

Małgorzata KOŁOPIEŃCZYK, Alexander BARKALOV, Larysa TITARENKO

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Synteza automatu Mealy'ego z wbudowanym blokiem pamięci w strukturach programowalnych

dr inż. Małgorzata KOŁOPIEŃCZYK

Absolwentka Wydziału Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego (1999 r.). Obecnie adiunkt w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Zajmuje się następującymi zagadnieniami: projektowaniem i syntezą sterowników logicznych oraz inżynierią oprogramowania.



e-mail: m.kolopienczyk@iie.uz.zgora.pl

prof. dr hab. inż. Alexander BARKALOV

Prof. Alexander A. Barkalov w latach 1976-1996 był pracownikiem Instytutu 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. 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

Streszczenie

W artykule zostanie przedstawiona metoda umożliwiająca syntezę skończonego automatu stanów typu Mealy'ego z wbudowanym blokiem pamięci (ang. Embedded Memory Blocks, EMB) w strukturach programowalnych typu FPGA. Metoda ta bazuje na przekształceniu strukturalnej tabeli przejść skończonego automatu stanów oraz na kodowaniu elementów podzbioru warunków logicznych. W artykule zostanie zaprezentowany przykład projektowania oraz wstępne wyniki badań.

Słowa kluczowe: Mealy FSM, FPGA, wbudowane bloki pamięci, projektowanie, układy logiczne.

Synthesis of Mealy FSM with EMB in FPGA structure

Abstract

In this article, we propose a method allowing decreasing the hardware amount of the block used for replacement of logical conditions. The method is based on transformation of FSM structure table and replacement of some logical conditions. Example of design and preliminary results of investigations are given.

Keywords: Mealy FSM, FPGA, embedded memory block, design, logic circuit.

1. Wprowadzenie

Zdecydowana większość systemów cyfrowych zawiera jednostkę sterującą odpowiedzialną za wzajemną współpracę wszystkich pozostałych elementów systemu [1]. W wielu praktycznych przypadkach do realizacji jednostki sterującej wykorzystywany jest skończony automat stanów Mealy'ego (ang. Finite State Machine, FSM) [2]. Obecnie układy programowalne FPGA, bardzo często są stosowane do implementacji takiego automatu [3, 4]. Większość układów FPGA bazuje na tablicach LUT połączonych z programowanymi przerzutnikami [5, 6]. Jedna tablica LUT razem z przerzutnikiem tworzy element logiczny (ang. Logic Element, LE). Dwa elementy LE tworzą blok *slice*. Dwa bloki *slice* tworzą konfigurowalny blok CLB (ang. Configurable Logic Block, CLB).

Wszystkie bloki CLB mogą być połączone za pośrednictwem programowanych połączeń, natomiast wewnątrz każdego CLB wykorzystywane są szybkie połączenia [4]. Przerzutnik elementu logicznego LE można pominąć, zatem wyjścia tablic LUT mogą być kombinacyjne lub rejestrowe. Regułą jest że, liczba wejść S tablicy LUT jest raczej mała $S \leq 6$ [5, 6]. To ograniczenie prowadzi do zastosowania dekompozycji funkcji boolowskich opisujących zachowanie automatu skończonego [7 - 9]. Zastosowanie dekompozycji powoduje zwiększenie liczby poziomów tablic LUT w układzie i komplikacje w połączeniach, co z kolei prowadzi do zwiększenia zużycia energii [10, 11]. W celu poprawienia parametrów układu logicznego, można zastosować wbudowane bloki pamięci do implementacji niektórych jego części [12, 16].

Bloki EMB, w nowoczesnych układach FPGA, mogą być konfigurowane. Oznacza to, że takie parametry jak liczba komórek i ich wyjść mogą być zmienione [4]. Typowe konfiguracje bloków EMB są następujące $16K \times 1$, $8K \times 2$, $4K \times 4$, $2K \times 8$, $1K \times 16$, 512×32 , 256×64 (bitów) [5, 6]. Tak więc, nowoczesne wbudowane bloki pamięci EMB są bardzo elastyczne i mogą być dostosowane do realizowania specyficznych zadań.

W pracach [17], zaproponowano metody implementacji automatów FSM z wbudowanymi blokami pamięci EMB. Do zmniejszenia liczby bloków EMB w układzie logicznym zastosowano kodowanie warunków logicznych [1]. Do kodowania zastosowano multiplexer. Multiplexer można zaimplementować za pomocą tablic LUT lub bloków EMB. W artykule zaproponowano metodę umożliwiającą zmniejszenie liczby zasobów sprzętowych bloku wykorzystywanego do kodowania warunków logicznych. Zaproponowana metoda polega na kodowaniu elementów tylko pewnego podzbioru zbioru warunków logicznych. Do opisu algorytmu sterowania zastosowano sieć działań (ang. Graph Scheme of Algorithm, GSA). Zwróćmy uwagę, że bazując na sieci działań możemy, w łatwy sposób, skonstruować graf przejść jak również tablicę przejść projektowanego układu.

