ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1993
Seria: INFORMATYKA z. 24 Nr kol. 1222

Stefan WĘGRZYN

SYSTEMY STEROWANE PRZEPŁYWEM OPERACJI I SYSTEMY STEROWANE PRZEPŁYWEM ARGUMENTÓW *

Streszczenie. W pracy scharakteryzowano istotę organizacji i struktury dwóch podstawowych rodzajów systemów realizacji algorytmów, a mianowicie:
1. systemów sterowanych przepływem operacji,
2. systemów sterowanych przepływem argumentów.
Wprowadzono pojęcie postaci kanonicznej algorytmu oraz jej macierzy zmiennych i zastosowano je do syntezy systemu sterowanego przepływem argumentów, realizującego zadany algorytm.

SYSTEMS CONTROLLED BY FLOW OF OPERATIONS AND SYSTEMS CONTROLLED BY FLOW OF ARGUMENTS

Summary. The article considers two basic types of systems in which algorithms may be performed, namely:
1. systems controlled by flow of operations,
2. systems controlled by flow of arguments.
Basic features of organization and structure of these systems are given. Notions of canonical form of the algorithm and of its variable matrix are presented and used to syntesise the latter type of systems.

SYSTEMES CONTROLES PAR LE FLUX DES OPERATIONS ET SYSTEMES CONTROLES PAR LE FLUX DES ARGUMENTS

Résumé. L'article considère deux types fondamentaux des systèmes dans lesquels des algorithmes peuvent être executés, et notamment:
1. systèmes controlés par le flux des opérations,
2. systèmes controlés par le flux des arguments.
On discute les bases de l'organisation et de la structure de ces systèmes. On introduit les notions de la forme canonique de l'algorithme et de sa matrice de variables et on utilse ces notions dans la sythèse des systèmes contolés par le flux des arguments.

*Praca zrealizowana w ramach Projektu Badawczego (GRANTU) nr KBN 3 P406 011 04

1. Systemy realizacji algorytmów

Przez algorytm rozumiemy skończony, uporządkowany zbiór operacji, które po wykonaniu na wprowadzonych danych dają rozwiązanie dowolnego zadania z określonej klasy zadań [1].

Można powiedzieć, że algorytm ustala operacje, które mają być wykonywane na określonych argumentach, czyli występujących w algorytmie danych i zmiennych.

Ale można to ująć inaczej i powiedzieć, że algorytm ustala argumenty, które mają być poddane określonym występującym w algorytmie operacjom.

Ilustracją tych dwóch różnych interpretacji algorytmu może być przykładowa macierz dla pewnego algorytmu rys.1, podająca, na których występujących w tym algorytmie argumentach a_1, a_2, a_3 mają być wykonywane odpowiednie, występujące w tym algorytmie operacje O_1, O_2, O_3, O_4.

		ARGUMENTY		
		a_1	a_2	a_3
OPERACJE	O_1	X		
	O_2	X	X	
	O_3	X	X	
	O_4			X

Rys. 1. Przykładowa macierz dla pewnego algorytmu podająca, na których argumentach a_1, a_2, a_3 mają być wykonywane odpowiednie występujące w tym algorytmie operacje O_1, O_2, O_3, O_4

Fig. 1. An exemplary matrix of an algorithm indicating arguments a_1, a_2, a_3 on which the operations O_1, O_2, O_3, O_4 should be executed.

Z tej macierzy (rys.1) odczytujemy, że np. operację O_1 możemy wykonać wtedy, gdy znamy argument a_1, a np. operację O_3 wtedy, gdy znamy argumenty a_1 i a_2. Ale możemy tę macierz odczytywać inaczej, a mianowicie tak, że np. gdy znamy argument a_1, to wtedy możemy na nim dokonać operacji O_1, a gdy np. znamy argumenty a_1 i a_2, to wtedy możemy na nich dokonać zarówno operację O_2, jak i O_3.

Z tych różnych możliwości interpretacji macierzy algorytmów wynikają dwa różne typy systemów ich realizacji.

