

Krzysztof Firląg  
Wydział Transportu Politechniki Warszawskiej

## REALIZACJA SPECJALIZOWANYCH STEROWNIKÓW RUCHU DROGOWEGO

**Streszczenie:** W artykule przedstawiono ważną rolę, jaką pełnią w systemach sterowania ruchem drogowym sterowniki lokalne. Wzrost wymagań stawianych tym sterownikom powoduje poszukiwanie nowych sposobów realizacji urządzeń sterowania ruchem drogowym. Do powstania nowych rozwiązań przyczynia się rozwój technologiczny i wprowadzenie do powszechnego użytku układów programowalnych. Połączenie tych zagadnień pozwoliło na budowę specjalizowanych sprzętowych sterowników ruchu drogowego w technice SoC.

W artykule przedstawiono główne punkty i założenia metody realizacji urządzeń sterowania ruchem drogowym w formie specjalizowanych urządzeń programowalnych, obejmujące specyfikację modeli urządzeń sterowania w językach opisu sprzętu z wykorzystaniem grafów przejść automatów skończonych gdzie wprowadzenie hierarchiczności wyeliminowało płaskość modelu. Przedstawiono również założenia weryfikacji modeli urządzeń oraz elementy analizy niezawodnościowej.

**Słowa kluczowe:** sterowanie ruchem drogowym, specjalizowane sterowniki ruchu drogowego, układy programowalne

### 1. WSTĘP

Rosnąca intensywność ruchu drogowego, pogarszająca jego warunki, zmusza do poszukiwania coraz nowocześniejszych i efektywniejszych metod sterowania ruchem na skrzyżowaniach, zapewniających również wysoką klasę bezpieczeństwa. Wymusza to ciągły rozwój systemów sterowania ruchem drogowym.

Duża dynamika procesów ruchu powoduje, że większość funkcji sterowania ruchem przekazywanych jest na poziom lokalny systemu gdzie znajdują się sterowniki lokalne ruchu drogowego [10]. Urządzenia te bezpośrednio odpowiadają za sterowanie ruchem na skrzyżowaniu. Te cyfrowe systemy reaktywne pracują w warunkach niepełnej informacji o procesie ruchu i są sterowane zdarzeniami występującymi w tym procesie. Informacje o stanie procesu dostarczane są poprzez ciec czujników, detektorów ruchu. Decyzje wpływające na bezpieczeństwo ruchu często muszą być podejmowane w czasie rzeczywistym, czemu z kolei nie sprzyjają złożone algorytmy sterowania ruchem drogowym.

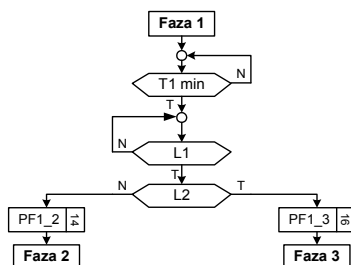
## 2. URZĄDZENIA STEROWANIA RUCHEM DROGOWYM

### 2.1. Algorytmy sterowania ruchem drogowym

Idea sterowania ruchem we współczesnych sterownikach lokalnych ruchu drogowego opiera się na realizacji przez sterownik algorytmu sterowania ruchem, czyli skończonego, uporządkowanego ciągu jasno zdefiniowanych czynności, koniecznych do wykonania założonej funkcji sterowania ruchem.

Większość stosowanych współcześnie sterowników ruchu drogowego pracuje w zamkniętych układach sterowania, realizując sterowanie adaptacyjne do warunków ruchu na skrzyżowaniu. Wytyczne, co do projektowania sygnalizacji zawiera rozporządzenie ministra infrastruktury nr 220 poz. 2181 z dnia 3 lipca 2003 r. Określa ono szereg etapów, które musi zawierać projekt sygnalizacji na skrzyżowaniu. W przypadku sygnalizacji adaptacyjnej efektem prac projektowych jest algorytm sterowania ruchem na skrzyżowaniu, opisujący warunki zakończenia i rozpoczęcia poszczególnych faz ruchu, opracowany stosownie do celu sterowania i z zastosowaniem racjonalnych metod sterowania.

