

Walery SOŁOWJEW, Marek GRUSZEWSKI

Politechnika Białostocka

SYNTEZA UKŁADÓW KOMBINACYJNYCH I SEKWENCYJNYCH NA PLD

Streszczenie. W pracy analizowane są własności struktur wewnętrznych programowalnych układów logicznych - PLD (*Programmable Logic Devices*) i sposoby ich efektywnego wykorzystania do syntezy układów cyfrowych.

W przypadku układów kombinacyjnych rozpatrywane są algorytmy syntezy złożonych układów, dla których brakuje odpowiedniej liczby wejść i termów związanych z jednym wyjściem układu PLD.

W przypadku układów sekwencyjnych główny problem polega na względnie niewielkiej liczbie termów związanych z jednym wyjściem. W pracy proponowane są metody usunięcia tej wady przez specjalne kodowanie wewnętrznych stanów automatu i zwiększenie liczby pozycji kodu.

SYNTHESIS COMBINATIONAL AND SEQUENTIAL CIRCUITS ON PROGRAMMABLE LOGIC DEVICES

Summary. Features of structural organization of Programmable Logic Devices (PLDs) and ways of their effective application at synthesis of the digital circuits are analyzed. The approaches to synthesis of the complex combinational circuits on PLD in case of restrictions on number of inputs, outputs and product terms are considered. The approach to elimination of the given defect at the expense of special coding of automata states and increase of number of bits in automata states code is offered.

1. Wstęp

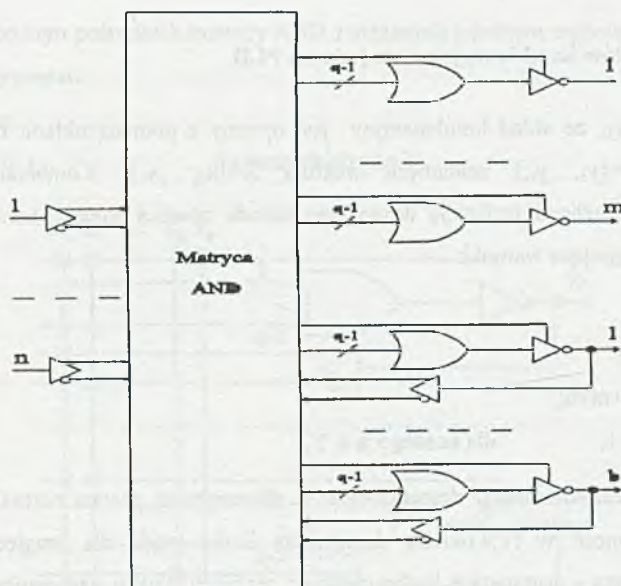
W ostatnich latach układy PLD (*Programmable Logic Devices*) stały się wiodącymi elementami wśród składników używanych do budowy cyfrowych urządzeń automatyki i techniki obliczeniowej [1]. Nowoczesne układy PLD charakteryzują się niską ceną, dużą szybkością pracy, dużymi możliwościami funkcjonalnymi (jeden układ programowalny może zastąpić nawet kilkadziesiąt tradycyjnych układów serii 74XXX), możliwością wielokrotnego przeprogramowywania, niską zużywaną mocą itd. Przy wykorzystaniu nowoczesnych środków automatyzowanego projektowania opracowanie urządzeń na bazie układów PLD sprowadza się do zapisu logicznych równań wyjściowych w sposób, który umożliwi ich realizację na PLD. Czas opracowania dość złożonych układów może być bardzo długi.

Niestety, znane metody syntezy [2] słabo uwzględniają specyficzne cechy nowej bazy elementowej, a metody realizowane w popularnych pakietach automatyzowanego projektowania (PALASM, CUPL, ABEL i innych) praktycznie przekompilowują wyjściowy opis projektu na format dopasowany do PLD i nie przeprowadzają optymalizacji wyjściowego opisu układu. Dlatego też jakość projektu zrealizowanego na bazie PLD zależy w dużym stopniu od umiejętności projektanta, jego doświadczenia i intuicji, a także umiejętności wykorzystania specyficznych cech nowoczesnych elementów. W pracy tej zostało omówione oryginalne podejście do projektowania układów kombinacyjnych i sekwencyjnych z wykorzystaniem PLD.

