ISSN 1232-3829
tts
10/2013
Technika Transportu Szynowego koleje ▪ tramwaje ▪ metro

**PNIEWSKI Roman**

# METODY KODOWANIA AUTOMATÓW ASYNCHRONICZNYCH

***Streszczenie***

*Głównym problemem, ograniczającym wykorzystanie automatów asynchronicznych w syntezie systemów cyfrowych jest możliwość wystąpienia wyścigów (gonitw) krytycznych, powodujących niepoprawną pracę układu. Występowaniu tych zjawisk zapobiega się poprzez odpowiednie kodowanie stanów. W artykule zaprezentowano, stosowane powszechnie metody kodowania oraz, opracowana przez autora metodę opartą na warunkach elementarnych*

## WSTĘP

Kodowanie tablicy przejść polega na jednoznacznym przyporządkowaniu stanom wewnętrznym S automatu kodów binarnych: Q1, Q2, ..., Qk. Do zakodowania tablicy przejść automatu o K stanach wewnętrznych potrzeba k sygnałów 2-stanowych (binarnych), (2k-1 < k < 2k). Ilość bramek potrzebna do realizacji automatu zależy od sposobu przypisania symbolicznym stanom automatu kodów binarnych. Oczywiste jest, że dla różnych sposobów binarnej reprezentacji symboli otrzymamy różną realizację automatu. Dla różnych przypadków funkcje logiczne pobudzeń przerzutników i funkcje wyjścia wymagają różnych ilości bramek. Z praktycznego punktu widzenia jedyną możliwością znalezienia optymalnego rozwiązania jest zrealizowanie wszystkich możliwych sposobów kodowania. Bez komputerowych narzędzi wspomagających projektowanie możemy ograniczyć się tylko do stosowania pewnych heurystycznych zasad, których stosowanie prowadzi do uzyskania lepszych rezultatów.

W odróżnieniu od automatów synchronicznych kodowanie automatów asynchronicznych jest bardzo ważnym i krytycznym etapem syntezy, gdyż wybór niewłaściwego kodu może prowadzić do występowania zjawiska tzw. wyścigów, wynikającego z różnic opóźnień w elementach pamięciowych. Zjawisko wyścigu może wystąpić we wszystkich układach sekwencyjnych (synchronicznych i asynchronicznych). W układach synchronicznych wyścig może przyczyną chwilowych przekłamań stanu wyjść. Stan nieustalony, spowodowany zjawiskiem musi się zakończyć przed następnym impulsem zegara sterującego, dzięki temu nie wpływa na następny stan układu. Rozróżnia się dwa rodzaje wyścigu:

Niekrytyczny – jeżeli zmiany stanów spowodowane wyścigiem prowadzą, ostatecznie do zgodnego z programem stanu stabilnego

Krytyczny – co najmniej jedna z dróg prowadzi do niezgodnego z założeniami stanu stabilnego lub oscyluje pomiędzy dwoma stanami

Skutkami wyścigu krytycznego może być przejście do błędnego stanu stabilnego lub oscylacja miedzy stanem (niestabilnym) poprawnym i błędnym.

# 1. METODY KODOWANIA ZAPEWNIAJĄCE LIKWIDACJĘ WYŚCIGÓW

Do najpowszechniej stosowanych metod kodowania automatów asynchronicznych należą:
- Kodowanie „One Hot Encoding”
- Rachunek podziałów
- Przekształcanie grafu przejść

Kodowanie „One Hot Encoding”

W mało rozbudowanych systemach asynchronicznych rezygnuje się czasami z kodowania stanów na jak najmniejszej ilości przerzutników. Możemy dodać dodatkowe przerzutniki, aby dzięki temu uprościć logikę sterującą i wyjścia. Metodą kodowania która opiera się na takim podejściu jest kodowanie „one hot” (kodowanie z gorącą jedynką). Wykorzystuje ono dokładnie tyle przerzutników ile jest stanów automatu. W rzeczywistości, do poprawnego zakodowania automatu asynchronicznego wystarczy użyć N-1 przerzutników (jeden ze stanów kodujemy ciągiem 0).

