

Michał DOLIGALSKI, Marian ADAMSKI

UNIwersytet Zielonogórski,
ul. Licealna 9, 65-417 Zielona Góra

Mechanizm obsługi sytuacji wyjątkowych oraz ponownego wznowienia sterowania w hierarchicznych sieciach Petriego

Mgr inż. Michał Doligalski

Absolwent Uniwersytetu Zielonogórskiego (2006). Ukończył studia o specjalności Inżynieria Komputerowa. Od roku 2006 pracuje jako asystent na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego. Zainteresowania badawcze obejmują metody projektowania i implementacji systemów cyfrowych. Członek Polskiego Towarzystwa Informatycznego, Polskiego Stowarzyszenia Zarządzania Wiedzą.



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

Prof. dr hab. inż. Marian ADAMSKI

Dyrektor Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Zainteresowania badawcze obejmują projektowanie mikrosterowników logicznych oraz formalne metody syntez, analizy i weryfikacji rekonfigurowalnych układów cyfrowych. Członek IEEE, IET, ACM, Polskiego Towarzystwa Elektrotechniki Teoretycznej i Stosowanej, Polskiego Towarzystwa Informatycznego oraz Brazylijskiego Towarzystwa Mikroelektronicznego.



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

Streszczenie

Opis programów dla rekonfigurowalnych sterowników logicznych jest przeprowadzany z wykorzystaniem dualnej specyfikacji tego samego, abstrakcyjnego automatu współbieżnego opisanego zarówno diagramem maszyny stanów UML jak i skorelowanym z nim diagramem hierarchicznej sieci Petriego [2, 3]. W artykule pokazano sposób obsługi sytuacji wyjątkowych oraz ponownego wznowienia sterowania współbieżnego, hierarchicznego procesu dyskretnego przedstawionego modułową siecią Petriego. W literaturze przedmiotu dotyczącej sieci Petriego, brak jest spójnej i prostej metodologii wprowadzenia tego rodzaju funkcjonalności, znanej z teorii hierarchicznych map stanów wprowadzonej przez Harela i rozwiniętej w pracach Gajskiego [7, 8]. Dotychczasowe sposoby odwzorowania sytuacji wyjątkowych w sieciach Petriego, znane ze standardu UML, prowadzą do zerwania związku między prostotą odwzorowania algorytmu a jego sensownością i intuicyjnością. W pracy zaproponowano sposób obsługi sytuacji wyjątkowych polegający na wprowadzeniu do hierarchicznych sieci Petriego dodatkowych miejsc konfiguracyjnych (aktywnych i spoczynkowych). W przypadku prawidłowego funkcjonowania sterownika miejsca konfiguracyjne aktywnych modułów są oznakowane. W przypadku wystąpienia konieczności wstrzymania pracy układu, aktywne miejsca konfiguracyjne traci marker na rzecz spoczynkowych miejsc konfiguracyjnych. W przypadku wznowienia procesów po awarii, następuje powrót markerów do aktywnych miejsc konfiguracyjnych. W przypadku obsługi sytuacji krytycznej, konfiguracyjne miejsca aktywne przekazują marker do miejsca konfiguracyjnego wywłaszczającego. Przeprowadzane jest opróżnienie z markerów wszystkich miejsc sterujących w sieci a wznowienie pracy powoduje wprowadzenie jej do stanu początkowego.

Słowa kluczowe: hierarchiczna sieć Petriego, wywłaszczenie, rekonfigurowalny sterownik logiczny, UML.

Exception handling mechanism and control resumption in hierarchical Petri nets

Abstract

The description of programs for reconfigurable logic controllers is performed with use of dual specification [5, 6], concurrent abstract state machine described both with the UML state machine diagram and correlated hierarchical Petri net [1, 2, 3]. The paper shows the exceptions handling mechanism and control resumption of the hierarchical concurrent binary process described by means of a modular Petri net. In the literature on Petri nets (Fig. 2), there is no coherent and simple methodology for introduction of this kind of functionality, known from the theory of hierarchical map of states introduced by Harel and developed in the work by Gajski [7, 8]. The paper proposes how to handle exceptions based on introducing additional configurational places (active and idle) into hierarchical Petri nets (Fig. 3). In the case of having to stop operation of the system, the active configurational place loses its marker for the idle configurational place. In the case of resumption processes after the failure (Fig. 6), markers will return to the active configurational places. In the case of an emergency (critical exceptions - Fig. 7), the active configurational place moves a marker to the resumption configurational place. Evacuation of all markers in the control places of the network and resumption of the net work to its original state are carried out. In the

paper the possibility and desirability of implementing a new method of exceptions handling in a binary logic controller for productive is presented.

