ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1982

Seria: INFORMATYKA z. 3 Nr kol. 708

Eugeniusz WRÓBEL

Instytut Informatyki Czasu Rzeczywistego
Politechniki Śląskiej

STRUKTURY JEDNORODNE JEDNOWYMIAROWE

Streszczenie: Przez pojęcie struktury jednorodnej rozumie się jedno-, dwu- lub więcej wymiarowy iteracyjny układ składający się z identycznych kombinacyjnych lub sekwencyjnych komórek i typowych, regularnych połączeń między nimi. Struktury jednorodne zyskały na popularności dzięki rozwojowi technologii wytwarzania układów cyfrowych LSI.

Artykuł przedstawia sposób opisu pewnej klasy struktur jednowymiarowych bazujący na odwrotnej notacji polskiej oraz algorytm określający sposób realizowania w oparciu o takie struktury funkcji logicznych, w szczególności słabo określonych.

1. WPROWADZENIE

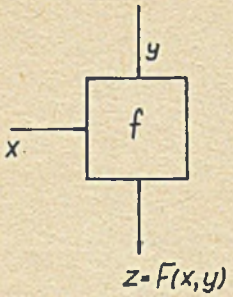
Rozwój technologii wytwarzania układów scalonych spowodował konieczność opracowania nowych, perspektywicznych koncepcji realizacji układów logicznych, bazujących na modułach dużej skali integracji. Jedna z nich polega na technologicznym wytworzeniu pewnych uniwersalnych struktur, które mogą być programowane w sposób umożliwiający realizację dowolnej funkcji kombinacyjnej lub sekwencyjnej określonej liczby zmiennych. Pewną klasę takich układów nazywa się strukturami jednorodnymi.

2. PODSTAWOWE OKREŚLENIA

Def. 1

Struktura jednorodna to jedno-, dwu- lub więcej wymiarowy iteracyjny układ składający się z identycznych kombinacyjnych lub sekwencyjnych komórek i typowych, regularnych połączeń między nimi.

Niniejsza praca zajmuje się strukturami jednorodnymi jednowymiarowymi, w których poszczególne komórki zdefiniowane są następująco:



Rys. 1. Komórka K struktury jednorodnej

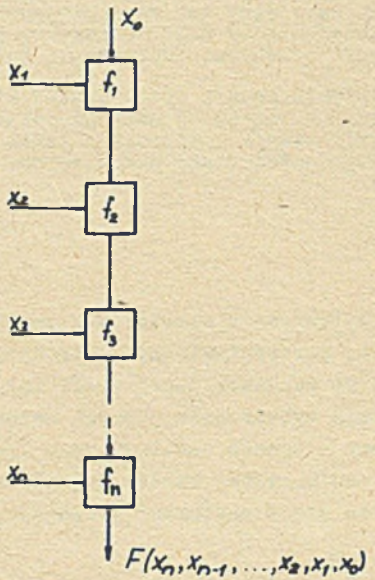
Def. 2

Komórka K to układ logiczny z dwoma wejściami x,y oraz z wyjściem z = F(x,y), gdzie F jest jedną z funkcji boolowskich dwóch zmiennych (rys. 1).

Struktura jednorodna jednowymiarowa (rys. 2), zwana także kaskadą, to układ składający się z n komórek K_i $(1 \leq i \leq n)$, przy czym wyjście z_i komórki i-tej połączone jest z wejściem y_{i+1} komórki i+1.

Def. 3

Baza struktury to zbiór funkcji dwóch zmienych logicznych, które mogą być realizowane przez każdą z komórek danej struktury.



Rys. 2. Struktura jednorodna jednowymiarowa

Def. 4

Programowanie (strojenie) struktury jednorodnej to proces, w którym pod działaniem sygnałów podawanych na wejścia programujące (bieguny strojenia) poszczególne komórki osiągają jeden ze swych stanów wewnętrznych, umożliwiający realizowanie którejś z funkcji należącej do bazy struktury.

Sposób programowania będzie różny w zależności od zastosowanej technologii oraz wybranej wersji rozwiązania struktury. Może być zatem zrealizowane np. przez podanie odpowiednich stanów logicznych na dodatkowe (nie

zaznaczone na rysunkach) wejścia programujące poszczególnych komórek, czy też na wejścia programujące oraz wejścia adresowe, wybierające programowaną komórkę. W innej wersji proces programowania struktury może być analogiczny do programowania pamięci typu PROM, EPROM itp.

3. METODA OPISU STRUKTUR JEDNOWYMIAROWYCH

Jednym z ważnych problemów związanych z realizacją układów logicznych w oparciu o struktury jednorodne jest opracowanie możliwie prostego, czytelnego i łatwego do wykorzystania przy implementowaniu określonych algorytmów na maszynę cyfrową sposobu ich opisu.

