

METODY SYNTEZY I REALIZACJI SPECJALIZOWANYCH STEROWNIKÓW RUCHU DROGOWEGO W STRUKTURACH REPROGRAMOWALNYCH

Kluczowe zadanie w systemach sterowania ruchem drogowym pełnią sterowniki lokalne ruchu drogowego. Obecnie stosowane sterowniki są urządzeniami mikroprocesorowymi, realizującymi zaawansowane algorytmy sterowania zarówno lokalnego jak i obszarowego.

W artykule przedstawiono argumentację na rzecz stosowania w tego typu urządzeniach rozwiązań sprzętowych, układów reprogramowalnych oraz wady i zalety tego typu realizacji. Przedstawiono proponowaną metodę realizacji i syntezy lokalnych sterowników ruchu drogowego w postaci specjalizowanych jednoukładowych urządzeń sprzętowych. Dzięki czemu, poza niezawodnością urządzeń, znacznie wrasta ich szybkość przetwarzania. Pozwala to na implementację dodatkowych funkcjonalności dostępnych dotychczas w klasycznych sterownikach jedynie jako oddzielne moduły.

SYNTHESIS AND REALIZATION METHODS OF SPECIALISED TRAFFIC CONTROLLERS IN REPROGRAMMABLE LOGIC DEVICES

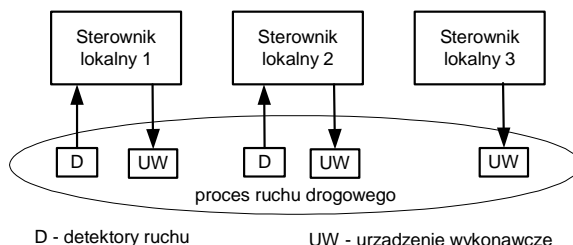
Local traffic control devices play a key role within traffic control systems. Currently used controllers are microprocessor devices realizing advanced algorithms of both local and area control. The paper presents argumentation substantiating the use of hardware solutions and programmable structures in this type of devices, as well as the advantages and disadvantages of realization of this kind. The proposed method has been presented of realization and synthesis of local traffic control devices in the form of specialized single-system hardware devices, due to which, apart from reliability of these devices, their processing speed increases significantly. This allows for implementation of additional functionalities available so far in classic controllers only as separate modules. The method at the same time allows for specification of controllers with the use of intuitive graphic models.

1. WSTĘP

Urządzeniem sterującym, realizującym sterowanie na najniższym poziomie systemu sterowania ruchem drogowym jest sterownik lokalny. 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

¹ Politechnika Warszawska, Wydział Transportu; 00-662 Warszawa; ul. Koszykowa 75; tel: + 48 22 234-75-85; e-mail: kfr@it.pw.edu.pl

występującymi w tym procesie. Informacje o stanie procesu dostarczane są poprzez sieć czujników, detektorów ruchu (rys. 1).



Rys. 1. Struktura systemu z bezpośrednim sterowaniem ruchem w strukturze zamkniętej (sterowniki 1 i 2) i otwartej (sterownik 3) [7]

Sterownik lokalny, jak każde urządzenie sterujące, realizuje założony algorytm sterowania ruchem [5]. O ile sama realizacja algorytmu sterowania ruchem drogowym nie wymaga dużej szybkości przetwarzania sterownika, to obsługa współczesnych układów detekcji i wideo-detekcji wymaga znacznych szybkości przetwarzania i zasobów logicznych. Stąd też szybkość przetwarzania i zasoby urządzeń powinny być kluczowymi kryteriami uwzględnionymi przy projektowaniu i budowie sterowników lokalnych.

W artykule przedstawione zostaną technologie stosowane we współczesnych sterownikach lokalnych ruchu drogowego oraz zalety i problemy związane z tymi technologiami. W efekcie przedstawione zostanie rozwiązanie specjalizowanego sterownika ruchu drogowego, zrealizowanego w technologii układów programowalnych, wraz z zaletami tego rozwiązania oraz metodą realizacji urządzeń w tej technologii.

