Konferencja naukowa KNWS'04
„Informatyka - sztuka czy rzemiosło"
23-25 czerwca 2004, Czocha

BINARNE DIAGRAMY DECYZYJNE W PROJEKTOWANIU STEROWNIKÓW CYFROWYCH

Piotr Miczulski

Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50

e-mail: P.Miczulski@iie.uz.zgora.pl

STRESZCZENIE

Szereg zagadnień teorii grafów i zbiorów znajduje swoje zastosowanie w wielu dziedzinach techniki, w tym również w procesie projektowania sterowników cyfrowych. Jedną z częściej stosowanych odmian grafów są binarne diagramy decyzyjne, będące efektywnym narzędziem reprezentowania funkcji logicznych. Mogą być one wykorzystywane zarówno na etapie opisu zachowania układu cyfrowego, reprezentacji jego przestrzeni stanów, weryfikacji poprawności specyfikacji, syntezy, oraz na etapie jego testowania. W niniejszym referacie przedstawione zostały główne nurty badań prowadzonych przez autora, w których istotną rolę odgrywają różne odmiany diagramów decyzyjnych.

1. WPROWADZENIE

W czasach powszechnego wykorzystania komputerów do wykonywania złożonych obliczeń matematycznych, istotnym punktem stało się opracowanie odpowiednich struktur danych, pozwalających na efektywne ich przeprowadzanie. W przypadku przekształceń na funkcjach logicznych odpowiedzią, na tak postawione zapotrzebowanie, stały się Binarne Diagramy Decyzyjne (ang. *Binary Decision Diagram*). Obecnie istnieje wiele rodzajów binarnych diagramów decyzyjnych dedykowanych dla różnych typów funkcji logicznych [2, 7].

Wstępem do prac prowadzonych obecnie przez autora były prace nad wykorzystaniem diagramów decyzyjnych do rozwiązywania problemów kombinatorycznych, na przykładzie algorytmów kolorowania i wyznaczania pokryć zbiorów. Algorytmy te znajdują zastosowanie między innymi w teorii kodowania, podziale logiki w komputerach oraz optymalizacji układów cyfrowych. Ich zalety mogą zostać ponadto wykorzystane, na każdym niemal etapie projektowania sterowników cyfrowych, opisywanych za pomocą jednopoziomowych lub hierarchicznych sieci Petriego [1, 6, 8]. Zatem binarne diagramy decyzyjne mogą być wykorzystywane zarówno na etapie opisu zachowania układu cyfrowego (predykaty tranzycji), reprezentacji jego przestrzeni stanów (funkcja charakterystyczna [1]), weryfikacji poprawności specyfikacji, analizy właściwości sieci Petriego [6], syntezy [3], oraz na etapie testowania.

Poza doborem odpowiedniego typu diagramu decyzyjnego dla otrzymywanych funkcji logicznych, ważnym elementem staje się wybór jak najlepszej (pod względem szybkości wykonywania operacji, jak również ilości zajmowanej pamięci operacyjnej) biblioteki diagramów decyzyjnych [5]. W przypadku klasycznych zredukowanych, uporządkowanych diagramów decyzyjnych ROBDD (ang. *Reduced Ordered Binary Decision Diagram*) oraz unarnych diagramów decyzyjnych ZBDD (ang. *Zero – Suppressed Binary Decision Diagram*) na szczególną uwagę zasługuje biblioteka CUDD (ang. *Colorado University Decision Diagram Library*).