Rachunek podziałów

Bezwyścigowe kodowanie automatów asynchronicznych opiera się na następujących własnościach:
- Jeżeli w kolumnie tablicy przejść odpowiadającej aktualnemu stanowi wejść występują dwa lub więcej stany stabilne to grozi wyścig krytyczny
- Jeżeli w kolumnie tablicy przejść odpowiadającej aktualnemu stanowi wejść jest tylko jeden stan stabilny, to może wystąpić tylko wyścig niekrytyczny
- Warunkiem wystarczającym uniknięcia wyścigów jest takie zakodowanie automatu, aby przy każdej zmianie stanu zmieniał się stan tylko jednego elementu pamięci

Przekształcanie grafu przejść

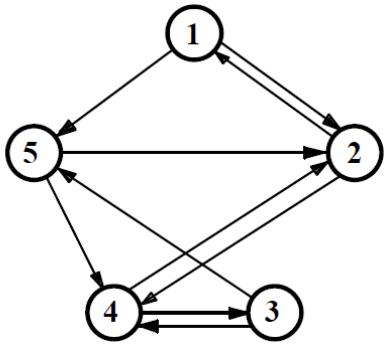
Metoda opiera się na zastosowaniu modelu geometrycznego przestrzeni stanów automatu. Model ten ma postać zbioru punktów odpowiadających poszczególnym stanom wewnętrznym, położonych w wierzchołkach k-wymiarowego hipersześcianu, którego sąsiednim wierzchołkom przyporządkowano kolejne liczby w kodzie Gray'a. W ten sposób punkty połączone krawędziami hipersześcianu przypisane mają kody różniące się zawsze tylko jednym bitem. W przypadku kodowania 2 stanów hipersześcianem jest odcinek łączący stany, dla 3 do 4 stanów (kod 2-bitowy) - kwadrat, dla 5 do 8 stanów (kod 3-bitowy) - sześcian. Dla kodowania wymagającego więcej niż 3 bitów metoda ta, stosowana bez pomocy komputera, zawodzi. Kodowanie polega na próbie takiego przypisania stanów automatu wierzchołkom hipersześcianu, aby wszystkie strzałki grafu przejść leżały na jego krawędziach. W większości przypadków nie daje się uniknąć przejść po przekątnych hipersześcianu, ale metoda pozwala nam przynajmniej zmniejszyć ich liczbę. Po każdej próbie zakodowania należy przeanalizować wszystkie przejścia po przekątnych, badając czy tworzą wyścigi krytyczne czy niekrytyczne. Jeśli istnieją wyścigi krytyczne, należy spróbować innego kodu, aż do wyczerpania możliwości. Jeśli nie istnieje możliwość uniknięcia wyścigów krytycznych, model geometryczny jest bardzo pomocny w projektowaniu przejść cyklicznych. Na rys.1.przedstawiono przykładową tablicę automatu oraz odpowiadający jej graf przejść. Rys.2.ilustruje różne warianty kodowania automatu.

(a)