2. ARCHITEKTURA STEROWNIKA LOKALNEGO RUCHU DROGOWEGO

2.1 Współczesne rozwiązania sterowników lokalnych

Technologicznie sterowniki ruchu drogowego od początków swojego istnienia nadążały 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 [5]. Sterowniki mikroprocesorowe stanowią większość produkowanych obecnie sterowników ruchu drogowego.

Głównym zadaniem sterownika ruchu drogowego jest sterowanie ruchem w obszarze skrzyżowania, co wiąże się z realizacją algorytmu sterowania ruchem na skrzyżowaniu. Ze względu na sposób realizacji algorytmu sterowania ruchem stosuje się podział na:

- sterowniki sprzętowe realizujące algorytmy sterowania;
- sterowniki programowo realizujące algorytmy sterowania.

Pierwszą grupę stanowią rozwiązania oparte na układach SSI i MSI, drugą rozwiązania mikroprocesorowe. 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. Główne zalety tego typu rozwiązań to pewność konfiguracji, wynikająca ze stałych połączeń elementów, szybkość działania wynikająca z równoległego przetwarzania różnych sygnałów i danych oraz specjalizacja do określonego zadania. Podstawowymi wadami są czas działania bramek i układów logicznych (np. w technologii TTL), duża ilość połączeń wpływająca na niezawodność urządzenia, problem modyfikacji parametrów działania układu (algorytmów sterowania) wymagający ingerencji w fizyczną strukturę urządzenia (lutowania połączeń lub układów).

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. Główną zaletą tego typu rozwiązań jest uniwersalność, wynikająca z łatwości zmiany sposobu działania urządzenia (zmiana programu zapisanego w pamięci). Natomiast 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 [1]. Problem ten uwidacznia się zwłaszcza przy rozbudowanych algorytmach i wielu portach WE/WY. Rosnące wymagania (adaptacja, wideo-detekcja) powodują, że producenci zmuszeni są do ciągłego udoskonalania i rozwoju tych urządzeń. Rośnie ilość modułów wchodzących w skład sterowników, zwiększana jest częstotliwość taktowania, dodawane są kolejne mikroprocesory, co zmusza producentów do zabiegów kompensujących wzrost zawodności urządzeń, co wiąże się ze wzrostem ich ceny.

Sytuacja ta zachęca do poszukiwania innych rozwiązań sprzętowych, kompensujących niedoskonałości wcześniej wymienionych technik, a jednocześnie posiadających ich zalety. Rozwiązaniem zwiększającym szybkość przetwarzania sterowników może stać się powrót do sprzętowych realizacji urządzeń sterowania ruchem drogowym.

2.2 Specjalizowane sterowniki lokalne

Skokiem technologicznym w dziedzinie rozwiązań sprzętowych są układy specjalizowane ASIC, układy zaawansowane technologicznie, wykonywane i projektowane na zamówienie w fabryce. Niestety, chociaż układy charakteryzują się dużą szybkością i wydajnością, to głównym problemem jest ich uniwersalność i wysokie koszty produkcji.

Ewolucją układów ASIC było pojawienie się układów programowalnych [6]. Układy te pozwalają na sprzętową realizację algorytmów sterowania, przy czym, technologia ich projektowania i realizacji łączy w sobie zalety rozwiązań sprzętowych z elastycznością rozwiązań programowych. Dodatkowe możliwości stwarzają układy reprogramowalne FPGA. Układy te, pozwalają na wielokrotną zmianę swojej struktury (realizowanego algorytmu).

Zasoby układów FPGA sięgające milionów bramek logicznych i tysięcy wyprowadzeń powodują, że możliwa jest realizacja całych rozbudowanych systemów sterowania wewnątrz jednego układu reprogramowalnego w technice SoC (System on Chip).