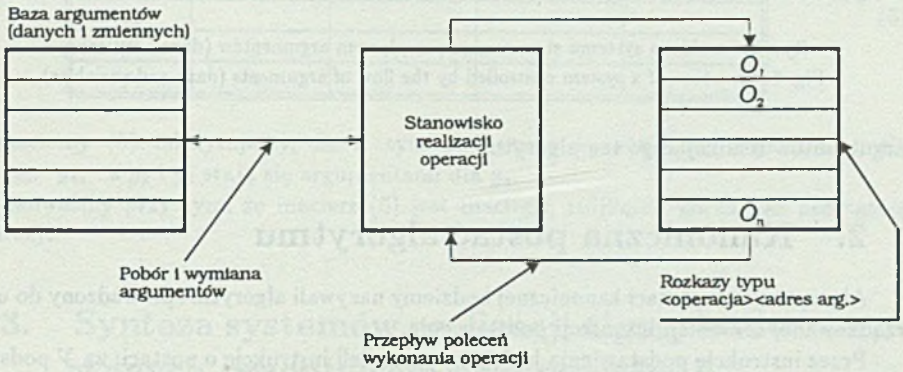
Pierwszy, to systemy sterowane przepływem operacji, w których wychodzi się od operacji i realizuje je kolejno, w ustalonym w algorytmie porządku, pobierając w miarę ich realizacji właściwe dla nich argumenty, to jest dane i zmienne.

Drugi, to systemy sterowane przepływem argumentów, w których wychodzi się od argumentów, a operacje przeprowadza się nie jedna po drugiej, w określonym z góry porządku, ale wykonuje się je wówczas, gdy w systemie pojawiają się przewidziane dla nich parametry.

Struktury systemów sterowanych przepływem operacji i systemów sterowanych przepływem argumentów przedstawione są odpowiednio na rys. 2 i na rys. 3.

Warto zauważyć, że struktura systemu przedstawionego na rys. 2 odpowiada strukturom konwencjonalnych komputerów typu von Neumana, w których rozkazy wykonania określonych operacji na wskazanych danych przesyłane są kolejno na stanowisko realizacji operacji. Po wykonaniu zadanej rozkazem operacji stanowisko zwalnia się dla następnej itd. aż do zakończenia realizacji całego algorytmu.

Natomiast strukturę sytemu przedstawionego na rys. 3 można by nazwać strukturą sieciową operacji sterowanych przepływem argumentów. Jej działanie polega na tym, że gdy w sieci pojawiają się argumenty potrzebne do wykonania określonej operacji, to zostaje ona wykonana, w rezultacie czego powstają i zostają wprowadzone do sieci nowe argumenty powodujące realizację następnych, odpowiednich operacji itd. aż do zakończenia realizacji całego algorytmu.



Rys. 2. Struktura systemu sterowanego przepływem operacji

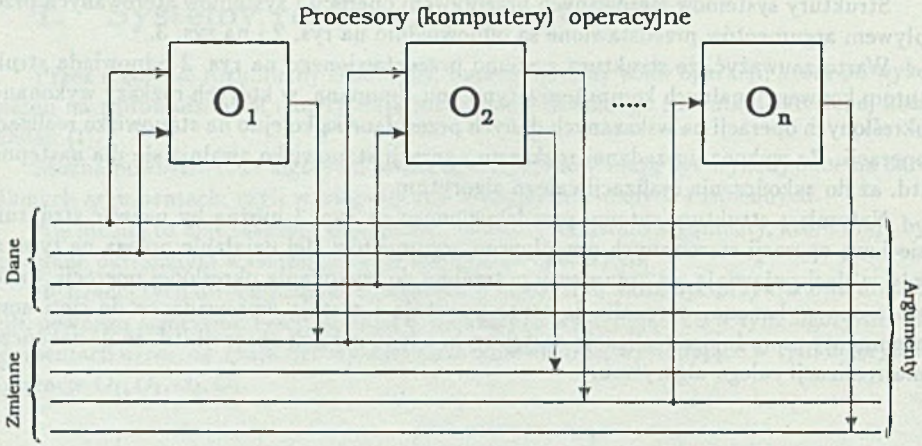
Fig. 2. Structure of a system controlled by the flow of operations

Można więc powiedzieć, że w systemach sterowanych przepływem operacji, operacje pobierają potrzebne do ich wykonania argumenty, a w systemach sterowanych przepływem argumentów — argumenty „szukają" przewidzianych do ich przekształcenia operacji.