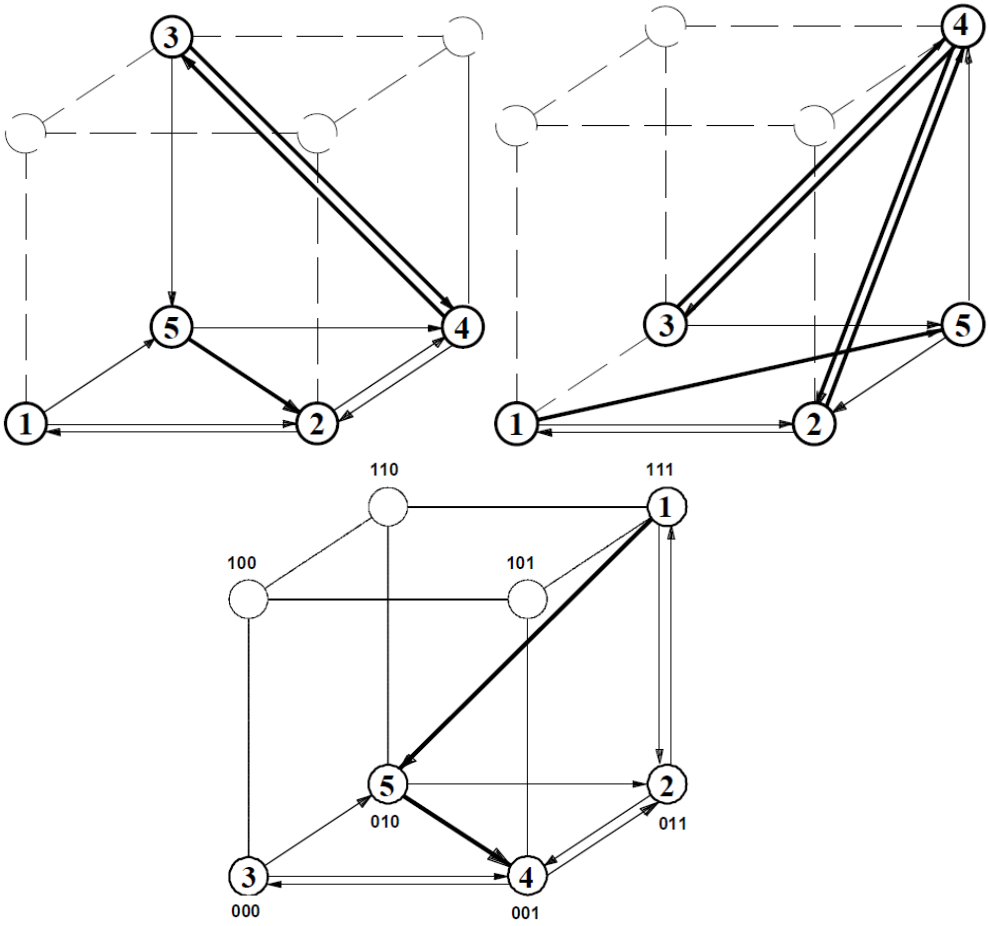
| $A\backslash X_1X_2$ | 00 | 01 | 11 | 10 | Y |
|---|---|---|---|---|---|
| 1 | ① | ① | 2 | 5 | 10 |
| 2 | 1 | ② | ② | 4 | 11 |
| 3 | 4 | 5 | ③ | ③ | -1 |
| 4 | ④ | 2 | 3 | ④ | 0- |
| 5 | 4 | ⑤ | 2 | ⑤ | 11 |

A'

(b)




**Rys.1** Przykład automatu sekwencyjnego [5]




**Rys.2.** Warianty kodowania automatu asynchronicznego [5]

## 2. METODA WARUNKÓW ELEMENTARNYCH

Prezentowana w tym rozdziale metoda wynika z rachunku podziałów, pozwala na pełną automatyzację procesu bezwyścigowego kodowania automatów asynchronicznych [2]. W metodzie tej są porównywane elementarne przejścia między stanami. Idea metody zostanie przedstawiona na prostym przykładzie automatu opisanego tabelą przejść Tab.1

W metodzie porównuje się (osobno dla każdego wektora wejściowego) przejścia między stanami mające różne następniki (inne stany docelowe w tabeli przejść). Przykładowo dla wektora wejściowego X1 w tabeli występują następujące przejścia:

1 -> 1 , 2 -> 2 , 3 -> 1 , 4 -> 2 .

**Tab1.**Przykładowa tabela przejść automatu

| | X1 | X2 | X3 | X4 |
|---|---|---|---|---|
| 1 | 1 | 4 | 1 | 2 |
| 2 | 2 | 2 | 1 | 2 |
| 3 | 1 | 2 | 3 | 4 |
| 4 | 2 | 4 | 3 | 4 |

Zgodnie z przyjętym założeniem porównujemy następujące przejścia:

- przejście 1 -> 1 z przejściem 2 -> 2
- przejście 1 -> 1 z przejściem 4 -> 2
- przejście 2 -> 2 z przejściem 3 -> 1
- przejście 4 -> 2 z przejściem 3 -> 1

W wyniku porównania tworzone są warunki (kody) w których stanom zawartym jednej parze przejść przypisana jest wartość „0” drugiej parze „1”, pozostałe stany otrzymują wartość „-”. W prezentowanym przykładzie dla poszczególnych przejść powstaną warunki:

[1234]
10--
10-0
010- (101-)
1010

Dla przejść, określonych przy wektorze wejściowym X2 powstaną warunki:

[1234]
10-1
1001
-1-0
-001

Dla wektora wejściowego X3:

[1234]
1-0-
1-00
110-
1100

Dla wektora wejściowego X4:

[1234]
1100
11-0
-100
-1-0

W pracy [3] pokazano metodę zoptymalizowanej minimalizacji warunków elementarnych. W pierwszym etapie minimalizacji można pominąć warunki podrzędne.

