ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ

Nr 102 Automatyka z.5 1964

ZDZISŁAW POGODA

Katedra Teorii Regulacji

ZASTOSOWANIE RACHUNKU MACIERZOWEGO DO SYNTEZY SIECI LOGICZNYCH

1. Pojęcie i określenia podstawowe

Jednoargumentową zmienną logiczną x nazywać będziemy literą. Litery mogą przyjmować wartości z pewnego alfabetu, za który w niniejszej pracy będzie przyjmowany system dwójkowy. Wartości liter oznaczać będziemy symbolem $\dot{x}$. W alfabecie dwójkowym może być $\dot{x}=0$ lub $\dot{x}=1$. Zmienną wieloargumentową X, stanowiącą zbiór n liter x_i nazywać będziemy słowem o długości $l\ X = n$. Każdemu słowu można przypisać pewną wartość liczbową określoną wzorem

$$\dot{X} = \sum_{i=0}^{n-1} x_i\, 2^i. \qquad (1)$$

Słowa będziemy zapisywać w postaci macierzy kolumnowych

$$\dot{X} = \begin{Vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Vmatrix} \qquad (2)$$

Na przykład słowu

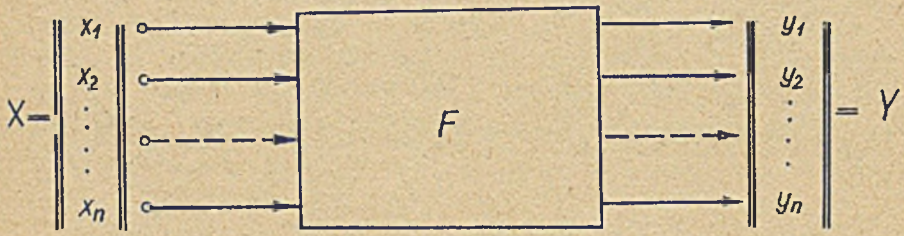
$$\dot{X} = \begin{Vmatrix} 1 \\ 0 \\ 1 \\ 1 \end{Vmatrix}$$

odpowiada wartość liczbowa

$$\dot{X} = \{1\ 1\ 0\ 1\}_2 = 13.$$

Z określeń tych wynika odwrotnie, że każdej liczbie zapisanej w systemie dwójkowym odpowiada słowo, którego pierwszą literę stanowi bit o najniższej wadze i ostatnią bit o najwyższej wadze.

Siecią logiczną nazwiemy urządzenie przekształcające słowo wejściowe X o długości lX=n w słowo wyjściowe Y o długości lY=m. Z określenia tego wynika, że sieć taka musi posiadać n kanałów wejściowych dla wprowadzenia słowa X, oraz m kanałów wyjściowych dla wyprowadzenia słowa Y (rys. 1).



Rys.1. Sieć logiczna

Jeżeli wartość słowa Y w pewnym momencie czasu zależy tylko i wyłącznie od wartości słowa X wprowadzonego na wejście sieci w tym samym momencie a nie zależy od historii sieci, to mówimy, że dana sieć nie posiada pamięci. W dalszym ciągu rozpatrywane będą tylko sieci nie posiadające pamięci. Dla sieci bez pamięci obowiązuje więc zależność

$$\underset{m\times 1}{Y} = \underset{m\times n}{F}\ \underset{n\times 1}{X} \qquad (3)$$

gdzie F - pewna macierz prostokątna charakteryzująca sposób działania sieci. Wzór (3) stanowi dokładną analogię odpowiedniego równania analogowego elementu wieloparametrowego. Dlatego celowym jest nazwać macierz F logiczną macierzą przejścia sieci. Logiczna macierz przejścia sieci jest więc macierzą prostokątną o ilości kolumn równej długości słów wejściowych i o ilości wierszy równej długości słów wyjściowych.

Słowa oraz logiczne macierze przejścia są więc macierzami, których elementy stanowią zmienne logiczne. Wartościami liczbowymi macierzy są macierze - zero-jedynkowe. Na macierzach logicznych można wykonywać pewne operacje algebraiczne.

Negacją (inwersją) macierzy logicznej nazwiemy taką macierz logiczną, której elementy są inwersjami odpowiednich elementów macierzy logicznej pierwotnej.
Jeśli

$$A = \|a_{ij}\| \quad \text{to} \quad \bar{A} = \|\bar{a}_{ij}\|. \tag{4}$$

Sumą macierzy logicznych nazywamy taką macierz logiczną, której elementy są sumami logicznymi odpowiednich elementów macierzy składowych. Jeśli

$$\left.\begin{array}{l} A = \|a_{ij}\|, \quad B = \|b_{ij}\| \quad \text{i} \quad C = A + B, \\ \text{to} \\ \qquad C = \|a_{ij} + b_{ij}\|. \end{array}\right\} \tag{5}$$

Iloczynem macierzy logicznych nazywamy taką macierz logiczną, której elementy stanowią sumy logiczne iloczynów logicznych elementów odpowiedniego wiersza pierwszej macierzy przez elementy odpowiedniej kolumny drugiej macierzy.
Jeśli

$$\left.\begin{array}{l} \underset{m\times p}{A = \|a_{ij}\|}, \quad \underset{p\times n}{B = \|b_{ij}\|} \quad \text{i} \quad \underset{m\times n}{C = AB}, \\ \text{to} \\ \qquad C = \left\|\sum_{k=1}^{p} a_{ik}\, b_{kj}\right\|. \end{array}\right\} \tag{6}$$

Oczywiście działania dodawania i mnożenia we wzorach (5) i (6) należy rozumieć w sensie logicznym. Dla dodawania i mnożenia macierzy logicznych obowiązują te same prawa dotyczące łączności, addytywności i przemienności jak dla zwykłych macierzy liczbowych.

2. Sposoby zapisu funkcji logicznych [1]

Dowolny człon koniunkcyjny lub dyzjunkcyjny funkcji logicznej słowa X, można przedstawić w następujący sposób

$$T_j^{\Pi}(A_j,X)=\prod_{k=0}^{n-1}(x_k\, a_{jk}+\bar{x}_k\, \bar{a}_{jk}), \tag{7}$$

$$T_j^{\Sigma}(A_j,X)=\sum_{k=0}^{n-1}(x_k\, a_{jk}+\bar{x}_k\, \bar{a}_{jk}), \tag{8}$$

gdzie wyrażenia pod znakiem iloczynu i sumy mają następujące wartości

$$x_k a_{jk}+\bar{x}_k\bar{a}_{jk}=\begin{cases} x_k, & \text{gdy } a_{jk}=1,\\ \bar{x}_k, & \text{gdy } a_{jk}=0,\end{cases} \tag{9}$$

oraz A_j i X – słowa o tej samej długości. Wyrażenia T_j^{Π} będziemy nazywali koniunkcjami elementarnymi a wyrażenia T_j^{Σ} – dyzjunkcjami elementarnymi słowa X a słowo A_j – słowem tworzącym koniunkcję lub dyzjunkcję o numerze porządkowym j. Dla danego słowa X o długości lX=n można utworzyć tyle różnych koniunkcyj i dyzjunkcyj ile istnieje różnych słów tworzących A_j, tzn. 2^n. Jeśli słowa tworzące uporządkować według wzrastających wartości, tzn. przyjąć $A_j = j$ $(j=0,1,\ldots,2^n-1)$, to dla słowa dwuliterowego X można utworzyć następujące koniunkcje i dyzjunkcje