Keywords: hierarchical Petri net, expropriation, reconfigurable logic controller, UML.

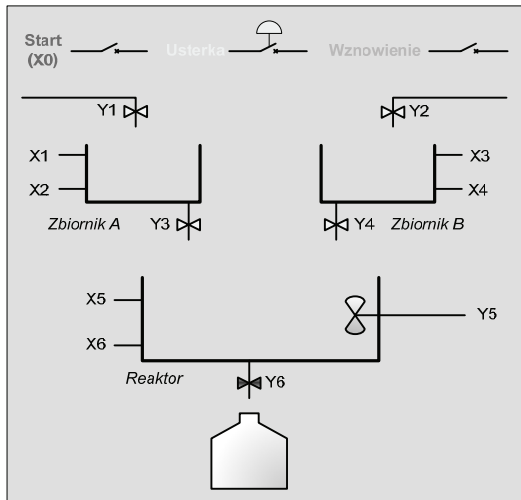
1. Wprowadzenie

Jedną z nowych metod specyfikacji behawioralnej systemów sterowania jest dualna specyfikacja [5, 6]. Składa się z dwóch komplementarnych i wzajemnie transformowalnych modeli: Maszyny stanów UML oraz hierarchicznej sieci Petriego. Model ten poprzez zastosowanie diagramu maszyny stanów UML pozwala na specyfikację sytuacji wyjątkowych. Wystąpienie sytuacji wyjątkowej o charakterze ostrzegawczym bądź krytycznym powoduje wywłaszczenie aktualnie aktywnego stanu i przekazanie sterowania (aktywacja stanu) do miejsca obsługi wyjątku. Mechanizm ten jest ujęty w standardzie języka UML. Drugi model składowy - hierarchiczna sieć Petriego - nie zawiera elementów które w sposób bezpośredni pozwalały by na obsługę sytuacji wyjątkowych. Może być ona realizowana poprzez wprowadzenie dodatkowych elementów w postaci miejsc zbierających żetony oraz tranzycji synchronizujących i scalających procesy współbieżne. Rozwiązanie nie jest efektywne pod względem wykorzystania zasobów sprzętowych na etapie implementacji sterownika. Mimo dużych zasobów sprzętowych układów FPGA, w których sterowniki mogą być implementowane, już na etapie specyfikacji sterownika winno się zwracać uwagę na wykorzystanie takich elementów jak przerzutniki czy też tablice LUT realizujące logikę kombinacyjną [1]. Znany z diagramów maszyny stanów UML mechanizm wznowienia nie ma swojego bezpośredniego odpowiednika w modelu hierarchicznej sieci Petriego. Podjęte próby odwzorowania mechanizmu wznowienia w hierarchicznej współbieżnej maszynie stanów HCFM powodują duże marnotrawstwo zasobów sprzętowych: każdy stan posiada kopię która przechowuje informację gdy maszyna stanów do której należy jest wywłaszczana [4].

W artykule przedstawiono propozycję odwzorowania mechanizmu obsługi sytuacji wyjątkowych zarówno o charakterze ostrzegawczym jak i krytycznym z szczególnym uwzględnieniem mechanizmu wznowienia.

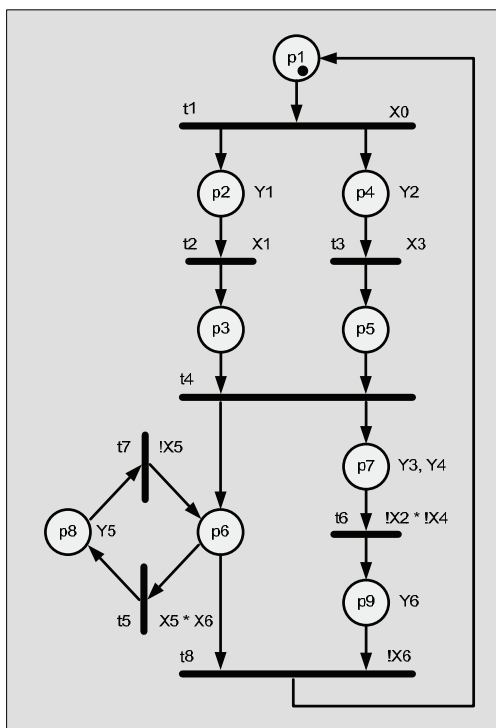
2. Specyfikacja sterownika z wykorzystaniem sieci Petriego

