

Kodowanie stanów energooszczędnych automatów sekwencyjnych wykorzystujące algorytm Kernighana-Lina

Streszczenie. W artykule przedstawiono oryginalny dekompozycyjny algorytm kodowania stanów wewnętrznych automatów skończonych, który ukierunkowany jest na minimalizację poboru mocy. W kolejnych krokach następuje podział grafu stanowiącego probabilistyczny opis automatu realizowany za pomocą zmodyfikowanego algorytmu Kernighana-Lina. Wyniki eksperymentów wskazują, że opracowana metoda kodowania prowadzi do redukcji poboru mocy oraz zmniejszenia powierzchni układu.

Abstract. An original decomposition algorithm for state assignment of FSM is presented in the paper. This approach is oriented to reduction of power consumption. In the sequential steps, the probabilistic model of automaton is partitioned into two part by means of modified Kernighan-Lin algorithm. Experimental results demonstrate that the proposed state assignment approach can produce significant power savings and reducing of circuits' area. (**Low power FSM state assignment using a Kernighan-Lin algorithm**)

Słowa kluczowe: pobór mocy, kodowanie stanów, układy synchroniczne, automat skończony.

Keywords: power dissipation, state assignment, synchronous circuit, finite state machine.

Wstęp

Minimalizacja mocy jest jednym z najistotniejszych problemów projektowania energooszczędnych układów cyfrowych. Do niedawna istota syntezy logicznej sprowadzała się do maksymalizacji szybkości pracy układu lub minimalizacji jego powierzchni. Wraz ze zwiększaniem się skali integracji układów scalonych i tym samym gęstości mocy, dominującą rolę zaczyna odgrywać problem odprowadzania ciepła. Minimalizacja poboru mocy związana jest z kilkoma aspektami realizacji układów. Pierwszy dotyczy metod odprowadzania ciepła. Układy scalone, przy projektowaniu których nie uwzględniono minimalizacji poboru mocy wydzielają większą ilość ciepła, co prowadzi do konieczności stosowania radiatorów, wentylatorów i tym samym zwiększa koszt oraz gabaryty finalnego produktu. Drugi aspekt dotyczy układów mobilnych. W układach zasilanych bateryjnie stałe, pewne źródło zasilania jest elementem krytycznym. Minimalizacja poboru mocy w tych układach prowadzi do wydłużenia czasu ich pracy. Nie bez znaczenia jest również aspekt ekologiczny tego zagadnienia, ponieważ większy pobór energii urządzeń przenośnych, przekłada się na większą liczbę zużytych ogniw galwanicznych.

Obecnie większość układów cyfrowych wykonywanych jest w technologii CMOS. Całkowity pobór mocy w tego typu układach jest sumą tzw. mocy statycznej oraz mocy dynamicznej [1]. Pierwszy ze składników związany jest z prądami upływu płynącymi w stanie ustalonym, natomiast drugi, z przeładowywaniem pojemności obciążenia występującym w momencie zmiany stanu logicznego. Dominującym składnikiem jest moc dynamiczna, która może być minimalizowana w procesie syntezy logicznej.

Średnia moc dynamiczna pobierana przez układ wykonany w technologii CMOS jest proporcjonalna do średniej aktywności przełączeń [2]. Średnią moc dynamiczną rozpraszaną w jednej bramce g , oznaczaną jako P_g można wyrazić zależnością (1)

$$(1) \quad P_g = \frac{1}{2} C_g V_{DD}^2 f_{CLK} P_g$$

gdzie: C_g – pojemność obciążenia bramki g , V_{DD} – napięcie zasilania, f_{CLK} – częstotliwość sygnału zegarowego, p_g - prawdopodobieństwo przełączenia wyjścia bramki g .

Prawdopodobieństwo przełączenia wyjścia bramki g definiuje się jako liczbę zmian stanu logicznego na wyjściu tej bramki w trakcie okresu T [3].