Dodatkową zaletą układów programowalnych jest wykorzystanie języków opisu sprzętu HDL, jako platformy opisującej urządzenia realizowane w tych układach [8]. Języki te pozwalają na specyfikację zarówno struktury sprzętu jak i funkcji realizowanych przez urządzenia, przy czym w sposób naturalny pozwalają na uzyskanie współbieżności działania, tzn. niezależnego, równoczesnego przetwarzania sygnałów wejściowych.

Podsumowując, technologia układów programowalnych łączy zalety wspomnianych w poprzednim rozdziale technologii. Zalety te, to pewna sprzętowa struktura urządzenia, szybkość działania sprzętu, wynikająca ze współbieżności realizacji procesów oraz łatwa programowa konfiguracja. Jednocześnie realizacja urządzeń w układach programowalnych pozbawiona jest głównych problemów obecnych technologii, takich jak sekwencyjność realizacji procesów sterowania i przetwarzania danych oraz konieczności fizycznej zmiany konfiguracji układu przy zmianie założeń co do jego działania.

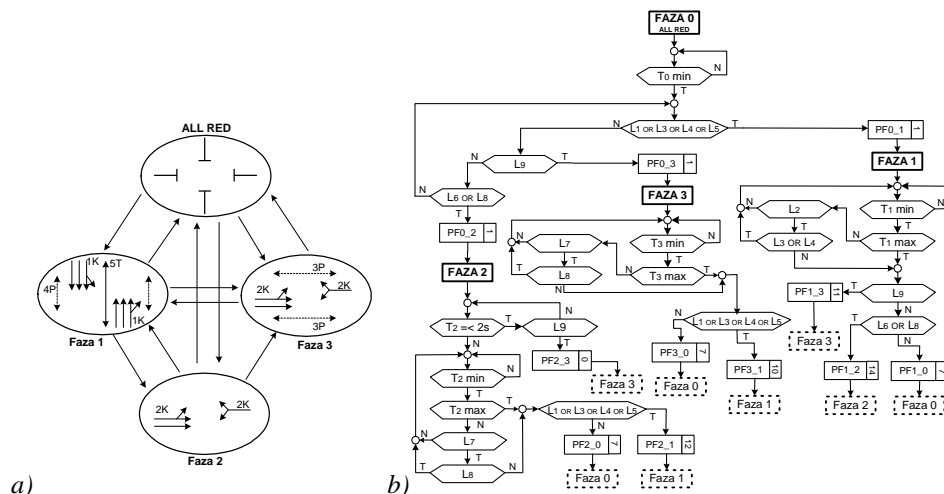
Należy podkreślić, że przeprowadzone badania [4] wykazały bardzo wysoką niezawodność urządzeń realizowanych w układach programowalnych, co stanowi kolejny argument na rzecz wykorzystania tej technologii.

3. REALIZACJA STEROWNIKA RUCHU DROGOWEGO W UKŁADACH FPGA

3.1 Projekt adaptacyjnej sygnalizacji świetlnej

Większość współcześnie stosowanych lokalnych sterowników ruchu drogowego realizuje sterowanie adaptacyjne do warunków ruchu na skrzyżowaniu. W przypadku sygnalizacji adaptacyjnej efektem prac projektowych jest algorytm sterowania ruchem na skrzyżowaniu, opisujący warunki zakończenia i rozpoczęcia faz ruchu, opracowany stosownie do celu sterowania i z zastosowaniem racjonalnych metod sterowania. Przykładowe sekwencję występowania faz ruchu na skrzyżowaniu przedstawia graf przejść pomiędzy fazami (rys. 2a).

Pomimo funkcjonowania kilku sposobów zapisu algorytmów sterowania dla skrzyżowań ruchu drogowego, najbardziej czytelną jest forma opisu bazująca na niemieckich wytycznych zawartych w RiLSA [3]. Forma graficzna algorytmów sterowania adaptacyjnego (rys. 2b) zawiera dwa podstawowe bloki, blok stanu i blok warunku, oraz trzeci rodzaj bloków, pełniący rolę bloku stanu wywołującego sekwencję zmian stanów sygnalizacji, tzw. przejście międzyfazowe.



