

Halina KAMIONKA-MIKUŁA
Politechnika Śląska, Instytut Informatyki

KOMPUTEROWE WSPOMAGANIE NAUCZANIA W ZAKRESIE ZAGADNIEŃ WYŚCIGU W UKŁADACH CYFROWYCH

Streszczenie. W nauczaniu, zwłaszcza nauczaniu na odległość z wykorzystaniem sieci komputerowych, ważnym elementem są programy edukacyjne umożliwiające nie tylko prezentacje tematyki nauczania, ale również ćwiczenia i sprawdzanie wiedzy. W pracy przedstawiono rozwiązania zastosowane w jednym z programów dydaktycznych uwzględniających nauczanie zagadnień wyścigu w układach cyfrowych. Omówiono rozwiązania umożliwiające tworzenie nowych zadań w zakresie tematyki nauczania.

Słowa kluczowe: program edukacyjny, układ cyfrowy, zagadnienie wyścigu.

COMPUTER AIDED TEACHING OF RACE IN DIGITAL CIRCUITS

Summary. The educational programs are very important for teach process. They allow either presentation of the material parts or check student's knowledge. It is important especially for standalone teaching based on computer networks. This article presents the program showing race problems appearing in the asynchronous sequential logic circuits. The presented program allows make new exercises and control solution correctness also.

Keywords: educational program, logic circuits, race problems.

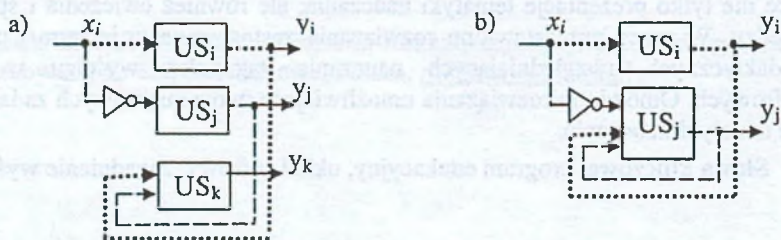
1. Wprowadzenie

W nauczaniu na odległość z wykorzystaniem sieci komputerowych kontakt osobisty ucznia z nauczycielem jest ograniczony. Dlatego w tym nauczaniu dużego znaczenia nabierają programy dydaktyczne. Powinny one nie tylko prezentować wiedzę teoretyczną, ale również umożliwiać wszechstronne ćwiczenia tematyki nauczania oraz sprawdzanie wiedzy. Ważnym walorem programu dydaktycznego jest: możliwość tworzenia z jego pomocą

nowych i możliwość modyfikacji wcześniej utworzonych danych dla przykładów i zadań. Podstawą rozwiązań umożliwiających „komputerową obsługę” nowych danych jest implementacja komputerowa ogólnych algorytmów dotyczących danej tematyki nauczania. Do tego potrzebny jest szczegółowy opis implementowanego algorytmu. W pracy omówiono rozwiązania zastosowane w jednym z programów wspomagających nauczanie z zakresie zagadnień wyścigu w układach cyfrowych.

2. Wyścig w układach cyfrowych

Wyścig jest jednym ze zjawisk występujących w układach sekwencyjnych z powodu opóźnień w elementach przełączających. Przyczyną jego jest różnica opóźnień dróg (rys.1) wiodących przez układy sekwencyjne (US_i oraz US_j), po których przenoszone są zmiany proste i zanegowane pewnego sygnału x_i do pewnego elementu.



Rys. 1. Schemat blokowy układu zawierającego wyścig

Fig. 1. Race containing circuits block diagram

W siatce programu wyścig objawia się tym, że istnieje bezpośrednie przejście między stanami niesąsiednimi logicznie. Rozróżnia się dwa rodzaje wyścigu: *niekrytyczny* - gdy wszystkie przejścia wynikające z wyścigu prowadzą do zgodnego z programem stanu stabilnego, *krytyczny* - gdy istnieje co najmniej jedna z dróg prowadząca do niezgodnego z programem stanu stabilnego lub do zamkniętego cyklu drgań. Złym skutkiem wyścigu można zapobiegać przez odpowiednie kodowanie stanów układu lub dobór opóźnień.

