II Konferencja Naukowa KNWS'05
"Informatyka- sztuka czy rzemiosło"
15-18 czerwca 2005, Złotniki Lubańskie

ROZPROSZONY SYSTEM STEROWANIA BEZPIECZNEGO Z REKONFIGUROWALNYMI UKŁADAMI LOKALNYMI

Marek Węgrzyn

Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50

e-mail: M.Wegrzyn@iie.uz.zgora.pl

STRESZCZENIE

W referacie przedstawiono propozycję systemu sterowania opartego na strukturze trzypoziomowej. Poziomem najwyższym jest aplikacja programowa, za pomocą której wybierane są odpowiednie podprogramy sterowania, na podstawie podejmowanych decyzji, dokonująca stosownych uaktualnień w nadrzędnej bazie danych. Dostęp do poziomu nadzorującego realizowany jest za pośrednictwem komputerowych sieci rozległych (okablowanie), wraz z rezerwowym kanałem komunikacyjnym (łączność radiowa lub telefonia komórkowa). Poprawność wykonania poszczególnych zadań jest potwierdzana odpowiednim wpisem do nadrzędnej bazy danych. W systemie istnieje możliwość wprowadzania dodatkowych stopni pośrednich, zarówno dodatkowych baz danych, jak i dedykowanych, w tym dynamicznie rekonfigurowanych, sprzętowo-programowych elementów sterujących i wykonawczych. Do specyfikacji algorytmów sterowania wykorzystywane są hierarchiczne sieci Petriego, które są reprezentowane w postaci odpowiednich strukturach bazy danych.

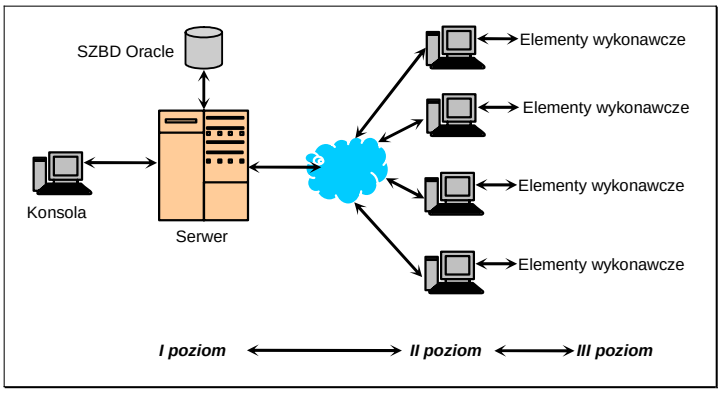
1. WPROWADZENIE

Realizacja złożonych algorytmów sterowania wymaga nowego podejścia zarówno do specyfikacji programu sterowania, jak również opracowania nowych modeli struktur takich systemów. Zaawansowane rozwiązania sprzętowe w połączeniu z nowoczesnymi technologiami informatycznymi pozwalają na usprawnienie procesu projektowania systemów sterowania. Efektywne jest wykorzystanie systemów zarządzania bazami danych do gromadzenia i dystrybucji informacji o aktualnie planowanych do realizacji programach sterowania Z drugiej strony realizacja sprzętowa, oprócz korzyści związanych z wydajnością ułatwia weryfikację działania systemu.

2. OGÓLNA KONCEPCJA SYSTEMU

Podstawowym, rozpatrywanym systemem sterowania jest struktura trzypoziomowa. Poziomem najwyższym jest aplikacja programowa, za pomocą której wybierane są

odpowiednie podprogramy sterowania, na podstawie podejmowanych decyzji, dokonująca stosownych uaktualnień w nadrzędnej bazie danych. Dostęp do poziomu nadzorującego realizowany jest za pośrednictwem komputerowych sieci rozległych (okablowanie), wraz z rezerwowym kanałem komunikacyjnym (łączność radiowa lub telefonia komórkowa). Komunikacja bazy danych ze sterownikami odbywać się będzie z wykorzystaniem mechanizmów dostępnych w systemie zarządzania bazami danych firmy ORACLE, nazywanych procedurami zewnętrznymi. Wymaga to implementacji wymaganych zadań bazując na niskopoziomowych instrukcjach języka C. W celu zapewnienia poprawnej wymiany danych należy również odpowiednio skonfigurować i zestroić moduł komunikacji sieciowej. Poprawność wykonania poszczególnych zadań powinna być potwierdzana odpowiednim wpisem do nadrzędnej bazy danych. Pomimo koncentracji badań na z góry określonej strukturze, istnieje możliwość wprowadzania dodatkowych stopni pośrednich, zarówno dodatkowych baz danych, jak i dedykowanych, w tym dynamicznie rekonfigurowanych, sprzętowo-programowych elementów sterujących i wykonawczych. Do specyfikacji wykorzystywane są hierarchiczne sieci Petriego [5]. Ogólny schemat proponowanej struktury systemu przedstawia rys. 1.