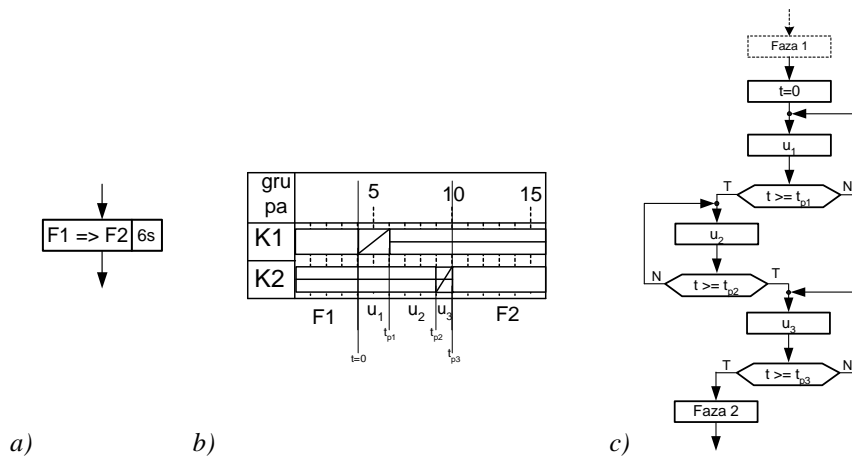
Rys. 2. Elementy projektu sygnalizacji świetlnej: a) graf dopuszczalnych przejść pomiędzy fazami, b) adaptacyjny algorytm sterowania ruchem na skrzyżowaniu

Przedstawione na rysunkach grafy są elementami projektu sterowania dla warszawskiego skrzyżowania Wołoska-Dąbrowskiego.

3.2 Budowa modelu bloku logicznego sterownika

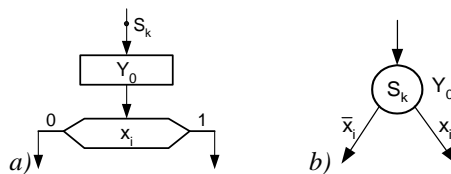
Do realizacji specjalizowanego sterownika ruchu drogowego w technologii układów FPGA wykorzystano metodę pozwalającą, w oparciu o adaptacyjne algorytmy sterownia na skrzyżowaniu, otrzymać model urządzenia w języku opisu sprzętu VHDL. Język ten jest uniwersalną i podstawową platformą opisu urządzeń realizowanych w układach programowalnych.

W pierwszym kroku algorytm sterowania ruchem w formacie RiLSA przekształcany jest do formalnej postaci GSA (graficzny schemat algorytmu). Przy tej zmianie najciekawszy etap stanowi zamiana klatki przejścia międzyfazowego w odpowiadający mu ciąg klatek sieci GSA (rys. 3).



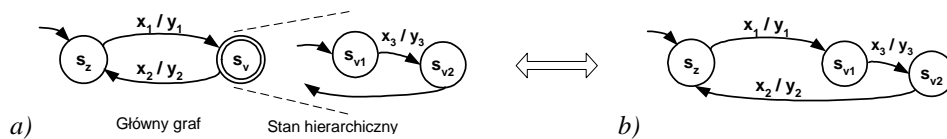
Rys. 3. Konwersja klatki przejścia międzyfazowego w GSA: a) klatka przejścia międzyfazowego, b) program przejścia międzyfazowego, c) sieć GSA przejścia międzyfazowego

Otrzymaną sieć działań GSA przekształca się w graf przejść automatu skończonego. Wykorzystywany jest graf automatu Moore'a, dla którego zasadę konwersji przedstawiono na rysunku 4.