W układach sekwencyjnych, zmniejszenie poboru mocy można uzyskać poprzez stosowanie różnorodnych rozwiązań układowych lub odpowiednich algorytmów kodowania stanów wewnętrznych automatu. Obydwa sposoby związane są z ograniczeniem liczby przełączeń poszczególnych elementów układu.

W literaturze przedstawiono szereg różnych metod kodowania stanów wewnętrznych automatów ukierunkowanych na minimalizację mocy [4, 5, 6, 7, 8, 9, 10]. Algorytm POW3 [4], znany również jako kolumnowy, polega na analizie licznosci tzw. klas nierozróżnialności. W pracy [5] zaproponowano metodę kodowania polegającą na tworzeniu słów kodowych, których długość jest odwrotnie proporcjonalna do prawdopodobieństwa statycznego poszczególnych stanów, zwaną kodowaniem Huffmana. Znane jest również kodowanie stanów w oparciu o tzw. drzewo rozpinające [6]. W procesie kodowania stosuje się algorytm wyżarzania [7] lub różnego typu algorytmy genetyczne [8]. W pracy [9] zaproponowano kilka różnych metod kodowania nazwanych odpowiednio metodami: Depth First, Min Distance, One Level, One Level Tree, Back Tracking. Każda z wymienionych powyżej strategii kodowania [9] składa się z dwóch etapów. W pierwszym wykonywane jest odpowiednie sortowanie stanów, a w drugim, następuje przypisanie kolejnym stanom słów kodowych w zależności od funkcji opisującej moc. W pracy [10] przedstawiono dwa algorytmy służące do wyznaczania kodów stanów: iteracyjny oraz sekwencyjny. Pierwszy z nich umożliwił zoptymalizowanie wyników kodowania, przeprowadzonego za pomocą innego dowolnego algorytmu. W algorytmie sekwencyjnym [10] proces kodowania uzależnia się od przypisanych uprzednio kodów.

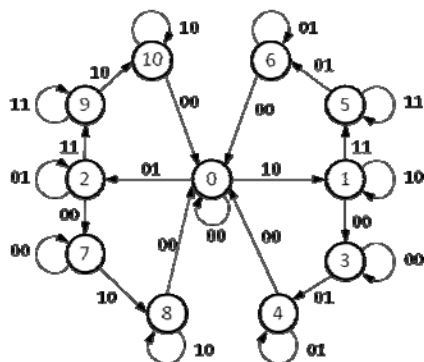
Dogłębna analiza znanych metod kodowania, wskazała na możliwość poprawy uzyskiwanych rezultatów, poprzez opracowanie dekompozycyjnego algorytmu kodowania [11]. Istota zaproponowanego w pracy [11] pomysłu związana była z oryginalnym kojarzeniem słów kodowych z symbolicznie określonymi stanami, odbywające się "bit to bicie". Okazuje się, że tego typu dekompozycyjne kodowanie stanów zachowuje elastyczność przyporządkowywania słów kodowych w całym procesie kodowania i zapewnia możliwość minimalizacji przełączeń elementów pamięci od początku do końca procesu kodowania. Istota pomysłu sprowadza się do tworzenia drzewa binarnego, którego węzły powstają w efekcie odpowiedniego podziału automatu skończonego.

Celem artykułu jest przedstawienie oryginalnej metody kodowania stanów automatów sekwencyjnych, stanowiącej rozwinięcie metody przedstawionej w pracy [11]. Istota proponowanego rozwiązania tkwi w optymalizacji podziału automatu w poszczególnych krokach tworzenia binarnego drzewa kodowania. W procesie podziału automatu wykorzystano jego probabilistyczny model i zmodyfikowany algorytm Kernighana-Lina.

Dekompozycyjna metodologia kodowania stanów wewnętrznych automatów sekwencyjnych

Automaty skończone można opisać za pomocą dyskretnych łańcuchów Markowa [12]. Dla podanego prawdopodobieństwa pojawienia się danych stanów logicznych na wejściach automatu, można wyznaczyć prawdopodobieństwa przejść pomiędzy poszczególnymi stanami. Prawdopodobieństwo statyczne, czyli prawdopodobieństwo znalezienia się automatu w określonym stanie po nieskończonej długim czasie, można obliczyć korzystając z równania Chapmana-Kołmogorowa [12]. Pozwala to na probabilistyczny opis automatu, w którym krawędziom łączącym odpowiednie stany nieskierowanego grafu, przyporządkowuje się wagi poszczególnych przejść.

