

Jacek LIPOWSKI

Instytut Informatyki Czasu Rzeczywistego
Politechniki Śląskiej

MODUŁOWA REALIZACJA SYSTEMÓW CYFROWYCH

Streszczenie. W pracy przedstawiono najważniejsze koncepcje Uniwersalnych Bloków Logicznych. Zaprezentowano spotykane w literaturze algorytmy syntezy modułowych struktur sekwencyjnych układów synchronicznych.

Przedstawiono nowy algorytm syntezy jednorodnych struktur sekwencyjnych, synchronicznych automatów Moore'a w oparciu o zaproponowany moduł D-UML_c.

1. WSTĘP

Aktualne osiągnięcia w produkcji elementów scalonych średniej i dużej skali integracji mają istotny wpływ na powstawanie nowych poglądów na całokształt procesu wytwarzania struktur i podzespołów systemów cyfrowych, a w szczególności na tworzenie nowych metod projektowania układów logicznych.

Jedną z koncepcji projektowych jest realizacja modułowych struktur układów logicznych.

Problem ten można rozpatrywać zarówno w aspekcie zaproponowania możliwie optymalnej struktury uniwersalnego bloku logicznego (UBL) (realizującego np. możliwie największą ilość funkcji dużej ilości zmiennych, zbudowanego w pojedynczym elemencie scalonym z najmniejszą możliwą ilością zewnętrznych wyprowadzeń) jak też podania algorytmów, które dla zaproponowanego UBL lub gotowego, produkowanego układu scalonego średniej skali integracji dadzą optymalną strukturę układu. Kryteriami optymalności mogą być przy tym np.: ilość użytych modułów, regularność struktury, ilość połączeń między modułami, łatwa testowalność układu, straty energetyczne układu w eksploatacji itd. Pod pojęciem regularności rozumie się np. regularność połączeń, ilość rodzajów użytych modułów i innych elementów.

Szczególnie uprzywilejowanym podejściem przy realizacji modułowych struktur układów cyfrowych jest zaproponowanie w oparciu o wybrany moduł lub zbiór modułów, prostych inżynierskich metod projektowych przy założeniu, że projektant nie ma specjalnego przygotowania z metodyki projektowania układów logicznych.

W pracy omówiono niektóre z ważniejszych propozycji UBL (w szczególności ich "lasy" tzw. uniwersalnych modułów logicznych (UML)) oraz niektóre

znane metody realizacji sekwencyjnych układów logicznych w oparciu o proponowane przez różnych autorów lub produkowane moduły. W rozdziale 4 przedstawiono nową metodę projektowania dowolnych sekwencyjnych synchronicznych automatów Moore'a w oparciu o zaproponowany moduł D-UMLc. Metoda ta jest przydatna do bezpośredniego, inżynierskiego zastosowania.

2. UNIWERSALNE BLOKI LOGICZNE

Uniwersalnym blokiem logicznym (UBL) nazywać będziemy element zrealizowany jako układ cyfrowy średniej (lub małej) skali integracji, w oparciu o który zrealizować można każdy układ logiczny w strukturze jednorodnej, tzn. w oparciu o ten tylko blok logiczny, pseudojednorodnej, tzn. w oparciu o 1 lub 2 rodzaje uniwersalnych bloków logicznych oraz podstawowe bramki logiczne (SSI) lub w strukturze mieszanej, tzn. w oparciu o 3 lub więcej uniwersalnych bloków logicznych oraz podstawowe bramki logiczne (SSI). Specyficznym rodzajem UBL jest uniwersalny moduł logiczny (UML).

Uniwersalnym modułem logicznym z zmiennych (UML-n) jest układ logiczny który może zrealizować każdą z 2^{2^n} funkcji logicznych n zmiennych.

W [1] po raz pierwszy zaproponowano UML jako wygodny podzespół do budowy układów cyfrowych.

W [2] wykorzystano do zdefiniowania UML dekompozycję Shannona funkcji kombinacyjnej n zmiennych w formie:

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2^{n-1}} x_1^{i_1} x_{n-1}^{i_{n-1}} f(i_1, \dots, i_{n-1}, x_n) \quad (1)$$

gdzie:

$$x_j^0 = \bar{x}_j, x_j^1 = x_j, j = 1, 2, \dots, n-1, \text{ a}$$

$i_1 i_2 \dots i_{n-1}$ jest binarną reprezentacją i.

Prowadzi to bezpośrednio do zdefiniowania tzw. T-UML-n (Tree-UML-n) jako modułu realizującego taką dekompozycję. Techniczną realizacją T-UML-n jest multiplexer.

W [2] zaproponowano strukturę tzw. Q-UML-n, który dla $n \geq 6$ daje minimalną ilość wejść i wyjść.

Szczególnym przypadkiem T-UML-n jest moduł posiadający $n+3$ wejść/wyjść. Przedstawiono go w [2]. Sygnały programujące wprowadzane są tu szeregowo do rejestru przesuwanego. Jest to tzw. SUML-n (szeregowy UML).

W wielu pracach, np. [4], [5], dyskutowano problem optymalnych struktur UML.