Systemy sterowane przepływem operacji, w których kolejna operacja jest realizowana wówczas, gdy poprzednia została zakończona, posiadają naturalne predyspozycje do realizacji algorytmów szeregowych.

Systemy sterowane przepływem argumentów, w których operacje realizowane są wówczas, gdy pojawiają się potrzebne do ich wykonania argumenty, posiadają naturalne predyspozycje do realizacji algorytmów równoległych. Systemy te muszą być jednorodne i często, a zwłaszcza w przypadku obszernych algorytmów złożonych z podalgorytmów, może być tak, że algorytm jako całość złożona z podalgorytmów będzie realizowany w systemie sterowanym przepływem argumentów, natomiast poszczególne podalgorytmy będą realizowane w systemach sterowanych przepływem operacji.

W niniejszej pracy wprowadzimy pojęcie kanonicznej postaci algorytmu i wskażemy na możliwość jej bezpośredniego wykorzystania do syntezy systemu sterowanego przepływem



Rys. 3. Struktura systemu sterowanego przepływem argumentów (dane i zmienne)
Fig. 3. Structure of a system controlled by the flow of arguments (data and variables)

argumentów realizującego ten algorytm.

2. Kanoniczna postać algorytmu

Algorytmem w postaci kanonicznej będziemy nazywali algorytm sprowadzony do uporządkowanej sekwencji instrukcji podstawienia.

Przez instrukcję podstawienia będziemy rozumieli instrukcję o postaci: za $\mathcal{Y}$ podstaw wynik pewnej operacji na argumentach, to jest danych $\mathcal{D}$ i zmiennych $\mathcal{X}$, co zapiszemy:

$$\mathcal{Y} = \mathcal{O}(\mathcal{D}, \mathcal{X}) \tag{1}$$

gdzie $\mathcal{Y}$ i $\mathcal{X}$ — wektory zmiennych,
$\mathcal{D}$ — wektor danych,
$\mathcal{O}$ — operacja.

W tym zrozumieniu instrukcją podstawienia będzie

$$y = (a + b) \cdot x \tag{2}$$

ale również

$$y = NWD(a, b)$$

gdzie $NWD(a, b)$ – operacja znajdowania największego wspólnego dzielnika dwóch dodatnich liczb całkowitych a i b, takich, że $a > b$. Instrukcją podstawienia będzie też:

$$y_1, y_2, \ldots, y_n = \text{Dzielniki liczby całkowitej } a \tag{3}$$

Wprowadzimy jeszcze pojęcia macierzy zmiennych formy kanonicznej algorytmu. Zilustrujemy to na przykładzie algorytmu o następującej postaci kanonicznej:

$$\begin{aligned} y_1 &= O_1(a_1) \\ y_2 &= O_2(y_1, a_2) \\ y_3 &= O_3(y_1, a_3) \\ y_4 &= O_4(y_1, y_2, y_3) \end{aligned} \tag{4}$$

gdzie a_1, a_2, a_3 — dane,
y_1, y_2, y_3, y_4 — zmienne.

Macierz zmiennych formy kanonicznej będzie miała w tym przypadku postać:

	y_1	y_2	y_3	y_4
y_1		×	×	×
y_2				×
y_3				×
y_4				

(5)

Z macierzy (5) odczytujemy, że w tym algorytmie y_1 staje się argumentem dla y_2, y_3, y_4, a y_2 i y_3 stają się argumentami dla y_4.

Zauważmy przy tym, że macierz (5) jest macierzą trójkątną górną bez przekątnej głównej.

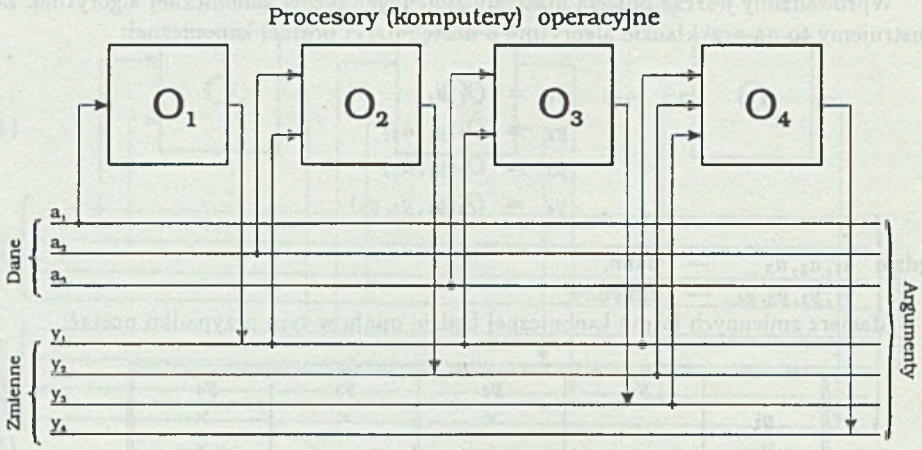
3. Synteza systemów realizacji algorytmów sterowanych przepływem argumentów