Opracowany dekompozycyjny algorytm kodowania zostanie przedstawiony na przykładzie kodowania stanów wewnętrznych jednego z automatów testowych udostępnianych przez MCNC (Microelectronics Center of North Carolina) [13]. Wybrano automat train11, którego graf przejść przedstawiono na rysunku 1.



Rys.1. Graf automatu train11

Macierz prawdopodobieństwa przejść, przy założeniu, że prawdopodobieństwo wystąpienia jedynki i zera na wejściu układu jest takie samo i wynosi 0,5, przyjmuje postać:

$$(2) \quad P = \begin{bmatrix} 0,333 & 0,333 & 0,333 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,333 & 0 & 0,333 & 0 & 0,333 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,333 & 0 & 0 & 0 & 0,333 & 0 & 0,333 & 0 & 0 \\ 0 & 0 & 0 & 0,5 & 0,5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,5 & 0 & 0 & 0 & 0,5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,5 & 0,5 & 0 & 0 & 0 & 0 \\ 0,5 & 0 & 0 & 0 & 0 & 0 & 0,5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 & 0,5 & 0 & 0 \\ 0,5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 & 0,5 \\ 0,5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 \end{bmatrix}$$

Na podstawie macierzy (2) wyznacza się wektor prawdopodobieństwa statycznego:

$$(3) \quad p_s = [0,214 \ 0,107 \ 0,107 \ 0,071 \ 0,071 \ 0,071 \ 0,071 \ 0,071 \ 0,071 \ 0,071 \ 0,071]$$

Kolejny etap tworzenia opisu probabilistycznego automatu wymaga wyznaczenia prawdopodobieństwa

przejścia ze stanu S_i do stanu S_j pod warunkiem, że automat znajduje się w stanie S_i , zgodnie zależnością (4).

$$(4) \quad p_t(S_i \rightarrow S_j) = p_s(S_i)P(S_{ij})$$

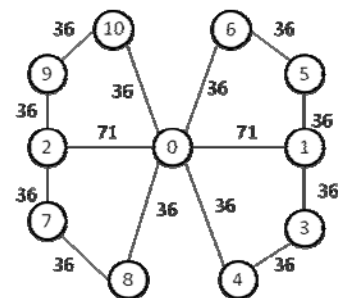
Uwzględnienie zależności (4) pozwala na utworzenie macierzy P_t

$$(5) \quad P_t = \begin{bmatrix} 0,071 & 0,071 & 0,071 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,036 & 0 & 0,036 & 0 & 0,036 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,036 & 0 & 0 & 0 & 0 & 0,036 & 0 & 0,036 & 0 \\ 0 & 0 & 0 & 0,036 & 0,036 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,036 & 0 & 0 & 0 & 0,036 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,036 & 0,036 & 0 & 0 & 0 & 0 \\ 0,036 & 0 & 0 & 0 & 0 & 0 & 0,036 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,036 & 0,036 & 0 & 0 \\ 0,036 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,036 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,036 & 0,036 \\ 0,036 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,036 \end{bmatrix}$$

Zmiana stanu automatu z S_i na S_j wymaga przełączenia dokładnie takiej samej liczby przerzutników, co zmiana ze stanu S_j na S_i . Można zatem przekształcić graf skierowany w graf nieskierowany, w którym wagi poszczególnych krawędzi będą przyjmować wartość:

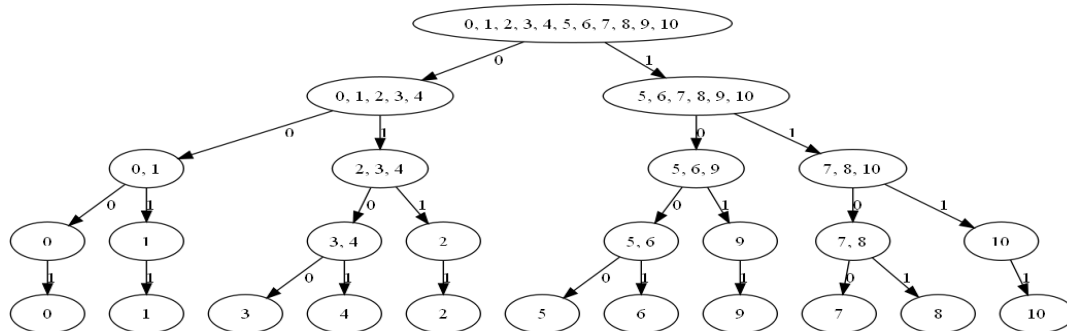
$$(6) \quad w_{ij} = p_t(S_i \rightarrow S_j) + p_t(S_j \rightarrow S_i)$$

Graf nieskierowany automatu train11 wraz z wyznaczonymi wagami poszczególnych krawędzi przedstawiono na rys. 2. Aby uzyskać lepszą czytelność rysunku wagi poszczególnych krawędzi przemnożono przez tysiąc.



Rys.2. Opis probabilistyczny automatu train11 (wagi krawędzi pomnożone przez 1000)

Otrzymany model probabilistyczny automatu jest wykorzystywany w algorytmie kodowania stanów LPBTE (Low Power Binary Encoding) [11]. Zgodnie z nim wybierana jest para stanów, które są połączone krawędzią o największej wadze, czyli $\{s_0, s_1\}$. Następnie do nowo tworzonego węzła drzewa binarnego dodawane są stany, dla których suma wag krawędzi pomiędzy stanami już należącymi do aktualnego węzła jest maksymalna. Zgodnie z tą zasadą do nowego węzła drzewa, w skład którego wchodzi stan s_0 i s_1 , dodane zostają stany s_2 , s_3 oraz s_4 . Po dodaniu tych trzech stanów, liczba stanów wchodzących w skład nowo tworzonego potomka wynosi 5. Ponieważ jest to maksymalna liczba stanów, które mogą zostać dodane do lewego potomka, pozostałe stany z dzielonego węzła drzewa (w tym przypadku korzenia), zostają przypisane do drugiego potomka. W analogiczny sposób wyznaczane są kolejne poziomy drzewa. Binarne drzewo kodowe rozpatrywanego automatu otrzymane w wyniku kodowania stanów za pomocą algorytmu LPBTE przedstawia rysunek 3. Precyzyjny opis algorytmu LPBTE można znaleźć w pracy [11].



Rys.3. Binarne drzewo kodowe automatu train11 otrzymane w wyniku działania algorytmu LPBTE

Algorytm Kernighana-Lina jako element podziału stanów automatu

Algorytm LPBTE przedstawiony w [11] ukierunkowany był na wybieranie wierzchołków połączonych krawędziami, skojarzonych z jak największymi wagami. Proponowana metoda jest najprostszą metodą podziału stanów, umożliwiającą kodowanie metodą „bit po bicie”, polegającą na sekwencyjnym tworzeniu binarnego drzewa kodowego. Proces podziału stanów może jednak uwzględniać szereg dodatkowych elementów wpływających na efektywność kodowania.

Proces tworzenia binarnego drzewa decyzyjnego można sprowadzić do odpowiedniego podziału zbioru wierzchołków grafu, stanowiącego probabilistyczny opis automatu sekwencyjnego. Istota zaproponowanego pomysłu tkwi w odpowiednim doborze stanów, wchodzących w skład poszczególnych węzłów drzewa binarnego.

Do podziału wierzchołków grafu można zaadaptować znany z literatury algorytm Kernighana-Lina [14]. Jest to algorytm opracowany na podstawie heurystycznych przesłanek, którego zadaniem jest podział wierzchołków grafu na dwa rozłączne podzbiory.

