Marek SAŁAMAJ, Krzysztof KUJAWA
UNIWERSYTET ZIELONOGÓRSKI, WYDZIAŁ MECHANICZNY,
ul. Prof. Z. Szafrana 4, 65-516 Zielona Góra

Bezpieczny Mikrosterownik Logiczny dla systemów krytycznych

Dr inż. Marek SAŁAMAJ

Ukończył studia magisterskie w 2001 roku na Wydziale Elektrycznym Politechniki Zielonogórskiej o specjalności inżynieria systemów komputerowych. W 2010 roku obronił rozprawę doktorską na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego w dyscyplinie informatyka. Tematyka badań naukowych: bezpieczne sterowanie systemami krytycznymi, projektowanie układów o podwyższonym ryzyku działania, metody podwyższania poziomu bezawaryjności projektów realizowanych w układach FPGA.

e-mail: M.Salamaj@iizp.uz.zgora.pl

Mgr inż. Krzysztof KUJAWA

Ukończył studia magisterskie w 2005 roku na Wydziale Mechanicznym Uniwersytetu Zielonogórskiego o specjalności technologia maszyn. W 2011 roku otworzył przewód doktorski na temat „Badania nad przebiegiem skurczu liniowego siluminów na Wydziale Mechanicznym Uniwersytetu Zielonogórskiego w dyscyplinie mechanika i budowa maszyn. Tematyka badań naukowych: badania przebiegu krzepnięcia odlewu w formie.

e-mail: K.Kujawa@iizp.uz.zgora.pl

Streszczenie

Artykuł przedstawia tematykę związaną z projektowaniem nowego układu bezpiecznego, który w przyszłości zdolny byłby nie tylko do bezpiecznego i precyzyjnego zarządzania określonymi systemami krytycznymi ale również rozwiązaniami uniwersalnymi. Prezentowane wyniki badań obejmują propozycję nowej koncepcji układowej Bezpiecznego Mikrosterownika Logicznego (BML) w zakresie dywersyfikacji jego rozwiązań, które przeprowadzono z użyciem jak najprostszych mechanizmów oraz rozwiązań technicznych. Tego typu podejście pozwoliło na otrzymanie prostego i bezpiecznego mikrosystemu decyzyjno-sterującego, w którym zastosowane mechanizmy oraz rozwiązania zwiększyły bezawaryjność oraz niezawodność jego funkcjonowania. W rezultacie, tak zaproponowana jednostka sterująca okazała się rozwiązaniem jak najbardziej uniwersalnym, która może być stosowana do zarządzania dowolnym systemem krytycznym jak również (w miarę postępu technologicznego przy jednoczesnym spadku cen rozwiązań technicznych) dowolnym uniwersalnym systemem czasu rzeczywistego.

Słowa kluczowe: bezpieczeństwo, bezawaryjność, sterownik, mikrosterownik, architektura układów, dywersyfikacja układów.

Safety Logic Microcontroller for critical systems

Abstract

The article presents the issues related to the design of a new safety unit, which in the future would not only be able to manage certain critical systems safely and accurately but also universal solutions. The research results presented include a proposal for a new systemic concept of a Safety Logical Microcontroller (BML), concerning the diversification of its solutions, which was carried out using the simplest mechanisms and technical solutions. This approach allowed to obtain a simple and safe decision-making and control microsystem, in which the applied mechanisms and solutions increase reliability of its operation. As a result, the proposed control unit has proved to be the very universal solution, which can be used to manage any critical system as well as (as technology advances with the simultaneous decrease in the price of technical solutions) any universal real-time system.

Keywords: safety, uninterrupted, controller, microcontroller, hardware, system architecture, unit diversification.

