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

BOCIAN Stanisław

MIKROSYSTEMY CYFROWE STOSOWANE W POJAZDACH SZYNOWYCH I ZŁOŻONOŚĆ OBLICZENIOWA PÓŁGRUPY CHARAKTERYSTYCZNEJ AUTOMATU ASYNCHRONICZNEGO SPÓJNEGO

Streszczenie

W artykule omówiono mikrosystemy cyfrowe stosowane w pojazdach szynowych. Umożliwiają one programowanie w oparciu o zaimplementowane bloki funkcjonalne wejść i wyjść oraz funkcje przejścia (maszyna stanowa – automat) w oparciu o sporządzony wcześniej graf automatu. Przedstawiono także twierdzenie i przeprowadzono dowód na złożoność obliczeniową półgrupy charakterystycznej automatu asynchronicznego spójnego.

WSTĘP

Rozwój technologii obwodów scalonych umożliwiają tworzenie programu w oparciu o sporządzony wcześniej graf maszyny stanowej (automatu). Teoria automatów, stosowana do lat dziewięćdziesiątych ubiegłego stulecia poprzez rozwój technologiczny układów scalonych, po wielu latach ponownie może być wykorzystywana do projektowania złożonych systemów technicznych i informatycznych. Modele sterowania podsystemów mechatronicznych pojazdu szynowego możemy przedstawić w postaci grafu automatu.

Wzrost funkcjonalności złożonych podzespołów cyfrowych przyczyniły się do znacznego zainteresowania złożonością obliczeniową dla wielu zagadnień. Złożoność obliczeniowa ma znaczenie ogólne, to znaczy praktyczne, przy projektowaniu algorytmów jak i teoretyczne, kiedy chcemy zrozumieć istotę złożoności obliczeniowej.

Automat jest abstrakcyjnym pojęciem matematycznym, modelem obliczeń lub modelem pewnego procesu. Algebraiczna teoria automatów z jednej strony jest teoretycznym uogólnieniem teorii układów logicznych, z drugiej strony może być traktowana jako dział algebry. Pojęcia z algebry w postaci sformalizowanej są analizowane i przekształcane do postaci dogodnej do optymalizacji. Z postaci abstrakcyjnej, w procesie syntezy można je przekształcić w schemat logiczny, w oprogramowanie lub ich kombinację. Tym samym uczy teoria automatów jak koncepcyjnie i obliczeniowo rozważać wzrastające co do wielkości i złożoności problemy w informatyce.

Półgrupa charakterystyczna jest szczególnie istotnym pojęciem w teorii automatów; jest ona nośnikiem ważnych informacji, tak w aspekcie strukturalnym jak i lingwistycznym. Ma więc bezpośrednio ważkie konsekwencje praktyczne w algebrze projektowania optymalnych układów logicznych.

Szczególne rozpatrywanie klas automatów asynchronicznych wynika z powszechnego stosowania tych klas automatów w realizacjach technicznych cyfrowych układów sterujących.

Można by przypuszczać, że olbrzymi postęp w szybkości obliczeń komputerów zmniejszy znaczenie efektywności algorytmów. Okazuje się, że komputery liczą coraz szybciej i mają coraz większą pamięć. W dalszym ciągu o wzroście rozmiaru zadania, które może być osiągnięte wraz ze wzrostem szybkości komputera, decyduje przede wszystkim złożoność algorytmu.

1. MIKROSYSTEMY CYFROWE STOSOWANE W POJAZDACH SZYNOWYCH

Ciągły rozwój transportu kolejowego na świecie charakteryzuje się m.in. wzrostem prędkości i niezbędną z tego powodu automatyzacją procesów sterujących ruchem pociągów. Dotyczyć to będzie także systemu i różnych podsystemów sterowania i automatyki stosowanych w pojazdach szynowych. Układ sterowania nowoczesnym pociągiem zawiera wiele układów automatycznego sterowania, które oddziałują na siebie lub są we wzajemnej zależności, tworząc złożony system automatycznego sterowania pojazdem.

Rozwój technologii układów scalonych umożliwia realizacje tych zagadnień wykorzystując mikrosystemy cyfrowe (ATMEL, CYPRESS, TRISCEND i inne).

Mikrosystemy cyfrowe PSoC CY29466 firmy CYPRESS umożliwia tworzenie programu w oparciu o sporządzony wcześniej graf maszyny stanowej (automatu).

