,4)=TRANMX(I,3)+2**(NU-1)
20    CØNTINUE
```

```
C LØØP 40: THE CØNTENTS ØF THE EVEN RØWS IS EVALUATED.
      DØ 40 I=1,TWØNU,2
C LØØP 30: THE PARENT STATES ØF THE STATE S(I+1) ARE THE SAME AS
C THE PARENT STATES ØF THE STATE S(I).
      DØ 30 J=1,4
30    TRANMX(I+1,J)=TRANMX(I,J)
C LØØP 35: THE ØUTPUTVALUES CØRRESPØNDING TØ THE TRANSITIØNS TØ
C THE STATE S(I+1) ARE THE CØMPLEMENT ØF THE TRANSITIØNS
C TØ THE STATE S(I).
      DØ 35 J=5,8
35    TRANMX(I+1,J)=1-TRANMX(I,J)
C IF D(1,MU1)=0, WE HAVE THE ØUTPUTPATTERN 0110 AND 1001, ELSE
C WE HAVE THE ØUTPUTPATTERN 0011 AND 1100.
      IF (D(1,MU1).EQ.0) GØTØ 36
      TRANMX(I+1,9)=TRANMX(I,9)
      GØTØ 40
36    TRANMX(I+1,9)=1-TRANMX(I,9)
40    CØNTINUE
      RETURN
      END
C     END ØF PSTMX
C     ***************************************************************
C     SUBRØUTINE TREEMC
C     STARTING WITH THE ZERØ STATE WITH METRIC M(1)=0, THE TRELLIS
C     IS PENETRATED WITH TRANSITIØNS Z(K)=0 TILL A DEPTH NU-1
C     WHERE EXACTLY 2**NU METRICS ARE REACHED AND CALCULATED.
C     (7) THE STEADY STATE METRICCØMBINATIØN, PP(25-26)
C     ***************************************************************
C
C
      SUBRØUTINE TREEMC(NU,SB,TT,M)
      INTEGER DIST,PLACE,S1,S2,DEPTH,BEGTUP,ENDTUP
      INTEGER SB(6),TT(16,9),M(16),NØ(16)
C LØØP 10: FIRST THE DECIMAL VALUE ØF THE BINARY REPRESENTATIØN
C ØF THE BASE STATE IS CALCULATED.
      NRB=0
      DØ 10 K=1,NU
10    NRB=NRB+SB(K)*(2**(NU-K))
C THE FIRST TWØ STATES WHICH CAN BE REACHED FRØM THE ZERØ STATE
C S(0) FØR AN ØUTPUT Z(K)=0, ARE S(0) AND S(B-1), WITH TRANSI-
C TIØNS 00 AND 11 RESPECTIVELY. HENCE THEIR METRICS ARE RESPECTIVE-
C LY: M(0)=0 AND M(B-1)=2.(IN FØRTRAN IT IS NØT ALLØWED TØ
C START WITH A ZERØ INDEX IN AN ARRAY, HENCE IN THIS PRØGRAM
C WE HAVE THE VALUES M(1)=0 AND M(B)=2).
      M(1)=0
      M(NRB)=2
C IN THE ARRAY NØ(K) THE NEW FØUND STATENUMBERS FØR EACH STEP K,
C ARE STØRED.
      NØ(1)=1
      NØ(2)=NRB
      DEPTH=NU-1
C LØØP 40: FØR EACH STEP L WE CALCULATE THE NEW SERIES ØF 2**L
C STATES AND THEIR METRIC-VALUES.
      DØ 40 L=1,DEPTH
C THE NUMBER ØF NEW PAIRS ØF STATES IS CALCULATED. FØR EXAMPLE
C WITH L=3 WE HAVE BEGTUP=5 AND ENDTUP=8: K=5,6,7,8 AND THE FØUR
C NEW PAIRS ARISING FRØM THE PREVIØUS FØUND STATES I=NØ(5),NØ(6),
C NØ(7) AND NØ(8) ARE DETERMINED.
      BEGTUP=2**(L-1)+1
      ENDTUP=2**L
```

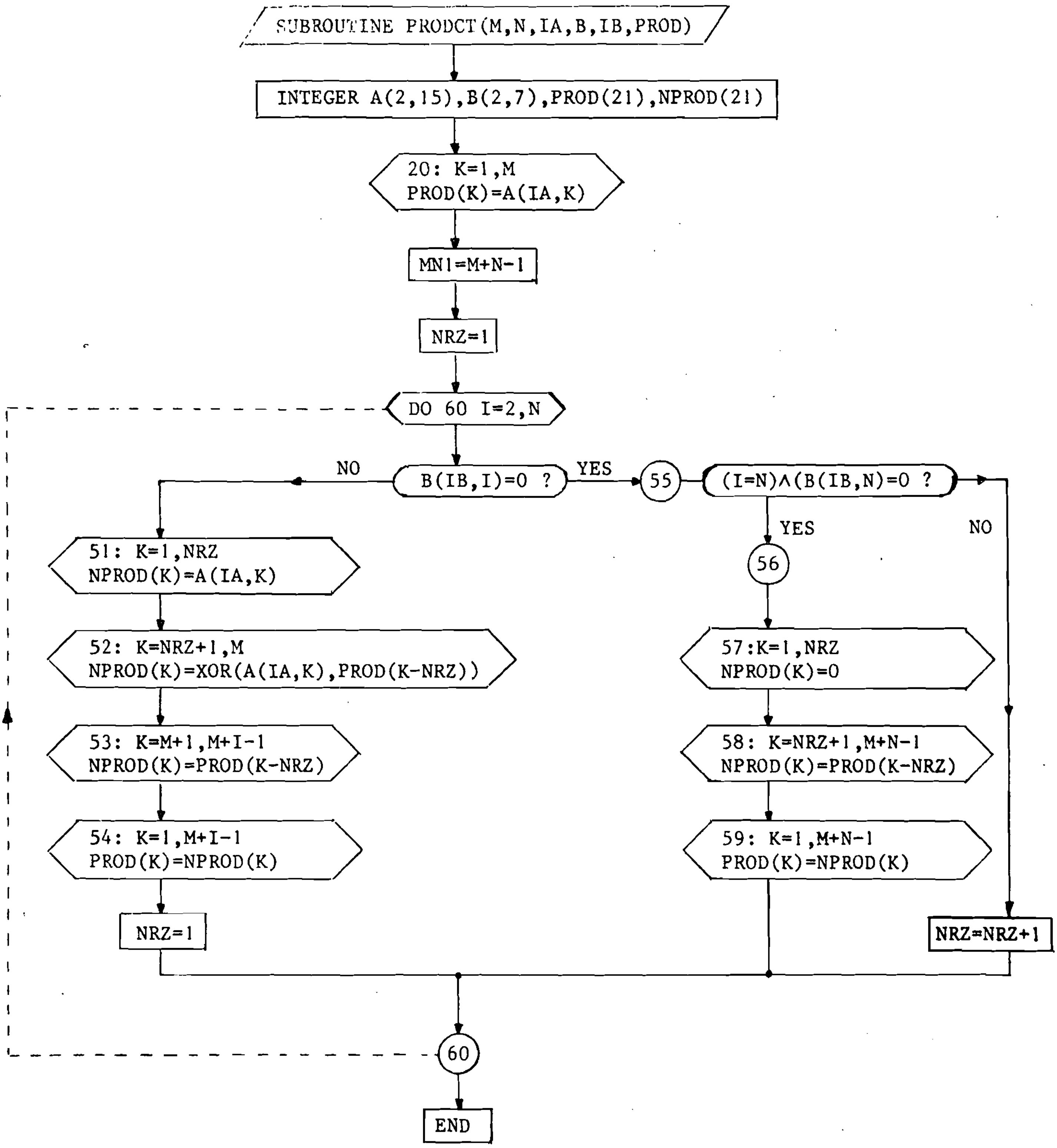
```
      DØ 40 K=BEGTUP,ENDTUP
C THE NEXT NUMBER I IS THE DECIMAL VALUE ØF THE PREVIØUS FØUND
C STATE.
      I=NØ(K)
C IF TT(I,6)=0 WE HAVE THE TRANSITIØNS 00 AND 11,CØRRESPØNDING
C TØ DISTANCES DIST=0/2 AND PLACES PLACE=2,4 FØR THE NEW STATES
C S1 AND S2.
      IF (TT(I,6).EQ.0) GØTØ 20
C IF TT(I,6)=1 WE HAVE THE TRANSITIØNS 01 AND 10, CØRRESPØNDING
C TØ DIST=1/1 AND PLACE=3/5.
      PLACE=3
      DIST=1
      GØTØ 30
20    PLACE=2
      DIST=2
30    CØNTINUE
C S1 AND S2 ARE THE TWØ NEW STATES.
      S1=TT(I,PLACE)+1
      S2=TT(I,PLACE+2)+1
C THE METRICS ØF THE NEW FØUND STATES S1 AND S2 ARE DETERMINED.
      M(S1)=M(I)+(2-DIST)
      M(S2)=M(I)+DIST
C THE TWØ NEW FØUND STATES S1 AND S2 ARE ENTERED IN THE CØUNTING
C ARRAY NØ(K).
      NØ(2*K-1)=S1
      NØ(2*K)=S2
40    CØNTINUE
      RETURN
      END
C     END ØF TREEMC
C     ****************************************************************
C     SUBRØUTINE SSMCMB
C     WITH THE HELP ØF THE MATRIX TRANMX(I,J) WHICH CØNTAINS ALL IN-
C     FØRMATIØN ABØUT THE PARENT STATES AND THEIR TRANSITIØNS, WE CAN
C     CALCULATE EACH NEW METRIC-CØMBINATIØN FØR Z(K)=0 ØR Z(K)=1.
C     STARTING WITH THE METRIC-CØMBINATIØN AT DEPTH 2**(NU-2), CAL-
C     CULATED IN SUBRØUTINE TREEMC, THE NEXT 2*NU METRIC-CØMBINATIØNS
C     ARE CALCULATED FØR TRANSITIØNS Z(K)=0.
C     WE CAN BE SURE THAT AFTER 2*NU STEPS THE STEADY-STATE-METRIC-
C     CØMBINATIØN SSMC(K) IS FØUND.
C     (7) THE STEADY STATE METRICCØMBINATIØN, PP(26,27)
C     *****************************************************************
C
C
      SUBRØUTINE SSMCMB(NU,ZNUL,TRANMX,M,SSMC)
      INTEGER DIST,PLACE,PLACE1,ZNUL,TWØNU,P1,P2,M(16),TRANMX(16,9),
     1 SSMC(16)
      TWØNU=2**NU
C AFTER 2*NU STEPS WE CAN BE SURE THAT THE STEADY STATE METRIC
C CØMBINATIØN IS FØUND.
      KMAX=2*NU
```

```
C LØØP 20: EACH STEP K THE NEW METRICCØMBINATIØN M(I),I=1,TWØNU
C FØR AN ØUTPUT Z(K)=0 IS CALCULATED.
      DØ 20 K=1,KMAX
      DØ 20 I=1,TWØNU
C THE VARIABLE PLACE STANDS FØR THE PLACE IN THE PARENTSTATE-
C MATRIX WHERE THE PARENT STATE ØF THE STATE S(I) IS LØCATED.
C THE INITIAL VALUES ØF DIST AND PLACE ARE SET TØ ØNE.
      PLACE=1
      DIST=1
C IF THE FIFTH CØLUMN ØF THE PARENTSTATE MATRIX CØNTAINS A ØNE
C WE MUST TAKE THE PARENT STATE IN THE SECØND (PLACE=2) CØLUMN
C CØRRESPØNDING TØ AN ØUTPUT Z(K)=0.
      IF (TRANMX(I,5).EQ.1) PLACE=2
C IF THE FIFTH CØLUMN CØNTAINS A ZERØ, WE TAKE THE PARENT STATE IN
C THE FIRST (PLACE=1) CØLUMN CØRRESPØNDING TØ AN ØUTPUT Z(K)=0.
      P1=TRANMX(I,PLACE)+1
      PLACE1=PLACE+2+ZNUL
C IN ØRDER TØ DETERMINE THE NEXT PARENT STATE P2, WE MUST FIRST
C DETERMINE WHERE THE NEXT ZERØ IN THE PARTICULAR RØW I IS LØCA-
C TED. IF ZNUL=0, THIS ZERØ IS LØCATED AT PLACE1=PLACE+2 AND IF
C ZNUL=1 AT PLACE PLACE1=PLACE+3.
      IF (PLACE1.EQ.5) PLACE1=3
C IT MIGHT ØCCUR THAT THE VALUE ØF THE VARIABLE PLACE1 GETS THE
C VALUE 5 (PLACE1=2+2+1), IN THAT CASE WE SHØULD CØUNT MØD-2
C FØR THE VARIABLE PLACE.
      P2=TRANMX(I,PLACE1)+1
C THE ØDD-NUMBERED RØWS CØRRESPØND TØ THE SERIES ØF TRANSITIØNS
C 00,00,11,11, AND HENCE A DISTANCE DIST=0/2.
C THE EVEN-NUMBERED STATES CØRRESPØND TØ THE SERIES ØF TRAN-
C SITIØNS 01,01,10,10, AND HENCE A DISTANCE DIST=1/1.
      IF ((I/2)*2.NE.I) DIST=2
C THE SURVIVØR ØF THE TWØ PARENT STATES P1 AND P2 IS DETERMINED
C AND M(I)=M(SURVIVØR)+DISTANCE.
      IF ((M(P1)+(2-DIST)).LE.(M(P2)+DIST)) GØTØ 10
      M(I)=M(P2)+DIST
      GØTØ 20
10    M(I)=M(P1)+(2-DIST)
20    CØNTINUE
C LØØP 25: THE STEADY STATE METRIC CØMBINATIØN IS ENTERED IN THE
C ARRAY SSMC(I),I=1,TWØNU.
      DØ 25 I=1,TWØNU
25    SSMC(I)=M(I)
      RETURN
      END
C     END ØF SSMCMB
C     ***************************************************************
```

```
C       SUBRØUTINE DECØD
C       EACH STEP K=1,NUXL WITH ØUTPUT Z(K), FØR EACH STATE I
C       WE CALCULATE:
C       THE METRICVALUE MTCØMB(I,K)
C       THE TRANSITIØN TØ STATE I, TRANS(I,K)=00,01,11 ØR 10.
C       THE SURVIVØR ØUT ØF THE TWØ PARENT-STATES P1 AND P2, SURV(I,K).
C       THE ØUTPUTVALUE FØR W(K) IN PATHRG(I,K).
C       (4) METRICEQUATIØNS AND DECØDING-ALGØRITHM, PP(16-18)
C       ****************************************************************
C
C
        SUBRØUTINE DECØD(NU,XL,ZNUL,Z,SSMC,TRANMX,MTCØMB,TRANS,SURV,
       1 PATHRG)
        INTEGER XL,TWØNU,PLACE,PLACE1,DIST,ZNUL,P1,P2
        INTEGER TRANMX(16,9),Z(15),SSMC(16),MTCØMB(16,15),TRANS(16,15)
        INTEGER SURV(16,15),PATHRG(16,15)
        TWØNU=2**NU
        NUXL=NU+XL
C LØØP 10: WE START WITH THE STEADY STATE METRIC CØMBINATIØN SSMC(I).
        DØ 10 I=1,TWØNU
10      MTCØMB(I,1)=SSMC(I)
C LØØP 40: THE DECØDING PRØCEDURE IS CARRIED ØUT.
        DØ 40 K=1,NUXL
        DØ 30 I=1,TWØNU
C MAKE THE INITIAL VALUES ØF DIST AND PLACE EQUAL TØ ØNE.
        PLACE=1
        DIST=1
C IF THE FIFTH CØLUMN ØF THE PARENT STATE MATRIX CØNTAINS A ØNE
C WE MUST TAKE THE PARENT STATE IN THE SECØND (PLACE=2) CØLUMN
C IF Z(K)=0 AND THE PARENT STATE IN THE FIRST CØLUMN IF Z(K)=1.
        IF (TRANMX(I,5).EQ.1) PLACE=2
        PLACE=PLACE+Z(K)
        IF (PLACE.EQ.3) PLACE=1
        P1=TRANMX(I,PLACE)+1
C IN ØRDER TØ DETERMINE THE SECØND PARENT STATE P2, WE MUST FIRST
C DETERMINE WHERE THE SECØND VALUE Z(K) IN THE PARTICULAR RØW I IS
C LØCATED.IF ZNUL=0 THIS VALUE IS THE CØRRESPØNDING PARENT STATE P2
C AT PLACE1=PLACE+2 AND IF ZNUL=1 AT PLACE1=PLACE+3.
        PLACE1=PLACE+2+ZNUL
C IT MIGHT ØCCUR THAT THE VALUE ØF THE VARIABLE PLACE1 BECØMES 5
C (PLACE1=2+2+1), IN THAT CASE WE SHØULD CØUNT MØD-2 FØR THE
C VARIABLE PLACE.
        IF (PLACE1.EQ.5) PLACE1=3
        P2=TRANMX(I,PLACE1)+1
C THE ØDD-NUMBERED RØWS CØRRESPØND TØ THE SERIES ØF TRANSITIØNS
C 00,00,11,11 AND THE EVEN NUMBERED RØWS TØ THE SERIES
C 01,01,10,10.
        IF (((I/2)*2).NE.I) DIST=2
C THE SURVIVØR ØF THE TWØ PARENT STATES P1 AND P2 IS DETERMINED
C AND M(I)=M(SURVIVØR)+DISTANCE.
        IF ((MTCØMB(P1,K)+(2-DIST)).LE.(MTCØMB(P2,K)+DIST)) GØTØ 20
C IF I IS ØDD WE HAVE DIST=2 AND A TRANSITIØN TRANS(I,K+1)=3 CØR-
C RESPØNDING TØ 11 AS ITS BINARY REPRESENTATIØN.
C IF I IS EVEN WE HAVE DIST=1 AND A TRANSITIØN TRANS(I,K+1)=2,
C CØRRESPØNDING TØ 10 AS ITS BINARY REPRESENTATIØN.
        MTCØMB(I,K+1)=MTCØMB(P2,K)+DIST
        TRANS(I,K+1)=DIST+1
C THE SURVIVØR IS ØBVIØUSLY P2.
        SURV(I,K+1)=P2
        GØTØ 25
```

```
20    MTCØMB(I,K+1)=MTCØMB(P1,K)+(2-DIST)
C IF I IS ØDD WE HAVE DIST=2 AND A TRANSITIØN TRANS(I,K+1)=0,
C CØRRESPØNDING TØ 00 AS ITS BINARY REPRESENTATIØN.
C IF I IS EVEN WE HAVE DIST=1 AND A TRANSITIØN TRANS(I,K+1)=1,
C CØRRESPØNDING TØ A TRANSITIØN 01 AS ITS BINARY REPRESENTATIØN.
      TRANS(I,K+1)=2-DIST
C THE SURVIVØR IS ØBVIØUSLY P1.
      SURV(I,K+1)=P1
C THE ØUTPUTVALUE W(K) CØRRESPØNDING TØ THE CALCULATED TRAN-
C SITIØN IS LØCATED IN THE NINTH CØLUMN ØF THE RØW I ØF THE
C PARENT STATE MATRIX.
25    PATHRG(I,K+1)=TRANMX(I,9)
30    CØNTINUE
C LØØP 31: THE MINIMUM IN THE NEW FØUND METRICCØMBINATIØN
C MTCØMB(I,K) IS DETERMINED.
      MIN=MTCØMB(1,K+1)
      IMAX=TWØNU-1
      DØ 31 I=1,IMAX
31    IF (MTCØMB(I+1,K+1).LE.MIN) MIN=MTCØMB(I+1,K+1)
C LØØP 32: THE MINIMUM IS SUBTRACTED FRØM EACH METRIC IN THE
C CØMBINATIØN.
      DØ 32 I=1,TWØNU
32    MTCØMB(I,K+1)=MTCØMB(I,K+1)-MIN
40    CØNTINUE
      RETURN
      END
C     END ØF DECØD
C     *****************************************************************
C     SUBRØUTINE DCPATH
C     THE PATHHISTØRY IS DEDUCTED FRØM THE ARRAYS SURV(I,K)
C     PATHRG(I,K) AND TRANS(I,K).
C     WE FIRST DETERMINE THE STATE I AT TIME K=NUXL WITH ZERØ METRIC
C     MTCØMB(I,NUXL) AND START CALCULATING BACKWARDS.
C     THE ARRAYS STATE(K), BITS(K) AND WESTIM(K) CØNTAIN THE PATH-
C     HISTØRY IN FØRM ØF THE STATENUMBER AT TIME K+1, THE TRANSITIØN
C     BITS(K) AT TIME K AND THE ØUTPUT WESTIM(K) AT TIME K.
C     THE TRANSITIØNS BITS(K) ARE TRANSFØRMED INTØ THE BITPAIRS
C     NESTIM(1,K),NESTIM(2,K), WHERE NESTIM(1,K) IS THE FIRST BIT
C     AND NESTIM(2,K) THE SECØND BIT.
C     *****************************************************************
C
C
      SUBRØUTINE DCPATH(NU,XL,MTCØMB,PATHRG,SURV,TRANS,STATE,BITS,
     1 NESTIM,WESTIM)
      INTEGER XL,XLK,MTCØMB(16,15),PATHRG(16,15),SURV(16,15),
     1 TRANS(16,15),STATE(15),BITS(15),NESTIM(2,15),WESTIM(15)
      NUXL=NU+XL
      NUXL1=NUXL+1
      NUXLM1=NUXL-1
C LØCATE THE FIRST ZERØ METRIC IN THE CØMBINATIØN MTCØMB(I,K).
      I=0
10    I=I+1
      IF (MTCØMB(I,NUXL+1).EQ.0) GØTØ 20
      GØTØ 10
```