Rozpatrzmy strukturę z rys. 2. Funkcję realizowaną przez nią możemy zapisać:

$$F(x_n,x_{n-1},\ldots,x_2,x_1,x_0) = F_n(x_n,F_{n-1}(x_{n-1},\ldots,F_2(x_2,F_1(x_1,x_0))\ldots)) \tag{1}$$

W pracy [1] zaproponowano zastosowanie do opisu tego typu struktury odwrotnej notacji polskiej (ONP). Aby móc w pełni wykorzystać przedstawioną tam metodę, musimy wprowadzić operatory (symbole) wszystkich funkcji logicznych dwóch zmiennych binarnych (por. [11]). Stosowane powszechnie oznaczenia funkcji (zgodnie z [9]) uzupełniono tak, jak przedstawia to tablica 1.

Jeżeli przez f oznaczymy operator funkcji F dwóch zmiennych, to dla pojedynczej komórki struktury jednorodnej możemy zapisać:

$$F(x,y) = xfy$$

oraz w ONP

$$F(x,y) = xyf \tag{2}$$

Analogicznie, wyrażenie (1) można zapisać w sposób następujący:

$$F(x_n,x_{n-1},\ldots,x_2,x_1,x_0) = x_nf_n(x_{n-1}f_{n-1}\ldots(x_3f_3(x_2f_2(x_1f_1x_0))\ldots) \tag{3}$$

Przekształcając wyrażenie (3) zgodnie z regułami ONP otrzyma się:

$$F(x_n,x_{n-1},\ldots,x_2,x_1,x_0) = x_nx_{n-1}\ldots x_3x_2x_1x_0f_1f_2f_3\ldots f_{n-1}f_n \tag{4}$$

Odpowiedność wyrażenia (4) oraz struktury z rys. 2 jest widoczna.

Tablica 1

Funkcje logiczne dwóch zmiennych binarnych

	Postać funkcji	Oznaczenie wg [9]	Nazwa funkcji	Oznaczenie w ONP	b_i
f_0	0	0	zero		0
f_1	xy	xy	koniunkcja	$xy*$	1
f_2	$x\bar{y}$	$x.\Delta y$	zakaz przez y	$xy\Delta$	2
f_3	x	x	zmienna x	$xy\bigcirc$	3
f_4	$\bar{x}y$	$y.\Delta x$	zakaz przez x	$xy\nabla$	4
f_5	y	y	zmienna y	$xy\square$	5
f_6	$x\bar{y}+\bar{x}y$	$x \oplus y$	nierównoznaczność	$xy\oplus$	6
f_7	$x+y$	$x+y$	dysjunkcja	$xy+$	7
f_8	$\bar{x}\bar{y}$	$x\|y$	funkcja Peirce'a	$xy\|$	8
f_9	$xy+\bar{x}\bar{y}$	$x \sim y$	równoznaczność	$xy\sim$	9
f_{10}	$\bar{y}$	$\bar{y}$	negacja y	$xy\boxminus$	10
f_{11}	$x+\bar{y}$	$y \rightarrow x$	implikacja od y do x	$xy\leftarrow$	11
f_{12}	$\bar{x}$	$\bar{x}$	negacja x	$xy\odot$	12
f_{13}	$\bar{x}+y$	$x \leftarrow y$	implikacja od x do y	$xy\rightarrow$	13
f_{14}	$\bar{x}+\bar{y}$	$x\|y$	funkcja Sheffera	$xy\|$	14
f_{15}	1	1	jedynka		15

P r z y k ł a d 1

Przyjmując bazę struktury $B = \left\{f_0, f_1, f_2, \ldots, f_i, \ldots, f_{14}, f_{15}\right\}$, gdzie f_i podane są w tablicy 1, funkcję

$$P(x_4, x_3, x_2, x_1, x_0) = x_4\bar{x}_3 + x_4\bar{x}_2\bar{x}_1 + x_4\bar{x}_2\bar{x}_0 + x_4x_2x_1x_0$$

można przekształcić do jednej z poniższych postaci:

$$P(x_4, x_3, x_2, x_1, x_0) = x_4\left[\bar{x}_3 + (x_2 \oplus x_1x_0)\right]$$

$$P(x_4, x_3, x_2, x_1, x_0) = x_4\left[\bar{x}_3 + (\overline{x_2 \sim x_1x_0})\right]$$

$$P(x_4, x_3, x_2, x_1, x_0) = x_4\left\{\bar{x}_3 + \left[x_2 \sim (\bar{x}_1 + \bar{x}_0)\right]\right\}$$

