ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ

Nr 102 Automatyka z.5 1964

OLGIERD PALUSIŃSKI

Katedra Teorii Regulacji

RACHUNEK KWANTYFIKATORÓW W ZASTOSOWANIU DO SYNTEZY SIECI LOGICZNYCH

Streszczenie. W artykule przedstawiono możliwości wykorzystania zawężonego rachunku kwantyfikatorów do obliczania sieci logicznych zawierających elementy pamięci. Ogólne zasady poparto przykładami syntezy konkretnych sieci logicznych często spotykanych w technice cyfrowej. Artykuł jest rezultatem prac nad zagadnieniami automatów skończonych prowadzonych w Katedrze Teorii Regulacji.

1. Wprowadzenie

Ze względu na metody obliczeń problemy syntezy sieci logicznych można podzielić na dwie grupy:

a) synteza sieci z elementami pamięci,
b) synteza sieci bez elementów pamięci.

Metody syntezy sieci nie zawierających elementów pamięci są stosunkowo dobrze znane. Zagadnienia syntezy sieci logicznych z elementami pamięci są natomiast mniej rozwinięte i stosunkowo mało spopularyzowane.

W artykule omawia się przede wszystkim syntezę sieci logicznych z elementami pamięci przy pomocy rachunku kwantyfikatorów.

Proces syntezy sieci logicznych można podzielić na trzy etapy:

I. Opisanie własności sieci (lub inaczej tzw. Operatora [1]) w jakimś języku matematycznym - w naszym przypadku w formie wyrażenia kwantyfikatorowego.

II. Przejście z zapisu w danym języku do równań kanonicznych.

III. Konstrukcja sieci logicznej (zbudowanej z określonych elementów) w oparciu o równania kanoniczne [1].

Przedmiotem artykułu są pierwsze dwa etapy syntezy sieci dających się opisać za pomocą formuły zawężonego rachunku kwantyfikatorów. Przedstawioną poniżej metodę dość łatwo rozszerza się na wszystkie sieci, które spełniają warunki automatu skończonego. Używa się wtedy rozszerzonego rachunku kwantyfikatorów nazywanego również rachunkiem kwantyfikatorów wyższych rzędów.

Problemy III etapu syntezy są to zagadnienia typu:jak na podstawie równań algebry Boole'a skonstruować sieć składającą się z określonych elementów rzeczywistych.Problemy te wykraczają poza ramy niniejszego artykułu.

2. Podstawowe własności rachunku kwantyfikatorów

P r e d y k a t e m jednoargumentowym nazywamy funkcję, która może przyjmować wartość 0 lub 1 w zależności od argumentu x.

Zmienna x może mieć dowolną wartość z obszaru M. Predykat A(x) = 1 gdy x należące do M posiada zadaną własność A. Można powiedzieć, że predykat A(x) wyznacza pewien obszar M_A (zawarty całkowicie w obszarze M) zmiennych przedmiotowych x dla których spełniona jest zależność A(x) = 1.

K w a n t y f i k a t o r e m s z c z e g ó ł o w y m lub e g z y s t e n c j a l n y m (Ex)A(x) nazywamy funkcję dwuwartościową, która jest prawdziwa (=1) jeżeli w obszarze M znajdzie się co najmniej jedna wartość x dla której A(x) = 1.

Kwantyfikator szczegółowy można zapisać w postaci nieskończonej lub skończonej dyzjunkcji

$$(Ex)A(x)=A(x_1) \oplus A(x_2) \oplus A(x_3) \oplus \ldots$$

x_i - wszystkie wartości zmiennej x z przedziału M.

Kwantyfikator szczegółowy wyraża sąd, że istnieje takie x należące do M dla którego A(x) = 1.

K w a n t y f i k a t o r e m o g ó l n y m $(x) A (x)$ nazywamy funkcję dwuwartościową, która jest prawdziwa (=1) gdy dla każdego x z obszaru M spełnione jest równanie $A(x) = 1$.
Kwantyfikator ogólny wyraża sąd, że każde x z obszaru M spełnia warunek $A(x) = 1$.
Kwantyfikator ogólny można zapisać w postaci nieskończonej lub skończonej koniunkcji

$$(x)A(x)=A(x_1)\ A(x_2)\ A(x_3)\ \ldots.$$

x_i – wszystkie wartości zmiennej x z przedziału M.

Wyrażeniami zawężonego rachunku kwantyfikatorów lub inaczej rachunku kwantyfikatorów I rzędu nazywamy wyrażenia w których pod znakiem kwantyfikatora występują symbole zmiennych przedmiotowych, nie występują natomiast nigdy symbole predykatowe.

Rachunek kwantyfikatorów wyższych rzędów dopuszcza występowanie zmiennych przedmiotowych i predykatowych pod znakiem kwantyfikatora.