Głównym kierunkiem prowadzonych aktualnie prac jest połączenie zalet hierarchicznych sieci Petriego oraz różnych odmian binarnych diagramów decyzyjnych do projektowania złożonych (o dużej liczbie stanów globalnych), współbieżnych sterowników cyfrowych. Obecnie prace te zmierzają do odzwierciedlenia hierarchicznej struktury układu również na poziomie jego przestrzeni stanów, w postaci hierarchicznego grafu znakowań. Jednym z proponowanych rozwiązań, będącym rdzeniem dalszych badań, jest zastosowanie połączonego systemu diagramów decyzyjnych. Przeprowadzona analiza struktury otrzymywanych sieci Petriego wykazała również celowość stosowania różnych typów diagramów decyzyjnych, do różnych jej fragmentów oraz wykorzystanie wybranych rodzajów, ogólnie znanych, krawędzi z atrybutami. Zatem celem prowadzonych prac jest próba dostarczenia odpowiednich rozwiązań i narzędzi, pozwalających na szybsze oraz wygodniejsze projektowanie złożonych sterowników cyfrowych, modelowanych za pomocą sieci Petriego.

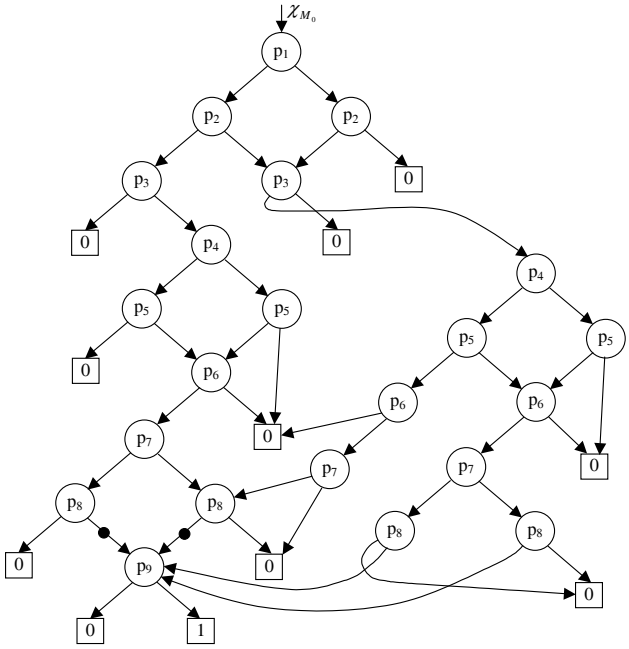
2. BINARNE DIAGRAMY DECYZYJNE

Klasycznym Zredukowanym, Uporządkowanym, Binarnym Diagramem Decyzyjnym ROBDD nazywamy skierowany i acykliczny graf z wyróżnionym węzłem, będącym korzeniem diagramu [7]. Diagram ROBDD posiada dwa typy węzłów: węzły nieterminalowe, reprezentujące zmienne funkcji boolowskiej oraz węzły terminalowe, o etykietach 0 i 1, reprezentujące wartości funkcji. Umownie przyjęto, że łuki łączące węzeł z jego lewym następnikiem odpowiadają wartości zerowej zmiennej decyzyjnej węzła, zaś łuki łączące węzeł z prawym jego następnikiem – wartości jeden zmiennej decyzyjnej węzła. Najczęściej do konstrukcji diagramu ROBDD, przedstawiającego zadaną funkcje boolowską, stosuje się tzw. rozkład Shannona. Rozkład ten dla funkcji *f* definiuje się jako:

$$f(x_1, ..., x_n) = x_1 f(1, x_2, ..., x_n) + \bar{x}_1 f(0, x_2, ..., x_n), \tag{1}$$

gdzie $f(1, x_2, ..., x_n)$ i $f(0, x_2, ..., x_n)$ są nazywane odpowiednio dodatnim i ujemnym kofaktorem funkcji *f*, zależnym od zmiennej x_i. Stosując rozkład Shannona (1) tak długo, aż w każdym z kofaktorów nie pozostanie żadna zmienna, otrzymujemy diagram BDD. Zatem

