dr inż. Magdalena Szymczyk, dr inż. Piotr Szymczyk
Akademia Górniczo-Hutnicza, Katedra Automatyki

ZASTOSOWANIE MECHANIZMU ZEGAROWEGO DO WYKRYWANIA ANOMALII ZACHOWANIA KOMPUTEROWEGO SYSTEMU STEROWANIA

Projektowanie niezawodnego, nieczułego na zakłócenia, wbudowanego systemu sterowania jest jednym z podstawowych aspektów tworzenia systemu sterowania. Artykuł ten przybliża ideę systemu monitorującego pracę systemu wbudowanego. Określa podstawowe założenia i wymagania dotyczące takiego mechanizmu, opisuje możliwe rozwiązania pewnych problemów.

WATCHDOG TIMER MECHANIZM FOR DETECTING EXECUTION COMPUTER ANOMALIES IN CONTROL SYSTEMS

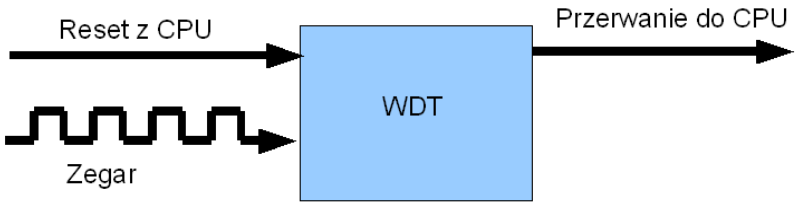
Designing reliable embedded control systems is one of the basic aspects of control systems. In this paper the idea of monitoring embedded system is presented. The main assumptions and requirements for such system are explained. Possible solutions of these problems are presented.

1. WPROWADZENIE

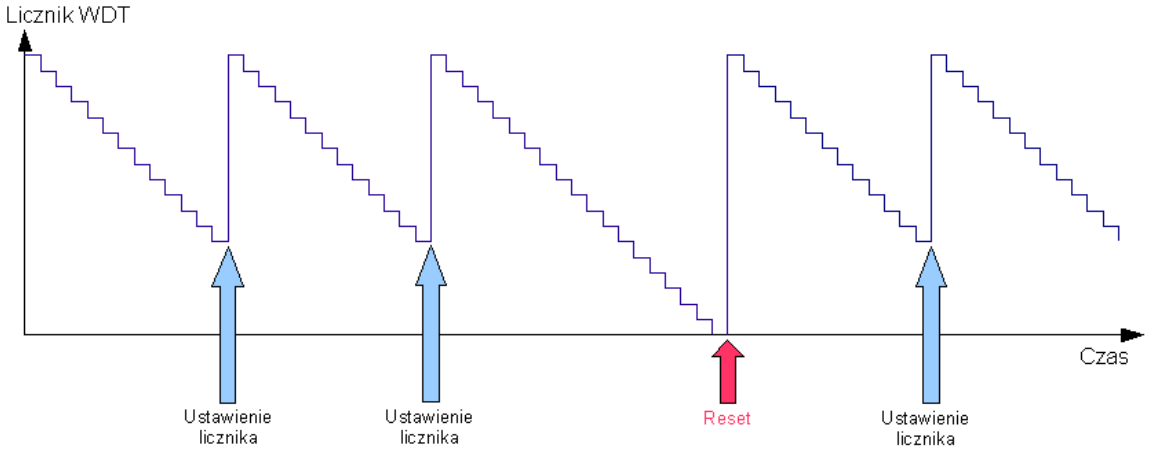
Gwałtowny postęp myśli technicznej zarówno w zakresie sprz ętu komputerowego jak i oprogramowania spowodował, że systemy uruchamiane na dużych i drogich konfiguracjach sprzętowych mogą być aktualnie urucham iane na pojedynczym układzie scalonym wykonującym specyficzne zadania. Mo żliwości wydajnościowe takich system ów wbudowanych, są porównywalne do tych system ów bazujących na kom puterach klasy PC sprzed 5 laty. Mo żliwości jak i z łożoność takich system ów stale ro śnie, konieczne jest zapewnienie ich odporno ści na b łędy. Systemy wbudowane m ogą doświadczać różnego rodzaju błędnych działań, ale ogólnie można je określić jako twarde i miękkie. Twarde awarie prowadzą do nieodwracalnych uszkodze ń w system ie (na przyk ład wpływ wysokiej temperatury na uk łady scalone), a awarie m iękkie wynikające na przyk ład z uszkodzenia danych zawartych w pam ięci są odwracalne. Pojawienie si ę mikroprocesorów zrewolucjonizowało także układy sterowania. Zbieranie danych z różnego rodzaju czujników, skomplikowane przetwarzanie tych inform acji, wypracowywanie odpowiedniego sygna łu sterowania, optymalizacja wydajności zamkniętego układu sterowania, nie by łoby możliwe bez obecności układu mikroprocesorowego. Od ich poprawnego dzia łania niejednokrotnie zależy nasze życie (rozruszniki serca, uk łady podtrzymujące podstawowe funkcje życiowe) czy w wi ększej skali, bezpieczne dzia łanie elektrowni atom owych. Są to zazwyczaj dedykowane układy mikroprocesorowe, wykonujące specyficzne dzia łania. Systemy wbudowane znacznie lepiej toleruj ą błędnie działający program niż układy typu desktop, nie wynika to jednak że z faktu, że oprogramowanie nie zawodzi, lecz z obecno ści mechanizmu umożliwiającego powrót do norm alnej pracy, gdy z jakich ś przyczyn oprogram owanie źle funkcjonuje. Częstym rozwiązaniem w tego typu system ach jest zastosowanie niezale żnego mechanizmu monitorującego przebieg sterowania programem.

