

Alexander BARKALOV, Larysa TITARENKO, Sławomir CHMIELEWSKI
UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Optymalizacja skończonych automatów Moore'a w układach CPLD

Prof. dr hab. inż. Alexander BARKALOV

Prof. Alexander A. Barkalov w latach 1976-1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki Donieckiej. Współpracował aktywno z Instytutem Cybernetyki im. V.M. Glushkova w Kijowie, gdzie uzyskał tytuł doktora habilitowanego ze specjalnością informatyka. W latach 1996-2003 pracował jako profesor w Instytucie Informatyki Narodowej Politechniki Donieckiej. Od 2004 pracuje jako profesor na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.



e-mail: A.Barkalov@iie.uz.zgora.pl

Dr hab. inż. Larysa TITARENKO

Dr hab. Larysa Titarenko w 2004 roku obroniła rozprawę habilitacyjną i uzyskała tytuł doktora habilitowanego ze specjalnością telekomunikacja. W latach 2004-2005 pracowała jako profesor w Narodowym Uniwersytecie Radioelektroniki w Charkowie. Od 2005 pracuje jako adiunkt na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.



e-mail: L.Titarenko@iie.uz.zgora.pl

Mgr inż. Sławomir CHMIELEWSKI

Mgr inż. Sławomir Chmielewski absolwent Uniwersytetu Zielonogórskiego. Ukończył studia informatyczne o specjalizacji inżynieria komputerowa. Obecnie student pierwszego roku na studiach doktoranckich.



e-mail: S.Chmielewski@weit.uz.zgora.pl

Streszczenie

W pracy przedstawiona została metoda zmniejszania ilości makrokomórek w układach typu PAL przy pomocy skończonych automatów stanów z wyjściami typu Moore'a. Metoda ta jest oparta na wykorzystaniu nieużywanych wyjść osadzonych obszarów pamięci w celu reprezentacji kodu klasy pseudo-równoważnych stanów. Zaproponowane podejście pozwala zmniejszyć ilość wymaganego zużycia sprzętowego bez zmniejszenia wydajności systemów cyfrowych. Podany również jest przykład aplikacji zaproponowanego rozwiązania.

Słowa kluczowe: Automat stanów, CPLD, wbudowany blok pamięci, algorytmiczna sieć działań.

Optimization of logic circuit of Moore FSM on CPLD

Abstract

Method of decrease of number of PAL macrocells in the circuit of Moore FSM is proposed. Method is based on usage of free outputs of embedded memory blocks to represent the code of the class of the pseudoequivalent states. Proposed approach permits to decrease the hardware amount without decrease of digital system performance. An example of application of proposed method is given.

Keywords: FSM, CPLD, embedded memory blocks, flow-chart of algorithm.

1. Wprowadzenie

Jednostka sterująca (ang. Control Unit, CU) jest bardzo ważnym blokiem we wszystkich systemach cyfrowych, która współpracuje z pozostałymi blokami systemu [1, 5]. Skończony automat stanów (ang. Finite State Machine, FSM) z wyjściami typu Moore'a jest bardzo często używany jako reprezentacja jednostki sterującej (CU) [3, 4]. Aktualny stan mikroelektroniki pozwala zaimplementować złożone systemy cyfrowe używając pojedynczych układów scalonych [8, 9]. Dowolny logiczny system cyfrowy może być zaimplementowany w strukturach programowalnych PAL (ang. Programmable Array Logic) przy użyciu układów CPLD (ang. Complex Programmable Logic Devices) [6]. Funkcje tabelaryczne mogą być implementowane

przy użyciu wbudowanego bloku pamięci (ang. Embedded Memory Blocks, EMB) [4]. Wzrost zużycia zasobów sprzętowych jest jednym z aktualnych problemów w logicznych układach jednostki sterującej [3]. Rozwiązanie tego problemu pozwala na zmniejszenie zajętej powierzchni w układzie scalonym przez układ CU, jak również daje potencjalną możliwość zwiększenia liczby funkcji systemu cyfrowego wewnątrz pojedynczego układu. Specyficzną cechą układu PAL jest duża ilość wejść makrokomórek i ilość termów na makrokomórkę [7, 9]. Cechą automatów Moore'a jest istnienie pseudo-równoważnych stanów i regularny charakter mikrooperacji, które daje się implementować z użyciem EMB [8, 9]. W tym artykule proponowana jest metoda optymalizacji liczby makrokomórek typu PAL w jednostkach sterujących automatu typu Moore'a, przy użyciu wyżej wspomnianych cech.

