


2 1971

prace

P. 2229 | 71



**Instytutu
Maszyn
Matematycznych**

rok XIII

P. 2229/71



prace

Instytutu

Maszyn

Matematycznych

Copyright © 1971 - by Instytut Maszyn Matematycznych
Poland

Wszelkie prawa zastrzeżone

Komitet Redakcyjny

Bartłomiej GŁOWACKI, Jerzy GRADOWSKI /redaktor naczelny/,
Andrzej KOJEMSKI, Włodzimierz MARDAL /z-ca redaktora naczelnego/,
Jan RELUGA

Sekretarz Redakcji: Romana NITKOWSKA

Adres Redakcji: Instytut Maszyn Matematycznych
Branżowy Ośrodek INTE
Warszawa, ul. Krzywickiego 34, tel. 28-37-29

SPIS TREŚCI ZESZYTU 2

СОДЕРЖАНИЕ
CONTENTS

str.

1. MADEJ W., MARDAL W., MUŁAWA A.

SEMMA 2 - język do symulacji procesów ciągłych z automa-
tyczną optymalizacją parametrów modelu 5

SEMMA 2 - язык для моделирования непрерывных про-
цессов с автоматической оптимизацией парамет-
ров модели /Резюме/ 26

SEMMA 2 language for simulation of continuous processes
with model parameter automatic optimalization /Summary/ 27

2. SINKIEWICZ T.

Analiza efektywności modelowania sieci przełączających
metodą listową i blokową 29

Анализ эффективности моделирования переключающих
сетей листовым и блочным методом /Резюме/ . . . 70

The analysis of the effectivity of switching networks
modelling by the list and the block method /Summary/. . 71

3. KUBERA W.

Zastosowanie EMC w pewnych zagadnieniach projektowania
okablowania systemów cyfrowych III generacji 73

Применение ЦВМ к некоторым проблемам проектиро-
вания линий соединения в ЭВМ III генерации/Резюме/ 94

Digital computer application to some problems of design-
ing III generation digital system wiring /Summary/. . . 96

4. AUTORZY ARTYKUŁÓW ZAMIESZCZONYCH W ZESZYCIE 2/71 "PRAC IMM". . 97

CEMMA 2 - JEZYK DO SYMULACJI PROCESÓW CIĄGŁYCH Z AUTOMATYCZNĄ OPTYMALIZACJĄ PARAMETRÓW MODELU^{*)}

Waldemar MADEJ
Włodzimierz MARDAL
Andrzej MULAŁA

Pracę złożono 2.04.1971

W artykule opisano charakterystyki języka CEMMA 2 przeznaczonego do symulacji procesów ciągłych i impulsowo-ciągłych na maszynie cyfrowej. Język ten zrealizowany na maszynie ZAM-41 należy do kategorii cyfrowych symulatorów maszyny analogowej i jest oparty na zasadzie blokowego opisu modelowanego procesu. Artykuł podaje budowę i znaczenie poszczególnych zdań języka. Opisano sposób rozwiązywania zadań optymalizacji, zasady definiowania zdań złożonych, a także postacie danych uzyskiwanych w procesie symulacji. W Dodatku do artykułu podano przykład zastosowania systemu CEMMA 2.

1. WSTĘP

Język CEMMA 2 stanowi rozbudowaną i udoskonaloną wersję języka CEMMA [6]. Rozszerzenie języka zostało dokonane w następujących głównych kierunkach:

- rozbudowa repertuaru zdań,
- wprowadzenie możliwości definiowania przez programującego własnych zdań złożonych /tzw. makrobloków/.

^{*)} Dokładny opis języka CEMMA 2 jest dostępny we wszystkich ośrodkach obliczeniowych wyposażonych w maszynę ZAM-41.

- stworzenie możliwości automatycznej optymalizacji parametrów modelu według zadanego kryterium i ograniczeń parametrów,
- wprowadzenie szeregu dyrektyw umożliwiających wpływanie na proces translacji i wykonanie programu.

Rozszerzenia te zostały podyktowane praktycznymi potrzebami użytkowników języka CEMMA, a także aktualnymi tendencjami rozwojowymi języków tego typu na świecie.

W następnych punktach artykułu zostaną omówione najważniejsze zagadnienia, które dotyczą nowej wersji języka CEMMA przy założeniu, że czytelnik zapoznał się z opisem poprzedniej wersji języka [6].

2. ZDANIA PODSTAWOWE

Zbiór zdań podstawowych służących do opisu zależności między zmiennymi modelu matematycznego procesu został rozszerzony i obejmuje obecnie 39 zdań, przy czym zasady budowy zdań strukturalnych pozostały niezmienione.

Kompletną listę tych zdań i ich znaczeń podano w Dodatku A. Rozszerzenie listy zdań polegało głównie na dołączeniu zdań:

- opisujących elementy logiczne i cyfrowe,
- opisujących wyznaczanie i działania na maksymalnych i minimalnych wartościach zmiennych,
- dodatkowych funkcji trygonometrycznych.

Istotną niedogodnością pierwszej wersji języka była konieczność takiego ułożenia schematu blokowego modelu i programu, aby nie wystąpiła tzw. pętla algebraiczna, tj. taka zamknięta pętla połączeń bloków, w której nie występuje operacja całkowania lub opóźnienia. W nowej wersji niedogodność ta została usunięta przez wprowadzenie specjalnego zdania PAL, które opisuje występującą w modelu pętlę algebraiczną i określa dopuszczalny

błąd przy iteracyjnym wyznaczaniu wartości zmiennej wyjściowej.

3. ZDANIA ZŁOŻONE

W celu ułatwienia opisu modeli, w których pewne fragmenty struktury powtarzają się wielokrotnie wprowadzono możliwość definiowania zdań złożonych jako sekwencji zdań podstawowych. Po takim zdefiniowaniu zdania złożonego może ono być używane w różnych miejscach programu przy różnych parametrach i wielkościach zmiennych. Definicja zdania złożonego ma następującą postać:

MAKRO	XYZ	Np	Mq		
-	-	}			
-	-				
-	-				ciąg zdań
-	-				podstawowych
-	-				

KONMAKRO;

gdzie MAKRO i KONMAKRO są dyrektywami początku i końca definicji, XYZ-mnemotechniczna nazwa definiowanego zdania, p określa liczbę zmiennych wejściowych zdania złożonego, q oznacza liczbę parametrów funkcji opisanej przez zdanie złożone. Np. jeśli definicja zdania złożonego miała postać

MAKRO	REG	N2	M3	
-	-	-	-	
-	-	-	-	
-	-	-	-	
-	-	-	-	

KONMAKRO

a zdefiniowane zdanie złożone zostało użyte w programie w następujący sposób

105 REG 10 15 --3 --1 P31

to symbolom N1, N2 i M1, M2, M3, występującym w zdaniach podstawowych definicji, zostaną przyporządkowane następujące elementy użyte w zdaniu złożonym:

N1 = 10

N2 = 15

M1 = -3

M2 = -1

M3 = P31 /parametr symboliczny/

Obowiązuje przy tym zasada, że definicja zdania złożonego musi poprzedzać pierwsze jego użycie.

Wprowadzenie możliwości definiowania zdań złożonych w języku GEMMA 2 pozwala na używanie w opisie procesu pojęć bliższych tym, które stosuje się w różnych dziedzinach badań zajmujących się procesami ciągłymi.

4. ROZWIĄZYWANIE ZADAŃ OPTYMALIZACJI

Języki symulacyjne służą głównie do uzyskiwania danych o właściwościach dynamicznych procesu przy określonych jego parametrach. Rozwiązywanie zadań doboru pewnych parametrów tak, aby uzyskać ekstremum określonej charakterystyki zachowania się procesu, wymaga wielokrotnej symulacji przy różnych wartościach parametrów. Wydaje się zatem celowe, aby to zadanie również powierzyć maszynie cyfrowej, która przez wielokrotne powtórzenie cyklu symulacji i badanie wartości określonej funkcji celu wyznaczałaby optymalne wartości parametrów przy zadanych ograniczeniach. W języku GEMMA 2 podjęto próbę rozwiązania tego problemu przez wprowadzenie dyrektywy OPTYMALIZUJ, o następującej budowie:

OPTYMALIZUJ <metoda><forma wyników>, N, P_i, g_i, h_i, P_j, g_j, h_j, ...

gdzie <metoda> - określa jedną z trzech metod optymalizacji, <forma wyników> - określa sposób rejestracji wyników, N - numer zdania /bloku/, którego wielkość wyjściowa winna być zminimalizowana, P_i, g_i, h_i - symboliczna nazwa optymalizowanego pa-

rametru oraz jego ograniczenie dolne $/g_1/$ i górne $/h_1/$. Należy zwrócić uwagę, że kryterium jakości układu, które stanowi podstawę procesu optymalizacji musi być, podobnie jak sam proces, opisane za pomocą zdań języka GEMMA 2.

W obecnym rozwiązaniu system GEMMA 2 pozwala optymalizować do sześciu parametrów równocześnie.

Należy tu podkreślić, że przy opracowywaniu koncepcji rozwiązywania problemów optymalizacyjnych w ramach programu GEMMA 2 konieczne było uwzględnienie charakterystyk sprzętu jakim dysponowali projektanci, tzn. maszyny ZAM-41. Stosunkowo mała pojemność pamięci operacyjnej maszyny zmuszała do wyboru takiego zakresu problemów i takich metod ich rozwiązywania, które pozwoliłyby na uzyskanie możliwie małych rozmiarów programu GEMMA 2. Dlatego też w obecnym rozwiązaniu języka ograniczono się do problemów optymalizacji przy ograniczeniach w postaci stałych wartości liczbowych, co uniemożliwia rozwiązanie wielu zagadnień praktycznych, w których występują ograniczenia funkcyjne. Podobnie, wybór metody optymalizacji został podyktowany dążeniem do uzyskania możliwie prostych programów, co spowodowało odrzucenie metod nawet bardziej efektywnych, wymagających mniejszej liczby obliczeń lecz bardziej skomplikowanych w sensie ich zaprogramowania.

W programie GEMMA 2, w przypadku minimalizacji funkcji jednej zmiennej stosuje się, niezależnie od zadeklarowanej metody, metodę złotego podziału [2]. Poszukiwanie minimum wartości kryterium przerywa się, gdy zachodzi warunek:

$$\left| \frac{f(p') - f(p'')}{p' - p''} \right| < e$$

gdzie p' , p'' - kolejne wartości zmiennej, $f(p')$, $f(p'')$ - wartości funkcji minimalizowanej, e - dokładność wyznaczania minimum, ustalana automatycznie na początku procesu optymalizacji.

Dla znalezienia minimum funkcji 2 ÷ 6 zmiennych z ograniczeniami zastosowanie metod Rosenbrocka i Powella-Zangwilla wymaga dokonania przekształceń według algorytmu Box'a [4], które powo-

dują, że problem optymalizacji z ograniczeniami zamienia się w problem bez ograniczeń. Następnie stosuje się metodę Rosenbrocka w wersji opisanej w [1] lub Powella-Zangwilla [3], [5], przy czym tę ostatnią stosuje się w dwóch wersjach, nazwanych POWELL i ZANGWILL. Obydwie metody wzajemnie się uzupełniają, gdyż pierwsza z nich wykazuje dobrą zbieżność z dala od poszukiwanego minimum, a druga jest szybko zbieżna w jego pobliżu.

Realizacja dwóch wersji algorytmu Powella - Zangwilla została podyktowana koniecznością zapewnienia dobrej zbieżności procesu optymalizacji zarówno dla szybko jak i wolno zmiennych funkcji celu, w zależności od zmian wartości parametrów optymalizowanych. W tym celu zastosowano różne procedury minimalizacji funkcji wzdłuż zadanego kierunku i różne kryteria zbieżności. W pierwszej wersji wykorzystuje się kryterium dokładne podane w [3] oraz uproszczoną procedurę minimalizacji funkcji wzdłuż zadanego kierunku. Polega ona na tym, że po wyznaczeniu dwóch wartości minimalizowanej funkcji $f_0 = f(p)$ i $f_1 = f(p + l \cdot q)$, gdzie p - punkt startowy, q - kierunek minimalizacji, l - wielkość kroku, dokonuje się porównania f_1 i f_0 . Jeśli $f_1 < f_0$ to przyjmuje się następny krok - $3l$ dla $f_1 > f_0$ natomiast krok zostaje zmniejszony dwukrotnie, a kierunek zmieniony na przeciwny.

W drugiej wersji metody Powella-Zangwilla procedura poszukiwania minimum jest nieco inna, mianowicie zapamiętuje się trzy najmniejsze wartości funkcji uzyskane w poprzednich krokach i jeśli

$$f(l_1) < f(l_2) < f(l_3) \quad \text{dla} \quad l_1 < l_2 < l_3$$

to minimum funkcji jednomodalnej znajduje się w przedziale $[l_1, l_3]$; można wtedy zastosować algorytm złotego podziału.

W tej wersji metody stosuje się kryterium identyczne jak w metodzie Rosenbrocka, mianowicie sprawdza się spełnienie warunku

$$\| \underline{a} \| < \epsilon$$

gdzie $\underline{a}^T = (a_1, a_2, \dots, a_n)$ jest wektorem globalnego przesunięcia w przestrzeni parametrów.

Wykorzystanie automatycznej optymalizacji w języku CEMMA 2 okazało się szczególnie użyteczne dla złożonych procesów. Np. wyznaczenie optymalnych parametrów układów regulacji kaskadowej kotła parowego, opisanego 62 równaniami różniczkowymi zwyczajnymi I rzędu metodą automatyczną, przyniosło pięciokrotne skrócenie czasu maszyny w stosunku do metody prób i błędów. Przykład wykorzystania możliwości automatycznej optymalizacji podano w Dodatku B.

5. STEROWANIE PROCESEM TRANSLACJI, OBLICZEŃ I REJESTRACJĄ WYNIKÓW

W języku CEMMA 2 przewidziano 24 dyrektywy, celem umożliwienia programującemu dostatecznego wpływu na przebieg translacji i wykonania programu oraz na postać wydruku wyników działania programu. Dyrektyw tych nie będziemy dokładnie opisywać, zainteresowanych odsyłamy do szczegółowego opisu języka. Warto tu tylko wspomnieć, że pozwalają one m.in. na:

- wstrzymanie procesu symulacji w celu dokonania zmian w programie i wznowienia obliczeń od momentu przerwania,
- określenie dopuszczalnego bezwzględnego błędu całkowania przy stosowaniu metody o zmiennym kroku,
- określenie warunków zakończenia procesu symulacji,
- stosowanie parametrów symbolicznych, których wartości podawane są poza programem lub określane w procesie optymalizacji modelu,
- wybór metody całkowania spośród dostępnych sześciu metod,
- zadawanie minimalnego kroku całkowania przy stosowaniu metody o zmiennym kroku,
- określenie postaci danych wyjściowych.

W systemie CEMMA 2 przewidziano zasadnicze trzy formy przedstawienia wyników symulacji:

- tablica wartości zmiennych modelu,
- przebiegi zmiennych przedstawione w formie zbioru punktów zaznaczonych wybranym znakiem drukarki /tzw. postać graficzna wyników/,
- przebiegi zmiennych w postaci linii reprezentujących kolejne rzędne z jednoczesnym wydrukiem ich wartości /tzw. postać numeryczno-graficzna/.

Wszystkie te formy mogą być wykorzystywane równocześnie w tym samym programie symulacji, jeśli wymaga tego dalszy proces analizy wyników symulacji.

6. ZAKOŃCZENIE

Przedstawiony w artykule język symulacyjny CEMMA 2 był stosowany do symulacji różnych procesów ciągłych. Dotychczasowe doświadczenia wskazują, że jest on łatwy do opanowania, szczególnie dla osób znających technikę modelowania analogowego. System CEMMA 2 jest również narzędziem bardzo efektywnym z punktu widzenia programującego, gdyż pozwala opisywać dość złożone procesy w bardzo zwartej i czytelnej formie. Możliwość automatycznego wyznaczania optymalnych wartości parametrów modelu pozwala na kilkakrotne skrócenie czasu w stosunku do metody prób i błędów dla procesów opisywanych kilkudziesięcioma równaniami różniczkowymi I rzędu. Należy jednak oczekiwać, że w miarę gromadzenia doświadczeń z wykorzystania języka CEMMA 2 pojawią się również zastrzeżenia odnośnie niektórych rozwiązań, a co za tym idzie potrzeba istotnych zmian. Autorzy będą wdzięczni wszystkim użytkownikom języka CEMMA 2 za nadsyłanie swoich uwag w tym zakresie.

DODATEK A

Podstawowe zdania języka GEMMA 2

Postać ogólna	Realizowana funkcja
k INT n, P	$Y_k = \int_0^t Y_n \cdot dt + P$
k DOD $n_1, P_1, n_2, P_2, \dots, n_m, P_m$	$Y_k = \sum_{i=1}^m P_i \cdot Y_{n_i}$
k SUM n_1, n_2, \dots, n_m	$Y_k = \sum_{i=1}^m Y_{n_i}$
k MNZ n_1, n_2, \dots, n_m, P	$Y_k = P \cdot \prod_{i=1}^m Y_{n_i}$
k DZL n_1, n_2	$Y_k = \frac{Y_{n_1}}{Y_{n_2}} ; \quad Y_{n_2} \neq 0$
k PWK n	$Y_k = \sqrt{Y_n} ; \quad Y_n > 0$

Postać ogólna	Realizowana funkcja
k EXP n	$Y_k = e^{Y_n}; \quad Y_n \leq 176$
k LGN n	$Y_k = \ln Y_n; \quad Y_n > 0$
k SIN n	$Y_k = \sin Y_n$
k COS n	$Y_k = \cos Y_n$
k ATG n	$Y_k = \operatorname{arctg} Y_n$
k ABS n	$Y_k = Y_n $
k AND n_1, n_2	$Y_k = 1$ gdy $Y_{n_1} > 1, Y_{n_2} > 1$ $Y_k = 0$ w innych przypadkach
k UOR n_1, n_2	$Y_k = 0$ gdy $Y_{n_1} \leq 0, Y_{n_2} \leq 0$ $Y_k = 1$ w innych przypadkach

Postać ogólna	Realizowana funkcja
k XOR n_1, n_2	$Y_k = 1 \text{ gdy } Y_{n_1} \leq 0, Y_{n_2} > 0$ $Y_k = 1 \text{ gdy } Y_{n_1} > 0, Y_{n_2} \leq 0$ $Y_k = 0 \text{ w innych przypadkach}$
k NOR n_1, n_2	$Y_k = 1 \text{ gdy } Y_{n_1} \leq 0, Y_{n_2} \leq 0$ $Y_k = 0 \text{ w innych przypadkach}$
k NEG n	$Y_k = 1 \text{ gdy } Y_n \leq 0$ $Y_k = 0 \text{ gdy } Y_n > 0$
k GNL P_1, P_2	$Y_k = 0 \text{ gdy } t < P_1$ $Y_k = P_2 \cdot (t - P_1) \text{ gdy } t \geq P_1$
k GSK P_1, P_2	$Y_k = 0 \text{ gdy } t < P_1$ $Y_k = P_2 \text{ gdy } t \geq P_1$

Postać ogólna	Realizowana funkcja
$k \text{ GGS } P_1, P_2, P_3$	<p>Zmienna Y_k o normalnym rozkładzie prawdopodobieństwa przy czym</p> <p>P_1 - wartość średnia</p> <p>P_2 - odchylenie standardowe</p> <p>$P_3 \neq 0$ wartość początkowa</p>
$k \text{ GRN } P_1, P_2, P_3$	<p>Zmienna Y_k o równomiernym rozkładzie prawdopodobieństwa przy</p> <p>P_1, P_2 - min i max wartość Y_k</p> <p>$P_3 \neq 0$ - wartość początkowa</p>
$k \text{ OGR } m, P_1, P_2, P_3$	$Y_k = P_3 \cdot P_1 \text{ gdy } Y_n < P_1;$ $Y_k = P_3 \cdot P_2 \text{ gdy } Y_n > P_2;$ $Y_k = P_3 \cdot Y_n \text{ gdy } P_1 < Y_n < P_2;$ $P_2 > P_1$
$k \text{ SHN } m, P_1, P_2, P_3$	$Y_k = 0 \text{ gdy } P_1 \leq Y_n \leq P_2$ $Y_k = P_3 \cdot (Y_n - P_1) \text{ gdy } Y_n < P_1$ $Y_k = P_3 \cdot (Y_n - P_2) \text{ gdy } Y_n > P_2;$ $P_1 < P_2$

Postać ogólna	Realizowana funkcja
k KNT n, P	$Y_k = P \cdot \text{ENT} \left\{ \frac{Y_n + \frac{P}{2}}{P} \right\}$
k LUZ n, P_1, P_2, P_3, P_4	Patrz rys. A.1
k PRZ n, P_1, P_2, P_3, P_4	Patrz rys. A.2
k PHZ $n, P_1, P_2, P_3, P_4, P_5, P_6$	Patrz rys. A.3
k COM n_1, n_2	$Y_k = 0 \quad \text{gdy} \quad Y_{n_2} > Y_{n_1}$ $Y_k = 1 \quad \text{gdy} \quad Y_{n_2} \leq Y_{n_1}$
k PWE n_1, n_2, n_3, n_4	$Y_k = Y_{n_2} \quad \text{gdy} \quad Y_{n_1} < 0$ $Y_k = Y_{n_3} \quad \text{gdy} \quad Y_{n_1} = 0$ $Y_k = Y_{n_4} \quad \text{gdy} \quad Y_{n_1} > 0$

Postać ogólna	Realizowana funkcja
<p>k RST n_1, n_2, n_3, P</p>	<p>a/ $Y_k = 0$ gdy $Y_{n_1} > 0$ $/Y_{n_2}, Y_{n_3} - \text{dowolne}/$</p> <p>b/ $Y_k = 1$ gdy $Y_{n_1} \leq 0$ $/Y_{n_3} - \text{dowolne}/$</p> <p>c/ $Y_k = 0$ gdy $Y_{n_1}, Y_{n_2} \leq 0$ i $Y_{n_3} > 0, Y_k(t - \Delta) = 1$ lub $Y_{n_3} \leq 0, Y_k(t - \Delta) = 0$</p> <p>d/ $Y_k = 1$ gdy $Y_{n_1}, Y_{n_2} \leq 0$ i $Y_{n_3} \leq 0, Y_k(t - \Delta) = 1$ lub $Y_{n_3} > 0, Y_k(t - \Delta) = 0$</p>
<p>k FUN n, F_1</p>	<p>Funkcja jednej zmiennej zadana w postaci tablicy F_1</p>
<p>k FUD n_1, n_2, F_1</p>	<p>Funkcja dwóch zmiennych zadana w postaci tablicy F_1</p>