Rys. 1. Ogólny schemat systemu

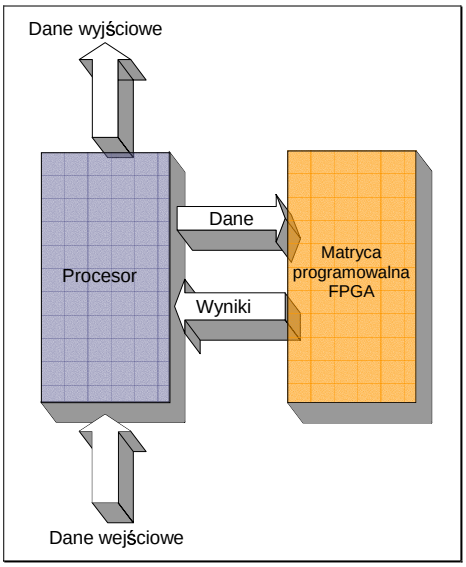
Wysoce niezawodne programowane układy elektroniczne przeznaczone do wykorzystania w układach sterowania o podwyższonym bezpieczeństwie (ang. *safety critical control system, SCCS*) stanowią nowy obszar badawczy, znajdujący się w dopiero początkowej fazie prac badawczych i inżynierskich. Znaczenie tematu wynika z jednej strony ze stale rosnących wymagań społeczeństwa, co do bezpieczeństwa, a z drugiej strony z technologicznych tendencji w kierunku bardziej wydajnych i elastycznych, tzn. kontrolowanych, sprzętowo-programowych urządzeń sterujących. Celem, do którego się dąży jest osiągnięcie sytuacji, w której systemy czasu rzeczywistego można będzie projektować z wystarczająco dużym stopniem pewności, co do ich niezawodności, co umożliwi wydawanie licencji na ich zastosowanie w układach sterowania SCSC przez odpowiednie autorytety (instytucje), na bazie formalnej aprobaty.

Do realizacji sprzętowej wykorzystywane są układy typu „*System-on-a-Chip*" (*SoC*) z możliwością selektywnego rekonfigurowania bloków logicznych struktury FPGA (np. FPSLIC firmy Atmel), w celu zdalnego dopasowania realizowanego programu (w szczególności awaryjnego) do zmieniających się wymogów. Programowalne układy logiczne oprócz zalet związanych z procesem prototypowania, spowalniają proces starzenia się urządzeń realizujących procesy sterowania, jak również usprawniają procesy obliczeniowe. Konieczność uaktualnienia realizowanego przez urządzenie cyfrowe algorytmu z reguły pociąga za sobą konieczność fizycznej obecności serwisu. W przypadku sterowania rozproszonego wymiana urządzeń w wielu punktach jednocześnie może spowodować zachwianie stabilności funkcjonowania systemu. Dlatego ważna jest możliwość zarządzania centralnego dopasowaniem poszczególnych składników systemu. Dotychczasowe możliwości rekonfiguracji FPGA bazowały na wykorzystaniu „szeregowego ładowania" konfiguracji i wymagały bezpośredniego podłączenia rekonfigurowanego urządzenia do systemu programującego. Nowe rozwiązania SoC, ze szczególnym uwzględnieniem układów FPSLIC firmy Atmel, umożliwiają zdalne programowanie matrycy FPGA przez zintegrowany rdzeń mikroprocesora AVR. Wykorzystując nowe możliwości SoC w zakresie rekonfiguracji możliwe jest przygotowanie danych konfiguracyjnych FPGA przez system CAD na zdalnym komputerze centralnym. Następnie za pośrednictwem medium transmisyjnego (TCP/IP) przekazanie danych konfiguracyjnych do zdalnego urządzenia, którego konfiguracja ma zostać zmieniona. W następnym etapie, w oparciu o dotychczas niedostępne metody, wykorzystując zintegrowany mikroprocesor przeprogramować sterownik zrealizowany w FPGA. Takie podejście umożliwia zdalną weryfikację oraz dopasowanie sterowników rozproszonych do zmieniających się wymagań, uwzględniając weryfikację bezpieczeństwa oraz poprawności realizowanych algorytmów.