Jeżeli macierz zmiennych formy kanonicznej danego algorytmu jest macierzą trójkątną górną bez przekątnej głównej, to wynika z niej wprost realizujący ten algorytm system sterowany przepływem argumentów. Dla przykładu, na rys. 4 przedstawiono system sterowany przykładem argumentów realizujący algorytm (4).

Działanie systemu z rys. 4 jest następujące. Po wprowadzeniu wartości danych a_1, a_2, a_3 zostaje uruchomiona operacja O_1 i powstająca w jej wyniku wartość zmiennej y_1 zostaje wprowadzona na linię tej zmiennej. Pojawienie się wartości zmiennej y_1 na linii zmiennej y_1 powoduje uruchomienie operacji O_2 i O_3, a powstające w ich wyniku wartości zmiennych y_2 i y_3 zostają wprowadzone na linie tych zmiennych, a to, łącznie z poprzednio otrzymaną wartością zmiennej y_1, uruchamia operację O_4. W wyniku wykonania operacji O_4 otrzymuje się rezultat algorytmu w postaci odpowiedniej wartości zmiennej y_4.

Warto zauważyć, że w rozpatrywanym przypadku w systemie przedstawionym na rys. 4 operacje O_2 i O_3 realizowane są równolegle.

Jak to bezpośrednio wynika z przeprowadzonej syntezy układu sterowanego przepływem argumentów, w przypadku gdy macierz zmiennych jest macierzą trójkątną górną bez przekątnej głównej, jest ona łatwa i wynika wprost z kanonicznej formy algorytmu.



Rys. 4. System sterowany przepływem argumentów realizujący algorytm (4)
Fig. 4. System controlled by the flow of arguments and performing algorithm (4)

4. Przykłady

4.1. Rozwiązywanie liniowych równań algebraicznych o stałych współczynnikach metodą Cramera

Dany jest układ równań:

$$\sum_{j=1}^{n} a_{ij}\, x_j \;=\; b_i, \qquad i \in [1,n] \tag{6}$$

o stałych współczynnikach a_{ij}, $i \in [1,n]$, $j \in [1,n]$ będących liczbami rzeczywistymi. Wyznacznik układu (6) zapiszemy w postaci $|a_{ij}|_1^n$. Algorytm rozwiązywania takiego układu równań w postaci kanonicznej jest następujący:

$$1.\quad d = \begin{cases} |a_{ij}|_1^n, & \text{jeżeli } |a_{ij}|_1^n \neq 0 \\ \text{STOP, układ nie ma rozwiązań,} & \text{jeżeli } |a_{ij}|_1^n = 0 \end{cases} \qquad \text{Operacja } O_1$$

$$2.\quad d_i = \begin{cases} \text{wyznacznik macierzy utworzonej z macierzy } [a_{ij}] \\ \text{przez zastąpienie } i\text{-tej kolumny przez kolumnę } [b_i] \end{cases} \qquad \text{Operacje } O_{2i},\ i \in [1,n] \tag{7}$$