W inżynierii ruchu drogowego stosuje się algorytmy przypominające sieć działań, rys. 1. Zawiera ona dwa podstawowe bloki, blok stanu i blok warunku (warunki czasowe lub logiczne). Dodatkowo klatki zawierające opis PF1\_2 i PF1\_3 pełnią rolę bloków stanu wywołujących sekwencję zmian stanów sygnalizacji nazywaną przejściem międzyfazowym.



Rys. 1. Fragment algorytmu sterowania na krzyżowaniu.

Algorytmy sterowania są obecnie bardzo rozwijane i rozbudowywane. Rozwój algorytmów sterowania wymusza coraz większą ilość i złożoność realizowanych funkcji sterowania na skrzyżowaniach, implementacja zaawansowanych algorytmów sterowania adaptacyjnego, czy też realizacja rozbudowanych funkcji bezpieczeństwa w sterownikach lokalnych ruchu drogowego.

### 2.2. Sterowniki lokalne ruchu drogowego

Sterowniki lokalne ruchu drogowego realizują bezpośrednie sterowanie w obrębie skrzyżowania. Z tego też względu urządzenia te mają największy wpływ na bezpieczeństwo procesu transportowego. Zadania sterownika ruchu drogowego to przede wszystkim sterowanie ruchem w obszarze skrzyżowania, co wiąże się z realizacją algorytmu sterowania ruchem na skrzyżowaniu.

Technologicznie sterowniki ruchu drogowego od początków swojego istnienia nadały za rozwojem ówczesnej elektroniki i automatyki. Początkowo stosowano rozwiązania mechaniczne i elektromechaniczne. Wraz z rozwojem elektroniki wprowadzono rozwiązania cyfrowych sterowników, początkowo realizowanych, jako platformy sprzętowe z układów małej i średniej skali integracji (SSI, MSI), następnie zaczęto realizować sterowniki na platformach programowych z układów wielkiej skali integracji VLSI jako rozwiązania mikroprocesorowe [10]. Sterowniki mikroprocesorowe stanowią większość produkowanych obecnie sterowników ruchu drogowego.

Sterowniki sprzętowe składają się z wielu elementów elektronicznych. Działanie sterowników sprzętowych wynika bezpośrednio z ich struktury, algorytmy sterowania w tych sterownikach implementuje się na etapie syntezy struktury sprzętowej sterowników. Zmiana algorytmów sterowania wiąże się z fizyczną zmianą konfiguracji sterownika.

Sterowniki programowo realizujące algorytmy sterowania zawierają mikroprocesor oraz dodatkowe układy peryferyjne jak pamięci, bloki detektorów itp. Działanie układu polega na realizacji przez standardową strukturę sprzętową, algorytmu sterowania ruchem zapisanego w pamięci w postaci programu, czyli ciągu instrukcji. Narzuca to sekwencyjny sposób realizacji takiego programu.

Rosnące wymagania powodują, że producenci zmuszeni są do ciągłego udoskonalania i rozwoju tych urządzeń. Rośnie ilość modułów wchodzących w skład sterowników, dodawane są kolejne mikroprocesory, co zmusza producentów do zabiegów kompensujących wzrost zawodności urządzeń, co wiąże się ze wzrostem ich ceny. Powoduje to, że producenci coraz częściej sięgają w stronę nowych, wydajniejszych a zarazem niezawodnych rozwiązań sprzętowych.

