

Dariusz KANIA, Józef KULISZ
POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI

Zastosowanie grafu niezgodności i dopełnień w procesie kodowania automatów asynchronicznych

Dr hab. inż. Dariusz KANIA

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej. Pracę doktorską obronił w 1995, habilitacyjną 2004r. Jest profesorem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe koncentrują się wokół programowalnych układów i systemów cyfrowych.



e-mail: dariusz.kania@polsl.pl

Dr inż. Józef KULISZ

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej. Pracę doktorską obronił w 2003 r. Jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe to układy logiki programowalnej, sterowniki programowalne i układy z przelączanymi pojemnościami.



e-mail: jozef.kulisz@polsl.pl

Streszczenie

W artykule przedstawiono przykład zastosowania nowego rodzaju grafu - grafu niezgodności i dopełnień. Specyficzną cechą tego grafu jest to, że zawiera on dwa rodzaje krawędzi: krawędzie skojarzone z relacjami niezgodności oraz krawędzie skojarzone z relacjami dopełniania. Graf może być wykorzystywany w szeregu problemów optymalizacyjnych, w których rozważane są relacje niezgodności i dopełniania wzorców bitowych. W artykule zaprezentowano wykorzystanie grafu w procesie kodowania stanów asynchronicznych układów sekwencyjnych. Przedstawiono też odpowiednie algorytmy tworzenia grafu i kolorowania jego wierzchołków.

Słowa kluczowe: problem pokrycia i dopełnienia, teoria grafów, kodowanie stanów, sekwencyjne układy asynchroniczne.

An application of the Incompatibility and Complement Graph to asynchronous FSM coding

Abstract

The paper presents an application of a novel concept of graph – the Incompatibility and Complement Graph. A specific feature of the graph is that it contains two kinds of edges: connecting mutually incompatible nodes, and connecting mutually complementing nodes [3, 4]. The graph can be useful in certain class of optimization problems, in which compatibility of bit patterns in both the true and the complemented form has to be analyzed [5]. An example of such a problem is covering analysis in asynchronous FSM coding. The relevant coding method was presented by Tracey [1]. The method consists of several steps. In one of the steps a Boolean matrix is built, describing partitions of the relevant state set, which are required to provide coding free from critical races. In the subsequent step the Boolean matrix has to be reduced. During this step compatibility of the matrix rows both in the true, and the complemented form has to be analysed. For this purpose the Row Incompatibility and Complement graph can be used. The paper presents a simple example explaining the method. Appropriate algorithms for the graph building (Fig. 3) and colouring (Fig. 4) are also presented.

Keywords: covering and complement problem, graph theory, state assignment, asynchronous FSMs.

1. Wprowadzenie

W ostatnim okresie czasu obserwuje się zwiększone zainteresowanie realizacją układów sterujących w postaci struktur asynchronicznych. Wpływa na to nie tylko chęć poprawy parametrów dynamicznych układów sekwencyjnych. Jedną z korzystnych właściwości układów asynchronicznych jest rozproszony w czasie charakter generowanych zakłóceń. Właściwość ta nabiera istotnego znaczenia w przypadku konieczności konstruowania urządzeń spełniających wymogi kompatybilności elektromagnetycznej. Na szczególną uwagę zasługują struktury realizowane w postaci układów GALS (Globally Asynchronous Locally Synchronous),

w których udało się połączyć korzystne cechy dwóch rodzajów implementacji: względną prostotę syntezy układu synchronicznego i poprawę jego właściwości poprzez nadrzędną rolę mechanizmów asynchronicznych. Głównym problemem syntezy układu asynchronicznego jest trudność zapanowania nad wyścigami krytycznymi. Istota eliminowania wyścigów krytycznych polega na odpowiednim kodowaniu stanów wewnętrznych automatu [1].

Celem artykułu jest przedstawienie sposobu wykorzystania grafu niezgodności i dopełnień w procesie kodowania stanów wykorzystującym ideę podziałów. Grafy niezgodności i dopełnień pierwotnie wprowadzone zostały w celu rozwiązania pewnych problemów pojawiających się w procesie dekompozycji [2 – 4]. Okazuje się jednak, że narzędzie to jest uniwersalne, a jego zastosowanie w procesie syntezy układów cyfrowych może być znacznie szersze [5].