$$3.\quad p_i = \frac{d_i}{d} \qquad \text{Operacje } O_{3i},\ i \in [1,n]$$

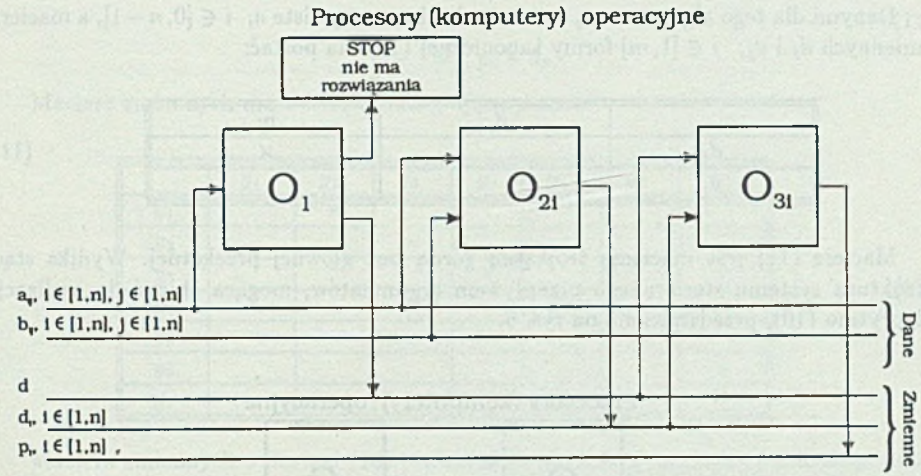
gdzie p_i — pierwiastki układu równań (6).

Danymi dla tego algorytmu są a_{ij}, b_i, $i, j \in [1, n]$, a macierz zmiennych formy kanonicznej (7) ma postać (8), z której wynika, że d i d_i stają się argumentami dla p_i.

	d	d_1	p_1
d			×
d_i			×
p_i			

(8)

Macierz (8) jest macierzą trójkątną górną bez przekątnej głównej. Wynika stąd struktura systemu sterowanego przepływem argumentów, mogąca służyć do realizacji algorytmu (7) przedstawionego na rys. 5.



Rys. 5. Struktura systemu sterowanego przepływem argumentów dla znajdowania pierwiastków układów metodą Cramera
Fig. 5. Structure of the system controlled by the flow of arguments which seeks the root of an equation system using Cramer's method

Warto zauważyć, że w systemie przedstawionym na rys. 5 operacja O_1 i operacje O_{2i} $i \in [1, n]$ będą realizowane równolegle.

4.2. Znajdowanie całkowitych pierwiastków równania algebraicznego

$$x^n + \sum_{i=n-1}^{0} a_i x^i = 0 \tag{9}$$

o współczynnikach stałych, całkowitych a_i, $i \in [0, n-1]$. Postać kanoniczna algorytmu znajdowania pierwiastków całkowitych równania (9) jest następująca:

1. $d_1, d_2, \cdots d_j, \cdots d_m$ = dzielniki a_0 Operacja O_1
2. od $j = 1$ do m wykonuj

$$p_j = \begin{cases} d_j, & \text{jeżeli } d_j^n + \sum_{i=n-1}^{0} a_i d_j^i = 0 \\ \phi, & \text{jeżeli } d_j^n + \sum_{i=n-1}^{0} a_i d_j^i \neq 0 \end{cases} \qquad \text{Operacje } O_{2j} \tag{10}$$

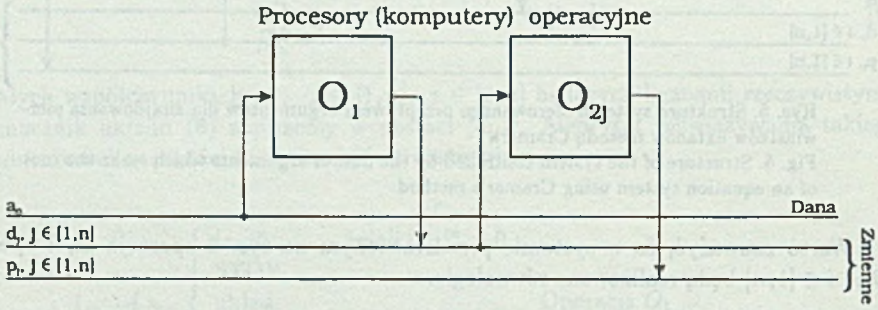
gdzie p_j, $j \in [1, m]$, to pierwiastki całkowite równania (9), a ϕ — symbol pusty.

Danymi dla tego algorytmu są całkowite liczby rzeczywiste a_i $i \in [0, n-1]$, a macierz zmiennych d_j i p_j, $j \in [1, m]$ formy kanonicznej (10) ma postać:

	d_j	p_j
d_j		×
p_j		

(11)

Macierz (11) jest macierzą trójkątną górną bez głównej przekątnej. Wynika stąd struktura systemu sterowanego przepływem argumentów, mogąca służyć do realizacji algorytmu (10), przedstawiona na rys. 6.