2.1. Algorytm wykrywania wyścigu

Algorytm wykrywania stanów, w których występuje wyścig jest prosty: dla każdego stanu układu sprawdza się, czy stanem docelowym jest stan niesąsiedni logicznie w stosunku do badanego stanu. Jeżeli tak, to w badanym stanie S_i występuje wyścig.

Algorytm rozpoznania rodzaju wyścigu

W algorytmie stosowane są następujące struktury pomocnicze:

ZW – lista stanów, w których występuje wyścig,

- SD – stan docelowy dla stanu S_i zgodnie z programem,
 T1 – lista stanów sąsiednich logicznie do stanu SB, do których układ może przejść zgodnie z programem lub w wyniku wyścigu,
 T2 – lista stanów, które wystąpiły na drodze badanego przejścia ze stanu S_i (gdzie: $S_i \in ZW$) do stanu SB.

Opis algorytmu rozpoznania rodzaju wyścigu w pseudokodzie

procedure ROZPOZNANIE_RODZAJU_WYŚCIGU

```

  for i = 1 to [liczba_elementów_listy_ZW] do
    Si:=ZW[i];
    SD:=rozpoznaj_docelowy(Si);
    T2:=czysta_lista;
    dodaj_element_do_T2(Si)
    if wyścig_krytyczny(Si,T2) then
      write („Wyścig krytyczny w” Si);
    else
      write („Wyścig niekrytyczny w” Si);
    end if
  end for
end

```

function wyścig_krytyczny(SB,T2_old)

{funkcja zwracające wartość boolowską}

```

  T1:=lista_przejsć(SB)
  k:=1;
  while k<=liczba_elementów_listyT1 do
    SN:=T1[k];
    T2:=T2_old;
    if (SN≠SD) then
      if SN jest stabilny then
        wyścig_krytyczny:=true;
        exit;
      else
        if (SN∈T2) then
          wyścig_krytyczny:=true;
          exit;
        else
          dodaj_element_do_T2(SN);
          (dodawany jest element SN na koniec listy T2)
          if wyścig_krytyczny(SN,T2) then
            wyścig_krytyczny:=true;
            exit;
          end if
        end if
      end if
    end while
    k:=k+1;
  wyścig_krytyczny:=false;
end

```

3. Opis rozwiązań zastosowanych w programie dydaktycznym wspomagającym nauczanie w zakresie zagadnień wyścigu

W bibliotece programów dydaktycznych Instytutu Informatyki Politechniki Śląskiej jest kilka programów, np.: [3], [6], uwzględniających zagadnienie wyścigu. W każdym z tych programów są zastosowane różniące się rozwiązania uwzględniające zagadnienie wyścigu. Dalej przedstawiono rozwiązania zastosowane w programie *Wyścig w układach cyfrowych*.

Program *Wyścig w układach cyfrowych* [3] jest 32-bitową aplikacją pracującą pod kontrolą systemu operacyjnego Windows 95/98/2000/XP. Do stworzenia aplikacji został wykorzystany pakiet Microsoft Visual Studio 6.0. Program *Wyścig w układach cyfrowych* został napisany w języku Visual C++ przy użyciu technik programowania obiektowego OOP (Object Oriented Programming). Jest to aplikacja MDI zaprojektowana przy użyciu struktury Document/View składającej się z jednego dokumentu i jednego widoku.

Aplikacja umożliwia: prezentację przebiegu analizy pod względem wyścigu dla układu opisanego binarną siatką przejść, realizację ćwiczeń i sprawdzanie wiedzy w zakresie tej tematyki, a także tworzenie nowych zadań i nowych prezentacji.

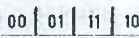
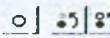
W programie zaimplementowano ogólny algorytm analizy układu pod względem wyścigu oparty na opisie podanym w rozdziale 2. Algorytm ten jest podstawą do tworzenia nowych prezentacji i do kontroli poprawności rozwiązywania zadań podczas ćwiczeń i w opcji sprawdzania wiedzy (Test).