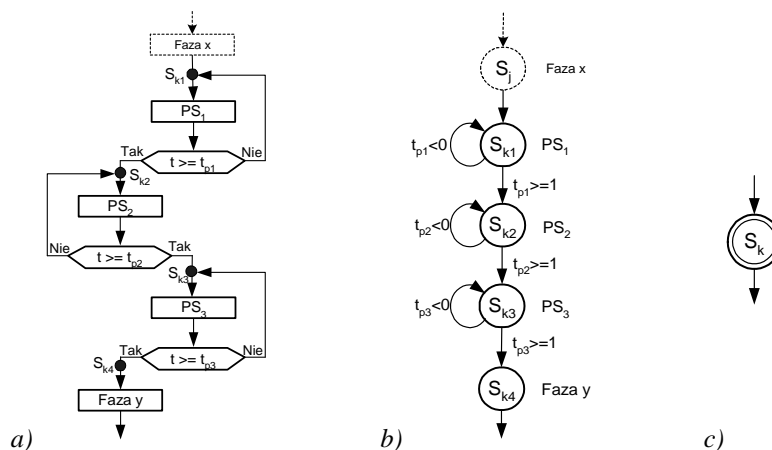
Rys. 4. Zamiana sieci działań w graf stanów automatu Moore'a: a) oznakowany fragment sieci działań, b) graf stanów automatu Moore'a odpowiadający oznakowanej sieci

Otrzymany według powyższej metody graf sterownia jest bardzo rozbudowany, na przykład dla algorytmu z rysunku 2b graf sterowania zawiera ponad 50 stanów. Sposobem na zwiększenie przejrzystości grafu, poprzez wyeliminowanie pewnej ilości stanów, jest wykorzystanie grafów hierarchicznych. W grafach tych, niektóre wierzchołki zawierają ukryte wewnątrz ciągi kolejnych stanów. Właściwie model ten różni się od automatu skończonego jedynie w warstwie graficznej (rys. 5).



Rys. 5. Graf hierarchiczny: a) graficzna reprezentacja hierarchicznego automatu skończonego z pamięcią, b) rozwinięcie grafu hierarchicznego na zwykły

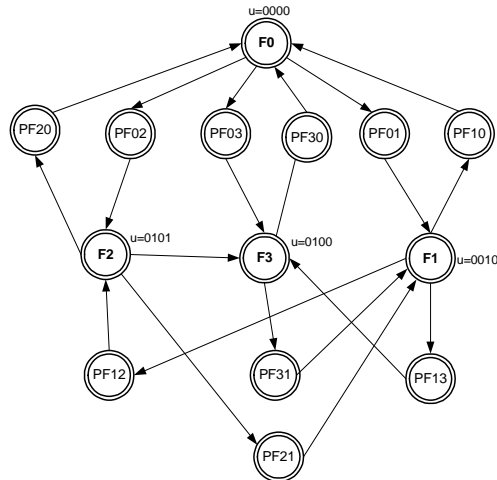
Stany hierarchie wykorzystano dla ukrycia sekwencji obsługi przejść międzyfazowych oraz obsługi faz ruchu. Zasadę budowy stanów hierarchicznych dla przejść międzyfazowych obrazuje rysunek 6.



Rys. 6. Etapy budowy stanu hierarchicznego dla przejścia międzyfazowego: a) GSA przejścia międzyfazowego, b) graf sterowania przejścia międzyfazowego, c) stan hierarchiczny dla przejścia międzyfazowego

Końcowym efektem jest model hierarchicznego grafu sterowania dla bloku logicznego specjalizowanego sterownika ruchu drogowego. Przykładowy graf sterowania, odpowiadający algorytmowi sterowania z rysunku 2b, przedstawiono na rysunku 7.