Całość artykułu została podzielona na 2 części. Po krótkim wprowadzeniu w rozdziale drugim na prostym przykładzie przedstawiono problemy kodowania stanów wewnętrznych automatów asynchronicznych. Istota kodowania sprowadza się do zastosowania znanej z literatury metody podziałów Tracey'a [1]. W kolejnym rozdziale zaprezentowano oryginalny algorytm kolorowania wierzchołków grafu niezgodności i dopełnień, który wykorzystano do optymalizacji procesu sklejenia wierszy pierwotnej tablicy podziałów. Przedstawiony heurystyczny algorytm o liniowej złożoności pozwala na szybkie wyznaczenie kodów poszczególnych stanów. Całość artykułu kończy podsumowanie.

2. Podstawy syntezy asynchronicznych układów sekwencyjnych

Kluczowym etapem syntezy automatu asynchronicznego jest etap kodowania stanów wewnętrznych. Rozważmy problem kodowania stanów dla automatu opisanego tablicą przejść-wyjść przedstawioną na rys. 1.

	000	001	011	010	110	111	101	100
1	1	1		5				
2	2	1	2		4	2	2	2
3	1	3	3	4	3	3		
4			2	4	4	5		
5			3	5	5	5		

Rys. 1. Tablica przejść-wyjść przykładowego automatu sekwencyjnego (zaciemnione pola oznaczają stany stabilne)

Fig. 1. Program table of the example FSM (shaded cells denote stable states)

Istota kodowania stanów automatów asynchronicznych polega na takim przypisaniu kodów poszczególnym stanom, aby wyeliminować wyścigi krytyczne. Jedną z metod, zaprezentowaną w pracy [1], rozpoczyna się od kroku polegającego na wyszukaniu dla każdej kolumny tablicy przejść pomiędzy stanami, które mu-

szą zostać rozdzielone od siebie przez odpowiednie kodowanie, aby przejścia te nie prowadziły do powstania wyścigów krytycznych. Podziały wyszukiwane są więc dla wszystkich wektorów wejściowych. Oddzielone od siebie muszą zostać wszystkie przejścia kończące się w różnych stanach stabilnych. I tak dla wektora 000 musi istnieć zmienna wewnętrzna rozdzielająca przejście $3 \rightarrow 1$, od stanu 2. Oznaczmy konieczność wystąpienia takiego podziału jako $\Pi_1 = \{13,2\}$. Analiza pozostałych przejść występujących w automacie z rys. 1 prowadzi do znalezienia następujących podziałów: $\Pi_2 = \{12,3\}$; $\Pi_3 = \{24,35\}$; $\Pi_4 = \{15,34\}$; $\Pi_5 = \{24,3\}$; $\Pi_6 = \{24,5\}$; $\Pi_7 = \{2,45\}$; $\Pi_8 = \{3,45\}$.

Po zdefiniowaniu wszystkich wymaganych podziałów w drugim kroku metody tworzy się tablicę podziałów. Kolumny tej tablicy skojarzone są ze stanami automatu. Wierszom natomiast przyporządkowuje się odpowiednie podziały. Elementami tablicy są wartości "0", "1" i "-". Rozdział poszczególnych przejść wprowadza się poprzez arbitralne przyporządkowanie jednej parze stanów wartości "0", a drugiej wartości "1". Wartość "-" wskazuje stany nie wchodzące do danego podziału. W ten sposób powstaje tablica przedstawiona na rys. 2.

	1	2	3	4	5
Π_1	0	1	0	-	-
Π_2	0	0	1	-	-
Π_3	-	0	1	0	1
Π_4	0	-	1	1	1
Π_5	-	0	1	0	-
Π_6	-	0	-	0	1
Π_7	-	0	-	1	1
Π_8	-	-	0	1	1

Rys. 2. Tablica podziałów dla przykładowego automatu

Fig. 2. The Boolean matrix corresponding to the example program table

Z każdym podziałem wprowadzonym w pierwszym kroku metody można utożsamić pewną prowizoryczną zmienną wewnętrzną. Tablica podziałów definiuje więc dla automatu pewne prowizoryczne kodowanie, które zapewni wprawdzie eliminację wyścigów krytycznych, jest jednak zwykle bardzo nadmiarowe.

