

Valery SALAUYOU, Tomasz GRZEŚ

POLITECHNIKA BIAŁOSTOCKA WYDZIAŁ INFORMATYKI, ul. Wiejska 45A, 15-351 Białystok

Algorytm kodowania stanów wewnętrznych automatu skończonego ze zmienną długością kodu do minimalizacji mocy

dr hab. inż. Valery SALAUYOU

Ukończył w 1978 r. studia na Wydziale Matematyki Stosowanej w Białoruskim Państwowym Uniwersytecie w Mińsku. W 1986 r. obronił rozprawę doktorską, a w 2003 r. uzyskał tytuł doktora habilitowanego. Od 25 lat pracuje w dziedzinie projektowania logicznego systemów cyfrowych.

e-mail: valsol@mail.ru



Dr inż. Tomasz GRZEŚ

Ukończył studia w Instytucie Informatyki Politechniki Białostockiej. Obronił rozprawę doktorską w 2010 r. na Wydziale Informatyki Politechniki Białostockiej, gdzie zajmuje stanowisko adiunkta. Jego zainteresowania naukowe to metody minimalizacji poboru energii układów sekwencyjnych, ze szczególnym uwzględnieniem układów o strukturze PLD/CPLD.

e-mail: t.grzes@pb.edu.pl



Streszczenie

W artykule przedstawiono dwie heurystyczne metody kodowania stanów wewnętrznych automatów skończonych, których celem jest zminimalizowanie poboru energii: ze stałą i ze zmienną długością kodu. Drugie podejście charakteryzuje się małym kosztem obliczeniowym. Badania eksperymentalne wykazują znaczące zmniejszenie poboru energii w przypadku pierwszej metody w porównaniu z algorytmem NOVA średnio o 39%. Druga metoda w porównaniu z pierwszą pozwala na zmniejszenie poboru energii nawet o 34%.

Słowa kluczowe: minimalizacja mocy, automat skończony, kodowanie stanów

Finite State Machine State Assignment Algorithm with Variable Code Length for Power Minimization

Abstract

We consider two heuristic methods of assigning the codes to the internal states of finite state machine to reduce energy consumption. In the first approach assumes a constant value of the code length of the internal states. In the second approach, the number of bits of code changed from minimum to the point where an increase in the length of code does not lead to a reduction in power consumption. The second approach is characterized by low computational cost, which allows its use for finite state machines with a large number of states. Experimental results have shown that the first method reduces the power consumption of finite automata, in comparison with the algorithm of NOVA, on average, 39%, and for some examples – by 68%. The second method, compared to the first, allows in some cases, reduce power consumption by 34%.

Keywords: power minimization, finite state machine, state assignment

1. Wprowadzenie

W ostatnich czasach, ze względu na upowszechnienie się pokładowych i mobilnych systemów wbudowanych (telefony komórkowe, odtwarzacze przenośne, nawigacje, kosmiczne systemy sterujące), ważnym problemem stało się zminimalizowanie poboru energii systemów cyfrowych. Istnieje wiele podejść do problemu zmniejszenia poboru energii automatów skończonych, np.

zastosowanie specjalnych metod kodowania stanów wewnętrznych [1-4], dekompozycja większych układów na mniejsze bloki [5], wykorzystanie wbudowanych bloków pamięci układów programowalnych (np. FPGA) [6] i inne.

W artykule przedstawiono dwie heurystyczne metody kodowania stanów wewnętrznych automatu skończonego do minimalizacji poboru energii. W pierwszym podejściu (Algorytm 1) budowany jest zbiór wszystkich możliwych kodów, z którego kolejno wybierane są najbardziej odpowiednie kody stanów wewnętrznych. W drugim podejściu (Algorytm 2) nie jest tworzony zbiór wszystkich możliwych kodów, a kod dla kolejnego stanu tworzony jest na podstawie określonych warunków. Cecha ta znacząco zmniejsza złożoność obliczeniową algorytmu i pozwala na zastosowanie w automatach o dużej liczbie stanów wewnętrznych.

2. Sformułowanie problemu

Celem zadania kodowania stanów wewnętrznych automatu prowadzącego do minimalizacji poboru energii jest takie kodowanie, przy którym prawdopodobieństwo zmiany stanu elementu pamięci będzie minimalne. Zadanie kodowania stanów wewnętrznych automatu skończonego w celu minimalizacji mocy można sformułować w następujący sposób.

Zadanie 1. Znaleźć takie kodowanie stanów wewnętrznych automatu skończonego, które spełnia zależność:

$$\sum_{i=1}^M \sum_{j=1}^M w_{i,j} \times H(k_i, k_j) = \min \quad (1)$$

przy założeniu następujących ograniczeń:

$$\sum_{i=1}^R k_i^r \oplus k_j^r \geq 1 \text{ dla każdego } k_i, i \neq j, i = \overline{1, R}, j = \overline{1, R} \quad (2)$$

gdzie: k_i^r – wartość bitu r kodu k_i , $r = \overline{1, R}$, $w_{i,j}$ – prawdopodobieństwo przejścia pomiędzy stanami a_i oraz a_j , a_i i $a_j \in A$.