Przykładowy proces produkcyjny (rys. 1) polegający na odmierzeniu odpowiednich ilości dwóch płynnych substratów a następnie połączeniu ich w kadzi reaktora. Produkt konfekcjonowany jest do zbiornika umieszczonego pod zaworem spustowym Y6. W przypadku wykrycia piany przez czujnik X5 następuje włączenie mieszadła Y5.



Rys. 1. Proces produkcyjny z obsługą sytuacji awaryjnych
Fig. 1. Productive process with exceptions handling

Sterowanie procesem może być za pomocą sterownika logicznego, którego specyfikację stanowi sieć Petriego, przedstawiona na rys. 2. Stanowi ona blok operacyjny sterownika konfigurowalnego.



Rys. 2. Uproszczona sieć Petriego bez obsługi sytuacji awaryjnych
Fig. 2. Simplified Petri net without exceptions handling

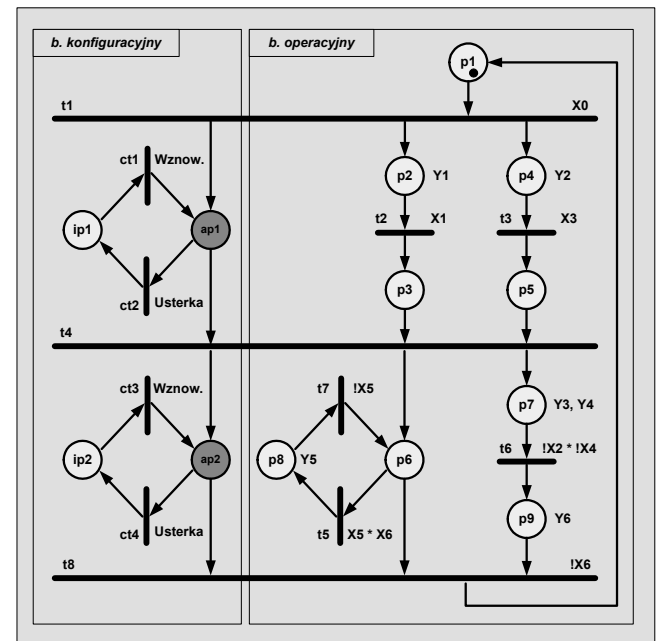
Przedstawiona sieć nie posiada mechanizmu obsługi sytuacji awaryjnych. Sygnały Usterka oraz Wznowienie nie warunkują wykonania zadanej z tranzycji. Tak przedstawiony sterownik nie realizuje w pełni postawionych przed nim zadań. Nie ma możliwości zatrzymania ani wstrzymania procesu produkcyjnego, dlatego też takie rozwiązanie jest niedopuszczalne.

3. Obsługa wyjątków o charakterze ostrzegawczym

Powstanie sytuacji awaryjnej (wyjątku) pociąga za sobą konieczność odpowiedniej reakcji. Proponowana metoda obsługi wyjątków rozdziela blok operacyjny sterownika realizujący właściwą funkcję sterowania procesem przemysłowym od bloku

konfiguracyjnego odpowiedzialnego za nadzór nad funkcjonowaniem sieci.

Realizacja mechanizmu obsługi wyjątków opiera się na wprowadzeniu dodatkowych miejsc i tranzycji o szczególnym charakterze. Są to miejsca konfiguracyjne: aktywne *ap1*, *ap2* (ang. *Active Place*) oraz spoczynkowe *ip1*, *ip2* (ang. *Idle Place*). Dla każdego z etapów została dodana para miejsc spięta tranzycjami: zamrażającymi *ct2*, *ct4*, oraz wznowiającymi *ct1*, *ct2*.