Układ scalony nazywa się mikrosystemem cyfrowy ponieważ w swej strukturze integruje rdzeń mikroprocesorowy oraz programowalny blok sprzętowy.

W procesie projektowania cyfrowych układów sterowania oprócz stosowania modeli specyfikacji formalnej bardzo ważne jest sprecyzowanie docelowej platformy realizacyjnej takiej jak: sprzętowej, programowej, i sprzętowo – programowej. Platforma sprzętowa to struktury układowe małej i średniej skali integracji oraz nowoczesne matryce reprogramowalne. Realizacja programowa to połączenie pewnego zestawu instrukcji (program) oraz odpowiednich struktur sprzętowych (np. mikroprocesor, pamięć) zdolnych do wykonywania określonych działań zapisanych w kodzie tego programu. Zalety i wady obu rozwiązań, tzn. realizacji sprzętowej oraz programowej, rozpatruje się pod względem dwóch podstawowych kryteriów:

1. czasu reakcji – szybsze rozwiązanie sprzętowe
2. kosztów prototypowania – przyjmuje się niższe dla rozwiązań programowych, ze względu na niższą cenę zarówno samych układów, jak i narzędzi wspomagających proces projektowania.

Własności te wydają się wzajemnie sprzeczne, dlatego należałoby zastosować jednocześnie obie metody stosując kompromis łączący zalety szybkości działania z niskimi kosztami produkcji. Funkcje takie spełniają zintegrowane w jednej strukturze scalonej mikrosystemy cyfrowe. Przykładem takiego systemu są systemy cyfrowe PSoC (Programmable System on Chip) firmy CYPRESS [1,2,3,4]. Układ ten różni się od typowych mikrokontrolerów tym, że posiada programowalne peryferia nie tylko cyfrowe, ale także analogowe. Należy wybrać tę rodzinę układów ze względu na bogate możliwości programowego tworzenia układów analogowych z rekonfigurowanych bloków. Ponadto dostępne są układy cyfrowe: liczniki, timery, generatory pseudolosowe itp. Komunikację zapewniają moduły RS–232, I^2C, USB.

Tworzenie oprogramowania jest możliwe za pomocą dwóch narzędzi:

1. PSoC Designer

– Umożliwia tworzenie oprogramowania z wykorzystaniem zaimplementowanych bloków funkcjonalnych uzupełnionych o programy w Asemmlerze (i/lub) języku C.

2. PSoC Express

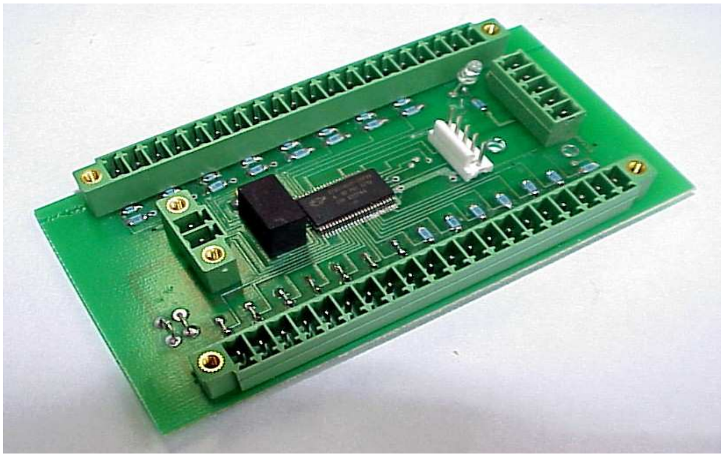
– Przeznaczony jest do tworzenia programu w oparciu o zaimplementowane bloki funkcjonalne we/wy oraz funkcje przejścia. Funkcje przejścia mogą być:
  – Kombinacyjne (Table Lookup)
  – Warunkowe (warunki w języku C) :

Status Encoder If x1 then y1
If x2 then y2
If x3 then y3

Priority Encoder If x1 then y1
Else If x2 then y2
Else If x3 then y3

– Stanowe (State Machine) – umożliwiają tworzenie programu w oparciu o sporządzony wcześniej graf automatu.