Spełnienie równania (1) zapewni znalezienie kodowania, które daje minimalną moc, natomiast spełnienie ograniczenia (2) – deterministyczność realizacji automatu skończonego.

Podczas rozwiązywania zadania kodowania stanów wewnętrznych zamiast grafu automatu wykorzystuje się graf G_p prawdopodobieństw przejść pomiędzy stanami automatu skończonego [1], w którym waga krawędzi $w_{i,j}$ jest określona za pomocą następującego wyrażenia:

$$w_{i,j} = P(a_i \rightarrow a_j) + P(a_j \rightarrow a_i), \quad (3)$$

gdzie: $P(a_i \rightarrow a_j)$ oraz $P(a_j \rightarrow a_i)$ – prawdopodobieństwa przejścia odpowiednio ze stanu a_i do a_j i ze stanu a_j do a_i .

3. Algorytm sekwencyjny rozwiązania zadania 1

Główną ideą przedstawionego algorytmu jest sekwencyjne (z wyłączeniem pierwszych dwóch stanów) przypisywanie kodów stanom wewnętrznym automatu skończonego. Niech K^R oznacza zbiór wszystkich możliwych binarnych kodów o rozmiarze R ; K – zbiór kodów wybranych do kodowania stanów wewnętrznych automatu skończonego. W procesie kodowania stanów kody będą sekwencyjnie wybierane ze zbioru K^R i dodawane do zbioru K . Niech również określone zostaną dwa podzbiory zbioru stanów: A_C – zbiór stanów, którym kody zostały przypisane i A_U – zbiór stanów, którym kodów jeszcze nie przypisano, przy czym $A = A_C \cup A_U$, oraz $A_C \cap A_U = \emptyset$.

Istota algorytmu 1 rozwiązania zadania 1 przedstawia się następująco: na początku w grafie G_P znajdowane są dwa stany połączone krawędzią o największej wadze i tym stanom przypisywane są dwa kody ze zbioru K^R z minimalną odległością Hamminga. Następnie ze zbioru stanów jeszcze nie zakodowanych A_U wybiera się stan a_i , który w grafie G_P jest najsilniej związany ze stanami już zakodowanymi ze zbioru A_C . W procesie wyboru kodu k_i dla stanu a_i obliczana jest wartość funkcji $\gamma(a_i, k_i)$, która określa aktywność przełączania elementów pamięci przy przejściu automatu skończonego ze stanu a_i do już zakodowanych stanów ze zbioru A_C :

$$\gamma(a_i, k_i) = \sum_{a_j \in A_C} w_{i,j} \times H(k_i, k_j), \quad k_i \in K^R, \quad (5)$$

gdzie: $w_{i,j}$ – waga krawędzi grafu G_P łącząca stany a_i i a_j ; k_j – kod stanu a_j , $a_j \in A_C$. Do stanu a_i przypisywany jest taki kod, dla którego wartość funkcji $\gamma(a_i, k_i)$ jest najmniejsza. Proces kodowania stanów wewnętrznych jest kontynuowany do momentu, gdy wszystkie stany automatu skończonego zostaną zakodowane.

4. Zmiana liczby R bitów kodu

Liczba R bitów kodu stanu wewnętrznego może zmieniać się od wartości $\text{intlog}_2 M$ do M , gdzie M – liczba stanów wewnętrznych. Najczęściej przyjmuje się wartość R równą $\text{intlog}_2 M$. Jest oczywistym, że przy zwiększeniu wartości parametru R w zbiorze K^R zwiększy się liczba kodów z najmniejszą odległością Hamminga, co zgodnie z (1) będzie skutkowało zmniejszeniem mocy.

Z drugiej strony zwiększenie wartości R prowadzi do zwiększenia liczby elementów pamięci i sprzężeń zwrotnych automatu skończonego, a w konsekwencji do zwiększenia kosztów realizacji automatu. Dlatego w algorytmie 2 zwiększanie parametru R będzie ograniczone do momentu, gdy wzrost wartości R nie prowadzi do zmniejszenia mocy.

Podczas wykonania algorytmu 2 mogą pojawić się trudności z realizacją, gdyż ze wzrostem parametru R znacznie zwiększa się złożoność obliczeniowa algorytmu 1. Dlatego też należy opracować taki algorytm kodowania stanów wewnętrznych, który nie tworzy zbioru K^R wszystkich możliwych kodów długości R , jak również właściwy kod nie jest wybierany ze zbioru K^R a tworzony na podstawie pewnych warunków. Ostatnie wymaganie pozwala sformułować następujący problem.

Zadanie 2. Dla stanu a_i , $a_i \in A_U$, zbudować kod k_i o długości R bitów, dla którego spełnione są zależności:

$$H(k_i, k_j) \geq 1 \text{ dla każdego } k_j \in K \quad (4)$$

oraz

$$\gamma(a_i, k_i) = \min, \quad (5)$$

gdzie: K – zbiór kodów stanów uprzednio zakodowanych.