2. Możliwości PLD przy syntezie układów kombinacyjnych

Wszystkie PLD można podzielić na dwie klasy: standardowe PAL (Programmable Array Logic) i uniwersalne PLD [3]. Ogólna struktura kombinacyjnego układu PAL pokazana jest na rys.1. Posiada on n dwufazowych wejść, programowalną matrycę AND, m jednokierunkowych i b dwukierunkowych wyjść. Matryca AND pozwala tworzyć elementarne iloczyny zmiennych wprowadzanych na wejścia. Wyjścia matrycy AND zwane są szynami pośrednimi, termami (product terms) lub termami PAL. Wszystkie termy PAL są poroździelane na sekcje po q termów w każdej sekcji. Jeden term sekcji jest wykorzystywany do sterowania inwerterem wyjściowym, a $q-1$ są wejściami bramki OR. Wyjścia dwukierunkowe mają dodatkowo pętlę sprzężenia zwrotnego łączącą je z wejściami matrycy AND. Jeśli na szynie sterującej inwertera wyjściowego ustali się logiczna jedynka, to wyjście przechodzi w stan wysokiej impedancji. Przy czym „czyste” wyjścia są po prostu odłączane od szyny, a wyjścia dwukierunkowe mogą być wykorzystywane jako wejścia. Element PAL, przedstawiony na rys.1, będziemy oznaczać jako $PAL(n,m,b,q)$, np. $PAL16L8$ zostanie przedstawiony jako $PAL(10,2,6,8)$.

Uniwersalne PLD (PAL o architekturze V) charakteryzują się elastycznością architektury, np. $PAL16V8$ może emulować 21 standardowych układów PAL [3]. Makrokomórki wyjściowe uniwersalnych PLD dają możliwość programowania ich jako wyjścia rejestrowe lub kombinacyjne. Mogą także pracować jako wejścia. Rozwinięte architektury uniwersalnych PLD ($PALCE29M16$, $PALCE610$) pozwalają programować każde ze swoich wyprowadzeń jako wejście lub wyjście (rejestrowe lub kombinacyjne), dopuszcza się także buforowanie sygnałów wyjściowych oraz wybór rodzaju przerzutnika wyjściowego.



Rys. 1. Ogólna struktura kombinacyjnego układu PAL
 Fig. 1. General structure of combinational PAL

We wszystkich układach PAL sygnały wejściowe i wyjściowe, biegnące wzdłuż pętli sprzężenia zwrotnego, są przedstawione kodem dwufazowym, dlatego nie ma konieczności specjalnego ich negowania; cena uzyskania sygnału x i \bar{x} jest jednakowa. Na termach PAL można realizować dowolną koniunkcję zmiennych wejściowych i sygnałów pętli sprzężenia zwrotnego. Maksymalna liczba zmiennych koniunkcji jest ograniczona liczbą wejść matrycy AND i wynosi $n+b$, np. PAL16L8 pozwala realizować koniunkcję do 16 zmiennych. Ponieważ z każdym wyjściem jest związana jego grupa termów, to do optymalizacji równań logicznych można stosować rozdzieloną minimalizację funkcji boolowskich, a nie minimalizację całego układu funkcji boolowskich. Ułatwia to zadanie minimalizacji i prowadzi do uzyskania lepszych rezultatów. Oprócz tego, wystarczy znaleźć najkrótszą alternatywną postać normalną, zawierającą minimalną liczbę różnych elementarnych koniunkcji.

Wiele układów PAL, a wśród nich wszystkie uniwersalne, umożliwiają programowe ustawienie polarności każdego wyjścia z osobna. Dlatego można realizować funkcję y lub też jej wartość zanegowaną \bar{y} , w zależności od złożoności równania logicznego, a zatem przekształcać sygnał wyjściowy do potrzebnego poziomu logicznego.