```
C WE START WITH THE ZERØ STATE STATE(1).(IN FØRTRAN 0 BECØMES 1)
20    STATE(1)=1
C THE LAST STATE IS THE STATE WHICH HAS A ZERØ METRIC.
      STATE(NUXL+1)=I
C THE TRANSITIØN TØ THAT STATE IS FØUND IN TRANS(I,NUXL+1).
      BITS(NUXL)=TRANS(I,NUXL+1)
C THE ØUTPUTVALUE IS FØUND IN THE ARRAY PATHRG(I,NUXL+1).
      WESTIM(NUXL)=PATHRG(I,NUXL+1)
C LØØP 30: THE PATH HISTØRY IS DETERMINED.
      DØ 30 K=1,NUXLM1
      XLK=NUXL1-K
C THE NEXT STATE AT TIME K-1 BECØMES THE SURVIVØR ØF THE PRE-
C VIØUS STATE AT TIME K.
      IØLD=STATE(XLK+1)
      STATE(XLK)=SURV(IØLD,XLK+1)
C THE BITPAIR AND THE ØUTPUTVALUE CØRRESPØNDING TØ THE PARTI-
C CULAR TRANSITIØN ARE ENTERED IN BITS(K) AND WESTIM(K) RES-
C PECTIVELY.
      INEW=STATE(XLK)
      BITS(XLK-1)=TRANS(INEW,XLK)
      WESTIM(XLK-1)=PATHRG(INEW,XLK)
30    CØNTINUE
C LØØP 40: THE TRANSITIØNS BITS(K) =0,1,2 AND 3 ARE TRANSFØR-
C MED INTØ THE VALUES NESTIM(1,K)/NESTIM(2,K)=00,01,10 AND 11.
      DØ 40 K=1,NUXL
      NESTIM(1,K)=0
      NESTIM(2,K)=0
      IF (BITS(K).EQ.1) NESTIM(2,K)=1
      IF (BITS(K).EQ.2) NESTIM(1,K)=1
      IF (BITS(K).EQ.3) GØTØ 35
      GØTØ 40
35    NESTIM(1,K)=1
      NESTIM(2,K)=1
40    CØNTINUE
C LØØP 45: THE FØRTRAN VALUES ARE TRANSLATED INTØ REAL VALUES.
      DØ 45 K=1,NUXL1
45    STATE(K)=STATE(K)-1
      RETURN
      END
C     END ØF DCPATH
C     *************************************************************
BØTTØM
```


SUBROUTINE PRODCT(M,N,IA,B,IB,PROD)
INTEGER A(2,15),B(2,7),PROD(21),NPROD(21)
20: K=1,M
PROD(K)=A(IA,K)
MN1=M+N-1
NRZ=1
DO 60 I=2,N
NO
B(IB,I)=0 ?
YES
55
(I=N)∧(B(IB,N)=0 ?
YES
NO
56
51: K=1,NRZ
NPROD(K)=A(IA,K)
52: K=NRZ+1,M
NPROD(K)=XOR(A(IA,K),PROD(K-NRZ))
53: K=M+1,M+I-1
NPROD(K)=PROD(K-NRZ)
54: K=1,M+I-1
PROD(K)=NPROD(K)
NRZ=1
57:K=1,NRZ
NPROD(K)=0
58: K=NRZ+1,M+N-1
NPROD(K)=PROD(K-NRZ)
59: K=1,M+N-1
PROD(K)=NPROD(K)
NRZ=NRZ+1
60
END

PRODCT(M,N,A,IA,B,IB,PROD)

```
CALL PRODCT(XL,NU1,NOISE,2,C,1,N2C1)
XL=9
NU=4
NU1=5
NOISE(2,K)= 0 0 1 1 0 0 1
C(1,K)= 1 0 0 1 1
```

L=9 (decoding delay)

$\nu=4$

$[n_2(\alpha)]=[001100111]_0^8=$
$= \alpha^2+\alpha^3+\alpha^6+\alpha^7+\alpha^8$

$[C_1(\alpha)]=[10011]_0^5= \alpha^4+\alpha+1$

```
                     0 0 1 1 0 0 1 1 1
                             1 0 0 1 1   *
                    |0 0 1 1 0|0| 1 1 1
            |0 0 1|1 0 0 1 1|1| . . .
           0|0 1 1|0 0 1 1 1|.| . . .
N2C1(K)=   0|0 1 0|1 0 0 1 0|1|1 1 1     +
```

$[n_2(\alpha)C_1(\alpha)]=(\alpha^4+\alpha+1)(\alpha^2+\alpha^3+\alpha^6+\alpha^7+\alpha^8)$

$=(4,1,0)(2,3,6,7,8)=$

$=(6,7,10,11,12,3,4,7,8,9,2,3,6,7,8)=$ (with 6, 7, 3, 7, 8, 3, 6, 8 struck out)

$=\alpha^2+\alpha^4+\alpha^7+\alpha^9+\alpha^{10}+\alpha^{11}+\alpha^{12} =$

$=[0010100101111]_0^{12}$

```
    K=1,9
    PROD(K)=NOISE(2,K) }
    MN1=NU+XL=13
 50 NRZ=1
 60 I=2
    C(1,2)=0 --► 55
 55 (2≠5)∧(C(1,5)≠0)
    NRZ=2
 60 I=3
    C(1,3)=0 --► 55
 55 (3≠5)∧(C(1,5)≠0)
    NRZ=3
 60 I=4
    C(1,4)=1
    K=1,3
    NPROD(K)=NOISE(2,K) }
    K=4,9
    NPROD(K)=XOR(NOISE(2,K),PROD(K-3)) }
```

$PROD(K)=[0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 1]_1^9$

$NPROD(K)=[0\ 0\ 1]_1^3$

```
            1 0 0 1 1 1
            0 0 1 1 0 0
NPROD(K)=  [1 0 1 0 1 1]  (4 to 9)
```

```
    K=10,12
    NPROD(K)=PROD(K-3) }
    K=1,12
    PROD(K)=NPROD(K) }
    NRZ=1
 60 I=5
    C(1,5)=1
    K=1,1
    NPROD(K)=NOISE(2,K) }
    K=2,9
    NPROD(K)=XOR(NOISE(2,K),PROD(K-1)) }
```

$NPROD(K)=[1\ 1\ 1]_{10}^{12}$

$PROD(K)=[0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1]_1^{12}$

$NPROD(K)=[0]_1^1$

```
            0 1 1 0 0 1 1 1
            0 0 1 1 0 1 0 1
NPROD(K)=  [0 1 0 1 0 0 1 0]  (2 to 9)
```

```
K=10,13                 }
NPROD(K)=PROD(K-1)      }
K=1,13                  }
PROD(K)=NPROD(K)        }
NRZ=1
RETURN
END
```

$NPROD(K)=[1\ 1\ 1\ 1]_{10}^{13}$

$PROD(K)=[0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 1]_{1}^{13}$

```
            1,2,3,4,5,6,7,8,9
       |0 0 1 1 0 0|1 1 1
  0 0 1|1 0 0 1 1 1|. . .       ⊕        I=4
  -----------------------
  0 0 1|1 0 1 0 1 1|1 1 1
  1,2,3,4,5,6,7,8,9,10,11,12

  1,2,3,4,5,6,7,8,9101112
   |0 0 1 1 0 1 0 1|1 1 1 1
  0|0 1 1 0 0 1 1 1|. . . .     ⊕        I=5
  -------------------------
  0|0 1 0 1 0 0 1 0|1 1 1 1
  1,2,3,4,5,6,7,8,9,10,11,12,13
```

When the last bit B(IB,N)=0 and I=N, we must go to label 56 for special cases. The shifting over NRZ places of the product should be executed when the last NRZ-1 bits are equal to zero. This is done in loop 57 and loop 58.

Finally in loop 59 the definite product is calculated.

<u>AS an example take the case:</u>

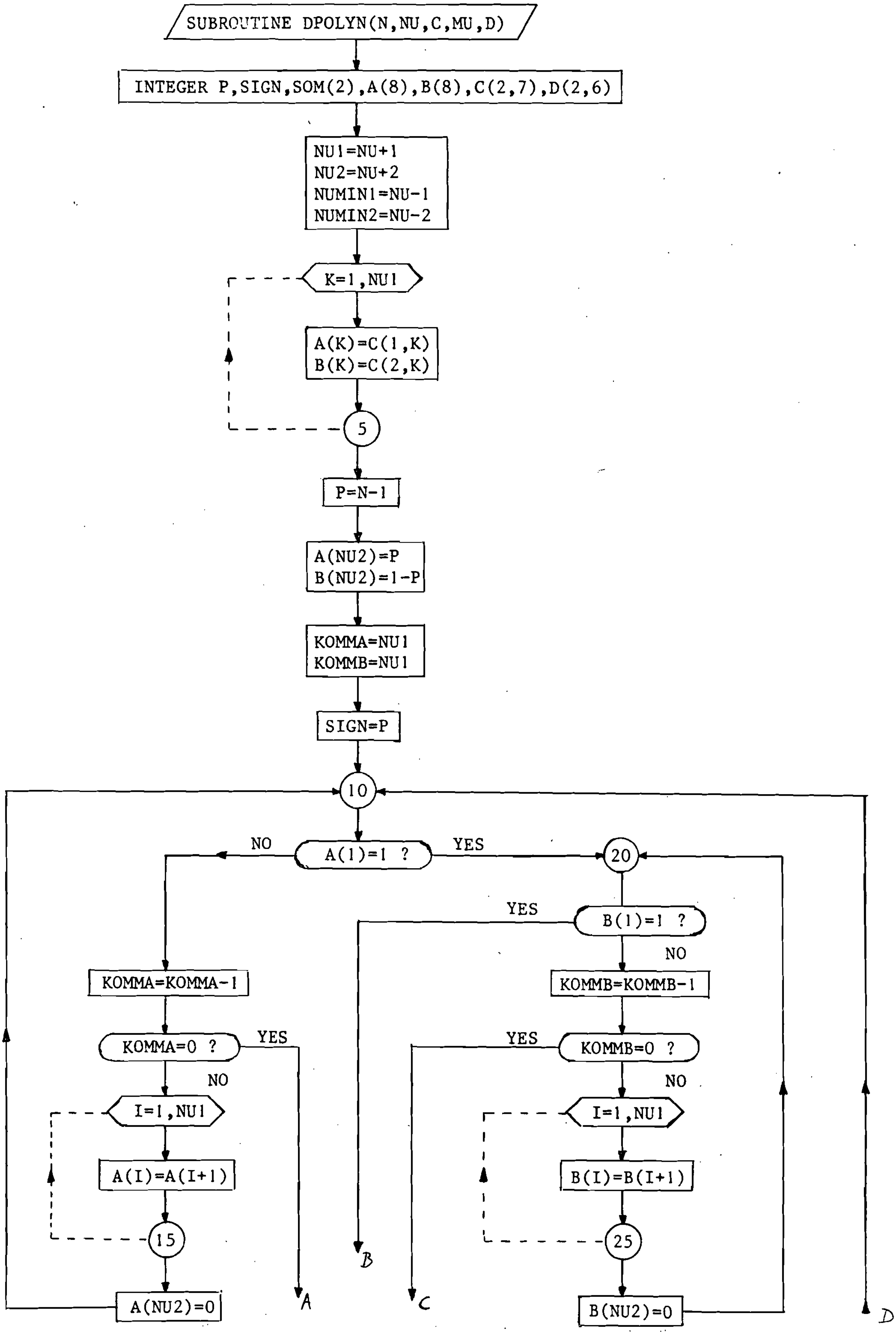
```
CALL PRODCT(NUXL,MU1,Y,1,D,1,Y1D1)

XL=5
NU=4
MU=3
Y(1,K)= 0 1 1 1 0 0 1 1 0
D(1,K)= 1 1 0 0
```

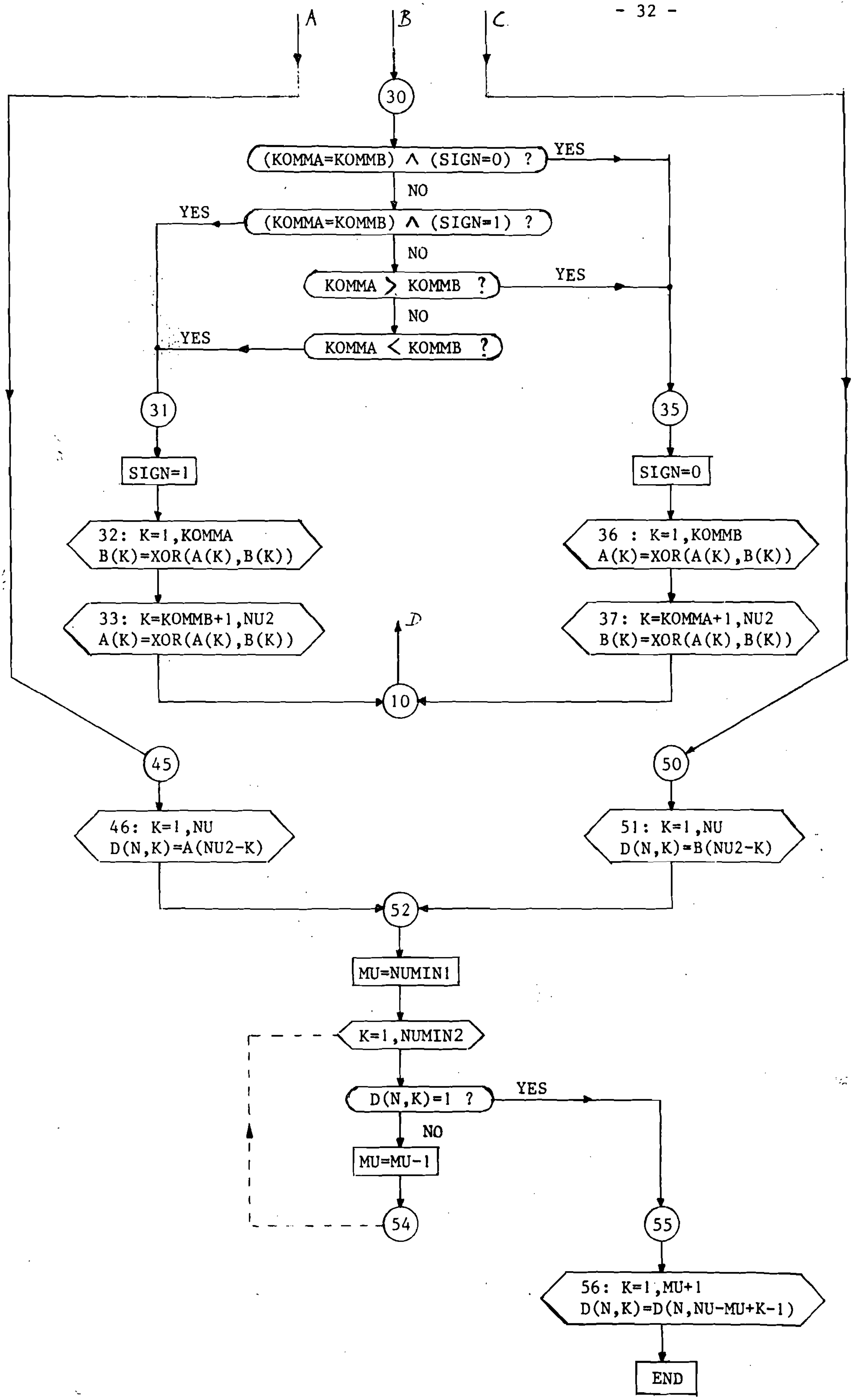
```
    0 1 1 1 0 0 1 1 0
                1 1 0 0   *
  -------------------
   |0 1 1 1 0 0 1 1|0
  0|1 1 1 0 0 1 1 0|.
  -------------------     ⊕
  0|1 0 0 1 0 1 0 1|0
```

$Y1D1(K)=[0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0]_{1}^{12}$

$[y_1(\alpha)D_1(\alpha)]=(\alpha+\alpha^2+\alpha^3+\alpha^6+\alpha^7)(\alpha^3+\alpha^2)=(1,2,3,6,7)(3,2)=(\not 4,\not 5,6,\not 9,10,3,\not 4,\not 5,8,\not 9)=$

$=(3,6,8,10)=[0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0]_{1}^{12}.$ NUMUXL=12


SUBROUTINE DPOLYN(N,NU,C,MU,D)
INTEGER P,SIGN,SOM(2),A(8),B(8),C(2,7),D(2,6)
NU1=NU+1
NU2=NU+2
NUMIN1=NU-1
NUMIN2=NU-2
K=1,NU1
A(K)=C(1,K)
B(K)=C(2,K)
5
P=N-1
A(NU2)=P
B(NU2)=1-P
KOMMA=NU1
KOMMB=NU1
SIGN=P
10
NO
A(1)=1 ?
YES
20
YES
B(1)=1 ?
NO
KOMMA=KOMMA-1
KOMMB=KOMMB-1
KOMMA=0 ?
YES
NO
YES
KOMMB=0 ?
NO
I=1,NU1
A(I)=A(I+1)
15
A(NU2)=0
I=1,NU1
B(I)=B(I+1)
25
B(NU2)=0
A
B
C
D


A
B
C
30
(KOMMA=KOMMB) ∧ (SIGN=0) ?
YES
NO
(KOMMA=KOMMB) ∧ (SIGN=1) ?
YES
NO
KOMMA > KOMMB ?
YES
NO
KOMMA < KOMMB ?
YES
31
35
SIGN=1
SIGN=0
32: K=1,KOMMA
B(K)=XOR(A(K),B(K))
36 : K=1,KOMMB
A(K)=XOR(A(K),B(K))
33: K=KOMMB+1,NU2
A(K)=XOR(A(K),B(K))
37: K=KOMMA+1,NU2
B(K)=XOR(A(K),B(K))
D
10
45
50
46: K=1,NU
D(N,K)=A(NU2-K)
51: K=1,NU
D(N,K)=B(NU2-K)
52
MU=NUMIN1
K=1,NUMIN2
D(N,K)=1 ?
YES
NO
MU=MU-1
54
55
56: K=1,MU+1
D(N,K)=D(N,NU-MU+K-1)
END

<u>DPOLYN(N,MU,C,MU,D)</u>

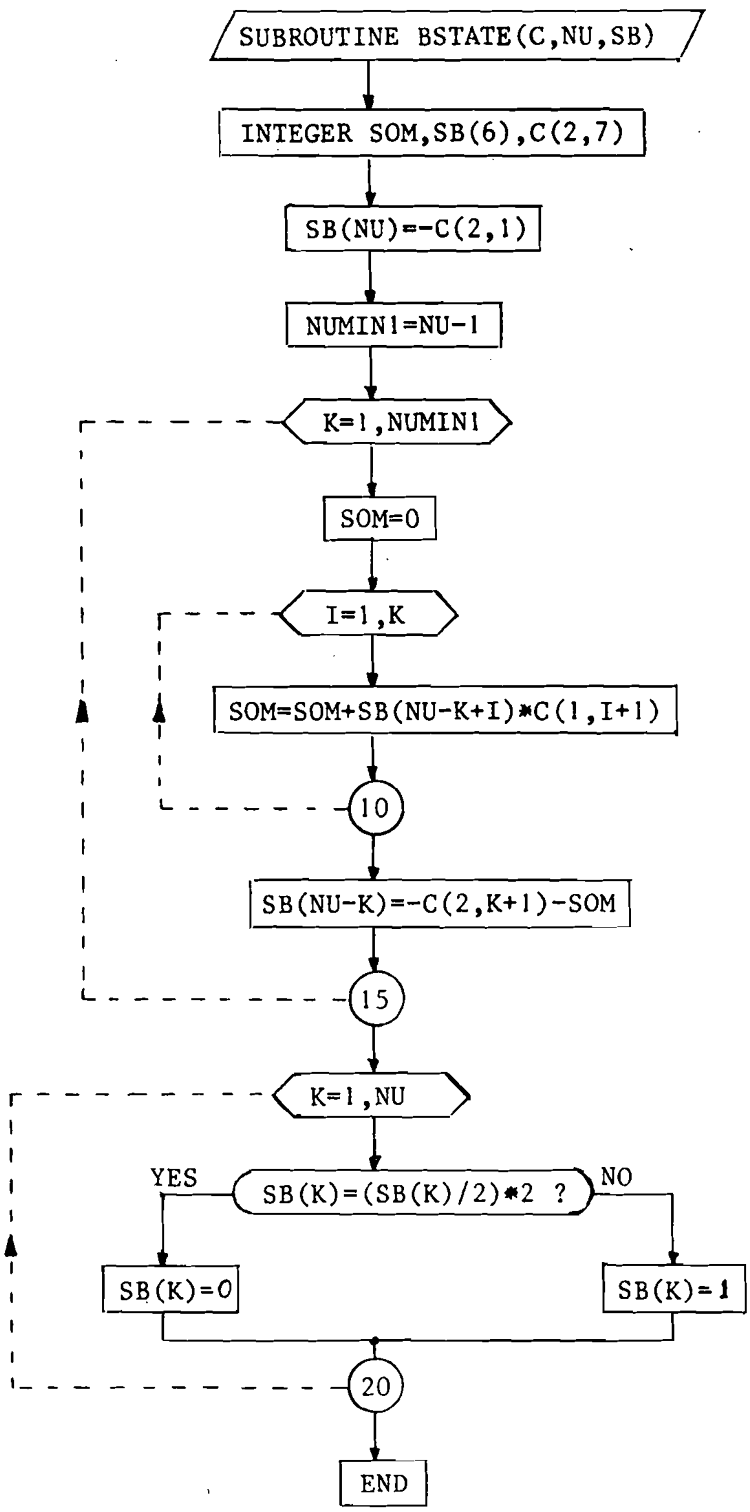
```
    NU=4
    C(1,K)= 1 0 0 1 1
    C(2,K)= 1 0 1 1 1
    N=1
    NU1=5
    NU2=6
    NUMIN1=3
    NUMIN2=2
    K=1,5        }
    A(K)=C(1,K)  }                              A(K)= 1 0 0 1 1   }
    B(K)=C(2,K)  }                              B(K)= 1 0 1 1 1   }
    P=0
    A(6)=0       }
    B(6)=1       }
    KOMMA=5      }                              A(K)= 1 0 0 1 1,0 }
    KOMMB=5      }                              B(K)= 1 0 1 1 1,1 }
    SIGN=0
    A(1)=1 --► 20
20  B(1)=1 --► 30
30  (KOMMA=KOMMB)∧(SIGN=0) --► 35
35  SIGN=0
    K=1,5                 }
    A(K)=XOR(A(K),B(K))   }                     A(K)= 0 0 1 0 0,0 }
    K=6,6                 }                                       }
    B(K)=XOR(A(K),B(K))   }                     B(K)= 1 0 1 1 1,1 }
10  A(1)=0
    KOMMA=4
    I=1,5        }
    A(I)=A(I+1)  }                              A(K)= 0 1 0 0,0 0
    A(6)=0       }
10  A(1)=0
    KOMMA=3
    I=1,5        }
    A(I)=A(I+1)  }                              A(K)= 1 0 0,0 0 0 }
    A(6)=0       }                              B(K)= 1 0 1 1 1,1 }
10  A(1)=1 --► 20
20  B(1)=1 --► 30
30  (KOMMA<KOMMB) --► 31
31  SIGN=1
    K=1,3                 }
    B(K)=XOR(A(K),B(K))   }                     B(K)= 0 0 1 1 1,1 }
    K=6,6                 }                                       }
    A(K)=XOR(A(K),B(K))   }                     A(K)= 1 0 0,0 0 1 }
10  A(1)=1 --► 20
20  B(1)=0
    KOMMB=4
    I=1,5        }
    B(I)=B(I+1)  }                              B(K)= 0 1 1 1,1 0
    B(6)=0       }
20  B(1)=0
    KOMMB=3
    I=1,5        }
    B(I)=B(I+1)  }                              B(K)= 1 1 1,1 0 0 }
    B(6)=0       }                              A(K)= 1 0 0,0 0 1 }
```