Jako uniwersalne bloki logiczne dla syntezy układów kombinacyjnych wielu autorów proponowało też dekodery (np. [18]), sumatory i półsumatory (np. [10]) jak też komparatory czy inne bloki cyfrowe.

W [19] pokazano, jak każdą funkcję kombinacyjną można przekształcić w funkcję symetryczną dodając pewne nadmiarowe argumenty.

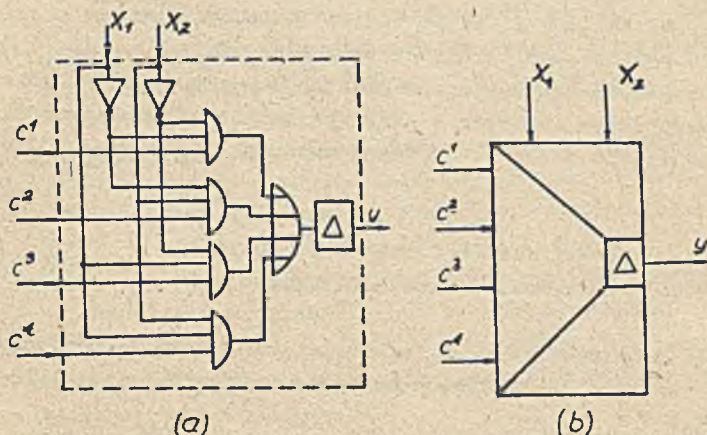
W [2], [3] i szeregu innych prac zaproponowano szereg modułów dla realizacji funkcji symetrycznych kombinacyjnych jak i symetrycznej dekompozycji funkcji sekwencyjnych.

Szereg modułów zaproponowano dla realizacji automatów sekwencyjnych. Podstawowe algorytmy bazują przy tym na module zaproponowanym przez Ullmana-Weinera [11] i Newborna [7], wykorzystywanym w licznych algorytmach np.: Hopcrofta [12], Arnolda [3], Williama [13] i innych.

Moduł ten jest kaskadowym połączeniem modułu T-UML-n z elementem pamięci (odpowiednim opóźnieniem czasowym dla realizacji asynchronicznych, synchronicznym przerzutnikiem D, J-K itd. dla realizacji synchronicznych).

Stał się on podstawą szczególnie algorytmów dla syntezy automatów synchronicznych.

Realizację i symbol modułu dla 2 zmiennych adresowych przedstawia rysunek 1.



Rys. 1. Schemat ideowy (a) i symbol (b) modułu T-UML_c-2

Funkcję, jaką realizuje wyjście proste tego modułu, można przez analogię do (1) zapisać jako:

$$y^{t+1} = \sum_{i=1}^{2^n} \left[(x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}) \cdot c^i \right]^t \quad (2)$$

gdzie:

$$x_j^0 = \bar{x}_j, x_j^1 = x_j, j = 1, \dots, n$$

$(i_1 i_2 \dots i_n)$ stanowi binarną reprezentację i

$t, t+1$ - poprzednia i następna chwila czasowa.

Moduł ten często w literaturze nazywany jest modułem AOD (And Or Delay). Nawiązując do oznaczenia modułu T-UML-n nazwijmy go T-UML_c-n (T-UML-n Clocked). Moduł T-UML_c produkowany jest seryjnie jako 4x(T-UML_c-1) w 1 elemencie MSI (Siemens 74298/84298).

Wielu autorów zaproponowało swoje rozwiązania uniwersalnych bloków dla synchronicznych realizacji struktur cyfrowych. Znaczna część z nich bazuje jednak na propozycji Ullmana-Weinera.

W wielu pracach proponowano też liczniki i rejestry przesuwne jako podstawowe uniwersalne bloki do jednorodnej lub pseudojednorodnej syntezy automatów synchronicznych, np. [14].

Specyficzne moduły zaproponowano dla modułowej syntezy automatów asynchronicznych. Szczególnie ciekawą propozycję dla realizacji jednorodnej wprost z grafu układu jest zaproponowany w [6], a produkowany obecnie seryjnie przez firmę SESCOSEM moduł CUSA (Cellule Universelle pour Sequences Asynchrones).

W [15] zaproponowano rodzinę 5 modułów, w oparciu o które zrealizować można każdy automat asynchroniczny dany siecią Petriego (moduły: JUNCTION, DISTRIBUTION, TRANSFERT, SELECTION, ATTRIBUTION) wprost z sieci.

W wielu pracach, np. [16], określa się rodziny modułów proponowanych lub produkowanych realizujących działania wynikające z funkcji przypisanych poszczególnym symbolom w sieciach działań (klatki warunkowe i operacyjne) czy innych podobnych formach opisu programu pracy układu.

3. REALIZACJA SEKWENCYJNYCH SYSTEMÓW CYFROWYCH W OPARCIU O UNIWERSALNE BLOKI LOGICZNE

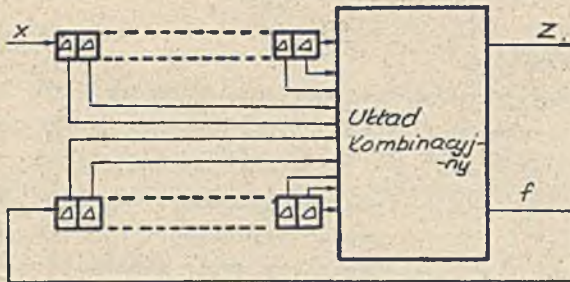
Poniżej omówiono niektóre istniejące algorytmy jednorodnej i pseudojednorodnej syntezy sekwencyjnych automatów synchronicznych, opartych o moduł T-UML_c.