A_j	0 0	1 0	0 1	1 1
T_j^{Π}	$\bar{x}_0\,\bar{x}_1$	$x_0\,\bar{x}_1$	$\bar{x}_0\,x_1$	$x_0\,x_1$
T_j^{Σ}	$\bar{x}_0+\bar{x}_1$	$x_0+\bar{x}_1$	$\bar{x}_0+x_1$	x_0+x_1

Wykonując inwersję równania (7), otrzymamy na podstawie praw de Morgana

$$\bar{T}_j^{\Pi}(A_j,X) = \sum_{k=0}^{n-1} \overline{(x_k a_{jk} + \bar{x}_k \bar{a}_{jk})} = \sum_{k=0}^{n-1} (\bar{x}_k + \bar{a}_{jk})(x_k + a_{jk}) =$$

$$= \sum_{k=0}^{n-1} (x_k \bar{a}_{jk} + \bar{x}_k a_{jk}).$$

Podobnie inwersja dyzjunkcji elementarnej będzie równa

$$\bar{T}_j^{\Sigma}(A_j,X) = \prod_{k=0}^{n-1} \overline{(x_k a_{jk} + \bar{x}_k \bar{a}_{jk})} = \prod_{k=0}^{n-1} (x_k \bar{a}_{jk} + \bar{x}_k a_{jk}).$$

Porównując znalezione inwersje z określeniami (7) i (8) otrzymujemy następującą parę zależności

$$\bar{T}_j^{\Pi}(A_j,X) = T_j^{\Sigma}(\bar{A}_j,\bar{X}), \tag{10}$$

$$\bar{T}_j^{\Sigma}(A_j,X) = T_j^{\Pi}(\bar{A}_j,X). \tag{11}$$

Dzięki symetrii pod znakiem iloczynu i sumy w określeniach (7) i (8), oczywista też jest następująca para zależności

$$T_j^{\Pi}(\bar{A}_j,X) = T_j^{\Pi}(A_j,\bar{X}), \tag{12}$$

$$T_j^{\Sigma}(\bar{A}_j,X) = T_j^{\Sigma}(A_j,\bar{X}). \tag{13}$$

Z zależności (3) wynikają następujące ważne własności koniunkcyj i dyzjunkcyj elementarnych

$$T_j^{\Pi}(A_j,A_j) = T_j^{\Sigma}(A_j,A_j) = 1, \tag{14}$$

$$T_j^{\Pi}(A_j,\bar{A}_j) = T_j^{\Sigma}(A_j,\bar{A}_j) = 0. \tag{15}$$

Własności te dowodzą, że wśród zbioru wszystkich 2^n koniunkcyj lub dyzjunkcyj elementarnych dla każdej możliwej wartości słowa X, istnieje koniunkcja lub dyzjunkcja elementarna równa jedności i zeru. Wynika stąd, że suma wszystkich koniunkcyj elementarnych jest tożsamościowo równa jedności

$$1 = \sum_{j=0}^{2^n-1} T_j^{\Pi}(A_j, X), \quad \text{gdzie } \dot{A}_j = j. \tag{16}$$

Podobnie iloczyn wszystkich dyzjunkcyj elementarnych jest tożsamościowo równy zeru

$$0 = \prod_{j=0}^{2^n-1} T_j^{\Sigma} (A_j, X), \quad \text{gdzie} \quad \dot{A}_j = j. \tag{17}$$

Koniunkcje elementarne bywają z tego powodu nazywane <u>składnikami jedynki</u> a dyzjunkcje elementarne - <u>czynnikami zera</u>.

Sumując tylko niektóre spośród koniunkcyj elementarnych i mnożąc tylko niektóre spośród dyzjunkcyj elementarnych, można uzyskać funkcje logiczne n argumentowe, przybierające wartość 1 tylko dla określonych słów X, dla pozostałych zaś - wartość zero. Operacja sumowania lub mnożenia wybranych koniunkcyj lub dyzjunkcyj może być zapisana za pomocą <u>słowa tworzącego</u> B o długości $lB = 2^n$, w następujący sposób

$$y(X) = R^{\Sigma}(B,X) = \sum_{j=0}^{2^n-1} \left[b_j\, T_j^{\Pi} (A_j, X)\right], \; A_j = j, \tag{18}$$

$$y(X) = R^{\Pi}(B,X) = \prod_{j=0}^{2^n-1} \left[b_j + T_j^{\Sigma}(A_j, X)\right], \; A_j = j. \tag{19}$$

Przedstawienie funkcji logicznej za pomocą operatora R^{Σ} nazywa się jej <u>normalną formą dyzjunkcyjną</u>, natomiast przedstawienie za pomocą operatora R^{Π} <u>normalną formą koniunkcyjną</u>. Normalna forma dyzjunkcyjna składa się tylko z tych koniunkcyj elementarnych, których numery odpowiadają numerom liter słowa tworzącego B - równym jedności. Normalna forma koniunkcyjna utworzona jest tylko z tych dyzjunkcyj, których numery odpowiadają numerom liter słowa tworzącego B - równym zeru.

Wykonując inwersję normalnej formy dyzjunkcyjnej, otrzymamy

$$\bar{R}^{\Sigma}(B,X) = \overline{\sum\left[b_j\, T_j^{\Pi}(A_j, X)\right]} = \prod \overline{\left[b_j\, T_j^{\Pi} (A_j, X)\right]} =$$

$$= \prod \left[\bar{b}_j + \bar{T}_j^{\Pi} (A_j, X)\right].$$

Podobnie inwersja normalnej formy koniunkcyjnej będzie równa

$$\bar{R}^{\Pi}(B,X)=\prod\overline{\left[b_j+T_j^{\Sigma}(A_j,X)\right]}=\sum\overline{\left[b_j+T_j^{\Sigma}(A_j,X)\right]}=\sum\left[b_jT_j^{\Sigma}(A_j,X)\right].$$