1. Wstęp

Dynamiczny rozwój nowych technologii oraz robotyzacji i automatyzacji wielu dziedzin życia codziennego, jak i przemysłu sprawia, że układy decyzyjno-sterujące znajdują coraz szersze zastosowanie w zarządzaniu różnego rodzaju urządzeniami, obiektami bądź procesami technicznymi. Najczęściej, tego typu układy reprezentowane są przez różnego rodzaju sterowniki, a nawet ze względu na niewielkie rozmiary mikrosterowniki logiczne, które głównie różnią się od siebie złożonością struktury oraz funkcjonalnością. Tego typu układy projektowane są z użyciem specjalistycznych narzędzi wspieranych rozwiązaniami typu CAD, a następnie fizycznie realizowane z bardzo dużą dokładnością. Pomimo stosowania procesu projektowania wspieranego przez CAD można zweryfikować oraz zdiagnozować wiele różnych błędów konstrukcyjnych lub projektowych tworzonego układu sterującego i je wyeliminować. Natomiast, niemożliwe jest już przewidzenie jakie błędy przypadkowe bądź losowe (w docelowym miejscu zastosowania w przyszłości) mogą się w nim pojawić, a następnie doprowadzić do ewentualnego zagrożenia życia człowieka lub zanieczyszczenia środowiska. Dlatego, systemy czasu rzeczywistego, a dokładniej krytyczne systemy czasu rzeczywistego [1, 2, 3, 4] coraz częściej wyposaża się w bezpieczne rozwiązania (układy) decyzyjno-sterujące, które precyzyjnie kontrolują własne działania, jak również działania sterowanych nimi obiektów. O ile błędy projektowe tych układów można w bardzo prosty skuteczny sposób diagnozować i eliminować, to błędów przypadkowych lub losowych już niestety nie. W tym przypadku, błędy przypadkowe lub losowe [1, 2] należy rozpatrywać jako efekt zdarzeń, które mogą pojawić się w dowolnym momencie pracy układu, wystąpienia których nie jesteśmy w stanie uniknąć, a jedynie zabezpieczyć się przed ich skutkami. Do tej grupy błędów można zaliczyć między innymi błędy układu wywołane np. promieniowaniem kosmicznym (neutronowym) modyfikującym zawartość pamięci, przypadkowym uszkodzeniem linii zasilającej, fizycznym uszkodzeniem całego układu sterującego lub jego części, czy zakłóceniami generowanymi przez inne urządzenia techniczne, itp. Niestety, tego typu błędów nie da się zamodelować na etapie tworzenia (projektowania) i fizycznej realizacji prototypowych bezpiecznych rozwiązań sterujących, ale logicznie myśląc należy stwierdzić, że mogą się one w nich uwidocznić dopiero w już działającym systemie (urządzeniu, układzie) - niekiedy nawet po wielu latach od momentu jego wdrożenia i uruchomienia. W związku z tym, rozwiązania bezpieczne układów sterujących powinny i są też tak projektowane, aby bez względu na ich przyszłe warunki otoczenia (w miejscu ich docelowego zastosowania) były zdolne do wykrycia, zidentyfikowania oraz do właściwego obsłużenia jak największej liczby błędów, których nie jest się w stanie przewidzieć, a tym bardziej rozpatrzyć na etapie tworzenia tego typu rozwiązań sterujących.

2. Bezpieczeństwo

Poruszając zagadnienia związane z bezpieczeństwem oraz zarządzaniem procesami technologicznymi bardzo często mamy na myśli krytyczne systemy czasu rzeczywistego. Wojskowym lub cywilnym krytycznym systemem czasu rzeczywistego określamy system czasu rzeczywistego, którego elementy spełniają krytyczne wymogi stawiane rozwiązaniom bezpiecznym [5]. Jakikolwiek konflikt lub awaria w tych systemach może doprowadzić do wielu przykrych następstw, między innymi do narażenia życia ludzkiego w jego zasięgu lub do powstania strat materialnych w przemyśle ale i nie tylko. Systemy te budowane są najczęściej z jednostki centralnej (procesora, mikroprocesora, serwera, itp.) oraz z różnego rodzaju urządzeń peryferyjnych wyposażonych w czujniki oraz

w moduły wykonawcze. O ile układy peryferyjne są prostymi w budowie oraz zasadzie działania urządzeniami, to jednostka centralna (np. sterownik) wręcz przeciwnie, już niestety nie. Do głównych zadań tego typu jednostki należy prawidłowe i precyzyjne zarządzanie całym systemem, analizowanie i interpretowanie sygnałów odpowiedzialnych za stan pracy nadzorowanych obiektów, sterowanie urządzeniami wykonawczymi, a ponadto musi wykrywać rożnego rodzaju konflikty, usterki i anomalie w funkcjonującym systemie i właściwie na nie reagować. Wynika z tego, że dowolny układ sterujący (zarządzający) tego typu systemem czasu rzeczywistego powinien charakteryzować się jak najprostszą budową oraz zasadą działania, gdyż wówczas przewidywalność jego działań doprowadziłaby do dużo efektywniejszej kontroli, a tym samym do bezawaryjnej pracy całego rozwiązania.