W czasie minimalizacji liczby warunków według proponowanej metody wykonuje się następujące operacje:

1. Z listy warunków elementarnych, utworzonych na podstawie tabeli przejść usuwane są wszystkie warunki podrzędne
2. Z pozostałych warunków tworzony jest graf niesprzeczności na którym zaznaczone są połączenia warunków niesprzecznych [3] i niesprzecznych po zanegowaniu [3]

W wyniku eliminacji warunków podrzędnych powstanie następujący zbiór warunków:
1010; 010-; 1001; 1100;

Minimalizacja sprowadza się do połączenia warunków niesprzecznych, przy czym połączenia dokonuje się dla warunków przedstawionych jak w powyższym przykładzie lub dla ich negacji (warunek 3 dla wejścia X1). Możliwość „sklejania" zanegowanych warunków wynika z dowolności przyjęcia „0" lub „1" dla stanów przydzielonych do wybranych przejść. W wyniku minimalizacji przedstawionych w przykładzie warunków otrzymuje się:

[1234]
1100
1001
1010

W automacie asynchronicznym, którego stany zostaną zakodowane według zminimalizowanych warunków nie wystąpią wyścigi krytyczne. W podanym przykładzie bezwyścigowe przejścia zostaną uzyskane przy przyjęciu następujących kodów dla poszczególnych stanów:

- Stan 1 – 111
- Stan 2 – 100
- Stan 3 – 001
- Stan 4 – 010

Jak wynika z przedstawionych kodów do poprawnej pracy automatu konieczne jest wykorzystanie 3 przerzutników. Choć z liczby stanów wynika, że do zakodowania wystarczą dwie zmienne kodowe to przy takim zakodowaniu wystąpią wyścigi krytyczne, konieczne jest więc zwiększenie liczby sprzężeń zwrotnych w układzie.

## PODSUMOWANIE

Przedstawiony w tym artykule, opracowany przez autora zmodyfikowany algorytm warunków elementarnych, zapewnia właściwe kodowanie stanów automatów asynchronicznych. Przedstawiona metoda pozwala na uniknięcie wyścigów krytycznych i może być stosowana w syntezie automatów asynchronicznych. Zaletą prezentowanej metody jest możliwość włączenia do systemów CAE przeznaczonych do projektowania systemów cyfrowych. Na podstawie prezentowanej metody został opracowany program do kodowania automatów. Prezentowane metody zostały ponadto wykorzystane w programach wsadowych, współpracujących z systemami LOGIC i WinCUPL. Programy te umożliwiają zakodowanie bezwyścigowe i syntezę automatów asynchronicznych, przy wykorzystaniu składni FSM stosowanej w tych programach. Powszechne zastosowanie automatycznego kodowania układów asynchronicznych (w systemach CAE) pozwoli na wyeliminowanie jednej z głównych wad tych układów.

## BIBLIOGRAFIA


1. Kawalec P. *Komputerowe wspomaganie projektowania cyfrowych urządzeń sterowania ruchem z wykorzystaniem grafów przejść automatu skończonego FSM*. Sprawozdanie z realizacji pracy własnej (grantu dziekańskiego). Wydział Transportu PW, Warszawa, 2001
2. Majewski W. *Cyfrowe układy telekomunikacyjne Podstawy teoretyczne i zasady syntezy* WKiŁ 1986.
3. Pniewski R. *Metoda oceny bezpieczeństwa cyfrowych systemów automatyki kolejowej*. Monografie Wyd. UTH Radom ISSN 1642-5278

4. Pniewski R., Strzyżakowski Z. *Symulacja trójwartościowa – wykrywanie hazardów w układach cyfrowych*, 10th International Conference "Computer Systems Aided Science, Industry and Transport", Transcomp 2006, vol.2, Zakopane 2006
5. Strona Internetowa „imio.pw.edu.pl”

# ASYNCHRONOUS AUTOMATA CODING METHOD

***Abstract***

*The main problem that limits the use of asynchronous automata for the synthesis of digital systems is the possibility of a race (race) critical, resulting in incorrect operation of the system. The occurrence of these phenomena is prevented by appropriate encoding states. The article presents, commonly used methods of coding and developed by the author's method based on the terms of elementary.*

***Autorzy****:*
dr inż. **Roman Pniewski** – Uniwersytet Technologiczno-Humanistyczny w Radomiu, Wydział Transportu i Elektrotechniki, e-mail: r.pniewski@uthrad.pl