Przeprowadzona analiza urządzeń sterowania ruchem drogowym, zarówno sterowników lokalnych realizujących sterowanie w tradycyjny sprzętowy sposób jak i w sposób programowy, wykazała wiele problemów wynikających z niedoskonałości powyższych rozwiązań. Braki tradycyjnych sprzętowych sterowników wynikają głównie z zaprzestania ich rozwijania na pewnym etapie i spowodowanej tym zaszłości technologicznej. Niedoskonałości programowej realizacji funkcji sterowania związane są z sekwencyjnym sposobem wykonywania programu oraz obsługi wejść i wyjść, wynikającym ze stosowania architektur von Neumanna lub Harvardzkiej w budowie mikroprocesorów [2]. Problem ten uwidacznia się zwłaszcza przy rozbudowanych algorytmach i wielu portach WE/WY. Sytuacja ta zachęca do poszukiwania innych rozwiązań sprzętowych, kompensujących niedoskonałości wcześniej wymienionych technik, a jednocześnie posiadających ich zalety.

### **2.3. Specjalizowane urządzenia sterowania ruchem**

Jako alternatywa do platform sprzętowych (VLSI, LSI) i programowych (mikroprocesorowych) nasuwają się układy specjalizowane realizowane z programowalnych struktur logicznych [11]. Układy te cechują się sprzętową strukturą przy programowym sposobie konfiguracji. Najnowsze układy programowalne charakteryzują się bardzo dużymi zasobami logicznymi rzędu milionów przeliczeniowych bramek logicznych, wieloma portami zewnętrznymi.

Wśród wielu rodzajów układów programowalnych wyróżniają się układy FPGA. Układy FPGA zbudowane są z macierzy bramek logicznych. Posiadają sieć programowalnych połączeń wewnątrz swojej struktury. Realizacja urządzeń z wykorzystaniem układów FPGA jest opłacalna już przy pojedynczych sztukach. Ponadto możliwość wielokrotnego programowania tych układów powoduje, że doskonale nadają się one do realizacji modeli i prototypów urządzeń.

Dużą zaletą układów specjalizowanych realizowanych w układach programowalnych jest fakt, że mogą być realizowane w technice SoC (System on Chip), gdzie całe urządzenie realizowane jest w postaci jednoukładowej.

Dodatkową zaletą układów programowalnych jest wykorzystanie języków opisu sprzętu HDL, jako platformy opisującej urządzenia realizowane w tych układach [13]. Języki te pozwalają na specyfikację zarówno struktury sprzętu jak i funkcji realizowanych przez urządzenia, przy czym w sposób naturalny pozwalają na uzyskanie współbieżności działania.

### **3. METODA REALIZACJI SPECJALIZOWANYCH URZĄDZEŃ STEROWANIA RUCHEM DROGOWYM**

Dotychczasowy brak badań zastosowań języków opisu sprzętu i układów programowalnych w modelowaniu urządzeń sterowania ruchem drogowym powoduje, że budowa urządzeń sterowania ruchem drogowym w układach programowalnych wymaga opracowania metody projektowania, łączącej w sobie zagadnienia modelowania algorytmów sterowania ruchem drogowym i realizacji architektur sprzętowych SoC.

Proces realizacji specjalizowanych urządzeń cyfrowych składa się z kilku etapów [3,7]. Kluczowym etapem jest etap projektowania, który wg [3] składa się z trzech zadań: modelowanie, synteza i optymalizacja, ocena poprawności.

Modelowanie jest niezwykle ważne, ponieważ pozwala na formalny opis działania układu. Modele muszą być czytelne dla konstruktorów i jednocześnie pozwalające na dalszą syntezę układu.

Klasyfikacja modeli opisu formalnego urządzeń cyfrowych, dzieli je, ze względu na możliwości ich zastosowania, do opisu części sterującej lub przetwarzania danych [3,7]:

- zorientowane na stan – modele behawioralne;
- zorientowane na aktywność;
- zorientowane na strukturę;
- zorientowane na dane;
- heterogeniczne.