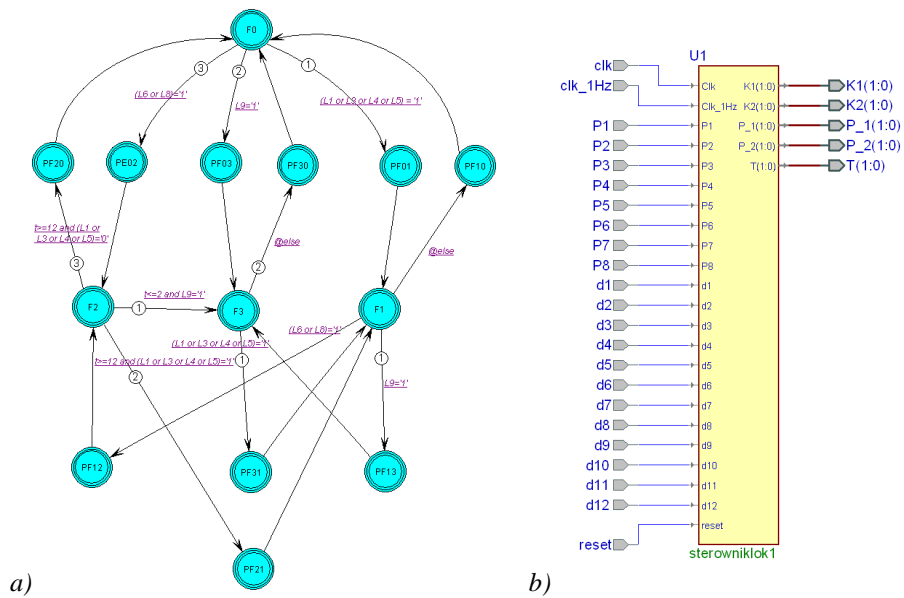
Graf ten odpowiada grafowi przedstawiającemu dopuszczalne przejścia faz ruchu na skrzyżowaniu (rys. 2a), wykorzystanym w początkowym etapie projektowania, co za tym idzie, model w tej formie jest przyjazny i intuicyjny dla każdego specjalisty ruchu drogowego nie znającego teorii automatów skończonych.



Rys. 7. Hierarchiczny graf sterowania ruchem drogowym na skrzyżowaniu

3.3 Specyfikacja bloku logicznego sterownika ruchu

Specyfikacja sterownika powinna doprowadzić do modelu bloków sterownika w językach opisu sprzętu VHDL, umożliwiającą dalszą realizację urządzenia praktycznie w dowolnej technologii, w tym również w układach programowalnych.



Rys. 8. Specyfikacja sterownika lokalnego: a) hierarchiczny graf sterowania ruchem na skrzyżowaniu, b) blok logiczny sterownika

Specyfikacja urządzenia bezpośrednio w języku VHDL wymaga znacznych umiejętności programistycznych. W związku z tym do specyfikacji szukano narzędzi CAD oferujących projektantowi intuicyjne graficzne środowisko. Ostatecznie wybrano pakiet Active-HDL firmy Aldec.

Otrzymany graf sterowania automatu skończonego (rys. 7) wyspecyfikowano w graficznym edytorze FSM programu Active-HDL (rys. 8a). Edytor ten pozwala również na budowę wierzchołków hierarchicznych.

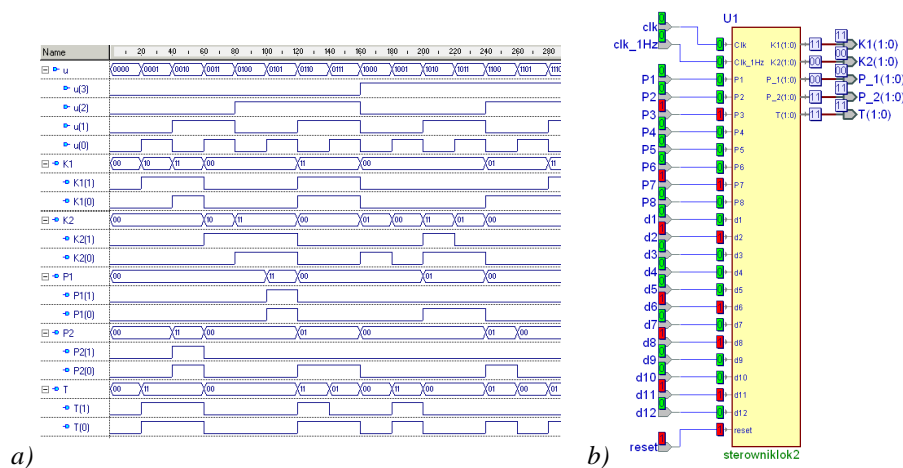
Opracowaną uniwersalną architekturę bloku logicznego sterownika ruchu drogowego wyspecyfikowano w formie modelu strukturalnego w edytorze BDE programu Active-HDL (rys. 8b).