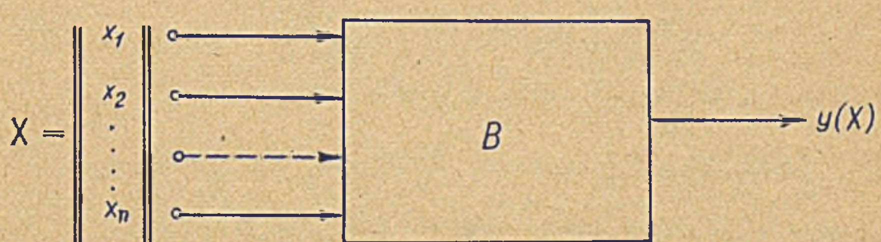
Porównując te wyniki z wyrażeniami (18) i (19) oraz uwzględniając zależności (10) - (13), znajdujemy ostatecznie

$$\bar{R}^{\Sigma}(B,X) = R^{\Pi}(\bar{B},\bar{X}), \tag{20}$$

$$\bar{R}^{\Pi}(B,X) = R^{\Sigma}(\bar{B},\bar{X}). \tag{21}$$

3. Sieci logiczne jednowyjściowe

Równania normalne (18) lub (19) opisują działanie sieci logicznej posiadającej n kanałów wejściowych dla wprowadzania słowa X oraz jeden kanał wyjściowy dla wyprowadzania jednoliterowego słowa wyjściowego y(X) (rys. 2.



Rys. 2. Sieć jednowyjściowa

Działanie lub też strukturę sieci jednowyjściowej określa całkowicie słowo B. Wynika stąd, że przy ograniczonej liczbie wejść, czyli długości słowa wejściowego, liczba różnych sieci jednowyjściowych przetwarzających informację binarną wynosi 2^n. Zgodnie z określeniem (3), macierz przejścia takiej sieci musi być macierzą wierszową.

Stosując definicję iloczynu macierzy logicznych (6), równanie dla normalnej formy dyzjunkcyjnej można zapisać w wygodny sposób za pomocą symboliki macierzowej

$$y(X) = R^{\Sigma}(B,X) = \|b_j\|^T \, \|T_j^{\,n}\|, \tag{22}$$

gdzie $\|b_j\|^T$ - transpozycja macierzy kolumnowej - słowa B czyli macierz wierszowa posiadająca 2^n elementów - liter słowa B oraz $\|T_j^{\Pi}\|$ - macierz kolumnowa, której elementami są wszystkie koniunkcje elementarne słowa X w ilości 2^n. Używając dużych liter dla oznaczenia macierzy, możemy (22) zapisać w postaci

$$y(X) = B^T \, T^{\Pi}. \tag{23}$$

Przy pomocy symboliki macierzowej łatwo jest zapisać macierz kolumnową dyzjunkcyj elementarnych. Otwierając nawiasy w określeniu (8), możemy napisać

$$T_j^{\Sigma}(A_j,X) = \sum_{k=0}^{n-1} x_k\, a_{jk} + \sum_{k=0}^{n-1} \bar{x}_k \bar{a}_{jk}, \tag{24}$$

lub zgodnie z definicją iloczynu macierzy (6)

$$T_j^{\Sigma}(A_j,X) = A_j^T\, X + \bar{A}_j^T\, \bar{X}. \tag{25}$$

Uwzględniając własność (10), otrzymujemy stąd

$$\bar{T}_j^{\Pi}(A_j,X) = T_j^{\Sigma}(\bar{A}_j,X) = \bar{A}_j^T\, X + A_j^T\, \bar{X}. \tag{26}$$

Ustawiając teraz wszystkie dyzjunkcje (26) w macierz kolumnową, znajdujemy

$$\bar{T}^{\Pi} = \begin{Vmatrix} \bar{A}^T X \\ \bar{A}_1^T X \\ \vdots \\ \bar{A}_{2^n-1}^T X \end{Vmatrix} + \begin{Vmatrix} A_0^T \bar{X} \\ A_1^T \bar{X} \\ \vdots \\ A_{2^n-1}^T \bar{X} \end{Vmatrix} = \begin{Vmatrix} \bar{A}_0^T \\ \bar{A}_1^T \\ \vdots \\ \bar{A}_{2^n-1}^T \end{Vmatrix} X + \begin{Vmatrix} A_0^T \\ A_1^T \\ \vdots \\ A_{2^n-1}^T \end{Vmatrix} \bar{X}.$$

Wprowadzając teraz oznaczenie

$$A = \begin{Vmatrix} A_0^T \\ A_1^T \\ \cdot \\ \cdot \\ \cdot \\ A_{2^n-1}^T \end{Vmatrix}, \quad \text{gdzie} \quad \dot{A}_j = j, \qquad (27)$$

otrzymujemy ostatecznie

$$\overline{T}^n = \overline{A}\,X + A\,\overline{X}, \qquad (28)$$

gdzie macierz A jest macierzą prostokątną o wymiarach $2^n \times n$. Wiersze tej macierzy stanowią zgodnie z (27) transponowane słowa A_j o wartościach kolejno wzrastających od zera do 2^n-1. Lewa kolumna macierzy A stanowi więc pozycję o najniższych wagach a prawa o najwyższych. Macierz A zbudowana jest więc z kolejnych liczb w systemie dwójkowym zapisanych w porządku odwrotnym, niż się to zwykle robi. Np. dla sieci o dwu wejściach macierz A będzie miała następującą budowę

$$A = \begin{Vmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{Vmatrix} \quad \text{oraz} \quad \overline{A} = \begin{Vmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{Vmatrix}.$$

Otrzymaliśmy więc w ten sposób następujący układ dwóch macierzowych równań logicznych opisujących działanie lub strukturę sieci logicznej bez pamięci o jednym wyjściu i n wejściach

$$\left.\begin{aligned} y(X) &= B\,T^n, \\ \overline{T} &= \overline{A}\,X + A\,\overline{X}. \end{aligned}\right\} \qquad (29)$$

Z równań tych wynika, że związek między słowem wyjściowym i wejściowym dany jest za pośrednictwem dwóch macierzy zero-jedynkowych: A i B. Budowa pierwszej z tych macierzy nie zależy od struktury sieci. Dokładniej, od struktury sieci zależą jedynie wymiary macierzy A: $2^n \times n$.
Dlatego macierz A będziemy nazywali macierzą ogólną sieci. Struktura sieci wyznaczona jest całkowicie przez macierz B, dla której przyjmiemy nazwę macierzy szczególnej sieci.

Wszystkie sieci o tej samej liczbie wejść posiadają identyczne macierze ogólne i różnią się jedynie macierzami szczególnymi.