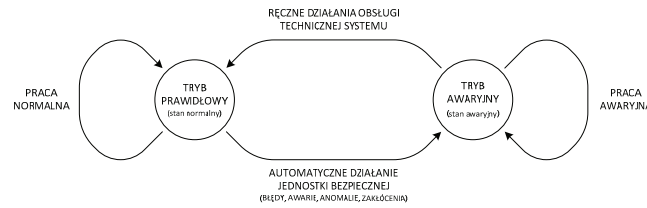
W zakresie projektowania rozwiązań bezpiecznych, na samym wstępie należy zaznaczyć, że zaproponowanie oraz wykonanie bezwzględnie bezpiecznego rozwiązania (układu, sterownika, mikrosterownika, urządzenia, itp.) sterującego nigdy nie było i nie będzie możliwe do zrealizowania. Stan bezwzględnego bezpieczeństwa [1, 2, 6], a tym bardziej stan całkowitej bezawaryjności układu jest niemożliwy do osiągnięcia z punktu widzenia technologii jego wykonania, ponieważ w realnym otoczeniu (zastosowaniu) zawsze może wystąpić możliwość niepożądanego zakłócenia jego pracy. Natomiast, realizacja układów bezpiecznych odmiennymi metodami oraz narzędziami, które wspierane są najnowszymi technologiami i rozwiązaniami typu CAD, pozwala jedynie na precyzyjne ich wykonanie, ale już nie na zabezpieczenie przed błędami w funkcjonowaniu. Dlatego, projektując tego typu rozwiązania należy wyposażać je w rozwiązania, które skutecznie zabezpieczyłyby je przed różnego rodzaju błędami. Niestety, tak przyjęty tok postępowania podczas projektowania tych układów skutkuje ich nadmierną rozbudową, a co za tym idzie prowadzi do nieprzewidzianych i niekontrolowanych zachowań w funkcjonowaniu. W związku z tym, aby rozwiązanie było bezpieczne to musi ono być możliwie jak najprostsze pod względem budowy i zasady działania [1], ale jednocześnie powinno być wyposażone w jak największą liczbę różnego rodzaju rozwiązań, które skutecznie zabezpieczyłyby je przed ewentualnymi błędami z zewnątrz. Pomimo takiego podejścia wciąż nie będzie gwarancji na zaprojektowanie układu bezwzględnie bezpiecznego, a jedynie układu funkcjonującego na dużo wyższym poziomie bezpieczeństwa i bezawaryjności w stosunku do pozostałych rozwiązań.

3. Bezpieczny Mikrosterownik Logiczny

Jednym z przykładów układu sterującego, który można zastosować do zarządzania krytycznymi systemami czasu rzeczywistego jest opracowany na bazie innych rozwiązań bezpiecznych [3, 6], a następnie zrealizowany Bezpieczny Mikrosterownik Logiczny (BML). Układ BML został zaprojektowany z myślą o rekonfigurowalnych układach FPGA, które pozwalają na szybkie i precyzyjne prototypowanie realizowanych projektów. W ten sposób, przeanalizowano wiele różnych rozwiązań z zakresu techniki, które po zweryfikowaniu zostały zaadoptowane i zaimplementowane w tak realizowanym układzie (BML). Projektując tego typu układ starano się nadać mu możliwie jak największą funkcyjność oraz wyposażyć go w różnego rodzaju mechanizmy zabezpieczające, które zabezpieczyłyby go na ewentualność wystąpienia w nim następujących błędów:

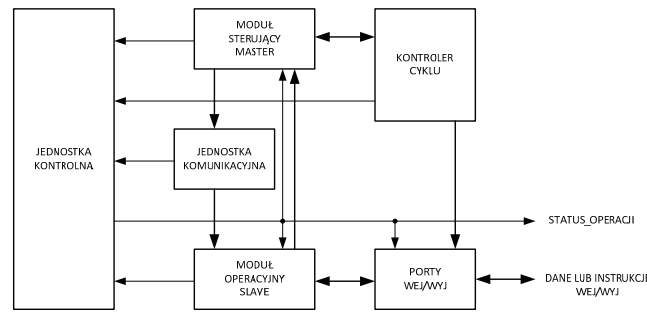
- błędów konstrukcyjnych, które najczęściej związane są z budową samego rozwiązania,
- błędów przetwarzania danych związanych z nieprecyzyjną analizą oraz weryfikacją przetwarzanych informacji na poziomie algorytmu działania gotowego rozwiązania,
- błędów przypadkowych lub losowych mogących być przyczyną nieprzewidzianych na etapie projektowania działań wywołanych przez czynniki zewnętrzne (np. wadliwie wyprodukowana seria elementów składowych danego rozwiązania lub oddziaływanie na dane rozwiązanie przez osoby postronne),
- błędów spowodowanych działaniem czynników zewnętrznych, jak różnego rodzaju zakłócenia (np. promieniowanie kosmiczne zwane również neutronowym, które w określonych sytuacjach może modyfikować zawartość pamięci wypełnionej informacjami w systemie binarnym.

W związku z powyższym, bezpieczeństwo jednostki sterującej należy rozumieć jako jej bezawaryjne funkcjonowanie (działanie) [1, 6], które otrzymuje się poprzez zastosowanie w niej różnego rodzaju mechanizmów oraz rozwiązań technicznych znacznie podwyższających poziom bezawaryjności ich pracy. Dlatego, wspomniany układ BML traktowany jest jako układ sterujący, którego praca może znaleźć się w jednym z dwóch możliwych trybów jego działania: w trybie awaryjnym lub w trybie prawidłowym – rys. 1.



Rys. 1. Dwa różne stany pracy układu BML
Fig. 1. Two different states of work of the BML unit

W tym przypadku, tryb prawidłowy BML szczegółowo określa wszelkie pożądane schematy zachowań tego układu na skutek zmian stanów logicznych, które zachodzą na jego wejściach. Wspomniane schematy zachowań początkowo definiowane są na poziomie założeń projektowych, a dopiero później odzwierciedlane w sprzęcie i oprogramowaniu realizowanego rozwiązania sterującego. W sytuacji, gdy układ BML samoczynnie diagnozując swoje działanie wykryje błąd w funkcjonowaniu to jego praca natychmiast zostaje przeniesiona z trybu (stanu pracy) prawidłowego do trybu awaryjnego – tzw. stanu bezpiecznego [1, 2, 6]. Stan bezpieczny w urządzeniu (BML) jest to stan, w którym w ekstremalnej sytuacji (w momencie wykrycia błędu) wszystkie jego wyjścia ustawiane są w ściśle określony stan logiczny zależny od zarządzanego nim krytycznego systemu czas rzeczywistego co uniemożliwi zagrożenie życia ludzkiego lub powstaniu strat materialnych w produkcji. Przykładowo, stanem bezpiecznym dla układu zarządzającego systemem sygnalizacji świetlnej na skrzyżowaniu ulic będzie stan, w którym na wyjściach układu podczas awarii systemu pojawią się sygnały sterujące wymuszające na elementach wykonawczych działanie w postaci pulsującego żółtego światła.



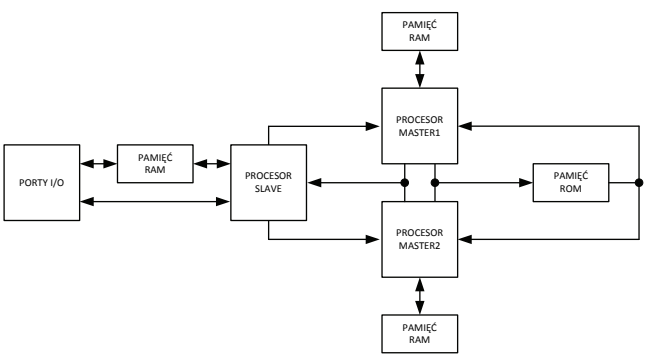
Rys. 2. Schemat blokowy koncepcji układu BML
Fig. 2. The block diagram of a BML conception