rozkład Shannona ustala związek miedzy funkcjami boolowskimi a Binarnymi Diagramami Decyzyjnymi. Zredukowane Binarne Diagramy Decyzyjne otrzymujemy przez zastosowanie odpowiednich reguł redukcji. Reguły te są inne dla klasycznych diagramów decyzyjnych, a inne np. dla diagramów unarnych [7]. Zastosowanie reguł redukcji dla diagramów ROBDD, powoduje, że nie istnieje w nich taki węzeł *v*, że lewy następnik tego węzła jest równy jego prawemu następnikowi oraz nie istnieją dwa takie węzły *v* i v' o identycznych zmiennych decyzyjnych, że poddiagramy o korzeniach *v* i v', są izomorficzne. Uporządkowanym BDD nazywamy diagram, w którym dla dowolnej ścieżki od korzenia do zmiennej terminalowej, zmienne występują w tym samym porządku leksykograficznym. Ważną cechą ROBDD jest fakt, że dla zadanej funkcji *f*, reprezentujące ją diagramy ROBDD są kanoniczne, co oznacza, że dwa diagramy ROBDD, reprezentują tą sama funkcję logiczną, wtedy i tylko wtedy, gdy rozpatrywane ROBDD są izomorficzne. Rysunek (rys. 1) przedstawia przykład diagramu ROBDD, reprezentującego funkcję charakterystyczną przestrzeni stanów układu, opisanego siecią Petriego z rysunku (rys. 2).



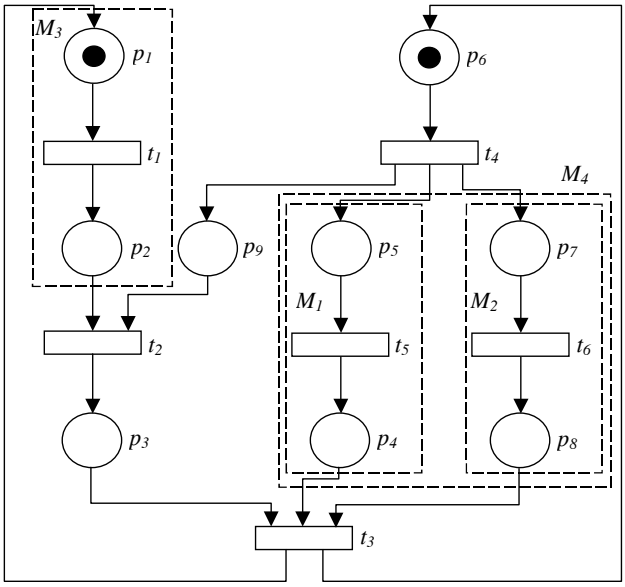
Rys. 1. Binarny Diagram Decyzyjny reprezentujący funkcję charakterystyczną

Efektywność binarnych diagramów decyzyjnych przy reprezentowaniu funkcji logicznych została potwierdzona wieloma badaniami, jakie były prowadzone na świecie w tym kierunku. Fakt ten jest również potwierdzany przez różnorodność dziedzin, w których znalazły one z powodzeniem zastosowanie. Warto tutaj jednak podkreślić, że ich zalety uwidaczniają się głównie dla funkcji logicznych o dużej liczbie zmiennych. Obok niewątpliwych zalet, istnieją

również ich trzy podstawowe negatywne właściwości, takie jak: wpływ uporządkowania zmiennych na rozmiar diagramu, wykładniczy wzrost liczby węzłów diagramu, niezależnie od uporządkowania zmiennych, dla pewnych rodzajów funkcji boolowskich oraz konieczność występowania takiego samego uporządkowania zmiennych logicznych, przy wykonywaniu operacji logicznych (w przeciwnym wypadku operacja ta jest NP – trudna).

