Mirosław CHMIEL, Jan MOCHA, Edward HRYNKIEWICZ
POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI,
ul. Akademicka 16, 44-100 Gliwice

Xilinx Virtex-4 jako platforma rozwojowa jednostek centralnych PLC

Dr inż. Mirosław CHMIEL

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w 1992 roku. Na tymże wydziale w roku 2003 obronił pracę doktorską z dziedziny elektronika. Obecnie jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe koncentrują się wokół cyfrowych układów sterowania, ze szczególnym uwzględnieniem konstrukcji oraz wykorzystania systemów opartych o sterowniki programowalne PLC.

e-mail: miroslaw.chmiel@polsl.pl

Mgr inż. Jan MOCHA

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w 2007 roku. Jest doktorantem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe koncentrują się wokół układów logiki programowalnej w szczególności dynamicznej częściowej rekonfiguracji układów programowalnych FPGA, praktycznych aspektów wykorzystania układów elektronicznych w aplikacjach przemysłowych i medycznych oraz zagadnień kompatybilności elektromagnetycznej.

e-mail: jan.mocha@polsl.pl

Dr hab. inż. Edward HRYNKIEWICZ

Studia ukończył na Wydziale Automatyki Politechniki Śląskiej. Rozprawę doktorską obronił w roku 1978, a habilitował się w roku 1992 w dyscyplinie elektronika. Zajmuje się syntezą logiczną, układami programowalnymi, programowalnymi sterownikami logicznymi oraz cyfrowymi układami przetwarzania częstotliwości. Obecnie pełni funkcję dyrektora Instytutu Elektroniki na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej.

e-mail: edward.hrynkiewicz@polsl.pl

Streszczenie

Artykuł prezentuje koncepcję platformy sprzętowo-programowej umożliwiającej testowanie różnych rozwiązań konstrukcyjnych jednostek centralnych sterowników programowalnych. Platforma do testowania jednostek bazuje na układzie FPGA Virtex-4 oraz opracowanym dedykowanym oprogramowaniu narzędziowym, umożliwiającym testowanie oraz badania właściwości opracowywanych jednostek. Przedstawiono wybrane dwuprocesorowe bitowo-bajtowe jednostki spotykane w literaturze, zorientowane na maksymalnie efektywne wykorzystanie obydwu procesorów. Szczególną uwagę zwrócono na szybkość wykonywania programu sterowania oraz funkcjonalność jednostki.

Słowa kluczowe: programowalny sterownik logiczny PLC, jednostka centralna, przetwarzanie współbieżne, układy programowalne, FPGA.

Xilinx Virtex-4 – based PLC CPUs development platform

Abstract

To develop fast central processing units (CPUs) of programmable logic controllers (PLC) one can employ the architecture with two processors: a bit and a byte processor. The bit processor shall be responsible for processing the bit variables, while the byte processor shall be meant to deal with the byte (word) variables [1, 2, 3, 4, 5, 6]. In case of the double-processor architecture it is extremely important to synchronize operation of data exchange between the processors. The literature references report various synchronization methods [9, 10, 11, 12] that are described in Section 3. Sections 4 and 5 outline the combined hardware and software platform intended to enable testing and comparison between various architectures of CPUs. The presented solution employs a programmable FPGA module from the Virtex-4 family [7, 8], that are described in Section 2. The newly developed software enables compilation of application programs dedicated for the presented architecture. To develop programs for the presented solution the authors used the assembler-type programming language very similar to STL language that is normally applicable to STL controllers from Siemens [13, 14]. The software developed for PC computers make it possible to define new instructions for processors both on hardware and software levels (Fig. 1). The presented solution takes advantage of components that are typical for FPGA modules, such as BockRAM memory units (Fig. 2). The presented platforms enable further research and development efforts intended to design fast CPUs for programmable logic controllers.

Keywords: PLC - programmable logic controller, central processing unit, concurrent operation, FPGA - field programmable gate array.