O procedurach realizacji sekwencyjnych automatów synchronicznych na bazie innych uniwersalnych bloków logicznych oraz o niektórych procedurach modułowej realizacji automatów asynchronicznych wspomniano już w rozdziale 2.

W [9] podano technikę realizacji każdego automatu synchronicznego Moore'a w oparciu o strukturę z jednym układowym sprzężeniem zwrotnym f i wyjściem Z (patrz rys. 2), przy czym sygnały wejściowe x i sygnały sprzężeń zwrotnych wprowadzane są do bloku kombinacyjnego przez szereg (n) opóźnień (elementarnych przerzutników S-R) lub synchroniczny rejestr przesuwający. Przy pomocy tworzonych tzw. grafu implikacji podano metodę wyznaczania funkcji f dla każdego ze stanów dowolnej, zadanej tablicy przejść.

W [3] zaproponowano w pełni regularną (tak pod względem jednego rodzaju modułu jak i połączeń między modułami) strukturę modułową służącą do

realizacji dowolnego układu synchronicznego Moore'a wg koncepcji z rysunku 2.



Rys. 2. Struktura realizacji automatu Moore'a z jedną układową pętlą sprzężenia zwrotnego

Struktura drzewiaste (kanoniczna i uproszczona), jako jedna z podstawowych koncepcji syntezy jednorodnych, modułowych struktur automatów sekwencyjnych (grupa I metod) stała się podstawą wielu algorytmów i przedmiotem wielu prac, np. [3], [10]. Algorytmy te dają w wyniku rozwiązania najbardziej regularne.

Druga grupa (II) metod syntezy modułowych struktur układów synchronicznych polega na sprowadzeniu automatu sekwencyjnego do struktury niedeterministycznej, jednorodnej w sensie jej realizacji w oparciu o T-UML_c ([7], [11], [12], [13]).

Metody te nie dają już rozwiązań tak regularnych jak struktury drzewiaste (nieregularna sieć połączeń), choć z reguły mniejszą ilość użytych modułów.

Charakterystyczny dla obu ww. grup metod jest stosunkowo skomplikowany i czasochłonny algorytm projektowy.

Trzecią grupą (III) metod są metody syntezy pseudojednorodnej lub mieszanej oparte na T-UML_c jako module podstawowym. Cechą charakterystyczną tej grupy jest prostota algorytmów projektowych.

W [21] zaproponowano metodę realizacji układów sekwencyjnych Moore'a wprost z grafu układu w oparciu o moduły T-UML_c oraz bramki OR. Metoda pozwala na bezpośrednie odwzorowanie struktury grafu w układ połączeń pomiędzy modułami, przy czym każdemu modułowi przypisuje się jeden stan układu (węzeł grafu).

W [8] zaproponowano metodę realizacji sekwencyjnych układów Moore'a w oparciu o T-UML_c i o bramki OR, przy czym każdemu modułowi przypisano zmienną stanu (kodowanie stanów), zmniejszając znacznie ilość użytych modułów. Konsekwencją wykorzystania kodowania jest jednak wprowadzenie do struktury indywidualnie dla każdego układu projektowanego dekodera oraz kombinacyjnego bloku (bramki OR), odwzorowującego sygnały stanów na zmienne programujące modułów T-UML_c.

4. METODA SYNTEZY JEDNORODNYCH MODUŁOWYCH STRUKTUR SEKWENCYJNYCH
 SYNCHRONICZNYCH AUTOMATÓW MOORE'A OPISANYCH GRAFEM
 W OPARCIU O ZAPROPONOWANY MODUŁ D-UML_c

Definicje i oznaczenia

Rozważmy sekwencyjny synchroniczny automat Moore'a o k stanach wewnętrznych, n wejściach, m wyjściach i zdefiniujmy następujące związane z nim, wielkości:

$$\begin{aligned} (X'_1, \dots, X'_1, \dots, X'_n) &= \tilde{X} && \text{wektor wejściowy; } X'_i \in \{X_1, \bar{X}_1\} \\ (Y'_1, \dots, Y'_1, \dots, Y'_m) &= \tilde{Y} && \text{wektor wyjściowy; } Y'_i \in \{Y_1, \bar{Y}_1\} \\ \{Q_1, \dots, Q_1, \dots, Q_k\} &= {}^*Q && \text{zamknięty zbiór stanów} \\ \{X_1, \dots, X_1, \dots, X_{2n}\} &= {}^*X && \text{zbiór wszystkich możliwych wektorów wejściowych. } \bar{X}_1 \text{ oznacza } \tilde{X}, \text{ w którym poszczególne } X'_i \in \{X_1, \bar{X}_1\} \text{ zgodnie z binarnym odpowiednikiem 1. } (X_1 - \text{najmniej znaczący bit, np. } n = 6, \bar{X}_5 = X_1 \bar{X}_2 X_3 \bar{X}_4 \bar{X}_5 \bar{X}_6). \\ \{\bar{Y}_1, \dots, \bar{Y}_1, \dots, \bar{Y}_{2m}\} &= {}^*\bar{Y} && \text{zbiór wszystkich możliwych wektorów wyjściowych. } \bar{Y}_1 \text{ oznacza } \tilde{Y}, \text{ w którym poszczególne } Y'_i \in \{Y_1, \bar{Y}_1\} \text{ (} Y'_i - \text{najmniej znaczący bit, np. } m = 6, \bar{Y}_9 = Y_1 \bar{Y}_2 \bar{Y}_3 Y_4 \bar{Y}_5 \bar{Y}_6). \end{aligned}$$