W Zakładzie Elektrotechniki Instytutu Pojazdów Szynowych „TABOR” w Poznaniu zaprojektowano i wykonano oprogramowanie modułu akwizycji sygnałów analogowych i cyfrowych oraz sterowania wyjść cyfrowych 24 V [1]. Na rys. 1 przedstawiono układ wyposażony w 8 wejść analogowych , 7 wejść cyfrowych oraz 7 wyjść cyfrowych. Przykładowy program z wykorzystaniem tego modułu realizuje pomiar częstotliwości w 4 kanałach oraz zbiera dane cyfrowe z 6 wejść cyfrowych. Następnie całą zebraną informację przesyła do sterownika nadrzędnego za pośrednictwem łącza RS – 232.



Rys. 1 . Widok przykładowego modułu mikroprocesorowego [1]

Na pomiar częstotliwości składają się następujące czynności:

– wzmocnienie wejściowego sygnału sinusoidalnego
– kształtowanie sygnału prostokątnego
– filtrowanie ew. wyższych częstotliwości
– realizacja programu do cyfrowego pomiaru częstotliwości.

Część analogową pomiaru zrealizowano z wykorzystaniem zasobów analogowych mikrokontrolera PSoC.

W pracy [2] przedstawiono układ przetwarzania sygnałów analogowych z czujników reluktancyjnych i program do pomiaru prędkości w pojazdach szynowych.

W pracy [3] przedstawiono zastosowanie maszyny stanowej do projektowania układów wejściowych wykorzystując mikrosystem cyfrowy firmy CYPRESS.

Mikrosystemy cyfrowe PSoC CY29466 firmy CYPRESS stosowane są obecnie do sterowania w pojazdach szynowych w lokomotywach i jednostkach elektrycznych.

2. ZŁOŻONOŚĆ OBLICZENIOWA PÓŁGRUPY CHARAKTERYSTYCZNEJ AUTOMATU ASYNCHRONICZNEGO SPÓJNEGO

Mikrosystem cyfrowy ma w swej strukturze także programowalny blok (maszyna stanowa – automat). Do tego celu wprowadzamy pojęcia algebry abstrakcyjnej i teorii automatów [5,6]

Grupoidem nazywamy parę uporządkowaną $(S,\circ)$ gdzie: S – niepusty zbiór, $(\circ)$ – operacja binarna na zbiorze stanów S. Operacją binarną na zbiorze S nazywamy przekształcenie niepustego podzbioru zbioru $(S \times S)$ w zbiór S. Binarną operację $(\circ)$ na zbiorze S nazywamy łączną (asocjatywną), jeśli $a \circ (b \circ c) = (a \circ b \circ c)$ dla wszystkich $a, b, c \in S$. Półgrupą, nazywamy taki grupoid $(S,\circ)$, w którym operacja $(\circ)$ jest asocjatywna. Niech Σ będzie dowolnym zbiorem niepustym. Zbiór Σ będziemy nazywali alfabetem, a jego elementy literami. Słowem x w alfabecie Σ nazywamy dowolny ciąg liter alfabetu napisanych obok siebie, a długością słowa (oznaczoną przez $|x|$) nazywamy liczbę tych liter σ.

Skończonym automatem zdeterminowanym bez wyjść nazywamy uporządkowaną trójkę (S, Σ, M), gdzie:

S – skończony, niepusty zbiór stanów
Σ – skończony, niepusty zbiór wejść
$M : S \times \Sigma \to S$: jest funkcją przejść.

Symbolem Σ^* oznaczać będziemy przeliczalny nieskończony zbiór ciągów o skończonej długości, utworzony z elementów zbioru Σ. Zbiór Σ^* razem z operacją konkatenacji (operacja połączenia dwóch słów, polegającą na napisaniu ich obok siebie w celu otrzymania nowego słowa), tworzy półgrupę wolną zwaną półgrupą wejściową. Symbolem Σ^* oznaczać będziemy monoid wejściowy, czyli $\Sigma^* = \Sigma^+ \cup \lambda \quad gdzie \quad \lambda$, jest ciągiem pustym.

Funkcję M rozszerzamy do obszaru określoności $S \times \Sigma^+$ w podany poniżej sposób. Niech: $M(s, x)$ będzie zdefiniowane, wtedy:

$$M(s, x\sigma) = M(M(s, x,), \sigma) \quad dla \quad każdego \quad s \in S, \quad x \in \Sigma^+, \quad \sigma \in \Sigma .$$

Na zbiorze Σ^* zdefiniujemy relację:
xRy wtedy i tylko wtedy, gdy $\forall_{s \in S} M(s, x) = M(s, y)$.