Program udostępnia wygodny dla użytkownika interfejs ułatwiający tworzenie nowych zadań i wprowadzanie odpowiedzi podczas rozwiązywania zadań.

Dostępne paski narzędzi, dostosowane do trybu pracy i etapu zadania, ułatwiają między innymi: wypełnianie siatki przejść, zaznaczanie stanów stabilnych i niestabilnych, zaznaczanie przejść wynikających z programu i przejść wynikających z wyścigu oraz zaznaczanie rodzaju wyścigu.

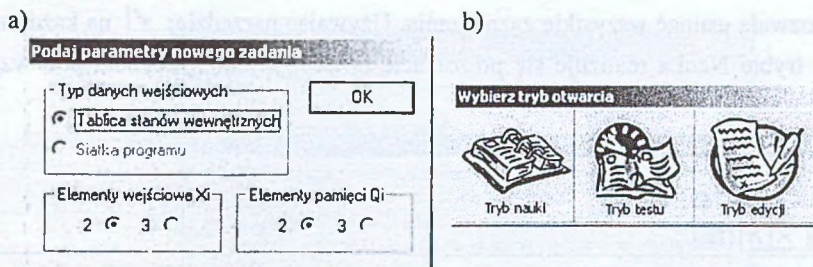
Oprócz opisu teoretycznego zagadnienia wyścigu program udostępnia pracę w trzech trybach: Edycja, Nauka i Test.

W trybie Edycja tworzona jest siatka opisująca badany układ. Podczas wypełniania siatki widoczne są dwie siatki: binarna siatka stanów i siatka przejść (w niej przejścia zaznaczone są strzałkami). Wypełnianie siatki może się odbywać dwoma sposobami:

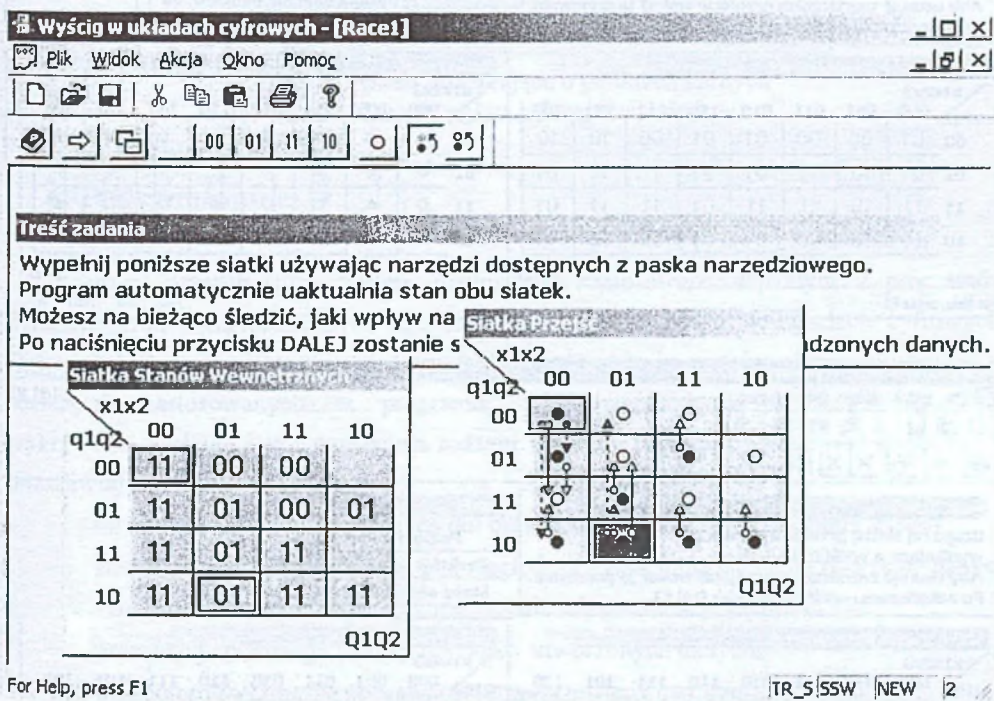
- używając narzędzi:  i „klikając” na pola siatki binarnej lub siatki przejść,
- z użyciem elementów:  i „klikając” na pola dowolnej z tych siatek.