4. Sieci logiczne wielowyjściowe

W przypadku sieci o m wyjściach, tzn. wydającej słowa wyjściowe Y o długości $lY=m$ przedstawionej na rys. 1, można wypisać m funkcji typu (18) opisujących litery składowe słowa wyjściowego

$$y_i(X)=R_i^{\Sigma}(B_i\ X) = \sum_{j=0}^{2^n-1} \left[b_{ij}\ T_j^{\Pi}\ (A_j,\ X),\right. \tag{30}$$

$$i = 1,2,\ldots,\ n, \quad \dot{A}_j = j.$$

Wszystkie równania (30) można zapisać w postaci macierzowej analogicznej do (23)

$$y_i(X) = B_i^T\ T^{\Pi}, \tag{31}$$

gdzie B_i - słowa - kolumny określające działanie każdego kanału wyjściowego. Ustawiając wszystkie litery y_i w macierz kolumnową otrzymujemy słowo wyjściowe

$$Y = B\ T^{\Pi}, \tag{32}$$

gdzie B - macierz prostokątna, której wiersze stanowią kolejne słowa tworzące o numerach odpowiadających kolejnym wyjściom sieci.

Macierzowe równania logiczne sieci o n wejściach i m wyjściach mają więc postać

$$\left.\begin{aligned} \underset{m\mathrm{x}1}{Y} &= \underset{m\mathrm{x}2^n}{B}\ \underset{2^n\mathrm{x}1}{T^{\Pi}}, \\ \underset{2^n\mathrm{x}1}{\bar{T}^{\Pi}} &= \underset{2^n\mathrm{x}n}{\bar{A}}\ \underset{n\mathrm{x}1}{X} + \underset{2^n\mathrm{x}n}{A}\ \underset{n\mathrm{x}1}{\bar{X}} \end{aligned}\right\} \tag{33}$$

Macierz szczególna sieci wielowyjściowej jest więc macierzą prostokątną o wymiarach m x 2^n, natomiast macierz ogólna, której wymiary zależą jedynie od ilości wejść, jest taka sama dla sieci jednowyjściowej jak i dla sieci wielowyjściowej.

5. Macierz przejścia sieci logicznej

Wprowadzimy operator negacji, określony zależnością

$$Nx_i = \bar{x}_i. \tag{34}$$

Drugie z równań (33) można wtedy zapisać formalnie w następujący sposób

$$\bar{T}^{II} = \bar{A}X + A\bar{X} = \bar{A}X + ANX = (\bar{A} + AN)X. \tag{35}$$

Macierz

$$Q = \bar{A} + A\,N \tag{36}$$

nazwiemy operatorem ogólnym sieci logicznej. Np. dla sieci o dwu wejściach operator ogólny ma postać

$$Q = \begin{Vmatrix} 1 & 1 \\ N & 1 \\ 1 & N \\ N & N \end{Vmatrix} \tag{37}$$

Równanie (35) przybiera więc formę

$$\bar{T} = Q\,X, \tag{38}$$

gdzie mnożenie macierzy Q oraz X rozumiane jest w zwykłym sensie. Ponieważ operator N działa w równaniu (35) zawsze ma macierz kolumnową, której wyrazami są pojedyncze zmienne logiczne, więc w bardziej złożonych równaniach, może on również działać tylko na jedną zmienną, a więc nie obowiązuje dla niego prawo łączności mnożenia względem dodawania

$$Nx_1 + Nx_2 = \bar{x}_1 + \bar{x}_2. \tag{39}$$

Oznacza to, że operatora N nie wolno wynosić przed nawias.

Wprowadzimy drugi operator negacji dla macierzy koniunkcyj elementarnych

$$M^{II}T = \bar{T}^{II}. \tag{40}$$

Tutaj operator M odnosi się do każdego wiersza macierzy kolumnowej T^{II} z osobna. Ponieważ jednak elementami macierzy T^{II} są funkcje logiczne, więc w celu oznaczenia do którego wiersza macierzy T^{Π} odnosi się operacja M; należy przyjąć

$$M = \begin{Vmatrix} M_o & & & \\ & M_1 & & \\ & & \ddots & \\ & & & M_{2^n-1} \end{Vmatrix} \tag{41}$$

Wynika stąd, że operator M_i musi działać łącznie na wszystkie zmienne; np.

$$M_i\, x_1 + M_i\, x_2 = M_i(x_1+x_2) = \bar{x}_1\, \bar{x}_2 \tag{42}$$

Operator M_i nie spełnia więc prawa rozdzielności mnożenia względem dodawania.

Stosując operator M, pierwsze z równań (33) można zapisać w postaci

$$Y = B\, M\, \bar{T}^{\Pi}. \tag{43}$$

Macierz

$$P = B\, M \tag{44}$$

nazwiemy <u>operatorem szczególnym</u> sieci logicznej. Podstawiając do równania (43) wyrażenie (38), otrzymamy ostateczną postać macierzowego równania sieci logicznej:

$$\underset{m \times 1}{Y} = \underset{m \times n}{F}\ \underset{n \times 1}{X}, \tag{45}$$

gdzie

$$\underset{m \times n}{F} = \underset{m \times 2^n}{P}\ \underset{2^n \times n}{Q} \tag{46}$$

poszukiwana macierz przejścia sieci. Macierz przejścia sieci jest więc iloczynem operatora szczególnego i ogólnego sieci. Dla znalezienia macierzy przejścia konieczna jest znajomość macierzy szczególnej sieci, wyznaczającej całkowicie jej strukturę. Macierz szczególna posiada prostą interpretację fizyczną, dzięki której łatwo ją znaleźć. Przypuśćmy, że na wejście sieci wprowadzone zostało słowo równe dowolnemu ze słów - wierszy macierzy ogólnej A_i, czyli $X = A_i$. Zgodnie z (26) mamy wtedy

$$\bar{T}_j^{\Pi}(A_j, A_i) = \bar{A}_j^T A_i + A_j^T \bar{A}_i. \tag{47}$$

Łatwo sprawdzić, że wśród wszystkich dyzjunkcyj $\bar{T}_j^n$ będzie tylko jedna równa zeru (dla i=j). Oznacza to, że wszystkie dyzjunkcje $\bar{T}_j^n$ z wyjątkiem $\bar{T}_i^n$ są równe jedności

$$T_j^n\,(A_j,\ A_i) = \begin{cases} 0 & \text{dla } j=i, \\ 1 & \text{dla } j\neq i. \end{cases} \tag{48}$$

Inwersje tych dyzjunkcyj mają więc następujące wartości

$$T_j^n\,(A_j,\ A_i) = \begin{cases} 1 & \text{dla } j=i, \\ 0 & \text{dla } j\neq i. \end{cases} \tag{49}$$

Macierz kolumnowa koniunkcyj T^n posiada więc w i-tym wierszu jedynkę a wszystkie pozostałe elementy równe są zeru. Podstawiając ten wynik do równania (32) i dzieląc macierz B na słowa kolumny, otrzymamy

$$Y = \left\| B_0,\ B_1,\ \ldots,\ B_i,\ \ldots\ B_{2^n-1} \right\| \begin{array}{c} \left\| \begin{matrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{matrix} \right\| \end{array} \begin{array}{l} 0 \\ 1 \\ \vdots \\ i \\ \vdots \\ 2^n-1 \end{array} = B_i \tag{50}$$

Równość ta oznacza, że dla $\dot{X} = i$ słowo wyjściowe $Y\,(A_i)$ równe jest słowu kolumnowemu B_i macierzy B. Porządkując więc wszystkie słowa wejściowe według wzrastającej wartości i wypisując kolejne żądane odpowiedzi sieci otrzymujemy uporządkowany zbiór słów kolumnowych, który jest identyczny z macierzą B. Macierz B stanowi więc program działania sieci dla wzrastających wartości słów wejściowych.