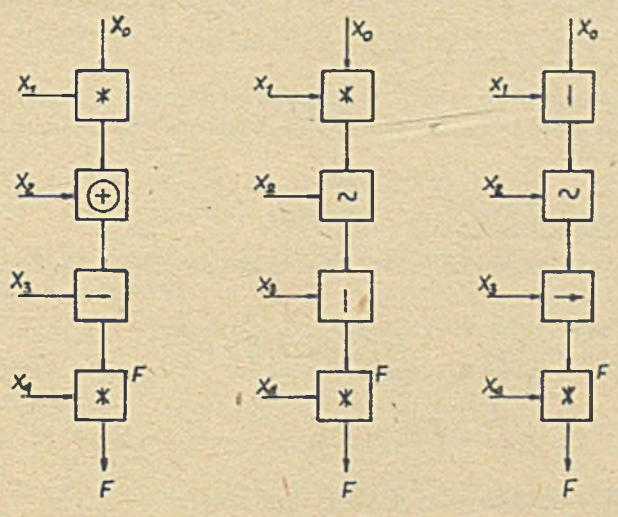
oraz w ONP:

$$P(x_4,x_3,x_2,x_1,x_0) = x_4x_3x_2x_1x_0 * \oplus - *$$

$$P(x_4,x_3,x_2,x_1,x_0) = x_4x_3x_2x_1x_0 * \sim | *$$

$$P(x_4,x_3,x_2,x_1,x_3) = x_4x_3x_2x_1x_0 | \sim - *$$

Z wyrażeń tych bezpośrednio wynikają realizacje przedstawione na rys. 3.



Rys. 3. Realizacje układowe zadanej w przykładzie funkcji

4. REALIZACJA UKŁADÓW LOGICZNYCH W OPARCIU O STRUKTURY JEDNORODNE JEDNOWYMIAROWE

W pracach [4, 5, 6, 7, 11] przedstawiono możliwości realizowania układów kombinacyjnych w oparciu o strukturę jednorodną jednowymiarową. W [5] wykazano, że ilość funkcji n zmiennych realizowanych w takiej strukturze wynosi $N_n = 0,8\ (3.6^{n-1} + 2)$ oraz zdefiniowano klasę tych funkcji. W pracy tej założono, iż bazę struktury tworzą wszystkie 16 funkcji dwóch zmiennych, W [6] udowodniono, że ograniczenie bazy struktury do kilku tylko funkcji dwóch zmiennych (np, $\bar{a}b$, ab, a, a+b, $\bar{a}$+b) nie zmniejszy ilości realizowanych w strukturze funkcji, wydłuży natomiast strukturę o co najwyżej jedną komórkę, znacznie upraszczając budowę pojedynczej komórki.

Przedstawione w ww. pracach metody syntezy tzn. algorytmy określające, jakie funkcje realizować muszą poszczególne komórki, aby struktura realizowała zadaną funkcję, w przeważającej większości nie nadają się do bez-

pośredniej implementacji maszynowej. W żadnej z tych pozycji nie rozważa się ponadto problemu wykorzystania przy syntezie układu stanów obojętnych występujących często w układach cyfrowych informatyki i automatyki. Właściwe wykorzystanie tych stanów umożliwia realizację w oparciu o strukturę jednowymiarową także takich układów, których nie da się zrealizować przy wykorzystaniu metod syntezy przedstawionych w cytowanej literaturze. Wynika to z konieczności przyjęcia w tych metodach a'priori stanów obojętnych jako jedynki lub zera logiczne.

Należy tutaj dodatkowo zaznaczyć, że funkcje słabo określone spotyka się w pracach inżynierskich stosunkowo często, czego potwierdzeniem jest m.in. zainteresowanie przemysłu wynikami prac przedstawionych w [10], a dotyczących tego typu funkcji.

Przedstawiona poniżej metoda realizowania układów logicznych w oparciu o strukturę jednowymiarową adresowana jest głównie dla funkcji słabo określonych.

4.1. Podstawy teoretyczne

Celem uproszczenia zapisu stosowane będą następujące oznaczenia i symbole:

F - skrócony zapis funkcji $F(x_n, x_{n-1}, \ldots, x_1, \ldots, x_2, x_1)$

$F(c)$ - skrócony zapis wyrażenia $F(x_n=c,\ x_{n-1}, \ldots, x_i, \ldots, x_2, x_1)$ np.:

$$F(1) = F(x_n = 1, x_{n-1}, \ldots, x_i, \ldots, x_2, x_1)$$

Σ - zbiór wszystkich składników jedynki,

Π - zbiór wszystkich czynników zera,

Σ^n - zbiór składników jedynki funkcji F wyrażonych w postaci liczb dziesiętnych. Przyjmuje się przy tym, że zmiennej x_n przypisana jest waga 2^{n-1}, natomiast zmiennej x_1 waga 2^0. Zatem:

$$\Sigma^n = \left\{ \sigma_i : \sigma_i \in \Sigma \wedge i \in A \wedge A \subset \left\{ 0, 1, \ldots, (2^n - 1) \right\} \right\}$$