Przykład wyrażenia rachunku zawężonego

$$(x)\left\{A_1(x) \oplus A_2(y) \,.\, A_3(z)\right\} \qquad (a)$$

w wyrażeniu tym występują zmienne x, y, z oraz symbole predykatowe A_1, A_2, A_3. Kwantyfikator ogólny (x) wiąże zmienną x, pozostałe zmienne y i z są zmiennymi swobodnymi, niezależnymi.
Przykład innego wyrażenia

$$(Ex)\ \left\{A_1(x) \oplus (y)\left[A_2(y).A_3(x) \oplus A_4(z)\right]\right\} \qquad (b)$$

W dalszym ciągu pod słowem "wyrażenię" rozumieć będziemy zawsze wyrażenie rachunku zawężonego.
W wyrażeniu (b) jest jedną zmienna swobodna z oraz dwie zmienne związane x i y. Kwantyfikator szczegółowy (Ex) nazywa się kwantyfikatorem wiodącym drugiego rzędu, ponieważ w obszarze jego działania znajduje się kwantyfikator rzędu pierwszego.

Ogólnie można powiedzieć, że kwantyfikator jest rzędu k jeśli najwyższy rząd kwantyfikatorów znajdujących się w obszarze jego działania jest (k – 1).

Wyrażenia zawierające predykaty i kwantyfikatory podlegają prawom rachunku zdań (algebra Boole'a), oprócz tego słuszne są cztery następujące tożsamości:

a) dla odwrotności

$$(Ex)A(x) = \overline{(x)\overline{A(x)}} \qquad (1.1)$$

$$(x)\,A(x) = \overline{(Ex)\overline{A(x)}}$$

b) dla kwantyfikatora ogólnego i koniunkcji

$$(x)[A_1(x)\ A_2(x)] = (x)A_1(x)\ (x)A_2(x) \qquad (1.2)$$

c) dla kwantyfikatora szczegółowego i dyzjunkcji

$$(Ex)[A_1(x) \oplus A_2(x)] = (Ex)A_1(x) \oplus (Ex)A_2(x) \qquad (1.3)$$

d) dla wyrażenia ze względną stałą

$$[x]\ [W \odot A(x)] = W \odot (x)\ A(x) \qquad (1.4)$$

gdzie:

[x] - oznacza kwantyfikator ogólny lub szczegółowy,
$\odot$ - oznacza dyzjunkcję lub koniunkcję,
W - wyrażenie nie zawierające zmiennej x,
W jest tak zwaną względną stałą.

Dla potrzeb syntezy sieci logicznych wystarcza ograniczyć obszar zmiennych przedmiotowych do zbioru liczb naturalnych, interpretowanych jako dyskretne wartości czasu.

Celem uproszczenia zapisu wprowadza się pojęcie kwantyfikatorów ograniczonych.

Ograniczony kwantyfikator szczegółowy $\underset{\tau<t}{(E\tau)}\, x(\tau)$ oznacza sąd: istnieje takie $\tau < t$, dla którego spełnione jest $x(\tau) = 1$.

Ograniczony kwantyfikator ogólny $\underset{\tau<t}{(\tau)}\, x(\tau)$ oznacza sąd dla każdego $\tau < t$ spełnione jest $x(\tau) = 1$.

Kwantyfikatory ograniczone można zapisać za pomocą kwantyfikatorów nieograniczonych:

$$\underset{\tau<t}{(E\tau)}x(\tau) = (E\tau)\ [\tau<t \cdot x(\tau)]$$

$$\underset{\tau<t}{(\tau)}x(\tau) = (\tau)\,[\tau<t \to x(\tau)] = (\tau)\ [\overline{\tau<t} \oplus x\ (\tau)]$$

Wyrażenie zawierające kwantyfikatory ograniczone podlegają prawom algebry Boole'a oraz tożsamościom (1.1) – (1.4), oprócz tego ważne są następujące twierdzenia:

$$\underset{\tau<1}{(\tau)}\, x(\tau) = 1 \qquad (1.5)$$

Dla kwantyfikatorów dwustronnie ograniczonych

$$\underset{t<\tau<t}{(E\tau)} x(\tau) = 0 \qquad \underset{t<\tau\leqslant t}{(E\tau)} x(\tau) = 0 \qquad (1.6)$$

$$\underset{t<\tau<t}{(\tau)}\, x(\tau) = 1 \qquad \underset{t<\tau\leqslant t}{(\tau)}\, x(\tau) = 1 \qquad (1.7)$$

twierdzenia o eliminacji ograniczeń $\tau \leqslant t$

$$\underset{\tau\leqslant t}{(E\tau)}\, x(\tau) = \underset{\tau<t}{(E\tau)}\, x(\tau) \oplus x(t) \qquad (1.8)$$

$$\underset{\tau\leqslant t}{(\tau)}\, x(\tau) = \underset{\tau<t}{(\tau)}\, x(\tau) \cdot x(t) \qquad (1.9)$$

odpowiednio dla kwantyfikatorów dwustronnie ograniczonych

$$\underset{\varrho\prec\tau\leqslant t}{(E\tau)}\, x(\tau) = \underset{\varrho\prec\tau<t}{(E\tau)}\, x(\tau) \oplus [\varrho\prec t \cdot x(t)] \qquad (1.10)$$

$$\underset{\varrho\prec\tau\leqslant t}{(\tau)}\, x(\tau) = \underset{\varrho\prec\tau<t}{(\tau)}\, x(\tau)\, [\varrho\prec t \rightarrow x(t)] \qquad (1.11)$$

W powyższych równaniach występuje symbol „$\prec$", który oznacza „$\leqslant$" lub „$<$".