Zdefiniujmy następujące operacje:

Definicja 1

Operacja osiągalności: $Q_1 \xrightarrow{b}$ oznacza, że stan (sytuacja) Q_1 jest uzyskiwany pod warunkiem zapisanym zdaniem b . (\nrightarrow oznacza zaprzeczenie operacji osiągalności).

Definicja 2

Operacja zajęcia zdarzenia warunkowego: $\bar{X}_r \odot {}^*P$ oznacza, że w sytuacji, gdy system był w jednym ze stanów ze zbioru stanów *P , wektor wejściowy przyjął wartość \bar{X}_r .

Zdefiniowanie modułu D-UML_c

W poniższej metodzie wykorzystano zaproponowany przez autora moduł D-UML_c ("Demultiplekserowy" Uniwersalny Moduł Logiczny - Synchroniczny (clocked)). Jego techniczną realizację oraz proponowany symbol przedstawiono na rys. 3.

Korzystając z wprowadzonych oznaczeń i definicji można w następujący sposób określić funkcje realizowane przez poszczególne wyjścia modułu:

$$W \leftarrow b_1 \vee \dots \vee b_t \quad (4)$$

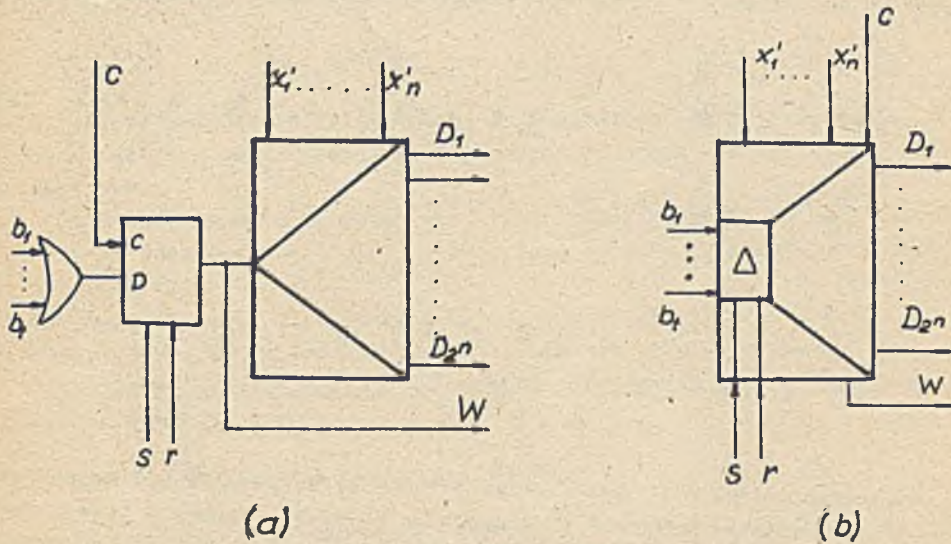
lub

$$W \leftarrow \bigvee_{i=1}^t b_i \tag{4.1}$$

oraz:

$$D_i = \bar{x}_i \odot W; \quad i = 1, \dots, N; \quad N = 2^n \tag{5}$$

Nazwijmy $x_1 \div x_n$ wejściami adresowymi, a $b_1 \div b_t$ wejściami strukturalnymi. Wejścia s, r są asynchronicznymi wejściami ustawiającymi aktualny stan wyjścia W .



Rys. 3. Schemat ideowy (a) i symbol (b) modułu D-UML_c-n

Poprzez rzęd modułu rozumiemy ilość wejść adresowych, poprzez stopień modułu - ilość wejść strukturalnych. Oznaczenie D-UML_c-(n,t) określa moduł rzędu n i stopnia t.

Wykorzystując produkowane seryjnie elementy MSI i SSI można zastąpić moduł D-UML_c szeregowym połączeniem bramki OR, przerzutnika synchronicznego, np. typu D i demultiplexera.

Przy realizacji modułu D-UML_c(2,4) jako elementu scalonego MSI wymagana byłaby obudowa o 16 wyprowadzeniach.

Podstawy teoretyczne metody

Dla dowolnego synchronicznego sekwencyjnego automatu Moore'a o k stanach, n wejściach i m wyjściach można napisać następujący układ zależności osiągalności stanów ze stanów poprzednich [20]:

$$Q_1 = \bar{x}_1 \odot *Q_1^1 \vee \bar{x}_2 \odot *Q_1^2 \vee \dots \vee \bar{x}_N \odot *Q_1^N \tag{3}$$

gdzie:

$$i = 1, \dots, k; \quad N = 2^n,$$

$*Q_1^r$ - zbiór zdefiniowany następująco:

$$*Q_1^r = \left\{ Q_1 \in \{Q_1, Q_2, \dots, Q_k\} : \delta(Q_1, X_r) = Q_1 \right\}$$

a δ - funkcja przejścia automatu.