Przykład. Sumator dwuwejściowy.
Zbiór słów wejściowych

$$\{X\} = \begin{Bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{Bmatrix} \begin{array}{l} \text{I składnik} \\ \text{II składnik} \end{array}$$

Zbiór słów wyjściowych

$$\{Y\} = \begin{Bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{matrix} \text{suma} \\ \text{przeniesienie} \end{matrix}$$

a więc macierz szczególna będzie równa

$$B = \begin{Vmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Vmatrix}$$

i stąd operator szczególny

$$P = BM = \begin{Vmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Vmatrix} \begin{Vmatrix} M_0 & & & \\ & M_1 & & \\ & & M_2 & \\ & & & M_3 \end{Vmatrix} = \begin{Vmatrix} 0 & M_1 & M_2 & 0 \\ 0 & 0 & 0 & M_3 \end{Vmatrix}.$$

Operator ogólny określony jest przez równość (37), wobec czego macierz przejścia sumatora będzie równa

$$F = PQ = \begin{Vmatrix} 0 & M_1 & M_2 & 0 \\ 0 & 0 & 0 & M_3 \end{Vmatrix} \begin{Vmatrix} 1 & 1 \\ N & 1 \\ 1 & N \\ N & N \end{Vmatrix} = \begin{Vmatrix} M_1N+M_2 & M_1+M_2N \\ M_3N & M_3N \end{Vmatrix}.$$

Równanie sumatora będzie więc miało postać

$$\begin{Vmatrix} y_1 \\ y_2 \end{Vmatrix} = \begin{Vmatrix} M_1\ N + M_2 & M_1 + M_2N \\ M_3N & M_3N \end{Vmatrix} \begin{Vmatrix} x_1 \\ x_2 \end{Vmatrix} =$$

$$= \begin{Vmatrix} M_1(\bar{x}_1+x_2) + M_2(x_1+\bar{x}_2) \\ M_3(\bar{x}_1+\bar{x}_2) \end{Vmatrix} = \begin{Vmatrix} x_1\bar{x}_2+\bar{x}_1x_2 \\ x_1x_2 \end{Vmatrix}$$

a stąd

$$y_1 = x_1\ \bar{x}_2 + \bar{x}_1\ x_2 \quad \text{suma},$$

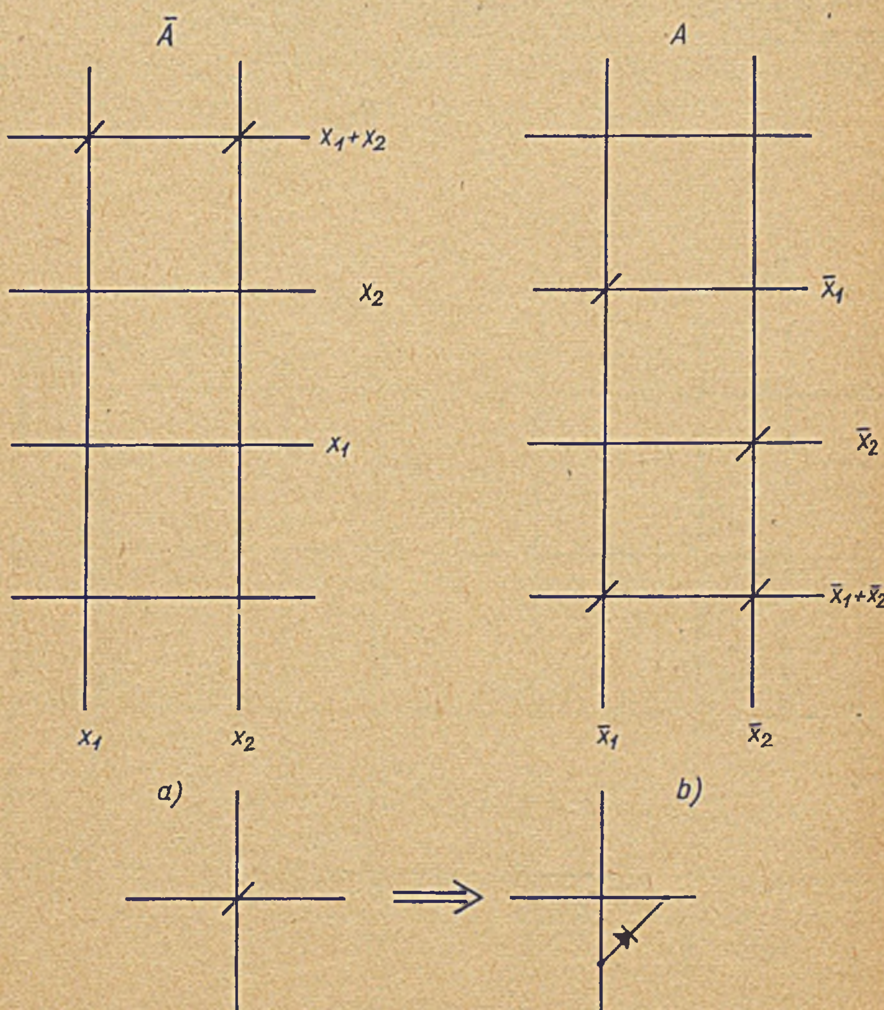
$$y_2 = x_1\ x_2 \quad \text{przeniesienie}.$$

6. Synteza sieci logicznych za pomocą matryc diodowych

Realizację techniczną sieci o danej macierzy przejścia przeprowadzimy w dwóch etapach.