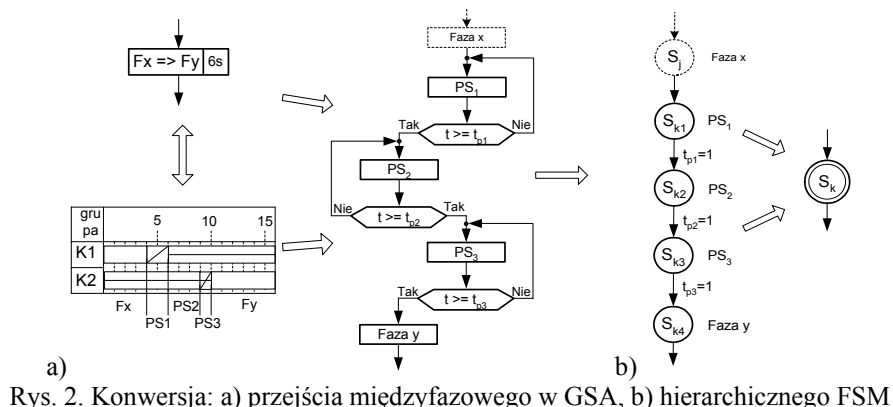
Pożądaną cechą modelu jest wspieranie przez niego opisu behawioralnego działania systemu oraz reprezentacja graficzna, ponieważ umożliwia opisywanie układu na poziomie jego zachowań bez konieczności znajomości szczegółów realizacyjnych. Przeanalizowano szereg modelei specyfikacji formalnej urządzeń cyfrowych [5], takich jak: grafy algorytmiczne ASM [7], Sieć działań GSA [7], Automaty FSM [3] i FSMD [7]; CFSM [9], HCFSM [9], Sieci Petriego [3], BDD [3], języki modelowania VHDL, Verilog.

### 3.1. Budowa modelu algorytmu sterowania ruchem drogowym

Po analizie przedstawionych modeli specyfikacji formalnej cyfrowych systemów sterowania dokonano wyboru modelu opisu i specyfikacji algorytmów sterowników ruchu drogowego realizowanych w układach programowalnych. Proponowana metoda modelowania urządzeń sterowania ruchem składa się z kilku etapów i łączy w sobie cechy kilku prezentowanych modeli.

Narzędzia syntezy do opisu urządzeń realizowanych w układach programowalnych wymagają języków opisu sprzętu. Fakt ten narzuca model opisu projektowanego układu, czyli jeden z języków opisu sprzętu. Języki opisu sprzętu, jak VHDL, pozwalają modelować zarówno strukturę układu jak i jego funkcje. Jednak języki te nie posiadają reprezentacji graficznej, przez co znacznie maleje przejrzystość modelowanego przy ich pomocy systemu.

Budowę modelu graficznego sterownika ruchu rozpoczęto od analizy algorytmów sterowania ruchem. Forma opisu algorytmów sterowania ruchem drogowym (rys. 1) w naturalny sposób nadaje się do specyfikacji przy pomocy modelu sieci działań GSA. Wymagane są jedynie drobne modyfikacje związane z klatkami przejść międzyfazowych. W efekcie modyfikacji otrzymujemy typową sieć działań GSA przedstawiającą algorytm sterowania ruchem (rys. 2a).



Rys. 2. Konwersja: a) przejścia międzyfazowego w GSA, b) hierarchicznego FSM

Modelem proponowanym do dalszego opisu sterownika ruchu drogowego jest model zorientowany na stany FSM. Do realizacji algorytmów srd wybrano automat Moore'a. W zasadzie rozwinięto model FSM wykorzystując model hierarchicznego FSM (rys. 2b). W stanach hierarchicznych można umieszczać zarówno sekwencje przejść międzyfazowych jak i obsługę poszczególnych faz ruchu. Wykorzystanie bloków hierarchicznych znacznie zwiększa przejrzystość grafu, eliminując konieczność operowania wszystkimi szczegółami specyfikacji.