Taka specyfikacja bloków pozwala na automatyczną generację kodu VHDL, przez pakiet Active-HDL. Otrzymuje się w ten sposób model realizowanego sterownika w języku opisu sprzętu. Opis w tym języku wymagany jest przez narzędzia do syntezy i implementacji urządzeń w struktury FPGA.

3.4 Weryfikacja działania sterownika ruchu

Weryfikacja jest niezwykle istotnym etapem realizacji urządzeń sterowania ruchem [2]. Weryfikację przeprowadza się na poziomie specjalistycznych narzędzi wspomaganie komputerowego, w których następuje specyfikacja układów. Podczas weryfikacji specyfikacji układów, za główny cel stawia się poprawną realizację algorytmów sterowania ruchem.

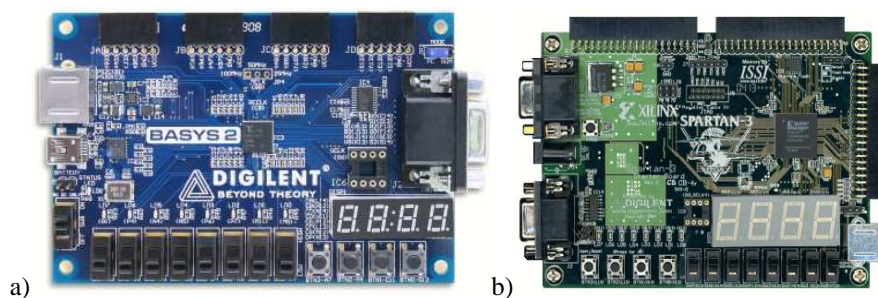
Weryfikacja przeprowadzana jest metodą weryfikacji funkcjonalnej z elementami weryfikacji strukturalnej. Do podawania wymuszeń na wyjścia układu zalecane jest wykorzystanie Testbench'y, automatyzując w ten sposób proces weryfikacji. Wyniki weryfikacji można analizować w formie przebiegów czasowych sygnałów (rys. 9a). Możliwa jest również obserwacja zmian stanów bezpośrednio na grafie sterowania lub sygnałów bezpośrednio na modelu strukturalnym (rys. 9b).



Rys. 9. Weryfikacja działania sterowania: a) weryfikacja na przebiegach czasowych, b) weryfikacja na schemacie blokowym

3.5 Realizacja prototypu urządzenia

Wykorzystywane oprogramowanie użytkowe pozwala na automatyczny proces syntezy i implementacji modeli sterowników specjalizowanych w układach programowalnych. Zastosowanie kart wyposażonych w układ programowalny i odpowiednie układy wejścia-wyjścia, umożliwiające prototypowanie projektowanych urządzeń (rys. 10).



Rys. 10. Karty do prototypowania specjalizowanych sterowników ruchu drogowego: a) Spartan3E-100 CP132, b) Spartan3 XC3S200 FT256

Kartę uruchomieniową łączy się komputerem wyposażonym w odpowiednie oprogramowanie (ISE). Dokonanie programowania odbywa się, w zależności od stanowiska uruchomieniowego, poprzez interfejs USB, JTAG (IEEE 1149.1) lub specjalistycznymi kablami (XChecker).

Tabela 1 zawiera parametry z przykładowego wykorzystania zasobów układów prototypowych Spartan3 XC3S200 FT256, dla sterownika którego algorytmy przedstawiono na rysunku 2b.