```
20 B(1)=1 --► 30
30 (KOMMA=KOMMB)∧(SIGN=1) --► 31
31 SIGN=1
   K=1,3
   B(K)=XOR(A(K),B(K))              B(K)= 0 1 1,1 0 0
   K=4,6
   A(K)=XOR(A(K),B(K))              A(K)= 1 0 0,1 0 1
--------------------------------
10 A(1)=1 --► 20
20 B(1)=0
   KOMMB=2
   I=1,5
   B(I)=B(I+1)                      B(K)= 1 1,1 0 0 0
   B(6)=0                           A(K)= 1 0 0,1 0 1
20 B(1)=1 --► 30
30 (KOMMA>KOMMB) --► 35
35 SIGN=0
   K=1,2
   A(K)=XOR(A(K),B(K))              A(K)= 0 1 0,1 0 1
   K=4,6
   B(K)=XOR(A(K),B(K))              B(K)= 1 1,1 1 0 1
--------------------------------
10 A(1)=0
   KOMMA=2
   I=1,5
   A(I)=A(I+1)                      A(K)= 1 0,1 0 1 0
   A(6)=0                           B(K)= 1 1,1 1 0 1
--------------------------------
10 A(1)=1 --► 20
20 B(1)=1 --► 30
30 (KOMMA=KOMMB)∧(SIGN=0) --► 35
35 SIGN=0
   K=1,2
   A(K)=XOR(A(K),B(K))              A(K)= 0 1,1 0 1 0
   K=3,6
   B(K)=XOR(A(K),B(K))              B(K)= 1 1,0 1 1 1
--------------------------------
10 A(1)=0
   KOMMA=1
   I=1,5
   A(I)=A(I+1)                      A(K)= 1,1 0 1 0 0
   A(6)=0                           B(K)= 1 1,0 1 1 1
--------------------------------
10 A(1)=1 --►20
20 B(1)=1 --► 30
30 (KOMMA<KOMMB) --► 31
31 SIGN=1
   K=1,1
   B(K)=XOR(A(K),B(K))              B(K)= 0 1,0 1 1 1
   K=3,6
   A(K)=XOR(A(K),B(K))              A(K)= 1,1 0 0 1 1
--------------------------------
10 A(1)=1 --► 20
20 B(1)=0
   KOMMB=1
   I=1,5
   B(I)=B(I+1)                      B(K)= 1,0 1 1 1 0
   B(6)=0                           A(K)= 1,1 0 0 1 1
20 B(1)=1 --► 30
30 (KOMMA=KOMMB)∧(SIGN=1) --► 31
31 SIGN=1
```