1. Wprowadzenie

Zdolność obliczeniowa sterownika programowalnego jest określona przez czas konieczny do wykonania tysiąca instrukcji [1, 2]. Im czas realizacji poszczególnych instrukcji przez jednostkę centralną jest krótszy, tym szersze są możliwości zastosowania sterownika do różnych zadań, w szczególności o znacznych wymaganiach czasowych. Współczesne obiekty przemysłowe wymagają coraz bardziej skomplikowanych układów sterujących, a co za tym idzie układ sterujący nimi musi posiadać możliwości realizacji algorytmów znacznych rozmiarów. A równocześnie algorytm ten powinien być realizowany przez sterownik w jak najkrótszym czasie. Zaprojektowanie i zbudowanie jednostki centralnej, która będzie wykonywała program sterowania w możliwie najkrótszym czasie staje się istotnym problemem i zadaniem do rozwiązania [3, 4, 5, 6]. Do budowy platformy, umożliwiającej konstruowanie szybko działających układów sterowania, można wykorzystać współczesne układy FPGA. Płyty prototypowe z układami programowalnymi dają możliwość wielokrotnej weryfikacji poprawności projektu na różnych etapach prac konstrukcyjnych: począwszy od etapu symulacji a skończywszy na sprzętowej weryfikacji działania przed etapem wdrożenia.

W literaturze występują w zasadzie dwa rozwiązania sprzętowe jednostek centralnych sterowników PLC: standardowy lub dedykowany mikroprocesor oraz struktury dwuprocesorowe bitowo-bajtowe czy bitowo-słowowe. Autorzy skupiają się w swoich rozważaniach na rozwiązaniach z dwoma procesorami, gdyż dają one więcej możliwości konstrukcyjnych.

2. Podstawowe informacje o układach FPGA

Przy budowie dedykowanych cyfrowych układów sterowania konstruktor mógł dotychczas skorzystać ze specjalizowanych układów scalonych wielkiej skali integracji albo zaprojektować własny system cyfrowy, korzystając z dostępnych na rynku cyfrowych układów scalonych. Obecnie pojawiła się możliwość wykorzystania układów programowalnych o bardzo dużych zasobach logicznych. Układy logiki programowalnej umożliwiają rekonfigurację, co oznacza możliwość zmiany ich funkcjonalności w zależności od chwilowych potrzeb. Umożliwiają one również wygodne prototypowanie na etapie testowania konstrukcji. Projekt, w przypadku wykorzystania układów programowalnych, tworzony jest najczęściej z wykorzystaniem języka opisu sprzętu HDL. Jako przykład można wymienić dwa najpopularniejsze języki: VHDL i Verilog. Wykorzystując układy logiki programowalnej mamy możliwość budowy prostych układów cyfrowych, jak również złożonych systemów mikroprocesorowych, opartych o wysokospecjalizowane mikroprocesory dedykowane do potrzeb konkretnej aplikacji. Z możliwości tej skorzystano w prezentowa-

nej pracy – wykorzystano język VHDL do zdefiniowania specjalizowanych mikroprocesorów: bitowego oraz bajtowego oraz zaprojektowano układy peryferyjne niezbędne do pracy całości, jako jednostki centralnej sterownika PLC.

Wśród układów programowalnych największymi zasobami oraz możliwościami charakteryzują się układy FPGA. Właśnie układy FPGA wykorzystano w niniejszej pracy, a konkretnie układy rodziny Virtex-4 firmy Xilinx [7]. Układy FPGA charakteryzują się symetryczną architektura składającą się z konfigurowalnych bloków logicznych CLB oraz sieci pionowych i poziomych konfigurowalnych połączeń pomiędzy blokami. Na każdy blok konfigurowalny CLB składa się kilka elementów składowych, wśród których jako najważniejsze należy wymienić: tablicowe generatory funkcji logicznych LUT oraz przerzutniki. Matryca konfigurowanych bloków logicznych CLB otoczona jest blokami wejścia-wyjścia, które umożliwiają podłączenie do układu FPGA układów zewnętrznych [7].

Aby zwiększyć funkcjonalność układów FPGA producenci umieszczają w strukturze wewnętrznej układu FPGA specjalizowane bloki sprzętowe, np. dwubramowe synchroniczne bloki pamięci RAM (BlockRAM), sprzętowe układy mnożące oraz układy zarządzania sygnałem zegarowym DCM (ang. *Digital Clock Manager*).