3. POŁĄCZONY SYSTEM DIAGRAMÓW DECYZYJNYCH

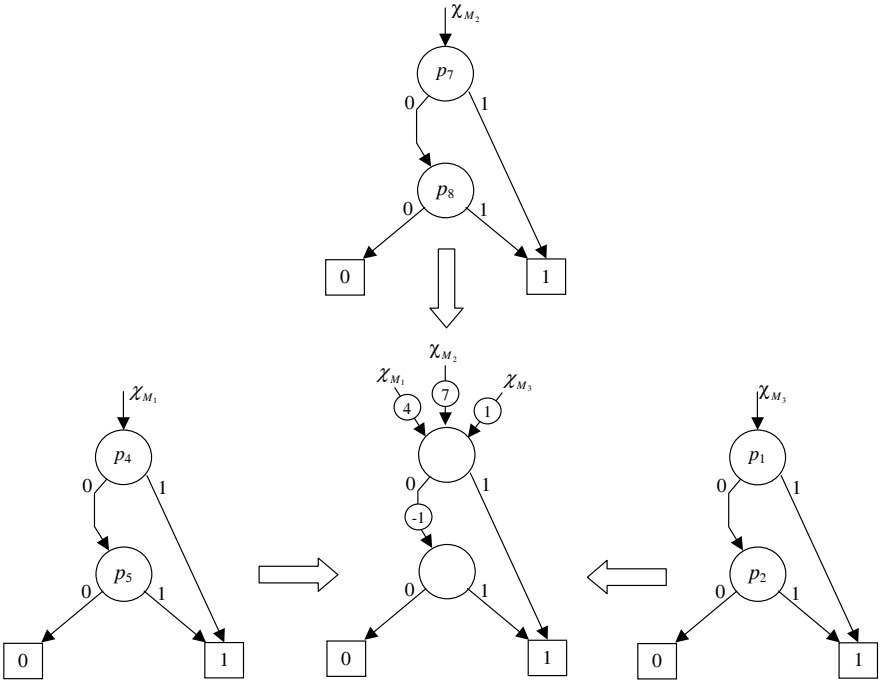
W dotychczas prowadzonych pracach, w kraju i na świecie [1, 8] przedstawiono możliwość reprezentowania całej przestrzeni stanów, jednopoziomowej sieci Petriego, z wykorzystaniem funkcji logicznej, reprezentowanej przez pojedynczy diagram BDD. Jednak pomimo zastosowania nowatorskich rozwiązań, przeprowadzone testy wykazały poważne trudności czasowe i pamięciowe przy projektowaniu układów powyżej miliona stanów. Zatem jednym z pierwszych etapów prowadzonych prac było opracowanie sposobu odzwierciedlania hierarchicznej struktury układu, opisywanego za pomocą hierarchicznej sieci Petriego, na poziomie jej przestrzeni stanów [4]. Rysunek 2 przedstawia jednopoziomową sieć Petriego z przykładowymi makromiejscami, utworzonymi za pomocą podstawowych reguł redukcji.



Rys. 2. Hierarchiczna sieć Petriego

Odzwierciedlenie hierarchicznej struktury przestrzeni stanów współbieżnego układu sterującego polega na przedstawieniu jego specyfikacji na różnych poziomach hierarchii (rys. 2), a następnie na wygenerowaniu dla każdego makromiejsca (poziomu hierarchii) jego przestrzeni stanów i przedstawieniu go w postaci funkcji logicznej. W kolejnym kroku każda z tak przygotowanych funkcji jest zapisywana jako Binarny Diagram Decyzyjny. Na przykład

przestrzeń stanów makromiejsc M_1, M_2 i M_3 została opisana następującymi funkcjami logicznymi: $\chi_{M_1} = p_4 + p_5$, $\chi_{M_2} = p_7 + p_8$ i $\chi_{M3} = p_1 + p_2$. Przy wykonanym podziale jednego diagramu decyzyjnego na większą ilość mniejszych diagramów decyzyjnych możliwa jest dalsza optymalizacja sposobu reprezentacji przestrzeni stanów. Jedną z proponowanych modyfikacji jest zastosowanie krawędzi z atrybutami, nazywanych *variable shifters* [7]. W rezultacie trzy osobne diagramy decyzyjne, opisujące funkcje charakterystyczne makromiejsc M_1, M_2 i M_3 mogą zostać zastąpione przez jeden diagram (rys. 3), co powoduje zmniejszenie liczby węzłów w połączonym systemie diagramów decyzyjnych. Zatem między innymi, w trakcie wykonywania analizy właściwości bezpieczeństwa hierarchicznej sieci Petriego, możliwa jest w danej chwili, przy tak przedstawionej przestrzeni stanów, analiza wybranego fragmentu układu [6]. Wówczas przeprowadzane operacje sprowadzają się do wykonywania operacji logicznych na ograniczonej liczbie diagramów, wchodzących w skład połączonego systemu diagramów.