3. Synteza sieci logicznych

Interesuje nas synteza sieci logicznych, które są automatami skończonymi. Operatory takich sieci można zawsze zapisać w postaci rozszerzonego wyrażenia kwantyfikatorowego, w którym jedyną zmienną niezależną jest czas t.

Ograniczymy się do przedstawienia syntezy sieci logicznych, których operatory dają się zapisać przy pomocy zawężonego wyrażenia kwantyfikatorowego z jedną zmienną niezależną; wyrażenia te będziemy dalej oznaczać przez A(t), B(t), W(t), Z(t) i krótko nazywać wyrażeniami kwantyfikatorowymi. Wyrażenie kwantyfikatorowe nazywamy regularnym o ile jego wartość w takcie t nie zależy od

predykatów w momencie t i następnych $t+1$, $t+2$, $t+3$ itd. Inaczej wyrażenie jest regularne o ile nie zawiera kwantyfikatorów ograniczonych typu $\underset{\alpha \prec \omega \leq t}{[\omega]}$ oraz predykatów typu $x_1(t)$, $x_2(t)$, ... $x_i(t)$, ... zależnych od taktu t.
Przykład wyrażenia regularnego

$$\tilde{\omega}(t) = \underset{\tau<t}{(\tau)} \left[x_1(\tau) \oplus \underset{\tau<\varrho<t}{(E\varrho)} x_2(\varrho)\right]$$

Regularność będziemy zaznaczać przez umieszczenie znaku $\sim$ nad symbolem wyrażenia.
Przykład wyrażenia nieregularnego

$$A(t) = \underset{\tau<t}{(\tau)} \left\{x_1(\tau) \oplus \underset{\tau<\varrho<t}{(E\varrho)}\left[x_2(\varrho) \oplus x_2(t)\right]\right\} \cdot x_1(t)$$

W wyrażeniu $A(t)$ są dwie nieregularności $x_2(t)$ i $x_1(t)$.
Inne wyrażenie nieregularne

$$B(t) = \underset{\tau<t}{(E\tau)} \left[x_1(\tau) \oplus \underset{\tau<\varrho\leq t}{(\varrho)} x_2(\varrho)\right].$$

Nieregularność powyższego wyrażenia polega na istnieniu ograniczenia $\varrho \leq t$ dla kwantyfikatora ogólnego (ϱ).
Można udowodnić prawdziwość następującego twierdzenia o ograniczeniu kwantyfikatora wiodącego[x].

Jeżeli dowolny kwantyfikator wiodący danego wyrażenia jest ograniczony typu $\underset{\lambda\prec\tau<t}{[\tau]}$ lub typu $\underset{\lambda\prec\tau\leq t}{[\tau]}$ to zdanie $\tau<t$ lub odpowiednio $\tau\leq t$ występujące w wyrażeniu podkwantyfikatorowym jest prawdziwe i można je zamienić na 1.

Oczywiście każdą jedynkę można zamienić na $\tau<t$ lub odpowiednio $\tau\leq t$.

Wniosek

Jeśli kwantyfikator wiodący jest typu $\underset{\varrho\prec\tau<t}{[\tau]}$ to dla wyrażeń będących w obszarze jego działania słuszne są tożsamości:

$$\underset{t\prec\omega\prec\tau}{(E\omega)} B(t) = 0 \tag{2.1}$$

---

[x] Dowód można znaleźć w [1] str.255-256.

$$\underset{t<\omega<\tau}{(\omega)} B(t) = 0. \qquad (2.2)$$

Dla przykładu udowodnimy pierwszą nierówność. Na podstawie twierdzenia I można napisać:

$$\underset{t<\omega<\tau}{(E\omega)} B(t) = \underset{t<\omega<\tau}{(E\omega)} B(t)\ 1 = \underset{t<\omega<\tau}{(E\omega)} B(t)\ \tau<t$$

Zamieniając kwantyfikator ograniczony na nieograniczony, otrzymujemy

$$\underset{t<\omega<\tau}{(E\omega)} B(t)\cdot\tau<t=(E\omega)[t<\omega\cdot\omega<\tau\cdot B(t)]\cdot\tau<t$$

Ostatnia forma zapisu dowodzi prawdziwości (2.1), ponieważ prawa strona równa jest zero.