R jest relacją równoważności (relacja Myhilla). Definicje relacji równoważności Myhilla na zbiorze stanów automatu oraz półgrupy charakterystycznej automatu pozwoliły wydobyć zeń możliwości obliczeniowe. Klasę równoważności zawierającą element $x \in \Sigma^*$ oznaczać będziemy $\overline{x}$, a zbiór wszystkich klas równoważności oznaczać będziemy $\overline{I}$. Zbiór $\overline{I}$ łącznie z operacją $(\circ)$ $gdzie \quad \overline{x} \circ \overline{y} = \overline{xy}$, tworzy półgrupę (odpowiednio monoid), zwaną półgrupą charakterystyczną (odpowiednio monoidem charakterystycznym). Półgrupę charakterystyczną automatu A oznaczać będziemy $\overline{I(A)}$.

Dla każdego $x \in \Sigma^*$ definiujemy przekształcenie f_x zbioru S w siebie, gdzie : $f_x(s) = M(s, x)$, dla każdego $s \in S$. Przekształcenie f_x jest implikowane przez x. Zbiór przekształceń zbioru S w siebie, implikowanych przez wszystkie elementy z Σ, będziemy oznaczać symbolem J. J ze względu na operację superpozycji, jest zbiorem generatorów pewnej półgrupy.

Automat $A=(S,\Sigma,M)$ będziemy nazywać asynchronicznym wtedy i tylko wtedy gdy, dla każdego $s\in S \;\; i \;\; \sigma\in\Sigma \;\; zachodzi \;\; M(s,\sigma)=M(s,\sigma\sigma)$.

Automat $A=(S,\Sigma,M)$ jest spójny, jeśli dla każdej pary $(s,s')\in S\times S$ istnieją słowa $x,y\in\Sigma^{+}$ takie, że $M(s,x)=s'$ lub $M(s',y)=s$.

Pojęcia: automat skończony, maszyna stanów (maszyna stanowa), możemy ogólnie traktować jako równoważne. Angielskie Finite State Machine (FMS), lub Finite State Automaton (FSA w liczbie mnogiej: automata) często tłumaczone są zamiennie jako automat skończony, lub maszyna stanowa.

Formalnie FSM nie pokrywa wszystkich możliwych wariantów FSA, który jest pojęciem szerszym. W informatyce teoretycznej problemy obliczalności rozwiązuje się z wykorzystaniem automatów. Możemy modelować prawdziwe „maszyny” korzystając z matematycznego modelu automatów.

Automat deterministyczny skończony asynchroniczny spójny (deterministic finite asynchronous connected) dla dwuliterowego alfabetu wejściowego $\Sigma=\{\sigma_0,\sigma_1\}$ oznaczamy DFAC$_2$. Dla tej klasy automatu dla słów $x_0=\sigma_0\sigma_1$; $x_1=\sigma_1\sigma_2$ wyznaczamy złożoność półgrupy charakterystycznej.

Dla tej klasy automatu przeprowadzam dowód:

Twierdzenie 1.

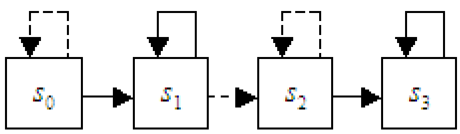
Niech A = (S, Σ, M) będzie automatem z klasy DFAC$_2$; wtedy półgrupa charakterystyczna
$\overline{I(A)}$ automatu A ma własność:

$$card\,\overline{I(A)}=2n-1 \tag{1}$$

gdzie:

$n=card(S)>2$; $card(\Sigma)=2$; $x_0=\sigma_0\sigma_1$; $x_1=\sigma_1\sigma_1$; $\sigma_0 -$ $\sigma_1 --$.

Dowód:



Rys. 2. Automat z klasy $DFAC_2$ dla czterech stanów

Dowód przeprowadzamy korzystając z zasady indukcji.

1^0

Dla automatu z klasy $DFAC_2$ (rys. 2) w pierwszym kroku indukcyjnym dla $n=card(S)=4$ obliczamy złożoność (moc) półgrupy charakterystycznej $\overline{I(A)}$ automatu A.

Rozpoczynając przekształcenie od liter σ_0 otrzymujemy następujące przekształcenia:

$f_{\sigma_0}=(s_1s_1s_3s_3)$

$f_{x_0} = (s_2 s_2 s_3 s_3)$

$f_{x_0\sigma_0} = (s_3 s_3 s_3 s_3)$.