Efekty wypełnienia siatki widoczne są jednocześnie w obydwu siatkach, niezależnie od tego, na pola której z nich „klikano”.

Rysunki 2 do 5 ilustrują pracę z programem w trybach: Edycja i Test.



Rys. 2. a) Okno określania parametrów zadania, b) okno wyboru trybu
 Fig. 2. a) Task parameters window b) mode selection window



Rys. 3. Okno programu w trybie Edycja
 Fig. 3. "Edit mode" window

Rozwiązywanie zadań obejmuje etapy: zaznaczanie stanów, w których występuje wyścig, zaznaczanie stanów stabilnych i niestabilnych, zaznaczanie przejść wynikających z programu i przejść wynikających z wyścigu, określenie rodzaju wyścigu.

W trybie Nauka uczeń może próbować rozwiązywać zadanie samodzielnie lub skorzystać z podpowiedzi z użyciem narzędzia: pokazującego dany etap rozwiązania. W przypadku błędnych odpowiedzi uczeń może też skorzystać z innej podpowiedzi, mianowicie użycie pierwszego z narzędzi: powoduje usunięcie tylko błędnych zaznaczeń. Drugie z tych

narzędzi pozwala usunąć wszystkie zaznaczenia. Używając narzędzia: ✓ na każdym etapie zadania w trybie Nauka realizuje się prezentację przebiegu analizy układu pod względem wyścigu.

Wyścig w układach cyfrowych - [z2.rce] _ [O] X
_ [E] X

plik Widok Akcja Okno Pomoc

Treść zadania

W poszczególnych kolumnach siatki przejść zaznacz przejścia ze stanów niestabilnych do odpowiadającym im stanów stabilnych wynikające z PROGRAMU.
Aby usunąć zaznaczone przejście wskaż je ponownie.
Po zakończeniu naciśnij przycisk DALEJ.

Wynik:

Poziom trudności: 2
Punktów do zdobycia: 140
Masz obecnie punktów: 40

Siatka Stanów Wewnętrznych

x1x2x3	000	001	011	010	110	111	101	100	
q1q2	00	01	00	00	01	01	00	10	10
	01	01	10	11	01	01	11	11	01
	11	11	10	11	11	01	11	11	01
	10	11	10	00	01	01	00	10	10

Q1Q2

Siatka Przejść

x1x2x3	000	001	011	010	110	111	101	100	
q1q2	00	01	00	00	01	01	00	10	10
	01	01	10	11	01	01	11	11	01
	11	11	10	11	11	01	11	11	01
	10	11	10	00	01	01	00	10	10

Q1Q2

For Help, press F1 PGM SSW TEST 4

Wyścig w układach cyfrowych - [z2.rce] _ [O] X
_ [E] X

plik Widok Akcja Okno Pomoc

Treść zadania

Uzupełnij siatkę przejść o przejścia wynikające z WYŚCIGU.
Aby usunąć zaznaczone przejście wskaż je ponownie.
Po zakończeniu naciśnij przycisk DALEJ.

Wynik:

Poziom trudności: 2
Punktów do zdobycia: 140
Masz obecnie punktów: 60

Siatka Stanów Wewnętrznych

x1x2x3	000	001	011	010	110	111	101	100	
q1q2	00	01	00	00	01	01	00	10	10
	01	01	10	11	01	01	11	11	01
	11	11	10	11	11	01	11	11	01
	10	11	10	00	01	01	00	10	10

Q1Q2

Siatka Przejść

x1x2x3	000	001	011	010	110	111	101	100	
q1q2	00	01	00	00	01	01	00	10	10
	01	01	10	11	01	01	11	11	01
	11	11	10	11	11	01	11	11	01
	10	11	10	00	01	01	00	10	10