W trakcie prowadzonych eksperymentów wykorzystano płytę uruchomieniową ML401 z układem Virtex-4 [8]. Oprócz układu FPGA na płycie znajdują się wszystkie niezbędne układy peryferyjne, z których może skorzystać projektant, m.in. pamięci SRAM oraz Flash, port RS-232, diody LED oraz przyciski do realizacji prostego interfejsu użytkownika [8].

3. Przegląd jednostek centralnych dla sterowników programowalnych PLC

Jednostki centralne o konstrukcji bitowo-bajowej (słowowej) są stosowane do konstrukcji jednostek centralnych sterowników PLC. W literaturze spotyka się różne metody współdziałania procesorów w jednostkach dwuprocesorowych, najczęściej jeden z procesorów wykorzystuje się od przetwarzania zmiennych dwustanowych, zaś drugi do przetwarzania zmiennych numerycznych. Poniżej zostanie zaprezentowanych kilka takich rozwiązań.

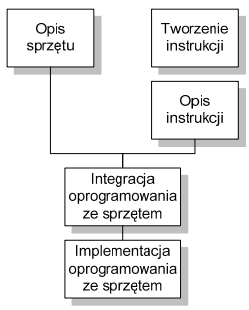
1. Połączenie procesora bajtowego z procesorem bitowym traktowanym jako jednostka podrzędna – koprocesor. W rozwiązaniu tym, rola procesora bitowego została ograniczona do szybkodziałającej jednostki, która na polecenie procesora nadrzędnego wykonuje operacje na zmiennych dwustanowych [3].
2. Jednostka centralna z układem dwóch procesorów ze wspólną częścią odpowiedzialną za pobieranie i dekodowanie rozkazów pochodzących ze wspólnej pamięci programu, w której każdy z procesorów ma swój własny, odrębny zestaw instrukcji do wykonania. Obydwa procesory współpracują ze wspólnymi pamięciami programu oraz odwzorowania wejść oraz wyjść [1].
3. Donandt [6] proponuje, aby procesory posiadały swoje odrębne pamięci programu, w których przechowywane będą odpowiednie części programu sterowania. W skład jednostki wchodzą dwa procesory, z których każdy ma swoją pamięć programu, co powoduje, że mogą one działać równolegle. W proponowanym układzie istnieją dwa typy synchronizacji. Procesory działają równolegle, przekazując sobie niezbędne informacje w czasie trwania kolejnych obiegów pętli oraz oczekując na siebie po zakończeniu realizacji każdej z nich. Jednak w układzie główna magistrala jednostki jest kontrolowana przez procesor bajtowy i z tej perspektywy procesor bitowy traktowany jest, jako element przestrzeni adresowej procesora bajtowego.
4. Następne rozwiązanie łączy w sobie idee zaprezentowane w punktach 2 oraz 3 i proponuje, aby procesor bajtowy posiadał dwie pamięci programu: pamięć programu procesora bajtowego, która przechowuje fragmenty mikroprogramów, które zawierają argumenty oraz pamięć programów standardowych, gdzie przechowywane będą stałe procedury rozkazów bezargumentowych. Pozwoli to zaoszczędzić na pojemności zarówno głównej pamięci programu, jak również na pamięci dla procesora bajtowego. Procesor bitowy posiada dodatkowo wbudowany dekoder instrukcji, który decyduje, dla którego procesora przeznaczony jest dany rozkaz [9, 4].
5. W przytoczonych pracach każdy z procesorów posiada pamięć danych, jednak pamięć odwzorowań wejść/wyjść jest jedna, co może powodować konflikty w dostępie do jej zasobów. Wydaje się, iż lepszym rozwiązaniem jest zdecydowanie się na bezkonfliktowy dostęp do sygnałów obiektowych, które dla obydwu procesorów stanowią zbiory rozłączne. Aby do minimum ograniczyć konflikty dostępu do zasobów wspólnych, układ będzie posiadał dwie magistrale danych, dwie magistrale adresowe oraz dwie magistrale sygnałów sterujących. Jak pokazano w pracy [10] procesor bitowy powinien pełnić rolę procesora równorzędnego bajtowemu (słowowemu). Jednak przy próbie całkowitego zrównoleglenia pracy procesorów, podstawowym problemem do rozwiązania staje się koordynacja wymiany danych pomiędzy nimi. Przekazywanie informacji odbywa się dzięki mechanizmowi zawierającemu dwa przerzutniki z układami odpowiedzialnymi za sprawdzanie aktualności zawartych w nich informacji. Takie rozwiązanie pozwala na wykonywanie operacji równolegle przed obydwa procesory aż do chwili, w której warunek wypracowany w jednym z nich musi zostać wykorzystany do wykonania instrukcji w drugim.
6. Jeszcze większe szybkości realizacji programu można uzyskać dzięki całkowicie równoległej pracy obydwu procesorów, poprzez zapewnienie dwóch zestawów przerzutników warunku – dla każdego kierunku przesyłania informacji jeden [11, 12]. Należy zauważyć, że procesory stanowią dla siebie nawzajem coś na kształt zestawu wejść, których stan ulega zmianie niezależnie od drugiego procesora. Platforma FPGA daje możliwość zastosowania dwóch rodzajów układów wymiany informacji: dwóch zestawów przerzutników warunku lub pamięci typu BlockRAM. Wymieniana informacja zapisywana jest do kolejnej komórki pamięci, a drugi procesor informację tę czyta. Rozwiązanie to pozwala na całkowite zrównoleglenie wykonywania programu. Jednak przy dużych dysproporcjach w czasie realizacji programu przez poszczególne procesory możemy mieć do czynienia z sytuacją, w której stan danej komórki wymiany ulegnie zmianie kilkakrotnie w czasie obiegu pętli procesora, który tę informację odczytuje. Aby proces dostępu do informacji warunkowych odbywał się dokładnie tak, jak w przypadku wejść/wyjść, układ został wyposażony w dodatkową pamięć, która przechowuje kopię stanu informacji o warunkach przekazywanych pomiędzy procesorami. Jednostka jednak została tak przygotowana, aby możliwa była praca z odczytem natychmiastowym nowego stanu warunku oraz odroczonym, czyli po zakończeniu kolejnego obiegu pętli.