Postać ogólna	realizowana funkcja
<p>k PAL n_1, n_2, \dots, n $/m_1, m_2, \dots, m_s/$ P_1, P_2</p>	<p>Zadanie PAL opisuje tzw. pętlę algebraiczną /nie zawierającą zdania całkowania i opóźnienia/ złożoną ze zdań o numerach m_1, m_2, \dots, m_s</p>
<p>k MAX n_1, n_2, \dots, n_m</p>	$Y_k = \max\{Y_{n_1}, Y_{n_2}, \dots, Y_{n_m}\}$
<p>k MIN n_1, n_2, \dots, n_m</p>	$Y_k = \min\{Y_{n_1}, Y_{n_2}, \dots, Y_{n_m}\}$
<p>k PMX n</p>	$Y_k(T) = \max[Y_n(t)] \quad 0 \leq t \leq T$
<p>k OPZ n, P</p>	$Y_k = Y_n(t - P) \quad \text{gdy } t > P$ $Y_k = 0 \quad \text{gdy } t < P$
<p>k IMP n, P_1, P_2</p>	$Y_k = Y_n(m \cdot P_1)$ <p style="text-align: center;">gdy $mP_1 \leq t \leq mP_1 + P_2$</p> $Y_k = 0$ <p style="text-align: center;">gdy $mP_1 + P_2 < t < (m + 1)P_1$</p>

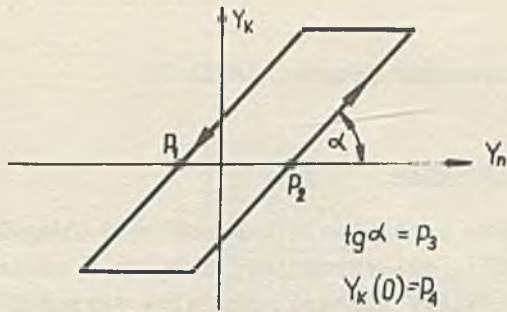
Postać ogólna	Realizowana funkcja
$k \text{ ZHD } n_1, n_2$	$Y_k = Y_{n_2} \quad \text{gdy } Y_{n_1} > 0$ $Y_k = Y_k(t - \Delta) \quad (\text{gdy } Y_{n_1} \leq 0)$

Oznaczenia: Y_k, Y_n - wielkości wyjściowe bloków o numerach k i n

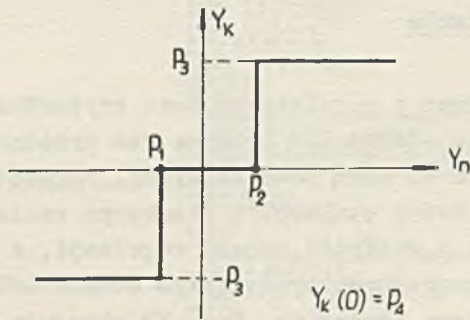
Δ - przyrost czasu

t - czas

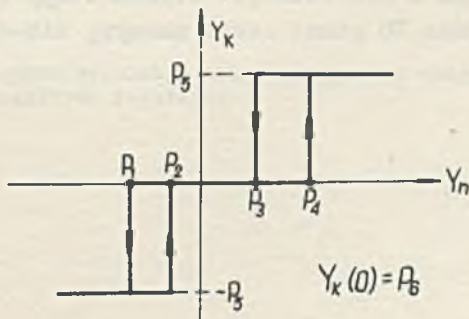
k, n, m, l, s - liczby naturalne



Rys. A.1. Funkcja zdania LUZ



Rys. A.2. Funkcja zdania PRZ



Rys. A.3. Funkcja zdania PHZ

DODATEK B

Przykład zastosowania języka CEMMA 2

Sformułowanie zadania

Absorber gazu z regulacją dopływu wody /regulator pneumatyczny typu PI/ opisywany jest układem 6 równań różniczkowych I rzędu. Należy znaleźć optymalne wartości parametrów regulatora do uzyskania minimum całki z kwadratu uchybu regulacji pomnożonego przez czas przy wymuszeniu skokowym.

Rozwiązanie zadania

Model absorbera z regulatorem oraz kryterium optymalizacji opisano w języku CEMMA 2, program ten przedstawiono na rys. B1 /program zawiera 68 zdań podstawowych/. Parametry regulatora optymalizowano w dwóch etapach. W pierwszym zastosowano zwykle kryterium całki z kwadratu uchybu regulacji, a następnie przy wartościach wyznaczonych w pierwszym etapie dokonano optymalizacji przy zadanym kryterium. Rys. B2 pokazuje odpowiedź na wymuszenie skokowe układu optymalnego, przy czym oprócz podanego przebiegu uzyskano szereg innych danych zarówno w formie tabelarycznej jak i graficznej. Obydwa etapy optymalizacji zajęły łącznie około 20 minut czasu maszyny ZAM-41.


```

[ DEFINICJE MAKROBLOKOW
MAKRO SINUS N1 M21
1 MZ2 N1 M11
2 SIN W11
3 MZ2 M21
KOMAKRO1
MAKRO REALPL. N1 M21
1 D00 N1 M2 W4 ==1 MZ2 = WZMOCNIENIE
2 GSK =0 N11 M1 = T
3 DZL =1 M21
4 INT W31
KOMAKRO1
[KONIEC DEFINICJI MAKROBLOKOW

[OPIS MODELU UKLADU
0 MZ2 51 P221
6 GSK =0 =0,11
9 D00 1 P7 6 P19 20 P201 WEJSCIE = Y0
10 C03 01
20 D00 6 =1 10 =0,11
1 GRN =0,1 =0,2 P11
2 D00 9 P2 7 ==1 11 =11
3 GSK =0 P111
4 DZL 2 31
5 INT 81
60 SINUS 5 P12 P131
7 MZ2 63 P41
8 MZ2 5 941
90 REALPL 104 P16 =11
100 REALPL 17 P17 =11
11 MZ2 104 151
12 D00 7 =1 11 ==1 163 P51
13 GSK =0 P101
14 DZL 12 131
15 INT 141
160 SINUS-15 P14 P151
17 D00 18 =1 27 P81 03L
18 GSK =0 P91
21 STN 163 ==0,002 =0,002 =1 STREFA WARTWA
[REGULATOR PNEUMATYCZNY P1
22 GSK =0 P191
23 D00 21 ==1 22 =11
24 INT 23 111
25 D00 23 P30 24 P311
33 D00 34 P41 50 P401
29 OPZ 26 P6 2001 OP0ZNIENIE
26 OGR 25 =0,2 =1 =11
27 POK 331
46 REALPL 29 =0,03 =11
34 DZL 36 351
35 GSK =0 P501
36 INT 371
37 PH2 40 ==0,2 ==0,1 =0,1 =0,2 =1 01
41 REALPL 37 P51 P521
40 D00 23 =1 45 ==11

[FUNKCJE CELU 1
51 GSK =0 =11
52 MZ2 51 541
53 INT 521 MR 1
54 MZ2 23 231
55 INT 541 NR 2
56 INT 571 NR 3
57 ABS 331
[KONIEC OPISU MODELU
STARTONE 11=0,61
PARAMETRY P19=1 P7=0 P20=0 P22=6,25 P1=0,15 P2=0,08 P4=0,08
P5=-0,08 P6=0,1 P8=0 P9=1,2 P10=2 P11=2 P12=0,8 P14=0,5
P13=31 P15=31 P16=0,15 P17=0,15 P18=0,05 P30=1 P31=1 P40=0
P41=0 P50=1 P51=1 P52=1 1
FINISZ 163 =0,21
ZEDAR =10 =0,01 =0,5 =0,5 =0,51
TYTUL
BADANIE UKLADU REGULACJI
OBIEKT ABSORBER GAZU
REGULATOR PNEUMATYCZNY TYPU P1
X3=0 GG=CONST Q3L=CONST1
DRUKUJ 5 7 94 11 14 15 1631
ZAKRES 5 7 11 15 1631
RYSUJ 1 X 15 = -0,7E-2 1
RYSUJ 1 5 5 = -4E-31
RYSUJ 1 7 7 = =1
RYSUJ 2 P 94 = =41
RYSUJ 2 0 104 = =1
WYKRES 163 = =0,031
GRAFIK 1
KONIEC 1
TYTUL
OPTIMALIZACJA REGULATORA PNEUMATYCZNEGO
FUNKCJA CELU NR 21
OPTIMALIZUJ ROSENBR0CK 2 55 P30 =0 =100 P31 =0 =1001
PARAMETRY P7=0 P19=1 P20=0 P8=1,6 P9=0 P40=1 P41=0 P31=1 P30=11
WYKRES 26 =1 WEJSCIE REGULATORA PNEUMATYCZNEGO
RYSUJ 3 W 21 = =1
ANULUJ DRUKUJ1
METODA PROSTOKATOW1
GRAFIKI
KONIECI
TYTUL
OPTIMALIZACJA REGULATORA PNEUMATYCZNEGO
FUNKCJA CELU NR 31
OPTIMALIZUJ ROSENBR0CK 2 56 P30 =0 =10 P31 =0 =1001
PARAMETRY P7=0 P19=1 P20=0 P8=1,6 P9=0 P40=1 P41=01
GRAFIKI
KONIECI
OPTIMALIZUJ ROSENBR0CK 2 55 P51 =0 =50 P52 =0 =501
PARAMETRY P40=0 P41=11
GRAFIKI
METODA PROSTOKATOW1
RYSUJ 4 K 37 = =1
RYSUJ 4 L 34 = =1
KONIECI
STOPI

```

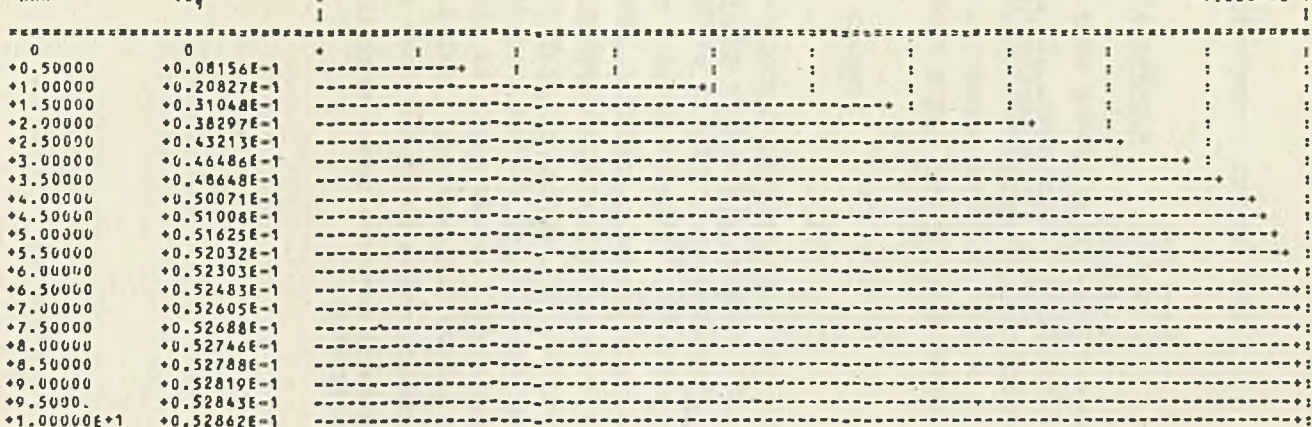
Rys. B.1. Program symulacji absorbera gazu z optymalizacją parametrów regulatora

WYKRES-1
XXX

163

0

+0.52862E-1



KONIEC-1

Rys. B.2. Odpowiedź regulatora optymalnego

Literatura

- [1] ROSENBROCK H. H., STOREY C.: Computational Techniques for Chemical Engineers, Pergamon Press, 1966.
- [2] WILDE D. J.: Optimum Seeking Methods, Prentice Hall, 1964.
- [3] POWELL M. J. D.: An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives, Computer J., 1964: 6, 155-162.
- [4] BOX M. J.: A Comparison of Several Current Optimization Methods and the Use of Transformations in Constrained Problems, Computer J., 1966: 9, 67.
- [5] ZANGWILL W. I.: Minimizing a Function without Calculating Derivatives, Computer J., 1967: 10, 293.
- [6] MADEJ W.: CEMMA - język do symulacji układów dynamicznych, Prace IMM, Warszawa, 1970: 2.
- [7] Krótkie informacje, ETO NOWOŚCI Nr 1, 1971.

СЕММА2 - ЯЗЫК ДЛЯ МОДЕЛИРОВАНИЯ НЕПРЕРЫВНЫХ ПРОЦЕССОВ С АВТОМАТИЧЕСКОЙ ОПТИМИЗАЦИЕЙ ПАРАМЕТРОВ МОДЕЛИ

Резюме

В статье описаны характеристики языка СЕММА2 предназначенного для моделирования непрерывных и импульсных процессов на цифровой машине. Язык СЕММА2 является развернутой и усовершенствованной версией языка СЕММА описанного в "Prace IMM" 2/70.

Язык СЕММА2 реализованный на машине ЗАМ-41 принадлежит к категории цифровых моделей аналоговой машины; он основан на принципе блочного описания моделируемого процесса. В статье указана структура и значение отдельных предложений языка. Описан способ решения задач оптимизации, принципы определения сложных предложений, а также вид данных полученных в процессе моделирования.

Из опыта следует, что языком СЕММА2 можно легко овладеть и что он описывает сложные процессы в очень компактной форме. Благодаря автоматическому определению оптимальных величин параметров модели, язык СЕММА2 в несколько раз сокращает время машины по отношению к методу многократных проб для процессов описываемых несколькими десятками дифференциальных уравнений I порядка.

В приложении к статье подан пример применения системы СЕММА2.

CEMMA 2 LANGUAGE FOR SIMULATION OF CONTINUOUS PROCESSES WITH MODEL
PARAMETER AUTOMATIC OPTIMALIZATION

Summary

The article describes the characteristics of the CEMMA 2 language destined to simulate continuous and sampled-data processes on a digital computer.

CEMMA 2 language is a developed and improved version of the CEMMA language described in "PRACE IMM" nr 2/70.

CEMMA 2, implemented on ZAM-41 computer belongs to the category of digital simulators of analogue computers and is based on the principle of a block description of the modelled process.

The article presents the structure and the meaning of separate language statements. It describes the way of solving optimization problems, principles of defining complex statements and the forms of data obtained in the process of simulation as well.

It results from the experience that the CEMMA 2 language is easy to be learned; it permits to describe complex processes in a concise way. Due to an automatic calculation of optimal values of the model parameters the computer time may be shortened several times as compared with the trials and errors method for processes described by dozens differential equations of I order.

An example of the CEMMA 2 system application is given in supplement.

ANALIZA EFEKTYWNOŚCI MODELOWANIA SIECI
PRZEŁĄCZAJĄCYCH METODĄ LISTOWĄ I BLOKOWĄ

Tadeusz SINKIEWICZ

Pracę złożono 25.05.1971

Praca zawiera porównawczą analizę parametrów modelowania dwu metod modelowania czasowego binarnych sieci przełączających. Po omówieniu ogólnych zasad organizacji procesu modelowania w metodzie listowej i blokowej przeprowadzono szacowanie parametrów modelowania wewnętrznych procesów przełączania dużych sieci przełączających /do 16-tu tysięcy elementów/ przy wykorzystaniu EMC II i III generacji. Szacowanie przeprowadzono na podstawie schematów blokowych programów po dokładnej analizie liczby rozkazów maszyny dla poszczególnych czynności modelowania. Czas modelowania wyznaczano w ilościach operacji maszyny zakładając, że program modelujący realizowany jest w języku wewnętrznym maszyny /język PJES EMC ZAM-41/.

Oszacowania liczbowe przeprowadzono dla typowych sieci przełączających części centralnej EMC III generacji.

WYKAZ PODSTAWOWYCH OZNACZEŃ

E_x	- liczba zdarzeń x
G	- liczba zdarzeń przełączania
H	- długość listy zdarzeń przełączania
H_{PB}	- liczba wierszy listy połączeń bloków
L	- maksymalna liczba wejść elementu przełączającego
$L_{\bar{sr}}$	- średnia liczba wykorzystanych wejść elementów przełączających modelowanej sieci

- N - maksymalna liczba elementów przełączających modelowanej sieci
- N_d - liczba elementów przełączających modelowanej sieci wzbudzonych dynamicznie
- N_{rz} - rzeczywista liczba elementów przełączających modelowanej sieci przełączającej
- n - maksymalna liczba elementów przełączających w bloku modelowanej sieci przełączającej
- n_n - maksymalna liczba "nadawczych" elementów przełączających w bloku modelowanej sieci
- n_o - maksymalna liczba "odbiorczych" elementów przełączających w bloku modelowanej sieci
- n_w - maksymalna liczba "wewnętrznych" elementów przełączających w bloku modelowanej sieci
- P - liczba typów elementów przełączających w modelowanej sieci
- p - typ elementu przełączającego / $p = 0 \div P$ /
- $r_o \div r_5$ - parametry pamięci zewnętrznych EMC
- s - krotność zdarzenia wielokrotnego
- T_p - czas trwania przejściowego procesu przełączania elementu przełączającego /w taktach modelowania/
- t_{podpr} - średnia liczba operacji przy realizacji podprogramu elementu przełączającego
- t_H - średnia liczba operacji maszyny dla wykonania czynności modelowania przy jednym obiegu listy zdarzeń przełączania
- t_{H1} - średnia liczba operacji maszyny dla wykonania czynności jednego taktu modelowania metodą listową
- t_w - średnia liczba operacji maszyny dla wykonania czynności jednego taktu modelowania metodą blokową
- U - liczba taktów modelowania
- w - liczba bloków, na którą podzielono sieć przełączającą do modelowania metodą blokową / $w = \frac{N}{n}$ /
- y - pojemność pamięci rdzeniowej modelującej EMC
- Y - całkowita pojemność pamięci /zewnętrznej i rdzeniowej/ modelującej EMC
- Z - maksymalna obciążalność wyjść elementów przełączających
- Z_{sr} - średnie obciążenie wyjść elementów przełączających modelowanej sieci
- τ - średni czas wykonywania operacji w maszynie modelującej
- θ - jednostkowa czasochłonność modelowania

1. WSTĘP

Programy modelujące stanowią nieodzowną część systemów automatycznego projektowania elektronicznych systemów cyfrowych. Programy te przyspieszają proces konstruowania i ułatwiają optymalizację projektu.

Celem niniejszej pracy jest ocena dwu metod modelowania logiczno-czasowego binarnych sieci przełączających, jakie mogą być zastosowane w systemie automatycznego projektowania EMC. Metody te charakteryzują się odmiennymi sposobami kodowania sieci i różną organizacją procesu modelowania.

Podstawowe przeszkody jakie towarzyszą porównaniu efektywności metod modelowania to przede wszystkim

- trudność dokładnej oceny metody bez praktycznej realizacji programu modelującego
- brak kryteriów oceny złożoności sieci przełączających i zachodzących w nich procesów przełączania pod względem efektywności ich modelowania.

Pierwszą z wymienionych trudności starano się w pracy pokonać drogą szacowania czasochłonności modelowania na podstawie schematów blokowych programów /wykresów czynności/ wykonanych dla obydwu rozpatrywanych metod. Zakładano przy tym, że programy modelowania zrealizowane są w języku wewnętrznym maszyny "wzorcowej" o ustalonych parametrach.

Dla pokonania drugiej trudności wprowadzono współczynniki charakteryzujące intensywność procesów przełączania.

W pracy zwrócono szczególną uwagę na zagadnienia modelowania dużych sieci przełączających /do 16 384 elementów/ charakterystycznych dla systemów automatycznego projektowania.

Czas realizacji czynności modelowania liczony będzie w jednostkach odpowiadających średniej wartości czasu wykonywania operacji w części centralnej maszyny cyfrowej. Umożliwi to wy-

korzystanie otrzymanych wyników do szacowania czasu modelowania dla maszyn o różnych szybkościach.

Podstawę przy szacowaniu ilości rozkazów do wykonania poszczególnych czynności stanowiły opracowane w IMM programy modelowania dla EMC ZAM 41 w języku wewnętrznym PJES /PJEG/.

Zakres rozpatrywanych zagadnień ograniczony został do omówienia podstawowego cyklu modelowania, oceny czasu modelowania wewnętrznych procesów przełączania w modelowanej sieci i wymaganej do tego celu objętości pamięci rdzeniowej. W rozważaniach pominięte zostały ważne skądinąd problemy ustawiania wartości początkowych, wyboru wielkości taktu modelowania, metod przetwarzania sygnałów pojawiających się jednocześnie na wejściach elementów przełączających, wprowadzania sygnałów zewnętrznych itp. Nie będą również rozpatrywane sposoby realizacji programów pomocniczych /np. drukowania/ nie wpływających bezpośrednio na przebieg modelowania.

2. ZAŁOŻENIA UPRASZCZAJĄCE I PARAMETRY MASZYN "WZORCOWYCH"

W celu uzyskania porównawczych wskaźników liczbowych dla rozpatrywanych w pracy metod modelowania, niezbędne stało się przyjęcie szeregu założeń upraszczających.

W dalszej części pracy dla obydwu metod zakłada się, że modelowana sieć przełączająca spełnia następujące założenia:

1. Wszystkie elementy przełączające sieci mają ten sam czas trwania przejściowego procesu przełączania T_p /sieć wykonana jest z elementów jednorodnych/
2. Maksymalna liczba wyjść elementów przełączających sieci wynosi $M=2$, natomiast maksymalna liczba ich biegunów /wejść i wyjść/ spełnia warunek $L+M \leq 16$
3. Maksymalna obciążalność wyjść elementów przełączających wynosi Z

4. Dynamiczne procesy przełączania przebiegają w modelowanych sieciach w sposób stacjonarny, charakteryzujący się stałą aktywnością przełączania /zgodnie z definicją podaną w p. 5/.

Do szacowania zajętości pamięci i czasu modelowania posługiwaliśmy się średnimi wartościami wykorzystanych wejść $/L_{\text{sr}}/$, wyjść $/M_{\text{sr}}/$ i obciążalności $/Z_{\text{sr}}/$ elementów przełączających. Wartości te pozwalają bardziej realistycznie ocenić parametry sieci, a co za tym idzie efektywność metod modelowania.

Celem zilustrowania podamy, że w wyniku przeprowadzonej analizy dla części centralnych EMC trzeciej generacji projektowanych w Instytucie Maszyn Matematycznych, przy wykorzystaniu układów scalonych serii TTL, wartości L_{sr} mieściły się w granicach $4 \div 6$, podczas gdy wartość L dla tych układów wynosi 10.

Podstawowe założenia przyjęte przy rozpatrywaniu metody listowej są zgodne z założeniami podanymi w pracy [4].