Rys. 3. Sieć Petriego z wyodrębnionym blokiem konfiguracyjnym
Fig. 3. Petri net with highlighted configurational block

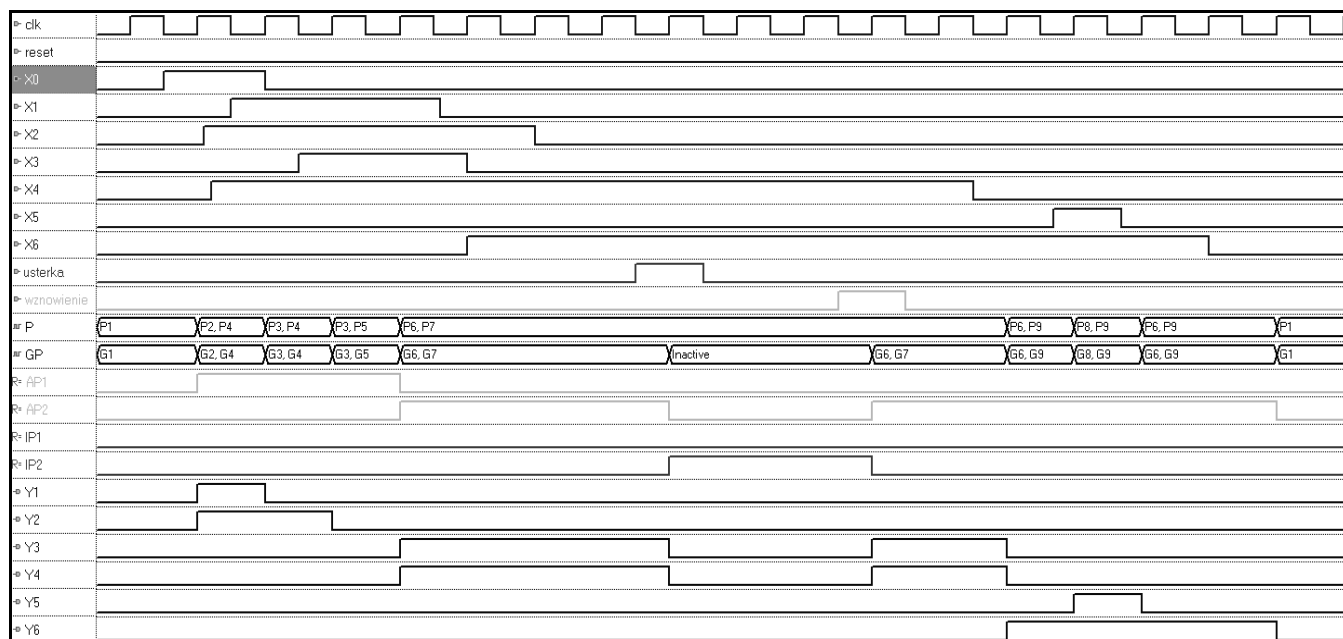
Sieć Petriego uzupełniona o miejsca konfiguracyjne z wyodrębnionymi blokami: operacyjnym i konfiguracyjnym, została przedstawiona na rys. 3.

Miejsca konfiguracyjne aktywne mają szczególny charakter. Gdy są oznakowane, możliwa jest realizacja tranzycji oraz generowanie sygnałów wyjściowych w odpowiadających im podsięciach. Przykładowo miejsce konfiguracyjne *ap1* warunkuje realizację tranzycji *t2*, *t3*, *t4* oraz generowanie sygnałów *Y1*, *Y3*. Miejsce *ap2* warunkuje realizację tranzycji *t5*, *t6*, *t7*, *t8* oraz generowanie sygnałów *Y3*, *Y4*, *Y5*, *Y6*.

W przypadku wystąpienia sytuacji awaryjnej funkcjonowanie bloku operacyjnego zostaje wstrzymane (zamrożone). Realizacja tranzycji wywłaszczających powoduje przejście markera z miejsca konfiguracyjnego aktywnego do miejsca konfiguracyjnego spoczynkowego. Usunięcie awarii pozwala na powrót markera do miejsca konfiguracyjnego i wznowienie sterowania.