Bloki przetwarzania danych, w metodzie projektowania sterowników specjalizowanych, realizuje się na poziomie języków opisu sprzętu, pozwalających w prosty, strukturalny lub funkcjonalny sposób na specyfikację bloków operacyjnych.

Do specyfikacji poziomego sterownika zawierającego układ sterujący i układy operacyjne korzystne jest wykorzystanie schematów blokowych (model strukturalny).

Dzięki zastosowaniu narzędzi wspomaganie komputerowego możliwa jest graficzna specyfikacja modelu sterownika ruchu w dedykowanych edytorach. Posiadając graficzną reprezentację modelu sterownika w automatyczny sposób generowany jest kod VHDL.

Proces ten kończy się otrzymaniem modelu sterownika ruchu w języku opisu sprzętu VHDL.

### 3.2. Weryfikacja działania algorytmów sterowania

Analiza metod weryfikacji i testowania [6], oraz prowadzone doświadczenia w oparciu o modele sterowników lokalnych, pozwoliły opracowywać metodę weryfikacji algorytmów sterowania oraz urządzeń sterowania ruchem drogowym modelowanych i realizowanych w układach programowalnych.

Weryfikację przeprowadza się na poziomie specjalistycznych narzędzi wspomaganie komputerowego, w których następuje specyfikacja układów. Wykorzystuje się pakiety oprogramowania: Active-HDL, a jako język specyfikacji VHDL. Podczas weryfikacji specyfikacji układów sterowania ruchem, za główny cel stawia się poprawną realizację algorytmów sterowania ruchem. Do testowania modelu wyspecyfikowanego w kodzie VHDL wykorzystywane są techniki testowania oprogramowania [12].

Weryfikację kodu VHDL układu sterowania rozpoczyna się od kompilacja projektu, eliminując na tym etapie błędy semantyczne i składniowe. Przy czym z powodu wielomodułowości układów sterowania zarówno kompilacja jak i dalsza weryfikacja prowadzona jest metodą wstępującą (bottom-up).

Weryfikacja przeprowadzana jest metodą weryfikacji funkcjonalnej z elementami weryfikacji strukturalnej. Do podawania wymuszeń na wyjścia układu zalecane jest wykorzystanie Testbench'y, automatyzując w ten sposób proces weryfikacji. Zbiór testów dla Testbench'y opracowuje się metodą Ad Hoc. Ponadto, dla grafów sterowania, zalecana jest automatyczna generacja Testbench'y.

W przypadku układów sterowania posiadających wiele wyprowadzeń, narzędziem pomagającym przy opracowaniu Testbench'y, zarówno przy weryfikacji funkcjonalnej jak i strukturalnej, jest Toggle Coverage. Narzędzie analizuje zmiany stanów portów układu, informując o ewentualnym braku aktywności na danym porcie.

W wyniku weryfikacji funkcjonalnej pokryte zostaną wszystkie funkcje układu, lecz nie cała jego struktura, należy więc włączyć do weryfikacji elementy testowania strukturalnego. Pomocne, do oceny jakości pokrycia strukturalnego, jest narzędzie Code Coverage. Narzędzie to automatycznie raportuje dwie główne miary z zakresu weryfikacji strukturalnej. Code coverage bada linie kodu, które zostały wykonane podczas wykonywania programu i sprawdza ile razy takie wywołanie nastąpiło. Branch Coverage z bada z kolei wywołania warunków typu IF, CASE.