Π^n - zbiór czynników zera funkcji F wyrażonych w postaci liczb dziesiętnych (jw.). Zatem:

$$\Pi^n = \left\{ \pi_i : \pi_i \in \Pi \wedge i \in A \wedge A \subset \left\{ 0, 1, \ldots, (2^n - 1) \right\} \right\}$$

Σ_o^{n-1} - podzbiór zbioru Σ^n utworzony przez usunięcie elementów większych lub równych 2^{n-1}

$$\Sigma_o^{n-1} = \left\{ \sigma_i : \sigma_i \in \Sigma^n \wedge i \in A \wedge A \subset \left\{ 0, 1, \ldots, (2^{n-1} - 1) \right\} \right\}$$

π_o^{n-1} - podzbiór zbioru π^n utworzony przez usunięcie elementów większych lub równych 2^{n-1}.

$$\pi_o^{n-1} = \left\{ \pi_i : \pi_i \in \pi^n \wedge i \in A \wedge A \subset \left\{0, 1, \ldots, (2^{n-1} - 1)\right\} \right\}$$

Σ_1^{n-1} - zbiór utworzony ze zbioru Σ^n przez usunięcie elementów mniejszych niż 2^{n-1} i odjęcie od pozostałych liczby 2^{n-1}

$$\Sigma_1^{n-1} = \left\{ \sigma_i : i = j - 2^{n-1} \wedge \sigma_j \in \Sigma^n \wedge j \in A \wedge A \subset \left\{2^{n-1}, (2^{n-1}+1), \ldots, (2^n-1)\right\} \right\}$$

π_1^{n-1} - zbiór utworzony ze zbioru π^n przez usunięcie elementów mniejszych niż 2^{n-1} i odjęcie od pozostałych liczby 2^{n-1}

$$\pi_1^{n-1} = \left\{ \pi_i : i = j - 2^{n-1} \wedge \pi_j \in \pi^n \wedge j \in A \wedge A \subset \left\{2^{n-1}, (2^{n-1}+1), \ldots, (2^n-1)\right\} \right\}$$

Można łatwo wykazać, że o ile zbiory Σ^n, π^n definiują funkcję F, to zbiory Σ_o^{n-1}, π_o^{n-1} definiują F(0), natomiast Σ_1^{n-1}, π_1^{n-1} funkcję F(1).

Dla dalszych rozważań zakładamy, że dana jest funkcja n zmiennych F zdefiniowana przez zbiory:

$$\Sigma^n = \left\{ \sigma_1, \sigma_2, \ldots, \sigma_j, \ldots, \sigma_m \right\}$$

$$\pi^n = \left\{ \pi_1, \pi_2, \ldots, \pi_k, \ldots, \pi_l \right\}$$

Należy znaleźć wyrażenie logiczne odpowiadające zadanej funkcji F w postaci:

$x_n x_{n-1} \ldots x_2 x_1 x_o b_1 b_2 \ldots b_{n-1} b_n$ gdzie $x_o \in [0,1]$ i jest parametrem, z którego bezpośrednio wynika realizacja funkcji F w strukturze jednorodnej jednowymiarowej.

Bazę struktury przyjmujemy $B = \left\{ f_1, f_4, f_5, f_7, f_9, f_{13} \right\}$, gdzie f_i zdefiniowane są w tablicy 1. Stosując wprowadzone oznaczenia można także napisać:

$$B = \left\{ *, \nabla, \square, +, \sim, - \right\} \tag{5}$$

Łatwo sprawdzić, że w oparciu o powyższe 6 funkcji każda funkcja dwóch zmiennych może być przedstawiona w strukturze jednowymiarowej o długości 2, co stanowi podstawowe kryterium wyboru bazy [4].

Poniższe stwierdzenia są podstawą omawianego algorytmu i podają szczegółowe warunki dekomponowania funkcji logicznej zgodnie z przyjętą bazą struktury. Dowody stwierdzeń są stosunkowo proste i ze względu na ograniczone ramy artykułu pomija się je.

Stwierdzenie 1

Funkcja F może być przedstawiona w postaci wyrażenia $F = x_n + y$ wtedy i tylko wtedy, gdy $\pi_1^{n-1} = \Phi$, przy czym $y=y(x_{n-1},\dots,x_2,x_1)$ i jest zdefiniowane przez zbiory:

$$\Sigma^{n-1} = \Sigma_0^{n-1}$$

$$\pi^{n-1} = \pi_0^{n-1}$$

Stwierdzenie 2

Funkcja F może być przedstawiona w postaci wyrażenia $F = \bar{x}_n + y$ wtedy i tylko wtedy, gdy $\pi_0^{n-1} = \Phi$, przy czym $y=y(x_{n-1},\dots,x_2,x_1)$ i jest zdefiniowane przez zbiory:

$$\Sigma^{n-1} = \Sigma_1^{n-1}$$

$$\pi^{n-1} = \pi_1^{n-1}$$

Stwierdzenie 3

Funkcja F może być przedstawiona w postaci wyrażenia $F = x_n y$ wtedy i tylko wtedy, gdy $\Sigma_0^{n-1} = \Phi$, przy czym $y=y(x_{n-1},\dots,x_2,x_1)$ i jest zdefiniowane przez zbiory:

$$\Sigma^{n-1} = \Sigma_1^{n-1}$$

$$\pi^{n-1} = \pi_1^{n-1}$$

Stwierdzenie 4

Funkcja F może być przedstawiona w postaci wyrażenia $F = \bar{x}_n y$ wtedy i tylko wtedy, gdy $\Sigma_1^{n-1} = \Phi$, przy czym $y=y(x_{n-1},\dots,x_2,x_1)$ i jest zdefiniowane przez zbiory:

$$\Sigma^{n-1} = \Sigma_0^{n-1}$$

$$\pi^{n-1} = \pi_0^{n-1}$$

Stwierdzenie 5

Funkcja F może być przedstawiona w postaci wyrażenia $F = y$ wtedy i tylko wtedy, gdy $\Sigma_1^{n-1} \cap \pi_0^{n-1} = \Phi$ i $\Sigma_0^{n-1} \cap \pi_1^{n-1} = \Phi$, przy czym $y=y(x_{n-1},\dots,x_2,x_1)$ i jest zdefiniowane przez zbiory:

$$\Sigma^{n-1} = \Sigma_0^{n-1} \cup \Sigma_1^{n-1}$$

$$\pi^{n-1} = \pi_0^{n-1} \cup \pi_1^{n-1}$$

Stwierdzenie 6

Funkcja F może być przedstawiona w postaci wyrażenia $F = x_n y + \bar{x}_n \bar{y}$ wtedy i tylko wtedy, gdy $\pi_1^{n-1} \cap \pi_0^{n-1} = \Phi$ i $\Sigma_0^{n-1} \cap \Sigma_1^{n-1} = \Phi$, przy czym $y=y(x_{n-1},\ldots,x_2x_1)$ i jest zdefiniowane przez zbiory:

$$\Sigma^{n-1} = \Sigma_1^{n-1} \cup \pi_0^{n-1}$$

$$\pi^{n-1} = \pi_1^{n-1} \cup \Sigma_0^{n-1}$$

4.2. Opis algorytmu

Metoda syntezy polega na sprawdzaniu możliwości dekomponowania funkcji kolejno dla poszczególnych zmiennych wejściowych x_i, począwszy od zmiennej x_n, której przypisana jest najwyższa waga. Danymi wejściowymi są zbiory Σ^n oraz π^n definiujące funkcję, oraz parametr n, określający ilość zmiennych.

Na początku sprawdza się czy ilość zmiennych wynosi 2. Jeżeli tak, to w związku z tym, iż każda funkcja 2 zmiennych realizowana jest w strukturze jednowymiarowej, oblicza się b_2 b_1 na podstawie tablicy 2. Jeżeli ilość zmiennych jest większa niż dwa, obliczenia przeprowadza się rozpoczynając od znalezienia zbiorów Σ_1^{n-1}, π_1^{n-1}, Σ_0^{n-1}, π_0^{n-1}. Następnie przeprowadza się sprawdzanie możliwości dekomponowania funkcji w następującej kolejności:

- czy $F = y$, ($b_n = 5$) warunki określone w stw. 5,
- czy $F = x_n y$, ($b_n = 1$) warunki określone w stw. 3,
- czy $F = \bar{x}_n y$, ($b_n = 4$) warunki określone w stw. 4,
- czy $F = x_n + y$, ($b_n = 7$) warunki określone w stw. 1,
- czy $F = \bar{x}_n + y$, ($b_n = 13$) warunki określone w stw. 2,
- czy $F = x_n \cdot y$, ($b_n = 9$) warunki określone w stw. 6.

W przypadku, gdy funkcja zdefiniowana przez zbiory Σ^n, π^n może być przedstawiona za pomocą jednego z ww. wyrażeń, zapamiętuje się b_n, a następnie definiuje się zbiory Σ^{n-1}, π^{n-1}, określające funkcję n-1 zmiennych, którą poddaje się próbie dekompozycji w następnym kroku. Jeżeli ta nowa funkcja może być przedstawiona za pomocą jednego z ww. wyrażeń, zapamiętuje się b_{n-1}, a następnie definiuje się zbiory Σ^{n-2}, π^{n-2}, określające funkcję n-2 zmiennych, którą poddaje się próbie dekompozycji w kolejnym kroku. Procedurę powtarza się aż do sytuacji, gdy ilość zmiennych wynosi dwa lub też w którymś z kroków okazało się, że funkcja nie może być dalej dekomponowana. W pierwszym przypadku określa się jeszcze b_2 b_1 i wyprowadza się otrzymane rozwiązanie w postaci wyrażenia $x_n x_{n-1} \ldots x_2 x_1 x_0 b_1 b_2 \ldots b_{n-1} b_n$. Jeżeli w i-tym kroku funkcja okaże się niedekom-