W artykule zaproponowano następującą metodę projektowania układu U_3 :

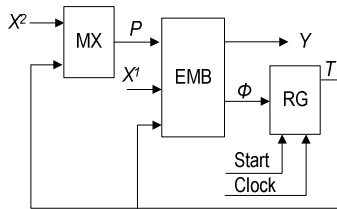
1. Utworzenie oznaczonej sieci działań Γ i znalezienie zbioru stanów A .
2. Zakodowanie stanów.
3. Utworzenie tabeli przejść dla układu Mealy FSM.
4. Utworzenie podziału zbioru X i znalezienie zbiorów X^1 i X^2 .
5. Utworzenie tabeli kodowania elementów podzbioru warunków logicznych.
6. Utworzenie przekształconej tabeli przejść dla projektowanego układu.
7. Implementacja układu w strukturach FPGA.

Strukturę projektowanego układu przedstawiono na rysunku 1. W układzie przedstawionym na rysunku 1, multiplexer MX generuje funkcje

$$P = P(X^2, T), \quad (1)$$

natomiast blok EMB implementuje system funkcji:

$$\begin{aligned}\Phi &= \Phi(P, X^1, T); \\ Y &= Y(P, X^1, T).\end{aligned}\quad (2)$$



Rys. 1. Struktura automatu Mealy U_3
Fig. 1. Structure diagram of Mealy U_3

2. Wstępne wyniki badań

W celu zbadania skuteczności proponowanych metod zastosowano zestaw standardowych testów z biblioteki LGSynth93 [21]. Zestaw ten obejmuje 44 moduły testowe zaczerpnięte z praktyki projektowania FSM.

Synteza została wykonana w oprogramowaniu firmy Xilinx dla układu Virtex-5 (XC5VLX30). Układ ten zawiera 32 bloki pamięci RAM (BRAM) o następującej konfiguracji: $36K \times 1$, $18K \times 2$, $9K \times 4$, $4K \times 9$, $2K \times 18$, $1K \times 36$, 512×72 . Na podstawie przeprowadzonych testów możemy stwierdzić, że dla 30 testów z biblioteki LGSynth93 odpowiednie automaty Mealy FSM mogą zostać zaimplementowane bez kodowania warunków logicznych. Oznacza to, że 68,2% modułów testowych może zostać zaimplementowana z wykorzystaniem układu z jednym blokiem pamięci EMB i rejestrem RG. Układ logiczny tych automatów wymaga tylko pojedynczego bloku BRAM.

Zastosowanie metody kodowania warunków logicznych konieczne jest, jeżeli chcemy zaimplementować automaty dla modułów testowych: *kirkman*, *s1a* i *s208*. Przedstawione w artykule rozważania pokazują, że zastosowanie układu U_3 zamiast układu z pracy [17] umożliwia zmniejszenie liczby tablic LUT w układzie multiplexera MX. Zmniejszenie to sięga do 23%. Zatem proponowana metoda poprawia właściwości sprzętowe multiplexera MX, jeżeli zastosowanie kodowania warunków logicznych jest wymagane.

Przeprowadzona analiza pokazuje, że układ U_3 nie może zostać zastosowany dla modułów testowych: *ex1*, *planet*, *planet1*, *s1488*, *s1494*, *s420*, *s510*, *s820*, *s832* i *sand* (11 testów).

W takim przypadku inne metody projektowania powinny zostać wykorzystane. Dla przykładu, metoda strukturalnej dekompozycji [16] może poprawić powierzchniowe parametry automatów. Drugim sposobem jest zastosowanie automatów hierarchicznych [22], które nadają się raczej do skomplikowanych algorytmów sterowania.

6. Podsumowanie

W artykule zaproponowano metodę umożliwiającą zmniejszenie zużycia zasobów sprzętowych w automatach Mealy'ego projektowanych z wykorzystaniem wbudowanego bloku pamięci EMB. Metoda ta bazuje na kodowaniu elementów tylko pewnego podzbioru warunków w zbiorze warunków logicznych. W rezultacie liczba tablic LUT zostaje zmniejszona w układzie logicznym multiplexera w porównaniu do przypadku, w którym kodowane są wszystkie warunki logiczne. Również liczba połączeń zostanie zmniejszona co skutkuje zmniejszeniem poboru mocy [10].