Z uwagi na ograniczenia PLD zaleca się wybrać względnie małą liczbę q (od 8 do 16) szyn pośrednich, związanych z jednym wyjściem.

3. Synteza układów kombinacyjnych na jednym PLD

Przyjmijmy, że układ kombinacyjny jest opisany z pomocą układu równań funkcji boolowskich $Y = \{y_1, \dots, y_N\}$ zmiennych wektora $X = \{x_1, \dots, x_L\}$. Kombinacyjny element PAL(n,m,b,q) umożliwia realizację dowolnego układu równań funkcji boolowskich, jeśli spełnione są następujące warunki:

$$\begin{aligned} L &\leq n+b; \\ N &\leq m+b; \\ L+N &\leq n+m+b; \\ Q(y_i) &\leq q-1, \quad \text{dla każdego } y_i \in Y, \end{aligned} \tag{1}$$

gdzie $Q(y_i)$ - liczba koniunkcji elementarnych w alternatywnej postaci normalnej funkcji y_i . Pierwsza nierówność w (1) określa dostateczną liczbę wejść dla przyjęcia zmiennych wejściowych, druga - dostateczną liczbę wyjść do realizacji funkcji wyjściowych, a trzecia - dostateczną liczbę wszystkich wyprowadzeń elementu PAL.

Jeśli dla którejkolwiek funkcji y_i , $y_i \in Y$, nie jest spełniony warunek na liczbę szyn pośrednich (czwarta nierówność w układzie (1)), to dla jej realizacji można wykorzystać terminy związane z niewykorzystywanymi wyjściami dwukierunkowymi. W rezultacie funkcja y_i będzie realizowana za pomocą dwupoziomowego układu kaskadowego, jak to pokazano na rys. 2, a jej logiczne równanie będzie opisane z pomocą pośrednich zmiennych z_1, \dots, z_g : $y = y(z_1, \dots, z_g)$. W ten sposób liczba $Q(y_i)$, przy której możliwa będzie realizacja wybranych funkcji y_i na PAL(n,m,b,q), jest opisana następującymi warunkami:

$$\begin{aligned} Q(y_i) &\leq (q-1)b + ((q-1)-b) && \text{przy } b < q-1; \\ Q(y_i) &\leq (q-1)(q-1) && \text{przy } b \geq q-1. \end{aligned} \tag{2}$$

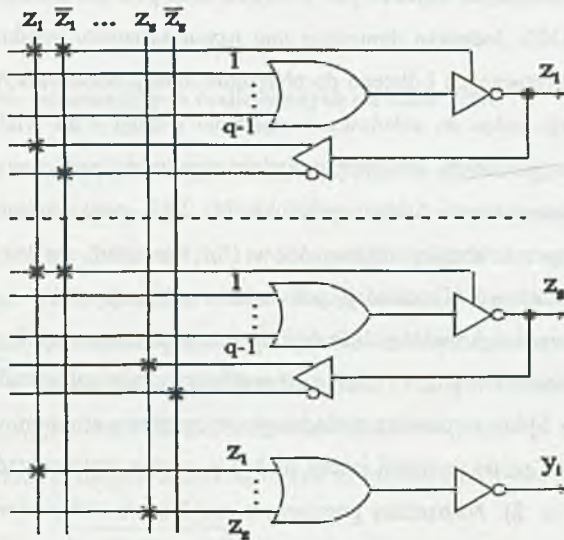
Pierwsza nierówność w (2) opisuje przypadek, gdy liczba dwukierunkowych wyjść PAL jest mniejsza od $q-1$. W tym wypadku na łączącej bramce OR można korzystać dodatkowo z $(q-1)-b$ termów matrycy AND. Druga nierówność w (2) uwzględnia przypadek gdy b jest większe od $q-1$ i na ostatniej bramce OR można korzystać z co najwyżej $q-1$ pośrednich funkcji z_1, \dots, z_g .