ponowana, wyprowadza się $x_n x_{n-1} \ldots x_i b_{i+1} \ldots b_{n-1} b_n$ oraz zbiory Σ^i, π^i. W takiej sytuacji można przeprowadzić zmianę kolejności zmiennych $x_i, \ldots, x_2 x_1$, zdefiniować nowe zbiory Σ^i, π^i i kontynuować obliczenia lub też zrealizować funkcję F w strukturze dwuwymiarowej. W szczególnych przypadkach struktura może ulec skróceniu o jedną lub dwie komórki:

- o jedną komórkę, gdy zbiór Σ^2 definiuje funkcję, która należy do bazy struktury lub też dla i-tej komórki $b_i = 5$,
- o dwie komórki, gdy zbiór Σ^2 jest zbiorem pustym lub zawiera cztery elementy.

Tablica 2

Tablica określająca b_2 b_1

Zbiór Σ^2	Realizowana funkcja	b_2	b_1	x_o	Uwagi
	0	-	-	0	struktura skrócona
		5	5	0	struktura pełna
$\{0\}$	$\bar{x}_2\bar{x}_1$	4	13	0	
$\{1\}$	$\bar{x}_2 x_1$	4	-	x_1	struktura skrócona
		4	7	0	struktura pełna
$\{2\}$	$x_2\bar{x}_1$	1	13	0	
$\{3\}$	$x_2 x_1$	1	-	$\bar{x}_1$	struktura skrócona
		1	7	0	struktura pełna
$\{0,1\}$	$\bar{x}_2$	13	5	0	
$\{0,2\}$	$\bar{x}_1$	5	13	0	
$\{0,3\}$	$x_2 x_1 + \bar{x}_2\bar{x}_1$	9	-	x_1	struktura skrócona
		9	7	0	struktura pełna
$\{1,2\}$	$\bar{x}_2 x_1 + x_2\bar{x}_1$	9	13	0	
$\{1,3\}$	x_1	5	7	0	
$\{2,3\}$	x_2	7	5	0	
$\{0,1,2\}$	$\bar{x}_2 + \bar{x}_1$	13	13	0	
$\{0,1,3\}$	$\bar{x}_2 + x_1$	13	7	0	
$\{0,2,3\}$	$x_2 + \bar{x}_1$	7	13	0	
$\{1,2,3\}$	$x_2 + x_1$	7	7	0	
$\{0,1,2,3\}$	1	-	-	1	struktura skrócona
		5	5	1	struktura pełna

4.3. Przykłady

Przykład 1

Funkcja $F=F(x_6,x_5,\ldots,x_1)$ jest zdefiniowana przez zbiory:

$$\Sigma^6 = \{16,17,23,24,25,31\}$$

$$\pi^6 = \{0,2,3,8,9,10,19,20,22,28,29,30,51,52,53,54,61\}$$

Funkcję F zrealizować w oparciu o strukturę jednowymiarową.

R o z w i ą z a n i e:

$i=6,\ 2^{i-1} = 31$

$\Sigma_1^5 = \{\ \}$ $\quad \Sigma_0^5 = \{16,17,23,24,25,31\}$

$\pi_1^5 = \{19,20,21,22,29\}$ $\quad \pi_0^5 = \{0,2,3,8,9,10,19,20,22,28,29,30\}$

Spełnione są warunki stwierdzenia 4, zatem $\underline{b_6 = 4}$ oraz:

$\Sigma^5 = \{16,17,23,24,25,31\}$

$\pi^5 = \{0,2,3,8,9,10,19,20,22,28,29,30\}$

$i=5,\quad 2^{i-1} = 16$

$\Sigma_1^4 = \{0,1,7,8,9,15\}$ $\quad \Sigma_0^4 = \{\ \}$

$\pi_1^4 = \{3,4,6,12,13,14\}$ $\quad \pi_0^4 = \{0,2,3,8,9,10\}$

Spełnione są warunki stwierdzenia 3, zatem $\underline{b_5=1}$ oraz:

$\Sigma^4 = \{0,1,7,8,9,15\}$ $\quad \pi^4 = \{3,4,6,12,13,14\}$

$i=4,\quad 2^{i-1} = 8$

$\Sigma_1^3 = \{0,1,7\}$ $\quad \Sigma_0^3 = \{0,1,7\}$

$\pi_1^3 = \{4,5,6\}$ $\quad \pi_0^3 = \{3,4,6\}$

Spełnione są warunki stwierdzenia 5, zatem $\underline{b_4=5}$ oraz

$\Sigma^3 = \{0,1,7\}$ $\quad \pi^3 = \{3,4,5,6\}$

$i=3\quad 2^{i-1}=4$

$\Sigma_1^2 = \{3\}$ $\quad \pi_1^2 = \{0,1,2\}$ $\quad \Sigma_0^2 = \{0,1,\}$ $\quad \pi_0^2 = \{3\}$

Spełnione są warunki stwierdzenia 6, zatem $\underline{b_3 = 9}$ oraz: $\Sigma^2 = \{3\}$

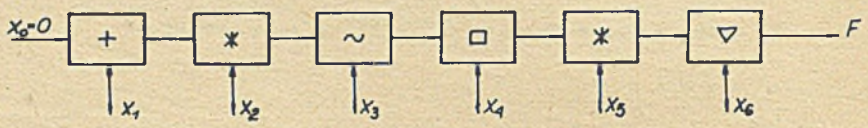
$i=2$

Dla $\Sigma^2 = \{3\}$ z tablicy można znaleźć: $\underline{b_2=4}$, $\underline{b_1=7}$, $x_0=0$.

Funkcję F zdefiniowaną w temacie można zatem przedstawić w postaci wyrażenia:

$$F(x_6,x_5,\ldots,x_1) = x_6x_5x_3x_2x_1x_0b_1b_2b_3b_4b_5b_6 = x_6x_5x_4x_3x_2x_1x_0 + * \sim \square * \nabla$$

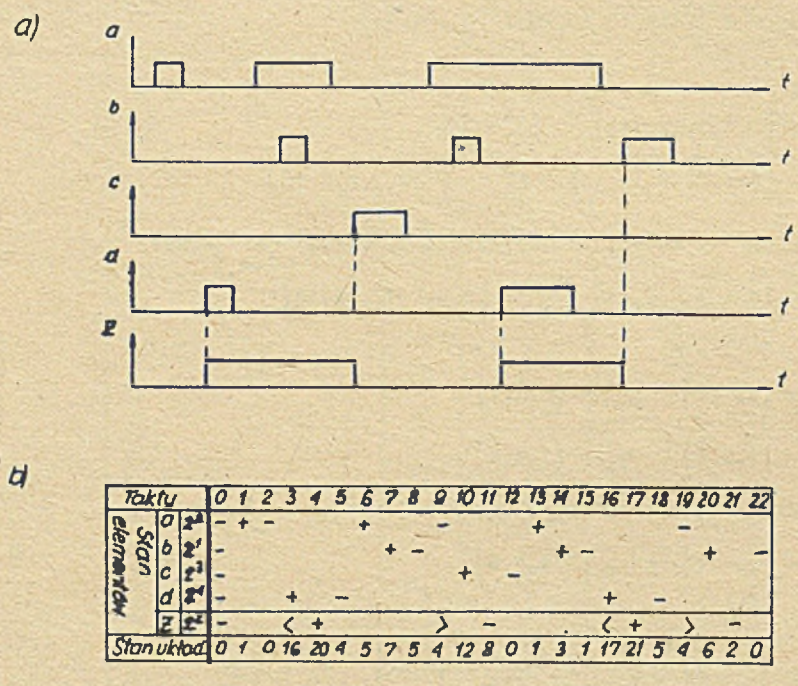
Realizację funkcji F w oparciu o strukturę jednowymiarową przedstawia rys. 4.



Rys. 4. Realizacja funkcji F z przykładu 1

Przykład 2

W oparciu o strukturę jednowymiarową zrealizować układ działający wg wykresu czasowego przedstawionego na rys. 5a. Układ posiada wejścia a,b, c,d i steruje wyjściem Z.



Rys. 5. Wykres czasowy oraz tablica kolejności łączeń do przykładu 2

R o z w i ą z a n i e:

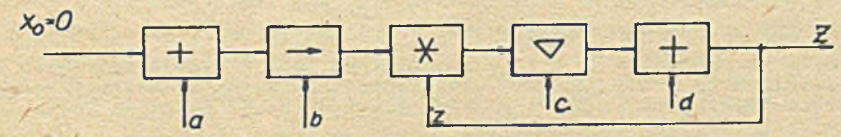
Na podstawie wykresu czasowego określamy tablicę kolejności łączeń (rys. 5b) [9], z której bezpośrednio znajdujemy zbiory Σ^5, π^5 definiujące funkcję Z:

$$\Sigma^5 = \{4,5,7,16,17,20,21\} \qquad \pi^5 = \{0,1,2,3,6,8,12\}$$

przy czym zmiennym a,b,z,c,d przypisane są kolejno wagi $2^0, 2^1, 2^2, 2^3, 2^4$. Prowadząc dalsze obliczenia wg przedstawionej wyżej metody szczegółowo zilustrowanej w poprzednim przykładzie, otrzymuje się wyrażenie:

$$Z = dczbax_0b_1b_2b_3b_4b_5 = dczbax_0 + - * \nabla +, \text{ gdzie } x_0 = 0,$$

z którego bezpośrednio wynika realizacja przedstawiona na rys.6.