Tego typu rozgraniczenie funkcjonalności układu BML na stan pracy prawidłowej (zgodny z założeniami projektowymi) oraz na stan krytyczny (bezpieczny, awaryjny) pozwoliło na zaadoptowanie w nim dywersyfikacji [1] rozwiązań, która polega głównie na tym, aby to samo rozwiązanie realizować całkowicie odmiennymi metodami, narzędziami oraz sposobami. Wprowadzenie dywersyfikacji podczas realizacji układu BML oraz dwóch odmiennych trybów jego pracy pozwoliło na skoncentrowaniu uwagi na rozwiązaniach,

które ostatecznie miały podwyższyć poziom bezpieczeństwa oraz bezawaryjności pracy układu sterującego oraz zarządzanego nim całego krytycznego systemu czasu rzeczywistego.

4. Architektura układu BML

Badania w zakresie realizacji układów bezpiecznych pozwoliły na zaproponowanie nowej koncepcji architektury układu sterującego (rys. 2) - Bezpiecznego Mikrosterownika Logicznego (BML) [1, 2]. Układ BML zaprojektowano jako 32-bitową jednostkę sterującą do zarządzania systemami krytycznymi czasu rzeczywistego, której trój-procesorową architekturę wraz ze wszystkimi niezbędnymi modułami funkcyjnymi w całości zaimplementowano w jednej strukturze reprogramowalnej FPGA – rys. 3. W architekturze tej wyszczególnić można trzy niezależne od siebie lokalne obszary, które globalnie precyzyjnie współpracują ze sobą. W obrębie obszarów lokalnych architektury BML można wyróżnić procesory sterujące MASTER1 oraz MASTER2 jak również procesor operacyjny SLAVE wraz ze wszystkimi niezbędnymi do ich funkcjonowania układami. W tym przypadku założono, że oba procesory MASTER będą pełniły w układzie BML funkcję procesorów sterujących, zaś procesor SLAVE funkcję koprocesora matematycznego wspierającego i odciążającego działania procesorów sterujących. Dywersyfikacja rozwiązań przystosowywanych do użycia w nowej koncepcji BML uwidoczniła się w momencie dywersyfikacji potoku sterowania w procesorach MASTER1 i MASTER2 oraz podczas dywersyfikacji potoku obliczeniowego w procesorze SLAVE. Potok sterowania rozbito na dwa niezależne współbieżne strumienie, w których wykonuje się dokładnie te same działania i operacje, ale całkowicie odmiennymi metodami, rozwiązaniami oraz sposobami. W podobny sposób zdywersyfikowano potok obliczeniowy w procesorze SLAVE, w którym wyszczególniono już trzy niezależne ale współbieżne względem siebie strumienie obliczeniowe. W rezultacie, bezpośrednia i nieustanna wymiana oraz weryfikacja wszelkich informacji generowanych we wszystkich strumieniach zarówno w potoku sterującym, jak i w obliczeniowym sprawiła, że otrzymano precyzyjnie funkcjonujący mechanizm, który pozwala w bardzo prosty sposób na wykrywanie wszelkiego rodzaju błędów oraz awarii.

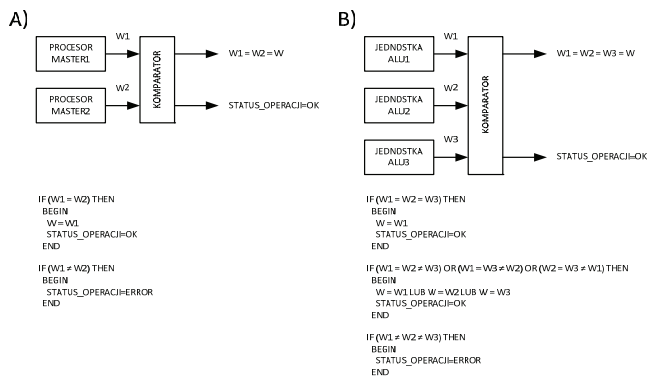


Rys. 3. Uproszczona architektura układu BML
Fig. 3. The architecture of a BML unit