Przekształcenie operatora danego w postaci wyrażenia kwantyfikatorowego w równania kanoniczne wymaga przeprowadzenia działań dwu typów:

I. Eliminacja kwantyfikatorów typu $\underset{\varrho<\omega\leqslant t}{[\omega]}$, $\underset{t<\omega<t}{[\omega]}$ i $\underset{t<\omega<\tau}{[\omega]}$

II. Rozkład wyrażenia na podformuły regularne.

Oba typy działań zilustrujemy na przykładzie syntezy prostej sieci logicznej realizującej element pamięci z kasowaniem.

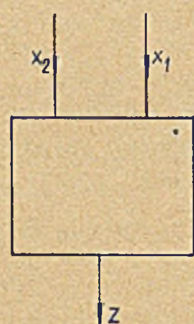
Przykład 1

1 bitowa komórka pamięci z kasowaniem jest automatem skończonym posiadającym dwa wejścia: x_1 - sygnał zapisywany, x_2 - sygnał kasujący i jedno wyjście z (rys.1). Wyjście z równe jest 1 w takcie t o ile w dowolnym takcie $\tau\leqslant t$ pojawił się sygnał x_1 i w żadnym takcie między τ i t nie było sygnału x_2. Zakłada się że sygnał kasujący nie może pojawić się jednocześnie z sygnałem zapisywanym. Opisany operator może być przedstawiony w postaci wyrażenia kwantyfikatorowego:

$$z(t) = \underset{\tau\leqslant t}{(E\tau)} \left[x_1(\tau)\cdot \underset{\tau<\varrho\leqslant t}{(\varrho)} \overline{x_2(\varrho)}\right] \qquad (2.3)$$

Wyrażenie (2.3) jest nieregularne, przeprowadzamy zatem eliminację kwantyfikatorów. Działanie to rozpoczynamy od kwantyfikatorów najwyższego rzędu.
W naszym przykładzie takim kwantyfikatorem jest $(E\tau)_{\tau\leq t}$ (drugi rząd).
Na podstawie (1.8) można napisać

$$z(t)= (E\tau)_{\tau<t} \left[x_1(\tau)\cdot(\varrho)_{\tau<\varrho\leq t} \overline{x_2(\varrho)}\right] \oplus \left[x_1(t)\cdot(\varrho)_{t<\varrho\leq t} \bar{x}_2(\varrho)\right] \qquad (2.4)$$



Rys. 1. Binarna komórka pamięci

Korzystając z (1.7) eliminujemy podformułę $(\varrho)_{t<\varrho\leq t} \overline{x_2(\varrho)}$ jako równą 1 i otrzymujemy:

$$z(t)= (E\tau)_{\tau<t} \left[x_1(\tau)\cdot(\varrho)_{\tau<\varrho\leq t} \overline{x_2(\varrho)}\right] \oplus x_1(t) \qquad (2.5)$$

Następnie eliminujemy kwantyfikator 1 rzędu $(\varrho)_{\tau<\varrho\leq t}$ korzystając z równania (1.11):

$$z(t)=(E\tau)_{\tau<t} \left\{x_1(\tau)\cdot(\varrho)_{\tau<\varrho<t} \bar{x}_2(\varrho) \left[\tau<t \rightarrow \bar{x}_2(t)\right]\right\} \oplus x_1(t) \qquad (2.6)$$

Implikację $\tau<t \rightarrow \bar{x}_2(t)$ można zapisać w postaci dyzjunkcji $\overline{\tau<t} \oplus \bar{x}_2(t)$. Na podstawie tw. o ograniczeniu kwantyfikatora wiodącego wiemy, że $\overline{\tau<t} = 0$, wobec tego (2.6) sprowadza się do wyrażenia:

$$z(t)=(E\tau)_{\tau<t} \left[x_1(\tau)\cdot(\varrho)_{\tau<\varrho<t} \bar{x}_2(\varrho)\cdot \bar{x}_2(t)\right] \oplus x_1(t) \qquad (2.7)$$

Na tym kończy się proces eliminacji kwantyfikatorów.

Rozkład wyrażenia na podformuły regularne zawierające nieregularności (tylko nieregularności typu $x_i(t)$) polega na przekształceniach, które dają w efekcie funkcję logiczną podformuł regularnych i nieregularności.

Stosując tożsamość (1.4) do wyrażenia (2.7) otrzymujemy

$$z(t) = \underset{\tau<t}{(E\tau)}[x_1(\tau) \underset{\tau<\varrho<t}{(\varrho)} \bar{x}_2(\varrho)] \cdot \bar{x}_2(t) \oplus x_1(t) \qquad (2.8)$$

W wyrażeniu (2.8) jest jedna podformuła regularna

$$\underset{\tau<t}{(E\tau)}[x_1(\tau) \underset{\tau<\varrho<t}{(\varrho)} \bar{x}_2(\varrho)] = \tilde{B}(t) \qquad (2.9)$$

Interesujący nas rozkład na podformuły regularne kończy się podstawieniem (2.9) do (2.8)

$$z(t) = \tilde{B}(t) \cdot \bar{x}_2(t) \oplus x_1(t) \qquad (2.10)$$

Wyrażenia zawierające nieregularności typu $x_i(t)$ można rozłożyć na podformuły regularne dwoma sposobami:

a) przez stosowanie przekształceń tożsamościowych,
b) przez rozkład wyrażenia wg zmiennych $x_i(t)$.