2. Opis projektowania automatów Moore'a

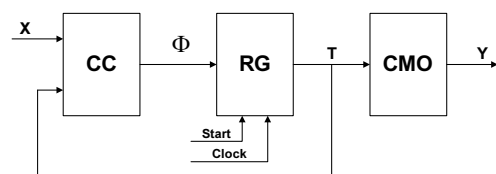
Algorytm sterujący systemem cyfrowym jest opisany poprzez sieć działań algorytmu (SDA) [2] $\Gamma = \Gamma(B, E)$, gdzie $B = \{b_0, b_E\} \vee E_1 \vee E_2$ jest zbiorem stanów automatu a E jest zbiorem krawędzi. Symbol b_0 jest początkowym a b_E jest końcowym stanem, E_1 jest operacyjnym, natomiast E_2 warunkowym stanem automatu. Stan $b_q \in E_1$ zawiera rodzinę mikrooperacji $Y(b_q) \subseteq Y$, gdzie $Y = \{y_1, \dots, y_N\}$ jest zbiorem mikrooperacji [5] systemu cyfrowego. Stan $b_q \in E_2$ zawiera warunek logiczny $x_e \in X$, gdzie $X = \{x_1, \dots, x_L\}$ jest zbiorem warunków logicznych [1]. Początkowy i końcowy stan SDA odpowiada wewnętrznemu stanowi $a_1 \in A$, gdzie $A = \{a_1, \dots, a_M\}$ jest zbiorem wewnętrznych stanów automatu Moore'a. Każdy operacyjny stan $b_q \in E_1$ odpowiada unikatowemu stanowi $a_m \in A$ i rodzinie $Y(b_q) = Y(a_m)$. Układy logiczne automatów Moore'a U_1 utworzone są przez system funkcji Boolowskich:

$$\Phi = \Phi(T, X), \quad (1)$$

$$Y = Y(T), \quad (2)$$

gdzie $T = \{T_1, \dots, T_R\}$ jest wewnętrzną zmienną zakodowanego stanu $a_m \in A$, $R = \lceil \log_2 M \rceil$; $\Phi = \{D_1, \dots, D_R\}$ jest zbiorem funkcji pobudzających pamięć automatu FSM.

Struktura diagramu FSM Moore'a U_1 jest pokazana na rys. 1.



Rys. 1. Struktura diagramu FSM Moore'a U_1
Fig. 1. Structural diagram of Moore FSM U_1

Układ kombinacyjny CC tworzy funkcje (1), które są funkcjami pobudzającymi przerzutniki typu „D” układów rejestrowych RG. Blok CMO tworzy mikrooperacje (2). Sygnał „Start” jest używany do załadowania kodu początkowego stanu do rejestru RG, sygnał zegarowy „Clock” powoduje przełączenie pamięci automatu z kodu $K(a_m)$ stanu obecnego $a_m \in A$ na kod $K(a_s)$ stanu następnego $a_s \in A$. W przypadku układów CPLD układ CC jest zaimplementowany z wykorzystaniem makrokomórek PAL, a układ CMO jest zaimplementowany z wykorzystaniem EMB.