W algorytmie 3 najpierw znajdowany jest zbiór kodów K_S , które spełniają warunek (4) i różnią się od kodu k_s stanu a_s na minimalnej liczbie pozycji. Następnie spośród kodów ze zbioru K_S wybierany jest kod k_i dla stanu a_i spełniający warunek (5). Takie stopniowe spełnianie kryteriów optymalizacyjnych pozwala znaleźć jedynie przybliżone rozwiązanie zadania 2, jednakże znacznie zmniejsza liczbę obliczeń dla dużych wartości R .

5. Wnioski

Sekwencyjny algorytm 1 kodowania stanów wewnętrznych automatu skończonego minimalizujący pobór energii wykazał bardzo wysoką efektywność w porównaniu ze znanymi algorytmami kodowania stanów wewnętrznych automatu. Może on być polecany do stosowania w przypadku automatów z małą liczbą stanów wewnętrznych. Jednocześnie, w przypadku, gdy liczba bitów kodu stanu wynosi $R = \text{intlog}_2 M$, pozwala na zmniejszenie kosztu realizacji automatu. Algorytm 2, wykorzystujący zmianę długości R kodu stanu wewnętrznego, posiada o wiele mniejszą złożoność

obliczeniową i w niektórych przypadkach znacząco zmniejsza pobór energii. W związku z tym algorytm 2 może być stosowany w przypadku automatów o dużej liczbie stanów wewnętrznych.

Wymienione algorytmy umożliwiają wzięcie pod uwagę prawdopodobieństw pojawienia się jedynek (zer) na każdym z wejść automatu. W wynikach badań eksperymentalnych założono, że prawdopodobieństwo pojawienia się jedynki (zera) na każdym z wejść jest równe 0,5. Proponowane podejście pozwala na ustalanie różnych wartości prawdopodobieństw dla każdego z wejść automatu skończonego, co znacznie zwiększa efektywność algorytmów.

Celem dalszych prac będzie poprawienie efektywności zmniejszania poboru energii automatów skończonych przez proponowane heurystyczne algorytmy oraz zastosowanie nowych modeli strukturalnych, co pozwoli na wykorzystanie cech architektonicznych współczesnych układów programowalnych.

6. Literatura

- [1] Benini L., De Micheli G.: State Assignment for Low Power Dissipation. IEEE Journal on Solid-state Circuits, Vol. 30, No. 3 (1995), pp. 259-268.
- [2] Chattopadhyay S.: Low Power State Assignment and Flipflop Selection for Finite State Machine Synthesis – a Genetic Algorithmic Approach. IEE Proceedings – Computers and Digital Techniques, Volume: 148, Issue: 45, 2001, pp. 147-151.
- [3] Salauyou V., Grzes T.: FSM State Assignment Methods for Low-power Design. Proceedings of 6th International Conference on Computer Information Systems and Industrial Management Applications: CISIM'2007, Elk, June 28-30, IEEE Computer Society, Los Alamitos 2007, pp. 345-348.
- [4] Shiue W.-T.: Novel State Minimization and State Assignment in Finite State Machine Design for Low-power Portable Devices. Integration, the VLSI Journal, Volume 38, Issue 4 (April 2005), pp. 549-570.
- [5] Xia Y., Ye X., Wang L., Tao W., Almaini A.: A Uniform Framework of Low Power FSM Partition Approach. International Conference on Communications, Circuits and Systems Proceedings, Guilin, 25-28 June 2006, Volume 4, p. 2642-2647.
- [6] Tiwari A., Tomko K.A.: Saving Power by Mapping Finite-state Machines into Embedded Memory Blocks in FPGAs. Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 16-20 Feb. 2004, Vol. 2, pp. 916-921.
- [7] Grzes T., Salauyou V., Bulatava I.: Algorithms of coding the internal states of finite-state machine focused on the reduced power consumption. Radioelectronics and Communications Systems, 2010, V. 53, No. 5, pp. 265-273.
- [8] Grzes T., Salauyou V., Bulatava I.: Power estimation methods in digital circuit design. Optoelectronics, Instrumentation and Data Processing, 2009, V. 45, No. 6, pp. 576-583.
- [9] Tsui C.-Y., Monteiro J., Devadas S., Despain A.M., Lin B.: Power Estimation Methods for Sequential Logic Circuits. IEEE Transactions on VLSI Systems, Vol. 3, No. 3 (1995), pp. 404-416.
- [10] Villa T., Vincentelli A.S.: NOVA: State Assignment of Finite State Machines for Optimal Two-level Logic Implementation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 9, Issue 9, Sep 1990, pp. 905-924.
- [11] Lin B., Newton A. R.: Synthesis of Multiple Level Logic from Symbolic High-Level Description Languages. Proceedings of the IFIP Conference on VLSI, Munich, West Germany, August 1989, pp. 187-196.
- [12] Sentovich E.M., Singh K.J., Lavagno L., et al.: SIS: A system for sequential circuit synthesis. Memorandum № UCB/ERL M92/41, Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, May 1992. <http://www.eesc.berkeley.edu/Pubs./TechRpts/1992/ERL-92-41.pdf>
- [13] Yang S.: Logic synthesis and optimization benchmarks user guide. Version 3.0. Technical Report. North Carolina. Microelectronics Center of North Carolina, 1991.