Przy rozpatrywaniu blokowej metody modelowania przyjęto dodatkowe założenie częściowego uporządkowania struktury połączeń między elementami przełączającymi. Zakłada się mianowicie, że połączenia międzyblokowe wykonywane są wyłącznie pomiędzy wydzielonymi grupami elementów przełączających bloku, nazywanych "odbiorczymi" i "nadawczymi" elementami bloku. Do wejść elementów odbiorczych doprowadzane są połączenia z zewnątrz bloku, natomiast wyjścia elementów nadawczych sterują elementami znajdującymi się na zewnątrz rozpatrywanego bloku. Wejścia i wyjścia tzw. elementów "wewnętrznych" nie są wprowadzane na zewnątrz bloku.

Drugim dodatkowym założeniem, przyjętym przy szacowaniu czasu modelowania metodą blokową, jest założenie pełnego wykorzystania pojemności bloków, tj. przyjęcie, że rzeczywista liczba elementów przełączających w bloku jest równa odpowiedniej dla tego bloku wartości maksymalnej $/n/$.

Parametry maszyny wzorcowej podane są w umownych jednostkach τ_1 , τ_1 , umożliwiającą interpretację otrzymanych wyników dla maszyn o różnych szybkościach działania części centralnych i różnych parametrach pamięci rdzeniowych i zewnętrznych. Jako podstawową jednostkę czasu przyjęto średni czas wykonania operacji w części centralnej maszyny. W tabeli 1 zestawiono parametry maszyny wzorcowej z typowymi parametrami krajowych EMC II i III generacji.

3. METODA LISTOWA^{*)}

Listowa metoda modelowania działania sieci neuronowych podana została w [5]. Zastosowanie tej metody do modelowania sieci logicznych przedstawiono w [3] i rozwinięto w [4]. Metodę wykorzystano w systemie automatycznego projektowania maszyn serii IBM-360. Podstawowe założenia tej metody stosowane są również w języku modelowania "Simscrip".

3.1. Kodowanie struktury sieci przełączającej

Rozpatrzmy postać informacji określającej strukturę logiczną sieci przełączającej, z której bezpośrednio korzysta program modelujący. Informacja ta otrzymywana jest z podstawowego pliku informacji w rezultacie działania specjalnego programu, tak zwanego ekstraktora.

Pojedynczy biegun /wejście lub wyjście/ elementu przełączającego określony jest przez następującą informację:

- wartość sygnału logicznego
- cecha bieguna /odróżniająca wejście od wyjścia/
- numer bieguna wejściowego połączonego z biegunem rozpatrywanym $/I_H/$

Gdy biegun nie jest połączony z innymi biegunami, w polu $/I_H/$ znajduje się zero. W przypadku, gdy biegun jest wyjściem,

^{*)} Metoda ta nazywana jest również metodą zdarzeń

Tabela 1

Zestawienie parametrów maszyny wzorcowej z parametrami EMC II i III generacji

Urządzenie maszyny	Maszyna wzorcowea	EMC II generacji	EMC III generacji
<u>Część centralna /procesor/</u>			
● długość słowa	24 bity	24 bity	24 bity
● średni czas wykonywania operacji	τ	$\tau = 25 \mu s$	$\tau = 4 \mu s$
● pojemność pamięci rdzeniowej /słów 24-bitowych/	y	y = 16k	y = 16k
<u>Moduł pamięci bębnowej /dyskowej/</u>			
● pojemność /słów 24-bitowych/	r_0	$r_0 = 32k$	$r_0 = 600k$
● średni czas dostępu	$\tau_1 = r_1 \tau$	20 ms, $r_1 = 800$	20 ms, $r_1 = 500$
● szybkość przekazywania informacji	r_2 słów/100 τ	6000 słów/s, $r_2 = 15$	30000 słów/s, $r_2 = 12$
<u>Stacja pamięci taśmowej</u>			
● pojemność /słów 24-bitowych/	r_3	$r_3 = 2,5$ mln	$r_3 = 5$ mln
● czas startu taśmy	$\tau_2 = r_4 \tau$	6 ms, $r_4 = 240$	3 ms, $r_4 = 750$
● szybkość przekazywania informacji	r_5 słów/100 τ	5300 słów/s, $r_5 = 13,2$	60000 słów/s, $r_5 = 24$

w polu $/I_H/$ umieszczony jest adres dołączonego do niego wejścia. Jeżeli do wyjścia dołączono kilka wejść, to ich numery umieszcza się kolejno w polach $/I_H/$ informacji kodującej poszczególne wejścia. W ten sposób powstają listy określające kolejne bieguny leżące na drodze sygnału. Listy połączeń grupowane są w określonych obszarach pamięci zależnie od typu elementu przełączającego [3] lub uzupełniane są dodatkową informacją definiującą [4]. W pierwszym przypadku zbędnym staje się kodowanie typu elementu lecz pojawiają się trudności z rozmieszczeniem informacji w pamięci.

3.2. Realizacja modelowania

Śledzenie dynamiki przełączania elementów sieci przeprowadza się za pomocą tak zwanej listy zdarzeń przełączania znajdującej się stale w pamięci rdzeniowej. Wymiary listy określone są przez maksymalną liczbę wyjść elementów przełączających, które mogą być przełączane równoległe oraz przez maksymalny czas przełączania T_P , mierzony w taktach modelowania. Jeżeli w danym takcie zmienia się wartość sygnału na wyjściu umieszczonym w liście, program modelujący na podstawie list połączeń określa nowe elementy przełączające, które mogą być uruchomione tą zmianą sygnału. Dla każdego potencjalnie pobudzonego elementu realizowane są odpowiadające mu podprogramy, i w przypadku gdy spełnione są warunki przełączania tego elementu do listy zdarzeń przełączania wpisywane są numery /adresy/ wyjść elementu z zaznaczeniem taktu modelowania, w którym następuje zmiana wartości sygnału wyjściowego. Jednocześnie z listy usuwany jest numer tego wyjścia, na którym nastąpiła zmiana wartości sygnału w danym takcie modelowania.

Tak więc zawartość listy zdarzeń przełączania jest aktualizowana w każdym takcie modelowania i znajdują się w niej numery wyjść tylko tych elementów, w których aktualnie odbywa się proces przełączania.

Jak wykazują statystyki [1], [3] procentowa liczba elementów, w których odbywa się proces przełączania do całkowitej

liczby elementów sieci przełączającej dla typowych układów maszyn matematycznych waha się w granicach $1 \div 20\%$ i w większości przypadków przyjmuje wartości poniżej 5%. Dzięki temu w pewnych warunkach można uzyskać znaczne skrócenie czasu modelowania.

Uproszczony wykres czynności programu zrealizowanego według omawianej metody modelowania podano na rys. 1. Działanie programu rozpoczyna się od ustalenia wartości początkowych. Dalsza realizacja programu sprowadza się w zasadzie do cyklicznego przebiegania trzech podstawowych pętli realizujących odpowiednio:

- dystrybucję czasu modelowania
- dystrybucję sygnałów zewnętrznych sieci przełączającej
- przetwarzanie sygnałów wewnętrznych sieci przełączającej

Jak zaznaczono we wstępie, w dalszej części artykułu ograniczymy się do rozpatrzenia pętli przetwarzania sygnałów wewnętrznych.

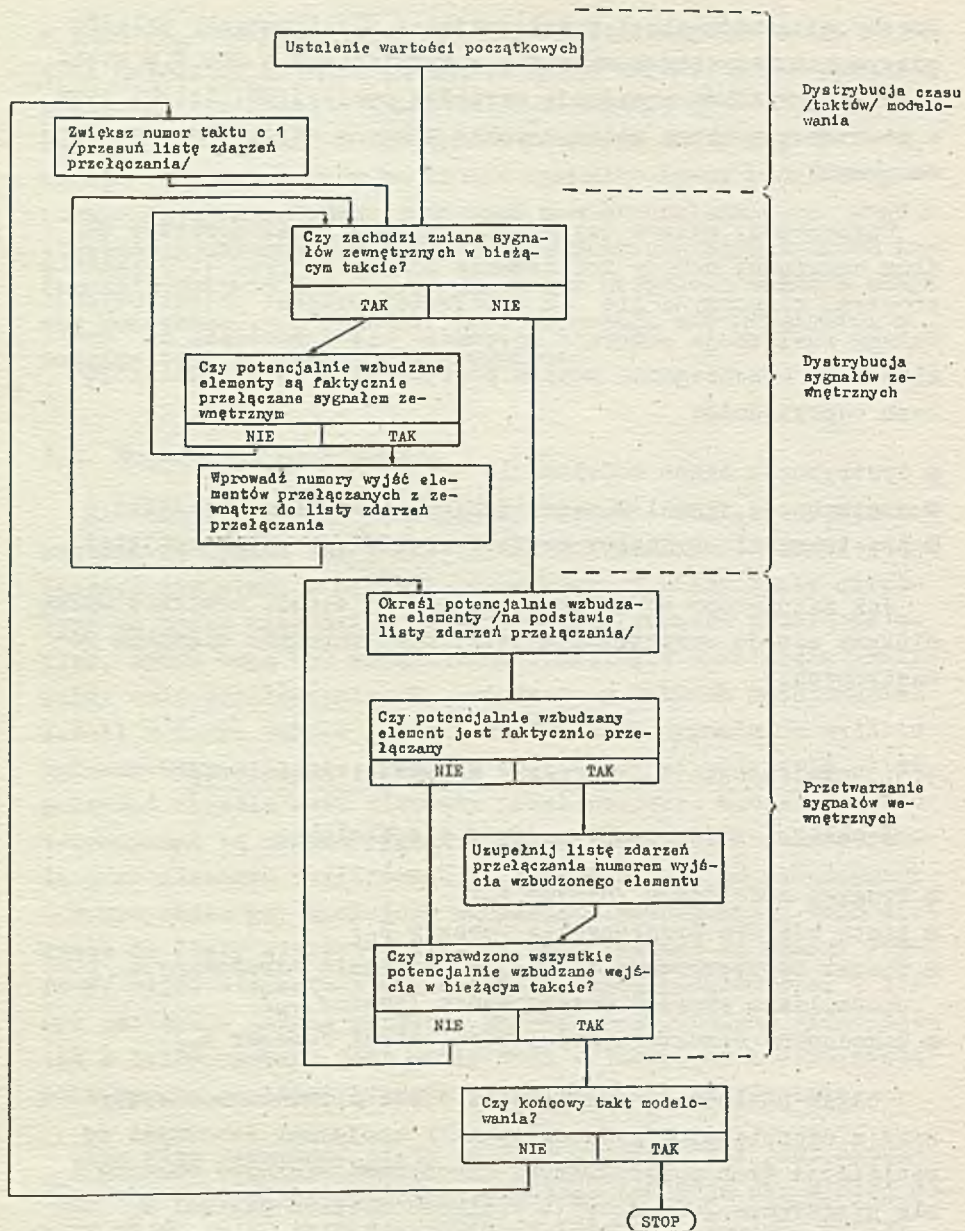
3.3. Roźmieszczenie informacji w pamięci rdzeniowej

W pamięci rdzeniowej znajdują się stale

- Program modelujący /obszar A_H /
- Lista zdarzeń przełączania /obszar B_H /
- Obszar zarezerwowany dla wyjść pośrednio adresowanych poprzez listę zdarzeń przełączania /obszar C_H /
- Zakodowana struktura modelowanej sieci /obszar D_H /

Obszar pamięci A_H zajęty jest przez program modelujący wraz z odpowiednimi podprogramami i miejscami roboczymi do realizacji tych podprogramów. Szacunkowa wielkość obszaru A_H nie przekracza 2000 słów^{*)}. Wielkość obszaru pamięci B_H za-

*) Jest to wartość przeciętna, gdyż wielkość obszaru A_H w dużym stopniu zależy od liczby typów modelowanych elementów przełączających



Rys. 1. Uproszczony wykres czynności modelowania metodą listową

leżna jest od przyjętego taktu modelowania i związana z tym długością listy zdarzeń przełączania. Zgodnie z [4] przyjmujemy typową długość listy zdarzeń przełączania równą 150 słowom.

Obszar C_H winien mieć taką wielkość, aby umożliwić dołączenie do listy zdarzeń przełączania maksymalnej liczby wyjść jaka może być wzbudzona dynamicznie przy jednym obiegu listy. Zgodnie z podanymi poprzednio oszacowaniami przyjmujemy, że wielkość obszaru C_H określona jest przez liczbę słów niezbędną do kodowania 20% wyjść modelowanej sieci przełączającej przy założeniu, że elementy mają maksymalnie dwa wyjścia.

Liczba bitów niezbędna do pośredniego adresowania pojedynczego wyjścia generującego sygnał wyjściowy jednocześnie z wyjściem znajdującym się na liście zdarzeń przełączania wynosi

$$F_{CH} = 1 - \text{entier}(-\log_2 N) - \text{entier}(-\log_2 0,2N) \quad /1/$$

gdzie: N - maksymalna liczba elementów modelowanej sieci przełączającej.

W tabeli 2 podano wartości F_{CH} obliczone wg /1/ dla niektórych N.

Tabela 2

$\log_2 N$	9	10	11	12	13	14
F_{CH}	17	19	21	23	25	27

Pośrednie adresowanie wyjść generujących jednocześnie sygnały wyjściowe może być przeprowadzone za pomocą jednego 24-bitowego słowa dla $N \leq 4096$. Do określania wielkości obszaru C_H pamięci rdzeniowej posługiwać się będziemy zależnością

$$C_H = \text{entier}(0,2 e_{CH} \cdot N \cdot M_{sr}) \quad /2/$$

gdzie: e_{CH} - liczba słów niezbędna do pośredniego adresowania pojedynczego wyjścia generującego jednocześnie sygnał wyjściowy

$$(e_{CH} = 1 \text{ dla } F_{CH} \leq 24; e_{CH} = 2 \text{ dla } 24 < F_{CH} \leq 48)$$

N - maksymalna liczba elementów modelowanej sieci przełączającej

M_{gr} - średnia liczba wykorzystanych wyjść elementów modelowanej sieci

W obszarze D_H znajduje się informacja kodująca strukturę modelowanej sieci. Określimy maksymalną wielkość sieci przełączającej jaka może być zakodowana w obszarze D_H pamięci rdzeniowej.

Aby uniknąć konieczności wprowadzania tablic odpowiedniości w programie modelującym muszą być zachowane zasady kodowania biegunów /wejść i wyjść/ elementów stosowanych w systemie automatycznego projektowania. Maksymalna liczba biegunów modelowanej sieci przełączającej wyniesie wobec tego

$$N (L + M) \quad /3/$$

gdzie: N - maksymalna liczba elementów logicznych w sieci

L - maksymalna liczba wejść elementu

M - maksymalna liczba wyjść elementu

Jeżeli elementy przełączające kodowane są według metody podanej w [4], to przy założeniu odrębnego kodowania numeru elementu i jego biegunów liczba pozycji binarnych niezbędna do zakodowania jednego bieguna określona będzie przez wyrażenie

$$F_{DH} = -\text{entier}(-\log_2 N) - \text{entier}[-\log_2(L+M)] - \text{entier}(-\log_2 L) \quad /4/$$

Dla $N = 16384$, $L+M \leq 16$, $L \leq 15$ otrzymujemy $F_{DH} = 22$, czyli do kodowania pojedynczego bieguna przy maksymalnej założonej ilości elementów przełączających wystarcza jedno słowo 24-bitowe.

Określmy średnią liczbę słów niezbędną do kodowania sieci przełączającej zawierającej N elementów. Wartość ta wyznacza zarazem średni obszar pamięci D_{Hsr}

$$D_{Hsr} = \text{entier} \left[N \left(1 + L_{sr} + M_{sr} \right) \right] \quad /5/$$

gdzie: L_{sr} - średnia liczba wykorzystanych wejść elementów sieci przełączającej

M_{sr} - średnia liczba wykorzystanych wyjść elementów sieci przełączającej

Jeżeli modelowana sieć przełączająca może zawierać do 16 384 elementów / $D_{Hsr} = 139\ 264/$, to przechowywanie informacji o jej strukturze w standardowych pamięciach rdzeniowych rozpatrywanych maszyn drugiej i trzeciej generacji /tab. 1/ nie jest możliwe. W przypadku gdy dostępna wielkość pamięci rdzeniowej maszyny jest mniejsza od wymaganej liczby słów dla obszarów $A_H \div D_H$ programu modelującego należy ocenić podstawowe parametry programu modelującego przy umieszczeniu części obszaru D_H w pamięci zewnętrznej.

3.4. Szacowanie czasu modelowania

Rodzaj czynności jakie wykonuje program przy przetwarzaniu sygnałów wewnętrznych w metodzie listowej uzależniony jest od bieżącej zawartości listy zdarzeń przełączania.

Przy badaniu tej listy można wyróżnić trzy przypadki:

- element listy nie zawiera zdarzeń przełączania
- element listy zawiera pojedyncze zdarzenie przełączania /wskaznik jednoczesności $s = 0/$
- element listy zawiera zdarzenie wielokrotne składające się z $s+1$ jednoczesnych zdarzeń przełączania

Omówimy czynności wykonywane przez program w każdym z wymienionych wyżej przypadków. W nawiasach podana będzie nie-

zbędna liczba elementarnych operacji maszyny dla wykonania tych czynności.

A. Wyznaczenie rodzaju zdarzenia w elemencie listy zdarzeń przełączania

1. Porównanie zawartości elementu listy ze wzorem określającym typ zdarzenia /2/.
2. Zwiększenie o jeden /modulo H/ numeru elementu listy /2/.

Czynności p. A wykonywane są dla każdego elementu listy niezależnie od rodzaju znajdującego się w tym elemencie zdarzenia. Szacunkowa liczba operacji części centralnej przy wykonywaniu czynności p. A wynosi

$$t_{AH} = 4 \quad /6/$$

B. Przetwarzanie pojedynczego zdarzenia przełączania

1. Wykonanie czynności p. A /4/
2. Wyznaczenie wartości sygnału na wyjściu elementu przełączającego /1/
3. Badanie oddziaływania sygnału wyjściowego na elementy przełączające dołączone do wyjścia
 - 3.1. wywołanie elementu połączonego z wyjściem / t_{wyw} /
 - 3.2. wykonanie podprogramu elementu / t_{podpr} /
4. Umieszczenie nowego numeru wyjścia w liście zdarzeń przełączania /2/
5. Likwidacja zdarzenia przełączania / t_{od} /

Czynności p. B3 wykonywane są średnio Z_{BR} -krotnie dla każdego wyjścia umieszczonego w liście zdarzeń przełączania. Dla czynności p. B3.1 $t_{wyw} = 0$, w przypadku gdy kod elementu znajduje się w pamięci rdzeniowej. Re-

alizacja p. B5 polega na przesłaniu kodu wyjścia elementu z nową wartością sygnału wyjściowego do miejsca stałego przechowywania. Liczba operacji zużywana na przesyłanie kodu elementu do pamięci zewnętrznej $t_{od} = 0$, w przypadku gdy informacja o elemencie przechowywana jest w pamięci rdzeniowej. Całkowitą liczbę operacji do wykonania czynności p. B wyznacza wobec powyższego szacunkowa zależność

$$t_{BH} = 7 + (t_{wyw} + t_{podpr}) Z_{\acute{s}r} + t_{od} \quad /7/$$

gdzie: $Z_{\acute{s}r}$ - średnie obciążenie wyjść elementów /w umownych jednostkach/

t_{wyw} , t_{podpr} , t_{od} - liczby operacji do wykonania czynności wywołania, realizacji podprogramu i odesłania "elementu"

C. Przetwarzanie wielokrotnego zdarzenia przełączania

1. Wykonanie czynności p. A1 /2/

2. Wykonanie czynności p. B2 ÷ B5

3. Przeadresowanie zdarzenia jednoczesnego, jeżeli nie jest ono ostatnim w danej grupie lub wykonanie czynności p. A2 w przypadku ostatniego zdarzenia jednoczesnego w danej grupie /2/

W interesującym nas przypadku dynamiki stacjonarnej, czynności p. C wykonywane są $s+1$ -krotnie dla każdego elementu listy zdarzeń przełączania ze zdarzeniem wielokrotnym.

Ze względu na konieczność pośredniego adresowania elementów listy zdarzeń przełączania, zawierających zdarzenia jednoczesne, liczba operacji przy wykonywaniu czynności p. B4 jest dla zdarzenia jednoczesnego o jeden większa, co powoduje konieczność wykonania dodatkowo s operacji dla każdego zdarzenia wielokrotnego.