3. METODY REALIZACJI LOKALNYCH UKŁADÓW STEROWANIA

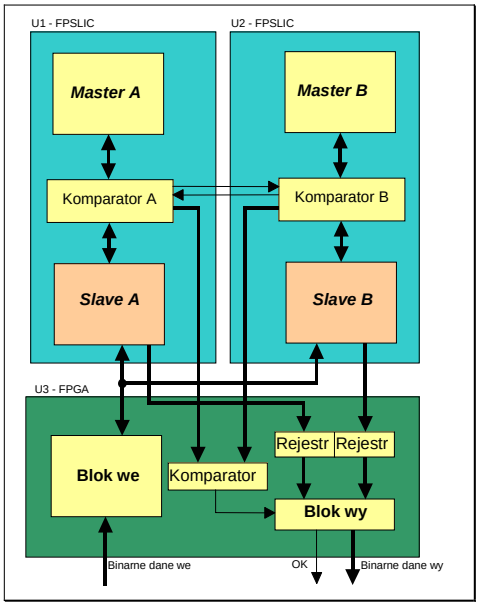
Lokalne układy sterowane rozpatrywane są jako sterowniki realizowane w nowoczesnych strukturach FPGA, w tym również takich, które zawierają, oprócz programowanej struktury także wbudowane rdzenie układów mikroprocesorowych (rys. 2).

W takim systemie możliwe jest zaimplementowanie oprogramowania, które odpowiedzialne będzie za zarządzane rekonfiguracją układu FPGA, czyli dopasowaniem struktury fizycznej układu do aktualnego algorytmu [3]. W celu zwiększenia elastyczności rekonfiguracji analizowane są możliwości wymiany jedynie części informacji, tzn. przeprogramowanie jedynie fragmentu układu. Dlatego układ należy programować odpowiednim strumieniem, który zawiera wyłącznie różnicę konfiguracji – strumieniem różnicowym. Ponadto, dla rozpatrywanego systemu mikroprocesorowego możliwe jest opracowanie systemu operacyjnego spełniającego wymagania czasu rzeczywistego (ang. *real-time operating system*, *RTOS*).



Rys. 2. Przykładowa architektura lokalnego układu sterowania

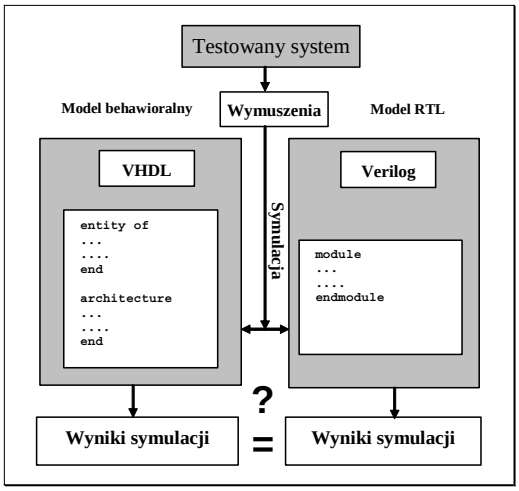
Przykładowe rozwiązanie sterownika o podwyższonym stopniu bezpieczeństwa opiera się na architekturze składającej się z dwóch par procesorów *master-slave* [2], których poprawność działania na bieżąco jest weryfikowana przez dodatkowe układy porównujące (komparatory). Realizację w strukturach FPGA z zintegrowanymi mikroprocesorami [1] przedstawiono na rys. 3.



Rys. 3. Realizacja lokalnego układu sterowania z wykorzystaniem struktur FPSLIC

Procesor *master* jest konfigurowany za każdym razem, zgodnie z aktualnym algorytmem sterowania. *Master* zleca oraz nadzoruje wykonanie zadań przez procesor *slave*. Raz zaprojektowany *slave* wykonuje tylko zadania ze stałego zbioru zaimplementowanych wcześniej procedur.