Mechanizm taki zwany ”watchdog” (skrót W DT) jest konieczny przy budowaniu odpornych na błędy systemów wbudowanych, (przyk ład misja Pathfinder na Marsa). Je śli oprogramowanie systemu zawiesi si ę, przy pom ocy odpowiednio zaprojektowanego mechanizmu, który zajm uje się detekcją nieprawidłowego stanu oprogram owania, możliwe

jest zresetowanie całego systemu. Wydaje się, że tego typu rozwiązane jest ostatnią linią obrony przed błędami, gdy wszystkie inne rozwiązania zawiodą. Mechanizm WDT znany jest od lat i stosowany na przykład w programowalnych sterownikach logicznych (PLC). Tylko niektóre procesory znanych producentów są zabezpieczone przed przekłamaniami na poziomie bitów (spowodowanych na przykład promieniowaniem kosmicznym). Jednakże większość systemów wbudowanych nie posiada możliwości resetowania systemu z zewnątrz. Można wyróżnić następujące typy mechanizmu WDT: wewnętrzne, zewnętrzne, śledzące zmienne stanu programu, resetujące lub nie zewnętrzne urządzenia. WDT może charakteryzować się różnym stopniem złożoności, a w swej najprostszej wersji jest to zwykły licznik zliczający w dół, od pewnej wartości początkowej do zera. Na rys. 1 przedstawiono typowe rozwiązanie, a z kolei rysunek 2 pokazuje jego przebiegi czasowe. Mechanizm WDT może być wbudowanym elementem procesora lub układem działającym niezależnie. Taki prosty WDT daje możliwość stwierdzenia czy dane CPU funkcjonuje poprawnie, to znaczy, czy w określonych odcinkach czasu wysyła sygnał resetujący licznik (oznacza to, że CPU jest „żywe").