Pokażemy na czym polega rozkład nieregularności wg zmiennych $x_i(t)$.

Dla prostoty załóżmy, że w formule $W(t)$ jest jedna nieregularność $x(t)$, zaznaczymy to pisząc nie $W(t)$ lecz $W[x(t),t]$.
Niech $W(0,t)$ będzie wyrażeniem otrzymanym z $W[x(t),t]$ przez podstawienie $x(t)=0$, analogicznie przez podstawienie $x(t)=1$ otrzymujemy $W(1,t)$.

Następnie korzysta się z tożsamości:

$$W[x(t),t] = x(t)\, W(1,t) \oplus \bar{x}(t) W(0,t)$$

Wróćmy do naszego przykładu, w którym dokonano już rozkładu pierwszym sposobem. Zamieńmy w (2.9) kwantyfikatory typu $\underset{\varrho<\omega<t}{[\omega]}$ na $\underset{\varrho<\omega\leq t}{[\omega]}$, otrzymamy wyrażenie nieregularne

$$B(t+1) = \underset{\tau\leq t}{(E\tau)}[x_1(\tau) \underset{\tau<\varrho\leq t}{(\varrho)} \bar{x}_2(\varrho)] \qquad (2.11)$$

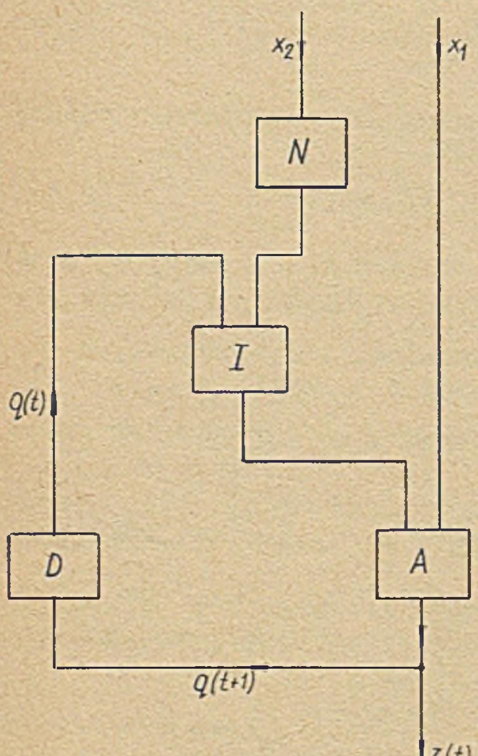
Porównując (2.11) i (2.3) spostrzegamy, że

$$B(t+1) = z(t) \qquad (2.12)$$

Jeżeli zastąpimy symbol wyrażenia kwantyfikatorowego B(t) symbolem predykatu q(t) to na podstawie (2.12) i (2.10) otrzymamy równania kanoniczne operatora:

$$z(t) = q(t+1) \qquad (2.13)$$

$$q(t+1) = q(t)\ \bar{x}_2(t) \oplus x_1(t)$$



Rys. 2. Sieć logiczna jednobitowej komórki pamięci

Zakładając, że dysponujemy podstawowymi elementami realizującymi koniunkcję, dyzjunkcję i negację oraz elementem opóźniającym sygnały o 1 takt możemy w oparciu o równanie kanoniczne (2.13) zbudować projektowaną sieć (rys. 2).

Jak łatwo sprawdzić uzyskana sieć spełnia wymagania stawiane komórce pamięci.

Pokażemy obecnie ogólnie jak przekształca się kwantyfikatorowy zapis dowolnego operatora (automatu skończonego) w równania kanoniczne tego operatora.

Aby wyrażenie kwantyfikatorowe W(t) przekształcić w równania kanoniczne postępujemy zgodnie z podanym niżej schematem.

1. Jeżeli w wyrażeniu W(t) są kwantyfikatory typu $[\omega]_{\varrho<\omega\leq t}$ to musimy je wyeliminować - stosując działanie I, następnie tak przekształcone wyrażenie rozkładamy na podformuły regularne - działanie II.

W rezultacie otrzymujemy:

$$W(t)=\phi[x_1(t), \ldots\ldots, x_n(t), \tilde{B}_1(t), \ldots\ldots, \tilde{B}_k(t)] \qquad (2.14)$$

gdzie

$\tilde{B}_i(t)$ - podformuły regularne,

ϕ - funkcja zdaniowa (funkcja logiczna).

Jeżeli $W(t)$ jest wyrażeniem regularnym lub dane jest w postaci (2.14) to działania rozpoczynamy od punktu 2.