Zależność wyznaczającą liczbę operacji potrzebną do zrealizowania czynności p. C otrzymujemy zakładając ten sam czas

wykonywania operacji przeadresowania zdarzenia jednoczesnego i zwiększenia o jeden /modulo H/ numeru elementu listy zdarzeń przełączania.

$$t_{CH} = t_{BH} (s + 1) + s \quad /8/$$

t_{BH} - liczba operacji maszyny potrzebna do zrealizowania czynności p. B

s - krotność zdarzenia wielokrotnego

Zachowując przyjętą zasadę dynamiki stacjonarnej sieci przełączającej określimy ilość zdarzeń przełączania E_H dla jednego cyklu przeglądania listy zdarzeń przełączania

$$E_H = E_B + E_C (s + 1) \quad /9/$$

Całkowitą liczbę operacji maszyny przy jednym obiegu listy zdarzeń przełączania t_H określa wzór:

$$t_H = (H - E_B - E_C) t_{AH} + E_B t_{BH} + E_C t_{CH} \quad /10/$$

gdzie: H - długość /liczba elementów/ listy zdarzeń przełączania

E_B - liczba pojedynczych zdarzeń przełączania dla jednego obiegu listy zdarzeń przełączania

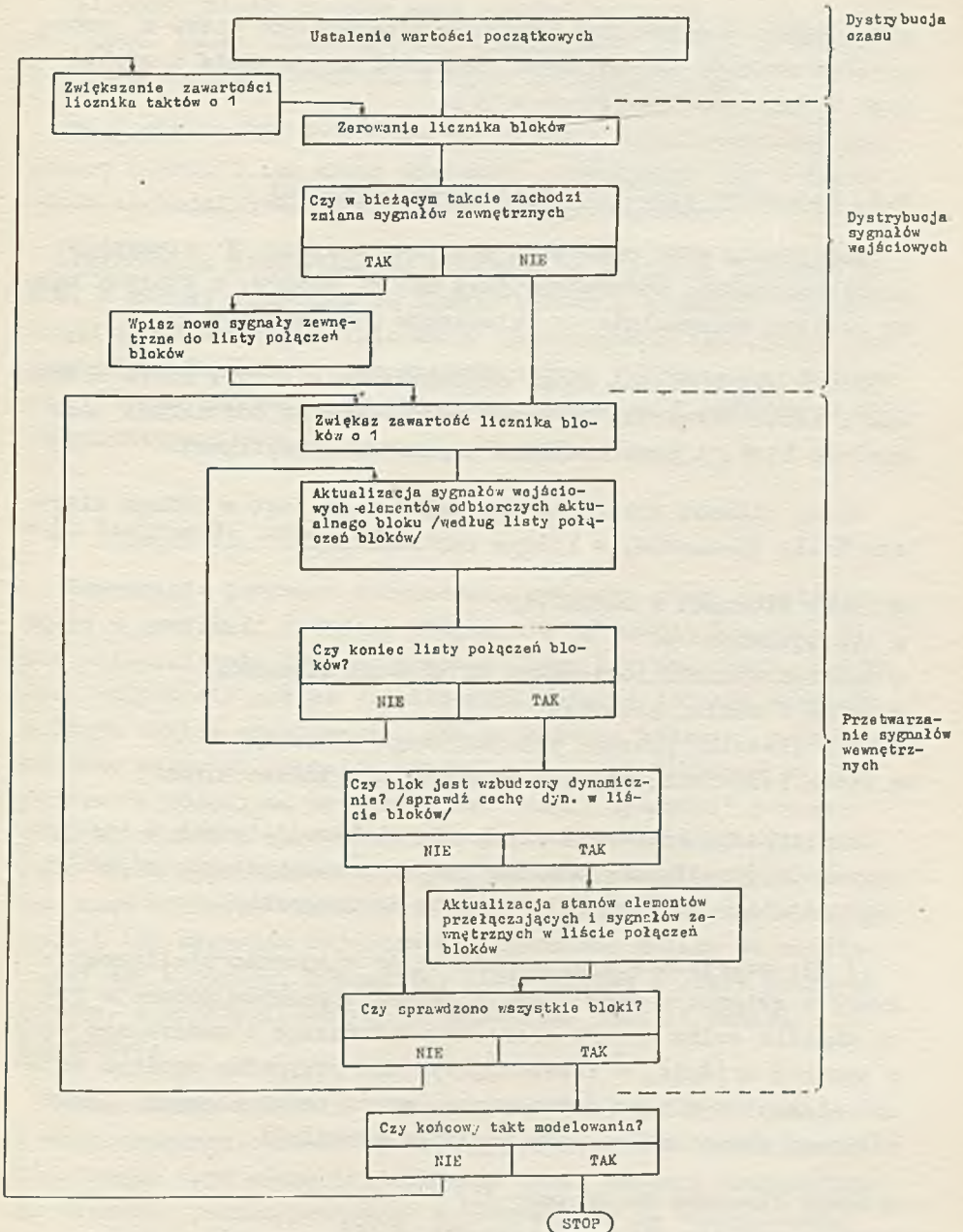
E_C - liczba wielokrotnych zdarzeń przełączania dla jednego obiegu listy zdarzeń przełączania

Średnią liczbę operacji maszyny do wykonania czynności jednego taktu modelowania określać będziemy z zależności

$$t_{H1} = \frac{t_H}{H} \quad /11/$$

4. METODA BLOKOWA

Blokowa metoda modelowania sieci przełączających zastosowana została w [2]. Dalej rozważana będzie ogólna organizacja



Rys. 2. Uproszczony wykres czynności modelowania metodą blokową

modelowania wewnętrznych procesów przełączania sieci z pominięciem szeregu pomocniczych czynności modelowania i niektórych ograniczeń przyjętych w [2].

4.1. Kodowanie struktury sieci przełączającej

Modelowana sieć przełączająca, zawierająca N elementów przełączających, podzielona jest na w bloków, z których każdy zawiera maksymalnie n elementów przełączających.

Strukturę logiczną bloku określają dwie listy: lista elementów i lista połączeń. Listy te wykonywane są oddzielnie dla każdego bloku i przechowywane w pamięci zewnętrznej.

Każdy element przełączający zakodowany jest w jednym wierszu listy elementów, w którym zapisane są:

- numer elementu w bloku /i/
- typ elementu /p/
- wartość sygnału logicznego na wyjściu elementu
- cecha dynamiki elementu
- stan licznika procesu przejściowego elementu
- cecha połączenia wyjścia elementu z wyjściem bloku

W przypadku kodowania elementów wielowyjściowych w liście elementów określa się ponadto numery poszczególnych wyjść i odpowiadające im wartości sygnałów logicznych.

W [2] przyjęto równoprawność wejść elementów przełączających, w związku z czym lista połączeń elementów podana w pracy określa tylko numery elementu odbiorczego i nadawczego wraz z numerem wyjścia. W interesującym nas przypadku ogólnym wejścia elementów nie są równoprawne, wobec czego element listy połączeń wewnątrzblokowych powinien określać:

- numer elementu nadawczego /i/
- numer wyjścia elementu nadawczego /j/
- numer elementu odbiorczego /l/
- numer wejścia elementu odbiorczego /k/

Elementy przełączające mają oddzielną numerację dla każdego bloku. Numer elementu sieci przełączającej składa się wobec tego z numeru elementu w bloku i numeru bloku.

Wszystkie bloki wyliczone są w liście bloków zawierającej numery bloków i ich cechy dynamiki, określające czy w danym bloku zachodzą dynamiczne procesy przełączania.

Połączenia międzyblokowe kodowane są w liście połączeń bloków, w której określone są numery bloków i elementów /nadawczego i odbiorczego/, informacja przekazywana jest przez połączenie aktualnie i w poprzednim takcie modelowania. Informacje o sygnałach wejściowych przechowywane są w dodatkowym niestandardowym bloku.

4.2. Realizacja modelowania

Sterowanie procesem modelowania dynamiki sieci przełączającej w omawianej metodzie odbywa się za pomocą licznika bloków odliczającego kolejne wiersze listy bloków. Bloki, w których odbywa się proces przełączania chociaż jednego elementu w danym takcie modelowania, lub na wejścia których podawane są nowe wartości sygnałów wejściowych identyfikowane są na podstawie różnej od zera wartości "cechy dynamiki" poszczególnych bloków i sprowadzane do pamięci rdzeniowej. Dla każdego bloku sprowadzonego do pamięci rdzeniowej program wyznacza nowe wartości stanów wewnętrznych elementów przełączających i ich sygnałów wyjściowych, wprowadza zmiany do znajdującej się w pamięci rdzeniowej listy bloków i listy połączeń bloków, a następnie odsyła przetworzony blok do pamięci zewnętrznej.

Modelowanie dynamiki przełączania poszczególnych elementów przełączających odbywa się według podprogramów odpowiednich dla danego typu elementu. Liczenie czasu procesu przejściowego elementu jednowyjściowego sprowadza się do zmniejszenia zawartości licznika procesu przejściowego danego elementu.

Operacja ta przeprowadzana jest w każdym takcie modelowania. W chwili gdy zawartość licznika procesu przejściowego elementu staje się równa zero, uważa się proces przełączania tego elementu za zakończony. Dla elementów wielowyjściowych należy przewidzieć w ogólnym przypadku indywidualne liczniki procesów przejściowych przyporządkowane poszczególnym wyjściom.

Skrócenie czasu modelowania w omawianej metodzie osiągane jest dzięki selektywnemu przetwarzaniu tylko tych bloków, w których przebiegają procesy dynamiczne. Pozwala to na zmniejszenie liczby przesłań pomiędzy pamięcią rdzeniową i zewnętrzną.

Uproszczony wykres czynności programu modelowania metodą blokową podano na rys. 2. Analogicznie do metody listowej można wydzielić te same podstawowe grupy czynności realizacji modelowania.

Dystrybucja czasu w metodzie blokowej polega na zmniejszeniu zawartości liczników procesów przejściowych poszczególnych elementów przełączających, w miarę przetwarzania sygnałów dynamicznych elementów znajdujących się w stanie przełączania i zwiększaniu zawartości centralnego licznika taktów modelowania.

Dystrybucja sygnałów zewnętrznych w metodzie blokowej przeprowadzana jest na początku każdego taktu modelowania poprzez wpisanie nowych wartości sygnałów zewnętrznych do znajdującej się w pamięci rdzeniowej tablicy połączeń bloków. Czynności te wykonywane są bez odwoływania się do pamięci zewnętrznej.

4.3. Rozmieszczenie informacji w pamięci rdzeniowej

W pamięci rdzeniowej znajdują się stale:

- Program modelujący wraz z miejscami roboczymi. /obszar A_w /
- Lista bloków /obszar B_w /
- Lista połączeń bloków /obszar C_w /
- Zakodowana struktura modelowanego bloku /obszar D_w /

Podobnie jak w metodzie listowej w obszarze pamięci rdzeniowej A_w umieszczony jest program modelujący z odpowiednimi podprogramami i koniecznymi do jego realizacji miejscami roboczymi. Przyjmujemy, że program modelujący zajmuje nie więcej niż 2000 słów ($A_w \leq 2000$ ^{*)}).

W liście bloków wyszczególnione są kolejne numery bloków i cecha dynamiki określająca czy w danym bloku znajdują się aktualnie elementy, w których odbywa się proces przełączania. Przyjmując, że w liście bloków każdemu blokowi przyporządkowane jest jedno słowo, otrzymujemy $B_w = w$.

Lista połączeń bloków określa połączenia między blokami, informację przekazywaną przez te połączenia aktualnie oraz w poprzednim takcie modelowania. Ze względu na to, że informacja do modelowania otrzymywana jest z podstawowego masywu danych systemu automatycznego projektowania, przyjmujemy, iż przy kodowaniu połączeń została zachowana stosowana w systemie numeracja elementów i ich biegunów. Określimy liczbę pozycji binarnych niezbędną do utworzenia jednostki informacyjnej listy połączeń bloków w zależności od wielkości sieci przełączającej.

Jeżeli sieć przełączająca może zawierać maksymalnie N elementów przełączających o maksymalnej liczbie wejść L i maksymalnie dwu wyjściach, to wymaganą liczbę pozycji binarnych F_{Cwe} do zakodowania pojedynczego wejścia elementu /przy założeniu, $N > 1$, $L > 1$ / określa wyrażenie

$$F_{Cwe} = -\text{entier}(-\log_2 N) - \text{entier}(-\log_2 L) \quad /12/$$

Niezbędną liczbę pozycji binarnych do zakodowania jednego wyjścia elementu /przy założeniu $N > 1$, $M = 2$ / wyznacza zależność

$$F_{Cwy} = -\text{entier}(-\log_2 N) + 1 \quad /13/$$

*) Podobnie jak w p. 33 jest to wartość przeciętna

Wobec powyższego do zakodowania połączenia wymagana będzie następująca liczba pozycji binarnych

$$F_{C_{Ww}} + F_{C_{Wy}} = 1-2 \text{ entier } (-\log_2 N) - \text{entier } (-\log_2 L) \quad /14/$$

Uwzględniając konieczność kodowania dla każdego połączenia bieżącej i poprzedniej wartości przesyłanego sygnału logicznego /2 bity/ otrzymujemy

$$F_{C_w} = 3-2 \text{ entier } (-\log_2 N) - \text{entier } (-\log_2 L) \quad /15/$$

W tabeli 3 podano zestawienie wartości $\log_2 N$ i F_{C_w} obliczonej wg /15/ dla $8 \leq L < 16$. Dodatkowo podano również liczbę słów 8-bitowych /bajtów/ potrzebną do kodowania połączeń między blokami w przypadku maszyny o zmiennej długości słowa.

Tabela 3

$\log_2 N$	8	9	10	11	12	13	14
F_{C_w}	23	25	27	29	31	33	35
Ilość bajtów	3	4	4	4	4	5	5

Z podanej tablicy wynika, że w przypadku $8 \leq L < 16$ i $M = 2$ za pomocą jednego słowa 24 bitowego można kodować połączenia między blokami dla $N \leq 256$. Kodowanie połączeń między blokami dla sieci o większej liczbie elementów wymaga dwu słów maszyny do zakodowania jednego połączenia. Widoczna jest oszczędność kodowania połączeń w przypadku zmiennej długości słowa maszyny.

Minimalna liczba połączeń międzyblokowych zachodzi w przypadku, gdy wszystkie bloki mają połączone tylko dwa bieguny /jedno wejście i jedno wyjście/

$$C_w \text{ min} = e_{C_w} w \quad /16/$$

gdzie e_{C_w} - liczba słów do zakodowania jednego połączenia między blokami

Maksymalna liczba połączeń międzyblokowych ograniczona jest z jednej strony liczbą wyjść elementów i ich dopuszczalną obciążalnością, z drugiej strony wartością L . Jeżeli dodatkowo przyjąć, że z pojedynczego wyjścia bloku do wejść dowolnego innego bloku prowadzone jest maksymalnie jedno połączenie^{*)}, to otrzymujemy warunek

$$C_{w \max 1} = e_{Cw} MN \min [Z, (w-1)] \quad /17/$$

gdzie: e_{Cw} - liczba słów niezbędna do zakodowania jednego połączenia między blokami

M - maksymalna liczba wyjść elementów sieci przełączającej

N - maksymalna liczba elementów przełączających w sieci

Z - dopuszczalna obciążalność wyjść elementów

w - liczba bloków

Drugi warunek, ograniczający liczbę połączeń, otrzymamy uwzględniając maksymalną liczbę wejść elementów L

$$C_{w \max 2} = e_{Cw} NL \quad /18/$$

Łącząc /17/ z /18/ otrzymujemy

$$C_{w \max} = e_{Cw} N \min \left\{ L, M \min [Z, (w-1)] \right\} \quad /19/$$

Obszar $C_{w \max}$ musi być zarezerwowany w pamięci rdzeniowej do kodowania połączeń międzyblokowych, w przypadku gdy połączenia te mogą być prowadzone w sposób dowolny.

W praktyce, w miarę wzrostu wielkości bloków $/n/$ liczba połączeń międzyblokowych staje się coraz mniejsza w stosunku do wartości określonych zależnością /19/. Jeżeli modelowana sieć spełnia podane w p. 2 założenia częściowego uporządkowania struktury połączeń, a równocześnie bloków jest dostatecznie dużo, dzięki czemu można korzystać z wartości Z_{gr} , M_{gr} , L_{gr} , to zależności /17 - 19/ mogą być zapisane w innej postaci, umożliwiającej oszacowanie średniej wielkości obszaru C_w

*) Jeżeli tak nie jest w realnym projekcie, to do celów modelowania informacja może być przekształcona do takiej postaci

$$C_{w\acute{s}r1} = e_{cw} M_{\acute{s}r} n_n w \min [Z_{\acute{s}r}, (w-1)] \quad /17a/$$

gdzie: n_n - liczba nadawczych elementów bloku

$$C_{w\acute{s}r2} = e_{Cw} n_o w L_{\acute{s}r} \quad /18a/$$

gdzie: n_o - liczba odbiorczych elementów bloku.

Zakładając $n_n = n_o = n_z$ otrzymujemy

$$C_{w\acute{s}r} = e_{Cw} n_z w \min \left\{ L_{\acute{s}r}, M_{\acute{s}r} \min [Z_{\acute{s}r}, (w-1)] \right\} \quad /19a/$$

W obszarze D_w pamięci rdzeniowej przechowywana jest zakodowana informacja o strukturze logicznej bloku aktualnie modelowanego. Jak zaznaczono w p. 4.1 informacja ta przechowywana jest w postaci dwu list.

W liście elementów za pomocą jednego bitu mogą być zakodowane: wartość sygnału logicznego na wyjściu, cecha dynamiki elementu i cecha połączenia elementu z wyjściem na zewnątrz bloku. Niezbędną liczbę bitów do zakodowania numeru elementu w bloku, typu elementu i stanu licznika procesu przejściowego określa zależność /20/ otrzymana przy założeniu $n, P, T_p > 1$

$$F_{n,P,T} = - \left[\text{entier}(-\log_2 n) + \text{entier}(-\log_2 P) + \text{entier}(-\log_2 T_p) \right] \quad /20/$$

gdzie: n - maksymalna liczba elementów w jednym bloku

P - ilość typów elementów w sieci przełączającej

T_p - maksymalny czas przełączania elementów sieci przełączającej /liczony w taktach modelowania/

Łączna liczba bitów do zakodowania informacji dla pojedynczego wiersza listy elementów bloku wyniesie wobec tego

$$F_{Dw} = 3 - \left[\text{entier}(-\log_2 n) + \text{entier}(-\log_2 P) + \text{entier}(-\log_2 T_p) \right] \quad /21/$$

Obszar pamięci D_{1e} , który należy zarezerwować dla listy elementów bloku, równać się będzie

$$D_{1e} = e_{Dw} n \quad /22/$$

gdzie: e_{Dw} - liczba słów wymagana do zakodowania jednego elementu w metodzie blokowej

$$/e_{Dw} = 1 \text{ dla } F_{Dw} \leq 24, e_{Dw} = 2 \text{ dla } 24 < F_{Dw} \leq 48/$$

n - maksymalna liczba elementów w modelowanym bloku

W liście połączeń elementów za pomocą jednego bitu kodowany jest numer wyjścia elementu nadającego sygnał /dla $M = 2/$. Niezbędną liczbę bitów do zakodowania jednego połączenia, dla którego określa się ponadto numer elementu nadającego i odbierającego sygnał oraz numer wejścia elementu odbierającego sygnał, wyznacza zależność:

$$F_{Dw1} = 1 - 2 \text{ entier}(-\log_2 n) - \text{entier}(-\log_2 L) \quad /23/$$

gdzie: n - maksymalna liczba elementów w jednym bloku

L - maksymalna liczba wejść elementów sieci przełączającej

Wielkość obszaru D_{1p} zajmowanego przez listę połączeń elementów bloku nie przekroczy wartości

$$D_{1p} = e_{Dw1} n M Z \quad /24/$$

gdzie: e_{Dw1} - liczba słów wymagana do zakodowania jednego połączenia wewnątrz bloku

$$/e_{Dw1} = 1 \text{ dla } F_{Dw1} \leq 24, e_{Dw1} = 2 \text{ dla } 24 < F_{Dw1} \leq 48/$$

n - maksymalna liczba elementów w modelowanym bloku

M - maksymalna liczba wyjść elementów sieci przełączającej

Z - maksymalna obciążalność pojedynczego wyjścia elementu

Górną granicę wielkości obszaru D_w otrzymujemy w rezultacie zsumowania ograniczeń dla D_{1e} i D_{1p}

$$D_{w \max} = n (e_{Dw} + e_{Dw1} M Z) \quad /25/$$

Dla sieci o dużych blokach, dla których można posługiwać się wartościami średnimi $M_{\acute{s}r}$, $Z_{\acute{s}r}$, spełniających ponadto założenia częściowego uporządkowania struktury połączeń, zależności /24/ i /25/ przyjmują nieco inną postać

$$D_{lp\acute{s}r} = e_{Dw1} (n_o + n_w) M_{\acute{s}r} Z_{\acute{s}r} \quad /24a/$$

gdzie: n_o - liczba odbiorczych elementów bloku

n_w - liczba wewnętrznych elementów bloku

$$D_{w\acute{s}r} = e_{Dw} n + e_{Dw1} (n_o + n_w) M_{\acute{s}r} Z_{\acute{s}r} \quad /25a/$$

4.4. Szacowanie czasu modelowania

Podobnie jak przy szacowaniu czasu modelowania metodą listową, również dla metody blokowej uwzględniać będziemy wyłącznie czynności przetwarzania sygnałów wewnętrznych sieci przełączającej. Szacowanie czasu modelowania przeprowadzane będzie dla dużych sieci, w których można posługiwać się wartościami średnimi $M_{\acute{s}r}$, $Z_{\acute{s}r}$, $L_{\acute{s}r}$ przy założeniu częściowego uporządkowania struktury połączeń zgodnie z p. 2.

Dalej podane zostaną podstawowe czynności modelowania metodą blokową. Liczba operacji maszyny niezbędna do wykonania poszczególnych czynności podana jest w nawiasie przy tych czynnościach.

A. Wymiana bloku

1. Przeglądanie listy bloków

1.1. Zwiększenie o jeden kolejnego numeru bloku /1/

1.2. Wyznaczenie wartości cechy dynamiki bloku /2/

2. Przesłanie bloku z pamięci zewnętrznej do pamięci rdzeniowej / t_{wb} /

3. Przesłanie bloku z pamięci rdzeniowej do pamięci zewnętrznej / t_{ob} /

Czynności p. A1 wykonywane są kolejno dla wszystkich bloków sieci; natomiast czynności pp. A2, A3 wykonywane są tylko dla bloków, w których zachodzą procesy przełączania.

Wyznamy liczbę operacji maszyny niezbędnych do wykonania czynności p. A w jednym takcie modelowania dla jednego bloku z różną od zera ceohą dynamiki

$$t_{AWd} = 3 + t_{wb} (n) + t_{ob} (n) \quad /26/$$

gdzie: n - maksymalna liczba elementów przełączających w pojedynczym bloku sieci

Liczbę operacji maszyny konieczną do przesyłania informacji pomiędzy zewnętrzną pamięcią taśmową a pamięcią rdzeniową wyznaczamy zakładając

$$t_{wb} (n) = t_{ob} (n) = t_{pt} (n)$$

$$t_{pt} (n) = r_4 + \frac{D_{w\acute{s}r} (n)}{r_5} 100 \quad /27/$$

gdzie: r_4, r_5 - parametry pamięci taśmowej

$D_{w\acute{s}r}$ - liczba słów niezbędna do zakodowania struktury logicznej bloku /wg 25a/

Łącząc /26/ z /27/ otrzymujemy

$$t_{AWd} = 3+2 \left(r_4 + \frac{D_{w\acute{s}r} (n)}{r_5} 100 \right) \quad /28/$$

Liczbę operacji p. A dla wszystkich bloków modelowanej sieci w jednym takcie modelowania określa zależność

$$t_{AW} = w_d \left[3+2 \left(r_4 + \frac{D_{w\acute{s}r} (n)}{r_5} 100 \right) \right] + 3 (w-w_d) \quad /28a/$$

gdzie: w - liczba bloków modelowanej sieci

w_d - liczba bloków modelowanej sieci wzbudzonych dynamicznie

B. Aktualizacja sygnałów wejściowych bloku

1. Odszukanie połączenia w liście połączeń bloków
 - 1.1. Wydzielenie numeru bloku /2/
 - 1.2. Porównanie zgodności numeru bloku z numerem szukanym /1/
2. Wyznaczenie wartości sygnału wejściowego i jego oddziaływania na element przełączający
 - 2.1. Sprawdzenie cechy dynamiki sygnału /2/
 - 2.2. Wydzielenie numeru elementu i numeru wejścia odbierającego sygnał /3/
 - 2.3. Wykonanie podprogramu elementu odbiorczego wzbudzonego dynamicznie / t_{podpr} /
3. Przeadresowanie numeru wiersza listy połączeń bloków
 - 3.1. Badanie końca listy połączeń bloków /2/
 - 3.2. Zwiększenie kolejnego numeru wiersza listy połączeń bloków /1/

Czynności p. B wykonywane są tylko dla bloków znajdujących się w pamięci rdzeniowej, tj. tych, w których zachodzą dynamiczne procesy przełączania.