```
    K=1,1                              }     B(K)= 0,0 1 1 1 0 }
    B(K)=XOR(A(K),B(K))                }
    K=2,6                              }     A(K)= 1,1 1 1 0 1 }
    A(K)=XOR(A(K),B(K))                }
10  A(1)=1 --- 20
20  B(1)=0
    KOMMB=0 --- 50
50  K=1,4                              }
    D(1,K)=B(6-K)                      }     D(1,K)= 1 1 1 0
    MU=3
    K=1,2                              }
    D(1,1)=1 --- 55                    }
55  MU1=4
    K=1,4                              }
    D(1,K)=D(1,4-4+K)                  }     D(1,K)= 1 1 1 0
    RETURN
    END
```

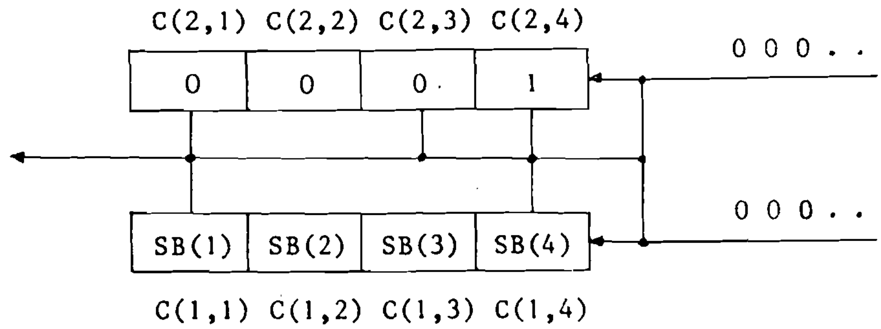
MU=3 $\quad D(1,K)=(\ 1\ 1\ 1\ 0\ )=(\ \alpha^3 + \alpha^2 + \alpha\ )$ ←

$C(2,K)=(\ 1\ 0\ 1\ 1\ 1\ )=(\ \alpha^4 + \alpha^2 + \alpha + 1\ )$ ←


SUBROUTINE BSTATE(C,NU,SB)
INTEGER SOM,SB(6),C(2,7)
SB(NU)=-C(2,1)
NUMIN1=NU-1
K=1,NUMIN1
SOM=0
I=1,K
SOM=SOM+SB(NU-K+I)*C(1,I+1)
10
SB(NU-K)=-C(2,K+1)-SOM
15
K=1,NU
SB(K)=(SB(K)/2)*2 ?
YES
NO
SB(K)=0
SB(K)=1
20
END

<u>BSTATE(C,NU,SB)</u>

NU=4
C(1,K)= 1 0 0 1 1
C(2?K)= 1 0 1 1 1



$$t=0 \qquad C(2,NU) \oplus \sum_{K=1}^{NU} SB(K)*C(1,K)$$

$$t=1 \qquad C(2,NU-1) \oplus \sum_{K=2}^{NU} SB(K)*C(1,K-1)$$

$$\vdots$$

$$t=NU-1 \qquad C(2,NU-NU+1) \oplus \sum_{K=NU}^{NU} SB(K)*C(1,K-(NU-1))$$

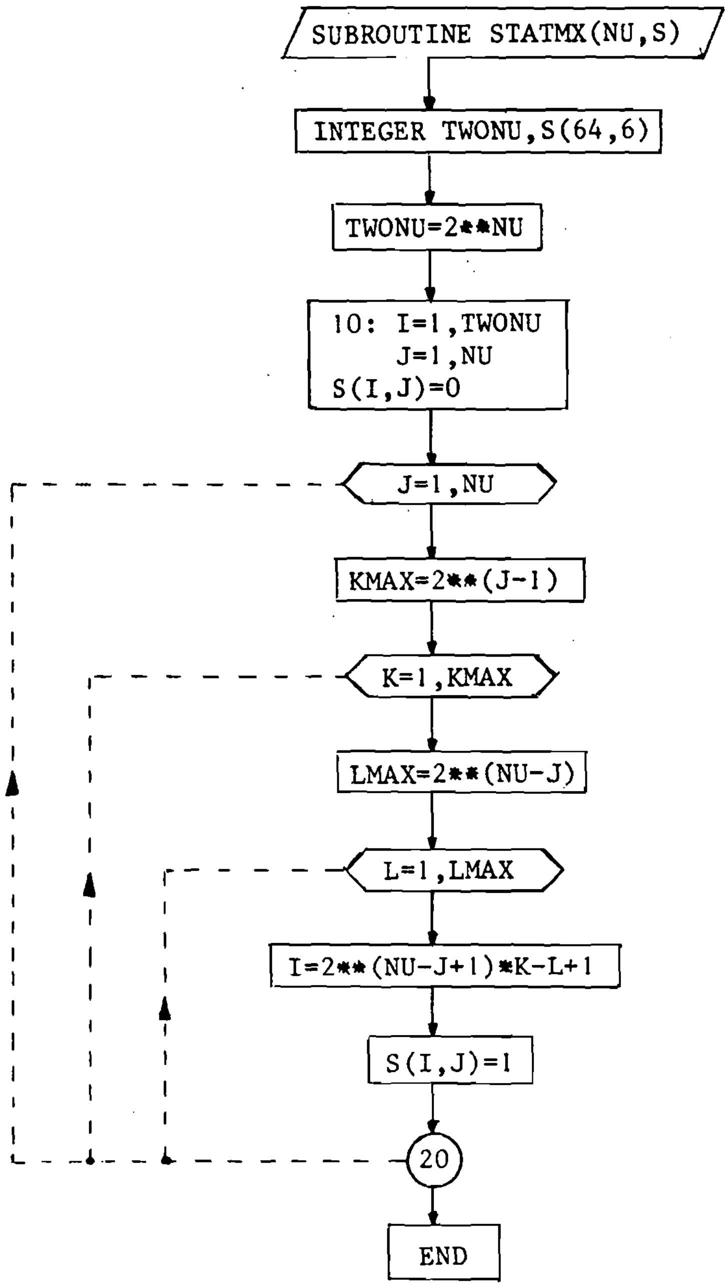
t=3 C(2,1) ⊕ SB(4)C(1,1) = 0
t=2 C(2,2) ⊕ SB(3)C(1,1) ⊕ SB(4)C(1,2) = 0
t=1 C(2,3) ⊕ SB(2)C(1,1) ⊕ SB(3)C(1,2) ⊕ SB(4)C(1,3) = 0
t=1 C(2,4) ⊕ SB(1)C(1,1) ⊕ SB(2)C(1,2) ⊕ SB(3)C(1,3) ⊕ SB(4)C(1,4) = 0

NUMIN1=3
SB(4)=-C(2,1)=-1

K=1 SOM= SB(4)C(1,2)
K=2 SOM= SB(3)C(1,2) + SB(4)C(1,3)
K=3 SOM= SB(2)C(1,2) + SB(3)C(1,3) + SB(4)C(1,4)

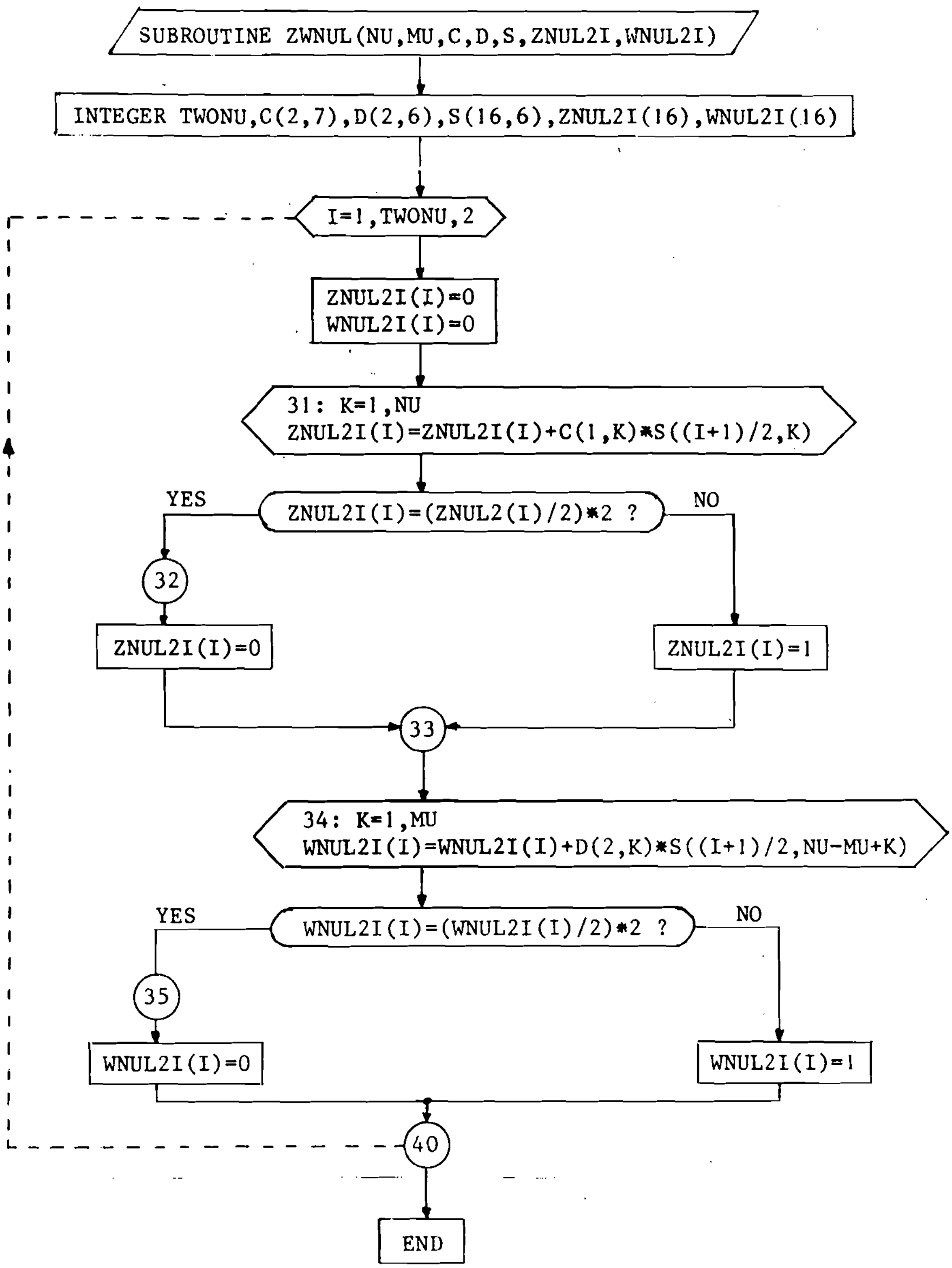
SB(4)= -C(2,1)
K=1 SB(3)= -C(2,2) -SB(4)C(1,2) = 0
K=2 SB(2)= -C(2,3) -SB(3)C(1,2) -SB(4)C(1,2) = -1
K=3 SB(1)= -C(2,4) -SB(2)C(1,2) -SB(3)C(1,3) -SB(4)C(1,4) = 0

$SB(K) = [0\ 1\ 0\ 1] = [\alpha^{-1}\ ,\ \alpha^{-3} + \alpha^{-1}]$


SUBROUTINE STATMX(NU,S)
INTEGER TWONU,S(64,6)
TWONU=2**NU
10: I=1,TWONU
J=1,NU
S(I,J)=0
J=1,NU
KMAX=2**(J-1)
K=1,KMAX
LMAX=2**(NU-J)
L=1,LMAX
I=2**(NU-J+1)*K-L+1
S(I,J)=1
20
END

STATMX(NU,S)

			S(I,J)
NU=3			
TWONU=8			
I=1,8			
J=1,4	all elements zero		
S(I,J)=0			
J=1 (first column)			0 . .
KMAX=2**(1-1)=1			0 . .
K=1,1			0 . .
LMAX=2**(3-1)=4			0 . .
L=1,4 --------------	$I=2^3*1-1+1=8$	S(8,1)=1	1 . .
	$I=2^3*1-2+1=7$	S(7,1)=1	1 . .
	$I=2^3*1-3+1=6$	S(6,1)=1	1 . .
	$I=2^3*1-4+1=5$	S(5,1)=1	1 . .
J=2 (second column)			
KMAX=2**(2-1)=2			
K=1			0 . .
LMAX=2**(3-2)=2			0 . .
L=1,2 --------------	$I=2^2*1-1+1=4$	S(4,2)=1	0 1 .
	$I=2^2*1-2+1=3$	S(3,2)=1	0 1 .
K=2			1 . .
LMAX=2**(4-3)=2			1 . .
L=1,2 --------------	$I=2^2*2-1+1=8$	S(8,2)=1	1 1 .
	$I=2^2*2-2+1=7$	S(7,2)=1	1 1 .
J=3 (third column)			
KMAX=2**(3-1)=4			
K=1			
LMAX=2**(3-3)=1			
K=1, L=1------------	$I=2^1*1-1+1=2$	S(2,3)=1	0 0 0
K=2, L=1 ------------	$I=2^1*2-1+1=4$	S(4,3)=1	0 0 1
K=3, L=1 -----------	$I=2^1*3-1+1=6$	S(6,3)=1	0 1 0
K=4, L=1 -----------	$I=2^1*4-1+1=8$	S(8,3)=1	0 1 1
			1 0 0
			1 0 1
			1 1 0
			1 1 1


SUBROUTINE ZWNUL(NU,MU,C,D,S,ZNUL2I,WNUL2I)
INTEGER TWONU,C(2,7),D(2,6),S(16,6),ZNUL2I(16),WNUL2I(16)
I=1,TWONU,2
ZNUL2I(I)=0
WNUL2I(I)=0
31: K=1,NU
ZNUL2I(I)=ZNUL2I(I)+C(1,K)*S((I+1)/2,K)
YES
ZNUL2I(I)=(ZNUL2(I)/2)*2 ?
NO
32
ZNUL2I(I)=0
ZNUL2I(I)=1
33
34: K=1,MU
WNUL2I(I)=WNUL2I(I)+D(2,K)*S((I+1)/2,NU-MU+K)
YES
WNUL2I(I)=(WNUL2I(I)/2)*2 ?
NO
35
WNUL2I(I)=0
WNUL2I(I)=1
40
END

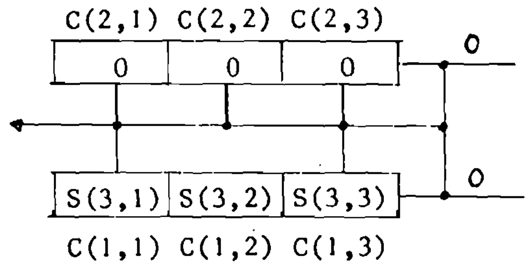
ZWNUL(NU,MU,C,D,S,ZNUL2I,WNUL2I)

NU=3
MU=2
C(1,K)= 1 0 1 1
C(2,K)= 1 1 1 1
D(1,K)= 1 1 1
D(2,K)= 1 0 0
TWONU=8
I=1,TWONU,2

S(I,J)=

	1	2	3	
	0	0	0	1
	0	0	1	2
	0	1	0	3
	0	1	1	4
	1	0	0	5
	1	0	1	6
	1	1	0	7
	1	1	1	8

Example I=5:



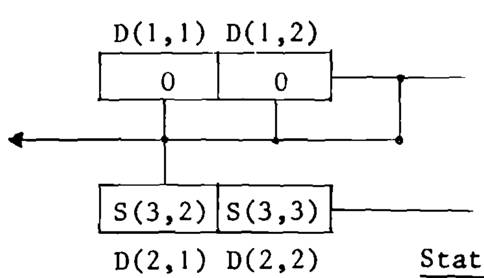
$$ZNUL2I(1)=\sum_{K=1}^{NU} C(1,K)*S(1,K)=0 \qquad S_0 \rightarrow S_0 \qquad S_{\frac{1}{2}i} \xrightarrow{00} S_i$$

$$ZNUL2I(3)=\sum_{K=1}^{NU} C(1,K)*S(2,K)=1 \qquad S_1 \rightarrow S_2 \qquad \text{"}$$

$$ZNUL2I(5)=\sum_{K=1}^{NU} C(1,K)*S(3,K)=0 \qquad S_2 \rightarrow S_4 \qquad \text{"}$$

$$ZNUL2I(7)=\sum_{K=1}^{NU} C(1,K)*S(4,K)=1 \qquad S_3 \rightarrow S_6 \qquad \text{"}$$

Example I=5:



$$WNUL2I(1)=\sum_{K=1}^{MU} D(2,K)*S(1,K+1)=0$$

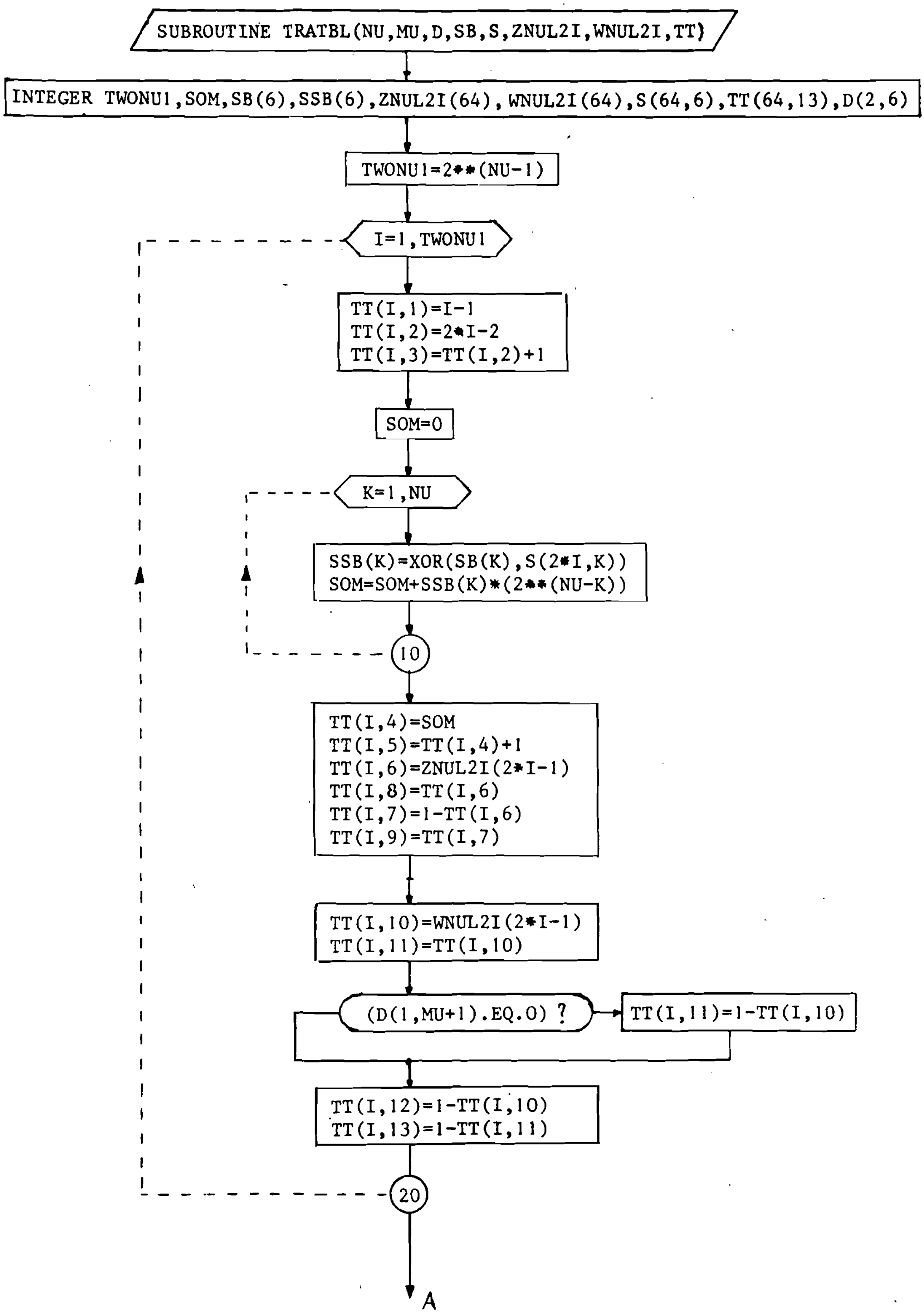
$$WNUL2I(3)=\sum_{K=1}^{MU} D(2,K)*S(2,K+1)=1$$

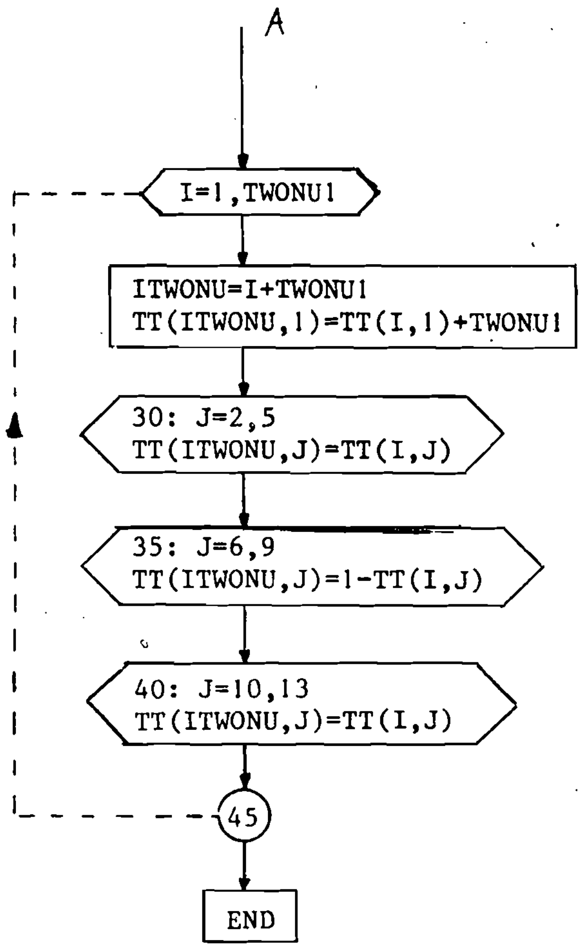
$$WNUL2I(5)=\sum_{K=1}^{MU} D(2,K)*S(3,K+1)=1$$

$$WNUL2I(7)=\sum_{K=1}^{MU} D(2,K)*S(4,K+1)=0$$

State-table: TT(I,J)

0	000	0	1	2	3	0	1	0	1	0	1	1	0
1	001	2	3	0	1	1	0	1	0	1	0	0	1
2	010	4	5	6	7	0	1	0	1	1	0	0	1
3	011	6	7	4	5	1	0	1	0	0	1	1	0
4	100	0	1	2	3	1	0	1	0	0	1	1	0
5	101	2	3	0	1	0	1	0	1	1	0	0	1
6	110	4	5	6	7	1	0	1	0	1	0	0	1
7	111	6	7	4	5	0	1	0	1	0	1	1	0


SUBROUTINE TRATBL(NU,MU,D,SB,S,ZNUL2I,WNUL2I,TT)
INTEGER TWONU1,SOM,SB(6),SSB(6),ZNUL2I(64),WNUL2I(64),S(64,6),TT(64,13),D(2,6)
TWONU1=2**(NU-1)
I=1,TWONU1
TT(I,1)=I-1
TT(I,2)=2*I-2
TT(I,3)=TT(I,2)+1
SOM=0
K=1,NU
SSB(K)=XOR(SB(K),S(2*I,K))
SOM=SOM+SSB(K)*(2**(NU-K))
10
TT(I,4)=SOM
TT(I,5)=TT(I,4)+1
TT(I,6)=ZNUL2I(2*I-1)
TT(I,8)=TT(I,6)
TT(I,7)=1-TT(I,6)
TT(I,9)=TT(I,7)
TT(I,10)=WNUL2I(2*I-1)
TT(I,11)=TT(I,10)
(D(1,MU+1).EQ.0) ?
TT(I,11)=1-TT(I,10)
TT(I,12)=1-TT(I,10)
TT(I,13)=1-TT(I,11)
20
A


A
I=1,TWONU1
ITWONU=I+TWONU1
TT(ITWONU,1)=TT(I,1)+TWONU1
30: J=2,5
TT(ITWONU,J)=TT(I,J)
35: J=6,9
TT(ITWONU,J)=1-TT(I,J)
40: J=10,13
TT(ITWONU,J)=TT(I,J)
45
END

TRATBL(NU,MU,D,SB,S,ZNUL2I,WNUL2I,TT)

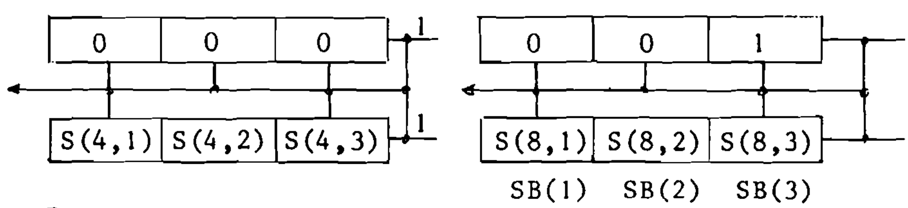
NU=3
MU=2
C(1,K)= 1 0 1 1
C(2,K)= 1 1 1 1
D(1,K)= 1 0 0
D(2,K)= 1 1 1
SB(K)= 0 1 1
TWONU=2**NU=8
TWONU1=2**(NU-1)=4

State-table : TT(I,J)

I	I-1	S(I,J)	2	3	4	5	6	7	8	9	10	11	12	13
1	0	000	0	1	2	3	0	1	0	1	0	1	1	0
2	1	001	2	3	0	1	1	0	1	0	1	0	0	1
3	2	010	4	5	6	7	0	1	0	1	1	0	0	1
4	3	011	6	7	4	5	1	0	1	0	0	1	1	0
5	4	100	0	1	2	3	1	0	1	0	0	1	1	0
6	5	101	2	3	0	1	0	1	0	1	1	0	0	1
7	6	110	4	5	6	7	1	0	1	0	1	0	0	1
8	7	111	6	7	4	5	0	1	0	1	0	1	1	0

Example I=4 :

TT(4,1)=3
TT(4,2)=6
TT(4,3)=7



K=1,3
SSB(K)=XOR(SB(K),S(8,K))

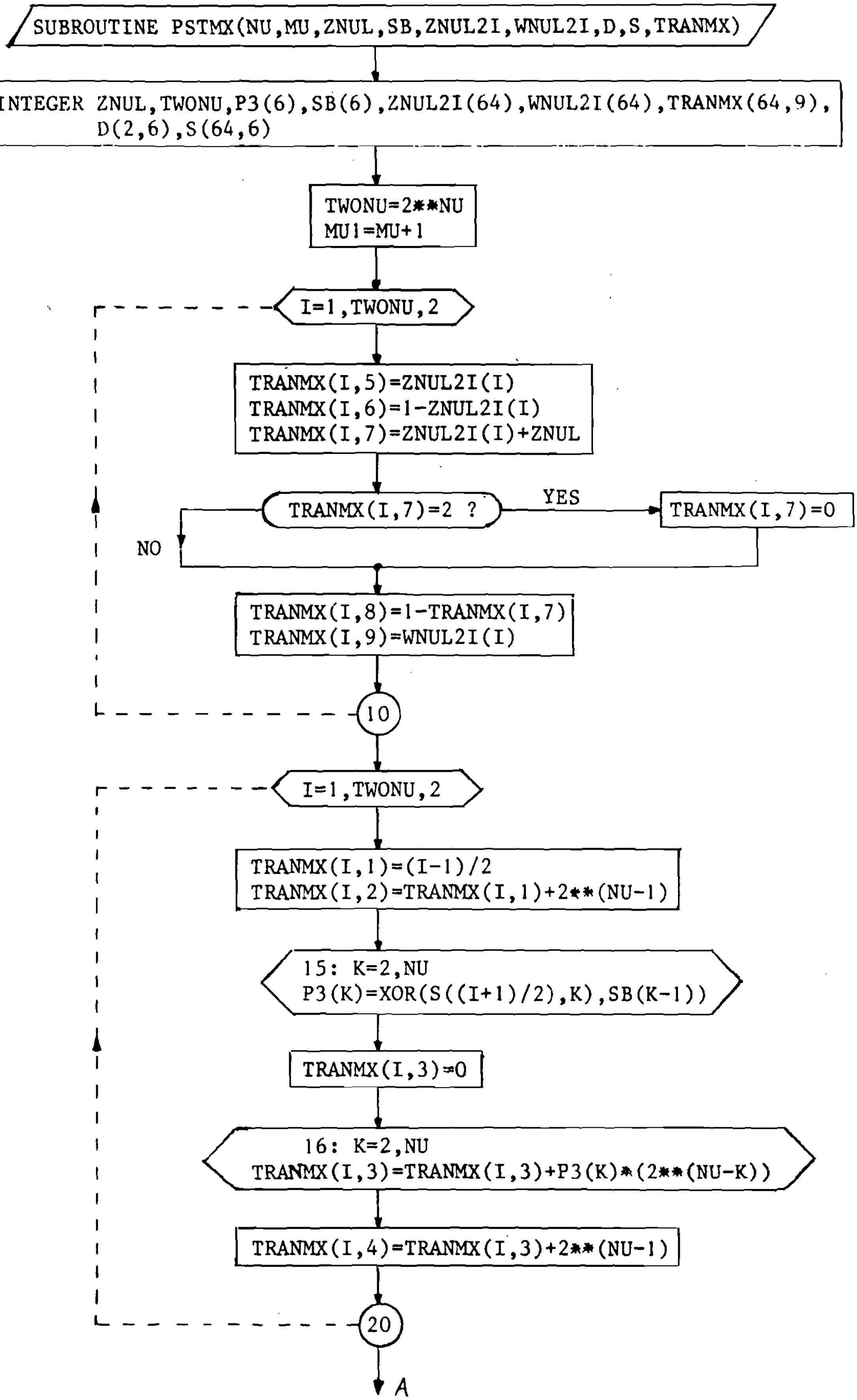
```
          0 1 1
          1 1 1
          ------ ⊕
SSB(K)=   1 0 0
```

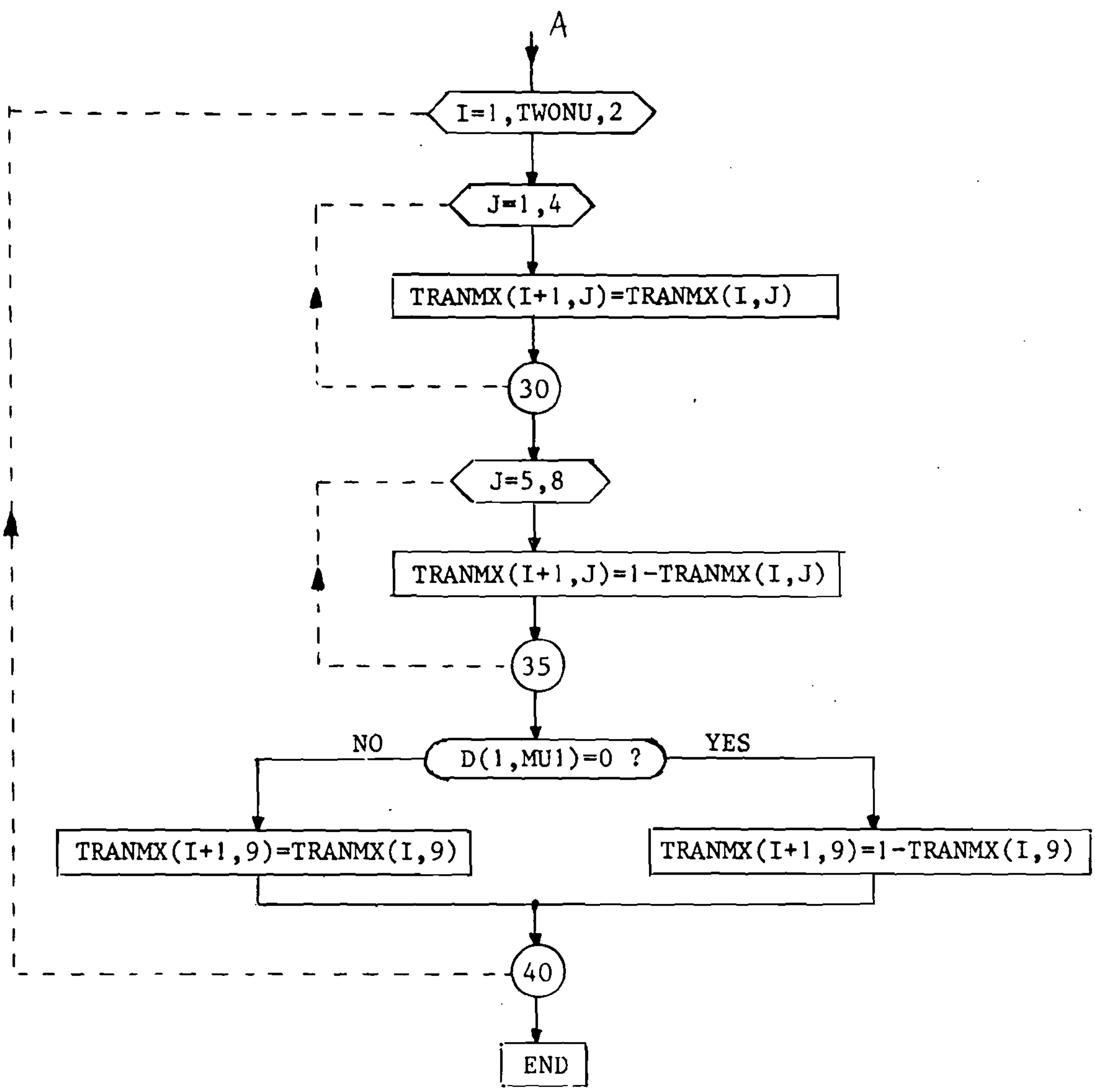
$$TT(4,4)=\sum_{K=1}^{NU} SSB(K)*2**(3-K)=1.2^2+0.2^1+0.2^0= 4$$

TT(4,5)=5
TT(4,6)=ZNUL2I(7)=1
TT(4,8)=1
TT(4,7)=0
TT(4,9)=0
TT(4,10)=WNUL2I(7)=0
TT(4,11)=0
D(1,3)=0 ---► TT(4,11)=1
TT(4,12)=1
TT(4,13)=0

LOOP 45:
TT(8,1)=TT(4,1)+TWONU1=7
LOOP 30:
TT(8,2)=TT(4,2)=6
TT(8,3)=TT(4,3)=7
TT(8,4)=TT(4,4)=4
TT(8,5)=TT(4,5)=5
LOOP 35:
TT(8,6)=1-TT(4,6)=0
TT(8,7)=1-TT(4,7)=1
TT(8,8)=1-TT(4,8)=0
TT(8,9)=1-TT(4,9)=1

LOOP 40:
TT(8,10)=TT(4,10)=0
TT(8,11)=TT(4,11)=1
TT(8,12)=TT(4,12)=1
TT(8,13)=TT(4,13)=0


SUBROUTINE PSTMX(NU,MU,ZNUL,SB,ZNUL2I,WNUL2I,D,S,TRANMX)
INTEGER ZNUL,TWONU,P3(6),SB(6),ZNUL2I(64),WNUL2I(64),TRANMX(64,9), D(2,6),S(64,6)
TWONU=2**NU
MU1=MU+1
I=1,TWONU,2
TRANMX(I,5)=ZNUL2I(I)
TRANMX(I,6)=1-ZNUL2I(I)
TRANMX(I,7)=ZNUL2I(I)+ZNUL
TRANMX(I,7)=2 ?
YES
TRANMX(I,7)=0
NO
TRANMX(I,8)=1-TRANMX(I,7)
TRANMX(I,9)=WNUL2I(I)
10
I=1,TWONU,2
TRANMX(I,1)=(I-1)/2
TRANMX(I,2)=TRANMX(I,1)+2**(NU-1)
15: K=2,NU
P3(K)=XOR(S((I+1)/2),K),SB(K-1))
TRANMX(I,3)=0
16: K=2,NU
TRANMX(I,3)=TRANMX(I,3)+P3(K)*(2**(NU-K))
TRANMX(I,4)=TRANMX(I,3)+2**(NU-1)
20
A


A
I=1,TWONU,2
J=1,4
TRANMX(I+1,J)=TRANMX(I,J)
30
J=5,8
TRANMX(I+1,J)=1-TRANMX(I,J)
35
NO
D(1,MU1)=0 ?
YES
TRANMX(I+1,9)=TRANMX(I,9)
TRANMX(I+1,9)=1-TRANMX(I,9)
40
END

PSTMX(NU,MU,ZNUL,SB,ZNUL2I,WNUL2I,D,S,TRANMX)

NU=3
MU=2
C(1,K)= 1 0 1 1
C(2,K)= 1 1 1 1
D(1,K)= 1 0 0
D(2,K)= 1 1 1
SB(K)= 0 1 1
ZNUL=1
TWONU=8
MU1=3

Parent state matrix: TRANMX(I,J)

I ⟶ J

I			1	2	3	4	5	6	7	8	9
1	0	000	0	4	1	5	0	1	1	0	0
2	1	001	0	4	1	5	1	0	0	1	1
3	2	010	1	5	0	4	1	0	0	1	1
4	3	011	1	5	0	4	0	1	1	0	0
5	4	100	2	6	3	7	0	1	1	0	1
6	5	101	2	6	3	7	1	0	0	1	0
7	6	110	3	7	2	6	1	0	0	1	0
8	7	111	3	7	2	6	0	1	1	0	1

Example I=5 :

TRANMX(5,5)=ZNUL2I(5)=0
TRANMX(5,6)=1
TRANMX(5,7)=0+1=1
TRANMX(5,8)=0
TRANMX(5,9)=0
TRANMX(5,1)=2
TRANMX(5,2)=2+4=6
K=2,3
P3(K)=XOR(S(3,K),SB(K-1)) } P3(K)= $\begin{array}{r} 1\ 0 \\ 0\ 1 \\ \hline 1\ 1 \end{array} \oplus$

$$\text{TRANMX}(5,3)=\sum_{K=2}^{NU} P3(K)*2**(3-K)=1.2^1+1.2^0= 3$$

$$S_{\frac{1}{2}i} \xrightarrow{00} S_i \quad z_{i0}$$

$$S_{\frac{1}{2}i+2^{\nu-1}} \xrightarrow{00} S_i \quad \bar{z}_{i0}$$

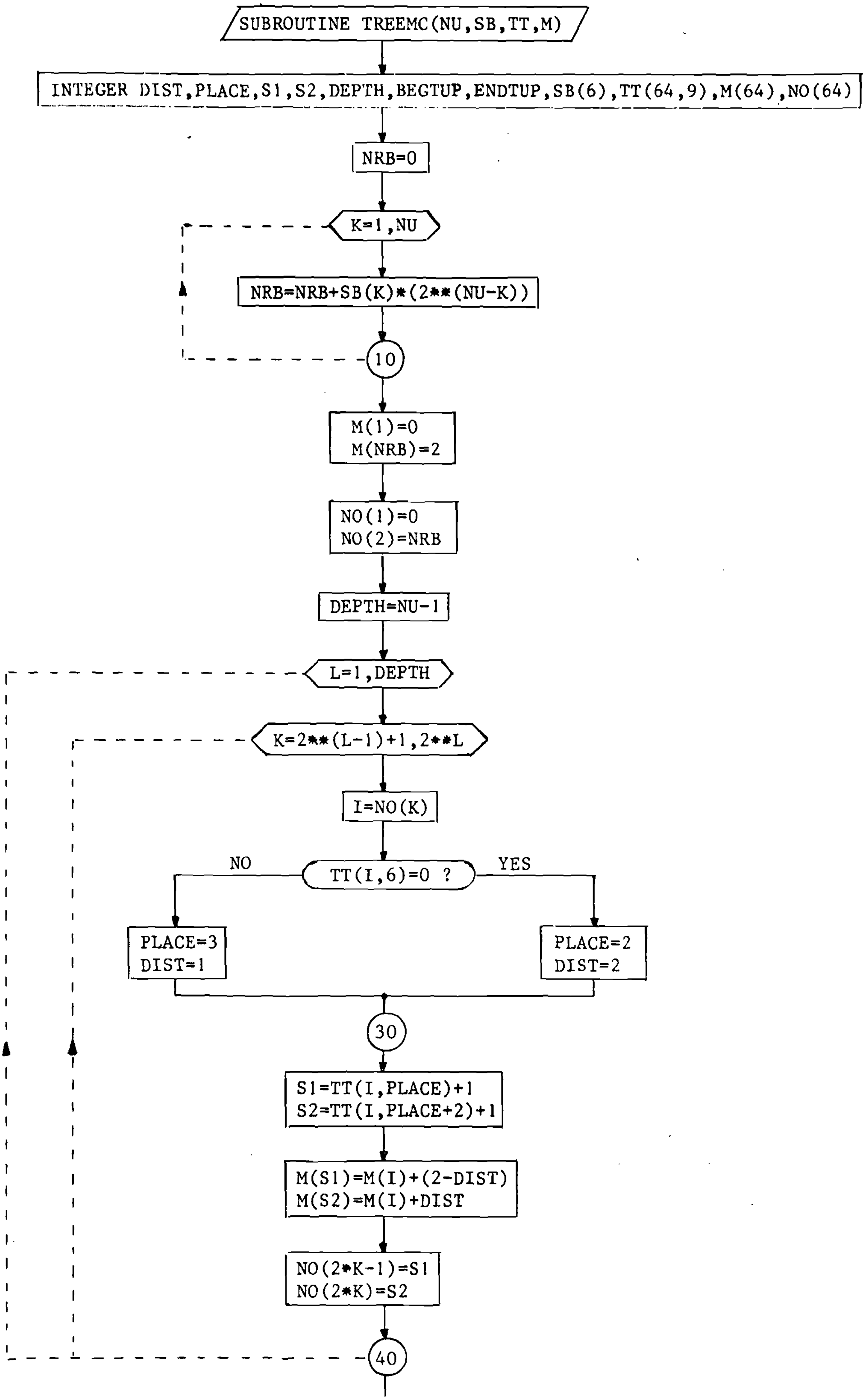
$$S_{\frac{1}{2}i} \oplus S_{\frac{1}{2}(b-1)} \xrightarrow{11} S_i \quad z_{i0} \oplus z_0$$

$$S_{\frac{1}{2}i+2^{\nu-1}} \oplus S_{\frac{1}{2}(b-1)} \xrightarrow{11} S_i \quad \bar{z}_{i0} \oplus z_0$$

$i=2k,\ k=0,1,\dots,2^{\nu-1}-1.$

TRANMX(5,4)=3+4=7
TRANMX(6,1)=2
TRANMX(6,2)=6
TRANMX(6,3)=3
TRANMX(6,4)=7
TRANMX(6,5)=1
TRANMX(6,6)=0
TRANMX(6,7)=0
TRANMX(6,8)=1

D(1,3)=0 --→ 36
36 TRANMX(6,9)=0
etç


SUBROUTINE TREEMC(NU,SB,TT,M)
INTEGER DIST,PLACE,S1,S2,DEPTH,BEGTUP,ENDTUP,SB(6),TT(64,9),M(64),NO(64)
NRB=0
K=1,NU
NRB=NRB+SB(K)*(2**(NU-K))
10
M(1)=0
M(NRB)=2
NO(1)=0
NO(2)=NRB
DEPTH=NU-1
L=1,DEPTH
K=2**(L-1)+1,2**L
I=NO(K)
TT(I,6)=0 ?
NO
YES
PLACE=3
DIST=1
PLACE=2
DIST=2
30
S1=TT(I,PLACE)+1
S2=TT(I,PLACE+2)+1
M(S1)=M(I)+(2-DIST)
M(S2)=M(I)+DIST
NO(2*K-1)=S1
NO(2*K)=S2
40

<u>TREEMC(NU,SB,TT,M)</u>

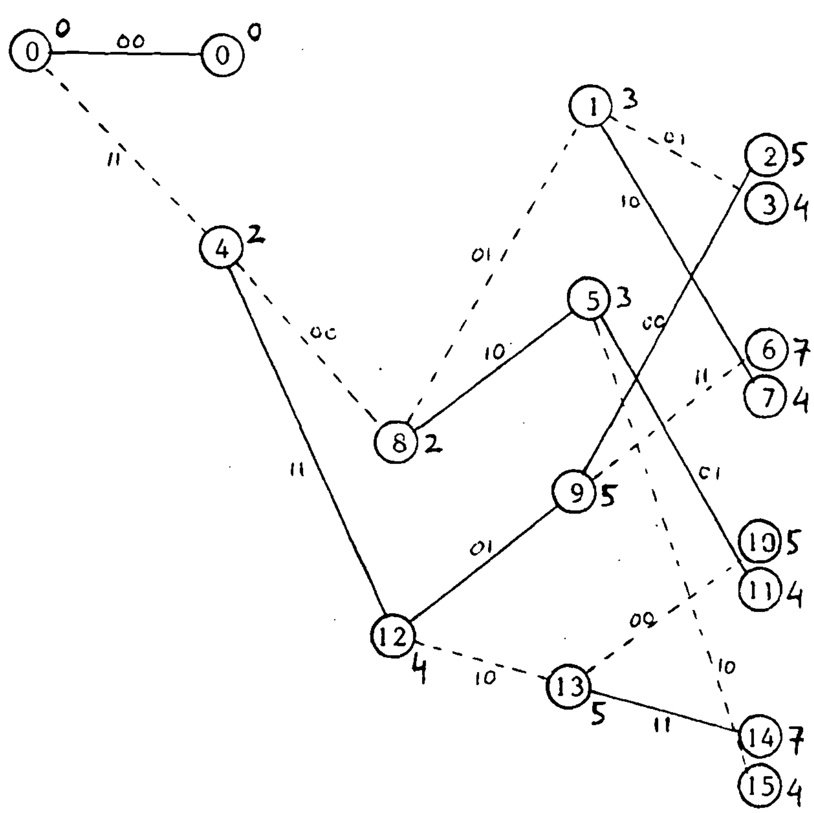
```
C(1,K)= 1 0 0 1 1
C(2,K)= 1 0 1 1 1
SB(K)= 0 1 0 1
NU=4
```

$$NRB= \sum_{K=1}^{NU} SB(K)*2**(4-K)=0.2^3+1.2^2+0.2^1+1.2^0= 5$$