1. Realizacja operatora ogólnego, którego wynikiem jest utworzenie wszystkich dyzjunkcyj $\bar{T}^n$. Operator ogólny składa się z dwóch części, zgodnie z równaniem (36)

$$T^n = Q\,X = \bar{A}\,X + A\,\bar{X}. \qquad (51)$$



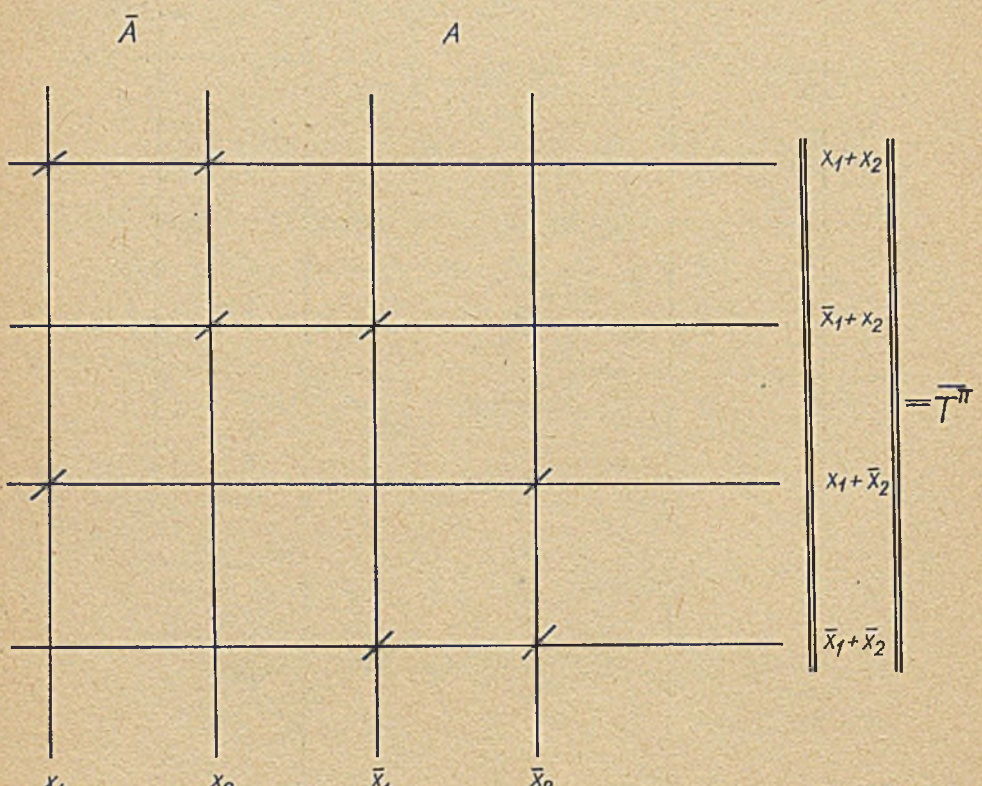
Rys. 3. Realizacja matrycowa operatora ogólnego

Realizację techniczną łatwo można uzyskać za pomocą matryc diodowych. W pierwszym składniku słowem wejściowym jest X a w drugim $\bar{X}$. Struktura obydwu matryc wyznaczona jest przez macierze A i $\bar{A}$. Przykłady takich matryc dla słów dwuliterowych pokazane są na rys. 3a i b.

Ponieważ w matrycach przedstawionych na rys. 2 na poszczególnych szynach wyjściowych następuje sumowanie, więc zgodnie z (51) można je połączyć równolegle, uzyskując w ten sposób wynik działania operatora ogólnego (rys. 4).

2. Zmieniając teraz kierunek włączenia diod (rys.5) oraz wprowadzając zasilanie szyn wyjściowych i zmieniając miejscami macierze A i $\bar{A}$, uzyskujemy inwersję wszystkich dyzjunkcyj, czyli wynik działania operatora

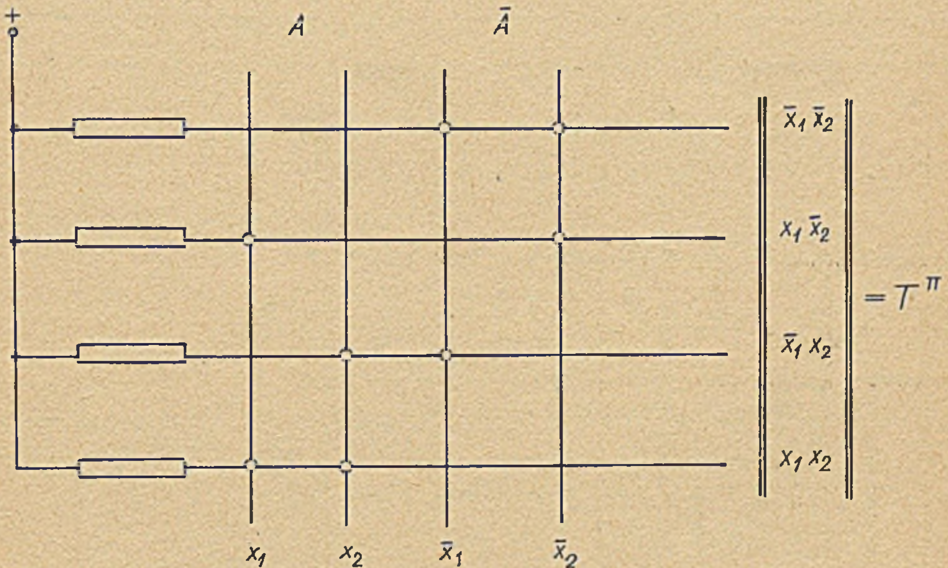
$$M\,Q\,X = T^{\pi}. \tag{52}$$

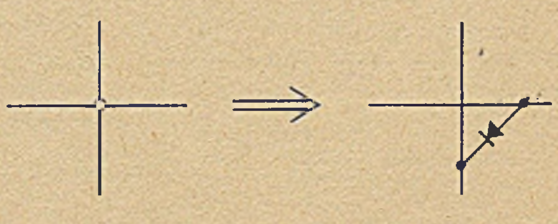


Rys. 4. Realizacja matrycowa operatora ogólnego

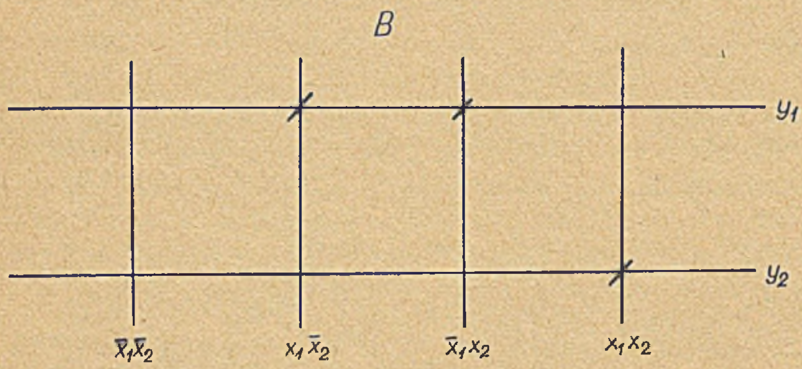
Ostatnim krokiem jest realizacja operacji reprezentowanej przez macierz B polegający na wybraniu odpowiednich koniunkcyj T_i^{Π} i zsumowaniu ich na szynach wyjściowych.
Wynik działania operatora B_i można uzyskać za pomocą matryc diodowych przedstawionych na rys. 3. Musi to być matryca posiadająca 2^n wejść i m wyjść, dla której sygnałami wejściowymi są wszystkie koniunkcje T_i^{Π}. Przykład realizacji operatora B dla sumatora dwuwejściowego pokazany jest na rys. 6.
Łącząc teraz kaskadowo matryce pokazane na rys. 5 i 6, otrzymujemy pełny schemat sumatora na dwa wejścia (rys. 7).