W kolejnym kroku metody tablica podziałów poddawana jest redukcji. Problem polega na tym, aby znaleźć minimalny zbiór zmiennych wewnętrznych zapewniających spełnienie wszystkich podziałów. Istota redukcji tablicy polega na sklejaniu wierszy zgodnych tzn. takich, które dla wszystkich stanów (kolumn tablicy podziałów) mają pary wartości różne od $\langle 0,1 \rangle$ i $\langle 1,0 \rangle$. Należy zauważyć, że wartości "0" i "1" przypisano do stanów automatu przy definiowaniu tablicy podziałów w sposób arbitralny. Dany podział byłby spełniony także wtedy, gdyby odpowiadający mu wiersz w tablicy został zanegowany. Stąd w procesie rozpatrywania zgodności trzeba uwzględnić również relację zgodności jednego wiersza z negacją drugiego.

Relacje pomiędzy wierszami tablicy podziałów można przedstawić w postaci grafu niezgodności i dopełnień wierszy [2 – 4], a sam proces redukcji można przeprowadzić w oparciu o ten graf. W celu przedstawienia istoty metody konieczne jest wprowadzenie kilku podstawowych pojęć:

Parę komórek (i, j) należących do tej samej kolumny tablicy podziałów nazywamy niezgodną, jeżeli wartości w nich zawarte wynoszą odpowiednio $(1,0)$ lub $(0,1)$.

Jeżeli w zbiorze par komórek należących do dwóch wierszy tablicy wystąpi przynajmniej jedna para komórek niezgodnych, to takie wiersze nazywamy **wierszami niezgodnymi**.

Jeżeli w zbiorze par komórek należących do dwóch różnych wierszy nie występują pary $(1,1)$ i $(0,0)$, to mówimy, że jeden wiersz jest **dopełnieniem** drugiego.

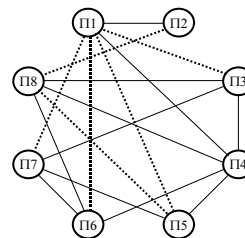
Grafem niezgodności i dopełnień wierszy nazywamy graf $G(Y,U)$, gdzie Y jest zbiorem wierzchołków odpowiadających wierszom tablicy, natomiast $U = U_n \cup U_d$ jest zbiorem krawędzi, gdzie: U_n - jest zbiorem krawędzi łączących wierzchołki odpowiadające wierszom niezgodnym, U_d - jest zbiorem krawędzi łączą-

cych wierzchołki odpowiadające wierszom, z których jeden jest dopełnieniem drugiego.

Krawędzie należące do zbioru U_n będą oznaczane linią ciągłą, natomiast krawędzie należące do zbioru U_d - linią przerywaną.

Przykład 1:

Wyznamy graf niezgodności i dopełnień wierszy dla tablicy przedstawionej na rys. 2. Analiza poszczególnych par wierszy pozwala utworzyć graf niezgodności i dopełnień wierszy przedstawiony na rys. 3.



Rys. 3. Graf niezgodności i dopełnień wierszy opisujący relacje pomiędzy wierszami tablicy podziałów z rysunku 2

Fig. 3. The Row Incompatibility and Complement Graph describing relationships between rows in the Boolean matrix presented in Fig. 2

Zaproponowany w dalszej części tej pracy algorytm kolorowania wierzchołków grafu pozwala na wyznaczenie minimalnego zbioru zmiennych wewnętrznych potrzebnych do zakodowania stanów wewnętrznych automatu. Sposób kodowania zapewni spełnienie wszystkich podziałów. Istota kolorowania grafu polega na przypisaniu wierzchołkom grafu minimalnej liczby różnych kolorów w taki sposób, aby każde dwa wierzchołki połączone ciągłą krawędzią otrzymały różne kolory.

3. Algorytm kolorowania wierzchołków grafu niezgodności i dopełnień wierszy

Algorytm kolorowania grafu niezgodności i dopełnień wierszy składa się z kilku kroków. W pierwszym kroku poszczególnym wierzchołkom przyporządkowywane są wszystkie możliwe kolory dozwolone i kolory dopełnień. Algorytm kolorowania wierzchołków oparty jest na sekwencyjnym wyborze wierzchołków, którym przypisywany jest **kolor dozwolony** (oznaczany dużą literą) lub **kolor dopełnienia** (oznaczany dużą literą z kreską u góry). W miarę możliwości kolejnym wierzchołkom przypisuje się kolory, które zostały już przypisane jakiemuś innemu wierzchołkowi. Po przypisaniu w i -tym kroku wybranemu wierzchołkowi koloru dozwolonego lub koloru dopełnienia, wszystkim wierzchołkom połączonym krawędziami ciągłymi z wybranym wierzchołkiem przypisywane są **kolory zabronione** (oznaczane małą literą), a wszystkim wierzchołkom połączonym z nim krawędziami przerywanymi - kolory dopełnienia.