Wychodząc z zespołu zależności osiągalności stanów (3) można strukturę automatu przedstawić w formie następującego zespołu zależności:

$$\left\{ \begin{array}{l} \bar{X}_1 \odot Q_1 \rightarrow Q_1^1 \\ \bar{X}_2 \odot Q_1 \rightarrow Q_2^1 \\ \vdots \\ \bar{X}_N \odot Q_1 \rightarrow Q_N^1 \end{array} \right. \quad \text{gdzie } Q_j^1 \in \{Q_r, \{\emptyset\}\}; \quad Q_r \in *Q; \quad Q_1 \in *Q \quad (6)$$

$$N = 2^n, \quad l = 1, \dots, k$$

lub krócej:

$$\bar{X}_j \odot Q_1 \rightarrow Q_j^1 \quad (6.1)$$

$$j = 1, \dots, N; \quad l = 1, \dots, k$$

$$N = 2^n$$

Po zastąpieniu znaku " \rightarrow " przez "=" zespół zależności (6) ma charakter równań demultipleksera o \bar{X}_j podanym na wejścia adresowe i niewykorzystywaniu aktualnie niektórych z wyjść ($Q_j^1 = \{\emptyset\}$). Ponadto po przyporządkowaniu $Q_1 \hat{=} W_1$ i $Q_j^1 \hat{=} D_j$ opisuje on funkcje wyjść D_j modułu D-UML_C.

Dla określenia czy Q_j^1 jest równe $\{\emptyset\}$, czy któremuś z elementów zbioru $*Q$, należy stwierdzić, czy w (3) $\bigvee_{i \in \{1, \dots, k\}} (Q_1 \in *Q_i^1)$. Jeśli tak, to $Q_j^1 = Q_i$. Jeśli nie, to $Q_j^1 = \{\emptyset\}$.

O ile zależności (3) odzwierciedlają strukturę grafu automatu w sensie pokazania, z jakich stanów poprzednich i pod wpływem jakich wektorów wejściowych osiągany jest dany stan Q_1 , tak zależności (6) odzwierciedlają strukturę grafu automatu przez pokazanie, jakie stany następne i pod wpływem jakich wektorów wejściowych osiągnane są z danego stanu Q_1 .

Oczywiście zbiór $\left\{ \bigcup_j Q_j^1 \right\}$ zawiera k niepustych elementów oraz $N-k$ elementów \emptyset .

Pozostawiając w (6) tylko te zależności, w których $Q_j^1 \neq \{\emptyset\}$, otrzymujemy zespół zależności (7):

$$\begin{aligned}
 \bar{x}_{j_1} \odot Q_1 &\rightarrow Q_{i_{1,1}} & l = 1, 2, \dots, k \\
 \bar{x}_{j_2} \odot Q_1 &\rightarrow Q_{i_{2,1}} & j_p \in *J, *J \subset *N, *N = \{1, \dots, N\} \\
 \dots &\dots & \\
 \bar{x}_{j_u} \odot Q_1 &\rightarrow Q_{i_{u,1}} & i_{p,1} \in *I, *I \subset *K, *K = \{1, \dots, k\}
 \end{aligned} \quad (7)$$

lub krócej:

$$\begin{aligned}
 \bar{x}_{j_p} \odot Q_1 &\rightarrow Q_{i_{p,1}}; & l = 1, 2, \dots, k & \quad (7.1) \\
 j_p &= j_1, \dots, j_u \\
 i_{p,1} &= i_{1,1}, \dots, i_{u,1}
 \end{aligned}$$

Zespół zależności (7) oznacza, że z danego stanu Q_1 pod wpływem u różnych wektorów wejściowych osiągnąć można u innych stanów. Maksymalna wartość $u=k$.

Logiczne dodanie stronami wszystkich zależności z (7), mających po prawych stronach to samo $Q_{i_{p,1}} = Q_1$, daje oczywiście w wyniku zespół zależności (3).

Rozpatrując (7.1) w kontekście zależności (4) i (5), opisujących działanie modułu $D-UML_c$, dokonajmy następujących przyporządkowań, zakładając przypisanie każdemu stanowi Q_1 automatu jednego modułu oznaczonego przez i (kodowanie "1 z k"):

$$Q_1 \hat{=} W_1 \text{ (przypisanie stanu } Q_1 \text{ modułowi } 1),$$

$$Q_{i_{p,1}} \hat{=} W_{i_{p,1}} \text{ (przypisanie stanu } Q_{i_{p,1}} \text{ modułowi } i_{p,1}).$$