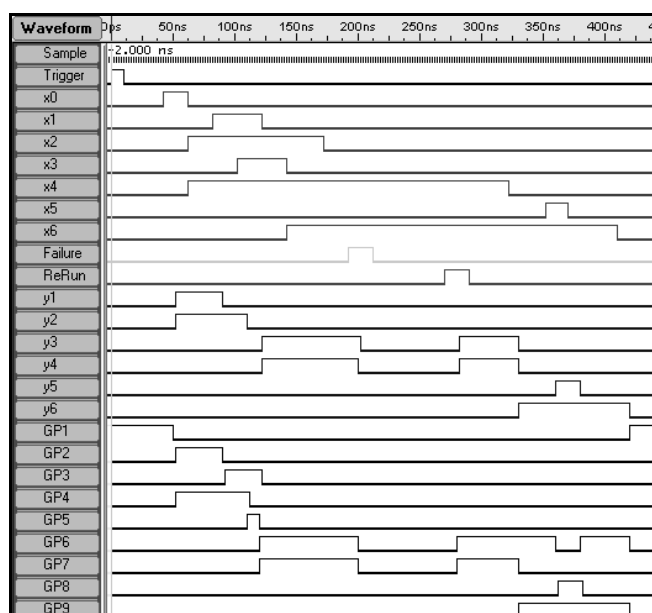
Symulacja sterownika (rys. 4) ukazuje mechanizm zamrażania stanu. Stany lokalne oznaczone są na przebiegu symulacji prefiksem *P* (rys. 4, 8), stany globalne prefiksem *GP* (rys. 4, 5). W drugim etapie produkcyjnym dochodzi do sytuacji awaryjnej - sygnał *Usterka* przyjmuje wartość 1. Miejsca *P6* i *P7* zostają zamrożone a generowanie sygnałów wyjściowych *Y3* i *Y4* wstrzymane. W celu lepszego zobrazowania wprowadzono sygnały globalnej konfiguracji *GP*. Ponieważ wyjątek miał jedynie charakter usterki pojawienie się sygnału *Wznowienie* powoła na kontynuację procesu od miejsca, w którym został on wstrzymany.

Specyfikacja posłużyła do syntezy a następnie implementacji układowej sterownika. Do tego celu wykorzystano zestaw uruchomieniowy Xilinx XUP Virtex-II Pro Development System. Model sterownika został opisany w języku Verilog, a następnie poddany syntezy i implementacji układowej. Przeprowadzone testy potwierdziły prawidłowe funkcjonowanie prototypu. Weryfikacji wykonano z wykorzystaniem analizatora stanów logicznych firmy Tektronix (rys. 5).



Rys. 4. Symulacja sterownika w środowisku Active HDL

Fig. 4. Simulation of the controller in the Active HDL environment



Rys. 5. Wyniki weryfikacji prototypu sterownika z obsługą wyjątków i wznowienia

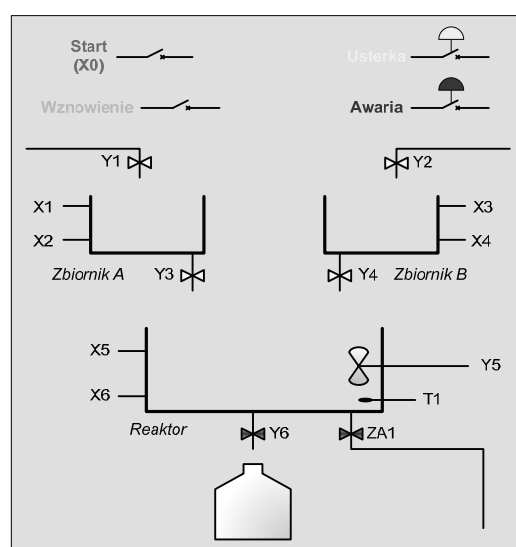
Fig. 5. Verification results of the controller with exceptions handling and resumption

4. Obsługa wyjątków krytycznych

System sterowania przedstawiony na rys. 1 w celu spełnienia bardziej rygorystycznych wymagań bezpieczeństwa może być rozszerzony obsługą wyjątków o charakterze krytycznym. Wyjątek taki może być wygenerowany z konsoli operatora (przycisk awaria), bądź pośrednio przez zewnętrzny moduł kontroli temperatury co jest istotne w przypadku reakcji egzotermicznych (rys. 6).