2. W każdej z podformuł regularnych $\tilde{B}_i(t)$ zamieniamy $[\omega]_{\varrho<\omega<t}$ na $[\omega]_{\varrho<\omega\leq t}$ otrzymujemy $B_i(t+1)$. W uzyskanym wyrażeniu wykonujemy działania I i II, w rezultacie dostajemy:

$$B_i(t+1)=\Psi_i[x_1(t), \ldots\ldots, x_n(t), \tilde{D}_1(t), \ldots\ldots, D_s(t)]$$

3. O ile wśród otrzymanych podformuł $\tilde{D}_i(t)$ są różne od $\tilde{B}_i(t)$ to każdą z nich przekształcamy za pomocą działań opisanych w punkcie 2 i otrzymujemy

$$D_i(t+1)=\mathcal{X}_i[x_1(t), \ldots\ldots, x_n(t), \tilde{E}_1(t), \ldots\ldots, \tilde{E}_l(t)]$$

4. Wszystkie podformuły $\tilde{E}_i(t)$ różne od $\tilde{B}_i(t)$ lub $\tilde{D}_i(t)$ przekształcamy wg 2 co daje:

$$E_i(t+1)=\Omega_i[x_1(t), \ldots\ldots, x_n(t), \tilde{G}_1(t), \ldots\ldots, \tilde{G}_m(t)]$$

Proces ciągnie się tak długo jak długo otrzymuje się podformuły różne od uzyskanych w poprzednich fazach.

Trachtenbrot i Kobryński podali w [1] dowód, że dla każdej formuły opisującej operator zdeterminowany i ograniczony proces ten kończy się po wykonaniu skończonej ilości kroków i daje skończony system równań:

$$W(t)=\phi[x_1(t), \ldots\ldots, x_n(t), \tilde{H}_1(t), \ldots\ldots, \tilde{H}_r(t)]$$

$$H_1(t)6=\Psi_1[x_1(t), \ldots\ldots, x_n(t), \tilde{H}_1(t), \ldots\ldots, \tilde{H}_r(t)]$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.15)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$H_r(t+1)=\Psi_r[x_1(t), \ldots\ldots, x_n(t), \tilde{H}_1(t), \ldots\ldots, \tilde{H}_r(t)]$$

Kładąc w równaniu (2.15) symbole predykatowe $q_i(t)$ zamiast symboli wyrażeń regularnych $\tilde{H}_i(t)$ otrzymujemy równania kanoniczne operatora

$$
\begin{aligned}
&z(t) = \phi[x_1(t),\dots,x_n(t),\ q_1(t),\dots,q_r(t)] \\
&q_1(t+1) = \Psi_1[x_1(t),\dots,x_n(t),\ q_1(t),\dots,q_r(t)] \\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
&q_r(t+1) = \Psi_r[x_1(t),\dots,x_n(t),\ q_1(t),\dots,q_r(t)]
\end{aligned} \tag{2.16}
$$

Warunki początkowe oblicza się z wyrażeń $\tilde{H}_i(t)$.

Przedstawiona metoda pozwala zaprojektować sieć logiczną, do której informacja jest dostarczona przez n - kanałów wejściowych a wynik jej przetwarzania jest wysyłany przez jeden kanał wyjściowy.

Operator sieci logicznej n - wejściowej i k - wyjściowej można zapisać w postaci układu wyrażeń kwantyfikatorowych:

$$
\begin{aligned}
&z_1(t) = W_1(x_1,\ x_2,\dots,x_n,t) \\
&z_2(t) = W_2(x_1,\ x_2,\dots,x_n,t) \\
&\dots\dots\dots\dots\dots\dots \\
&\dots\dots\dots\dots\dots\dots \\
&z_k(t) = W_k(x_1,\ x_2,\dots,x_n,t)
\end{aligned} \tag{2.17}
$$

gdzie

$x_1, x_2,\dots, x_n$ - sygnały przesyłane kanałami wejściowymi 1, 2, ... n,

$z_1, z_2,\dots, z_n$ - sygnały przesyłane kanałami wyjściowymi 1, 2, ... k.

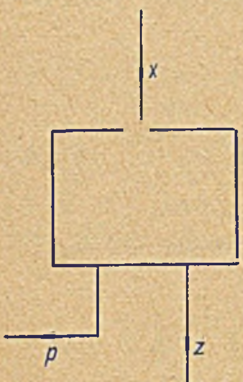
Aby otrzymać równania kanoniczne operatora (2.17) należy każde z wyrażeń $z_1(t), z_2(t),\dots,z_k(t)$ przekształcić oddzielnie na równania kanoniczne.

Otrzymuje się w ten sposób układ równań kanonicznych sieci n-wejściowej i k-wyjściowej.

Na zakończenie podamy jeszcze jeden przykład syntezy sieci logicznej.

Przykład 2

Zaprojektujemy sieć logiczną licznika dwójkowego. Zajmiemy się elementarną komórką licznika (rys. 3).



Rys. 3. Podstawowa komórka licznika

Pełny licznik otrzymać można przez szeregowe połączenie identycznych komórek elementarnych, z których każda jest siecią logiczną jednowejściową i dwuwyjściową. Sygnał wejściowy oznaczmy symbolem x, a sygnały wyjściowe: z - stan komórki i p - przeniesienie do wyższej pozycji.