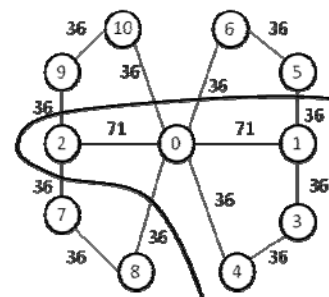
Analizując problem kodowania ukierunkowanego na minimalizację poboru mocy i sprowadzając go do odpowiedniego podziału grafu, można dojść do wniosku, że należy tak „ciąć” graf opisujący przełączenia automatu, aby suma wag „przeciętych” krawędzi była jak najmniejsza. Powyższy wniosek wynika z chęci minimalizacji przełączeń występujących w automacie, występujących w czasie zmiany jego stanów.

W ramach prac opracowano oryginalny sposób wykorzystania algorytmu Kernighana-Lina w procesie kodowania stanów. Pozwala on na optymalizację podziału wierzchołków tworzonego binarnego drzewa kodowego, prowadząc go tak, aby minimalizować odległość Hamminga kodów stanów, dla których w opisie probabilistycznym automatu występują krawędzie z największymi wagami. Podział stanów może być wstępnie dokonywany zgodnie z zaproponowaną w pracy [11] metodą, a następnie powinien być optymalizowany za pomocą algorytmu Kernighana-Lina.

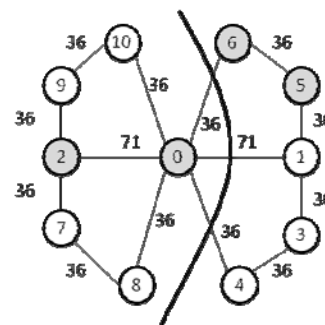
Dla analizowanego przykładu (train11) wstępny podział grafu, uzyskany za pomocą algorytmu LPBTE przedstawia rys. 4. W skład lewego potomka binarnego drzewa kodowego wchodzi stany s0, s1, s2, s3 i s4. Dla takiego podziału suma wag „ciętych” krawędzi, nazywana współczynnikiem cięcia, wynosi 216.

Zaproponowana optymalizacja podziału wierzchołków grafu polega na takim doborze krzywej cięcia, aby współczynnik cięcia przyjmował wartość minimalną. Oczywiście konieczne jest zapewnienie odpowiedniej liczności, uzyskanych w wyniku podziału podzbiorów. W wyniku poszukiwania optymalnego podziału wierzchołków

grafu realizowanego za pomocą algorytmu Kernighana-Lina uzyskujemy podział wierzchołków grafu przedstawiony na rysunku 5. Efektem cięcia jest więc podział wierzchołków, dla którego współczynnik cięcia wynosi tylko 143.