Czynności p. B1 i p. B3 wykonywane są dla kolejnych wierszy listy połączeń bloków do znalezienia poszukiwanego wiersza. Przy założeniu nieuporządkowanego rozkładu informacji w liście połączeń bloków, czynności p. B1 i p. B3 należy dla jednego taktu modelowania wszystkich bloków /całej sieci/ wykonać średnio ξ_1 -krotnie, zgodnie z zależnością

$$\xi_1 = \frac{H_{PB} (H_{PB} + 1)}{2} \cdot \frac{w_d}{w} \quad /29/$$

gdzie: H_{PB} - liczba wierszy listy połączeń bloków
 w - liczba bloków sieci przełączającej
 w_d - liczba bloków sieci przełączającej wzbudzonych dynamicznie

Do oszacowania czasu modelowania posługiwać się będziemy średnią wartością H_{PB} wyznaczoną z zależności

$$H_{PB} = w n_o L_{\acute{s}r} \quad /30/$$

gdzie: w - liczba bloków modelowanej sieci
 n_o - liczba odbiorczych elementów bloku
 $L_{\acute{s}r}$ - średnia liczba wykorzystanych wejść elementu

Podstawiając /30/ do /29/ otrzymujemy

$$g_1 = \frac{1}{2} n_o L_{\acute{s}r} w_d (w n_o L_{\acute{s}r} + 1) \quad /31/$$

Sprawdzenie cechy dynamiki /czynność p. B2.1/ przeprowadzane jest dla każdego wejścia elementu odbiorczego modelowanego bloku. Jeżeli blok zawiera n_o elementów odbiorczych, to czynności p. B2.1 dla pojedynczego bloku należy wykonać $n_o L_{\acute{s}r}$ -krotnie.

Jeżeli założyć, że w danym takcie modelowania na wejściach elementu może wystąpić tylko jeden sygnał z cechą dynamiki różną od zera, to czynności p. B2.2 dla wzbudzonych dynamicznie elementów odbiorczych poprawnie działającej sieci logicznej są powtarzane nie częściej niż co T_p taktów modelowania. Wobec tego efektywna liczba operacji maszyny jaką należy wykonać do zrealizowania czynności p. B2.2 w jednym takcie modelowania pojedynczego bloku wyniesie $n_{od} \frac{3}{T_p}$, gdzie n_{od} - liczba wzbudzonych dynamicznie elementów odbiorczych bloku.

Wyznamy średnią liczbę operacji niezbędną do wykonania czynności p.B dla wszystkich bloków w jednym takcie modelowania z pominięciem czynności p.B 2.3, które zostaną rozpatrzone łącznie z czynnościami p. C 2.3.

$$\begin{aligned} t_{BW-BW2.3} &= 6g_1 + 2 n_o L_{\acute{s}r} w_d + n_{od} \frac{3}{T_p} w_d = \\ &= w_d \left\{ n_o L_{\acute{s}r} [3(w n_o L_{\acute{s}r} + 1) + 2] + \frac{3n_{od}}{T_p} \right\} \quad /32/ \end{aligned}$$

C. Aktualizacja sygnałów wewnętrznych i wyjściowych bloku

1. Ustawienie warunków początkowych przeglądania listy elementów
 - 1.1. Zerowanie cechy dynamiki w liście bloków /2/
 - 1.2. Zerowanie licznika wierszy listy elementów /1/
2. Wyznaczenie nowych stanów wewnętrznych i sygnałów wejściowych elementów wewnętrznych i nadawczych bloku
 - 2.1. Przeglądanie listy połączeń elementów /4 operacje dla jednego elementu listy/
 - 2.2. Poszukiwanie wyjścia sterującego w liście elementów bloku i wyznaczenie wartości sygnału wyjściowego /4 operacje dla jednego elementu listy/
 - 2.3. Wykonanie podprogramu elementu wzbudzonego dynamicznie / t_{podpr} /
3. Czynności końcowe przeglądania listy elementów
 - 3.1. Zwiększenie o jeden zawartości licznika wierszy listy elementów /1/
 - 3.2. Badanie końca listy elementów /2/

Czynności p. C realizowane są tylko dla bloków, w których zachodzą dynamiczne procesy przełączania.

Czynności p. C1 wykonywane są jednorazowo, natomiast czynności p. C3 n-krotnie dla każdego bloku przesłanego do pamięci rdzeniowej.

Wszystkie czynności p. C2 są czynnościami wielokrotnymi. Przeglądanie listy połączeń elementów bloku /p. C 2.1/ ma na celu wyznaczenie wartości sygnałów na wejściach elementów wewnętrznych i nadawczych rozpatrywanego bloku. Jeżeli zgodnie z [2] informacja w liście połączeń elementów ma charakter nieuporządkowany, to łączna liczba / ξ_2 / czynności przeglądania listy połączeń elementów pojedynczego bloku zawierającej $(n_w + n_n)$ L_{sr} wierszy wyniesie

$$\varepsilon_2 = \frac{(n_w + n_n) L_{\text{sr}} [(n_w + n_n) L_{\text{sr}} + 1]}{2} \quad /33/$$

gdzie: n_w - liczba wewnętrznych elementów bloku
 n_n - liczba nadawczych elementów bloku
 L_{sr} - średnia liczba wykorzystanych wejść elementów

Łączną liczbę operacji niezbędną do zrealizowania czynności p. C 2.1 wyznaczano przy założeniu, że wykonanie cyklu przeglądania pojedynczego wiersza listy połączeń elementów zajmuje cztery operacje maszyny.

$$t_{C2.1} = 4 \varepsilon_2 \quad /34/$$

Czynności p. C 2.2 wykonywane są dla każdego wykorzystanego wejścia elementu wewnętrznego i nadawczego.

Jeżeli wyjścia elementów odbiorczych i wewnętrznych w blokach obciążone są średnio przez Z_{sr} wejść, to można przyjąć, że wyjścia sterujące poszukiwane są w liście elementów bloku średnio Z_{sr} -krotnie.

Zakładając częściowo uporządkowaną postać listy elementów /wydzielone elementy nadawcze/ wyznaczmy średnią liczbę ε_3 czynności p.C 2.2, jaką należy wykonać dla jednego bloku.

$$\varepsilon_3 = Z_{\text{sr}} \frac{(n_o + n_w) M_{\text{sr}} [(n_o + n_w) M_{\text{sr}} + 1]}{2} \quad /35/$$

gdzie: Z_{sr} - średnie obciążenie pojedynczego wyjścia elementu
 M_{sr} - średnia liczba wyjść elementu
 n_o, n_w - liczba odbiorczych i wewnętrznych elementów bloku

Całkowita liczba operacji niezbędnych do realizacji czynności p. C 2.2 dla jednego bloku wyznaczona została przy założeniu, iż obróbka pojedynczego wiersza listy elementów wymaga czterech operacji

$$t_{CW2.2} = 4 \varepsilon_3$$

/36/

Czynności p. C 2.3 i nie rozpatrywane poprzednio czynności p. B 2.3 wykonywane są tylko dla elementów bloku wzbudzonych dynamicznie. Przy szacowaniu niezbędnej liczby operacji do realizacji tych punktów, przyjmiemy, że pełny podprogram elementu t_{podpr} wykonywany jest co T_p taktów modelowania dla każdego elementu wzbudzonego dynamicznie.

W pozostałych taktach modelowania dla każdego elementu wzbudzonego dynamicznie przeprowadzana jest aktualizacja zawartości licznika procesu przejściowego tego elementu, co zajmuje łącznie 4 operacje. Wobec powyższego dla n_d wzbudzonych dynamicznie elementów bloku średnia liczba operacji w jednym takcie modelowania wyniesie

$$t_{BW2.3+CW2.3} = n_d \frac{(T_p - 1) 4 + t_{podpr}}{T_p} \quad /37/$$

gdzie: n_d - liczba dynamicznie wzbudzonych elementów bloku.

Jeżeli przyjąć, że elementy wzbudzone dynamicznie rozmieszczone są równomiernie we wszystkich blokach modelowanej sieci przełączającej, to wartość n_d może być wyznaczona z zależności

$$n_d = \frac{N_d}{w} \quad /38/$$

Wartość N_d można natomiast wyznaczyć z /47/

$$N_d = h_2 \cdot N \cdot T_p \quad /39/$$

Przy nierównomiernym rozmieszczeniu elementów wzbudzonych dynamicznie n_d znajduje się w przedziale

$$0 \leq n_d \leq n \quad /40/$$

Łączną liczbę operacji niezbędną do wykonania czynności p. C wraz z p. B 2.3 dla całej sieci w jednym takcie modelowania wyznaczać będziemy z wzoru

$$t_{CW+BW2.3} = w_d \left\{ 3(n+1) + 4(\xi_2 + \xi_3) + n_d \frac{(T_p - 1)4 + t_{\text{podpr}}}{T_p} \right\} \quad /41/$$

gdzie: w_d - liczba bloków sieci przełączającej wzbudzonych dynamicznie

Średnią liczbę operacji maszyny $/t_w/$ do realizacji jednego taktu modelowania całej sieci przełączającej metodą blokową /dla $w_d > 1/$ otrzymujemy w wyniku zsumowania prawych stron zależności /28a/, /32/, /41/

$$t_w = w_d \left\{ 3(n+2) + 2 \left(r_3 + \frac{D_{w_{sr}}}{r_4} 100 \right) + n_o L_{sr} [3(w_n L_{sr} + 1) + 2] + 4(\xi_2 + \xi_3) + \frac{1}{T_p} [3n_{od} + 4n_d (T_p - 1) + n_d t_{\text{podpr}}] \right\} + 3(w - w_d) \quad /42/$$

5. OCENA PORÓWNAWCZA

Dynamiczną aktywność przełączania sieci charakteryzować będziemy jednostkową aktywnością przełączania h wyznaczaną z zależności

$$h = \frac{G}{M_{sr} N_{rz} U} \quad /43/$$

gdzie: G - liczba zdarzeń przełączania w modelowanej sieci
 U - liczba taktów modelowania
 N_{rz} - rzeczywista liczba elementów przełączających w modelowanej sieci

M_{sr} - średnia liczba wyjść elementu przełączającego modelowanej sieci

Dla metody listowej zależność /43/ może być zapisana w postaci

$$h = \frac{E_H}{M_{\text{sr}} N_{\text{rz}} H} \quad /44/$$

gdzie: E_H - liczba zdarzeń przełączania w modelowanej sieci dla jednego obiegu listy zdarzeń przełączania
 H - liczba elementów listy zdarzeń przełączania
 $M_{\text{sr}}, N_{\text{rz}}$ - jak w /43/

Wartość jednostkowej aktywności przełączania sieci przełączającej, zbudowanej z jednorodnych elementów przełączających, nie może być większa niż

$$h_{\text{max}} = \frac{1}{T_p} \quad /45/$$

gdzie: T_p - czas trwania przejściowego procesu przełączania elementów przełączających liczony w taktach modelowania

Jednostkowa aktywność przełączania ma wartość maksymalną $h = h_{\text{max}}$, w przypadku gdy wszystkie elementy sieci przełączającej są wzbudzone dynamicznie $/N_{\text{rz}} = N_d/$. Liczba elementów sieci wzbudzonych dynamicznie może być wyznaczona z /43/ po podstawieniu /45/ i zamianie oznaczeń $h = h_{\text{max}}, N_{\text{rz}} = N_d$.

$$N_d = \frac{G T_p}{M_{\text{sr}} U} \quad /46/$$

Inną zależność dla N_d otrzymuje się w wyniku podstawienia do /46/ wartości G wyznaczonej z /43/

$$N_d = h N_{\text{rz}} T_p \quad /47/$$

Z /47/ wynika, iż dla danej sieci liczba elementów przełączających wzbudzonych dynamicznie N_d jest wprost proporcjonalna do jednostkowej aktywności przełączania h .

Ocenę efektywności modelowania przeprowadzać będziemy za pomocą tzw. jednostkowej oszczędności modelowania θ , zdefiniowanej następująco

$$\theta = \frac{t_m}{G(t_m)} \quad /48/$$

gdzie: t_m - czas modelowania /liczba operacji maszyny/
 $G(t_m)$ - liczba zdarzeń przełączania, jaka miała miejsce w modelowanej sieci w czasie modelowania t_m

Otrzymane w artykule zależności dają możliwość wykonania ilościowych oszacowań podstawowych wskaźników modelowania wewnętrznych procesów przełączania sieci w funkcji parametrów:

$$t_{\text{podpr}}, M_{\text{śr}}, Z_{\text{śr}}, L_{\text{śr}}, T_p, h, (N_d, n_d), r_0 \div r_5, \gamma, N, n, (n_o, n_w, n_n)$$

Przy porównywaniu różnych metod modelowania obliczenia odpowiadających im jednostkowych oszczędności modelowania należy przeprowadzać dla sieci przełączających o jednakowych wartościach parametrów $L_{\text{śr}}, M_{\text{śr}}, Z_{\text{śr}}$ i jednakowej aktywności przełączania h /lub N_d /. Do uzyskania pełniejszej oceny badanej metody modelowania, obliczenia należy przeprowadzać dla zbiorów parametrów określających realne warunki modelowania.

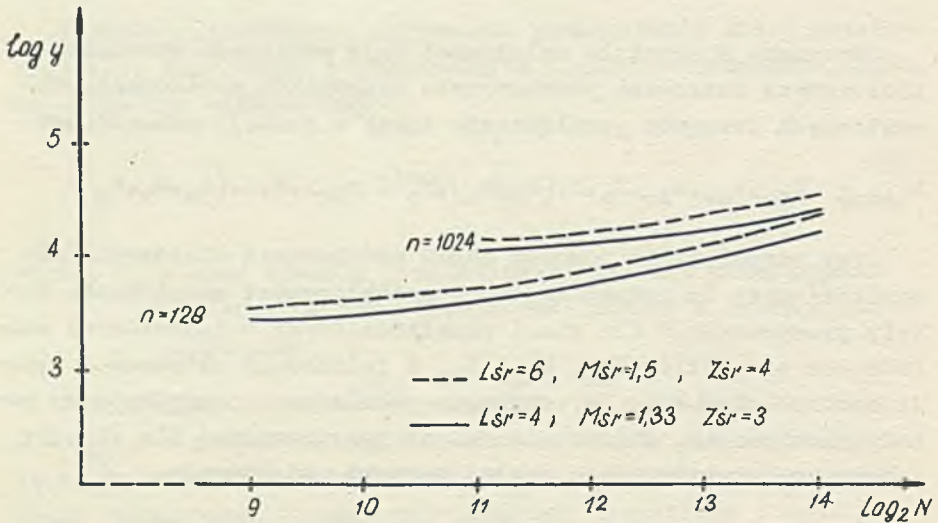
Niżej podano wybrane wyniki oszacowań podstawowych wskaźników obydwu rozpatrywanych metod modelowania, przeprowadzonych dla dwu typowych sieci przełączających części centralnej EMC III generacji

- sieć I o parametrach $Z_{\text{śr}} = 3, M_{\text{śr}} = 1,33, L_{\text{śr}} = 4$
- sieć II o parametrach $Z_{\text{śr}} = 4, M_{\text{śr}} = 1,5, L_{\text{śr}} = 6$

Obliczenia wykonywano przy założeniu $N_d = 0,1 N, t_{\text{podpr}} = 10$ dla modelowania na wzorcowych EMC II i III generacji wyposażonych w pamięć rdzeniową o pojemności 16K i taśmy magne-

tyczne o parametrach określonych w tabeli 1. Wykresy otrzymanych wyników wykonane zostały w skali logarytmicznej.

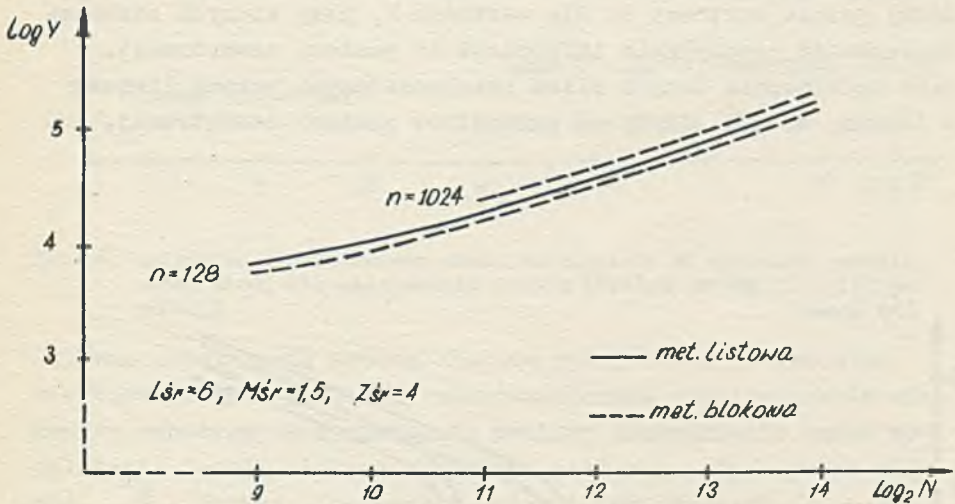
Rysunek 3 ilustruje zależność wymaganej do modelowania metodą blokową wielkości pamięci rdzeniowej y /maszyny modelującej od liczby elementów sieci przełączającej N / dla dwu różnych sieci przełączających i dla dwu wielkości bloków. Wykresy wykonano dla przypadku równomiernego rozmieszczenia elementów wzbudzonych dynamicznie w poszczególnych blokach modelowanej sieci.



Rys. 3. Zależność wymaganej wielkości pamięci rdzeniowej od liczby elementów sieci przełączającej dla metody blokowej

Na rys. 4 pokazano zależność wymaganej do modelowania całkowitej wielkości pamięci Y /łącznie zewnętrznej i rdzeniowej/ od liczby elementów modelowanej sieci N dla listowej i blokowej metody modelowania. Dla metody listowej założono, że program modelujący wraz z miejscami roboczymi i listą zdarzeń przełączania nie przekracza 2150 słów, natomiast wielkości ob-

szarów dla listy wyjść pośrednio adresowanych i zakodowanej struktury modelowanej sieci szacowano według /2/ i /5/. W przypadku metody blokowej obliczenia wykonano dla dwu wielkości bloków / $n = 128$ i $n = 1024$ /. Wielkość obszaru, jaką zajmuje program modelujący, równa się 2000 słów, wielkość obszaru zajmowanego przez listę połączeń bloków wyznaczono według /19a/, a wielkość obszaru przeznaczoną dla zakodowanej struktury modelowanych bloków - według /25a/.

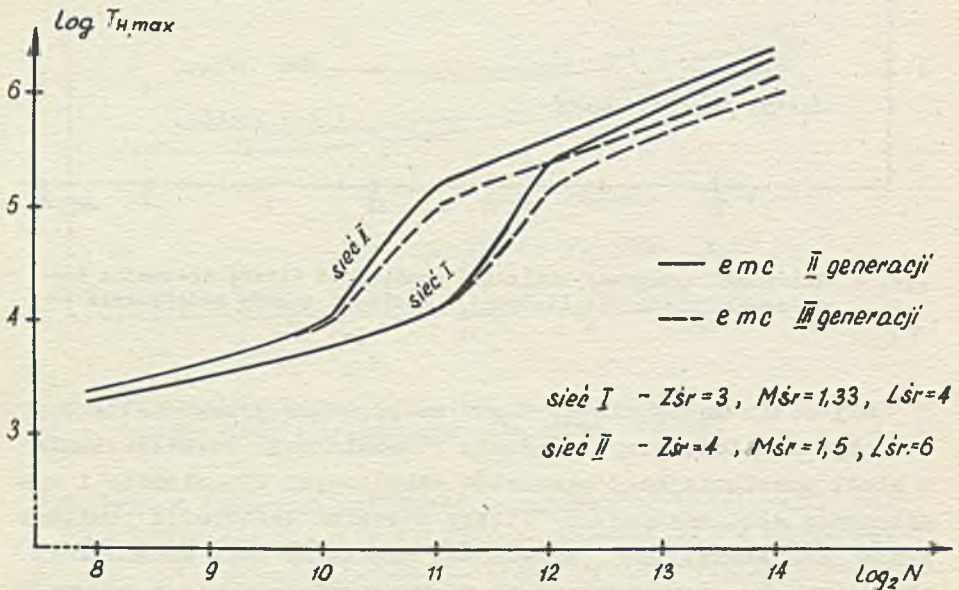


Rys. 4. Zależność wymaganej wielkości pamięci od liczby elementów modelowanej sieci dla listowej i blokowej metody modelowania

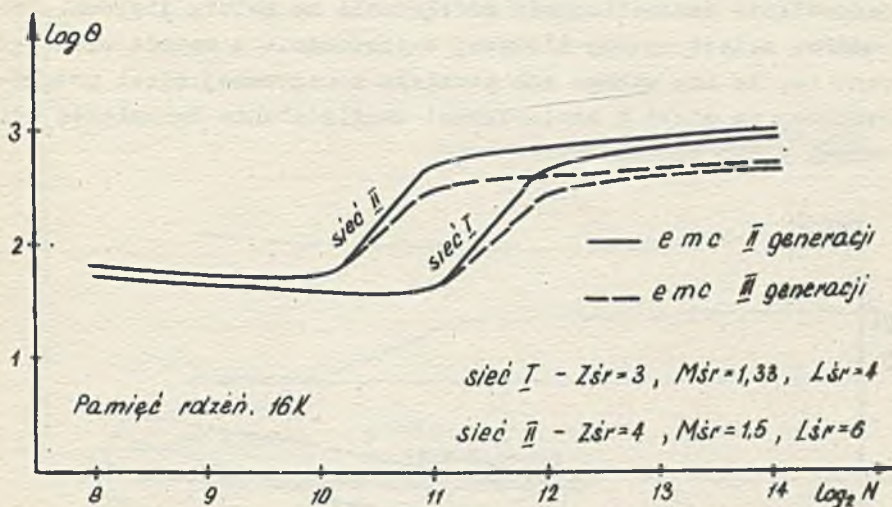
Obliczenia maksymalnego czasu modelowania przeprowadzono dla obydwu metod, przy założeniu równomiernego rozmieszczenia w sieci przełączającej elementów wzbudzonych dynamicznie i wynikającej stąd maksymalnej liczby przesłań informacji pomiędzy pamięciami zewnętrzną i rdzeniową. Minimalny czas modelowania określano dla nierównomiernego rozmieszczenia w sieci elementów wzbudzonych dynamicznie, dla którego liczba przesłań informacji pomiędzy pamięciami jest minimalna. Dla metody blokowej przyjmowano w związku z tym odpowiednio maksymalną i minimalną liczbę bloków wzbudzonych dynamicznie.