```
   M(1)=0
   M(5)=2
   NO(1)=1
   NO(2)=5
   DEPTH=3
   L=1
   BEGTUP=2
   ENDTUP=2
   ----------
   K=2
   I=NO(2)=5
   TT(5,6)=0
20 PLACE=2
   DIST=2
   S1=TT(5,2)+1=9
   S2=TT(5,4)+1=13
   M(9)=M(5)=0=2
   M(13)=M(5)+2=4
   NO(3)=9
   NO(4)=13
   L=2
   BEGTUP=3
   ENDTUP=4
   ----------
   K=3
   I=NO(3)=9
   TT(9,6)=1
   PLACE=3
   DIST=1
   S1=TT(9,3)+1=2
   S2=TT(9,5)+1=6
   M(2)=M(9)+1=4
   M(6)=M(9)+1=4
   NO(5)=2
   NO(6)=6
   ----------
   K=4
   I=NO(4)=13
   TT(13,6)=1
   PLACE=3
   DIST=1
   S1=TT(13,3)+1=10
   S2=TT(13,5)+1=14
   M(10)=M(13)+1=5
   M(14)=M(13)+1=5
   NO(7)=10
   NO(8)=14
```

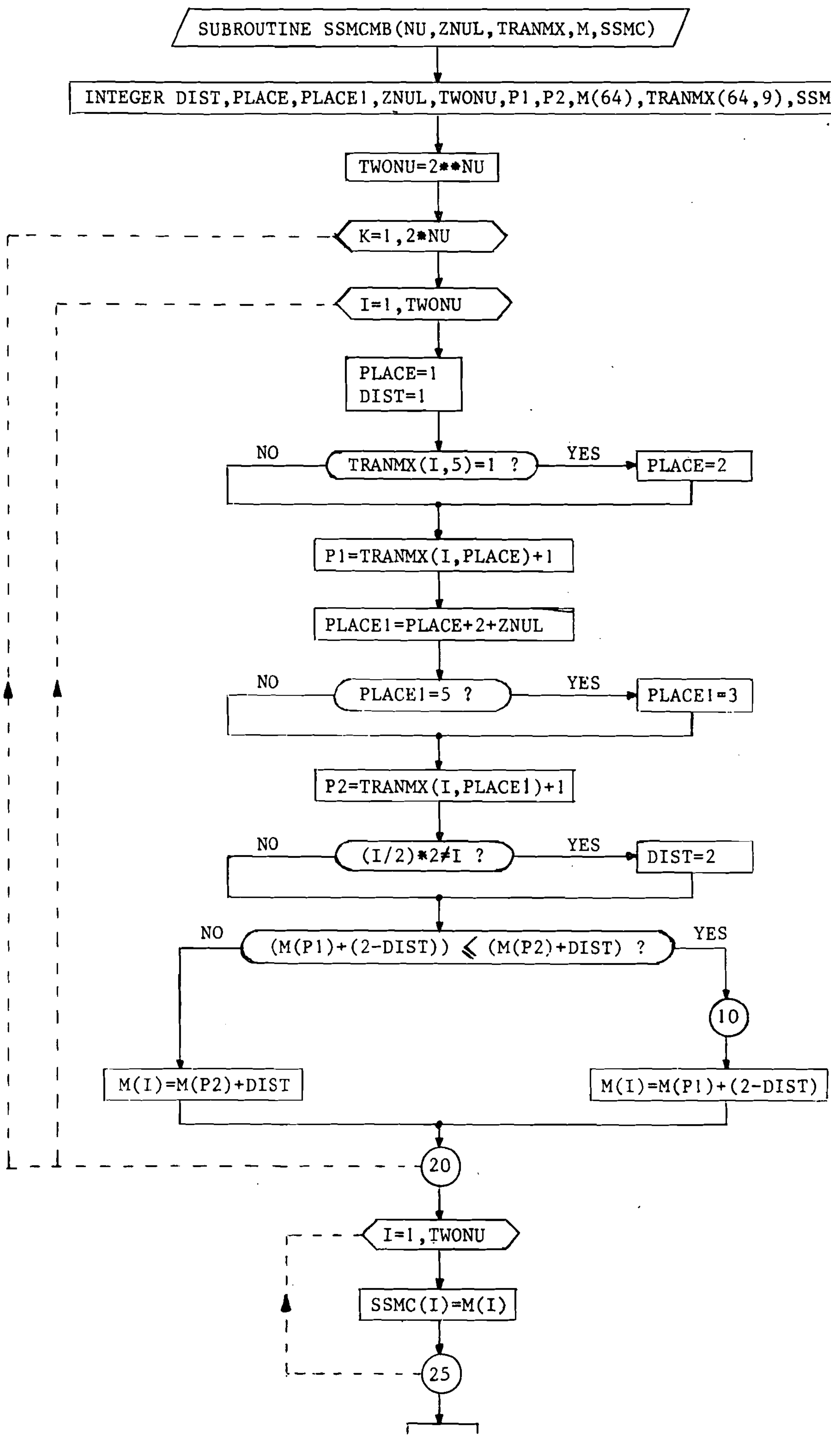
<u>State table:</u> TT(I,J)

I	J →					
1	0	0	1	4	5	0
2	1	2	3	6	7	1
3	2	4	5	0	1	0
4	3	6	7	2	3	1
5	4	8	9	12	13	0
6	5	10	11	14	15	1
7	6	12	13	8	9	0
8	7	14	15	10	11	1
9	8	0	1	4	5	1
10	9	2	3	6	7	0
11	10	4	5	0	1	1
12	11	6	7	2	3	0
13	12	8	9	12	13	1
14	13	10	11	14	15	0
15	14	12	13	8	9	1
16	15	14	15	10	11	0



M(I)

0
3
5
4
2
3
7
4
2
5
5
4
4
5
7
4


SUBROUTINE SSMCMB(NU,ZNUL,TRANMX,M,SSMC)
INTEGER DIST,PLACE,PLACE1,ZNUL,TWONU,P1,P2,M(64),TRANMX(64,9),SSMC(64)
TWONU=2**NU
K=1,2*NU
I=1,TWONU
PLACE=1
DIST=1
NO
TRANMX(I,5)=1 ?
YES
PLACE=2
P1=TRANMX(I,PLACE)+1
PLACE1=PLACE+2+ZNUL
NO
PLACE1=5 ?
YES
PLACE1=3
P2=TRANMX(I,PLACE1)+1
NO
(I/2)*2≠I ?
YES
DIST=2
NO
(M(P1)+(2-DIST)) ≤ (M(P2)+DIST) ?
YES
10
M(I)=M(P2)+DIST
M(I)=M(P1)+(2-DIST)
20
I=1,TWONU
SSMC(I)=M(I)
25

SSMCMB(NU,ZNUL,TRANMX,M,SSMC)

```
C(1,K)= 1 0 0 1 1
C(2,K)= 1 0 1 1 1
NU=4
ZNUL=0   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
M(I)=( 0 3 5 4 2 3 7 4 2 5 5 4 4 5 7 4 )
KMAX=2+NU=8
K=1
_______________________________
I=1
PLACE=1
DIST=1
TRANMX(1,5)=0
P1=TRANMX(1,1)+1=1
PLACE1=1+2+0=3
P2=TRANMX(1,3)+1=3
ODD ----► DIST=2
(M(1)+0=0)<(M(3)+2=7) --► 10
10 M(1)=M(1)+0=0
_______________________________
I=2
PLACE=1
DIST=1
TRANMX(2,5)=1 --► PLACE=2
P1=TRANMX(2,2)+1=9
PLACE1=2+2+0=4
P2=TRANMX(2,4)+1=11
EVEN ---► DIST=1
(M(9)+1=3)<(M(11)+1=6) --► 10
10 M(2)=M(9)+1=3
_______________________________
I=3
PLACE=1
DIST=1
TRANMX(3,5)=1 --► PLACE=2
P1=TRANMX(3,2)+1=10
PLACE1=2+2+0=4
P2=TRANMX(3,4)+1=12
ODD -----► DIST=2
(M(10)+0=5)<(M(12)+2=6) --► 10
10 M(3)=M(10)+0=5
```

Parent state matrix : TRANMX(I,J)

PARENT-STATE MATRIX:

I										
1	0-	0	8	2	10	0	1	0	1	0
2	1-	0	8	2	10	1	0	1	0	1
3	2-	1	9	3	11	1	0	1	0	0
4	3-	1	9	3	11	0	1	0	1	1
5	4-	2	10	0	8	0	1	0	1	1
6	5-	2	10	0	8	1	0	1	0	0
7	6-	3	11	1	9	1	0	1	0	1
8	7-	3	11	1	9	0	1	0	1	0
9	8-	4	12	6	14	0	1	0	1	1
10	9-	4	12	6	14	1	0	1	0	0
11	10-	5	13	7	15	1	0	1	0	1
12	11-	5	13	7	15	0	1	0	1	0
13	12-	6	14	4	12	0	1	0	1	0
14	13-	6	14	4	12	1	0	1	0	1
15	14-	7	15	5	13	1	0	1	0	0
16	15-	7	15	5	13	0	1	0	1	1
	J →	1	2	3	4	5	6	7	8	9

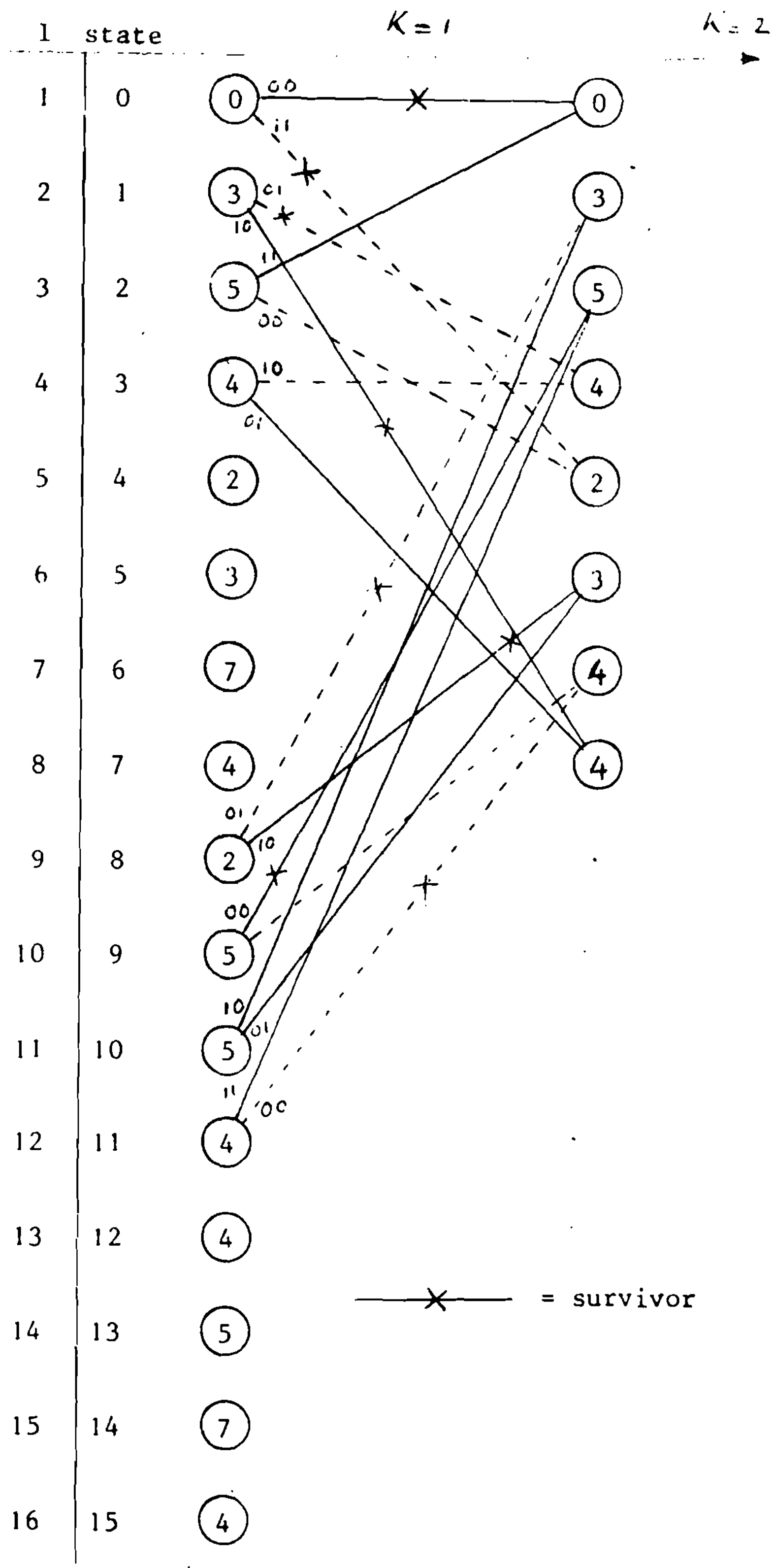
I=4
PLACE=1
DIST=1
TRANMX(4,5)=0
P1=TRANMX(4,1)+1=2
PLACE1=1+2+0=3
P2=TRANMX(4,3)+1=4
EVEN --► DIST=1
(M(2)+1=4) < (M(4)+1=5) --► 10
10 M(4)=M(2)+1=4

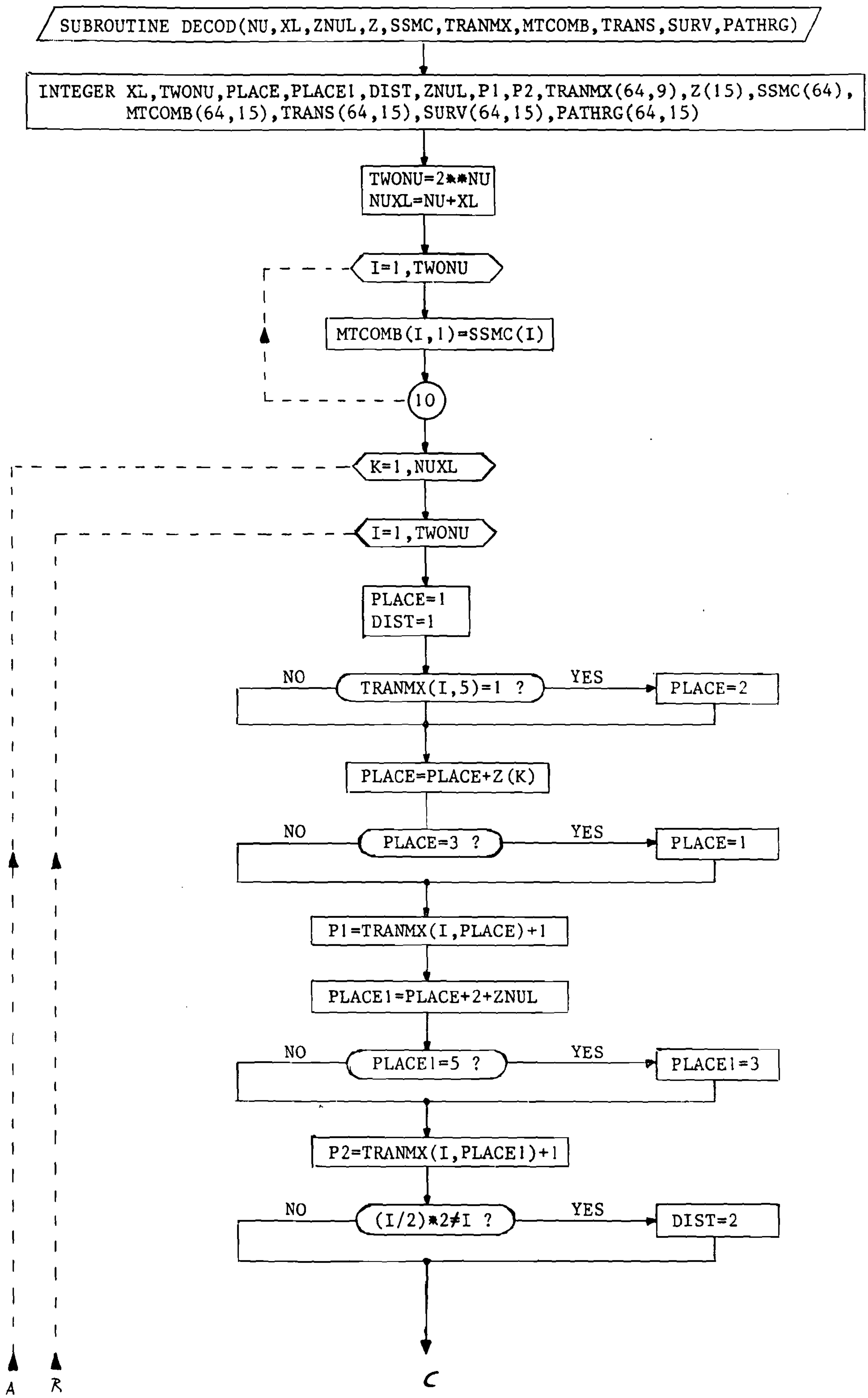
I=5
PLACE=1
DIST=1
TRANMX(5,5)=0
P1=TRANMX(5,1)+1=3
PLACE1=1+2+0=3
P2=TRANMX(5,3)+1=1
ODD ----► DIST=2
(M(3)+0=5) > (M(1)+2=2)
M(5)=M(1)+2=2

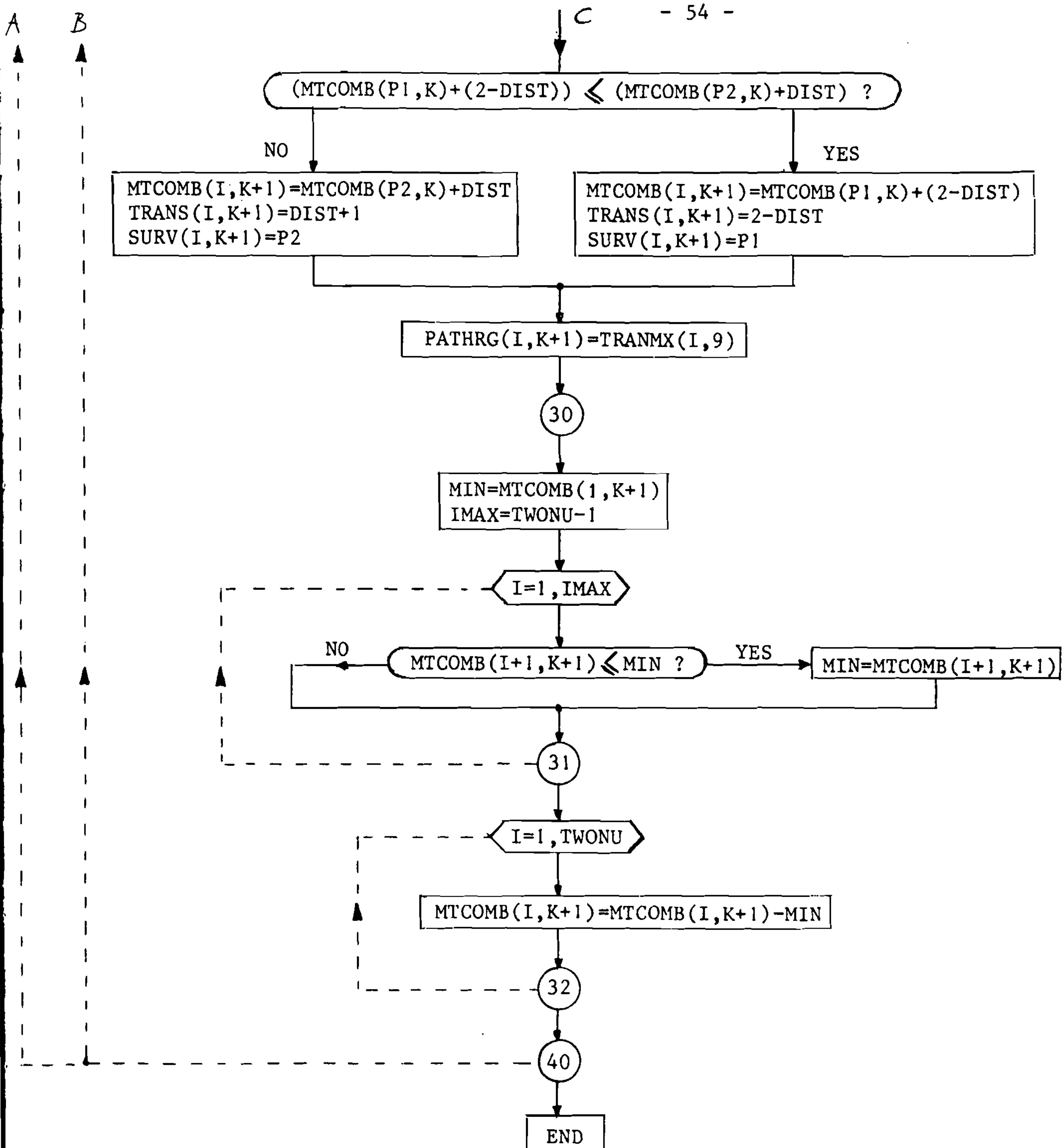
I=6
PLACE=1
DIST=1
TRANMX(6,5)=1 --► PLACE=2
P1=TRANMX(6,2)+1=11
PLACE1=2+2+0=4
P2=TRANMX(6,4)+1=9
EVEN --► DIST=1
(M(11)+1=6) > (M(9)+1=3)
M(6)=M(9)+1=3

I=7
PLAÇE=1
DIST=1
TRANMX(7,5)=1 --► PLACE=2
P1=TRANMX(7,2)+1=12
PLACE1=2+2+0=4
P2=TRANMX(7,4)+1=10
ODD ---► DIST=2
(M(12)+0=4)   (M(10)+2=7) --► 10
10 M(7)=M(12)+0=4

I=8




SUBROUTINE DECOD(NU,XL,ZNUL,Z,SSMC,TRANMX,MTCOMB,TRANS,SURV,PATHRG)
INTEGER XL,TWONU,PLACE,PLACE1,DIST,ZNUL,P1,P2,TRANMX(64,9),Z(15),SSMC(64),
MTCOMB(64,15),TRANS(64,15),SURV(64,15),PATHRG(64,15)
TWONU=2**NU
NUXL=NU+XL
I=1,TWONU
MTCOMB(I,1)=SSMC(I)
10
K=1,NUXL
I=1,TWONU
PLACE=1
DIST=1
NO
TRANMX(I,5)=1 ?
YES
PLACE=2
PLACE=PLACE+Z(K)
NO
PLACE=3 ?
YES
PLACE=1
P1=TRANMX(I,PLACE)+1
PLACE1=PLACE+2+ZNUL
NO
PLACE1=5 ?
YES
PLACE1=3
P2=TRANMX(I,PLACE1)+1
NO
(I/2)*2≠I ?
YES
DIST=2
A
R
C


A
B
C
(MTCOMB(P1,K)+(2-DIST)) ≤ (MTCOMB(P2,K)+DIST) ?
NO
YES
MTCOMB(I,K+1)=MTCOMB(P2,K)+DIST
TRANS(I,K+1)=DIST+1
SURV(I,K+1)=P2
MTCOMB(I,K+1)=MTCOMB(P1,K)+(2-DIST)
TRANS(I,K+1)=2-DIST
SURV(I,K+1)=P1
PATHRG(I,K+1)=TRANMX(I,9)
30
MIN=MTCOMB(1,K+1)
IMAX=TWONU-1
I=1,IMAX
NO
MTCOMB(I+1,K+1) ≤ MIN ?
YES
MIN=MTCOMB(I+1,K+1)
31
I=1,TWONU
MTCOMB(I,K+1)=MTCOMB(I,K+1)-MIN
32
40
END

DECOD(NU,XL,ZNUL,Z,SSMC,TRANMX,MTCOMB,TRANS,SURV,PATHRG)