Rys.1. Koncepcja mechanizmu WDT [4]



Rys. 2. Diagram czasowy dla mechanizmu WDT z rys. 1

2. WEWNĘTRZNY WDT

Wewnętrzny WDT to taki, który jest wbudowany w układ scalony procesora (Maxima MAXQ2000). Wysokointegrowane procesory zawierają w sobie wiele dodatkowych układów, w których z kolei często wbudowane są proste mechanizmy WDT (np. Toshiba TPM96141), elementy te generują NMI, gdy ich licznik przekroczy limit czasu, co nie zawsze jest rozwiązaniem najlepszym [5]. Korzystne jest, gdy po wygenerowaniu przerwania NMI, system odczeka przez chwilę i dopiero potem wykonywany jest jego reset (czego procesor Toshiby nie potrafi zrealizować). Czas ten potrzebny jest do wykonania pewnych

pomocniczych operacji związanych z zapisaniem informacji o stanie procesora, co jest niezbędne do późniejszego określenia przyczyny problemu. Aby WDT pracował poprawnie, konieczny jest sprzętowy reset umożliwiający powrót procesora do ponownej, poprawnej pracy. Ustawienie samego licznika instrukcji (rejestr PC) może nie zainicjować poprawnej pracy CPU. Dobrą cechą WDT jest także to, że dopiero dwie, bezpośrednio następujące po sobie operacje zapisu danych o wartościach na przykład 0x55 i 0xaa, mogą dezaktywować timer, tak że przypadkowe operacje pisania programu głównego do pamięci z małym prawdopodobieństwem mogą wyłączyć ten mechanizm. Ze szczególną ostrożnością należy podchodzić do takich WDT, których rejestry sterujące mogą zostać zmodyfikowane w trakcie wykonywania programu, a to także grozi wyłączeniem watchdoga. Jeżeli przekroczenie limitu czasu WDT nie powoduje ustawienia odpowiedniego stanu na nodze procesora, to konieczne jest dołożenie sprzętu resetującego każde urządzenie zewnętrzne. Bez tego, taki reset przywróci tylko CPU do życia, lecz niepoprawnie działające peryferia spowodują dalszą niepoprawną pracę systemu.

3. ZEWNĘTRZNE WDT

Część urządzeń nadzorujących pracę procesora i zrządzających sygnałem resetu posiada wbudowane mechanizmy WDT. Nie wszystkie jednak zawsze zadziałają poprawnie (np. chip TI UCC3946, SMT czy też Maxim MAX 823), dlatego należy być świadomym ich zalet i wad [5]. Dobrze jest, gdy WDT działają w pewnym oknie czasowym. Oznacza to, że sygnał świadczący o poprawnej pracy procesora jest przyjmowany w ściśle określonym odcinku czasu i błędnie działający program raczej z małym prawdopodobieństwem prześle prawidłowo w jego trakcie, dwie ściśle określone wartości, co z kolei spowoduje wyzerowanie licznika. Takim procesorem jest Maxima MAX6752, posiada on zewnętrzne kondensatory do ustawiania czasu trwania okresu watchdoga oraz odpowiedniego dostosowania czasu trwania sygnału resetującego. Często WDT muszą zostać nieaktywne podczas startu pracy systemu (boot procesora). Procesor MAX 6369 jest wzbogacony o mechanizm wyczekiwania dodatkowy dłuższy odcinek czasu podczas inicjalizacji systemu. Dodatkowo zewnętrzne WDT mogą monitorować poziom napięcia zasilającego i resetować system w przypadku przekroczenia zbyt niskiej wartości. Niektóre z nich posiadają kodowane wejścia sygnału resetującego.