Dla modułu 1:

$$\bar{x}_{j_p} \odot Q_1 = D_{j_p}$$

Ponieważ dla modułu i :

$$W_{i_{p,1}} \leftarrow \bigvee_{v=1}^t b_v$$

tak więc dla zrealizowania dowolnego z równań osiągalności stanu następnego $Q_{i_{p,1}}$ ze stanu aktualnego Q_1 pod wpływem wektora \bar{x}_{j_p} (7.1) należy zapewnić takie połączenie pomiędzy modułami 1 i $i_{p,1}$, aby:

$$\bar{x}_{j_p} \odot Q_1 = D_{j_p} = b_i; \quad i \in \{1, \dots, t\} \quad (8)$$

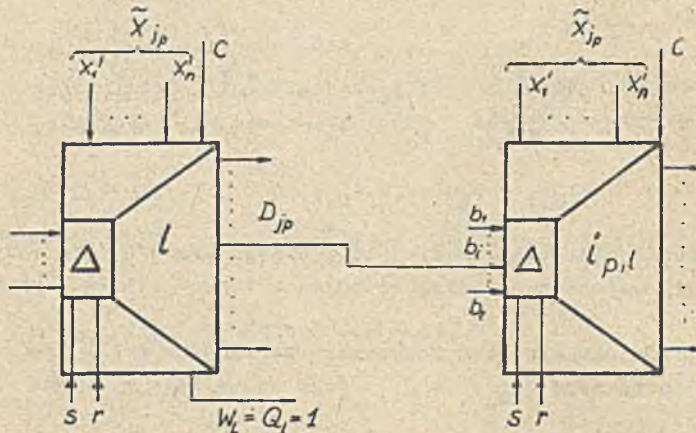
gdzie:

D_{j_p} - wyjście j_p modułu 1;

t - stopień modułu $i_{p,1}$;

b_i - wejście strukturalne i modułu $i_{p,1}$.

Realizację jednego z równań z (7.1) przedstawia rys. 4.



Rys. 4. Połączenie modułów 1 oraz $i_{p,1}$ odpowiadające osiągnięciu stanu $Q_{i_{p,1}}$ ze stanu Q_1 pod wpływem wektora \tilde{x}_{j_p}

Tak więc każdy sekwencyjny automat synchroniczny zrealizować można w oparciu o zespół zależności (6), realizując je na bazie modułów D-UML_c w myśl (8).

Ponieważ (6) jest matematyczną formą opisu grafu układu, stąd łączenie modułów D-UML_c zgodnie z (6), (8) i rys. 4 odbywać się może bezpośrednio z grafu układu bez jakichkolwiek etapów pośrednich. Ilość użytych modułów równa jest ilości stanów wewnętrznych układu (kodowanie 1 z k), a więc ilości węzłów w grafie. Ilość wszystkich połączeń pomiędzy wyjściami D_i modułów a wejściami strukturalnymi równa jest ilości gałęzi grafu (z wyjątkiem sytuacji, gdy jakiś stan(y) osiągnąć jest ze wszystkich stanów pod wpływem tego samego wektora).

Każdemu węzłowi grafu przypisany jest jeden moduł D-UML_c, każdej gałęzi grafu - połączenie typu jak na rys. 4. Jeśli gałąź łącząca stan Q_i z Q_j opisana jest wektorem \tilde{x}_1 , to połączenie pomiędzy modułami 1 oraz j zaczyna się na wyjściu D_1 modułu 1, a kończy na dowolnym z wejść b modułu j. Kolejne wejścia strukturalne modułu Q_j połączone są analogicznie ze wszystkimi modułami reprezentującymi stany, z których stan reprezentowany jest przez moduł 1 osiągnąć w myśl grafu układu.

Widać więc, że struktura rozwiązania jest skopiowaniem struktury grafu wyjściowego.

D e f i n i c j a 3

Automatem sekwencyjnym stopnia p nazywać będziemy automat, w którym każdy ze stanów osiągnany jest z nie więcej niż p stanów poprzednich. Stopień automatu jest miarą "odejścia" struktury automatu od struktury liniowej.

Posługując się definicją 3 można określić stopień modułów $D-UML_c(n,t)$ konieczny do realizacji danego automatu. Do syntezy automatów stopnia p potrzebne są moduły stopnia nie wyższego niż p ($D-UML_c(n,p)$).

Funkcje wyjść automatu zrealizować należy poprzez wygenerowanie zbioru \mathcal{X}_1 o elementach $\mathcal{X}_1 = \{Q_p \in \{Q_1, \dots, Q_k\} : \lambda_1(Q_p) = 1\}$; λ_1 - funkcja wyjść związana z Y_1 . Dla uzyskania kolejnych Q_1 postąpić można w zależności od struktury realizowanego automatu, jego przeznaczenia i technicznych możliwości realizacji (stopień automatu, stopień modułów, dysponowanie bramkami OR, możliwość realizacji tzw. iloczynu montażowego) wg jednego z poniższych sposobów:

1. Wyjścia W modułów reprezentujących $Q_1 \in \mathcal{X}_1$ wprowadzić na wejścia strukturalne modułu realizującego dany zbiór \mathcal{X}_1 (dane wyjście Y_1), tzw. modułu wyjściowego 1. Wejścia adresowe i wyjścia typu D tego modułu pozostają niewykorzystane. Jego wyjście W reprezentuje $Q_1(Y_1)$. Prawidłowa wartość sygnału wyjściowego pojawia się jednak z opóźnieniem 1 taktu.
2. O ile pozwala na to stopień modułów i stopień automatu, zrealizować zbiór \mathcal{X}_1 wg tych samych zasad jak dowolne z Q_1 , przy czym zbiór stanów poprzednich zbioru \mathcal{X}_1 będzie sumą zbiorów stanów poprzednich wszystkich $Q_1 \in \mathcal{X}_1$. Jest to możliwe tylko wtedy, gdy liczebność tak utworzonego zbioru stanów poprzednich zbioru Q_1 nie jest większa od stopnia modułu.