Rys.4. Podział grafu automatu train11 dla algorytmu LPBTE

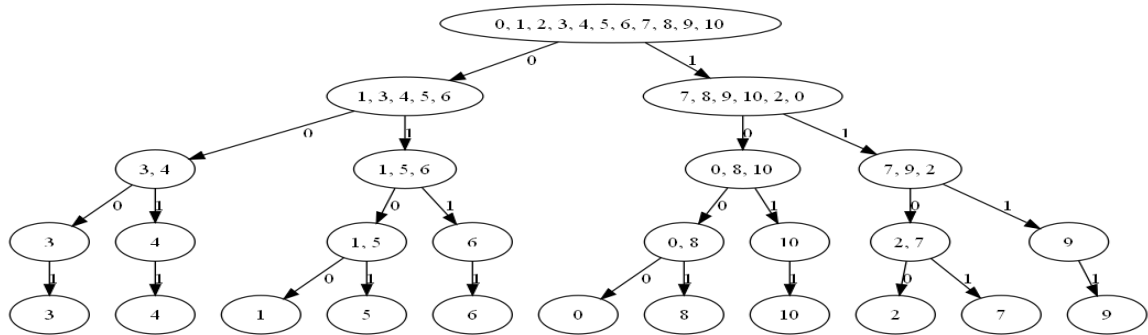


Rys.5. Podział grafu automatu train11 dla algorytmu LPBTE-KL

Kolorem szarym na rysunku 5 zostały wyróżnione wierzchołki, które na skutek zastosowanej optymalizacji przeniesiono z jednego do drugiego podzioru. Powyższy podział pozwala na utworzenie pierwszego poziomu binarnego drzewa kodowego. W kolejnych krokach tworzenia binarnego drzewa kodowego, algorytm Kernighana-Lina wykorzystywany jest do dalszego podziału wierzchołków, w efekcie czego powstaje kolejny poziom binarnego drzewa kodowego. W efekcie sekwencyjnego podziału powstających podgrafów powstaje binarne drzewo kodowe automatu train11 przedstawione na rysunku 6.

Wyniki eksperymentów

Przedstawiony algorytm został zaimplementowany w języku C++. Do estymacji mocy wykorzystano pakiet SIS. Obliczenia mocy zostały przeprowadzone dla: $f=20\text{MHz}$, $V_{DD}=5\text{V}$. Minimalizację części kombinacyjnej wykonano korzystając ze skryptu script.rugged. Eksperymenty wykonano, wykorzystując automaty testowe (ang. benchmarks) udostępniane przez MCNC (Microelectronics Center of North Carolina) [13]. Kodowanie stanów rozpatrywanymi algorytmami przeprowadzono na zbiorze dziesięciu automatów.



Rys.6. Binarne drzewo kodowe automatu train11 otrzymane w wyniku zastosowania algorytmu Kernighana-Lina

Tabela 1 przedstawia wyniki syntezy wybranych automatów testowych. Poszczególne kolumny zawierają pobieraną moc (P) oraz powierzchnię zajmowaną przez uzyskiwany układ (A). Porównano trzy algorytmy kodowania: popularne kodowanie 1 z N, algorytm JEDI wbudowany w środowisko SIS oraz przedstawiony w niniejszej publikacji algorytm LPBTE-KL (Low Power Binary Tree Encoding - Kernighan-Lin). Ostatni wiersz tabeli 1 zawiera liczby określające sumaryczną moc i całkowitą powierzchnię.

Tabela 1. Wyniki eksperymentów, P-moc, A-powierzchnia

FSM	One hot		JEDI		LPBTE-KL	
	P μ W	A	P μ W	A	P μ W	A
bbara	168,1	220	158,7	130	170,8	135
dk15	393,8	140	448,6	122	373,2	111
dk512	456,9	252	408,3	122	292,9	131
ex1	570,8	526	732,7	363	608,8	332
fetch	396,5	384	705,2	248	564,8	249
kirkman	663,8	347	763,9	291	654,1	269
log	161,3	261	386,4	191	295,9	205
s1	1076,0	668	513,1	219	600,1	303
sync	446,3	830	1107	480	899,2	514
train11	434,5	202	201,4	80	163,4	85
suma	4768,0	3830	5425,3	2246	4623,2	2334

Przedstawiony algorytm doprowadził do uzyskania najmniejszej mocy sumarycznej. Biorąc pod uwagę sumaryczną powierzchnię najlepszy wynik uzyskano za pomocą algorytmu JEDI, który z kolei najgorzej wypadł pod względem sumarycznej mocy. Algorytm LPBTE-KL doprowadził do rozwiązań cechujących się mniejszą mocą w przypadku pięciu automatów w porównaniu do rozwiązań uzyskanych za pomocą popularnego algorytmu one hot. Należy jednak zaznaczyć, że sumaryczna powierzchnia automatów uzyskanych w przypadku tego algorytmu jest aż o 64% większa w porównaniu do powierzchni automatów, których stany kodowano za pomocą algorytmu LPBTE-KL. Szczególną przewagę zaproponowanego algorytmu widać w przypadku automatu train11. W porównaniu do kodowania metodą „one hot”, za pomocą algorytmu LPBTE-KL uzyskano rozwiązanie, zużywające 62% mniej energii, przy jednoczesnej redukcji powierzchni o 58%.