Dalsze kierunki prac autorów obejmują:

- opracowanie efektywnej metody podziału zbioru warunków logicznych; metoda ta powinna umożliwić zastosowanie układu U_3 z minimalnym rozmiarem multiplexera;
- opracowane narzędzi umożliwiających projektowanie zaproponowanych automatów;

- przekształcenie opracowanych metod dla przypadku automatu Moore'a.

7. Literatura

- [1] Baranov S. Logic Synthesis for Control Automata. – Boston: Kluwer Academic Publishers, 1994.
- [2] DeMicheli G. Synthesis and Optimization of Digital Circuits. – New York: McGraw – Hill, 1994.
- [3] Maxfield C. The Design Warrior's Guide to FPGAs. – Orlando: Academic Press, 2004
- [4] Grout I. Digital Systems Design with FPGAs and CPLDs. – Amsterdam: Elsevier, 2008.
- [5] www.altera.com
- [6] www.xilinx.com
- [7] Scholl C. Functional Decomposition with application to FPGA Synthesis. – Norwell: Kluwer Academic Publishers, 2001.
- [8] Kim T., Vella T., Brayton R., Sangiovanni-Vincentelli A. Synthesis of finite state machines: functional optimization. – Boston: Kluwer Academic Publishers, 1997.
- [9] Nowicka M., Łuba T., Rawski M. FPGA-based decomposition of Boolean functions: algorithms and implementation // Advanced Computer Systems, 1999. – pp. 502 – 509.
- [10] Sutteer G., Todorowich E., Lopez-Buedo S., Boemo E. Lower-power FSMs in FPGA: Encoding Alternatives. – Lecture Notes in Computer Science 2451. – Berlin: Springer, 2002. – pp. 363 – 370.
- [11] Wu X., Pedram M., Wang L. Multi-code state assignment for low-power design. – In: IEEE Proceedings on Circuits, Devices and Systems. – 2000, V. 147, № 5. pp. 271 – 275.
- [12] Cong J., Yan K. Synthesis for FPGAs with embedded memory blocks. – In: Proceedings of the 2000 ACM / SIGDA 8th International Symposium on FPGAs. – 2000, pp. 75 – 82.
- [13] Rawski M., Selvaraj H., Łuba T. An application of functional decomposition in ROM-based FSM implementation in FPGA devices // Journal of System Architecture. – 2005, V. 51, № 6 – 7. – pp. 424 – 434.
- [14] Rawski M., Tomaszewicz P., Borowski G., Łuba T. Logic Synthesis Method of Digital Circuits Designed for Implementation with Embedded Memory Blocks on FPGAs. – In: M. Adamski, A. Barkalov, M. Węgrzyn (Editors). Design of Digital Systems and Devices. LNEE 79. – Berlin: Springer, 2011. – pp. 121 – 144.
- [15] Barkalov A., Titarenko L. Logic Synthesis for FSM-Based Control Units. – Berlin: Springer, 2009.
- [16] Barkalov A. Multilevel programmable logic arrays schemes for microprogrammed automata. // Cybernetics and System Analysis. – 1994, V. 30, № 4. – pp. 489 – 496.
- [17] Barkalov A., Titarenko L., Barkalov A. Structural decomposition as a tool for the optimization of an FPGA-based implementation of a Mealy FSM. // Cybernetics and system analysis. – 2012, V. 48, №2. – pp. 313 – 323.
- [18] Sklyarov V. Synthesis and implementation of RAM-based finite state machines in FPGAs. – In: Proceedings of Conference on Field Programmable Logic. – Villach, 2000. – pp. 718 – 728.
- [19] Tiwari A., Tomko K. Saving power by mapping finite state machines into embedded memory blocks in FPGAs. – In: Proceedings of Design Automation and Test in Europe. – 2004, V. 2. – pp. 916 – 921.
- [20] Garcia-Vargas I., Senhadji-Navarro R., Civit-Balcells A., Guerra-Gutierrez P. ROM-based finite state machine implementation in low cost FPGAs. – In: IEEE International Symposium on Industrial Electronics. – Vigo, 2007. – pp. 2342 – 2347.
- [21] Kubatova H., Becvar M. FEL-Code: FSM Internal State Encoding Method. – In: Proceedings of 5th International Workshop on Boolean Problems. – 2002. – pp. 109 – 114.
- [22] Kubatova H. Finite State Machine Implementation in FPGAs. – Design of Embedded Control Systems. – New York: Springer, 2005. – pp. 175–184.
- [23] International Workshop on logic synthesis benchmark suite (LGSynth93): <http://www.lab13.kuee.kyotou.ac.jp/eda/benchmark/mcnc/benchmarks/LGSynth93/LGSynth93.tar>.