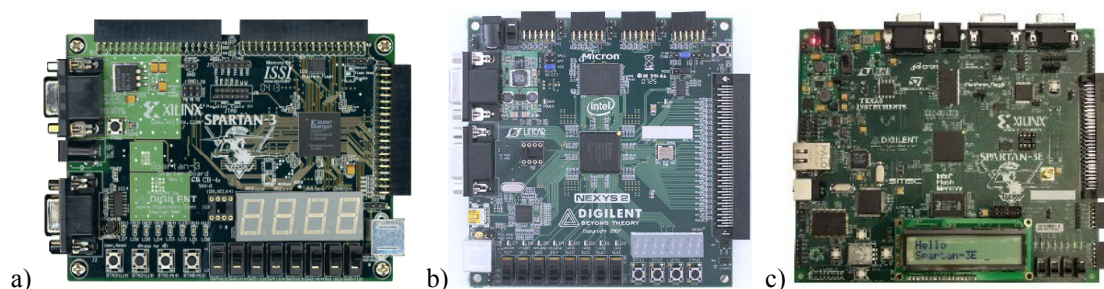
Budując Testbench'e na podstawie analizy kodu należy pamiętać o warunkach brzegowych. Należy zweryfikować działanie układu dla skrajnych wartości zadeklarowanych zmiennych, wykonać poszczególne pętle maksymalną ilość razy, sprawdzić działanie liczników, ich maksymalne zakresy i warunki zerowania.

Przeprowadzenie w ten sposób weryfikacji opracowanego kodu układu sterowania, pozwala znaleźć wszystkie błędy etapu specyfikacji i usunąć je w kolejnych rewizjach kodu VHDL opisującego układ sterowania ruchem.

## 4. PROTOTYPOWANIE URZĄDZEŃ STEROWANIA RUCHEM

Posiadając modele algorytmów sterowania ruchem drogowym w postaci hierarchicznych grafów przejść i modele strukturalne sterownika lokalnego ruchu drogowego, narzędzia CAD pozwalają na automatyczną generację modelu specjalizowanego sterownika ruchu w kodzie VHDL. Język ten jest platformą wyjściową dla narzędzi syntezy i implementacji. Po procesie syntezy i implementacji możliwa jest realizacja prototypu specjalizowanego sterownika ruchu drogowego.

Do realizacji prototypu wykorzystano karty uruchomieniowe, będące na wyposażeniu pracowni techniki cyfrowej Wydziału Transportu PW (rys. 3). Realizacja prototypów sterowników umożliwia badania symulacyjne rzeczywistych urządzeń, nie tylko modeli.



Rys. 3. Karty do prototypowania specjalizowanych sterowników ruchu drogowego: a) Spartan3 XC3S200 FT256, b) Spartan 3E-500 FG320, c) Spartan3 XC3S500E

Otrzymanie prototypu urządzenia umożliwia oszacowanie jego parametrów niezawodnościowych [8]. Wyznaczenie współczynnika MTBF dla specjalizowanego sterownika ruchu zrealizowanego na układach programowalnych przeprowadzono metodą wykorzystującą parametr FIT (Failure In Time) [uszkodzeń/ $10^9$  godziny].

Parametr FIT dla układów programowalnych jest wyznaczany przez producentów układów na podstawie szeregu testów opisanych w MIL-STD-883 [4]. Dodatkowo realizacja sterowników w formie modułów jednocukłowych SoC, wymaga wyznaczenia parametrów niezawodności tylko dla jednego modułu (obiektu). W tabeli 1 umieszczono podstawowe parametry niezawodnościowe dla przykładowego sterownika ruchu w różnych układach programowalnych.

Tablica 1.

Parametry niezawodności urządzeń sterowania wyznaczone z wykorzystaniem FIT					
Struktura logiczna	Virtex xcv50sc144	Virtex2 xc2v40cs144	Virtex4 xc4vfx12sf363	Spartan2 xc2s15vq100	Spartan3 xc3s200ft256
$\lambda$ [ $10^{-9}h^{-1}$ ]	13	17	3	12	3
R	0,9870841	0,98314368	0,997004496	0,98807171	0,997004496
MTBF [ $10^6h$ ]	76,923076	58,8235294	333,3333333	83,3333333	333,3333333

Wartość otrzymanego parametru MTBF sterownika ruchu drogowego potwierdza zalety jego realizacji w strukturach programowalnych.