Rys. 3. Wykorzystanie krawędzi z atrybutami do redukcji liczby węzłów

4. ZAKOŃCZENIE

Autor w swych pracach proponuje połączenie zalet hierarchicznych sieci Petriego z zaletami wynikającymi z zastosowania diagramów decyzyjnych. Na podstawie opracowanego sposobu reprezentowania przestrzeni stanów układu przyjęte zostały następujące główne kierunki badań:

- zdefiniowanie zasad oraz algorytmu automatycznego podziału płaskiej sieci Petriego na hierarchiczną strukturę makromiejsc,
- opracowanie zasad konstruowania przestrzeni stanów na dowolnym poziomie abstrakcji z zastosowaniem powiązanego systemu diagramów decyzyjnych,
- uwzględnienie w konstruowaniu przestrzeni stanów zarówno łuków zezwalających jak i zabraniających,
- dokonanie analizy przydatności innych typów diagramów decyzyjnych (np. UDD, KFDD [2]) i wybór najlepszego rozwiązania do reprezentowania funkcji charakterystycznych, opisujących przestrzeń stanów układu cyfrowego,
- opracowanie metod analizy wybranych właściwości hierarchicznych sieci Petriego z wykorzystaniem diagramów decyzyjnych,
- zaprojektowanie autorskiego, komputerowego systemu do analizy hierarchicznych sieci Petriego, w którym zostaną wykorzystane i przetestowane algorytmy opracowane w ramach badań.

Przedstawione prace prowadzone są w ramach grantu KBN, 4T11C 006 24.

LITERATURA

[1] K. Biliński: *Application of Petri Nets in parallel controller design*, PhD. Thesis, University of Bristol, Electrical and Electronic Departement, 1996

[2] R. Drechsler: *Binary Decision Diagram. Theory and Implementation*, Kluwer Academic Publishers, 1998

[3] T. Kozłowski, E. L. Dagless, J. M. Saul, M. Adamski and J. Szajna: *Parallel controller* synthesis using Petri nets, IEE Proceedings-E, Computer and Digital Techiques, July 1995, Vol. 142, No. 4, pp. 262-271

[4] P. Miczulski: *Algorytm konstruowania hierarchicznego grafu znakowań z wykorzystaniem przekształceń na funkcjach logicznych*, Reprogramowalne Układy Cyfrowe, Materiały V Krajowej Konferencji Naukowej, Szczecin, 2002

[5] P. Miczulski: *Analiza efektywności wybranych bibliotek BDD*, Międzynarodowe Sympozjum Naukowe Studentów i Młodych Pracowników Nauki, 2000

[6] P. Miczulski: *Weryfikacja poprawności opisu współbieżnych sterowników cyfrowych z wykorzystaniem diagramów decyzyjnych*, Reprogramowalne Układy Cyfrowe, Materiały VII Krajowej Konferencji Naukowej, Szczecin, 2004

[7] S. Minato: *Binary Decision Diagrams and Applications for VLSI CAD*, Kluwer Academic Publishers, 1996

[8] E. Pastor, O. Roig, J. Cortadella, M. R. Badia: *Petri Net Analysis Using Boolean Manipulation*, In Proceedings of 15th International Conference: Application and Theory of Petri Nets, volume 815 of Lecture Notes in Computer Science, Springer-Verlang, 1994