Maksymalny czas modelowania metodą listową wyznaczano dla zdarzeń wielokrotnych o maksymalnej dla danej sieci krotności s /przy określonej wartości h /, natomiast minimalny czas modelowania dla maksymalnej liczby zdarzeń pojedynczych.

Zależność maksymalnej liczby operacji maszyny od wielkości modelowanej sieci N dla modelowania metodą listową przeprowadzanego za pomocą wzorcowych EMC II i III generacji pokazano na rys. 5, natomiast na rys. 6 przytoczono odpowiednie wykresy dla jednostkowego czasu modelowania θ . Widoczny jest istotny wzrost wartości θ dla wartości N , przy których zachodzi konieczność przesyłania informacji do pamięci zewnętrznej. Czas modelowania jużych sieci przełączających metodą listową w istotny sposób zależy od parametrów pamięci zewnętrznej.



Rys. 5. Zależność maksymalnej liczby operacji od wielkości modelowanej sieci dla modelowania metodą listową na EMC II i III generacji



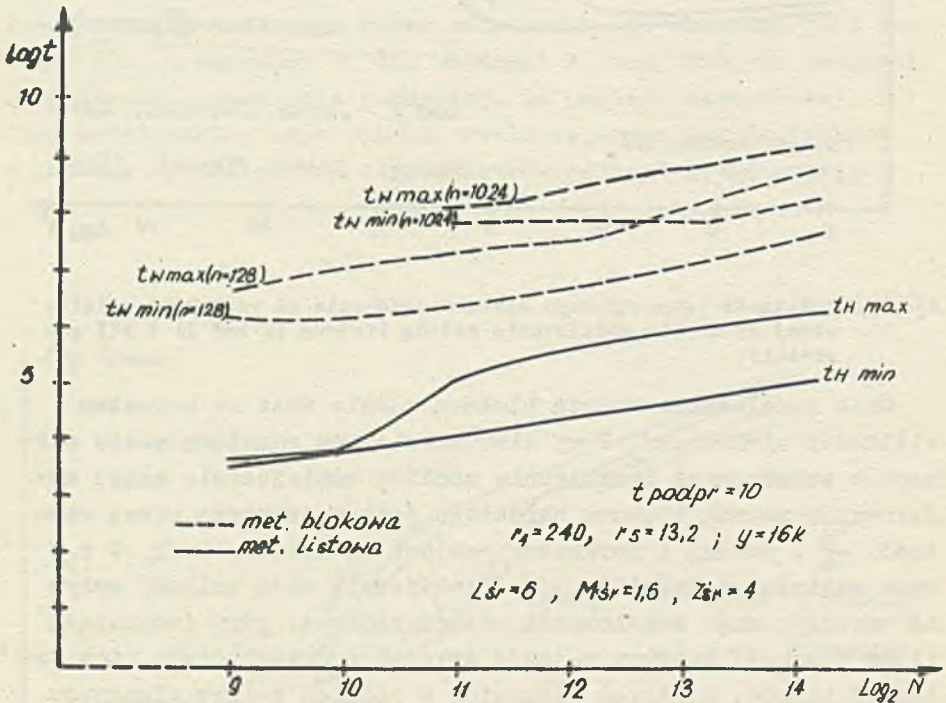
Rys. 6. Zależność jednostkowego czasu modelowania od wielkości modelowanej sieci dla modelowania metodą listową na EMC II i III generacji

Czas modelowania metodą blokową rośnie wraz ze wzrostem wielkości bloków $/n/$. Przy nierównomiernym rozmieszczeniu elementów wzbudzonych dynamicznie możliwe zmniejszenie czasu modelowania metodą blokową określone jest w praktyce przez wartość $\frac{d}{w}$. Rodzaj i parametry pamięci zewnętrznych $/r_0 \div r_5/$ oraz wahania wartości t_{podpr} praktycznie mają znikomy wpływ na wartość czasu modelowania metodą blokową, gdyż dominującą ilość operacji maszyny zajmują czynności przeglądania list połączeń bloków, połączeń elementów w blokach i list elementów bloków.

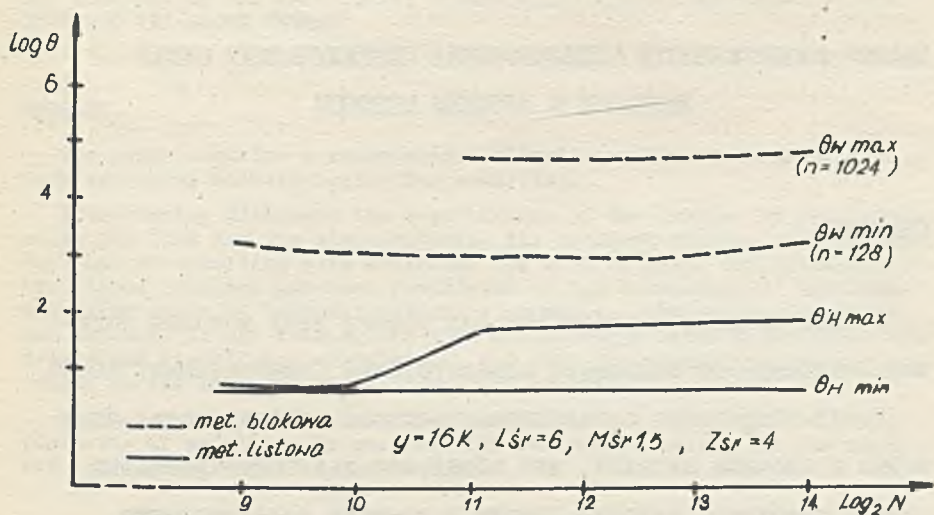
Porównawcze wykresy zależności maksymalnej i minimalnej liczby operacji modelowania od wielkości $/N/$ sieci przełączających /typu I/ dla obu porównywanych metod podano na rys. 7. Widoczna jest znacznie wyższa czasochłonność modelowania metodą blokową.

Rys. 8 ilustruje przebieg analogicznej zależności dla maksymalnego i minimalnego jednostkowego czasu modelowania θ . Nie-

zależnie od wielkości modelowanej sieci N znacznie mniejszą jednostkową czasochłonność modelowania ma metoda listowa. Dodatkową zaletą metody listowej w porównaniu z metodą blokową jest to, że nie wymaga ona podziału modelowanej sieci przełączającej na bloki i konieczności uwzględniania ograniczeń związanych z tym podziałem.



Rys. 7. Zależność maksymalnej i minimalnej liczby operacji modelowania od wielkości modelowanej sieci dla metody listowej i blokowej



Rys. 8. Zależność maksymalnego i minimalnego jednostkowego czasu modelowania od wielkości modelowanej sieci dla metody listowej i blokowej

Literatura

- [1] CASE P.W., GRAFF H.H., GRIFFITH L.E., LE CLERQ A.R., MURBEY W.B., SPENCE T.M.: Solid Logic Design Automation, IBM, J. Res. and Dev. April 1964
- [2] OLEFIR A.K.: Modelirovanije cifrovych sistem, Sbornik, Vychislitelnyje sistemy, 1966:25, 75-84
- [3] ULRICH E.G.: Time Sequenced Logical Simulation Based on Circuit Delay and Selective Tracing of Active Network Paths, Proc. of the 20th National Conference, 1965: 437-448
- [4] ULRICH E.G.: Exclusive Simulation of Activity in Digital Networks, Communications of the ACM, February 1969: 12, 2
- [5] REISS R.F.: The Digital Simulation of Neuro-Muscular Organisms, Behavioral Science, October 1960: 343-58
- [6] Podstawowy język symboliczny - ZAM 41, Oprogramowanie, Instytut Maszyn Matematycznych - Warszawa

АНАЛИЗ ЭФФЕКТИВНОСТИ МОДЕЛИРОВАНИЯ ПЕРЕКЛЮЧАЮЩИХ СЕТЕЙ ЛИСТОВЫМ И БЛОЧНЫМ МЕТОДОМ

Резюме

Работа заключает сравнительную оценку двух методов логическо-временного бинарного моделирования переключающих сетей.

После обсуждения организации процесса моделирования листовым и блочным методом, для обоих методов была проведена оценка требуемой емкости памяти и времени моделирования.

Объем рассматриваемых проблем был ограничен к оценке основного цикла моделирования внутренних процессов переключения в моделированной сети. В оценке принято, что моделированная сеть построена согласно с определенными упрощающими предположениями, а моделирование проводится на так называемых "образцовых" ЦВМ II и III генерации.

Полученные зависимости позволяют провести числовую оценку эффективности моделирования для разных параметров моделированной сети и моделирующих машин.

THE ANALYSIS OF THE EFFECTIVITY OF SWITCHING NETWORK MODELLING BY THE LIST AND THE BLOCK METHOD

Summary

The paper contains a comparative estimation of two methods of binary switching network logic-time modelling.

After having discussed the organization of the problem of modelling using the list and the block methods, the required storage capacity and the time of modelling were estimated for both methods. The scope of the considered problems has been restricted to the estimation of the basic modelling cycle of internal switching processes occurring in the modelled network. It has been agreed that the modelled network satisfies the determined simplifying assumptions, and that the modelling would be performed on the standard DC of II and III generations.

Due to the dependences obtained a quantitative estimation of the effectivity of modelling is possible for various parameters of the modelled network and modelling machines.

ZASTOSOWANIE EMC W PEWNYCH ZAGADNIENIACH
PROJEKTOWANIA OKABLOWANIA SYSTEMÓW CYFROWYCH
III GENERACJI

Wojciech KUBERA
Pracę złożono 5.09.1970

Opisano dwa proste modele linii długich: model transmisji sygnałów w linii długiej przy nieliniowych warunkach brzegowych i model procesów zachodzących w parze linii sprzężonych /model powstawania przeniików/. Wskazano sposób realizacji opartych na tych modelach programów dla EMC, przydatnych do oceny amplitudy i czasu trwania zakłóceń, powstających w okablowaniu szybkich bloków cyfrowych, zbudowanych na układach scalonych.

1. WSTĘP

Charakterystyczną cechą systemów cyfrowych III generacji jest duża szybkość działania /czasy narastania od kilku do kilkunastu nanosekund/. Stwarza to równocześnie szereg nowych trudności w projektowaniu tych systemów i wymaga uwzględnienia takich aspektów działania układów, które były mało istotne, a nawet pomijane w wolnych systemach tradycyjnych. Między innymi konieczne jest rozpatrywanie przewodów łączących jako obwodów o parametrach rozłożonych.

Z drugiej strony, przy stosowaniu w nowoczesnych systemach układów o coraz większym stopniu scalenia, punkt ciężkości prac projektowych w znacznym stopniu przesuwa się na okablowa-

nie. Wadliwie zaprojektowane okablowanie może stać się przyczyną silnych zakłóceń, uniemożliwiających prawidłową pracę systemu, a nawet może powodować uszkodzenia układów. Dlatego też konieczne jest opracowanie odpowiednich wytycznych dla projektantów, pozwalających na określenie zasad okablowania systemu.

Eksperymentalne badania okablowania są szczególnie pracochłonne oraz kosztowne, i dlatego, celowe wydaje się uzyskanie potrzebnych wyników drogą modelowania matematycznego zjawisk zachodzących w okablowaniu. Ze względu na złożoność modeli okablowania, konieczność analizowania dużej liczby wariantów, a także celem zmniejszenia możliwości występowania błędów w obliczeniach, pożądana jest realizacja programowa wspomnianych modeli na EMC.

W pracy niniejszej opisane są dwa proste modele linii długich: model transmisji sygnałów w linii długiej zakończonej opornościami nieliniowymi i model procesów zachodzących w dwóch liniach sprzężonych /model powstawania przeników/. Modele te są łatwe do realizacji programowej i zdaniem autora mogą być pomocne w projektowaniu okablowania nowoczesnych systemów. Dla większej przejrzystości, przytaczane w tekście artykułu szkice programów napisane są w języku ALGOL-60, co z pewnością ułatwi ich ewentualne wykorzystanie.

2. OBLICZANIE PRZEBIEGÓW NAPIĘCIA W DWÓCH SPRZĘŻONYCH LINIACH DŁUGICH OBCIĄŻONYCH OPORNOŚCIAMI LINIOWYMI

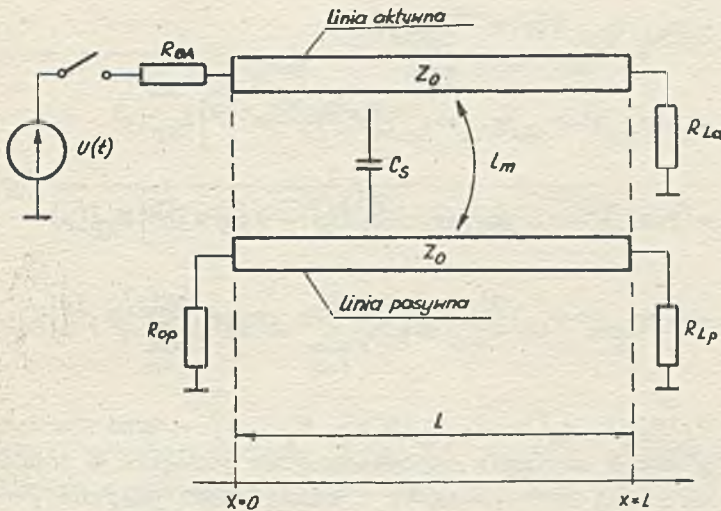
2.1. Zależności podstawowe

W niniejszym punkcie będziemy rozważać układ dwóch sprzężonych linii długich przedstawiony na rys. 1.

Przyjęto następujące założenia:

- warunki początkowe są zerowe,
- obie linie są jednakowe i jednorodne,

- straty w obu liniach mogą być pominięte,
- oddziaływanie linii pasywnej na aktywną sprowadza się wyłącznie do zmiany jej parametrów falowych,
- amplitudy przebiegów w linii pasywnej są na tyle małe, że indukowane przez nie przebiegi w linii aktywnej mają amplitudy znacznie mniejsze niż przebiegi wymuszone przez źródło $U(t)$ /rys. 1/,
- wszystkie oporności są rzeczywiste i liniowe.



Rys. 1. Rozpatrywany układ dwóch sprzężonych linii długich

Szczegółową analizę takiego układu można znaleźć w pracy [1]. Jej wynikiem są następujące wyrażenia operatorowe dla przebiegów napięcia na końcach obu linii /przyjęto, że jednostka czasu równa jest czasowi przejścia sygnału przez linię/:

$$U_A(0,p) = \left[\frac{1 - r_{OA}}{2} U(p) + r_{LA} U(p) e^{-2p} \right] K_A(p) \quad /1.1/$$

$$U_A(L,p) = \frac{(1 - r_{OA})(1 + r_{LA})}{2} [U(p)e^{-p}] K_A(p) \quad /1.2/$$

$$U_p(O,p) = U_F(O,p) + U_D(O,p)$$

$$U_p(L,p) = U_F(L,p) + U_D(L,p)$$

$$U_F(O,p) = (1+r_{op})(1-r_{OA})\frac{\Theta}{8} [U(p) - (1-r_{LA} r_{Lp})U(p)e^{-2p} - r_{LA} r_{Lp} U(p)e^{-4p}] K_{AP}(p) \quad /1.3/$$

$$U_D(O,p) = (1+r_{op})(1-r_{OA})(r_{LA}+r_{Lp})\frac{\Theta}{4} \left[\mathcal{L} \left(\frac{du(t)}{dt} \right) e^{-2p} \right] K_{AP}(p) \quad /1.4/$$

$$U_F(L,p) = (1+r_{Lp})(1-r_{OA})(r_{LA}+r_{op})\frac{\Theta}{8} [U(p) - U(p)e^{-2p}] K_{AP}(p) \quad /1.5/$$

$$U_D(L,p) = (1+r_{Lp})(1-r_{OA})\frac{\Theta}{4} \left[\mathcal{L} \left(\frac{du(t)}{dt} \right) e^{-p} + r_{LA} r_{op} \mathcal{L} \left(\frac{du(t)}{dt} \right) e^{-3p} \right] K_{AP}(p) \quad /1.6/$$

gdzie: $U_A(O,p)$, $U_A(L,p)$, $U_p(O,p)$, $U_p(L,p)$ - transformaty Laplace'a napięcia odpowiednio na początku i końcu linii aktywnej oraz początku i końcu linii pasywnej;

$U(p)$ - transformata Laplace'a napięcia wymuszającego $u(t)$;

L - długość linii;

r_{OA} , r_{LA} , r_{op} , r_{Lp} - współczynniki odbicia /niedopasowania/ określone zależnościami /oznaczenia jak na rys.1/:

$$\left. \begin{aligned} r_{OA} &= \frac{R_{OA} - Z_0}{R_{OA} + Z_0} & r_{op} &= \frac{R_{op} - Z_0}{R_{op} + Z_0} \\ r_{LA} &= \frac{R_{LA} - Z_0}{R_{LA} + Z_0} & r_{Lp} &= \frac{R_{Lp} - Z_0}{R_{Lp} + Z_0} \end{aligned} \right\} \quad /2/$$

$\theta, \bar{\theta}$ - współczynniki sprzężenia, określone zależnościami:

$$\bar{\theta} = \frac{lm}{l} + \frac{C_S}{C}; \quad \theta = \frac{lm}{l} - \frac{C_S}{C} \quad /3/$$

w których: lm - indukcyjność wzajemna między liniami,
 l - indukcyjność własna,
 C_S - pojemność sprzęgająca,
 C - suma pojemności własnej i sprzęgającej.

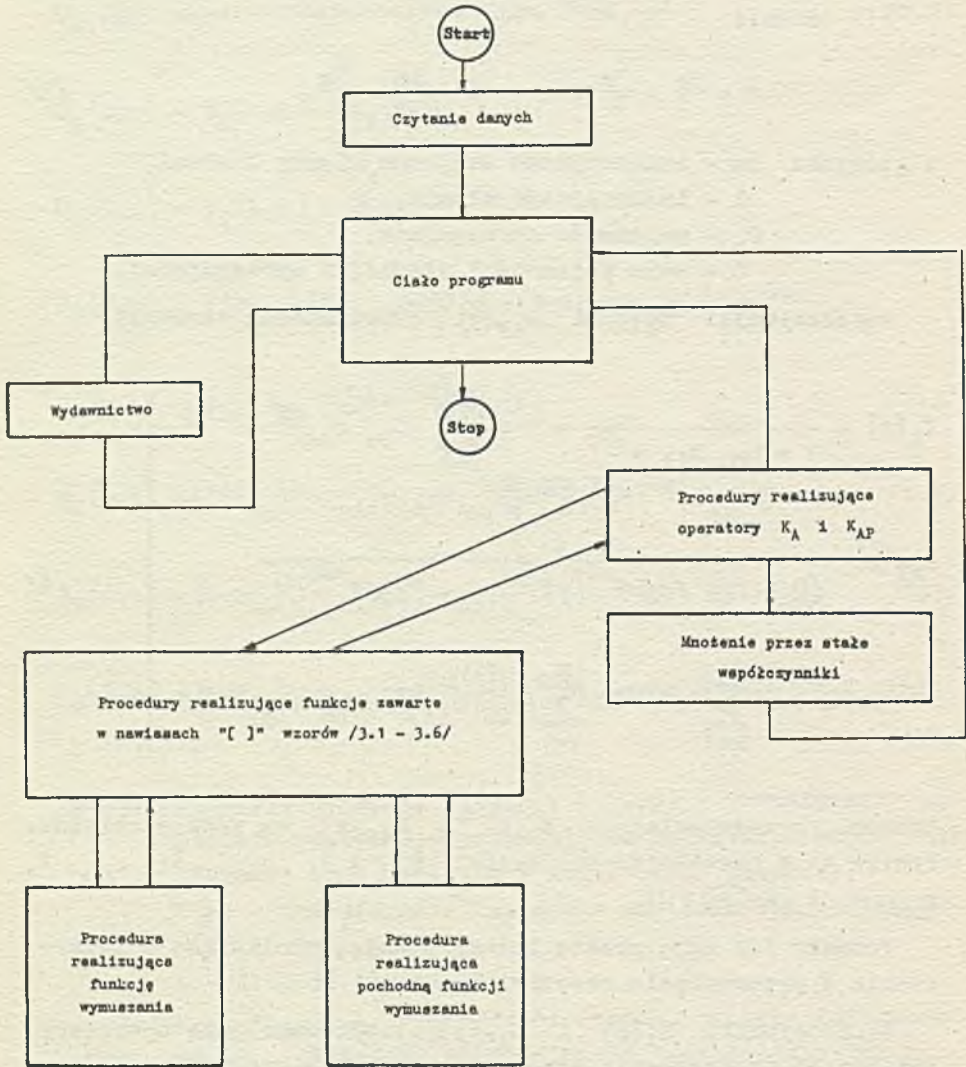
Współczynniki $K_A(p)$ i $K_{AP}(p)$ określone są wzorami:

$$\left. \begin{aligned} K_A(p) &= \frac{1}{1 - r_{OA} r_{LA} e^{-2p}} = 1 + \sum_{k=1}^{\infty} (r_{OA} r_{LA})^k e^{-2kp} \\ K_{AP}(p) &= \frac{1}{(1 - r_{OA} r_{LA} e^{-2p})(1 - r_{Op} r_{Lp} e^{-2p})} = \\ &= 1 + \sum_{m=1}^{\infty} e^{-2mp} \sum_{k=1}^m (r_{OA} r_{LA})^k (r_{Op} r_{Lp})^{m-k} \end{aligned} \right\} /4/$$

Szeregi, przedstawiające $K_A(p)$ i $K_{AP}(p)$ są zawsze zbieżne. Wynika to z następujących faktów: $r_{OA} < 1$, $r_{LA} < 1$, $r_{Op} < 1$, $r_{Lp} < 1$ i $\text{Re}[p] > 0$.

Formuły /4/ mają prostą interpretację, wynikającą z twierdzenia o przesunięciu rzeczywistym.

Współczynniki $K_A(p)$ i $K_{AP}(p)$ traktowane jako operatory działające na wyrażenia zawarte w nawiasach kwadratowych formuł /1/, powodują przesuwanie /opóźnianie/ tych wyrażen, mnożenie przez stałe współczynniki, a następnie zsumowanie otrzymanych składników. Działanie operatorów $K_A(p)$ i $K_{AP}(p)$ odpowiada "składaniu" przebiegu napięcia w linii z fal padających i odbitych.



Rys. 2. Ilustracja struktury programu modelującego

Wyrażenia zawarte w nawiasach kwadratowych formuł /1/ są kombinacjami poprzesuowanych funkcji: wymuszającej $U(t)$ lub jej pochodnej.

Z powyższych uwag wynika struktura programu modelującego, pokazana na rys. 2.