Projektując tego typu jednostkę (BML) starano się zaadoptować w jej architekturze konkretne rozwiązania oraz mechanizmy, które ostatecznie dość znacznie podwyższyłyby poziom bezpieczeństwa i niezawodności jej pracy. Dlatego, podczas realizacji układu BML starano się, aby układ ten pod względem budowy oraz zasady działania był jak najprostszym rozwiązaniem, ale jednocześnie wyposażony w różnego rodzaju rozwiązania zabezpieczające. Wydaje się, że im więcej rozwiązań zabezpieczających jest zaimplementowanych w tego typu rozwiązaniu to tym bezpieczniejszą i bezawaryjnie działającą jednostkę otrzymamy. Jednak w sytuacji, gdy nawet najprostszy układ sterujący zostanie wyposażony w różnego rodzaju zabezpieczenia to wtedy będzie on nieprzewidywalny w swoim działaniu, a to oznacza, że nie będzie możliwe do przewidzenia jego zachowanie w krytycznej sytuacji. W związku z tym, podczas realizacji układu BML doprowadzono do pewnego rodzaju kompromisu pomiędzy jego złożonością, a funkcjonalnością i bezawaryjnością.

5. Dywersyfikacja układu BML

Proponując nową koncepcję architektury układu BML starano się w miarę możliwości dywersyfikować względem siebie te rozwiązania, które mogłyby być kluczowe dla bezpiecznego i bezawaryjnego jego funkcjonowania. Dlatego, tak jak już wcześniej wspomniano w architekturze BML wyszczególniono trzy niezależne od siebie obszary obliczeniowe, które precyzyjnie współpracują ze sobą podczas realizacji określonego algorytmu do sterowania obiektem (systemem) krytycznym. Uogólniając, każdy tego typu obszar obliczeniowy składa się z procesora realizującego określoną funkcję oraz z modułów pamięci, do których nieustannie on się odwołuje. Zgodnie z ideą dywersyfikacji założono, że każdy tego typu obszar (szczególnie oba procesory MASTER) powinien zostać przygotowany przez niezależne grupy inżynierów, którzy podczas projektowania nie komunikowaliby się ze sobą, a wszelkie zadania realizowaliby różnymi metodami i sposobami wyłącznie na podstawie wytycznych (założeń) projektowych. W ten sam sposób, dywersyfikacji należałoby poddać procesor operacyjny SLAVE, jednak w architekturze BML przewidziano tylko jedną tego typu jednostkę obliczeniową – dywersyfikację dwóch procesorów SLAVE w swoim rozwiązaniu dokonali Halang-Śnieżek [3, 6]. Redukując liczbę procesorów z dwóch do jednego w bardzo dużym stopniu uproszczono budowę oraz zasadę działania układu bezpiecznego ale niejako pozbawiono się możliwości dywersyfikacji potoku obliczeniowego. W związku z tym dywersyfikacji rozwiązania dokonano wewnątrz procesora SLAVE, na poziomie przetwarzania informacji w jednostce ALU. Podobnie jak to miało miejsce podczas dywersyfikacji sterowania, teraz w potoku obliczeniowym procesora SLAVE wyodrębniono trzy niezależne od siebie strumienie, które zgodnie z ideą dywersyfikacji i założeniami projektowymi powinny być zrealizowane odmiennymi metodami i sposobami przez niezależne grupy badawcze (inżynierów, specjalistów). Realizując tego typu koncepcję bezpiecznego układu sterującego w rekonfigurowalnej strukturze typu FPGA zdecydowano się na zaproponowane powyżej sposoby dywersyfikacji architektury BML, gdyż w ten sposób zamodelowano w nim wszystkie te elementy, które pozwoliły na szczegółowe przeanalizowanie w nim różnych rozwiązań układowych. Między innymi analizie poddano korzyści z zastosowania podwójnej dywersyfikacji potoku sterującego w porównaniu z potrójną dywersyfikacją strumienia obliczeniowego.



Rys. 4. Dywersyfikacja rozwiązania: a) podwójna, b) potrójna
Fig. 4. Diversification of solution: a) double, b) triple