Przykładowo, dla elementu PAL16L8 zgodnie z pierwszą nierównością (2) $Q(y_i) \leq 43$. Uniwersalne PLD można wykorzystywać jako PAL(n,0,b,q), gdzie n i b - łączna liczba wejść i wyjść PLD. W takim przypadku należy uwzględnić warunki (1) i (2). Rozwinięte architektury PLD charakteryzują się dwoma parametrami: całkowitą liczbą wyprowadzeń z oraz liczbą q

określającą liczbę szyn pośrednich matrycy AND związanych z jednym wyjściem. Warunek (1) przyjmuje wtedy postać:

$$L+N \leq z;$$

$$Q(y_i) \leq q-1, \quad \text{dla wszystkich } y_i \in Y.$$



Rys. 2. Wykorzystanie dwukierunkowych wyjść układu PAL w przypadku braku termów
Fig. 2. Using of bidirectional pins of PAL in case defect of product terms

4. Synteza złożonych układów równań funkcji boolowskich

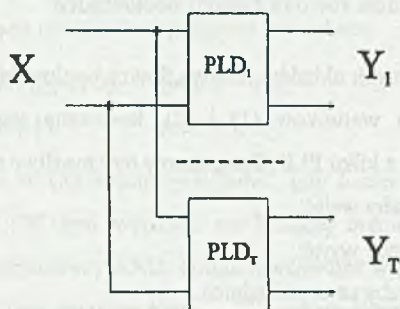
Przy realizacji złożonych układów równań funkcji boolowskich, dla których nie daje się dobrać PLD spełniających warunków (1) i (2), konieczna staje się realizacja układów kombinacyjnych złożonych z kilku PLD. Rozpatrzmy trzy możliwe sytuacje:

- 1) nie ma wystarczającej liczby wejść;
- 2) nie ma wystarczającej liczby wyjść;
- 3) nie ma wystarczającej liczby szyn pośrednich.

W przypadku deficytu wejść metody syntezy są w pełni zgodne z metodami przyjmowanymi wcześniej przy syntezie układów równań funkcji boolowskich na Programmable Logic Arrays (PLAs). Ponieważ metoda ta jest dobrze znana [4, 5], nie będziemy jej tutaj omawiać.

W przypadku braku wyjść przy syntezie układów na rejestrowych elementach PAL lub uniwersalnych PLD można wykorzystać następujące oryginalne rozwiązanie. Część zmiennych wejściowych zapisujemy w wewnętrzne rejestry PLD, a pozostałą część podajemy na wejścia. Dzięki istnieniu pętli sprzężenia zwrotnego w określeniu wartości funkcji boolowskiej biorą udział obie części zmiennych wejściowych. Podejście takie pozwala maksymalnie wykorzystać wejścia matrycy AND. Jednakże dopuszcza ono użycie schematu kombinacyjnego tylko w urządzeniach synchronicznych i dlatego do obliczenia funkcji boolowskich potrzebne są dwa takty synchronizacji: jeden do załadowania rejestrów i drugi - do właściwego wyliczenia wartości funkcji wyjściowych. Podobne podejście stosuje się przy syntezie sumatorów na PAL16X4. Niewystarczalność liczby wyjść układów PAL przy realizacji systemu funkcji boolowskich (naruszenie drugiej nierówności w (1)) rozwiązuje się bardzo prosto: łącząc równolegle kilka układów PAL posiadających wspólne wejścia (rys. 3).

Jeżeli przy realizacji niektórych funkcji naruszone zostaną ograniczenia na liczbę szyn pośrednich (nierówność czwarta w (1)), można spróbować zminimalizować te funkcje jedną ze znanych metod [4, 5] lub za pomocą posiadanego wyposażenia programowego. Jeżeli istnieją wolne wyjścia PAL, to do realizacji jakiejś funkcji y_i można wykorzystać terminy związane z tymi wyjściami (rys. 2). Najbardziej popularnym przykładem związanym z deficytem szyn pośrednich PLD jest realizacja zamiast funkcji y_i , $y_i \in Y$, jej odwrotności \bar{y}_i (negacja następuje w buforze wyjściowym). Można również minimalizować jednocześnie obydwie funkcje y_i i \bar{y}_i , a co za tym idzie, dobierać najbardziej stosowny rezultat.