Wyjątek o charakterze krytycznym powoduje wstrzymanie pracy układu (bloku operacyjnego) i przekazanie sterowania do układu nadrzędnego. Tam np. może być podjęta decyzja o konieczności awaryjnego opróżnienia kadzi reaktora (zawór ZA1). W przypadku wyjątków krytycznych wznowienie pracy układu nie jest możliwe. Wyłączenie miejsc z tokenów ma charakter perma-

nentny. Obsługa wyłączenia o charakterze krytycznym realizowana jest poprzez dodanie do sieci kolejnego miejsca konfigurowanej sieci, która w przypadku wystąpienia wyjątku krytycznego Awaria wyłączy miejsca operacyjne P6, P7, P8, P9, poprzez skasowanie tokenów w nich zawartych. Na poziomie realizacji układowej w postaci automatu sekwencyjnego odbywa się to poprzez zresetowanie przetrzutników odpowiadających wyłączanym miejscom.

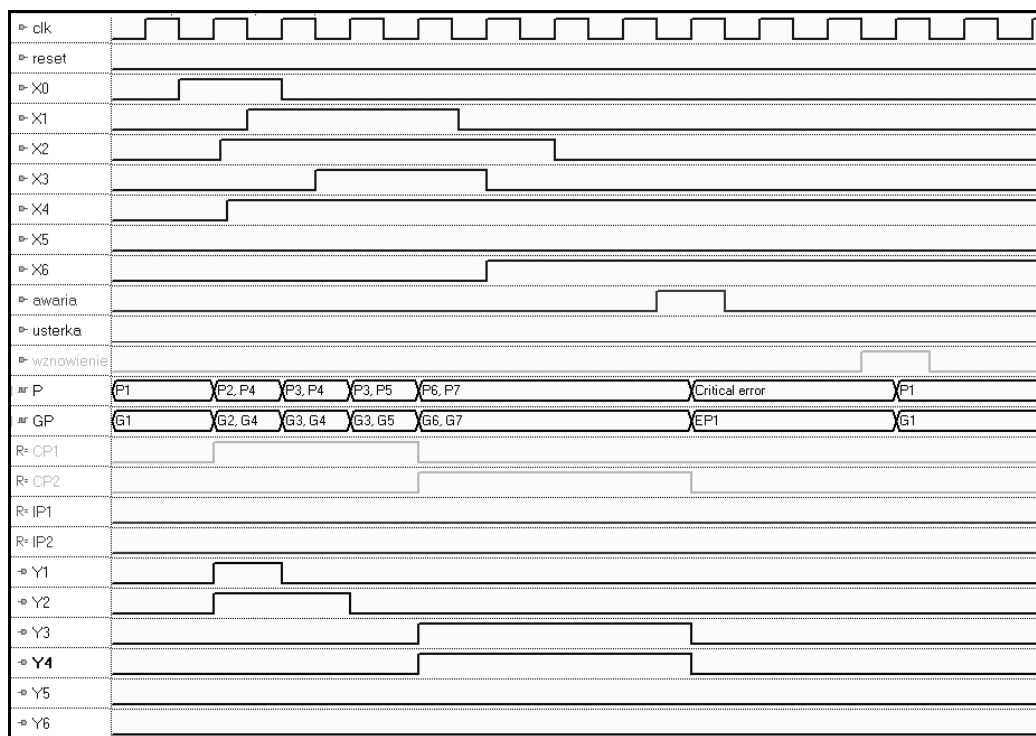


Rys. 6. Schemat procesu przemysłowego z pomiarem temperatury reaktora i awaryjnym zrzutem