## 5. WNIOSKI

Proponowana metoda projektowania i realizacji urządzeń sterowania ruchem drogowym pozwala zarówno na budowę modeli formalnych adaptacyjnych algorytmów sterowania ruchem, jak i na realizację w oparciu o zbudowane modele specjalizowanych sterowników ruchu. Koncepcja realizacji sterowników w formie układów SoC w układach FPGA pozwala na uzyskanie łatwo programowalnych urządzeń, o wysokich parametrach użytkowych i niezawodnościowych. Dodatkową zaletą metody jest wykorzystanie do modelowania sterowników modeli specyfikacji mających czytelne reprezentacje graficzne.

Proponowana metoda weryfikacji pozwala na wykrycie i eliminację błędów specyfikacji modelu urządzenia a dodatkowo badanie prototypu urządzenia pozwala na eliminację błędów realizacji.

### Bibliografia

1. Adamski M., Barkalov A.: Architectural and sequential synthesis of digital devices, Zakład Poligraficzny Uniwersytetu Zielonogórskiego, Zielona Góra, 2006.
2. Clements A.: Microprocessor Systems Design: 68000 Hardware, Software and Interfacing. PWS 97.
3. De Micheli G.: Synteza i optymalizacja układów cyfrowych, Wydawnictwo Naukowo-Techniczne, Warszawa, 1998.
4. Device Reliability Report, Fourth Quarter 2009. www.xilinx.com, Xilinx, 2009
5. Firląg K.: Metody modelowania algorytmów sterowania w specjalizowanych sterownikach ruchu drogowego. Logistyka nr 6/2009, Instytut Logistyki i Magazynowania, Poznań, 2009.
6. Firląg K., Kawalec P.: Weryfikacja i testowanie urządzeń sterowania ruchem drogowym realizowanych w układach FPGA. Logistyka nr 6/2009, Instytut Logistyki i Magazynowania, Poznań, 2009.
7. Gajski D.: Principles of digital design, Prentice Hall International, '97.
8. Kawalec P., Firląg K.: Reliability analysis of specialized traffic control devices. Archives of transport, volume 19, issue 1-2, Warszawskie Drukarnie Naukowe PAN, Warszawa, 2007 str. 75-82.
9. Lee B., Lee E. A.: Hierarchical Concurrent Finite state Machine in Ptolemy, Proceeding International Conference on Application of Concurrency to System Design, Fukushima, Japan, 1998, p. 34-40.
10. Leśko M., Guzik J.: Sterowanie ruchem drogowym. Sterowniki i systemy sterowania i nadzoru ruchu, Wydawnictwo Politechniki Śląskiej, Gliwice, 2000.
11. Pasierbiński J., Zbysiński P.: Układy programowalne w praktyce, WKŁ, Warszawa, 2001.
12. Patton R.: Software testing. 2-nd edition. USA, Pearson Education, copyright by SAMS Publishing, 2005.
13. Wrona W.: VHDL język opisu i projektowania układów cyfrowych. Gliwice, Wydawnictwo Jacka Skalmierskiego, 1998.

### REALIZATION OF SPECIALIZED TRAFFIC CONTROLLERS

**Abstract:** The article indicates a key role of local traffic controllers in the system of road traffic control, emphasizing at the same time the increasing requirements that these devices are presented with resulting from constantly growing road traffic intensity.

The article indicates the need for realization of road traffic control devices in the form of specialized controllers realized as single-system devices of SoC type with the use of programmable devices. Merits of such a realization have been pointed out.

After the analysis of formal specification models of digital controllers, a finite automaton with FSM memory has been chosen for realization of model controller. Additional extension of the model by hierarchization eliminated flatness of the model.

The paper includes methods of testing digital traffic control devices. The paper focuses on the tests of control devices which are realized in programmable logic devices FPGA. For the obtained prototype of traffic control devices reliability measures have been calculated.

**Keywords:** specialized traffic controllers, control devices verification, programmable devices