```
NU=2
XL=2
X(K)=1 1
NOISE(1,K)= 1 0
NOISE(2,K)= 0 1                 Z(K)= 1 1 0 1
C(1,K)= 1 1 1
C(2,K)= 1 0 1
NUXL=4
I=1,TWONU           }
MTCOMB(I,1)=SSMC(I) }  MTCOMB(I,1)=( 0 3 2 3 )
K=1
 I=1
 PLACE=1
 DIST=1
 TRANMX(1,5)=0
 PLACE=1+1=2
 P1=TRANMX(1,2)+1=3
 PLACE1=2+2+1=5
 PLACE1=3
 P2=TRANMX(1,3)+1=2
 ODD --> DIST=2
 (MTCOMB(3,1)+0=2)<(MTCOMB(2,1)+2=5) --> 20
20 MTCOMB(1,2)=2
 TRANS(1,2)=0
 SURV(1,2)=3
 PATHRG(1,2)=0
 I=2
 PLACE=1
 DIST=1
 TRANMX(2,5)=1 --> PLACE=2
 PLACE=2+1=3
 PLACE=1
 P1=TRANMX(2,1)+1=1
 PLACE1=1+2+1=4
 P2=TRANMX(2,4)+1=4
 EVEN --> DIST=1
 (MTCOMB(1,1)+1=1)<(MTCOMB(4,1)+1=4) --> 20
20 MTCOMB(2,2)=1
 TRANS(2,2)=1
 SURV(2,2)=1
 PATHRG(2,2)=1
 I=3
 PLACE=1
 DIST=1
 TRANMX(3,5)=1 --> PLACE=2
 PLACE=2+1=3
 PLACE=1
 P1=TRANMX(3,1)+1=2
 PLACE1=1+2+1=4
 P2=TRANMX(3,4)+1=3
 ODD --> DIST=2
 (MTCOMB(2,1)+0=3)<(MTCOMB(3,1)+2=4) --> 20
20 MTCOMB(3,2)=3
 TRANS(3,2)=0
 SURV(3,2)=2
 PATHRG(3,2)=1
```

Parent state matrix: TRANMX(I,J)

PARENT-STATE MATRIX:

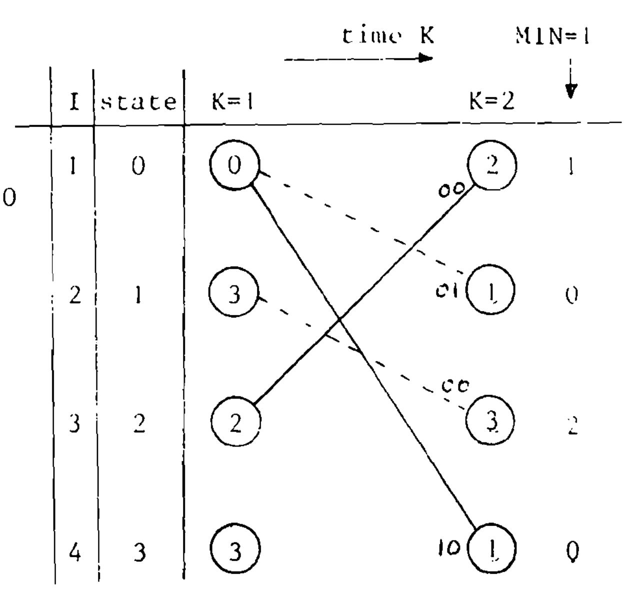
I		1	2	3	4	5	6	7	8	9
1	0-	0	2	1	3	0	1	1	0	0
2	1-	0	2	1	3	1	0	0	1	1
3	2-	1	3	0	2	1	0	0	1	1
4	3-	1	3	0	2	0	1	1	0	0

(I ↓, J →)

DECØDING SCHEME: (MTCOMB, SURV, PATHRG, TRANS)

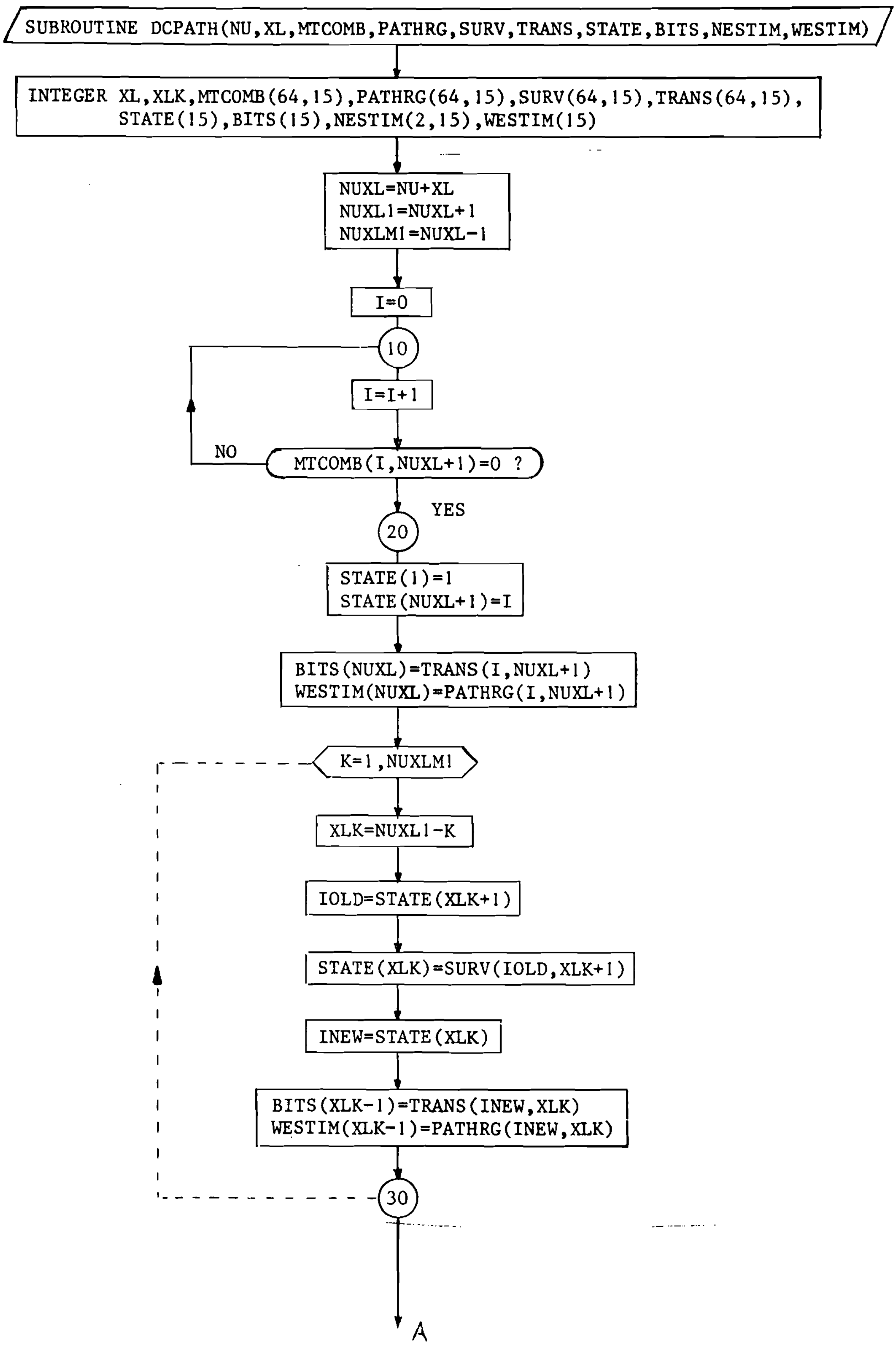
I	1	2	3	4	5
1	0	1 200	2 200	1 000	0 200
2	3	0 011	1 312	0 211	1 312
3	2	2 110	0 110	0 310	0 110
4	3	0 002	1 301	0 202	1 301

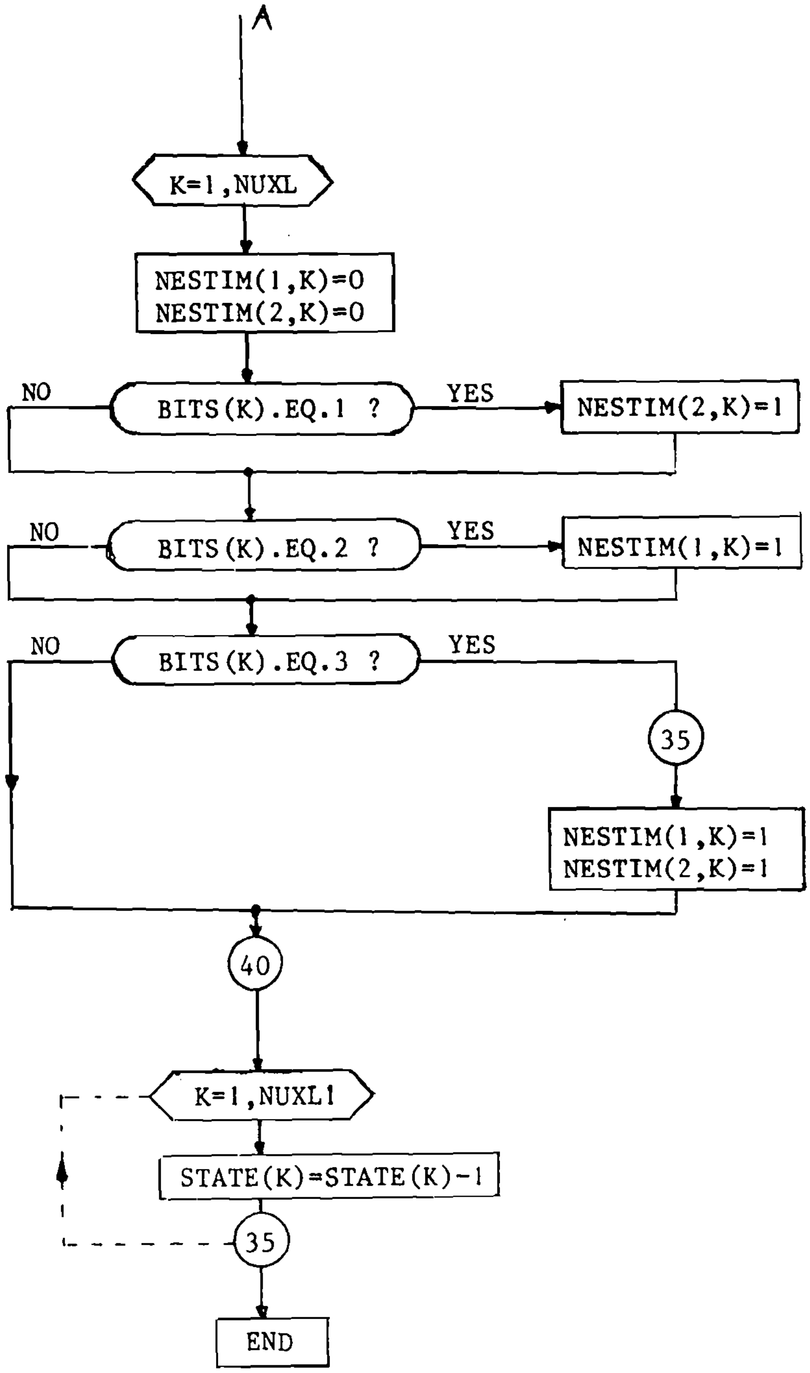
(I ↓, K →)



Z(1)=1

```
   I=4
   PLACE=1
   DIST=1
   TRANMX(4,5)=0
   PLACE=2
   P1=TRANMX(4,2)+1=4
   PLACE1=2+2+1=5
   PLACE1=3
   P2=TRANMX(4,3)+1=1
   EVEN --> DIST=1
   (MTCOMB(4,1)+1=4) > (MTCOMB(1,1)+1=1)
   MTCOMB(4,2)=1
   TRANS(4,2)=2
   SURV(4,2)=1 --> 25
25 PATHRG(4,2)=0
-----------------------------------------------
   K=1    MTCOMB(I,2)=( 2 1 3 1 )
-----------------------------------------------
   LOOP 31:
   MIN=MTCOMB(1,1)=2
   I=1,3                                     }
   IF (MTCOMB(I+1,2) ≤ MIN)  MIN=MTCOMB(I+1,2) }      MIN=MTCOMB(4,2)=1
   I=1,4                          }
   MTCOMB(I,K+1)=MTCOMB(I,K+1)-MIN }                 MTCOMB(I,2)=( 1 0 2 0 )
-----------------------------------------------
   K=2
   I=1
   PLACE=1
   DIST=1
   TRANMX(1,5)=0
   etc.              ------------------------->  MTCOMB(I,3)=( 2 1 0 1 )
                                                 MTCOMB(I,4)=( 1 0 0 0 )
                                                 MTCOMB(I,5)=( 0 1 0 1 )
```

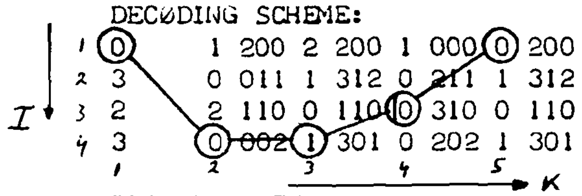

SUBROUTINE DCPATH(NU,XL,MTCOMB,PATHRG,SURV,TRANS,STATE,BITS,NESTIM,WESTIM)
INTEGER XL,XLK,MTCOMB(64,15),PATHRG(64,15),SURV(64,15),TRANS(64,15),
STATE(15),BITS(15),NESTIM(2,15),WESTIM(15)
NUXL=NU+XL
NUXL1=NUXL+1
NUXLM1=NUXL-1
I=0
10
I=I+1
NO
MTCOMB(I,NUXL+1)=0 ?
YES
20
STATE(1)=1
STATE(NUXL+1)=I
BITS(NUXL)=TRANS(I,NUXL+1)
WESTIM(NUXL)=PATHRG(I,NUXL+1)
K=1,NUXLM1
XLK=NUXL1-K
IOLD=STATE(XLK+1)
STATE(XLK)=SURV(IOLD,XLK+1)
INEW=STATE(XLK)
BITS(XLK-1)=TRANS(INEW,XLK)
WESTIM(XLK-1)=PATHRG(INEW,XLK)
30
A


A
K=1,NUXL
NESTIM(1,K)=0
NESTIM(2,K)=0
NO
BITS(K).EQ.1 ?
YES
NESTIM(2,K)=1
NO
BITS(K).EQ.2 ?
YES
NESTIM(1,K)=1
NO
BITS(K).EQ.3 ?
YES
35
NESTIM(1,K)=1
NESTIM(2,K)=1
40
K=1,NUXL1
STATE(K)=STATE(K)-1
35
END

DCPATH(NU,XL,MTCOMB,PATHRG,SURV,TRANS,STATE,BITS,NESTIM,WESTIM)

```
   NU=2
   XL=2
   X(K)=11
   C(1,K)= 1 1 1
   C(2,K)= 1 0 1
   NOISE(1,K)= 1 0
   NOISE(2,K)= 0 1
   NUXL=4
   NUXL1=5
   NUXLM1=3
   MTCOMB(1,5)=0 --→ 20
20 STATE(1)=1
   STATE(5)=1
   BITS(4)=TRANS(1,5)=0   (00)
   WESTIM(4)=PATHRG(1,5)=0
   ----------------------------
   LOOP 30:
   K=1
   IOLD=STATE(5-1+1)=STATE(5)=1
   STATE(4)=SURV(1,5)=3
   INEW=STATE(4)=3
   BITS(3)=TRANS(3,4)=0   (00)
   WESTIM(3)=PATHRG(3,4)=1
   ----------------------------
   K=2
   IOLD=STATE(5-2+1)=STATE(4)=3
   STATE(3)=SURV(3,4)=4
   INEW=STATE(3)=4
   BITS(2)=TRANS(4,3)=1   (01)
   WESTIM(2)=PATHRG(4,3)=0
   ----------------------------
   K=3
   IOLD=STATE(5-3+1)=STATE(3)=4
   STATE(2)=SURV(4,3)=4
   INEW=STATE(2)=4
   BITS(1)=TRANS(4,2)=2   (10)
   WESTIM(1)=PATHRG(4,2)=0
   ----------------------------
   LOOP 40:
   K=1
   NESTIM(1,1)=0
   NESTIM(2,1)=0
   BITS(1)=2 --- NESTIM(1,1)=1
   ----------------------------
   K=2
   NESTIM(1,2)=0
   NESTIM(2,2)=0
   BITS(2)=1 --- NESTIM(2,2)=1
   ----------------------------
   K=3
   NESTIM(1,3)=0
   NESTIM(2,3)=0
   BITS(3)=0
   ----------------------------
   K=4
   NESTIM(1,4)=0
   NESTIM(2,4)=0
   BITS(4)=0
   ----------------------------
   K=1,5                }
   STATE(K)=STATE(K)-1  }
```

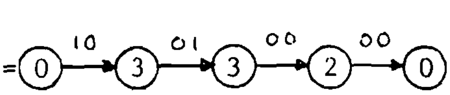
DECØDING SCHEME:



DECØDED PATH:

```
Z(K)        =  1    1    0    1

WESTIM(K)   =  0    0    1    0

STATE(K)    = 0--- 3--- 3--- 2--- 0·

NESTIM(1,K) =  1    0    0    0
NESTIM(2,K) =  0    1    0    0


SØMYD(K)    = 1110
WESTIM(K)   = 0010
---------------------------+
XESTIM(K)   = 1100

X(K)        = 11
```

```
NESTIM(1,K)= 1 0 0 0
NESTIM(2,K)= 0 1 0 0

WESTIM(K)  = 0 0 1 0
```

STATE(K)=


```
ØK, ED DISPLY
GØ
EDIT
P1000
.NULL.
C      THIS PRØGRAM IS EXECUTABLE ØN THE TELETYPE AS WELL AS THE TER-
C      MINAL CØNTRØL SYSTEM T-4014-1.
C      ØNLY THE TCS-VERSIØN IS PRØVIDED WITH INTRØDUCTØRY TEXT.
C      THE USER CAN CØNTRØL THE PACE ØF THE VARIØUS STEPS IN THE
C      DECØDING PRØCEDURE BY MEANS ØF A SIMPLE PUSH ØN THE CAR-
C      RIAGE RETURN KEY.
C      ****************************************************************
       INTEGER XL,XL1,XLK,ZNUL,TWØNU,TWØNU1,X1(2,15)
       INTEGER C(2,7),D(2,6),X(15),NØISE(2,15),Z(21),Y(2,15),N2C1(21),
      1 N1C2(21),Y1D1(21),Y2D2(21),XC1(21),XC2(21),SB(6),S(16,6),
      1 ZNUL2I(16),WNUL2I(16),TT(16,13),TRANMX(16,9),M(16),TRMC(16),
      1 SSMC(16),MTCØMB(16,15),SURV(16,15),PATHRG(16,15),TRANS(16,15),
      1 BITS(15),STATE(15),WESTIM(15),SØMYD(15),XESTIM(15),
      1 DECØUT(16,50),NESTIM(2,15),IADE(5)
       REAL XD(16),YD(16)
       WRITE(1,1000)
1000   FØRMAT('WHEN USING THE TELETYPE ------> MØDE=1'/
      1 'WHEN USING THE DISPLAY  ------> MØDE=2')
       CALL TNØUA('MØDE= ',6)
       READ(1,5) MØDE
       IF (MØDE.EQ.1) GØTØ 1001
       CALL INITT(120)
       CALL ANMØDE
C PRINT THE INTRØDUCTØRY TEXT.
       CALL TEXT1
       READ(1,2)
2      FØRMAT(A1)
       CALL ERASE
       CALL TSEND
C PRINT THE TABLE ØF FACTØRIZED CØNNECTIØNPØLYNØMIALS.
       CALL FACPØL
       READ(1,2)
       CALL ERASE
       CALL TSEND
1001   WRITE(1,3)
3      FØRMAT('   NØW GIVE AS INPUT:'/)
       IF (MØDE.EQ.1) WRITE(1,1)
1      FØRMAT('- THE NUMBER NØ ØF THE CØDE')
       WRITE(1,4)
4      FØRMAT('- THE CØDES CØNSTRAINTLENGTH NU'/
      1 '- THE CØNNECTIØNPØLYNØMIALS C(N,K), K=1,2 AND K=1,NU+1'/
      1 '- THE DATASEQUENCELENGTH XL, WHERE XL >1'/
      1 '- THE DATASEQUENCE X(K), K=1,XL'/
      1 '- THE NØISESEQUENCE-PAIR NØISE(N,K), N=1,2 AND K=1,XL'//
      1 '   REMEMBER THAT NU+XL < 16 !!'///)
C      ****************************************************************
```

```
C     THIS IS THE MAIN PRØGRAM WHERE ALL NECESSARY CALCULATIØNS TAKE
C     PLACE. ALL SUBRØUTINES ARE CALLED AND THE CØMPLETE DECØDING-
C     PRØCEDURE IS CARRIED ØUT.
C     AS INPUT THE PRØGRAM ASKS LINE BY LINE:
C     NØ (THE NUMBER ØF THE CØDE WHEN USING THE TELETYPE)
C     NU
C     C(1,K)
C     C(2,K)
C     XL
C     X(K)
C     NØISE(1,K)
C     NØISE(2,K)
C     ************************************************************
C
C
C THE READING ØF THE NUMBER ØF THE CØDE IS ØMITTED WHEN THE
C DISPLAY IS USED.
      IF (MØDE.EQ.2) GØTØ 1002
      CALL TNØUA('NØ= ',4)
C READ THE NUMBER ØF THE CØDE
      READ(1,1111) NØ
1111  FØRMAT(3A2)
1002  CALL TNØUA('NU= ',4)
C READ THE CØDE'S CØNSTRAINTLENGTH NU.
      READ(1,5) NU
      NU1=NU+1
      NU2=NU+2
      NUMIN1=NU-1
      TWØNU=2**NU
      TWØNU1=2**(NU-1)
      CALL TNØUA('C(1,K)= ',8)
C READ THE PØLYNØMIALS C(N,K),N=1,2 AND K=1,NU1.
      READ(1,6) (C(1,K),K=1,NU1)
      CALL TNØUA('C(2,K)= ',8)
      READ(1,6) (C(2,K),K=1,NU1)
      CALL TNØUA('XL= ',4)
C READ THE LENGTH XL ØF THE DATASEQUENCE X(K).(NU+XL NØT GREATER
C THEN 11).
      READ(1,5) XL
      XL1=XL+1
      NUXL=NU+XL
      NUXL1=NUXL+1
      CALL TNØUA('X(K)      = ',12)
C READ THE DATASEQUENCE X(K),K=1,XL.
      READ(1,6) (X(K),K=1,XL)
      CALL TNØUA('NØISE(1,K)= ',12)
C READ THE NØISE-SEQUENCES NØISE(N,K),N=1,2 AND K=1,XL.
      READ(1,6) (NØISE(1,K),K=1,XL)
      CALL TNØUA('NØISE(2,K)= ',12)
      READ(1,6) (NØISE(2,K),K=1,XL)
5     FØRMAT(I2)
6     FØRMAT(15I1)
```

```
C CALCULATE THE PRØDUCTS NØISE(1,K)*C(2,K) AND NØISE(2,K)*C(1,K)
C ØF LENGTH NUXL.
      CALL PRØDCT(XL,NU1,NØISE,1,C,2,N1C2)
      CALL PRØDCT(XL,NU1,NØISE,2,C,1,N2C1)
C LØØP 9: THE MØD-2 SUM ØF THE PRØDUCTS ABØVE FØRM THE SYNDRØMEØUT-
C PUTSEQUENCE Z(K),K=1,NUXL.
      DØ 9 K=1,NUXL
9     Z(K)=XØR(N1C2(K),N2C1(K))
C LØØP 10: MAKE A TWØ-DIMENSIØNAL ARRAY ØUT ØF THE ØNE-DIMENSIØNAL
C ARRAY X(K), IN ØRDER TØ SATISFY THE FØRMAT ØF THE SUBRØUTINE
C PRØDCT.
      DØ 10 K=1,XL
10    X1(1,K)=X(K)
C CALCULATE THE ØUTPUTSEQUENCES X(K)*C(1,K) AND X(K)*C(2,K) ØF THE
C ENCØDER.
      CALL PRØDCT(XL,NU1,X1,1,C,1,XC1)
      CALL PRØDCT(XL,NU1,X1,1,C,2,XC2)
C LØØP 11: MAKE THE LAST NU BITS ØF THE NØISE-SEQUENCES EQUAL ZERØ.
      DØ 11 I=1,2
      DØ 11 K=XL1,NUXL
11    NØISE(I,K)=0
C LØØP 15: CALCULATE THE RECEIVED LINE-SEQUENCES Y(1,K)=X(K)*C(1,K)
C + NØISE(1,K) AND Y(2,K)=X(K)*C(2,K) + NØISE(2,K) ØF LENGTH NUXL.
      DØ 15 I=1,2
      DØ 15 K=1,NUXL
      Y(1,K)=XØR(XC1(K),NØISE(1,K))
15    Y(2,K)=XØR(XC2(K),NØISE(2,K))
C EVALUATE THE PØLYNØMIALS D(N,K),N=1,2 ØF DEGREE MU.
      CALL DPØLYN(1,NU,C,MU,D)
      CALL DPØLYN(2,NU,C,MU,D)
      MU1=MU+1
      NUMUXL=NU+MU+XL
C CALCULATE THE PRØDUCTS Y(1,K)*D(1,K) AND Y(2,K)*D(2,K) ØF
C LENGTH NUMUXL.
      CALL PRØDCT(NUXL,MU1,Y,1,D,1,Y1D1)
      CALL PRØDCT(NUXL,MU1,Y,2,D,2,Y2D2)
C LØØP 16: CALCULATE THEIR SUM MØD-2.
      DØ 16 K=1,NUMUXL
16    SØMYD(K)=XØR(Y1D1(K),Y2D2(K))
C CALCULATE THE BASE-STATE SB(K),K=1,NU.
      CALL BSTATE(C,NU,SB)
C CALCULATE THE SPECIFIC ØUTPUTVALUE ZNUL ØF THE TRANSITIØN ØF STATE
C S(B/2) TØWARDS THE ZERØ STATE S(0) WITH INPUTBITS 11.
      ZNUL=0
      DØ 20 K=2,NU
20    ZNUL=ZNUL+C(1,K)*SB(K-1)
C IF ZNUL IS EVEN THEN ZNUL=0 ELSE ZNUL=1.
      IF (ZNUL.EQ.((ZNUL/2)*2)) GØTØ 25
      ZNUL=1
      GØTØ 30
25    ZNUL=0
30    CØNTINUE
```

```
C EVALUATE THE STATE-MATRIX S(I,J),I=1,TWØNU AND J=1,NU. (IN FORTRAN
C A ZERØ INDEX IN AN ARRAY IS NØT ALLØWED, HENCE S(I,J),I=1,J=1,NU
C HØLDS THE ZERØ STATE 00000...0 ØF LENGTH NU).
      CALL STATMX(NU,S)
C EVALUATE THE ØUTPUTVALUES ZNUL2I(I) AND WNUL2I(I), I=1,TWØNU,2.
      CALL ZWNUL(NU,MU,C,D,S,ZNUL2I,WNUL2I)
C CØNSTRUCT THE TRANSITIØN TABLE.
      CALL TRATBL(NU,MU,D,SB,S,ZNUL2I,WNUL2I,TT)
C EVALUATE THE FØUR PARENT STATES ØF EACH STATE S(I) WITH THEIR
C PARTICULAR TRANSITIØNS AND ØUTPUTVALUES Z(K) AND W(K).
      CALL PSTMX(NU,MU,ZNUL,SB,ZNUL2I,WNUL2I,D,S,TRANMX)
C CALCULATE THE METRIC-VALUES ØF THE STATES S(I) AT DEPTH NUMINI
C STARTING WITH M(1)=0.
      CALL TREEMC(NU,SB,TT,M)
C GIVE THIS METRICCØMBINATIØN THE NAME TRMC.
      DØ 45 I=1,TWØNU
45    TRMC(I)=M(I)
C CALCULATE THE STEADY STATE METRIC CØMBINATIØN SSMC(I),I=1,TWØNU.
      CALL SSMCMB(NU,ZNUL,TRANMX,M,SSMC)
C CALCULATE EACH TIME K: THE NEW METRICCØMBINATIØN MTCØMB(I,K),
C THE SURVIVØRS SURV(I,K), THEIR TRANSITIØNS TRANS(I,K) AND THE
C CØRRESPØNDING ØUTPUTVALUES FØR W(K) IN PATHRG(I,K).
      CALL DECØD(NU,XL,ZNUL,Z,SSMC,TRANMX,MTCØMB,TRANS,SURV,PATHRG)
C EVALUATE THE DECØDED PATH IN TERMS ØF THE NEW STATE(K), THE
C TRANSITIØN TØ THAT STATE, NESTIM(1,K)/NESTIM(2,K) AND THE
C CØRRESPØNDING ØUTPUTVALUE WESTIM(K).
      CALL DCPATH(NU,XL,MTCØMB,PATHRG,SURV,TRANS,STATE,BITS,
     1 NESTIM,WESTIM)
C LØØP 90: EVALUATE THE ESTIMATED DATASEQUENCE.
      DØ 90 K=1,NUXL
90    XESTIM(K)=XØR(SØMYD(K),WESTIM(K))
C LØØP 95: AS THE SURVIVØRS ARE THE DECIMAL VALUE ØF THE BINARY
C REPRESENTATIØN ØF THE STATE, WE MUST SUBTRACT THE VALUE ØNE.
      DØ 95 K=1,NUXL1
      DØ 95 I=1,TWØNU
      SURV(I,K)=SURV(I,K)-1
95    CØNTINUE
C     ***********************************************************
C     ØUTPUT PRØGRAM
C     THE CØMPLETE DECØDINGPRØCEDURE IS PRINTED ØUT.
C     THE CØMPLETE TRELLIS UP TØ TIME K=NUXL IS GIVEN IN THE FØRM
C     ØF CØLUMNS AT EACH TIME K, THAT CØNTAIN THE METRICCØMBINATIØN
C     MTCØMB(I,K), THE SURVIVØRS SURV(I,K), THE ØUTPUTS W IN
C     PATHRG(I,K) AND THE TRANSITIØNS TRANS(I,K).
C     THE SPECIFIC DECØDED PATH IS INDICATED WITH AT EACH TIME K,
C     THE ØUTPUTS Z(K) AND WESTIM(K), THE NEW STATE AT TIME K+1 AND
C     THE TRANSITIØNS TØ THAT NEW STATE.
C     AT LAST THE ESTIMATED DATASEQUENCE XESTIM(K) IS CØMPARED WITH
C     THE ØRIGINAL DATASEQUENCE X(K).
C     ***********************************************************
C
C
      WRITE(1,102) (D(1,K),K=1,MU1)
102   FØRMAT('D(1,K)= '6I1)
      WRITE(1,103) (D(2,K),K=1,MU1)
103   FØRMAT('D(2,K)= '6I1)
      WRITE(1,104) (SB(K),K=1,NU)
104   FØRMAT('SB(K) = '6I1)
      WRITE(1,105) ZNUL
105   FØRMAT('ZNUL  = 'I1/)
```

```
      WRITE(1,106) (XC1(K),K=1,NUXL)
106   FØRMAT(/'XC1(K)',T11,'= '20I1)
      WRITE(1,107) (NØISE(1,K),K=1,NUXL)
107   FØRMAT('NØISE(1,K)= '20I1)
      WRITE(1,108) (Y(1,K),K=1,NUXL)
108   FØRMAT('----------------------------+'/
     1'Y(1,K)',T11,'= '20I1/)
      WRITE(1,109) (XC2(K),K=1,NUXL)
109   FØRMAT(/'XC2(K)',T11,'= '20I1)
      WRITE(1,110) (NØISE(2,K),K=1,NUXL)
110   FØRMAT('NØISE(2,K)= '20I1)
      WRITE(1,111) (Y(2,K),K=1,NUXL)
111   FØRMAT('----------------------------+'/
     1'Y(2,K)',T11,'= '20I1/)
      WRITE(1,112) (N1C2(K),K=1,NUXL)
112   FØRMAT(/'N1C2(K)',T11,'= '20I1)
      WRITE(1,113) (N2C1(K),K=1,NUXL)
113   FØRMAT('N2C1(K)',T11,'= '20I1)
      WRITE(1,114) (Z(K),K=1,NUXL)
114   FØRMAT('----------------------------+'/
     1'Z(K)',T11,'= '20I1/)
      WRITE(1,115) (Y1D1(K),K=1,NUMUXL)
115   FØRMAT(/'Y1D1(K)',T11,'= '25I1)
      WRITE(1,116) (Y2D2(K),K=1,NUMUXL)
116   FØRMAT('Y2D2(K)',T11,'= '25I1)
      WRITE(1,117) (SØMYD(K),K=1,NUXL)
117   FØRMAT('-----------------------------------+'/
     1'SØMYD(K)',T11,'= '20I1)
C WHEN THE TELETYPE IS USED, THE LINE-FEEDS MUST BE ØMITTED.
      IF (MØDE.EQ.2) WRITE(1,118)
      WRITE(1,119)
118   FØRMAT(////////////////)
119   FØRMAT(/'STATE-TABLE:'/)
      DØ 120 I=1,TWØNU
120   WRITE(1,130) (TT(I,J),J=1,13)
130   FØRMAT(I3,'-',4I3,1X,4I3,1X,4I3)
      WRITE(1,131)
131   FØRMAT(/'PARENT-STATE MATRIX:'/)
      DØ 135 I=1,TWØNU
135   WRITE(1,140) TT(I,1),(TRANMX(I,J),J=1,9)
140   FØRMAT(I3,'-',4I3,1X,4I3,3X,I1)
      WRITE(1,141) (TRMC(I),I=1,TWØNU)
141   FØRMAT(/'TREEMETRIC(I)= '64I1)
      WRITE(1,142) (SSMC(I),I=1,TWØNU)
142   FØRMAT('SSMC(I)       = '64I1)
C THE DISPLAY REQUIRES THE ØUTPUT-PRØCEDURE STARTING AT LABEL
C 1004.
      IF (MØDE.EQ.2) GØTØ 1004
      WRITE(1,143)
143   FØRMAT(/'DECØDING SCHEME:')
C LØØP 150: IN ØRDER TØ FACILITATE THE DESIRED LAYØUT ØF THE
C DECØDING-SCHEME, THE ARRAY DECØUT IS INTRØDUCED.
      DØ 150 I=1,TWØNU
      DECØUT(I,1)=MTCØMB(I,1)
      DØ 150 K=1,NUXL
      DECØUT(I,4*K-2)=MTCØMB(I,K+1)
      DECØUT(I,4*K-1)=SURV(I,K+1)
      DECØUT(I,4*K  )=PATHRG(I,K+1)
      DECØUT(I,4*K+1)=TRANS(I,K+1)
150   CONTINUE
```

```
C THE NUMBER ØF CØLUMNS (SECTIØNS) EQUALS NUXL1.
      JMAX=4*NUXL1-3
      DØ 151 I=1,TWØNU
151   WRITE(1,152) (DECØUT(I,J),J=1,JMAX)
152   FØRMAT(I1,4X,39(2I2,2I1))
      WRITE(1,156) (Z(K),K=1,NUXL)
156   FØRMAT(/'DECØDED PATH:'//'Z(K)         ='I4,19I5)
      WRITE(1,157) (WESTIM(K),K=1,NUXL)
157   FØRMAT(/'WESTIM(K)    ='I4,19I5)
      WRITE(1,158) (STATE(K),K=1,NUXL1)
158   FØRMAT(/'STATE(K)     =',20(I2,'---'))
      WRITE(1,160) (NESTIM(1,K),K=1,NUXL)
160   FØRMAT(/'NESTIM(1,K) ='I4,19I5)
      WRITE(1,161) (NESTIM(2,K),K=1,NUXL)
161   FØRMAT('NESTIM(2,K) ='I4,19I5)
      WRITE(1,162) (SØMYD(K),K=1,NUXL)
162   FØRMAT(/'SØMYD(K)     = '20I1)
      WRITE(1,164) (WESTIM(K),K=1,NUXL)
164   FØRMAT('WESTIM(K)    = '20I1)
      WRITE(1,166) (XESTIM(K),K=1,NUXL)
166   FØRMAT('---------------------------+'/
     1'XESTIM(K)    = '20I1)
      WRITE(1,168) (X(K),K=1,XL)
168   FØRMAT(/,'X(K)         = '20I1)
C WHEN USING THE TELETYPE, THE PRØGRAM MUST BE ENDED.
      GØTØ 1010
C     ***********************************************************
1004  WRITE(1,170)
170   FØRMAT(//'*******************************************************'/
     1 ///'   BY GIVING A CARRIAGE RETURN, THE TRELLIS IS DRAWN'/
     1 'WITH IN THE FIRST RØW THE STEADY STATE METRICCØMBINA-'/
     1 'TIØN.'//
     1 '   NØW,BY GIVING A CARRIAGE RETURN AT EACH TIME K, THE'/
     1 'K TH SECTIØN WITH ITS DECØDING DATA WILL BE SHØWN.'/)
      READ(1,2)
      CALL ERASE
      CALL ANMØDE
      C A RECTANGLE ØF 600 TAGPØINTS IN HEIGHT IS RESERVED BENEATH THE TRELLIS
      C FØR ØUTPUT.
            CALL TWINDØ(0,4095,650,3119)
      C THE RADIUS R ØF THE CIRCLE IS ADJUSTED TØ THE NUMBER ØF SECTIØNS.
      C THE MINIMUM VALUE (NU=4 AND XL=11) ØF R EQUALS LINWDT(1) AND
      C AND THE MAXIMUM VALUE (NU=2 AND XL=2) EQUALS 3*LINWDT(1).
            R=LINWDT(1)*(41-2*NUXL)/11
            CALL TRELLI(TWØNU,NUXL,R)
            LY=LINHGT(1)/2
            LX=LINWDT(1)/2
      C LØØP 180: PRINT THE STEADY STATE METRICCØMBINATIØN IN THE NØDES.
            DØ 180 I=1,TWØNU
            CALL MØVEA(0.,FLØAT(1-I))
            CALL MØVREL(5-LX,7-LY)
            CALL ANCHØ(SSMC(I)+48)
      180   CØNTINUE
      C THE ØUTPUTVALUES Z(K),K=1,NUXL ARE TØ BE PRINTED IN THE MIDDLE
      C BETWEEN THE SECTIØNS.
            IXMULT=4096./FLØAT(NUXL+1)
            CALL TWINDØ(0,4095,0,3119)
      C MØVE THE CURSØR TØWARDS THE CØØRDINATE (10,600) BENEATH
      C THE TRELLIS.
            CALL MØVABS(10,600)
            IX=LINWDT(1)
```