4. Platforma programowa

Aby móc w pełni wykorzystać możliwości, jakie dają badane jednostki centralne, należało stworzyć własny kompilator, który umożliwiałby pisanie oraz kompilowanie programu według wymagań poszczególnych jednostek. Jako naturalnym językiem programowania uznano język asemblerowy. Zaproponowany język posiada cechy zarówno języka STL dla sterowników rodziny S7-300/400 [13], jak i sterowników rodziny S7-200 [14] firmy Siemens. Głównym zadaniem napisanego kompilatora jest analiza programu pod względem składniowym oraz jego podział na dwa strumienie instrukcji – dla procesora bajtowego oraz dla procesora bitowego. Programista pisze program w postaci jednego ciągu instrukcji. Natomiast kompilator rozdziela ten ciąg na dwa zestawy, które w dalszej części procesu są kompilowane z wykorzystaniem przygotowanego wcześniej zestawu instrukcji, następnie lista kodów instrukcji zapisywana jest w pamięciach programu poszczególnych procesorów.

Bardzo ważną w przypadku badań różnych jednostek centralnych, jest możliwość definiowania nowych rozkazów. Tworzenie nowych instrukcji polega na nadaniu nazwy mnemonicznej rozkazu oraz określeniu jego unikalnego kodu. Tworzeniu instrukcji

towarzyszy także zdefiniowanie zakresu parametrów, z jakimi dany rozkaz będzie mógł być wykorzystywany. Po zdefiniowaniu nowej instrukcji przechodzi się do etapu opisu jej funkcjonalności sprzętowej w języku opisu sprzętu (w tym wypadku użyto języka VHDL). Tak przygotowana nowa instrukcja musi zostać dołączona do zasobów jednostki centralnej, a następnie należy przeprowadzić proces implementacji jednostki centralnej w zasobach układu FPGA. Proces tworzenia nowej instrukcji został pokazany schematycznie na rys. 1.