Porównując ze sobą oba warianty dywersyfikacji podwójnej z potrójną w układzie BML stwierdzono korzyści, które wynikają ze zwiększenia dywersyfikowanego rozwiązania z dwóch do trzech kanałów. W przypadku układu BML, jako kanał należy rozumieć pojedynczy potok sterujący z udziałem procesora MASTER1 lub MASTER2 oraz jako pojedynczy strumień obli-

czeniowy wewnątrz procesora operacyjnego SLAVE. Wnioski jakie wyciągnięto w tym przypadku, pozwoliły na realizację pierwszego mechanizmu, który jedynie wykrywał awarię (błąd) jednostki sterującej oraz drugiego mechanizmu realizującego dokładnie to samo zadanie ale dodatkowo identyfikującego rodzaj wykrytego błędu. Oba sposoby omawianej dywersyfikacji przedstawiono na rys. 4. Na rys. 4a przedstawiono wszystkie warianty jakie mogą wystąpić w przypadku podwójnej dywersyfikacji rozwiązania. Jak można zauważyć sygnał poprawności STATUS_OPERACJI=OK uzyskujemy tylko i wyłącznie w przypadku porównywania (za pośrednictwem komparatora) ze sobą dwóch poprawnych oraz dwóch takich samych błędnych wyników działań bliźniaczych jednostek sterujących. Przy założeniu, że fragmenty zdywersyfikowanego rozwiązania zostałyby zrealizowane przez niezależne grupy inżynierów to uzyskanie dwóch takich samych błędnych efektów działania układu sterującego wydaje się być mało prawdopodobne, a wręcz nieprawdopodobne. W efekcie, tak zaproponowane rozwiązanie wykrywa zgodności efektów działań w obu kanałach i na tej podstawie stwierdza poprawność funkcjonowania całego układu BML w systemie krytycznym czasu rzeczywistego. Inaczej wygląda sytuacja w przypadku potrójnej dywersyfikacji (rys. 4b), gdzie w komparatorze logicznym weryfikowane są pomiędzy sobą już efekty trzech niezależnych, ale dotyczących tej samej operacji, działań układu BML. Ten wariant układowy funkcjonuje dokładnie tak samo jak poprzedni, jednak tutaj mamy możliwość weryfikacji poprawności oraz zgodności ze sobą analizowanych wyników. Ta opcja rozwiązania pozwala na precyzyjne stwierdzenie, czy porównywany efekt funkcjonowania układu jest poprawny czy też nie, oraz, który z nich jest niepoprawny w przypadku wykrycia awarii. W przypadku pierwszej dywersyfikacji (podwójnej) wykrycie na wejściach komparatora jakiekolwiek różnicy skutkowało, przejściem całego układu do jego stanu bezpiecznego, zaś w drugiej wersji dywersyfikacji (potrójnej) wykrycie różnic na wejściu komparatora pozwala na warunkowe funkcjonowanie układu bezpiecznego, ponieważ sygnał poprawności STATUS_OPERACJI=OK wtedy generowany był na podstawie dwóch pozostałych poprawnych porównywanych ze sobą wielkości. Wówczas, tak wykryty pojedynczy błąd (zgodnie z zasadą, że każdy kanał powinien być dywersyfikowany przez niezależną grupę inżynierów) mógł być bez żadnych konsekwencji dla bezpiecznej pracy układu odrzucony, a układ nadal funkcjonowałby bezawaryjnie. Oczywiście w bezpiecznie funkcjonującym układzie czy systemie żadnego błędu lub awarii nie można zbagatelizować, ale pomimo jego wykrycia nadal zachowana została ciągłość bezawaryjnego funkcjonowania układu BML, a w tym czasie obsługa techniczna zostaje powiadomiona o zaistniałym fakcie. W rezultacie, na skutek dywersyfikacji układu BML uzyskano rozwiązanie, które podczas pracy na bieżąco samoczynnie się testuje oraz diagnozuje, a dodatkowo informuje o tym fakcie obsługę techniczną systemu czy nawet samego konstruktora (projektanta).

6. Wnioski

Tak jak przedstawiono, dywersyfikując rozwiązanie standardowego układu sterującego uzyskano całkowicie nową koncepcję układu przeznaczonego do sterowania i nadzorowania krytycznymi systemami czasu rzeczywistego. Pomimo, że w artykule skupiono uwagę głównie na architekturze oraz dywersyfikacji rozwiązania, to układ BML został również wyposażony w inne rozwiązania oraz mechanizmy, które w połączeniu ze sobą dość znacznie podwyższyły poziom bezpieczeństwa i bezawaryjności jego pracy. Jednak, w przypadku zaproponowanej jednostki to architektura w połączeniu z dywersyfikacją rozwiązań w niej zaimplementowanych ma największe i kluczowe znaczenie dla bezpieczeństwa i bezawaryjności pracy krytycznego systemu czasu rzeczywistego, którym ona zarządza. Wynika to z faktu, że tak zaproponowany mechanizm działania układu BML może bez problemu zarządzać, a tym bardziej analizować i weryfikować poprawność pracy dowolnego zestawu rozwiązań, w które mógłby on w razie potrzeby być wyposażony. W rezultacie, otrzymano bezpieczną jednostkę sterującą do zarządzania systemami krytycznymi oraz odporną na różnego rodzaju błędy i awarie zgodnie z założeniami projektowymi.