Podstawą do formułowania systemu (1)-(2) jest struktura tabeli (ang. Direct Structural Table, DST) [5], gdzie kolumna: a_m oznacza stan bieżący automatu FSM; $K(a_m)$ będąca zakodowaną postacią stanu bieżącego posiadająca R-bitów; a_s jest stanem przejścia, $K(a_s)$ będąca zakodowaną postacią tego stanu, X_h jest koniunkcją elementów stanu X (lub ich dopełnieniem), która wywołuje przejście $\langle a_m, a_s \rangle$; $\Phi_h \subseteq \Phi$ jest zbiorem funkcji, która jest równa 1 układowi rejestrowemu RG z $K(a_m)$ do $K(a_s)$; h określa numer przejścia ($h = 1, \dots, H_1(\Gamma)$). Kolumna a_m zawiera rodzinę $Y(a_m) \subseteq Y$.

Z reguły liczba przejść $H_1(\Gamma)$ automatu Moore'a przekracza liczbę przejść $H_2(\Gamma)$ równoważnego automatu Mealy'ego. Prowadzi to do zwiększenia zajętości makrokomórek PAL i czasami liczby poziomów w układach kombinacyjnych automatu Moore'a w porównaniu z równoważnym automatem Mealy'ego [4]. Wartość $H_1(\Gamma)$ może być zmniejszona poprzez odpowiednie wykorzystanie istniejących pseudo-równoważnych stanów automatu Moore'a [1]. Stany $a_m, a_s \in A$ są stanami pseudo-równoważnymi, jeżeli wyjścia odpowiednich stanów operacyjnych są połączone z wejściem tego samego stanu SDA Γ . Niech $\Pi_A = \{B_1, \dots, B_I\}$ jest podzbiorem zbioru A przez klasy pseudo-równoważnych stanów ($I \leq M$). Oznaczmy klasę $B_i \in \Pi_A$ poprzez kod binarny $K(B_i)$ z $R_1 = \lceil \log_2 I \rceil$ [bitami]. Użyjmy zmiennych $\tau_r \in \tau$ dla takiego kodowania, gdzie $|\tau| = R_1$.

W tym przypadku specjalny konwerter kodu TC może być wprowadzony do układu U_1 . Tworzy on kody $K(B_i)$ na bazie kodów $K(a_m)$, gdzie $a_m \in B_i$. W tym przypadku CC tworzy funkcje

$$\Phi = \Phi(\tau, X), \quad (3)$$

a układ TC realizuje funkcje

$$\tau = \tau(T). \quad (4)$$

Układ TC jest implementowany przy użyciu EMBs [4].

Zostaje udowodnione, że system (3) ma $H_2(\Gamma)$ wyrażeń. Takie podejście ma jedną wadę: zużywa dodatkowe zasoby EMB, aby zaimplementować układ TC.

W pracy proponowana jest metoda projektowania, która pozwala zmniejszyć liczbę zużycia sprzętowego w układach CC bez konwertera kodu. Zaproponowana metoda oparta jest na następujących właściwościach CPLD [6, 7, 9]:

- liczba wejść makrokomórek PAL przekracza znacząco maksymalną możliwą ilość literałów w wyrażeniach systemu (1), który jest równoważny $L+R$;
- liczba wyjść EMB może być wybrana z jakiegoś ograniczonego obszaru $\{1, 2, 4, 8\}$.

3. Idea proponowanej metody

Oznaczmy t_F jako ustaloną liczbą wyjść bloku EMB oraz niech q jest ilością jego słów. Wartość t_F dla FSM U_1 jest określona przez

$$t_F = \lceil q / M \rceil. \quad (5)$$

Całkowita liczba wyjść t_S całego bloku EMBs w układzie CMO jest zdefiniowana jako

$$t_S = \lceil N / t_F \rceil * t_F. \quad (6)$$

W tym przypadku

$$\Delta_t = t_S - N \quad (7)$$

wyjścia nie są w użyciu, w celu reprezentowania mikrooperacji $y_n \in Y$.