Zagadnienia bezpieczeństwa stanowią bardzo ważne aspekty realizacji rozpatrywanych sterowników. Jednak, ze względu na złożoność układu pełna weryfikacja formalna jest bardzo utrudniona, a często wręcz niemożliwa. Dlatego należy szukać innych metod uwiarygodnienia procesu projektowania i wykonania. Najczęściej stosowanymi metodami są złożone metody symulacyjne [4]. Proponowana metoda wymaga opracowania różnych modeli, i to zarówno na różnym poziomie abstrakcji (modele behawioralne oraz strukturalne), jak również przy zastosowaniu różnych języków opisu sprzętu (np. VHDL i Verilog) [6]. Zgodność wyników przy tak różnorodnych podejściach podnosi poziom wiarygodności całego etapu weryfikacji. Środowisko symulacyjne bazujące na różnych modelach sterowników przedstawiono na rys. 4.



Rys. 4. Środowisko do weryfikacji symulacyjnej układu

Alternatywnym rozwiązaniem sprzętowym lokalnego układu sterowania może być dobrze znany z literatury układ mikroprogramowalny. Wymaga on jednak odpowiednich modyfikacji w celu lepszej weryfikacji poprawności działania. Ponadto, zastosowanie dedykowanych bloków pamięci dostępnych w układach FPGA umożliwia łatwiejsze, bezpośrednio odzwierciedlenia struktury układu w rzeczywistym elemencie.

4. PODSUMOWANIE

W referacie przedstawiono propozycję hierarchicznego systemu sterowania o strukturze trzypoziomowej. Poziom nadrzędny stanowi aplikacja z bazami danych, w której zapisany

jest algorytm sterowania. Przy realizacji lokalnych układów sterowania wykorzystane są sprzętowo-programowe rozwiązania bazujące na logice programowalnej, ze szczególnym uwzględnieniem struktur z wbudowanymi układami mikroprocesorowymi. W celu lepszego dopasowania układów lokalnych do wykonywanych zadań, rozpatrywane jest dynamiczne rekonfigurowanie tych układów z wykorzystaniem różnicowych strumieni konfigurujących.

W rozpatrywanym systemie szczególny nacisk położono na zagadnienia bezpieczeństwa w zakresie realizacji sterowników lokalnych. Wymaga to skomplikowanych zabiegów w czasie projektowania układów. Związane jest to między innymi z uwiarygodnieniem każdego etapu projektowania. Rozpatrywane są zarówno metody formalne, jak również rozbudowane metody symulacyjne bazujące na porównywaniu wyników dla różnych modeli.

Praca naukowa (częściowo) finansowana ze środków Komitetu Badań Naukowych w latach 2004-2006 jako projekt badawczy (grant nr 3 T11C 046 26).

LITERATURA

[1] A. Bukowiec, M. Węgrzyn: *Design of Logic Controllers for Safety Critical Systems Using FPGAs With Embedded Microprocessors*, w: M.Colnaric, W.A.Halang, M.Węgrzyn, (Eds.), Real-Time Programming 2004, (A Proceedings volume from the 28th IFAC/IFIP Workshop on Real-Time Programming, WRTP 2004 and the International Workshop on Software Engineering, IWSS 2004, Istanbul, Turkey, 6–8 September 2004), Elsevier, Oxford (UK), 2005

[2] W. A. Halang, M. Śnieżek, S.-K. Jung: *A Real-Time Computing Architecture for Applications with High Safety and Predictability Requirements*, Proceedings of the First IEEE International Workshop on Real-Time Computing System and Applications, 1994, ss.153-157

[3] R. Hartenstein: *Reconfigurable Computing: a New Business Model - and its Impact on SoC Design*, Proceedings of the EUROMICRO Symposium on Digital Systems Design, DSD 2001, ss.103-110

[4] A. Węgrzyn, M. Węgrzyn: *A New Approach to Simulation of Concurrent Controllers*, w: M.Adamski, A.Karatkevich, M.Węgrzyn, (Eds.), Design of Embedded Control Systems, Springer, Boston, 2004, ISBN: 0-387-23630-9

[5] M. Wegrzyn, M. Adamski: *Hierarchical Approach for Design of Application Specific Logic Controller*, Proceedings of the IEEE International Symposium on Industrial Electronics ISIE'99, Bled, Slovenia, 12-16.07.1999, Vol.3, ss.1389-1394

[6] M. Węgrzyn: *Implementation of Safety Critical Logic Controller by Means of FPGA*, Annual Reviews in Control, Vol.27, 2003, ss.55-61