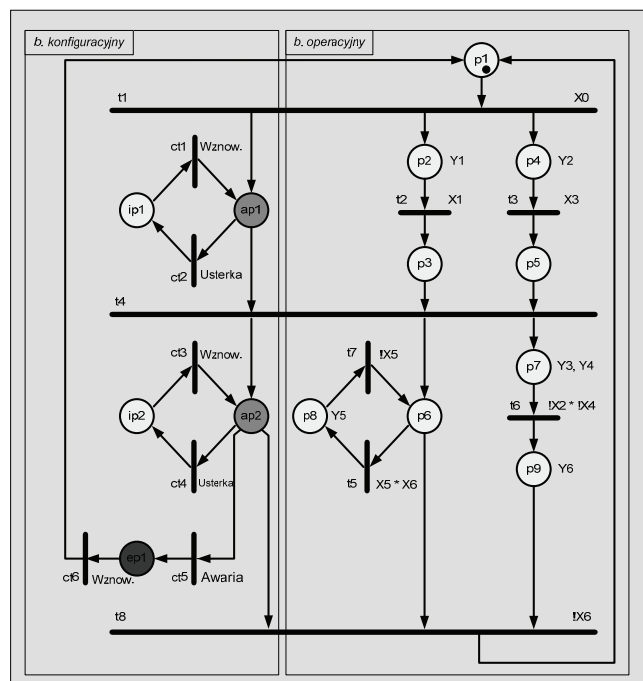
Fig. 6. Productive process with temperature measurement and emergency dumping schema

Sterowanie może zostać przekazane do sieci nadrzędnej, gdzie po usunięciu awarii i oczyszczeniu kadzi reaktora zostanie wygenerowany sygnał Wznowienie. Praca układu może zostać przywrócona a substraty znajdujące się w zbiornikach pomocniczych nie ulegną zniszczeniu.

Wyłączenie miejsc P6 i P7 następuje w chwili znakowania miejsca EPI (rys. 8).



Rys. 8. Symulacja sterownika z obsługą wyjątków krytycznych w środowisku Active HDL
Fig. 8. Simulation of the controller critical exception handling in the Active HDL environment



Rys. 7. Schemat procesu sterowania z obsługą wyjątków o charakterze krytycznym
Fig. 7. Control process with critical exception handling

5. Podsumowanie

W artykule omówiono metodę obsługi sytuacji wyjątkowych w Hierarchicznych sieciach Petriego. Jest ona niezwykle istotna gdy układ projektowany jest w oparciu o dualną specyfikację. Hierarchiczna sieć Petriego rozszerzona o mechanizm obsługi sytuacji wyjątkowych umożliwia wzajemną konwersję do diagramu maszyny stanów UML. Przedstawiona metoda może być również wykorzystana gdy diagram Hierarchicznej sieci Petriego jest podstawową formą specyfikacji sterownika. Wystąpienie

sytuacji wyjątkowej powoduje zamrożenie bądź wyłączenie sieci i przekazanie sterownika do sieci nadrzędnej (nadzorcy). Hierarchiczna sieć Petriego rozszerzona o mechanizm obsługi sytuacji wyjątkowych może być przekształcana do diagramu maszyny stanów UML.

Praca naukowa finansowana ze środków budżetowych na naukę w latach 2010 - 2013 jako projekt badawczy własny Nr N N516 513939.

6. Literatura

- [1] Adamski M., Karatkevich A., Węgrzyn M. (red): Design of embedded control systems, Springer, New York 2005.
- [2] Andreu D., Souquet G., Gil T.: Petri Net Based Rapid Prototyping of Digital Complex System, IEEE Computer Society Annual Symposium on VLSI, IEEE, 2008.
- [3] Basile F., Chiacchio P., Del Grosso D.: Modelling automation systems by UML and Petri Nets, Proceedings of the 9th International Workshop on Discrete Event Systems, Gooteborg, IEEE, 2008.
- [4] Bazydło G., Adamski M.: Obsługa wyjątków w maszynie stanowej UML realizowanej w mikrosystemach cyfrowych, Pomiary Automatyka Kontrola, 2010, Vol. 56, nr 7, s. 728-731.
- [5] Doligalski M., Adamski M.: Obsługa wyjątków oraz stanów wznowienia w ramach dualnej specyfikacji, Przegląd Elektrotechniczny, 2010, nr 9, s. 123-125.
- [6] Doligalski M., Adamski M.: Specyfikacja sterowników cyfrowych zorientowana na niezawodność, Pomiary Automatyka Kontrola, 2010, Vol. 56, nr 7, s. 671-674.
- [7] Gajski D. D., Wahid F., Narayan S., Gong J.: Specification and Design of Embedded Systems, P T R Prentice Hall, New Jersey 1994.
- [8] Łabiak G.: Wykorzystanie hierarchicznego modelu współbieżnego automatu w projektowaniu sterowników cyfrowych, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra 2005.