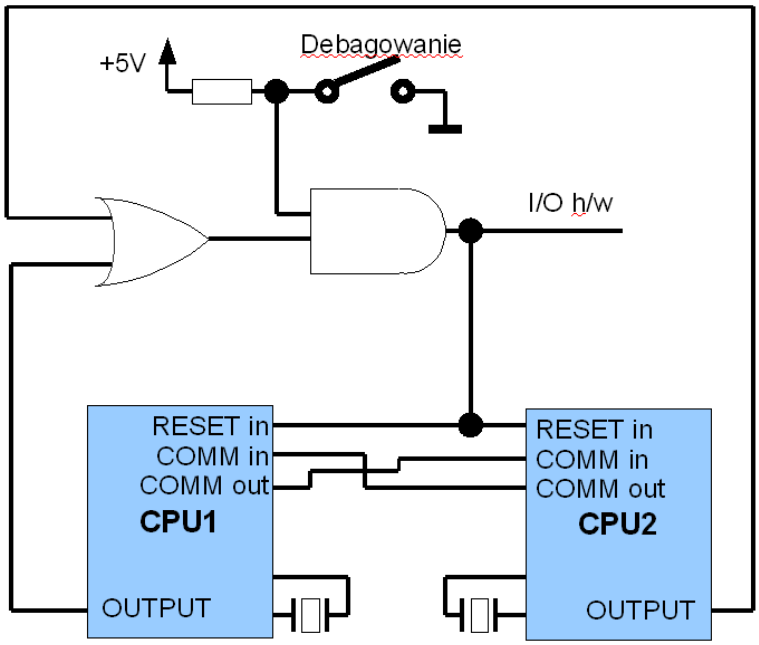
4. CECHY DOBREGO WDT

Podstawowym wymaganiem dla systemu sprawdzającego poprawność pracy systemu wbudowanego jest konieczność jego dalszej pracy, niezależnie od tego, co stało się z kodem głównego systemu, a w konsekwencji i z współpracującymi urządzeniami zewnętrznymi. Gdyby błędnie działający kod modyfikował kod czy rejestry watchdoga, to taki system nie ma szans na powrót do normalnej pracy. Stworzenie nawet prostego mechanizmu WDT nie jest zatem zadaniem prostym. Mechanizm taki nie powinien robić żadnych założeń co do aktualnego stanu oprogramowania czy sprzętu. Jeżeli coś poszło źle, to jego zadaniem jest przywrócić system do normalnej pracy lub ewentualnie wprowadzić w stan bezpiecznej pracy. Watchdog powinien dawać możliwość zapisania pewnych informacji, które umożliwiłyby później odtworzenie przyczyny błędu. Dobry mechanizm WDT jest niezależny od systemu głównego. Jest jeden poprawny sposób interwencji WDT, a mianowicie wysłanie sygnału resetu do procesora oraz do wszystkich podłączonych urządzeń zewnętrznych, a nie wysłanie tylko przerwania NMI. W niektórych rozwiązaniach stosuje się sygnał NMI, aby

zapisać informacje pomocne do odtworzenia przyczyny problem u jak i przes łać wiadomość do urządzeń zewnętrznych o tym, że CPU przestaje pracowa ć na krótką chwilę. Natychmiast po tych operacjach wysy łany jest sygna ł resetu. Czas zw łoki pomiędzy stwierdzeniem błędnego działania systemu wbudowanego, a odpowiedzi ą watchdoga powinien by ć jak najkrótszy. Należy pamiętać o tym, że system wbudowany powinien pozostać po uszkodzeniu w takim modzie pracy, aby by ł nieszkodliwy dla otoczenia, zatem sprzęt powinien zosta ć przywrócony do życia lub wprowadzony w stan bezpiecznej pracy. Zegar licznika powinien być niezależny od zegara system u wbudowanego, a zatem każdy WDT wbudowany w CPU systemu głównego nie jest bezpieczny. Tworz ąc watchdoga trzeba by ć pewnym, że żadne okoliczności związane z prac ą oprogramowania systemu wbudowanego nie spowoduj ą przeprogramowania WDT (jego rejestrów, pinów). Prawidłowo zbudowany mechanizm WDT monitoruje pracę całego systemu, a nie tylko przyjm uje sygnały „jest żywy" (w dom yśle procesor) pochodzące z pętli głównej (może okazać się, że jest więcej procesów wysyłających te sygnały, a któryś aktualnie został zawieszony). Użytkownik powinien być poinformowany w jakiś sposób o fakcie resetu systemu (świecąca dioda, komunikat).