Rys. 6.Realizacja funkcji Z z przykładu 2

5. UWAGI KOŃCOWE

Przy obecnym rozwoju technologii wytwarzania elementów scalonych produkcja układu LSI, zawierającego strukturę jednorodną umożliwiającą realizację funkcji kilkunastu, a nawet kilkudziesięciu zmiennych, nie powinna nastręczać trudności, Duża regularność (powtarzalność) budowy struktury umożliwia uzyskanie dużej gęstości upakowania elementów na płytce układu. Jest to w chwili obecnej [2] jedno z najważniejszych kryteriów powodujących zaliczenie układu do grupy układów katalogowych.

Wydaje się, że duża część układów logicznych projektowana indywidualnie, dla małoseryjnych lub wręcz jednostkowych zastosowań, a składana dotychczas z elementów małej i średniej skali integracji, mogłaby być realizowana w oparciu o struktury jednorodne.

Przedstawiona metoda opisu struktury jednowymiarowej, wykorzystująca odwrotną notację polską, w sposób zasadniczy upraszcza zapis funkcji realizowanej przez taką strukturę i to zarówno w obliczeniach ręcznych, jak i przy wykorzystaniu do prac projektowych maszyny cyfrowej.

Omówiony algorytm realizowania układów logicznych w oparciu o strukturę jednowymiarową adresowany jest głównie do funkcji słabo określonych, jednak poprawne wyniki otrzymuje się także dla funkcji w pełni określonych.

LITERATURA

[1] BERGES G.A.: Use of Postix Notation for the Descripton of Microcellular Arrays. IEEE Trans. on Comp. No 2/77 pp. 185-191.

[2] CARR W.N., MIZE J.P.: Projektowanie i zastosowanie układów MOS wielkiej skali integracji. WNT, Warszawa 1976.

[3] DUNCAN F.G., ZISSES D., WALLS M.: A Postifix Notation for Logic Circuits. The Computer Journal Vol. 18, No 1, pp. 63-69.

[4] MAJEWSKI W., ALBICKI A. (red.): Projektowanie cyfrowych układów telekomunikacyjnych. WKŁ, Warszawa 1977.

[5] MAITRA K.K.: Cascaded Switching Networks of Two-Input Flexible Cells. IRE Trans. on El. Comp. No 2/62, pp. 136-143.

[6] MINNICK R.C.: Cutpoint Cellular Logic IEEE Trans on El. Comp. No 6/64, pp. 685-698.

[7] MUKHOPADHYAY A. (edited by): Recent Developments in Switching Theory. Academic Press, New York and London 1972.

[8] PAPAKONSTANTINON G.K.: A Synthesis Method for Cutpoint Cellular Arrays. IEEE Trans. on Comp. No 12/1972, pp. 1286-1292.

[9] SIWIŃSKI J.: Układy przełączające w automatyce. WNT, Warszawa 1968.

[10] WRÓBEL E., MAŁYSIAK H. i in.: Komputerowa implementacja wybranych algorytmicznych metod minimalizacji funkcji logicznych. Materiały Sympozjum "Zastosowanie EMC w pracach inżynierskich" Katowice 1979.

[11] WRÓBEL E.: Realizacja układów przełączających w oparciu o struktury jednorodne. Zeszyty Naukowe Pol.Sl., s. Automatyka, z. 49, Gliwice 1979.

Wpłynęło do Redakcji: 17.11.1980 r.
W ostatecznej formie przyjęto: 14.01.1981 r.

Recenzent: Prof.dr inż. Jerzy Bromirski

ОДНОМЕРНЫЕ ОДНОРОДНЫЕ СРЕДЫ

Р е з ю м е

Цифровые автоматы на базе однородных сред представляют собой новый класс схем, состоящий из однотипных элементов-ячеек комбинационных или с памятью, соединенных друг с другом одинаковым образом. Статья посвящена некоторым вопросам осуществления логических систем на основе одномерных однородных сред, а также корреспонденции между выраженными в обратной польской нотации булевыми функциями и этими средами.

ONE-DIMENSIONAL CELLULAR ARRAYS

S u m m a r y

A cellular logic array is a one-, two-, or many-dimensional iterative arrangement of identical combinational or sequential cells with a regular interconnection pattern on the cells. Iterative arrays are becoming increasingly popular with the development of LSI.

The paper presents some problems of synthesis of one-dimensional cellular arrays and a correspondence between arbitrary switching functions expressed in Polish postfix notation and this one-dimensional arrays.