Na tym kończymy wyznaczanie półgrupy charakterystycznej automatu dla słów rozpoczynających się od litery σ_0. W następnych krokach uzyskujemy przekształcenia, które już poprzednio zostały wygenerowane.
Rozpoczynając przekształcenia od litery σ_1 (rys. 2) otrzymujemy następujące przekształcenia:

$f_{\sigma_1} = (s_0 s_2 s_2 s_3)$

$f_{x_1} = (s_1 s_3 s_3 s_3)$

$f_{x_1\sigma_1} = (s_2 s_3 s_3 s_3)$

$f_{x_1^2} = (s_3 s_3 s_3 s_3)$ gdzie $x_1^2 = \sigma_1\sigma_2\sigma_1\sigma_2$.

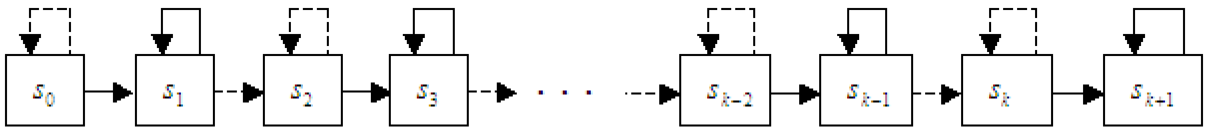
Na tym kończy się wyznaczanie półgrupy charakterystycznej automatu dla słów rozpoczynających się od litery σ_1. W następnych krokach uzyskujemy przekształcenia, które już poprzednio zostały wygenerowane. Sumując liczbę powyższych przekształceń dla automatu przedstawionego na rys. 1 otrzymujemy 7 różnych przekształceń.

2^0

Z założenia indukcyjnego mamy dla $n = card(S) = k$
$card\ \overline{I(A)} = 2k - 1$.

3^0

W kroku indukcyjnym otrzymujemy dla $n = k + 2$
$card\ \overline{I(A)} = 2(k + 2) - 1$.



Rys. 3. Automat DFAC$_2$ o $card(S) = k + 2$ stanów

Na rys. 3 przedstawiony został automat z klasy DFAC$_2$ dla którego obliczamy moc jego półgrupy charakterystycznej.

Rozpoczynając przekształcenia od litery $\sigma_0 \in \Sigma$ otrzymujemy:

$f_{\sigma_0} = (s_1 s_1 s_3 s_3, \ldots, s_{k-1} s_{k-1} s_{k+1} s_{k+1})$

$f_{x_0} = (s_2 s_2 s_4 s_4, .., s_k s_k s_{k+1} s_{k+1})$

$f_{x_0\sigma_0} = (s_3 s_3 s_5 s_5, \ldots, s_{k+1} s_{k+1} s_{k+1} s_{k+1})$

$f_{x_0^2} = (s_4 s_4 s_6 s_6, \ldots, s_{k+1} s_{k+1} s_{k+1} s_{k+1})$

.

.

.

$$f_{x_0^{\left(\frac{k}{2}\right)}} = \left(s_k s_k s_{k+1} s_{k+1}, \ldots, s_{k+1} s_{k+1} s_{k+1} s_{k+1}\right)$$

$$f_{x_0^{\left(\frac{k}{2}\right)}\sigma_0} = \left(s_{k+1} s_{k+1} s_{k+1} s_{k+1}, \ldots, s_{k+1} s_{k+1} s_{k+1} s_{k+1}\right).$$

Na tym kończy się wyznaczanie półgrupy charakterystycznej automatu DFAC$_2$ dla słów rozpoczynających się od litery σ_0

Rozpoczynając przekształcenia od litery $\sigma_1 \in \Sigma$ otrzymujemy:

$$f_{\sigma_1} = (s_0 s_2 s_2 s_4, \ldots, s_{k-2} s_k s_k s_{k+1})$$

$$f_{x_1} = (s_1 s_3 s_3 s_5, \ldots, s_{k-1} s_{k+1} s_{k+1} s_{k+1})$$

$$f_{x_1\sigma_1} = (s_2 s_4 s_4 s_6, \ldots, s_k s_{k+1} s_{k+1} s_{k+1})$$

.

.