Tab. 1. Wykorzystanie zasobów układów programowalnych przez sterownik

Struktura logiczna	Wyprowadzeń WE/WY	Bloków Slice	Komórek LUT 4 wejściowych	Przerzutników	Max częstotliwość pracy
Spartan3 xc3s200ft256	31/173 (16%)	95/1920 (4%)	174/3840 (4%)	81/3840 (2%)	190,767 MHz

4. WNIOSKI

Przedstawiona metoda realizacji i syntezy specjalizowanych sterowników ruchu drogowego pozwala na tworzenie nowej klasy rozwiązań sprzętowych w urządzeniach SRD. Przeprowadzone badania prototypów sterowników logicznych ruchu drogowego pozwalają na stwierdzenie, że technologia realizacji sterowników specjalizowanych w układach FPGA powoduje, że częstotliwości realizacji funkcji sterujących są ograniczone jedynie czasami propagacji sygnałów w strukturze FPGA, przy czym naturalna współbieżność układów FPGA znacznie przyspiesza przetwarzanie. Otrzymanej częstotliwości taktowania układów rzędu 190 [MHz] (tab. 1) miałby szansę dorównać układ mikroprocesorowy taktowany częstotliwością przynajmniej kilku GHz.

Koleiną zaletą przedstawianych rozwiązań jest możliwość jednocukładowej realizacji specjalizowanego sterownika logicznego ruchu drogowego. Modele sterowników logicznych, sprowadzone do jednocukładowych realizacji, zostały zaimplementowane do pojedynczych struktur FPGA w stanowiskach badawczych, tworząc architektury SoC (cały sterownik w jednym układzie scalonym).

Na korzyść tych rozwiązań wskazują, nie zamieszczone w tym artykule, parametry niezawodnościowe otrzymanych specjalizowanych sterowników ruchu drogowego [4].

Co do samej metody syntezy sterowników specjalizowanych, jest ona przyjazna dla projektanta, gdyż wykorzystuje graficzne modele formalne oraz intuicyjne narzędzia wspomagania komputerowego. Intuicyjność tej metody powoduje, że cały proces realizacji specjalizowanych sterowników może być przeprowadzony przez inżyniera ruchu drogowego bez zaawansowanej znajomości elektroniki. Ponadto wykorzystanie układów reprogramowalnych do realizacji sprzętowych sterowników specjalizowanych ruchu drogowego, ułatwia proces prototypowania, praktycznie likwidując koszty związane z budową kolejnych prototypów. Otrzymane na etapie produkcji sterowniki oparte na układach reprogramowalnych również posiadają cechy wyjątkowo łatwej rekonfiguracji. Gdyż wgranie nowej konfiguracji układu odbywa się analogicznie do rozwiązań programowych.

5. BIBLIOGRAFIA

- [1] Clements A.: *Microprocessor Systems Design: 68000 Hardware, Software and Interfacing*. PWS, 1997.
- [2] 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.
- [3] FGSV – Forschungsgesellschaft fuer Strassen- und Verkehrswesen (2003a). *Guidelines for Traffic Signals. English Version of Richtlinien fuer Lichtsignalanlagen RiLSA. Edition 1992 (with minor modifications), Translation 2003*. FGSV 321/S, FGSV-Verlag, Cologne, 2003.
- [4] Kawalec P., Firląg K.: Reliability analysis of specialized traffic control devices. *Archives of transport*, volume 19, issue 1-2, Warszawska Drukarnia Naukowa PAN, Warszawa, 2007 str. 75-82.
- [5] Leško M., Guzik J.: *Sterowanie ruchem drogowym. Sterowniki i systemy sterowania i nadzoru ruchu*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2000.
- [6] Pasierbiński J., Zbysiński P.: *Układy programowalne w praktyce*, WKŁ, Warszawa, 2001.
- [7] Wawrzyński W.: *Bezpieczeństwo systemów sterowania w transporcie*. Wydawnictwo Instytutu Technologii Eksploatacji, Warszawa-Radom, 2004.
- [8] Wrona W.: *VHDL język opisu i projektowania układów cyfrowych*. Gliwice, Wydawnictwo Jacka Skalmierskiego, 1998.