Z rys. 7 wynika, że węzły mnożące należy umieścić w miejscach, gdzie w macierzach A i $\bar{A}$ występują zera. Natomiast węzły sumujące pojawiają się w miejscach, gdzie w macierzy B występują jedynki. Macierz B należy obrócić w płaszczyźnie rysunku o 180°.

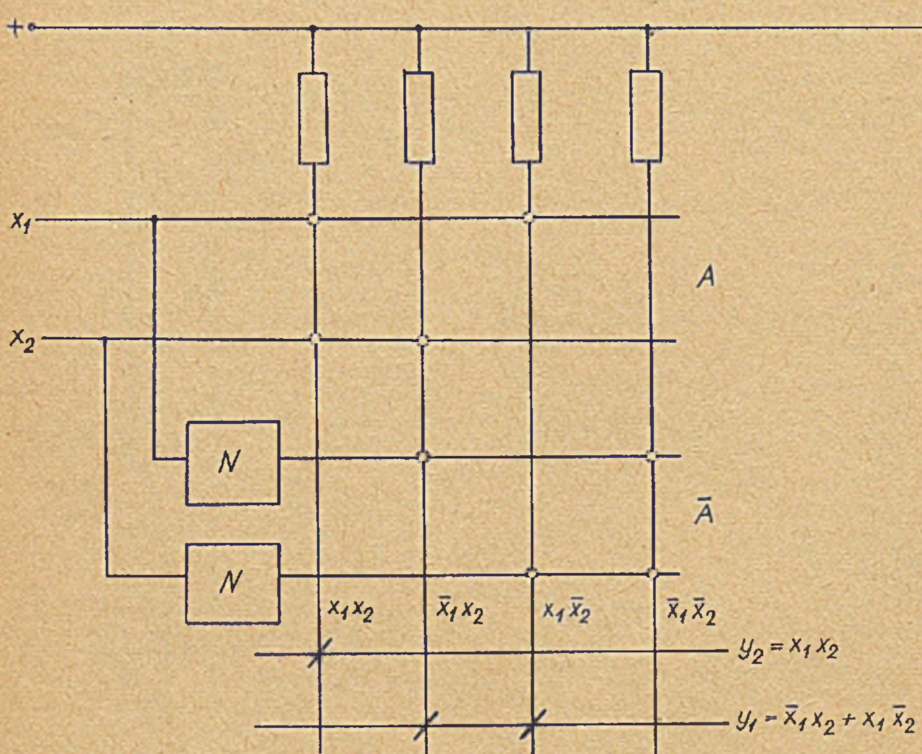




Rys. 5. Realizacja matrycowa operatora MQ



Rys.6. Realizacja operatora B dla sumatora dwuwejściowego



Rys.7. Realizacja matrycowa półsumatora

Przykład 1. Sumator na trzy wejścia.

Zbiór słów wejściowych

$$\{X\} = \begin{Bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{Bmatrix}$$

Zbiór słów wyjściowych

$$\{Y\} = \begin{Bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{Bmatrix}$$

Macierz szczególna

$$B = \begin{Vmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{Vmatrix}$$

Realizacja matrycowa sumatora pełnego pokazana jest na rys. 8.

Przykład 2. Deszyfrator kodu 5421 na kod 753-6.

Zbiór słów wejściowych

$$\{X\} = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{Bmatrix}$$

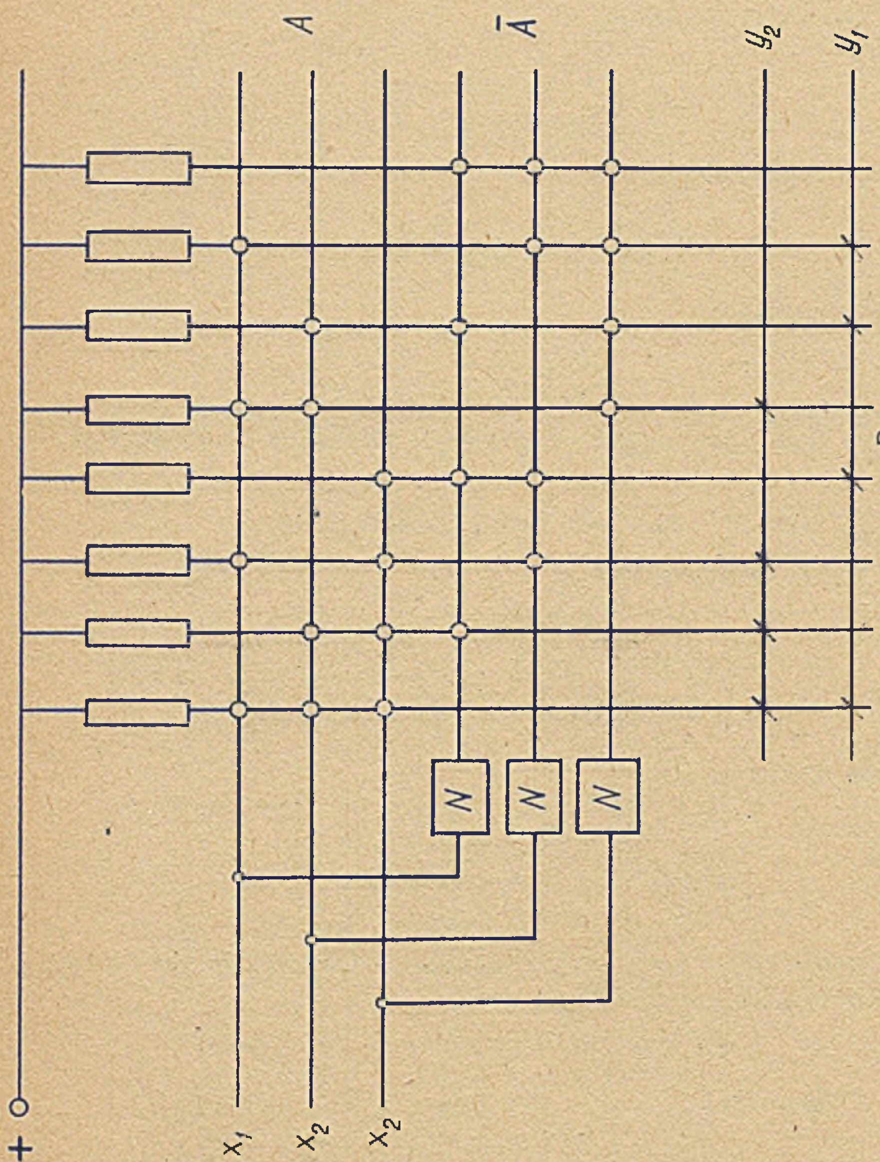
Zbiór słów wyjściowych

$$\{Y\} = \begin{Bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{Bmatrix}$$