Rys. 6. Struktura systemu sterowanego przepływem argumentów dla znajdowania pierwiastków całkowitych równania (9)
Fig. 6. Structure of a system controlled by the flow of arguments and finding the integral roots of equation (9)

W systemie przedstawionym na rys. 6 operacje O_{2j} dla $j \in [1, m]$ wykonywane są równolegle.

5. Efektywność czasowa w systemach sterowanych przepływem argumentów

Efektywność czasowa realizacji algorytmów w systemach sterowanych przepływem argumentów zależy od postaci macierzy zmiennych, a ta od przyjętej organizacji postaci kanonicznej algorytmu. Ten sam algorytm można przedstawić w różnych formach organizacyjnych, które będą miały decydujący wpływ na czas jego realizacji w systemie sterowanym przepływem argumentów. Zilustrujemy to przykładem algorytmu dodawania ośmiu liczb a_i, $i \in [1,8]$, [2].

Najprostsza jego postać kanoniczna jest następująca:

$$\begin{aligned} y_1 &= a_1 + a_2 \\ y_2 &= y_1 + a_3 \\ \dots \quad &\dots \quad \dots \\ y_7 &= y_6 + a_8 \end{aligned} \tag{12}$$

Macierz zmiennych ma wówczas następującą postać:

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
y_1		×					
y_2			×				
y_3				×			
y_4					×		
y_5						×	
y_6							×
y_7							

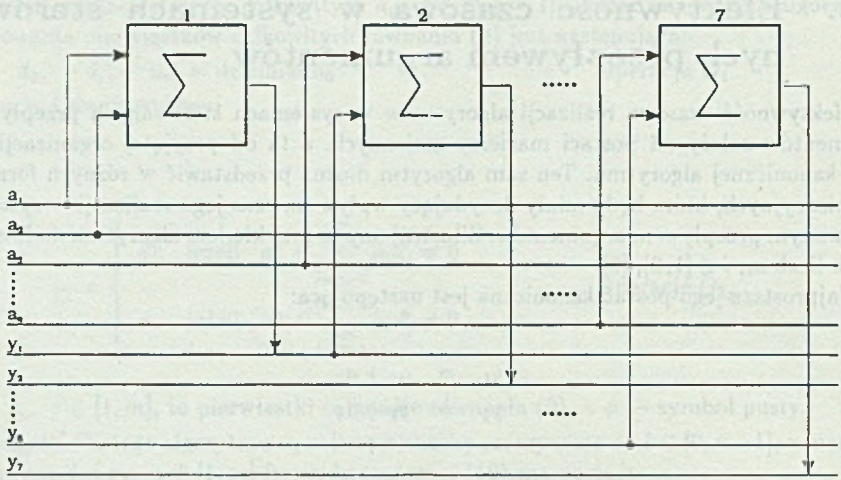
(13)

Macierz zmiennych o postaci (13) prowadzi do struktury systemu sterowanego przepływem algorytmów przedstawionego na rys. 7.

Jeżeli czas wykonania jednej operacji sumowania przyjąć równy 1 jednostce czasu, to całkowity czas realizacji algorytmu w systemie przedstawionym na rys. 7 będzie równy 7 jednostkom czasu, nie licząc czasu transmisji. Wynika to stąd, że w strukturze na rys. 7 nie wykorzystuje się możliwości prowadzenia procesów równoległych i w zasadzie jest to realizacja szeregowa procesu opisanego algorytmem.

Organizację algorytmu (12) można jednak przedstawić w inny sposób, a mianowicie:

$$\begin{aligned} y_1 &= a_1 + a_2 \\ y_2 &= a_3 + a_4 \\ y_3 &= a_5 + a_6 \\ y_4 &= a_7 + a_8 \\ y_5 &= y_1 + y_2 \\ y_6 &= y_3 + y_4 \\ y_7 &= y_5 + y_6 \end{aligned} \tag{14}$$



Rys. 7. Struktura systemu sterowanego przepływem argumentów realizującego algorytm (12)
Fig. 7. Structure of the system controlled by the flow of arguments and performing algorithm (12)

W tym przypadku macierz zmiennych będzie miała postać:

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
y_1					×		
y_2					×		
y_3						×	
y_4						×	
y_5							×
y_6							×
y_7							

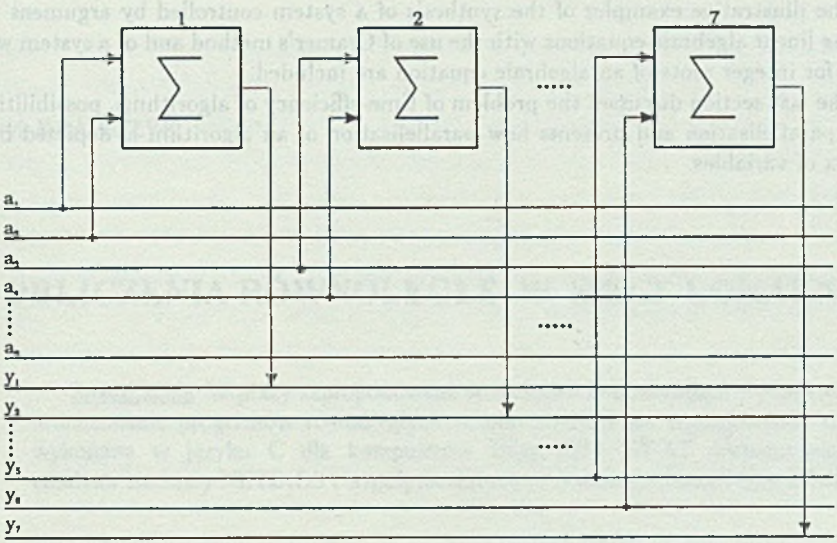
(15)

co prowadzi do systemu przedstawionego na rys. 8.

W tym przypadku łączny czas realizacji całego algorytmu będzie wynosił tylko 3 jednostki czasu. Wynika to stąd, że operacje obliczeniowe y_1, y_2, y_3, y_4 prowadzone są równolegle. Równolegle też prowadzone są operacje obliczenia y_5 i y_6.

Porównując te dwa czasy, a więc 7 jednostek czasu i 3 jednostki czasu, widzimy, że przez odpowiednią organizację procesu obliczeniowego czas realizacji całego algorytmu zmniejszył się dwukrotnie.

Ogólnie można wykazać, że jeżeli liczba danych byłaby równa 2^m, to całkowity czas realizacji algorytmu w strukturze rys. 7 byłby równy $2^m - 1$, a w strukturze rys. 8 byłby równy m, a więc $\log_2 2^m$. W tym przypadku można więc mówić o logarytmicznym skróceniu czasu obliczeń.



Rys. 8. Struktura systemu sterowanego przepływem argumentów realizująca algorytm (14)
Fig. 8. Structure of the system controlled by the flow of arguments and performing algorithm (14)

LITERATURA

[1] S. Węgrzyn, *Podstawy informatyki*, PWN, Warszawa 1978, s. 12.
[2] S. Węgrzyn, T. Czachórski, P. Vidal, J.C. Gille, *Algorytmy i procesy równoległe w informatyce, w biologicznych systemach rozwojowych i w systemach masowej obsługi*, Archiwum Informatyki Teoretycznej i Stosowanej, 1994 (w druku).

Recenzent: Doc. dr hab. inż. Adam Mrózek

Wpłynęło do Redakcji dnia 27.09.1993.

Abstract

Traditional von Neuman computers are systems controlled by flow of operations. The article considers two types of systems in which algorithms may be performed, namely: systems controlled by flow of operations and systems controlled by flow of arguments. Basic features of organization and structure of these systems are compared. We introduce the notions of canonical form of the algorithm and of its variable matrix and use them to syntesise the systems controlled by flow of arguments.

It is demonstrated that if the variable matrix of canonical form of the algorithm is the upper triangular one (main diagonal excluded) then it can be used in direct synthesis of the system controlled by argument flow and performing the corresponding algorithm.

The illustrative examples of the synthesis of a system controlled by argument flow, solving linear algebraic equations with the use of Cramer's method and of a system which seeks for integer roots of an algebraic equation are included.

The last section discusses the problem of time-efficiency of algorithms, possibilities of their parallelisation and presents how parallelisation of an algorithm is depicted by its matrix of variables.