Podsumowanie

W artykule przedstawiono nową metodę kodowania stanów automatów sekwencyjnych, która stanowi rozwinięcie opracowanej poprzednio dekompozycyjnej metody kodowania stanów. Metoda ta bazuje na tworzeniu drzewa binarnego w oparciu o probabilistyczny opis automatu. W trakcie tworzenia drzewa główną rolę odgrywa wykorzystany w oryginalny sposób algorytm Kernighana-Lina.

Wyniki eksperymentów wskazują jednoznacznie, że opracowana metoda kodowania umożliwia zmniejszenie poboru mocy uzyskiwanych rozwiązań w stosunku do innych, znanych z literatury metod kodowania. Z przeprowadzonych badań wynika również, że minimalizacja

mocy nie musi iść w parze z nadmierną ekspansją powierzchni, co stanowi istotną dodatkową wartość zaproponowanej metody kodowania stanów automatów sekwencyjnych.

LITERATURA

- [1] Pedram M., Power Minimization in IC Design, Principles and Applications, *ACM Trans. Design Automat. Electron. Syst.*, Vol.1, No. 1 (1996), 3-56
- [2] Cirit M.A., Estimating dynamic power consumption of CMOS circuits, *Proc. IEEE Int. Conf. CAD*, Nov. (1987), 534-537
- [3] Kentzer K., Ghosh A., Devadas S., White J., Estimation of average switching activity in combinational and sequential circuits, *Proc. Design Automation Conf.*, June (1992), 253-259
- [4] Benini L., DeMicheli G., State Assignment for Low Power Dissipation, *In IEEE Journal on Solid-state Circuits*, Vol. 30, No. 3 (1995), 258-268
- [5] Surti P., Chao L.F., Lower Power FSM Design Using Huffman-Style Encoding, *IEEE European Design and Test Conference*, (1997), 521-525
- [6] Nöth W., Kolla R., Spanning Tree Based State Encoding for Lower Power Dissipation, *Technical report, Dept. of Computer Science, University of Würzburg*, (1998)
- [7] Roy K., Prasad S. C., Circuit Activity Based Logic Synthesis for Low Power Reliable Operations, *IEEE Transactions on VLSI Systems*, Vol. 1, No. 4 (1993), 503-513
- [8] Venkataraman G., Reddy S.M., Pomeranz I., GALLOP: Genetic Algorithm Based Low Power FSM Synthesis by Simultaneous Partitioning and State Assignment, *16th International Conference on VLSI Design*, (2003), 533-538
- [9] Baccheta P., Daldoss L., Sciuto D., Silvano C., Lower-Power State Assignment Techniques for Finite State Machines, *IEEE International Symposium on Circuits and Systems*, Vol.2 (2000), 641-644
- [10] Salauyou V., Grześ T., Badania algorytmów kodowania stanów wewnętrznych automatu skończonego zorientowanych na minimalizację poboru mocy, *Pomiary, Automatyka, Kontrola*, R. 54, Nr 8 (2008), 499-501
- [11] Kajstura K., Kania D., Dekompozycyjna metoda kodowania stanów wewnętrznych automatu skończonego ukierunkowana na minimalizację mocy, *Przegląd Elektrotechniczny*, Nr 6 (2011), 146-149
- [12] Freitas A. T., Oliveira A. L., Implicit Resolution of the Chapman-Kolmogorov Equations for Sequential Circuits: An Application in Power Estimation, *Proceedings of DATE*, (2003), 764-769
- [13] Yang S., Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0, *Technical Report, Microelectronics Center of North Carolina*, (1991)
- [14] Kernighan B. W., Lin S., An efficient heuristic procedure for partitioning graphs, *Bell Systems Technical Journal*, Vol. 49, No. 1 (1970), 291-307

Autorzy: dr inż. Krzysztof Kajstura, Akademia Techniczno-Humanistyczna, Katedra Elektrotechniki i Automatyki, ul. Willowa 2, 43-309 Bielsko-Biała, E-mail: kkajstura@ath.bielsko.pl; Prof. dr hab. inż. Dariusz Kania, Politechnika Śląska, Instytut Elektroniki, Akademicka 16, 44-100 Gliwice, dkania@polsl.pl.