W przypadku tworzenia wewn ętrznego WDT (dla bardzo tanich system ów wbudowanych) istnieje kilka praktycznych rad daj ących możliwość tworzenia bardziej niezawodnego rozwiązania opartego w g łównej mierze o rozwiązania programistyczne. Dobrym zwyczajem jest tworzenie tak zwanych punktów kontrolnych w program ie watchdoga, które m odyfikują pewne zmienne w program ie pętli głównej. W przypadku, gdy warto ść sprawdzana w watchdogu różni się od tej w program ie, następuje wzbudzenie W DT, w przeciwnym przypadku wysyłany jest sygnał życia.

Do budowy zewn ętrznych WDT można użyć prostych m ikroprocesorów, tak aby system monitorujący pracę systemu wbudowanego m iał niezależny zegar, w łasną pamięć i wbudowane liczniki. Nie wydaje si ę sensowne wykorzystanie całego komputera do budowy WDT innego system u, choć można użyć procesorów jako m onitorów poprawnej pracy pozostałych, na przykład w systemie wieloprocesorowym.



Rys. 3. WDT dla systemu z dwoma procesorami wzajemnie sprawdzającymi się

5. WNIOSKI

Coraz większa liczba systemów wbudowanych stosowanych w urządzeniach o znaczeniu krytycznym ze względu na bezpieczeństwo powoduje, że wymagania dotyczące ich odporności na uszkodzenia stale rosną. Dość znaczącym elementem tego typu aplikacji staje się monitor, który obserwuje sposób wykonywania się programu oraz stan pracy systemu po błędzie. Prosty licznik stosowany jest już od lat, lecz współczesne rozwiązania mechanizmu WDT stają się coraz bardziej odporne na błędy. Analizując zebrane informacje, zewnętrzne WDT powinny być używane dla systemów krytycznych, czyli takich, które potrzebują wyższych wartości współczynnika niezawodności. Ale podłączenie pinu resetowania zewnętrznego WDT do mikroprocesora (czyli WDT może zostać zresetowany programistycznie) powoduje, że obniża się jego niezawodność. Rozwiązania oparte na wewnętrznych WDT stosuje się w aplikacjach niekrytycznych i dla redukcji kosztów.
Mechanizm WDT daje możliwość znacznego zwiększenia niezawodności systemu sterowania przy naprawdę niewielkim nakładzie kosztów.

6. LITERATURA

[1] BARANOWSKI R.: *Mikrokontrolery AVR ATmega w praktyce*, 2005, BTC

[2] BARR M.: *Introduction to Watchdog Timers, Embedded Systems Design, 2001, http://www.embedded.com/columns/beginerscorner/9900324?_requestid=301900*

[3] BERGER A.: *Embedded System Design*, 2002,CMP Books

[4] CHAKRAVARTY S., TOMAR R., ARORA M.: *Need a watchdog for improved system fault tolerance? Freescale Semiconductor, CommsDesign , 2008 http://www.commsdesign.com/showArticle.jhtml;jsessionid=G0N4FX04LDO5QQSNDLRSKHSCJUNN2JVN?articleID=211600055*

[5] LABROSSE J. [ET AL.]: *Embedded Software*, 2008, Elsevier

[6] LAPLANTE P.*:Real-Time Systems Design and Analysis*, IEEE Press, Wiley-Interscience, 2004

[7] MURPHY N.: *Watchdog Timers, Embedded Systems Design, 2003, http://www.embedded.com/columns/beginerscorner/9900324?_requestid=301900*

[8] SZYMCZYK M. *Projektowanie wbudowanych systemów sterowania odpornych na zakłócenia,* PAR kwiecień 2009