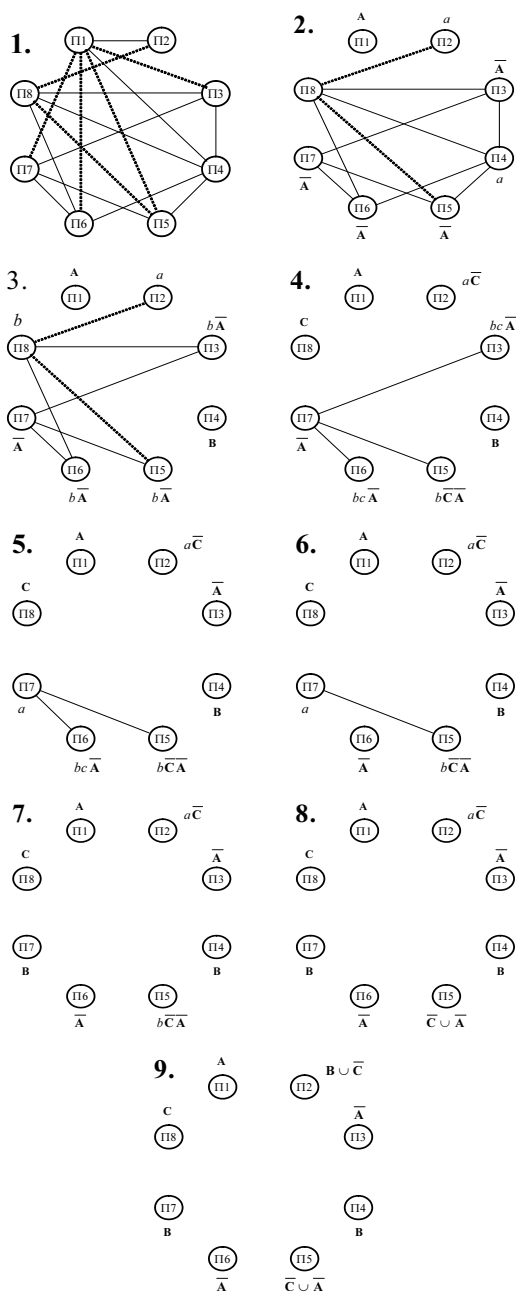
Wybór i -tego wierzchołka prowadzony jest zgodnie z następującymi zasadami:

- wybierany jest wierzchołek z maksymalną liczbą kolorów zabronionych; przyporządkowywany jest mu kolor dozwolony (jeżeli jest to możliwe, kolor ze zbioru już użytych kolorów);
- spośród wierzchołków z identyczną liczbą kolorów zabronionych wybierany jest ten, do którego dochodzi maksymalna liczba krawędzi;
- spośród wierzchołków z identyczną liczbą kolorów zabronionych i maksymalną liczbą krawędzi wybierany jest ten, który ma dodatkowo maksymalną liczbę kolorów dopełnień (jeżeli jest to możliwe, przyporządkowywany jest mu kolor dopełnienia);
- spośród wierzchołków z identyczną liczbą kolorów zabronionych, kolorów dopełnień i krawędzi wybierany jest ten, do którego dochodzi maksymalna liczba krawędzi ciągłych.

Po wybraniu wierzchołka i przypisaniu odpowiednim wierzchołkom kolorów dozwolonych, kolorów dopełnień i kolorów zabronionych, wykonywana jest redukcja grafu, polegająca na eliminacji krawędzi łączących wybrany wierzchołek z innymi wierzchołkami grafu. Następnie wybierany jest kolejny (i+1) wierzchołek, tym razem na podstawie analizy grafu zredukowanego. Na koniec wierzchołkom, które mają kolory zabronione i kolory dopełnień, przypisywane są wszystkie możliwe występujące na grafie kolory dozwolone i kolory dopełnień.

Przykład 2:

Zastosujmy przedstawiony algorytm kolorowania wierzchołków grafu dla grafu uzyskanego w przykładzie 3. Kolejne etapy kolorowania wierzchołków przedstawione są na rys. 4.



Rys. 4. Kolejne etapy kolorowania wierzchołków grafu niezgodności i dopełnień z rys. 3.