A l g o r y t m

Dla syntezy dowolnego opisanego grafem, synchronicznego, sekwencyjnego automatu Moore'a rzędu t i o n wejściach w oparciu o moduły $D-UML_c(n,t)$ należy korzystając bezpośrednio z grafu, zrealizować następujące kroki:

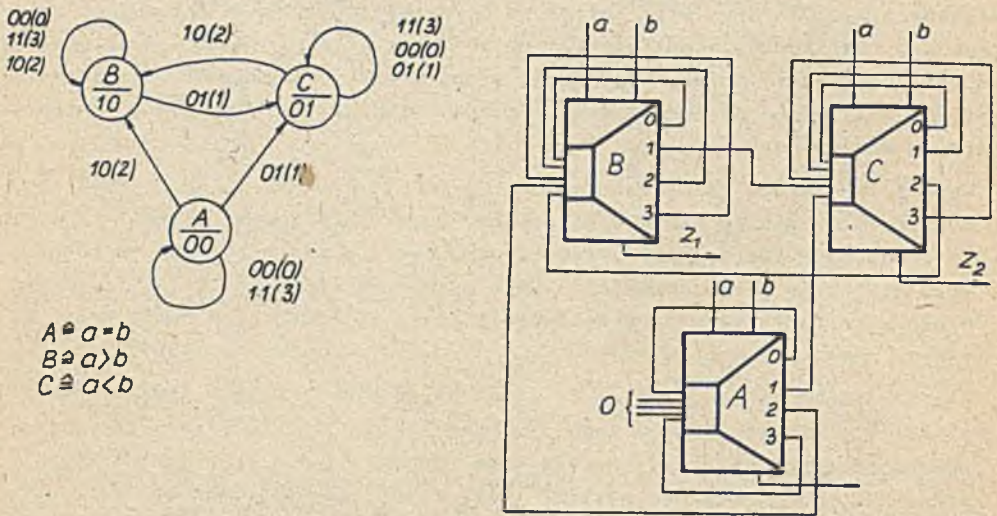
1. Każdemu stanowi układu (węzłowi grafu) przypisać jeden moduł $D-UML_c$.
2. Zrealizować sieć połączeń pomiędzy modułami odzwierciedlając strukturę gałęzi grafu wg następujących zasad:
 - gałąź grafu pomiędzy stanem Q_i i Q_j odpowiada połączeniu pomiędzy modułami i oraz j ,
 - połączenie zaczyna się na wyjściu D_r modułu i o ile odpowiadająca gałąź grafu opisana jest wektorem \vec{x}_r ,
 - połączenie kończy się na dowolnym z wejść b_1 modułu j .
3. Na wejścia adresowe modułów wprowadzić zmienne wejściowe układu.
4. Zrealizować wyjścia jednym ze sposobów podanych powyżej.

Przykład

Zrealizować w oparciu o zaprezentowaną metodę zadany grafem szeregowy komparator o wejściach a, b i wyjściach Z_1, Z_2 .

Całość procesu projektowego sprowadza się do odwzorowania grafu w układ połączeń pomiędzy trzema modułami reprezentującymi trzy stany układu. Zbiory wyjściowe: $\check{Q}_1 = B$ (odpowiada wyjściu Z_1), $\check{Q}_2 = C$ (odpowiada wyjściu Z_2).

Graf i schemat połączeń komparatora przedstawiono na rys. 5 (pominięto wejścia c, s, r).



Rys. 5. Realizacja przykładu

5. WNIOSKI

Zaproponowany w pkt. 4 algorytm należy do grupy najprostszych z istniejących algorytmów syntezy sekwencyjnych synchronicznych automatów zadanych grafem, dających strukturę połączeń pomiędzy elementami układu w sposób natychmiastowy z bezpośredniej analizy grafu. Do tej samej grupy należy znany algorytm Le Van-Van Houtte'a [21], bazujący na modułach T-UML_c.

O ile jednak algorytm oparty na zaproponowanych modułach D-UML_c generuje struktury jednorodne, to algorytm z [21] daje rozwiązania pseudojednorodne tak ze względu na konieczność użycia bramek OR do sumowania stanów na wejściach programujących modułów T-UML_c jak też do realizacji funkcji wyjść.

Propozycja modułu T-UML_c z wielowejściowymi brankami OR na wszystkich wejściach programujących, co spowodziłoby realizację wg [21] do struktury

ry jednorodnej względem wygenerowania wszystkich stanów układu (pozostałaby niejednorodność wynikająca z realizacji funkcji wyjść), jest technologicznie nie do zrealizowania ze względu na ograniczoną liczbę wyprowadzeń zewnętrznych elementu scalonego. W praktyce algorytm z [21] oparty o taki moduł dawałby też nieekonomiczne rozwiązania ze względu na bardzo dużą ilość niewykorzystanych wejść modułów.