Dla reprezentacji zbioru Π_A jako $\Pi_A = \Pi_B \cup \Pi_C$, gdzie $B_i \in \Pi_B$, jeżeli warunek

$$|B_i| > 1 \quad (8)$$

jest spełniony, natomiast $B_i \in \Pi_C$ w przypadku, gdy warunek (8) nie jest spełniony. Blok TC powinien tworzyć tylko kody klasy $B_i \in \Pi_B$. Zakodujemy każdą klasę $B_i \in \Pi_B$ przez binarny kod $K(B_i)$ z

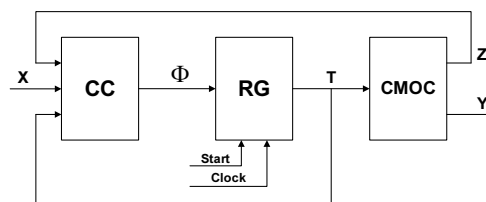
$$R_2 = \lceil \log_2 (M_1 + 1) \rceil \quad (9)$$

bitów, gdzie $|M_1| = \Pi_B$ oraz dodajmy 1, aby wskazać przypadek, kiedy $B_i \notin \Pi_B$.

Oznaczmy $z_r \in Z$ dla takiego kodowania, gdzie $|Z| = R_2$. W przypadku, kiedy warunek

$$\Delta_t \geq R_2 \quad (10)$$

będzie spełniony wtedy SDA Γ może być interpretowany przez proponowany automat Moore'a U_2 (rys. 2).



Rys. 2. Struktura diagramu FSM Moore'a U_2
Fig. 2. Structural diagram of Moore FSM U_2

W tym przypadku układ CC realizuje następujący system funkcji

$$\Phi = \Phi(T, Z, X), \quad (11)$$

działanie układu CMO jest zgodne z funkcją (2) oraz

$$Z = Z(T). \quad (12)$$

Zmienne $T_r \in T$ reprezentowane przez kody $K(a_m)$, gdzie $a_m \in B_i$ i $B_i \in \Pi_C$. Proponowana metoda pozwala zmniejszyć liczbę wyrażeń w systemie Φ aż do $H_2(\Gamma)$ i liczba EMBs jest równa dla obu układów CMO i CMO. Dostrzec można, że układ U_2 nie zawiera układu TC.

Liczba wejść w makrokomórkach PAL U_2 wzrasta aż do $L+R+R_2$, ale to nie wpływa na zużycie sprzętowe w układach CC w porównaniu z automatami Moore'a z układem TC.

Metoda projektowania układów cyfrowych U_2 różni się od metody projektowania U_1 [2] tylko w niektórych szczegółach. Rozważmy przykład projektu automatu Moore'a $U_2(\Gamma_1)$, gdzie symbol $U_i(\Gamma_i)$ pozostanie dla interpretacji SDA Γ_i przez automat Moore'a z struktury U_i .

4. Przykład proponowanej metody

Przykład algorytmu sterowania, który jest reprezentowany przez SDA Γ_1 przedstawiony został na rys. 3, gdzie $M = 16$. Automat SDA Γ_1 jest zbiorem przepływu stanów [2], gdzie stan $b_q \in E_2$ jest wymieniany przez przepływ stanu $a_m \in A$:

$$\begin{aligned} a_1 &\rightarrow a_2; a_2 \rightarrow a_3; a_7 \rightarrow a_8; a_{15} \rightarrow a_{16}; a_{16} \rightarrow a_1; \\ a_3 &\rightarrow x_1x_2a_4 \vee x_1/x_2a_5 \vee /x_1x_3a_6 \vee /x_1/x_3a_7; \\ a_4, a_5, a_6 &\rightarrow x_3x_4a_8 \vee x_3/x_4a_9 \vee /x_3x_5a_{10} \vee /x_3/x_5a_7; \\ a_8, a_9, a_{10} &\rightarrow x_4x_3a_{11} \vee x_4/x_3a_{12} \vee /x_4x_5a_{13} \vee /x_4/x_5a_{14}; \\ a_{11}, a_{12}, a_{13}, a_{14} &\rightarrow a_{15}. \end{aligned} \quad (13)$$