2.2. Opis procedur standardowych

2.2.1. Procedury realizujące funkcję wymuszającą i jej pochodne

Można przyjąć, że funkcja wymuszająca i jej pochodna jest zapisana w formie

$$f(t, X) = g(t) \cdot 1(t - X)$$

Oznacza to, że oprócz parametrów bliżej określających charakter zmienności funkcji i parametru czasowego, wprowadzone zostaje przesunięcie X .

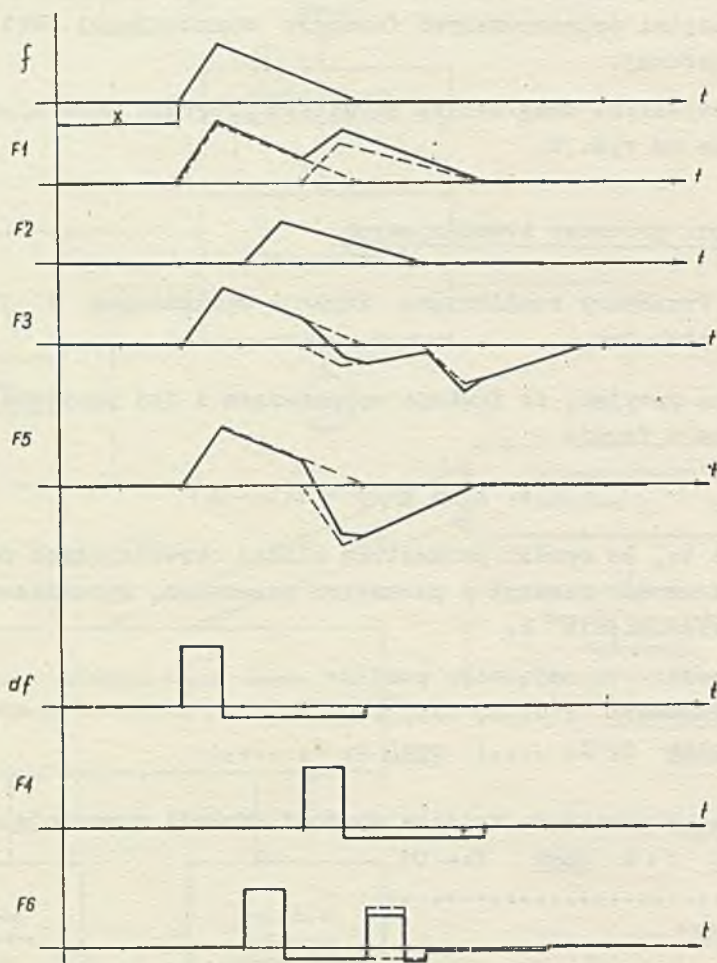
Procedury te mają więc postać:

```
real procedure f(t, X, ...);  
  value t, X, ...; real t, X, ...;  
begin  
  comment procedura wylicza wartość funkcji wymuszającej;  
  if t < X then f := 0;  
  .....  
  end
```

Procedura df wyliczająca wartość pochodnej funkcji wymuszającej ma postać analogiczną.

2.2.2. Procedury realizujące oryginały wyrażeń zawartych w nawiasach kwadratowych formuł 1.1. - 1.6.

Procedury te będziemy oznaczali: F1, F2, ..., F6, /gdzie cyfra w oznaczeniu związana jest z numerem wzorów p. 2.1 /1.1-1.6/.



Rys. 3. Ilustracja współzależności funkcji f , df i $F1 \div F6$

Współzależności między funkcjami f , df i $F1 \div F6$ ilustruje rys. 3. Procedury $F1 \div F6$ przedstawiają się następująco:

```

real procedure F1 (t, X,...); value t, X,..., real t, X,...;
F1:= f(t, X,...) + rLA x f(t, X + 2,...);
real procedure F2(t, X,...); value t, X,...; real t, X,...;
F2:= f(t, X + 1, ...);
real procedure F3(t, X,...); value t, X,...; real t, X,...;
F3:= f(t, X,...) - (1 - rLA x rLP) x f(t, X + 2,...);
real procedure F4(t, X,...); value t, X,...; real t, X,...;
F4:= df(t, X + 2,...);
real procedure F5(t, X,...); value t, X,...; real t, X,...;
F5:= f(t, X + 1,...) - f(t, X + 3,...);
real procedure F6(t, X,...); value t, X,...; real t, X,...;
F6:= df(t, X + 1,...) + rLA x rOP x df(t, X + 3,...);
    
```

2.2.3. Procedury realizujące operatory $K_A(p)$ i $K_{AP}(p)$

Ponieważ funkcja wymuszająca $U(t)$ spełnia warunek:

$$U(t) = 0 \quad \text{dla } t < 0$$

nie jest konieczne uwzględnianie nieskończonej ilości składników sum /4/, Wystarczy wziąć pierwszych

$$S = \text{Ent} (t/2) + 1$$

składników. Dla uproszczenia zapisu wprowadzamy następującą procedurę standardową:

```

real procedure POT (A, B): real A; integer B;
POT:= if B = 0 then 1 else A ↑ B;
    
```

która umożliwia przedstawienie pierwszego wyrazu rozwinięć $K_A(p)$ i $K_{AP}(p)$ jako potęg o zerowym wykładniku.

Procedury realizujące operatory $K_A(p)$ i $K_{AP}(p)$ są procedurami нефunkcyjnymi i mają postać /A1 - odpowiada czasowi, A3 - rezultat, FUN - oznacza jedną z procedur $F1 \div F6$, tj. funkcję, na którą działają operatory/:

```
real procedure GONA (A1, FUN, A3,...);  
  value A1; real A1, A3; real procedure FUN;  
  begin  
    comment procedura realizuje operator  $K_A(p)$ ;  
    integer s, m;  
      s:= entier (A1/2) + 1;  
      A3:= 0;  
      for m:= 0 step 1 until s do  
        A3:= A3 + FUN (A1, 2 x m,...) x POT (rOA x rLA, m);  
  end GONA;  
real procedure GONP (A1, FUN, A3,...);  
  value A1; real A1, A3; real procedure FUN;  
  begin  
    comment procedura realizuje operator  
       $K_{AP}(p)$ ;  
    integer s, m, k; real Sum;  
      s:= entier (A1/2) + 1;  
      A3:= 0;  
      for m:=0 step 1 until s do  
  begin  
    Sum:= 0;  
    for k:= 0 step 1 until m do  
      Sum:= sum + POT (rOP x rLP, m - k) x  
        POT (rOA x rLA, k);  
      A3:= A3 + FUN (A1, 2 x m,...) x Sum;  
    end;  
  end GONP;
```

2.2.4. Ciało programu

Blok ten może być zorganizowany przez programistę w dowolny sposób, zgodnie z konkretnymi wymaganiami użytkownika. Procedury opisane w pkt. 2.2.1 ÷ 2.2.3 umożliwiają obliczenie wartości napięcia na każdym z końców obu linii w dowolnym momencie czasowym.

Ich organizacja jest dostatecznie elastyczna: pozwala na przeprowadzenie obliczeń dla zmieniających się parametrów

charakteryzujących modelowany układ i badanie wpływu tych zmian na procesy zachodzące w obu liniach. Zadaniem bloku, nazwanego ciałem programu, jest sterowanie tymi zmianami w sposób, który wynika ze specyfiki problemów rozwiązywanych za pomocą modelu.

2.3. Przykład zastosowania

Opisana wyżej metodyka była zastosowana między innymi do obliczenia przeników między równoległymi ścieżkami obwodów drukowanych. Przyjęto, że napięcie wymuszające ma kształt trapezoidalny. Aproksymacja taka jest szczególnie prosta i daje dostatecznie dokładne wyniki. Na rys. 4 pokazano przykładowo rezultaty obliczeń dla układu dwu linii /wykresy uzyskane na drukarce wierszowej/. Za jednostkę czasu przyjęto czas przejścia sygnału przez linię, za jednostkę napięcia poziom napięcia na linii aktywnej w stanie ustalonym. Parametry linii wydrukowane są na tabulogramach.

3. METODYKA OBLICZANIA PROCESÓW PRZEJŚCIOWYCH W LINII DŁUGIEJ POBUDZANEJ SKOKIEM JEDNOSTKOWYM PRZY NIELINIOWYCH WARUNKACH BRZEGOWYCH

Obliczanie przebiegów napięcia w linii długiej obciążonej opornością nieliniową i sterowanej źródłem o nieliniowej oporności wewnętrznej ma duże znaczenie w analizie układów elektro- nicznych. Zdaniem autora szczególnie dobrze nadaje się do tych obliczeń metoda podana przez Singletona [2], odznaczająca się prostotą i dająca wystarczająco dokładne wyniki. Poniżej wy- prowadzimy podstawowe zależności opisujące procesy zachodzące w linii długiej, na których opiera się wspomniana metoda.

Rozpatrzmy bezstratną linię długą pobudzaną skokiem jednostko- wym. Transformaty Laplace'a prądu i napięcia w takiej linii wy- rażają się wzorami:

DANE:
 WARIANT NUMER 1
 OPORNOSC FALOWA = 52.000 OMK
 OPORNOSCI : R40 = 15 R80 = 150 uAL = 400 RBL = 41ESKONR20WOSC
 TMETAF = 0.250
 TMETAP = 0.000
 T2 = 0.60

UAL, UAO -1.5											X1.5	
UFOS, UPLS -0.25											X0.25	
0.00	:	:	:	:	:	:	:	:	:	:	:	:
0.20	:	:	:	:	:	:	:	:	:	:	:	:
0.40	:	:	:	:	:	:	:	:	:	:	:	:
0.60	:	:	:	:	:	:	:	:	:	:	:	:
0.80	:	:	:	:	:	:	:	:	:	:	:	:
1.00	:	:	:	:	:	:	:	:	:	:	:	:
1.20	:	:	:	:	:	:	:	:	:	:	:	:
1.40	:	:	:	:	:	:	:	:	:	:	:	:
1.60	:	:	:	:	:	:	:	:	:	:	:	:
1.80	:	:	:	:	:	:	:	:	:	:	:	:
2.00	:	:	:	:	:	:	:	:	:	:	:	:
2.20	:	:	:	:	:	:	:	:	:	:	:	:
2.40	:	:	:	:	:	:	:	:	:	:	:	:
2.60	:	:	:	:	:	:	:	:	:	:	:	:
2.80	:	:	:	:	:	:	:	:	:	:	:	:
3.00	:	:	:	:	:	:	:	:	:	:	:	:
3.20	:	:	:	:	:	:	:	:	:	:	:	:
3.40	:	:	:	:	:	:	:	:	:	:	:	:
3.60	:	:	:	:	:	:	:	:	:	:	:	:
3.80	:	:	:	:	:	:	:	:	:	:	:	:
4.00	:	:	:	:	:	:	:	:	:	:	:	:
4.20	:	:	:	:	:	:	:	:	:	:	:	:
4.40	:	:	:	:	:	:	:	:	:	:	:	:
4.60	:	:	:	:	:	:	:	:	:	:	:	:
4.80	:	:	:	:	:	:	:	:	:	:	:	:
5.00	:	:	:	:	:	:	:	:	:	:	:	:
5.20	:	:	:	:	:	:	:	:	:	:	:	:
5.40	:	:	:	:	:	:	:	:	:	:	:	:
5.60	:	:	:	:	:	:	:	:	:	:	:	:
5.80	:	:	:	:	:	:	:	:	:	:	:	:
6.00	:	:	:	:	:	:	:	:	:	:	:	:
6.20	:	:	:	:	:	:	:	:	:	:	:	:
6.40	:	:	:	:	:	:	:	:	:	:	:	:
6.60	:	:	:	:	:	:	:	:	:	:	:	:
6.80	:	:	:	:	:	:	:	:	:	:	:	:
7.00	:	:	:	:	:	:	:	:	:	:	:	:
7.20	:	:	:	:	:	:	:	:	:	:	:	:
7.40	:	:	:	:	:	:	:	:	:	:	:	:
7.60	:	:	:	:	:	:	:	:	:	:	:	:
7.80	:	:	:	:	:	:	:	:	:	:	:	:
8.00	:	:	:	:	:	:	:	:	:	:	:	:
8.20	:	:	:	:	:	:	:	:	:	:	:	:
8.40	:	:	:	:	:	:	:	:	:	:	:	:
8.60	:	:	:	:	:	:	:	:	:	:	:	:
8.80	:	:	:	:	:	:	:	:	:	:	:	:
9.00	:	:	:	:	:	:	:	:	:	:	:	:
9.20	:	:	:	:	:	:	:	:	:	:	:	:
9.40	:	:	:	:	:	:	:	:	:	:	:	:
9.60	:	:	:	:	:	:	:	:	:	:	:	:
9.80	:	:	:	:	:	:	:	:	:	:	:	:
10.00	:	:	:	:	:	:	:	:	:	:	:	:
10.20	:	:	:	:	:	:	:	:	:	:	:	:
10.40	:	:	:	:	:	:	:	:	:	:	:	:
10.60	:	:	:	:	:	:	:	:	:	:	:	:
10.80	:	:	:	:	:	:	:	:	:	:	:	:
11.00	:	:	:	:	:	:	:	:	:	:	:	:
11.20	:	:	:	:	:	:	:	:	:	:	:	:
11.40	:	:	:	:	:	:	:	:	:	:	:	:
11.60	:	:	:	:	:	:	:	:	:	:	:	:
11.80	:	:	:	:	:	:	:	:	:	:	:	:
12.00	:	:	:	:	:	:	:	:	:	:	:	:

Rys. 4. Rezultaty przykładowych obliczeń napięć dla dwóch równoległych ścieżek obwodów drukowanych.
 Oznaczenia: "•" - przebieg napięcia na początku linii aktywnej
 "X" - przebieg napięcia na końcu linii aktywnej
 "-" - przebieg napięcia na początku linii pasywnej
 "0" - przebieg napięcia na końcu linii pasywnej

$$I(X, p) = \frac{U_0}{p(R_L + Z_0)} e^{-\gamma X} \frac{1 - r_L e^{-2\gamma(L-X)}}{1 - r_0 r_L e^{-2\gamma L}} \quad /5/$$

$$U(X, p) = \frac{Z_0 U_0}{p(R_L + Z_0)} e^{-\gamma X} \frac{1 + r_L e^{-2\gamma(L-X)}}{1 - r_0 r_L e^{-2\gamma L}} \quad /6/$$

gdzie:

U_0 - amplituda napięcia wymuszającego

R_L - oporność obciążenia

Z_0 - oporność falowa linii

$$r_L = \frac{R_L - Z_0}{R_L + Z_0} \quad r_0 = \frac{R_G - Z_0}{R_G + Z_0}$$

R_G - oporność wewnętrzna generatora

$$\gamma = p\sqrt{lc}$$

l, c - jednostkowa indukcyjność i pojemność linii

L - długość linii.

Wzory /5, 6/ możemy przedstawić w postaci szeregów potęgowych i otrzymać z nich wyrażenia dla transformat prądu i napięcia na końcach linii:

$$I(0, p) = \frac{U_0}{p(R_L + Z_0)} \sum_{k=0}^{\infty} \left\{ (r_L r_0)^k e^{-2\gamma L k} - r_L^{k+1} r_0^k e^{-2\gamma L(k+1)} \right\} \quad /7/$$

$$I(L, p) = \frac{U_0(1 - r_L)}{p(R_L + Z_0)} \sum_{k=0}^{\infty} (r_0 r_L)^k e^{-\gamma L(2k+1)} \quad /8/$$

$$U(0, p) = \frac{U_0 Z_0}{p(R_L + Z_0)} \sum_{k=0}^{\infty} \left\{ (r_L r_0)^k e^{-2k\tau L} + r_L^{k+1} r_0^k e^{-2\tau L(k+1)} \right\} \quad /9/$$

$$U(L, p) = \frac{U_0 Z_0 (1 + r_L)}{p(R_L + Z_0)} \sum_{k=0}^{\infty} (r_L r_0)^k e^{-\tau L(2k+1)} \quad /10/$$

W przypadku dopasowania $r_L r_0 = 0$ w powyższych formułach należy przyjąć $r_0 r_L^0 = 1$.

Łatwo zauważyć, że wartości prądów i napięć na początku linii zmieniają się tylko w momentach $t = n\tau$ / τ - czas propagacji sygnału przez linię/, gdzie $n = 0, 2, 4, 6, \dots$; na końcu linii w momentach $t = m\tau$, gdzie $m = 1, 3, 5, \dots$. Ze względu na prostotę funkcji wymuszającej łatwo można przejść z dziedziny operatora p do oryginału i obliczyć:

$$\begin{aligned} U[L, (n+1)\tau] - U[0, n\tau] &= \\ &= \frac{Z_0 U_0}{R_L + Z_0} \left\{ (1 + r_L) \sum_{k=0}^{n/2} (r_0 r_L)^k - \sum_{k=0}^{n/2} (r_0 r_L)^k - \right. \\ &\quad \left. - r_L \cdot \sum_{k=0}^{n/2-1} (r_0 r_L)^k \right\} = \\ &= \frac{Z_0 U_0}{R_L + Z_0} \left\{ r_L \cdot \sum_{k=0}^{n/2-1} (r_0 r_L)^k - r_L \sum_{k=0}^{n/2-1} (r_0 r_L)^k + \right. \\ &\quad \left. + r_L^{n/2+1} r_0^{\frac{n}{2}} \right\} = \frac{Z_0 U_0}{R_L + Z_0} r_L^{\frac{n}{2}+1} \cdot r_0^{\frac{n}{2}} \end{aligned}$$

$$\begin{aligned}
 I[L, (n+1)\tau] - I[0, n\tau] &= \\
 &= \frac{U_0}{R_L + Z_0} \left\{ (1 - r_L) \sum_{k=0}^{n/2} (r_0 r_L)^k + \right. \\
 &+ r_L \sum_{k=0}^{(n/2)-1} (r_0 r_L)^k - \left. \sum_{k=0}^{n/2} (r_0 r_L)^k \right\} = \\
 &= - \frac{U_0}{R_L + Z_0} r_L^{\frac{n}{2}+1} \cdot r_0^{\frac{n}{2}}
 \end{aligned}$$

i wreszcie:

$$\frac{I[L, (n+1)\tau] - I[0, n\tau]}{U[L, (n+1)\tau] - U[0, n\tau]} = - \frac{1}{Z_0} \quad /11/$$

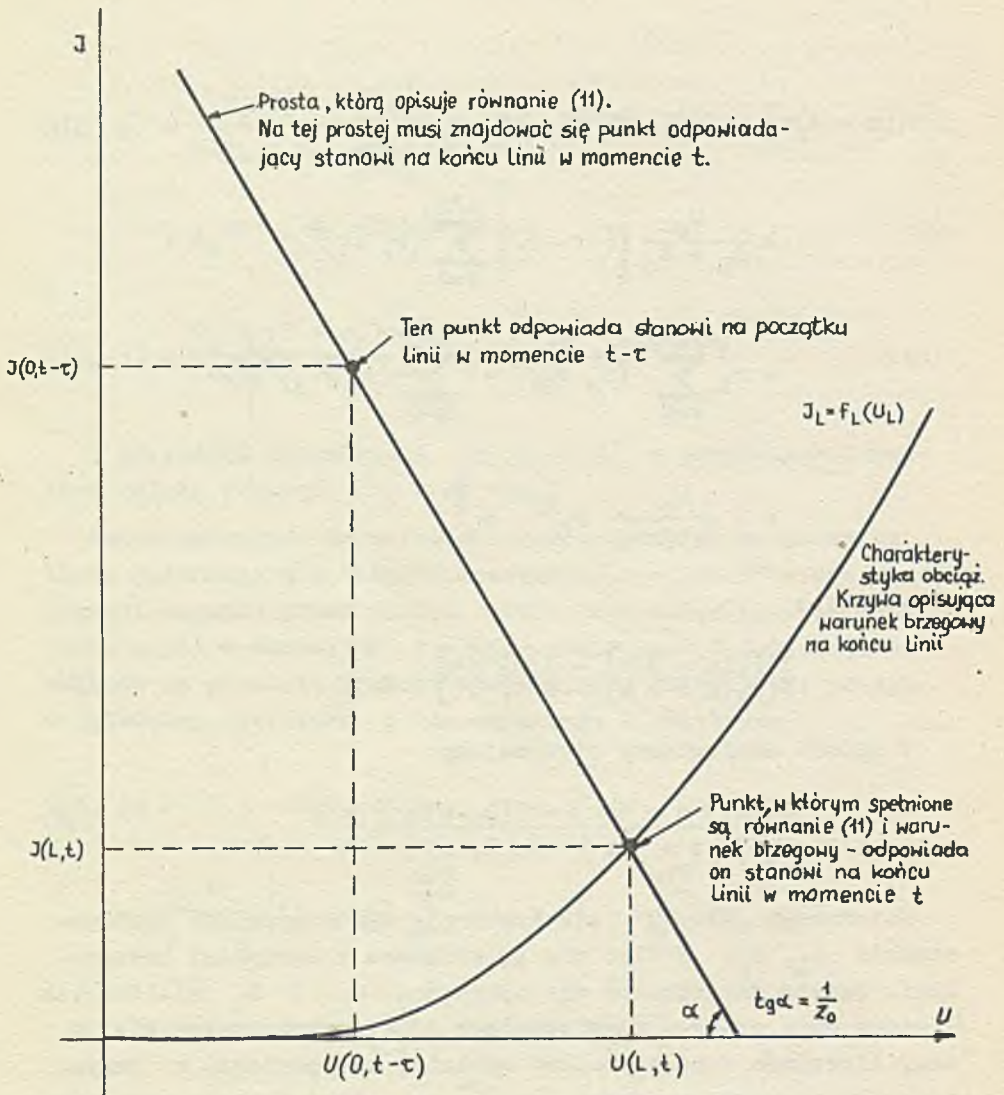
W sposób analogiczny otrzymujemy:

$$\frac{I[0, (m+1)\tau] - I[L, m\tau]}{U[0, (m+1)\tau] - U[L, m\tau]} = \frac{1}{Z_0} \quad /12/$$

Zależności /11-12/ nie zawierają współczynników niedopasowania r_L , r_0 , a więc nie są związane z warunkami brzegowymi. Będzie tak również dla oporności R_G i R_L nieliniowych /wówczas sumy potęg w rozwinięciach /10/, przekształca się w sumy iloczynów współczynników odbicia, ale ponieważ w danym momencie czasowym wartości r_0 i r_L , jako funkcje napięcia są we wszystkich sumach takie same - wszystkie zależności przytoczone powyżej pozostaną słuszne/.

Wzory /11-12/ są podstawą opisaną przez R. S. Singletona [2] metody graficznej wyznaczania przebiegów prądu i napięcia w liniach długich. Idea tej metody jest następująca.