.

$$f_{x_1^{\frac{k}{2}}} = (s_{k-1} s_{k+1} s_{k+1} s_{k+1}, \ldots, s_k s_{k+1} s_{k+1} s_{k+1})$$

$$f_{x_1^{\frac{k}{2}}\sigma_1} = (s_k s_{k+1} s_{k+1} s_{k+1}, \ldots, s_k s_{k+1} s_{k+1} s_{k+1})$$

$$f_{x_1^{\frac{k}{2}+1}} = (s_{k+1} s_{k+1} s_{k+1} s_{k+1}, \ldots, s_{k+1} s_{k+1} s_{k+1} s_{k+1}).$$

Na tym kończy się wyznaczanie półgrupy charakterystycznej automatu DFAC$_2$ dla słów rozpoczynających się od litery σ_1.

Zliczając ilość powyższych przekształceń otrzymujemy

$card(I(A)) = 2(k+2) - 1$.

Udowodniliśmy tym samym, że moc półgrupy charakterystycznej automatu A z klasy $DFAC_2$ wynosi: $card(I(A)) = 2n - 1$.

PODSUMOWANIE

Definicje relacji równoważności Myhilla na zbiorze stanów automatu oraz półgrupy charakterystycznej automatu pozwoliły wydobyć zeń możliwości obliczeniowe . Tak więc na automat spojrzeć można zarówno pod kątem widzenia systemu algebraicznego o charakterze strukturalno – językowym, jak i modelu obliczeń.

Półgrupa charakterystyczna określa zdolność do przetwarzania informacji.

Wykorzystując teorię automatów możemy oszacować lub obliczyć złożoność półgrup charakterystycznych automatów. Ma to istotny wpływ na oszacowanie złożoności programów i czasu wizualizacji stanów automatów. Wykorzystując narzędzia programistyczne PSoC – Expres mikrosystemu cyfrowego możemy przedstawić model sterowania pojazdu szynowego i realizować program w oparciu o sporządzony wcześniej graf automatu. Umożliwia to analizę graficzną zjawisk podczas symulacji sterowania pojazdem szynowym. Mikrosystemy cyfrowe stosowane są obecnie do sterowania hamulców (tablic pneumatycznych) w lokomotywach i jednostkach elektrycznych.

BIBLIOGRAFIA

1. Bocian S., Iwanowski J: Uniwersalne układy wejść – wyjść do zastosowania w pojazdach szynowych z wykorzystaniem mikrosystemu cyfrowego PSoC CY29466 firmy CYPRESS. Pojazdy Szynowe nr. 3/2008.
2. Bocian S. Iwanowski J: Układ przetwarzania sygnałów analogowych z czujników reluktancyjnych i program do pomiaru prędkości w pojazdach szynowych. Pojazdy szynowe, Nr 2/2010.
3. Bocian S.,Iwanowski J.: Zastosowanie maszyny stanowej do projektowania układów wejściowych wykorzystujący mikrosystem cyfrowy firmy CYPRESS z programem PSoC EXPRESS. Transcomp –XIV International Conference Computer Systems Aided Science, Industry and Transport, Zakopane 2010, (Logistyka Nr. 6/2010).
4. Bocian S.: Inteligentne podsystemy mechatroniczne w badaniach i sterowaniu pojazdów szynowych, Poznań 2012.
5. Grzymała – Busse J.W.: Podautomaty automatów skończonych związane ze zmianą czasu pracy, Politechnika Poznańska, Rozprawy nr.46, Poznań 1972.
6. Mikołajczak B.: Uogólnione przekształcenia okresowe automatów skończonych, Politechnika Poznańska, Rozprawy nr.98, Poznan1994.

DIGITAL MICROSYSTEMS USED IN THE RAIL VEHICLES AND THE COMPUTATIONAL COMPLEXITY OF THE CHARACTERISTIC SEMIGROUP OF THE ASYNCHRONOUS CONSISTENT AUTOMATON

Abstract

The article discusses the digital microsystems used in the rail vehicles. They allow to program based on the functional implemented blocks of the input and outputs and the transfer functions (state machine – automation) based on the previously drawn graph of the automation. The theorem is also presented and the evidence of the computational complexity of the characteristic semigroup of the asynchronous consistent automaton is also carried out.

Autorzy:
dr inż. **Stanisław Bocian –** Instytut Pojazdów Szynowych „TABOR" w Poznaniu