Realizacja algorytmu z modułem D-UML_c i algorytmu z [21] w oparciu o zastępcze, kaskadowe połączenia bramek, przerzutników i multiplekserów przemawia również za zaprezentowanym algorytmem ze względu na znacznie większą regularność połączeń struktur bazujących na koncepcji modułów D-UML_c.

O ile bowiem przy realizacji z [21] bramki OR pojawiają się na różnych wejściach programujących różnych modułów T-UML_c w zależności od struktury grafu wyjściowego, tak przy realizacji wg przedstawionego algorytmu z jednym modułem związana jest tylko jedna bramka OR i znajduje się ona w stałym dla każdego modułu miejscu, tworząc wejścia strukturalne.

Stopień modułu D-UML_c nie jest praktycznym ograniczeniem opisanej metody, gdyż w większości rzeczywistych przypadków stopień automatu jest liczbą niewielką (3,4,5).

LITERATURA

- [1] Forslund D., Waxman R.: The Universal Logic Block (ULB) and its Application to Logic Design. Seventh Annual Symp. Switching and Automata Theory, Conf. Rec. 1966.
- [2] Yau S.S., Tang C.K.: Universal Logic Modules and Their Applications IEEE Trans. Computers Vol. C-19, February 1970.
- [3] Arnold T.F., Tan C.J., Newborn M.N.: Iteratively Realized Sequential Circuits IEEE Trans. Computers, vol. C-19, January 1970.
- [4] Patt Y.N.: Optimal and Near Optimal Universal Logic Modules with Interconnected External Terminals IEEE Trans. Computers, October 1973.
- [5] Butler I.T., Breeding K.I.: Some Characteristics of Universal Cell Nets IEEE Trans. Computers, October 1973.
- [6] Rene D.: Modular Design of Asynchronous Circuits Defined by Graphs. IEEE Trans. Computers Vol. C-26, August 1977.
- [7] Newborn M.M.: Synthesis Technique for Binary Input - Binary Output Synchronous Sequential Moore Machines. IEEE Trans. Computers, Vol. C-17, July 1968.
- [8] Voith R.P.: Minimum Universal Logic Module Sequential Circuits with Decoders. IEEE Trans. on Computers VC-26, October 1977.
- [9] Friedman A.D.: Feedback in Synchronous Sequential Switching Circuits. IEEE Trans. Computers, No 3, 1966.
- [10] Unger S.: Tree Realization of Herative Circuits. IEEE Trans. on Comp. April 1977.
- [11] Ullman J.D., Weiner P.: Universal Two State Machines, Charakterization Theorems and Decomposition Schemes Proc. 9th Ann. Symp. Switching and Automata Theory, Oct. 1968.

- [12] Weiner P., Hopcroft J.E.: Modular Decomposition of Synchronous sequential Machines in Conf. Rec. IEEE 8th ASSAT, Oct. 1967.
- [13] Williams G.H.: Uniform Decomposition of Incompletely Specified Sequential Machine. IEEE Trans. on Comp. August 1975.
- [14] Keefe K.H.O.: Modularity in Design: "Shift Registers and Counters Used as System Building Blocks. IEEE Trans. on Comp. Febr. 1973.
- [15] Covarroc J.C., Blanchard M., Gillon J.: An Approach to the Modular Design of Industrial Switching Systems IFAC, Discrete Systems, Riga, 1974.
- [16] Nordman B.J., Cormick B.H.: Mc Modular Asynchronous Control Design. IEEE Trans. on Comput. March 1977.
- [17] Almaini A.E.A.: Sequential Machine Implementations Using Universal Logic Modules. IEEE Trans. on Comp. Oct. 1978.
- [18] Greer C.R., Thompson R.A.: Combinational Logic Design with Decoders. IEEE Trans. on Comp. Sept. 1978.
- [19] Yau S.S., Tang C.K.: Transformation of an Arbitrary Switching Function a Totally Symetric Function. IEEE Trans. on Comp. Dec. 1971.
- [20] Mołski M.: Synteza synchronicznych struktur logicznych zbudowanych z modułów typu M-D. Archiwum Aut. i Telem. T. XXII. Z. 1-2.
- [21] Le Van T., Van Houtte H.: Delayed Universal Logic Modules and Sequential Machine Synthesis. IEEE Trans. on Comp., August 1975.

Р е з ю м е

В работе описаны универсальные модули для синтеза комбинационных и последовательных схем. Представлены некоторые алгоритмы синтеза последовательных синхронных автоматов из модулей типа Т-УМЛ_ц (Т-УМЛ_с).

Показан новый алгоритм синтеза последовательных синхронных автоматов из модулей нового типа Д-УМЛ_ц (Д-УМЛ_с).

S u m m a r y

Several main concepts of Universal Logic Circuits for the combinational and sequential machines synthesis procedures are described. In the paper some basic and well known algorithms for the synchronous, sequential machine synthesis with T-UML_c modules in the uniform and quasi-uniform way are presented.

The new synthesis algorithm for the uniform modular structures of the synchronous, sequential Moore machines with the proposed D-UML_c modules is given.