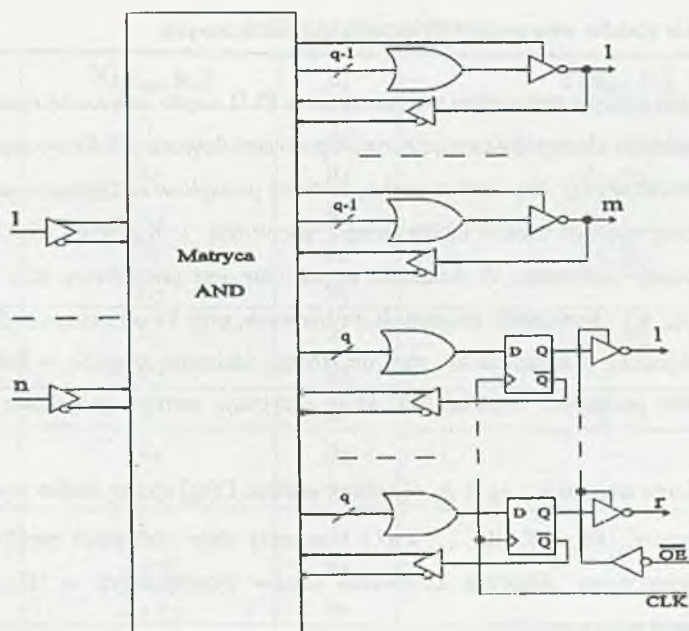
Rys. 3. Równoległe połączenie PLD w przypadku braku wyjść
Fig. 3. Parallel connection PLD in case defect of outputs

Jeżeli za pomocą żadnej z przedstawionych metod nie udało się spełnić ograniczeń na liczbę termów, to można skorzystać ze sposobu [6]. Jego istota polega na stworzeniu schematu kaskadowego na jednym układzie PAL. W związku z tym na niewykorzystywanych wyjściach

PAL realizowane są funkcje faktoryzujące, których wartości podawane są poprzez pętle sprzężenia zwrotnego na wejście matrycy AND. Powoduje to powstanie następujących możliwości: realizacja zanegowanych wartości funkcji faktoryzujących (negacja ich następuje w matrycy AND), wykorzystanie w roli funkcji faktoryzujących już zrealizowanych funkcji lub też ich inwersji.

5. Synteza układów sekwencyjnych realizowanych na bazie PAL

Do syntezy układów sekwencyjnych wykorzystuje się zwykle rejestrowe układy PAL. Uogólniona struktura rejestrowego układu PAL przedstawiona została na rys. 4. Od kombinacyjnych układów PAL odróżnia je obecność r wyjść rejestrowych. Makrokomórki rejestrowych wyjść PAL zawierają dodatkowo przerzutniki typu D. Pętla sprzężenia zwrotnego jest tutaj stworzona poprzez połączenie inwersyjnego wyjścia przerzutnika z wejściem matrycy AND. Wszystkie przerzutniki układu PAL synchronizowane są wspólnym sygnałem CLK, a wyjściowe inwertery sterowane wspólnym sygnałem uaktywnienia wyjść \overline{OE} . Rejestrowy układ PAL (przedstawiony na rys. 4) oznaczają będziemy PAL (n, m, r, q).