```
C THE ASCII VALUES FØR THE FIVE CHARACTERS IN THE EXPRESSIØN Z(K)=
C ARE ENTERED IN THE ARRAY IADE(5).
      IADE(1)=90
      IADE(2)=40
      IADE(3)=75
      IADE(4)=41
      IADE(5)=61
      CALL ANSTR(5,IADE)
      CALL MØVABS(IXMULT-IX/2,600)
      CALL ANCHØ(Z(1)+48)
C LØØP 190: THE VALUES Z(K) ARE PRINTED IN THE MIDDLE BETWEEN THE
C NUXL1 SECTIØNS.
      DØ 190 K=2,NUXL
      CALL MØVABS(K*IXMULT-IX,600)
      CALL ANCHØ(Z(K)+48)
190   CØNTINUE
      CALL TWINDØ(0,4095,650,3119)
C EVALUATE THE STARTING PØINTS ØF THE BRANCHES AT THE EDGE ØF
C THE NØDES.
      CALL EDGCIR(R,TWØNU,NUXL,XD,YD)
C LØØP 200: EACH TIME K, FØR EACH STATE I AT TIME K+1, THE
C FØLLØWING ACTIØNS TAKE PLACE:
C THE BRANCH TØ THE SURVIVØR SURV(I,K+1) IS DRAWN, SØLID ØR
C DASHED, DEPENDENT ØN THE VALUE ØF W(K+1) IN PATHRG(I,K+1).
C THE METRICVALUE MTCØMB(I,K+1) IS PRINTED IN THE NØDE.
C THE TRANSITIØN TRANS(I,K+1) IN THE FØRM 00,01,10 ØR 11,
C IS PRINTED AT THE LEFT ØF THE NØDE.
      DØ 200 K=1,NUXL
      CALL ANMØDE
      READ(1,2)
      DØ 200 I=1,TWØNU
C CALCULATE THE DIRECTIØN ØF THE BRANCH.
      SGN=I-SURV(I,K+1)-1
C CALCULATE THE INDEX IDIS ØF THE DISTANCE BETWEEN THE TWØ
C NØDES.
      IDIS=IABS(IFIX(SGN))+1
C CALCULATE THE STARTING PØINT ØF THE BRANCH.
      XCØØR=FLØAT(K-1)+XD(IDIS)
      YCØØR=FLØAT(-SURV(I,K+1))-SIGN(YD(IDIS),SGN)
      CALL MØVEA(XCØØR,YCØØR)
C CALCULATE THE END PØINT ØF THE BRANCH.
      XCØØR=FLØAT(K)-XD(IDIS)
      YCØØR=FLØAT(1-I)+SIGN(YD(IDIS),SGN)
C DRAW THE BRANCH, SØLID ØR DASHED, DEPENDENT ØN THE VALUE
C ØF PATHRG(I,K+1).
      CALL DASHA(XCØØR,YCØØR,PATHRG(I,K+1))
C MØVE TØWARDS THE CENTRE ØF THE NØDE.
      CALL MØVEA(FLØAT(K),FLØAT(1-I))
      CALL MØVREL(5-LX,7-LY)
C PRINT THE METRICVALUE.
      CALL ANCHØ(MTCØMB(I,K+1)+48)
C PRINT THE TWØ CHARACTERS AT THE FIXED DISTANCE LINWDT(3) FRØM
C THE EDGE R ØF THE CIRCLE.
      CALL MØVREL(-IFIX(R)-LINWDT(3),0)
      ICHAR1=48
      ICHAR2=48
      IF (TRANS(I,K+1).GE.2) ICHAR1=ICHAR1+1
      IF (TRANS(I,K+1).EQ.1 .ØR. TRANS(I,K+1).EQ.3) ICHAR2=ICHAR2+1
      CALL ANCHØ(ICHAR1)
      CALL ANCHØ(ICHAR2)
200   CØNTINUE
```

```
C LØØP 210: THE DECØDED PATH IS INDICATED IN THE TRELLIS BY A RE-
C DRAWING ØF THE SEPERATE BRANCHES AT A SLIGHTLY SHIFTED PØSITIØN.
      DØ 210 K=1,NUXL
      CALL ANMØDE
      READ(1,2)
C CALCULATE THE DIRECTIØN ØF THE BRANCH.
      SGN=STATE(K+1)-STATE(K)
C CALCULATE THE INDEX IDIS ØF THE DISTANCE BETWEEN THE TWØ
C NØDES.
      IDIS=IABS(IFIX(SGN))+1
C SHIFT THE X-CØØRDINATE ØVER A DISTANCE 0.2*XD(IDIS).
      XCØØR=FLØAT(K-1)+1.2*XD(IDIS)
C IF A HØRIZØNTAL BRANCH MUST BE DRAWN, THE SHIFT IN THE
C X-DIRECTIØN MUST BE SKIPPED.
      IF (IDIS.EQ.1) XCØØR=FLØAT(K-1)+XD(IDIS)
C SHIFT THE Y-CØØRDINATE ØVER A DISTANCE 0.2*YD(IDIS).
      YCØØR=-FLØAT(STATE(K))-0.8*SIGN(YD(IDIS),SGN)
C IF A HØRIZØNTAL BRANCH MUST BE DRAWN, WE MUST
C SHIFT THE Y-CØØRDINATE ØVER A DISTANCE 0.2*YD(2).
      IF (IDIS.EQ.1) YCØØR=-FLØAT(STATE(K))+0.2*YD(2)
      CALL MØVEA(XCØØR,YCØØR)
C ANALØGØUS TØ THE CALCULATIØN ØF THE BEGIN PØINT, THE
C END PØINT IS DETERMINED.
      XCØØR=FLØAT(K)-0.8*XD(IDIS)
      IF (IDIS.EQ.1) XCØØR=FLØAT(K)-XD(IDIS)
      YCØØR=-FLØAT(STATE(K+1))+1.2*SIGN(YD(IDIS),SGN)
      IF (IDIS.EQ.1) YCØØR=-FLØAT(STATE(K+1))+0.2*YD(2)
C     CALL DASHA(XCØØR,YCØØR,WESTIM(K))
      CALL DASHA(XCØØR,YCØØR,WESTIM(K))
210   CØNTINUE
C MØVE THE CURSØR TØWARDS THE CØØRDINATE (10,600) BENEATH
C THE TRELLIS.
      CALL MØVABS(10,600)
      CALL ANMØDE
C LABEL 271-276: THE LAST CØNCLUDING ØUTPUT IS PRINTED.
      WRITE(1,271) (NESTIM(1,K),K=1,NUXL)
271   FØRMAT(//'NESTIM(1,K) = '15I2)
      WRITE(1,272) (NESTIM(2,K),K=1,NUXL)
272   FØRMAT('NESTIM(2,K) = '15I2)
      WRITE(1,273) (WESTIM(K),K=1,NUXL)
273   FØRMAT(/'WESTIM(K)   = '15I2)
      WRITE(1,274) (SØMYD(K),K=1,NUXL)
274   FØRMAT('SØMYD(K)    = '15I2)
      WRITE(1,275) (XESTIM(K),K=1,NUXL)
275   FØRMAT('---------------------------------------',
     1 '----+'/'XESTIM(K)   = '15I2)
      WRITE(1,276) (X(K),K=1,XL)
276   FØRMAT(/'X(K)        = ',15I2)
C THE CURSØR IS DIRECTED TØ A NEUTRAL PØSITIØN.
      CALL FINITT(0,0)
C TØ AVØID THE PRINTING ØF THE MESSAGE 'ØK' WE BUILD IN AN
C EXTRA CARRIAGE RETURN.
      READ(1,2)
1010  CALL EXIT
      END
C *****************************************************************
BØTTØM
```

```
C     ************************************************************************
C     SUBRØUTINE TEXT1
C     THE INTRØDUCTØRY TEXT FØR APPRØPIATE USE ØF THE VIDEØ-DISPLAY
C     PRØGRAM IS PRINTED.
C     ************************************************************
C
C
      SUBRØUTINE TEXT1
      WRITE(1,1)
1     FØRMAT('**********************************************************'
     1 /'   THIS FØRTRAN GRAPHICAL-DISPLAY PRØGRAM SHØWS YØU THE'/
     1 'CØMPLETE SYNDRØME-DECØDING PRØCEDURE ØF BINARY R=1/2 CØN'/
     1 'VØLUTIØNAL CØDES ØF CØNSTRAINTLENGTH NU=2,3 AND 4 IN THE '/
     1 'CLASS T(NU,1).'/
     1 '   THIS CLASS CØNSISTS ØF CØDES FØR WHICH BØTH THE FIRST'/
     1 'AND LAST STAGES ØF THE ENCØDERS SHIFTREGISTER ARE CØNNEC-'/
     1 'TED WITH THE TWØ MØD-2 ADDERS.'/
     1 '**********************************************************'////
     1 '   AS INPUT THE PRØGRAM ASKS THE CØDES CØNSTRAINTLENGTH,'/
     1 'ITS CØNNECTIØNPØLYNØMIALS, THE DATASEQUENCE AT THE ENCØ-'/
     1 'DER SIDE AND THE NØISESEQUENCE-PAIR ØN THE CHANNEL.'//
     1 '   THE CØDE, IN TERMS ØF THE TWØ CØNNECTIØNPØLYNØMIALS'/
     1 'C(N,K), N=1,2 AND K=1,NU+1 ØF DEGREE NU, MAY BE CHØØSEN'/
     1 'FRØM A TABLE. IN THIS TABLE ALL PØSSIBLE PØLYNØMIALS'/
     1 'HAVE BEEN FACTØRIZED IN IRREDUCIBLE PØLYNØMIALS IN ØRDER'/
     1 'TØ HELP THE USER TØ CØNSTRUCT A NØN-CATASTRØPHIC CØDE.'//
     1 '   THE DATASEQUENCE X(K), K=1,XL ØF LENGTH XL, AT THE'/
     1 'ENCØDER INPUT MAY BE CHØØSEN FREELY WITH THE RESRICTIØN'/
     1 'THAT XL > 1 AND THAT SUM NUXL=NU+XL < 16 .'//
     1 '   THE NØISE-SEQUENCEPAIR NØISE(N,K), N=1,2 AND K=1,XL ,'/
     1 'MAY ALSØ BE CHØØSEN FREELY.'////
     1 ' AT FIRST ALL THE ESSENTIAL DATA REQUIRED FØR THE DECØ-'/
     1 'DING PRØCEDURE, IS PRINTED ØUT.'//
     1 '- THE PØLYNØMIALS D(N,K), N=1,2 AND K=1,MU+1, ØF DEGREE'/
     1 '  MU<NU WHICH SATISFY C(1,K)*D(1,K) + C(2,K)*D(2,K) = 1.'/
     1 '- THE CØDES BASE STATE SB(K), K=1,NU.'/
     1 '- THE SPECIFIC ØUTPUTVALUE ZNUL.'/
     1 '- THE CØNSTRUCTIØN ØF THE CHANNEL-SEQUENCES Y(N,K), N=1,2'/
     1 '  AND K=1,NUXL AT THE DECØDER SIDE.'/
     1 '- THE ØUTPUTSEQUENCE Z(K), K=1,NUXL ØF THE W(ALPHA)-'/
     1 '  FØRMER.'/
     1 '- THE STATE-TABLE WITH THE 2**NU STATES, THE FØUR STATE-'/
     1 '  TRANSITIØNS AND THE FØUR SYNDRØME-ØUTPUTS Z FØR THE'/
     1 '  TRANSITIØNS 00,01,11 AND 10 RESPECTIVELY.'/
     1 '- THE PARENT-STATE MATRIX WITH THE 2**NU STATES, THE'/
     1 '  FØUR PARENT-STATES, THE FØUR SYNDRØME-ØUTPUTS Z AND'/
     1 '  THE ØUTPUTVALUE W.'/
     1 '  FØR THE EVEN-NUMBERED STATES WE HAVE THE FØUR PARENT-'/
     1 '  STATES FØR THE TRANSITIØNS 00,00,11 AND 11, AND FØR'/
     1 '  THE ØDD-NUMBERDED STATES WE HAVE THE FØUR PARENT-'/
     1 '  STATES FØR THE TRANSITIØNS 01,01,10 AND 10 RESPEC-'/
     1 '  TIVELY.'/
     1 '- THE METRICVALUE TREEMETRIC(I), I=1,2**NU AT DEPTH NU.'/
     1 '- THE STEADY-STATE-METRICCØMBINATIØN SSMC(I), I=1,2**NU.'////
```

```
     1 7////////////
     1 '  HEREAFTER THE COMPLETE TRELLIS-DECODING PROCEDURE IS'/
     1 '  SHOWN, STARTING WITH THE STEADY-STATE-METRIC-'/
     1 '  COMBINATION AND THE SYNDROMESEQUENCE Z(K).'///
     1 '  EACH TIME K A COMPLETE SECTION OF THE TRELLIS IS SHOWN.'/
     1 '  IN EACH SECTION FOR EACH STATE I, THE FOLLOWING DATA'/
     1 '  ARE INDICATED:'//
     1 ' - THE NEW METRICVALUE, PRINTED IN THE SPECIFIC NODE.'/
     1 ' - THE SPECIFIC NODE (STATE), CONNECTED WITH ITS SURVI-'/
     1 '   VOR BY A SOLID BRANCH FOR A PATHREGISTERBIT 0, AND'/
     1 '   BY A DASHED BRANCH FOR A PATHREGISTERBIT 1.'/
     1 ' - THE SPECIFIC TRANSITION (00,01,11 OR 10), PLACED AT'/
     1 '   THE LEFT SIDE OF THE NODE.'///
     1 '  EACH SECTION IN THE TRELLIS CAN BE GENERATED BY THE '/
     1 '  USER BY A CARRIAGE RETURN.'//
     1 '  AT THE END, THE DECODED PATH IN THE TRELLIS IS SHOWN'/
     1 '  BY MEANS OF A DOUBLING OF THE SEPERATE BRANCHES'//
     1 '  IN CONCLUSION THE ESTIMATED DATASEQUENCE XESTIM(K), '/
     1 '  K=1,XL, IS EVALUATED.'/
     1 '  FOR THAT PURPOSE THE FOLLOWING OUTCOMES ARE PRINTED'/
     1 '  OUT BENEATH THE TRELLIS:'/
     1 '- NESTIM(1,K), K=1,NUXL'/
     1 '- NESTIM(2,K), K=1,NUXL'/
     1 '- WESTIM(K) = NESTIM(1,K)*D(1,K) + NESTIM(2,K)*D(2,K)'/
     1 '- SOMYD(K)  = Y(1,K)*D(1,K) + Y(2,K)*D(2,K)'/
     1 '- XESTIM(K) = SOMYD(K) + WESTIM(K)'///
     1 '**********************************************************'////
     1 '  BY GIVING A CARRIAGE RETURN THE TABLE OF FACTORIZED'/
     1 '  POLYNOMIALS IS PRINTED.'/)
      RETURN
      END
C     END OF TEXT1
C     **********************************************************
C     SUBROUTINE FACPOL
C     ALL POLYNOMIALS OF CONSTRAINTLENGTH NU=2,3,4 AND 6 HAVE BEEN
C     FACTORIZED IN IRREDUCIBLE POLYNOMIALS.
C     THEY ARE PRINTED IN THE APPROPIATE WAY.
C     **********************************************************
C
C
      SUBROUTINE FACPOL
      WRITE(1,3)
3     FORMAT('FACTORIZED CONNECTIONPOLYNOMIALS FOR BINARY R=1/2 CONVO-'/
     1 'LUTIONAL CODES OF CONSTRAINTLENGTH NU=2,3,4,5 AND 6 IN'/
     1 'THE CLASS T(NU,1).'/
     1 'THE NOTATION FOR THESE CONNECTIONPOLYNOMIALS IS:'//
     1 'C(N,K), N=1,2 AND K=1,NU+1    WHERE C(N,1)=C(N,NU+1)=1.'//
     1 'THE POLYNOMIALS ARE OF THE FORM C(N,K)*X**(NU+1-K),K=1,NU+1, '/
     1 'WHERE X IS A FORMAL PARAMETER WHICH SERVES AS A PLACEHOLDER.'/
     1 '**********************************************************'
     1 7//' NU   POLYNOMIAL   FACTORS'//
     1 ' 2      101          (11)*(11)'/
     1 '        111          IRREDUCIBLE'//
     1 ' 3      1001         (11)*(111)'/
     1 '        1011         IRREDUCIBLE'/
     1 '        1101         IRREDUCIBLE'/
     1 '        1111         (11)*(11)*(11)'//
```

```
     1 ' 4     10001          (11)*(11)*(11)*(11)'/
     1 '       10011          IRREDUCIBLE'/
     1 '       10101          (111)*(111)'/
     1 '       10111          (11)*(1101)'/
     1 '       11001          IRREDUCIBLE'/
     1 '       11011          (11)*(11)*(111)'/
     1 '       11101          (11)*(1011)'/
     1 '       11111          IRREDUCIBLE'//
     1 ' 5     100001         (11)*(11111)'/
     1 '       100011         (111)*(1101)'/
     1 '       100101         IRREDUCIBLE'/
     1 '       100111         (11)*(11)*(1011)'/
     1 '       101001         IRREDUCIBLE'/
     1 '       101011         (11)*(11001)'/
     1 '       101101         (11)*(11)*(11)*(111)'/
     1 '       101111         IRREDUCIBLE'/
     1 '       110001         (111)*(1011)'/
     1 '       110011         (11)*(11)*(11)*(11)*(11)'/
     1 '       110101         (11*)*(10011)'/
     1 '       110111         IRREDUCIBLE'/
     1 '       111001         (11)*(11)*(1101)'/
     1 '       111011         IRREDUCIBLE'/
     1 '       111101         IRREDUCIBLE'/
     1 '       111111         (11)*(111)*(111)'////////////
     1 '////////////' 6     1000001       (11)*(11)*(111)*(111)'/
     1 '       1000011        IRREDUCIBLE'/
     1 '       1000101        (1011)*(1011)'/
     1 '       1000111        (11)*(111101)'/
     1 '       1001001        IRREDUCIBLE'/
     1 '       1001011        (11)*(11)*(11)*(1101)'/
     1 '       1001101        (11)*(111011)'/
     1 '       1001111        (111)*(11001)'/
     1 '       1010001        (1101)*(1101)'/
     1 '       1010011        (11)*(111)*(1011)'/
     1 '       1010101        (11)*(11)*(11)*(11)*(11)*(11)'/
     1 '       1010111        IRREDUCIBLE'/
     1 '       1011001        (11)*(110111)'/
     1 '       1011011        IRREDUCIBLE'/
     1 '       1011101        (111)*(11111)'/
     1 '       1011111        (11)*(11)*(10011)'/
     1 '       1100001        IRREDUCIBLE'/
     1 '       1100011        (11)*(11)*(11111)'/
     1 '       1100101        (11)*(111)*(1101)'/
     1 '       1100111        IRREDUCIBLE'/
     1 '       1101001        (11)*(11)*(11)*(1011)'/
     1 '       1101011        (111)*(111)*(111)'/
     1 '       1101101        IRREDUCIBLE'/
     1 '       1101111        (11)*(100101)'/
     1 '       1110001        (11)*(101111)'/
     1 '       1110011        IRREDUCIBLE'/
     1 '       1110101        IRREDUCIBLE'/
     1 '       1110111        (11)*(11)*(11)*(11)*(111)'/
     1 '       1111001        (111)*(10011)'/
     1 '       1111011        (11)*(101001)'/
     1 '       1111101        (11)*(11)*(11001)'/
     1 '       1111111        (1011)*(1101)'////
     1 '**********************************************************'////
     1 '   WHEN YØU HAVE CHØØSEN A PARTICULAR CØDE, YØU MAY GIVE'/
     1 'A CARRIAGE RETURN IN ØRDER TØ PRØCEED WITH INPUT')
      RETURN
      END
C     END ØF FACPØL
C     **********************************************************
```

```
C     ********************************************************
C     SUBRØUTINE TRELLI
C     THE STATE-NUMBERS 0,1,...,TWØNU-1 ARE PRINTED IN A CØLUMN
C     AT THE LEFT ØF THE SCREEN AND THE TWØNU*NUXL1 CIRCLES ØF
C     THE TRELLIS ARE DRAWN CØLUMN BY CØLUMN IN THE WINDØW
C     (X,Y)=(0-4095,0-3119).
C     ********************************************************
C
C
      SUBRØUTINE TRELLI(TWØNU,NUXL,R)
      INTEGER TWØNU
C THE NUMBER ØF TAGPØINTS IN THE X-DIRECTIØN AND THE Y-DIRECTIØN
C IS 4096 AND 3120 RESPECTIVELY. THE TRELLIS CØNSISTS ØF NUXL1
C SECTIØNS AND TWØNU RØWS.
      XMULT=4096./FLØAT(NUXL+1)
      YMULT=3120./FLØAT(TWØNU)
C THE APPRØPIATE SPACE FØR THE TRELLIS ØN THE SCREEN IS RESERVED.
      CALL DWINDØ(-0.5,FLØAT(NUXL)+0.5,-0.5-FLØAT(TWØNU-1),0.5)
C THE STATE-NUMBERS ARE PLACED IN THE CØLUMN AT PØSITIØN -0.25.
      CALL NUMBER(TWØNU,-0.25)
      CALL DWINDØ(0.,4095.,0.,3119.)
C LØØP 20: THE CIRCLES (NØDES) ARE DRAWN SECTIØN BY SECTIØN.
      DØ 20 K=0,NUXL
      XC=(FLØAT(K)+0.5)*XMULT
      DØ 20 I=1,TWØNU
      YC=3120.-(FLØAT(I)-0.5)*YMULT
      CALL CIRCLE(XC,YC,R)
20    CØNTINUE
      CALL DWINDØ(-0.5,FLØAT(NUXL)+0.5,-0.5-FLØAT(TWØNU-1),0.5)
      RETURN
      END
C     END ØF TRELLI
C     ********************************************************
C     SUBRØUTINE NUMBER
C     THE DECIMAL STATE-NUMBERS 0,1,...,TWØNU-1, ARE PRINTED IN
C     THE CØLUMN AT THE ABSØLUTE PØSITIØN XC=PLACE.
C     ********************************************************
C
C
      SUBRØUTINE NUMBER(TWØNU,PLACE)
      INTEGER ADE(7),TWØNU
C THE UNITY LINWDT(1) IS ATTACHED TØ THE INTEGER CHARACTER WIDTH
C ICHWDT AND THE UNITY LINHGT(1) TØ THE INTEGER CHARACTER
C HEIGHT ICHHGT.
      ICHWDT=LINWDT(1)
      ICHHGT=LINHGT(1)
      DØ 10 I=1,TWØNU
      CALL NRCNVT(FLØAT(I-1),0,NCHAR,ADE)
      CALL MØVEA(PLACE,-FLØAT(I-1))
      CALL MØVREL(-NCHAR*ICHWDT,-ICHHGT/2+7)
      CALL ANSTR(NCHAR,ADE)
10    CØNTINUE
      RETURN
      END
C     END ØF NUMBER
C     ******************************************************
```

```
C      SUBRØUTINE EDGCIR
C      THE BEGIN- AND ENDPØINTS ØF THE BRANCHES AT THE EDGES
C      ØF THE CIRCLES, ARE CALCULATED AND ENTERED IN THE AR-
C      RAYS XD(I) AND YD(I). XD(I) AND YD(I) ARE THE RELATIVE
C      CØØRDINATES (WITH REGARD TØ THE CENTRE) ØF THE INTER-
C      SECTIØN ØF THE BRANCH AND THE EDGE ØF A CIRCLE.
C      THIS INTERSECTIØN IS DETERMINED BY THE ANGLE
C      Q=ATAN(Y-UNIT/X-UNIT).
C      ******************************************************
C
C
       SUBRØUTINE EDGCIR(R,TWØNU,NUXL,XD,YD)
       REAL XD(16),YD(16),NURTØR
       INTEGER TWØNU
C CALCULATE THE X-UNIT.
       DENTØR=4096./FLØAT(NUXL+1)
       DØ 10 I=1,TWØNU
C CALCULATE THE Y-UNIT.
       NURTØR=3120.*FLØAT(I-1)/FLØAT(TWØNU)
       XD(I)=R*CØS(ATAN(NURTØR/DENTØR))*FLØAT(NUXL+1)/4096.
       YD(I)=R*SIN(ATAN(NURTØR/DENTØR))*FLØAT(TWØNU)/3120.
10     CØNTINUE
       RETURN
       END
C      END ØF EDGCIR
C      ******************************************************
```