Sygnał wyjściowy określający stan z komórki w takcie t równa się 1, o ile w dowolnym takcie $\tau < t$ pojawił się sygnał x i jeżeli w każdym innym takcie $\eta \leq t$ nie było sygnału x, można to zapisać w postaci wyrażenia kwantyfikatorowego:

$$z(t)=(E\tau)_{\tau<t}[x(\tau).(\varrho)_{\tau<\varrho<t}\bar{x}(\varrho).(\lambda)_{\lambda<\tau}\bar{x}(\lambda)] \quad ..... \quad (3.1)$$

Sygnał przeniesienia p w takcie t jest równy 1 o ile w dwu dowolnych taktach poprzedzających $\tau < t$ i $\lambda < t$ pojawił się sygnał wejściowy i jeżeli we wszystkich pozostałych taktach $\eta \leq t$ był spełniony warunek

$$x(\eta) = 0$$

Zapis kwantyfikatorowy

$$p(t)=(E\tau)_{\tau<t}\left\{x(\tau).(\varrho)_{\varrho<\tau}\bar{x}(\varrho).(E\lambda)_{\tau<\lambda\leq t}[x(\lambda).(\eta)_{\tau<\eta<\lambda}\bar{x}(\eta).(\xi)_{\lambda<\xi\leq t}\bar{x}(\xi)]\right. \quad (3.2)$$

Wyrażenia (3.1) i (3.2) opisują operator komórki licznika. Odpowiadają one układowi równań (2.17).
Przekształćmy wyrażenie (3.1), w tym celu należy wyeliminować kwantyfikator nadrzędny $(E\tau)_{\tau<t}$ a następnie kwan-

tyfikator$\underset{\tau<\varrho\leqslant t}{(\varrho)}$ rzędu 1, po prostych przekształceniach uzyskujemy:

$$z(t)=\underset{\tau<t}{(E\tau)}\left\{x(\tau).\underset{\lambda<\tau}{(\lambda)}\bar{x}(\lambda).\underset{\tau<\varrho<t}{(\varrho)}\bar{x}(\varrho).[\tau<t\rightarrow\bar{x}(t)]\right\}\oplus$$

$$\oplus\, x(t).\underset{\lambda<t}{(\lambda)}\,\bar{x}(\lambda) \tag{3.3}$$

Implikacja $[\tau<t\rightarrow\bar{x}(t)]$ jest równa $\bar{x}(t)$ (na podstawie twierdzenia I), zatem

$$z(t)=\underset{\tau<t}{(E\tau)}\left\{x(\tau).\underset{\lambda<t}{(\lambda)}\bar{x}(\lambda).\underset{\tau<\varrho<t}{(\varrho)}\bar{x}(\varrho).\bar{x}(t)\right\}\oplus x(t)\,\underset{\lambda<t}{(\lambda)}\,\bar{x}\,(\lambda) \tag{3.4}$$

W wyrażeniu (3.4) należy usunąć nieregularności $\bar{x}(t)$ z obszaru działania kwantyfikatora $\underset{\tau<t}{(E\tau)}$ wykorzystujemy w tym celu tożsamość (1.4) i uzyskujemy:

$$z(t)=\underset{\tau<t}{(E\tau)}\left\{x(\tau).\underset{\lambda<\tau}{(\lambda)}\bar{x}(\lambda).\underset{\tau<\varrho<t}{(\varrho)}\bar{x}(\varrho)\right\}\bar{x}(t)\oplus x(t)\,\underset{\lambda<t}{(\lambda)}\,\bar{x}\,(\lambda) \tag{3.5}$$

W wyrażeniu (3.5) są dwie podformuły regularne:

$$\underset{\tau<t}{(E\tau)}\left\{x(\tau)\,.\,\underset{\lambda<\tau}{(\lambda)}\,\bar{x}(\lambda).\underset{\tau<\varrho<t}{(\varrho)}\,\bar{x}(\varrho)\right\}=\tilde{B}_1(t)$$

oraz

$$\underset{\lambda<t}{(\lambda)}\,\bar{x}(\lambda)=\tilde{B}_2(t)$$

Na podstawie tożsamości (1.1) widocznym jest, że

$$\tilde{B}_2(t)=\overline{\tilde{B}_1(t)}$$

Podstawiając symbole podformuł do (3.5) otrzymamy:

$$z(t)=\tilde{B}_1(t)\,\bar{x}(t)\oplus\overline{\tilde{B}_1(t)}\,x\,(t) \tag{3.6}$$

Tworząc wyrażenie $B_1(t+1)$ i porównując je z (3.1) z łatwością spostrzegamy, że

$$B_1(t+1)=z(t) \tag{3.7}$$

Aby otrzymać równania kanoniczne w klasycznej formie podstawiamy symbol predykatowy $q_1(t)$ zamiast $\tilde{B}_1(t)$ i otrzymujemy:

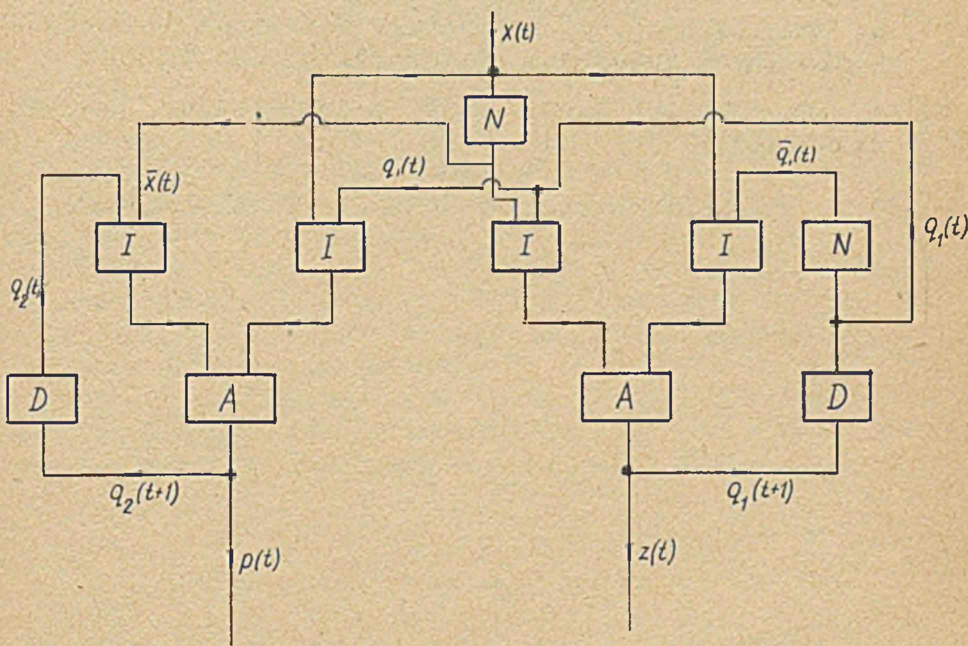
$$z(t)=q_1(t+1)$$

$$q_1(t+1)=q_1(t)\ \overline{x(t)} \oplus \overline{q_1(t)}x(t) \tag{3.8}$$

Postępując podobnie przekształcamy (3.2) w równania kanoniczne sygnału przeniesienia:

$$p(t)=q_2(t+1) \tag{3.9}$$

$$q_2(t+1)=q_2(t)\ \overline{x(t)} \oplus q_1(t)x(t)$$



Rys. 4. Sieć logiczna licznika binarnego

Jeżeli przyjmiemy, że dysponujemy elementami realizującymi podstawowe operacje logiczne oraz elementem opóźniającym sygnały o 1 takt, możemy na podstawie układów równań (3.8) i (3.9) zbudować sieć logiczną jednobitowej komórki licznika (rys.4).

Bez trudności stwierdzamy, że przedstawiona na rys. 4 sieć sumuje sygnały przychodzące na wejście. Sygnał stanu w pewnym takcie t jest równy 1, o ile ilość sygnałów jedynkowych pojawiających się na wejściu w poprzednich taktach była nieparzysta. Sygnał przeniesienia p(t)=1 jeżeli ilość sygnałów jedynkowych w poprzednich taktach była parzysta.

Pragnę złożyć serdeczne podziękowanie Panu Profesorowi Doktorowi Stefanowi Węgrzynowi za pomoc w opracowywaniu tematu oraz wiele cennych uwag w czasie pisania pracy.

LITERATURA

[1] B. Trachtenbrot, N. Kobrynski: Wwiedienie w tieoriju koniecznych awtomatow. Fizmatgiz Moskwa 1962.

[2] A. Grzegorczyk: Zarys logiki matematycznej, PWN Warszawa 1961.

Rękopis złożono w Redakcji w dniu 8.7.1963 r.

ПРИМЕНЕНИЕ ИСЧИСЛЕНИЯ ПРЕДИКАТОВ ДЛЯ СИНТЕЗА ЛОГИЧЕСКИХ СХЕМ

Р е з ю м е

В статье приводится возможность применения узкого исчисления предикатов для расчета логичес ких схем, в которые включены запоминающие элемен ты. Даны основы узкого исчисления предикатов. Да лее представлено некоторые вопросы синтеза логических схем, которых операторы можно написать с помощью выражений узкого исчисления предикаторов. Теорию проиллюстрировано простыми примерами синтеза логических схем. Представлен синтез элементарной ячейки памяти с установкой нуля и элемен тарной ячейки двоичного счетчика.

A QUANTIFIKATOR CALCULUS

S u m m a r y

The article shows how to use the limited quantificator calculus method for the design of logical networks containing memory elements. Essential properties of the limited quantificator calculus are given.

The author discusses the problem of designing logical networks in such a case where their operators can be performed by means of limited quantificator expressions, having one independent variable.

Two simple examlpes: the synthesis of 1-bit memory element with reset possibility and the synthesis of a binary counter element illustrate the described method.