Rys. 4. Ogólna struktura rejestrowego układu PAL

Fig. 4. General structure of registered PAL

Niech w charakterze matematycznego modelu układu sekwencyjnego przyjęty będzie automat skończony, który będziemy charakteryzować liczbą L wejściowych zmiennych zbioru $X = \{x_1, \dots, x_L\}$, liczbą N wyjściowych zmiennych zbioru $Y = \{y_1, \dots, y_N\}$, liczbą M stanów wewnętrznych zbioru $A = \{a_1, \dots, a_M\}$ i liczbą R funkcji wzbudzenia komórek pamięci zbioru $H = \{h_1, \dots, h_R\}$. Warunki realizacji automatu skończonego na PAL (n, m, r, q) będą następujące:

$$\begin{aligned} R &\leq r; \\ N + R &\leq m + r; \\ L + N + R &\leq n + m + r; \\ Q(y_i) &\leq q, \quad i = \overline{1, N}; \\ Q(h_j) &\leq q, \quad j = \overline{1, R}. \end{aligned} \quad (4)$$

Jeżeli zbiory Y i H połączymy w jeden zbiór funkcji wyjściowych W , $W = Y \cup H$, to synteza układów sekwencyjnych sprowadza się do rozpatrzonej powyżej syntezy systemu funkcji boolowskich (SFB). Otwartym pozostaje pytanie kodowania wewnętrznych stanów automatu.

7. Kodowanie stanów wewnętrznych automatów skończonych

Podczas syntezy automatów skończonych na PLD często zdarza się naruszenie przez funkcje wzbudzenia elementów pamięciowych ograniczeń dotyczących liczby szyn pośrednich (piąta nierówność w (4)). Aby temu zapobiec, możemy postąpić w następujący sposób.

Załóżmy automat zadany tablicą przejść, jak w tabl. 1. Każdy jej wiersz odpowiada jednemu przejściu automatu. W kolumnie a_m zapisany jest początkowy stan przejścia, w kolumnie $X(a_m, a_s)$ - koniunkcja zmiennych wejściowych, przy której następuje dane przejście (dla wartości jeden), w kolumnie a_s - stan, do którego następuje przejście, w kolumnie $Y(a_m, a_s)$ - podzbiór zmiennych wyjściowych, które przyjmują wartość jedynekową dla danego przejścia.

Każdemu stanowi $a_m, a_m \in A$, określimy wartość $C(a_m)$ równą liczbie wystąpień stanu a_m w kolumnie a_s . Przez $K = \{k_1, \dots, k_{R^2}\}$ oznaczmy zbiór wszystkich możliwych kodów stanów wewnętrznych. Algorytm kodowania stanów wewnętrznych w takim przypadku wygląda w następujący sposób:

1. Określa się liczbę R rodzajów kodu wewnętrznych stanów automatu: $R = \text{int } \log_2 M$.

2. Zeruje się wagi rodzajów kodu $v_i=0$, $i = \overline{1, R}$; określa się zbiór K kodów stanów wewnętrznych; przyjmuje się $A^* := A$.
3. Ze zbioru A^* wybiera się stan a_m z maksymalną wartością $C(a_m)$.
4. Spośród kodów ze zbioru K wyszukuje się kod k_i , dla którego spełnione są warunki: $v_i + C(a_m) \leq q$ dla wszystkich rodzajów i , w których kod k_i przyjmuje wartości jedynek. Jeśli jest wiele takich kodów, to spośród nich wybiera się kod przyjmujący minimalną liczbę jedynek.
5. Jeżeli w punkcie 4 nie udało się znaleźć odpowiednich kodów dla stanu a_m , to przyjmuje się $R := R + 1$ i przechodzi się do punktu 2. W przeciwnym przypadku znaleziony kod k_i wyznacza się stanem a_m , ze zbioru A^* usuwa się stan a_m , a ze zbioru K - kod k_i , dokonuje się korekty wag rodzajów kodów, przyjmuje się:

$$A^* := A^* \setminus \{a_m\};$$

$$K := K \setminus \{k_i\};$$

$$v_i := v_i + C(a_m), \text{ dla wszystkich } i, \text{ w których kod } k_i \text{ przyjmuje wartości jedynek.}$$
 Przejście do punktu 3.
6. Koniec.