Stan /napięcie i prąd/ w jakim znajduje się jeden z końców linii w danym momencie czasowym można interpretować jako punkt na płaszczyźnie /U, I/. Ponieważ stan końca linii zmienia się



Rys. 5. Ilustracja graficzna zależności występujących przy wyznaczaniu przebiegów prądów i napięcia w linii długiej metodą graficzną.

w dyskretnych momentach czasowych, proces przejściowy w linii będzie w pełni scharakteryzowany ciągiem punktów w płaszczyźnie $/U, I/$. Ze wzorów $/11 - 12/$ wynika, że punkt odpowiadający stanowi na jednym z końców linii w danym momencie czasowym t , leży na prostej o nachyleniu $\pm \frac{1}{Z_0}$ i przechodzącej przez punkt w płaszczyźnie $/U, I/$ odpowiadający stanowi na przeciwnym końcu linii zaistniałemu tam od chwili $(t - \tau)$.

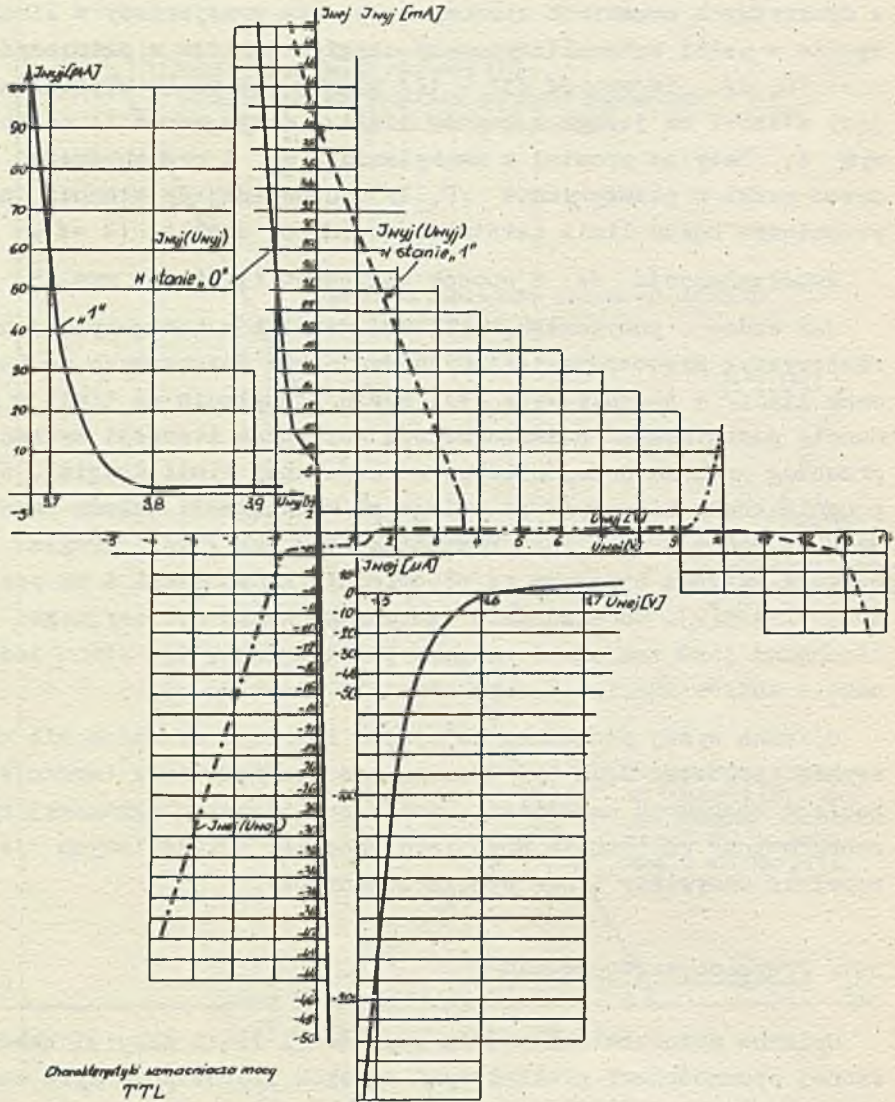
Współzależność tę w sposób poglądowy ilustruje rys. 5.

Jak widać z powyższego, znajomość warunków brzegowych /charakterystyk prądowo-napięciowych dwójników dołączonych na końcach linii/ i początkowych /tj. prądu i napięcia na linii w momencie początkowym/ pozwala metodą kolejnych iteracji wyznaczyć przebieg procesu przejściowego w rozważanej linii długiej. Wykonanie kroku iteracyjnego polega na rozwiązaniu układu dwóch równań; jedno z nich jest równaniem $/11/$ lub $/12/$, drugie opisuje warunek brzegowy na odpowiednim końcu linii i ma postać: $I = f(U)$. Do wykonania pierwszego kroku iteracyjnego niezbędna jest znajomość warunku początkowego, tj. stanu jednego z końców linii w pewnym momencie czasowym t_0 .

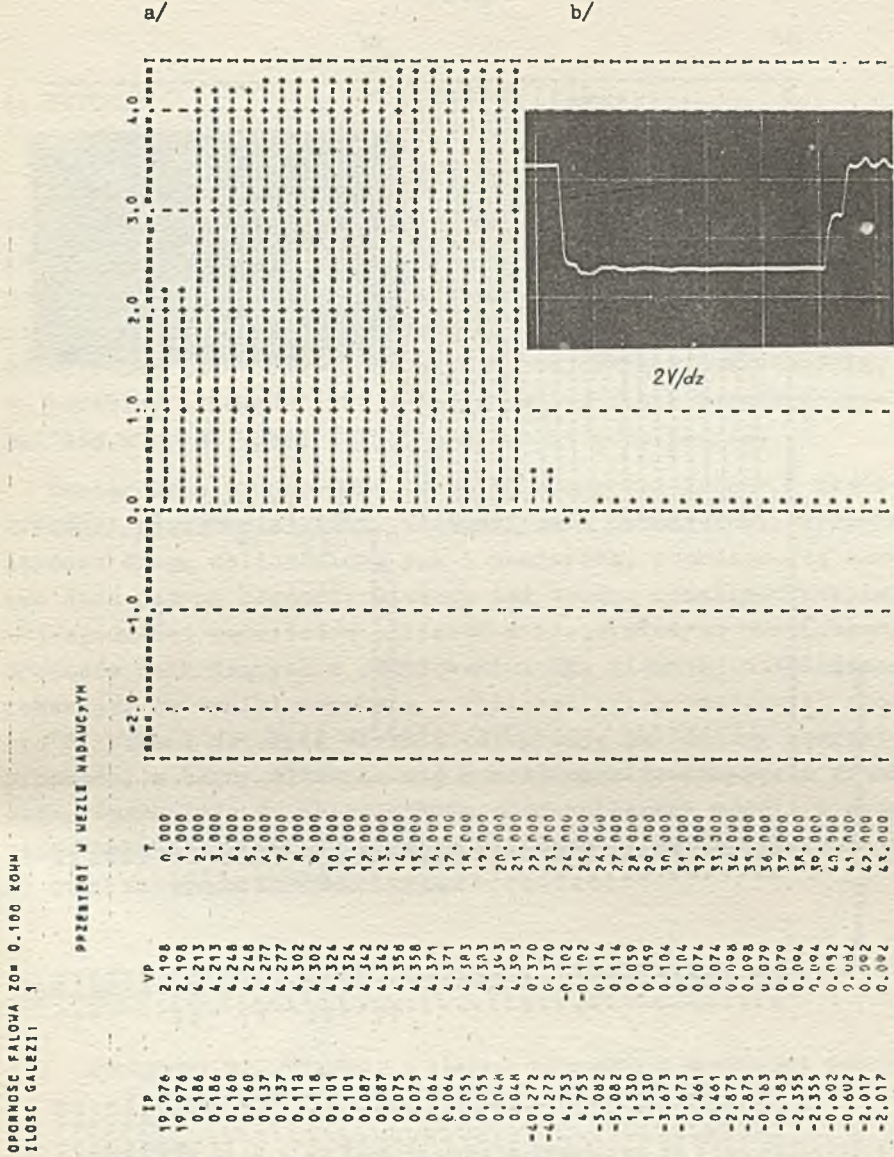
Opisana wyżej procedura kolejnych iteracji odznacza się niezwykłą prostotą. Może być stosowana w rozwiązywaniu bardzo szerokiego wachlarza zagadnień. Sposób realizacji programowej opisanej metody obliczania przebiegu procesów przejściowych jest zupełnie oczywisty i nie wymaga komentarzy.

3.1. Przykład zastosowania

Opisana metodyka obliczania odpowiedzi linii długiej zakończonej opornościami nieliniowymi na skok jednostkowy była zastosowana do wyznaczania procesów przejściowych w blokach cyfrowych zbudowanych z układów mikroelektronicznych typu TTL. Potrzebne do tego celu charakterystyki prądowo-napięciowe /rys. 6/ uzyskano drogą pomiarową. Na rys. 7 i 8 pokazano rezultaty obliczeń dla przypadku współpracy wzmacniaczy mocy poprzez kabel koncentryczny o oporności falowej 100Ω . Dla porównania dołączono zdjęcie fotograficzne przedstawiające rzeczywiste prze-



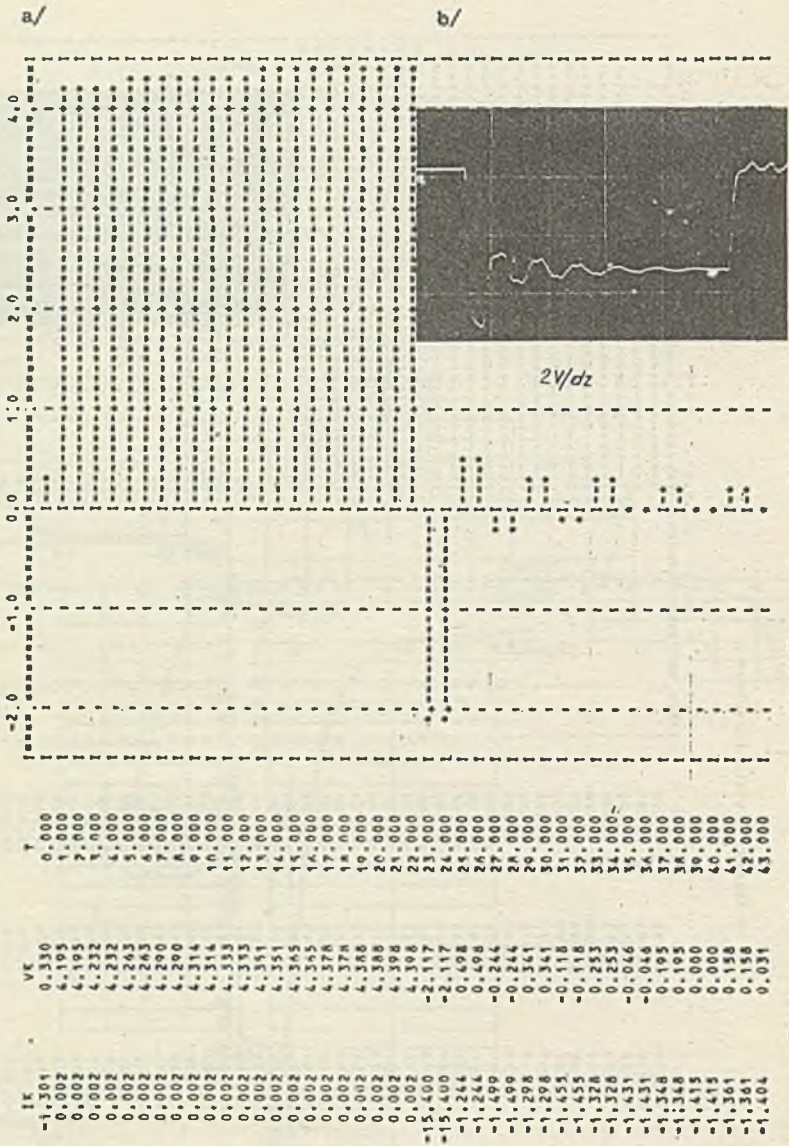
Rys. 6. Charakterystyki prądowo-napięciowe dla wzmacniacza mocy TTL stosowane przy obliczeniach przykładowych



Rys. 7. Rezultaty przykładowych obliczeń napięcia w węzle nadawczym /a/ oraz fotografia rzeczywistego przebiegu /b/.
 Oznaczenia: IP - prąd w węzle nadawczym [V]
 VP - napięcie w węzle nadawczym [mA]
 T - czas względny

OPORNOŚĆ FALOWA $Z_0 = 0,100 \text{ KON}$
 IŁOŚĆ GAŁĘZI 1

PRZENIEŚNIKI W WĘZŁACH ODBIORCZYCH



Rys. 8. Rezultaty przykładowych obliczeń napięcia w węźle odbiorczym /a/ oraz fotografia rzeczywistego przebiegu /b/.
 Oznaczenia: IK - prąd w węźle odbiorczym [A]
 VK - napięcie w węźle odbiorczym [mA]
 T - czas względny

4. WNIOSKI

Opisane w punktach 2 i 3 modele matematyczne były stosowane w praktyce do opracowania reguł okablowania bloków EMC III generacji. Analiza otrzymanych rezultatów pokazała ich pełną przydatność. Konieczne jest jednak dokładne określenie danych, które są podstawą przeprowadzanych obliczeń. Jeżeli parametry modelowanej linii określone są z wystarczającą dokładnością, to przebieg procesów przejściowych w tej linii będzie pokrywał się z przebiegami obliczonymi drogą modelowania.

Szczególnie trudno jest określić oporności falowe i współczynniki sprzężenia linii. Wartości tych parametrów, uzyskane zarówno drogą obliczeniową jak i pomiarową, obarczone są zawsze dość dużymi błędami. Dlatego też trudno oczekiwać absolutnej zgodności rezultatów obliczeń z rzeczywistymi przebiegami procesów zachodzących w okablowaniu. Tym niemniej doświadczenie pokazuje, że wyniki uzyskane z obliczeń przeprowadzanych zgodnie z opisaną metodyką wiernie odtwarzają jakościowy charakter procesów, a błędy mieszczą się w możliwych do przyjęcia granicach. Gwarantuje to pełną przydatność opisanych modeli w projektowaniu okablowania szybkich systemów cyfrowych, zwłaszcza na jego wstępnym etapie.

Literatura

- [1] KOWALEWSKA M., JOZANIS M.: Analiza przeników i transmisji sygnałów w okablowaniu urządzeń cyfrowych, Prace IMM, Warszawa, 1969: 2, XI.
- [2] SINGLETON R. S.: No Need to Juggle Equations to Find Reflection - Just Draw Three Lines, Electronics, October 1968: 28.
- [3] KOWALEWSKA M., KUBERA W.: Ocena długości przewodów okablowania bloków logicznych na układach TTL, Opracowanie wewnętrzne Zakładu Elektroniki Cyfrowej IMM, kwiecień 1969.

ПРИМЕНЕНИЕ ЦВМ К НЕКОТОРЫМ ПРОБЛЕМАМ ПРОЕКТИРОВАНИЯ ЛИНИЙ
СОЕДИНЕНИЯ В ЭВМ III ГЕНЕРАЦИИ

Резюме

В проектировании быстродействующих цифровых блоков, построенных на интегральных схемах, основную роль играют процессы происходящие в линиях соединения. Ошибочно запроектированные соединения могут привести к чрезмерным помехам не допускающим правильной работы устройства и даже к разрушению элементов.

Ввиду весьма коротких времен переключения интегральных схем соединения следует рассматривать как длинные линии, что значительно осложняет анализ происходящих процессов. Это затруднение можно решить применяя ЭВМ к проектированию линий соединения.

В работе указаны принципы создания программ для ЭВМ, позволяющих провести анализ передачи сигналов в длинной линии оконченной нелинейными сопротивлениями, а также в двух длинных линиях с индуктивным и емкостным сопряжением.

Действие первой из описанных программ основано на следующих зависимостях

$$\frac{I(L, t_0 + \tau) - I(0, t_0)}{U(L, t_0 + \tau) - U(0, t_0)} = - \frac{1}{Z_0} \frac{I(0, t_0 + \tau) - I(L, t_0)}{U(0, t_0 + \tau) - U(L, t_0)} = \frac{1}{Z_0}$$

где: $I /x, t/$ - ток проходящий через линию в точке x ,
во время t ;

$U /x, t/$ - напряжение на линии в точке x ,
во время t ;

L - координата x конца линии /длина линии/

t_0 - любой, начальный момент времени

τ - время распространения сигнала через
линию

Z_0 - характерное сопротивление линии.

Выше указанные уравнения и граничные условия, определяющие зависимость тока от напряжения на концах линии, позволяют определить переходные процессы /ток и напряжение/ выступающие в рассматриваемой длинной линии при данных начальных условиях, применяя метод шаг за шагом.

Вторая из описанных программ реализует математическую модель возникновения помех, указанная в работе [1] Эта модель позволяет определить напряжение в двух сопряженных однородных линиях без потерь при следующих предложениях:

- линии являются одинаковые и параллельные,
- начальные условия нулевые,
- обе линии с сопротивлением нагрузки на обоих концах,
- взаимное воздействие между линиями незначительное.

В работе подан список процедур языка АЛГОЛ 60 позволяющий вычислить величину напряжения для любого момента времени на обоих концах линии.

DIGITAL COMPUTER APPLICATION TO SOME PROBLEMS OF DESIGNING III GENERATION DIGITAL SYSTEM WIRING

Summary

The main role in designing fast digital blocks build on integrated circuits is played by the processes occurring within the wiring. A wrong wiring can lead to excessive disturbances disabling a correct work of the device and even to a destruction of elements.

Because of a very short time of integrated circuit switching - the connecting wires must be considered as long lines which significantly complicates the analysis of processes occurring within the wiring.

The paper gives the principles of writing programs for digital computers, enabling the analysis of signal transmission along a long line terminated with nonlinear resistances, and within two long lines inductive and capacitive coupled.

The first described program operates on the basis of the following dependences:

$$\frac{I(L, t_0 + \tau) - I(0, t_0)}{U(L, t_0 + \tau) - U(0, t_0)} = -\frac{1}{Z_0} \quad \frac{I(0, t_0 + \tau) - I(L, t_0)}{U(0, t_0 + \tau) - U(L, t_0)} = \frac{1}{Z_0}$$

where: $I/x, t/$ - current passing through the line in point x , in time t ,

$U/x, t/$ - voltage on the line in point x in time t ,

L - coordinate x of the end of line /line length/,

t_0 - assumed initial time moment,

τ^0 - the time of signal propagation through the line,

Z_0 - characteristic resistance of the line

The above equations and boundary conditions, determining current dependences on the voltages at the ends of lines, permit to denote transient processes of the current and voltage waveforms occurring within the considered long line under the given initial conditions using the step by step method.

The second described program realizes an arising mathematical model of crosstalk given in /1/. This model makes it possible to determine the voltage waveform in two coupled uniform long lossless lines under the following assumptions:

- the lines are similar and parallel,
- initial conditions are zero,
- both lines are terminated with linear resistances on both ends,
- the mutual influence between the lines is negligible.

The paper contains a set of ALGOL-60 language procedures permitting to compute the voltage value for any time moment on both ends of the line.

AUTORZY ARTYKUŁÓW ZAMIESZCZONYCH W ZESZYCIE 2 "PRAC IMM"

CEMMA 2 - JĘZYK DO SYMULACJI PROCESÓW CIĄGŁYCH
Z AUTOMATYCZNĄ OPTIMALIZACJĄ PARAMETRÓW MODELU

Waldemar MADEJ mgr inż. Studia ukończył w 1967 r. na Wydziale Elektroniki Politechniki Warszawskiej. W 1967 r. rozpoczął pracę w Instytucie Maszyn Matematycznych w Zakładzie Sterowania, Początkowo zajmował się zagadnieniami wykorzystania maszyny cyfrowej do badania procesów dynamicznych i opracowywaniem języków symulacyjnych ułatwiających takie badania. Jest autorem szeregu publikacji na ten temat. Od połowy 1970 r. zajmował się problematyką oprogramowania maszyn cyfrowych pracujących w systemie "real-time". Obecnie pracuje w Zakładach Wytwórczych Przyrządów Pomiarowych "ERA".

Włodzimierz MARDAL, mgr inż. W roku 1959 ukończył studia na Wydziale Łączności Politechniki Warszawskiej. Od 1958 roku pracuje w Instytucie Maszyn Matematycznych, specjalizując się początkowo w zagadnieniach analizy, syntezy i badania systemów sterowania opartych na maszynach cyfrowych, a następnie w zastosowaniu maszyn cyfrowych do modelowania układów automatyki. Od kilku lat poświęcił się problemom oprogramowania systemów przetwarzania informacji na bieżąco. Jest autorem szeregu publikacji na temat zastosowań maszyn w systemach sterowania oraz modelowania cyfrowego układów automatyki. Pełni funkcję kierownika Zakładu Sterowania w IMM.

Andrzej MULARA mgr inż. Ukończył studia na Wydziale Elektroniki Politechniki Warszawskiej w 1968 r. W latach 1968-1970 pracował w Instytucie Maszyn Matematycznych, w Zakładzie Sterowania. Początkowo zajmował się symulacją obiektów energetycznych za pomocą języka CEMMA. Następnie brał udział w pracach nad zdefiniowaniem języka symulacyjnego CEMMA 2 i konstrukcją translatora tego języka. Od 1 października 1970 r. jest na studium doktoranckim przy Instytucie Automatyki Politechniki Warszawskiej.

ANALIZA EFEKTYWNOŚCI MODELOWANIA SIECI PRZEŁĄCZAJĄCYCH
METODĄ LISTOWĄ I BLOKOWĄ

Tadeusz SINKIEWICZ mgr inż. Ukończył w 1961 r. Wydział Automatyki i Techniki Obliczeniowej Moskiewskiego Instytutu Energetycznego w Moskwie. Następnie rozpoczął pracę w Instytucie Maszyn Matematycznych, zajmując się konstrukcją układów podstawowych, przyrządów pomiarowych i urządzeń cyfrowych. Obecnie pracuje w Zakładzie Organizacji Maszyn Cyfrowych IMM nad zagadnieniami modelowania układów przełączających. Jest autorem szeregu publikacji.

ZASTOSOWANIE EMC W PEWNYCH ZAGADNIENIACH PROJEKTOWANIA
OKABLOWANIA SYSTEMÓW CYFROWYCH III GENERACJI

Wojciech KUBERA mgr inż., ur. w 1943 roku. Studia ukończył na Wydziale Automatyki i Elektronicznej Techniki Obliczeniowej Moskiewskiego Instytutu Energetycznego. W Instytucie Maszyn Matematycznych pracuje od 1967 roku do chwili obecnej, zajmując się problematyką zastosowania EMC w analizie zakłóceń powstających w okablowaniu szybkich maszyn cyfrowych oraz w projektowaniu pakietów mikroelektronicznych.