Fig. 4. Subsequent steps of colouring of the Incompatibility and Complement Graph presented in Fig. 3

Kolorowanie wierzchołków grafu niezgodności i dopełnień kolumn prowadzi do sklejania wierszy: $\{\pi_{11}, \pi_{13}, \pi_{15}, \pi_{16}\} \leftrightarrow A$;

$\{\pi_{2}, \pi_{4}, \pi_{7}\} \leftrightarrow B$ oraz $\{\pi_{8}\} \leftrightarrow C$. W efekcie sklejania powstaje zredukowana tablica podziałów przedstawiona na rys. 5.

	1	2	3	4	5
$\pi_{11}, \pi_{13}, \pi_{15}, \pi_{16}$	0	1	0	1	0
$\pi_{2}, \pi_{4}, \pi_{7}$	0	0	1	1	1
π_{8}	-	-	0	1	1

Rys. 5. Zredukowana tablica podziałów - wynik kolorowania grafu niezgodności i dopełnień wierszy

Fig. 5. A reduced Boolean matrix - the result of the Row Incompatibility and Complement Graph colouring

W poszczególnych wierszach tablicy z rys.5 znajdują się już minimalne kody stanów, które zapewniają brak wyścigów krytycznych przy wszystkich przejściach dla rozpatrywanego automatu.

4. Podsumowanie

W artykule przedstawiono przykład zastosowania nowego rodzaju grafu - grafu niezgodności i dopełnień. Krawędzie grafu niezgodności i dopełnień mogą reprezentować dwa rodzaje relacji: relacje niezgodności (oznaczane linią ciągłą) i relacje dopełniania (oznaczane linią przerywaną). Tego typu relacje pomiędzy wzorcami bitowymi są charakterystyczne dla szeregu problemów optymalizacyjnych występujących w procesach syntezy układów cyfrowych. Można je znaleźć między innymi w problemach opisujących zagadnienia dekompozycji [3, 4, 5] oraz sposobach wykorzystania elementów XOR [6, 7].

W niniejszym artykule pokazano wykorzystanie zaproponowanego grafu w procesie optymalizacji kodowania asynchronicznych układów sekwencyjnych. Opracowany algorytm może w znacznym stopniu uprościć proces sklejania podziałów, prowadząc do efektywnego uzyskiwania kodów wewnętrznych asynchronicznych automatów sekwencyjnych spełniających wszystkie wymagane podziały.

Należy podkreślić, że przedstawiony w artykule heurystyczny algorytm o złożoności czasowej $O(n)$ z powodzeniem może być wykorzystywany w wielu innych problemach optymalizacyjnych, związanych z rozwiązywaniem zadań pokrycia, w których występuje element dopełnienia. Tego typu technika stanowi istotne rozszerzenie klasycznego rozwiązania problemu pokrycia, w którym poszukuje się zbioru klik o minimalnej liczności, uzyskiwanego w wyniku analizy funkcji pokrycia.

5. Literatura

- [1] J. Tracey, Internal state assignments for asynchronous sequential machines, IEEE Transactions on Electronic Computer, Vol. EC-15, NO. 4 August 1966, pp. 551-560
- [2] D. Kania, Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL, Zeszyty Naukowe Politechniki Śląskiej, Nr 1619, Wydawnictwo Politechniki Śląskiej, Gliwice 2004
- [3] D. Kania, J. Kulisz, Logic synthesis for PAL-based CPLD-s based on two-stage decomposition, The Journal of Systems and Software 80, 2007, pp. 1129-1141
- [4] D. Kania, J. Kulisz, A. Milik, A novel method of two-stage decomposition dedicated for PAL-based CPLDs, Proceedings of Euromicro Symposium on Digital System Design, IEEE Computer Society Press, Porto, September, 2005, pp.114-121
- [5] D. Kania, J. Kulisz, The row incompatibility and complement graph - a novel concept of graph for decomposition, Programmable Devices and Embedded Systems, PDES 2006, Brno 14-16 February, 2006, pp.169-173
- [6] D. Kania, W. Grabiec, Synteza logiczna przeznaczona dla struktur CPLD z elementami XOR, Pomiary, Automatyka, Kontrola vol. 53, nr 7, 2007, ss. 54-56
- [7] D. Kania, W. Grabiec, Synteza logiczna dla struktur CPLD typu PAL wykorzystująca elementy XOR, Biuletyn WAT, Vol. LVI 3, Nr 3, (647), 2007, ss.229-241