Koncepcja układu BML opracowana została pod kątem jej realizacji w rekonfigurowalnych strukturach typu FPGA, a więc układu, który umieszczony w jednej płytce krzemowej funkcjonowałby jako wieloprocesorowe rozwiązanie sterujące do zarządzania systemem krytycznym. Z przeprowadzonych badań i analiz gotowego już rozwiązania wynika, że tak zaproponowane rozwiązanie spełnia wszystkie oczekiwania konstruktora (projektanta) zgodnie z założeniami projektowymi. Jednak uzyskanie certyfikatu bezpieczeństwa dla tego typu wersji rozwiązania układu BML, który pozwoliłoby na jego wykorzystywanie do zarządzania systemami krytycznymi, jest niestety niemożliwe. Brak tej możliwości wynika z realizacji układu BML w strukturze FPGA, natomiast certyfikowane rozwiązania powinny pozwolić się zweryfikować zgodnie z kryteriami inżynierii odwrotnej (ang. Reverse Engineering), a tutaj tak nie jest. Przykładowo, implementując gotowe rozwiązanie w strukturze FPGA za każdym razem uzyskujemy całkowicie odmienną mapę połączeń i dlatego, w celu uzyskania certyfikatu bezpieczeństwa dla układu BML równolegle prowadzone są badania, które mają na celu fizyczne odseparowanie od siebie wspomnianych ośrodków sterujących MASTER1 i MASTER2 oraz obliczeniowego SLAVE. Jednak tak modyfikując pierwotną koncepcję rozwiązania układu BML trzeba zwrócić szczególną uwagę na to, aby nowe rozwiązanie było zdolne do wykonywania swoich zdywersyfikowanych zadań oraz działań w sposób współbieżny, aby wyeliminować zbędne opóźnienia związane z komunikacją, analizą oraz przetwarzaniem informacji. W rezultacie, spełnienie tych wymagań pozwoli na pozytywne przejście procedur oraz uzyskanie przez koncepcję układu BML certyfikatu bezpieczeństwa.

Autor jest stypendystą w ramach Poddziałania 8.2.2 „Regionalne Strategie Innowacji", Działania 8.2 „Transfer wiedzy", Priorytetu VIII „Regionalne Kadry Gospodarki" Programu Operacyjnego Kapitał Ludzki współfinansowanego ze środków Europejskiego Funduszu Społecznego Unii Europejskiej i z budżetu państwa.

7. Literatura

[1] Sałamaj M.: A new conception of safety logic microcontroller. In-ternational Journal of Electronics and Telecommunications, vol. 58, no. 4, s. 419-424, 2012.
[2] Adamski M., Sałamaj M.: Programowalny sterownik logiczny. Zgłoszenie patentowe P388721, 03.08.2009.
[3] Halang W. A., Śnieżek M.: A safe programmable electronic system, Bulletin of the Polish Academy of Sciences, Technical Sciences, vol. 58, no. 3, s. 423-434, 2010.
[4] Sacha K.: Real Time Systems Education at Warsaw University of Technology. Proceedings of ttp://www.xilinx.com/) the Third IEEE Real-Time Systems Education Workshop, pp. 36-40, IEEE Computer Society, 1999.
[5] Rinaldi J.: Control IEC 61131-3 - The Fast Guide to IEC 61131-3 Open Control Standard & Software, Real Time Automation, 2010.
[6] Śnieżek M., Halang W. A.: Bezpieczny programowalny sterownik logiczny. Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów, Polska, 1998.

otrzymano / received: 05.06.2014
przyjęto do druku / accepted: 01.10.2014 *artykuł recenzowany / revised paper*