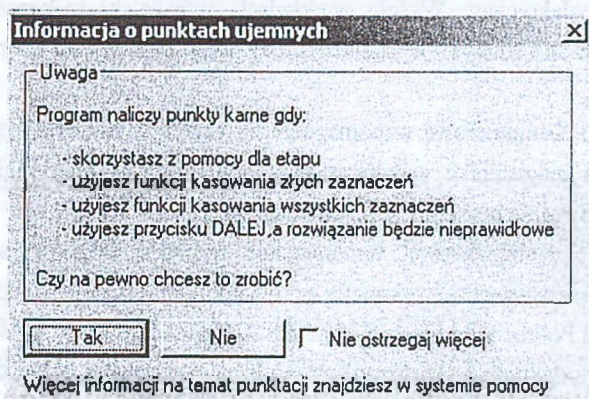
Q1Q2

For Help, press F1 WSC SSW TEST 5

Rys. 4. Okna programu w trybie Test- zaznaczanie przejść

Fig. 4. Transition selecting in "Test mode" windows

Praca z programem w trybie Test podlega punktowej ocenie. Naliczane są punkty: dodatnie za poprawne odpowiedzi i ujemne w przypadkach widocznych na rys.5.



Rys. 5. Okno informujące o punktach karnych

Fig. 5. Penalty points information window

4. Podsumowanie

W pracy przedstawione zostały rozwiązania zastosowane w jednym z programów dydaktycznych uwzględniających nauczanie zagadnień wyścigu w układach cyfrowych. Opisano algorytm wykrywania wyścigu i określania rodzaju wyścigu, który był podstawą rozwiązań zastosowanych w programie. Zaimplementowanie ogólnego algorytmu wykrywania wyścigu i rozpoznawania rodzaju wyścigu pozwoliło opracować program [3] umożliwiającą:

- łatwe tworzenie nowych danych do: prezentacji, ćwiczeń i sprawdzania wiedzy,
- śledzenie właściwości układu pod względem wyścigu już na etapie tworzenia danych - dzięki temu widać, czy tworzone zadanie lub prezentacja będą ciekawe,
- prezentację przebiegu analizy dla nowo utworzonych danych,
- komputerową kontrolę poprawności rozwiązywania utworzonych zadań.

Wygodny interfejs ułatwia pracę z programem.

LITERATURA

1. Kamionka-Mikuła H., Małysiak H., Pochopień B.: Układy cyfrowe – teoria i przykłady. Wydanie IV zmienione, PJS, Gliwice 2002.
2. Papała M.: Komputerowe wspomaganie nauczania w zakresie zagadnień wyścigu. Praca dyplomowa magisterska realizowana pod kierunkiem H. Kamionki-Mikuły w Instytucie Informatyki Politechniki Śląskiej, Gliwice 2002.

3. Program dydaktyczny: Wyścig w układach cyfrowych, wykonany przez: M. Papalę, w Instytucie Informatyki w ramach pracy dyplomowej pod kierunkiem H. Kamionki-Mikuły.
4. Trznadel J.: Komputerowe wspomaganie nauczania w zakresie metody Huffmana. Praca dyplomowa magisterska wykonana pod kierunkiem H. Kamionki-Mikuły w Instytucie Informatyki Politechniki Śląskiej, Gliwice 2001.
5. Samol M.: Komputerowe wspomaganie syntezy układów sekwencyjnych. Praca dyplomowa magisterska wykonana pod kierunkiem H. Kamionki-Mikuły w Instytucie Informatyki Politechniki Śląskiej, Gliwice 1998.
6. Programy dydaktyczne: HuffmanLab, Huffman wykonane przez: J. Trznadla, M. Samol w Instytucie Informatyki w ramach prac dyplomowej pod kierunkiem H. Kamionki-Mikuły.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 15 kwietnia 2003 r.

Abstract

The educational programs are very important for teach process. They allow either presentation of the material parts or check student's knowledge. It is important especially for standalone teaching based on computer networks.

This article presents the program showing race problems appearing in the asynchronous sequential logic circuits.

The algorithm implemented in the "Race in the logic circuits" program [2], [3] allows either detection of the race or recognition type of race. The presented program allows make new exercises and control solution correctness also.

Presented figures (2-5) illustrate program interface. Presented toolbars depends on mode and stage in the selected mode.

Adres

Halina KAMIONAKA-MIKUŁA: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, hkm@polsl.gliwice.pl