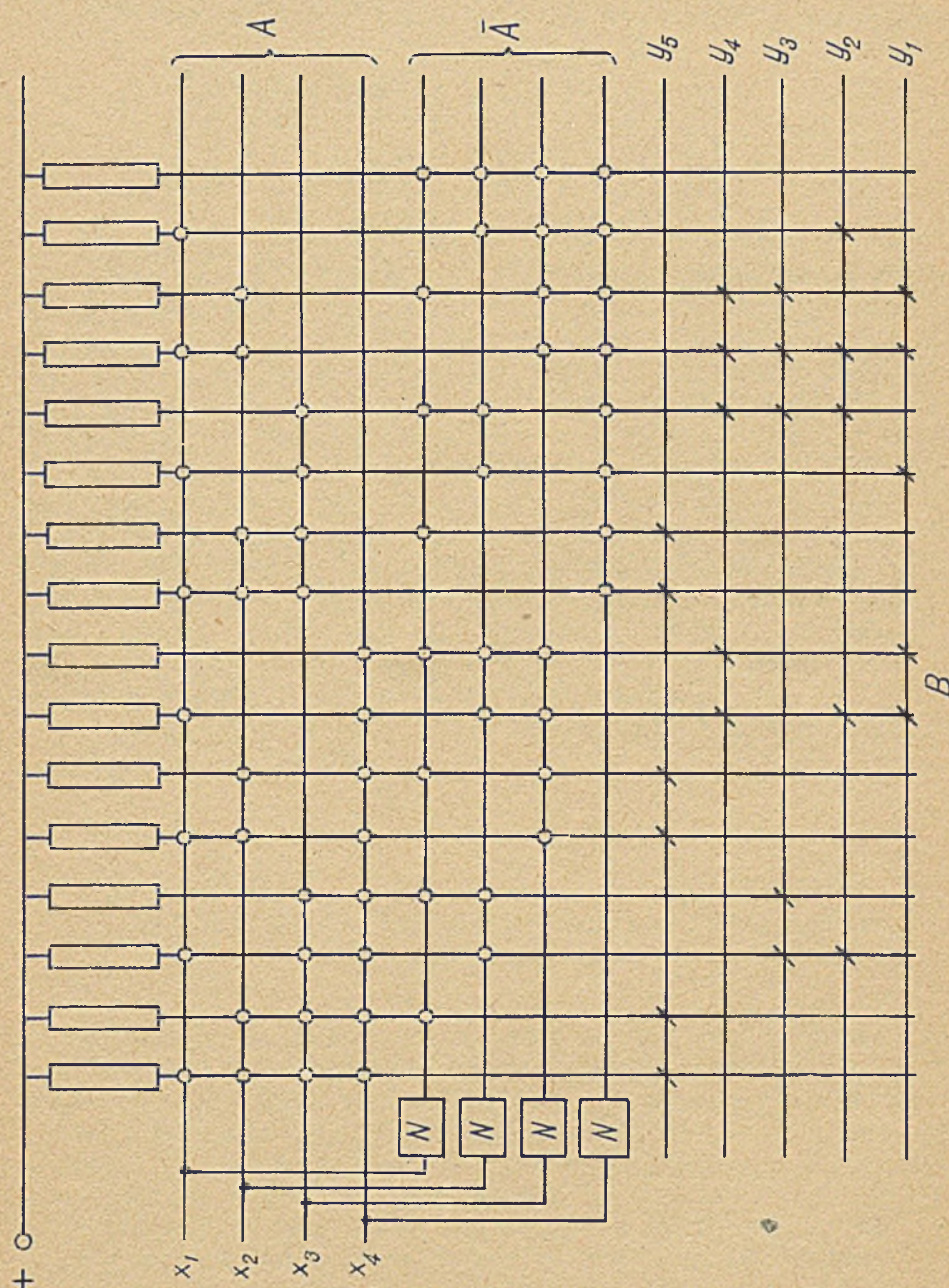
Uporządkujemy teraz wszystkie możliwe słowa wejściowe według wzrastających wartości

$$\{X\} = \begin{Bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{Bmatrix}$$

W zbiorze tym występują słowa, które nie posiadają sensu. Można zażądać aby w przypadku pojawienia się takiego słowa na wejściu słowo wyjściowe miało wartość zero i aby jednocześnie w piątym bicie sytuacja taka została zasygnalizowana.



Rys.8. Realizacja matrycowa sumatora pełnego



Rys.9. Realizacja matrycowa deszyfratora kodów

Uporządkowany zbiór słów wyjściowych, równoważny macierzy B będzie

$$B = \begin{Vmatrix} 0&0&1&1&0&1&0&0&1&1&0&0&0&0&0&0 \\ 0&1&0&1&1&0&0&0&0&1&0&0&0&1&0&0 \\ 0&0&1&1&1&0&0&0&0&0&0&0&1&1&0&0 \\ 0&0&1&1&1&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1 \end{Vmatrix}$$

Schemat deszyfratora przedstawiony jest na rys. 9.

LITERATURA

[1] Bazylewski, J.J.: Woprosy wremiennych logiczeskich funkcyj, Woprosy matematiczeskich maszyn I, Moskwa 1958.

Rękopis złożono w Redakcji w dniu 15.8.1963r.

ПРИМЕНЕНИЕ МАТРИЧНОГО ИСЧИСЛЕНИЯ ДЛЯ СИНТЕЗА ЛОГИЧЕСКИХ СХЕМ

Резюме

В статье представлена попытка применения матричного исчисления для описания логических схем не включающих элементов памяти. Рассматриваются схемы преобразующие n - разрядное входное слово на m - разрядное выходное слово. Введен матричный способ записи n - аргументных логических функций. Синтез распределен на два шага: синтез всех возможных коньюнктивных членов описанных так называемым общим оператором и выборка коньюнктивных членов, входящих в данную функцию. Последняя операция задана с помощью частного оператора. Произведение этих двух матричных операторов названо переходной матрицей логической схемы. Умножая входное слово на переходную матрицу, получаем выходное слово.

В заключении даны примеры синтеза сумматоров и кодирующих схем.

APPLICATION OF MATRIC-CALCULUS FOR SYNTHESIS OF LOGICAL NETWORKS

Summary

The author describes an attempt in the application of matric-calculus for description of complicated logical networks in which memory elements are not presented. Networks converting n-bit input-word into m-bit output-word are discussed. The author uses a matrix way to perform a n-argument logical function. The process of synthesis is devided into two steps: the first one being a synthesis of all possible conjunctions represented by a so called "common operator" and the second one accomplished by a choise of the proper conjunctions creating the given function and represented by the s.c. "particular operator".

The product of these two matrix operators has been called "transfer matrix" of the logical network. So defined a trans fer matrix converts the input-word into the output-word. Some examples illustrating the described method of the synthesis of adder and coding systems are given in the paper.