Tablica 1

Tablica przejść automatu

a_m	$X(a_m, a_s)$	a_s	$Y(a_m, a_s)$
a_1	1	a_2	-
a_2	$\frac{x_1}{x_1}$	a_6 a_3	y_1, y_2
a_3	$\frac{x_2}{x_2}$	a_4 a_7	y_2, y_3
a_4	$\frac{x_3}{x_3}$	a_5 a_3	y_4, y_5
a_5	1	a_1	y_2, y_3
a_6	$\frac{x_4}{x_4}$	a_7 a_6	-
a_7	$\frac{x_5}{x_5} \frac{x_3}{x_3}$ $\frac{x_5}{x_5} \frac{x_3}{x_3}$	a_8 a_5 a_3	y_4, y_5
a_8	1	a_1	y_4

Jako przykład prześledzimy kodowanie stanów wewnętrznych automatu zadanego tablicą 1, przy $q=6$. W przedstawionym przykładzie udało się zakodować wszystkie stany wewnętrzne automatu bez zwiększania początkowej liczby rodzajów kodu $R = \text{int} \log_2 8 = 3$. Kolejność kroków realizacji algorytmu, wybierane na każdym kroku stany a_m , ich wartości $C(a_m)$, wyznaczony kod K_i i zmiany wartości wag rodzajów kodu v_i , $i = \overline{1, R}$ przedstawione zostały w tablicy 2.

Tablica 2

Kolejność kroków realizacji algorytmu kodowania stanów wewnętrznych

L.p.	a_m	$C(a_m)$	K_i	v_1	v_2	v_3
1	a_3	3	000	0	0	0
2	a_1	2	001	0	0	2
3	a_5	2	010	0	2	2
4	a_6	2	011	2	2	2
5	a_7	2	100	2	4	4
6	a_2	1	101	3	4	5
7	a_4	1	110	4	5	5
8	a_8	1	111	5	6	6

LITERATURA

1. PAL Device Data Book and Design Guide. Advanced Micro Devices, 1995.
2. Pałagin A. W., Barkałow A. A., Jusifow S. I., Starodubow K. E., Szwec A. G.: Realizacja automatów mikroprogramowych na PLD. USiM, nr 8, 1991, s. 18-22.
3. Sołowiew W. W.: Projektowanie układów funkcjonalnych systemów cyfrowych na bazie programowalnych układów logicznych. Bestprint, Mińsk 1996, s. 252.
4. Zakrzewski A. D.: Synteza układów kaskadowych. Nauka, Mińsk 1981, s. 416.
5. Bibiło P. N.: Synteza kombinacyjnych struktur PLA dla VLSI. Nauka i technika, Mińsk 1992, s. 232.
6. Sołowiew W. W.: Wykorzystanie programowalnych macierzy logicznych przy syntezie układów kombinacyjnych. AWT, nr 4/5, 1995, s. 53-56.
7. Gruszkow W. M.: Synteza automatów cyfrowych. Fizmatgiz, Mińsk 1962, s. 476.

Recenzent: Prof. dr hab. inż. Józef Ober

Wpłynęło do Redakcji do 30.06.1996 r.

Abstract

Now Programmable Logic Devices (PLDs) find more and more wide application as new element base for construction of various digital devices of automatics and robots. Modern PLD are characterized in low cost, high speed, significant functional opportunities (one PLD can replace tens of cases of traditional "rigid" logic), low consumed capacity and other. By use of modern the development software on the basis PLD even is enough complex projects can make total some hours. However the traditional methods of synthesis of digital devices on the basis PLD do not take into account features of new element base. In work the original approaches to designing of combinational and sequential circuits on the basis PLD are considered.

At synthesis of the combinational circuits are considered algorithms of synthesis of the complex circuits in case of insufficient number of inputs PLD and terms, connected to the one output PLD. Such opportunities PLD as using of internal circuits of a feedback for transfer of significances of factor-functions on an input PLD, realization of inverse significances of factor-functions with the subsequent their inverting on a matrix AND, use of factors-functions already of realized logic functions, realization of the combinational circuits on register PLD for two steps of synchronization are thus used.

At synthesis of the sequential circuits on PLD a main problem is rather small number of terms, connected to one output. In work ways of elimination of this defect at the expense of special coding of internal condition of the automatic device and increase of number of the categories of a code of condition are offered.