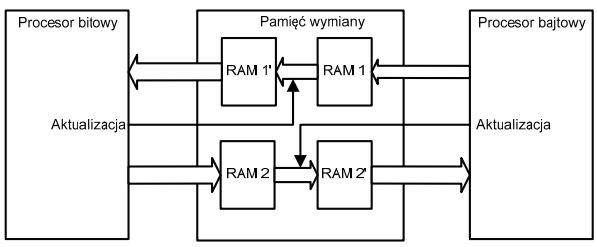


Rys. 1. Proces tworzenia nowej instrukcji
Fig. 1. Process of new instruction creation

Aby uzyskać możliwość testowania działania zbudowanych jednostek centralnych, wykorzystano występujące na płycie prototypowej dodatkowe elementy: mikroprzełączniki – jako wejścia dwustanowe, kilka diod LED – jako wyjścia dwustanowe, natomiast wejścia i wyjścia analogowe zostały zaimplementowane na wyprowadzonych na płycie złączach.

5. Platforma sprzętowa

Jako punkt wyjścia do badań przyjęto konstrukcję posiadającą dwa odrębne procesory. Jednak proces wykonywania rozkazów jest szeregowy z możliwością realizacji złożonych operacji bajtowych równolegle z następną operacją przeznaczoną dla procesora bitowego. Jako przeciwwaga została do badań wykorzystana struktura umożliwiająca pracę obydwu procesorów w sposób całkowicie równoległy, z przekazywaniem informacji na dwa różne sposoby: z potwierdzeniem oraz z przekazywaniem informacji pomiędzy procesorami poprzez pamięć wymiany, zrealizowaną w oparciu o pamięć BlockRAM (rys. 2).



Rys. 2. Jednostka centralna z pamięcią wymiany (Block RAM)
Fig. 2. CPU with exchange memory (Block RAM)

Na potrzeby prezentowanej platformy przygotowany został dedykowany procesor bajtowy oraz bitowy. Lista instrukcji obu procesorów zawiera typowe dla sterowników PLC operacje, takie jak: operacje logiczne dla procesora bitowego oraz operacje arytmetyczne, komparatory, operacje licznikowe oraz czasowe dla procesora bajtowego. Wszystkie operacje procesora nazywanego bajtowym (słowowym) wykonywane są na zmiennych 16-bitowych. W przeważającej większości instrukcje wykonywane są w dwóch taktach zegarowych, co przy sygnale zegarowym o wartości 50MHz, daje czas realizacji pojedynczej instrukcji wynoszący 40ns. Zaimplementowane wstępnie jednostki centralne zajmują 1841 bloków *slice* oraz 11 bloków BlockRAM wykorzystywanego układu FPGA Virtex-4.

6. Podsumowanie

Przygotowana platforma, jak i oprogramowanie pozwalają na testowanie możliwości, jakie dają bitowo-słowowe konstrukcje jednostek centralnych sterowników programowalnych PLC, realizujących sterowanie w sposób całkowicie programowy. Platforma została w chwili obecnej wyposażona w trzy jednostki centralne, w których wykorzystane zostały te same konstrukcje procesorów, jednak wyposażone w inne mechanizmy pobierania rozkazów z pamięci programu oraz różne mechanizmy przekazywania informacji pomiędzy procesorami. Na tym etapie prowadzone są badania nad optymalizacją zaproponowanych struktur oraz wypracowaniem algorytmu ich testowania i porównywania z istniejącymi na rynku konstrukcjami. Ponadto trwają prace nad zapisaniem programów testujących, tak, aby dawały one jak najwięcej informacji porównawczych. Rezultaty tych badań zostaną porównane z komercyjnymi rozwiązaniami i zostaną przedstawione w kolejnych publikacjach.

Praca finansowana ze środków Ministerstwa Nauki i Szkolnictwa Wyższego (5391/B/T02/2010/38).

7. Literatura

[1] Getko Z.: Programowalne systemy sterowania binarnego PLC. Elektronizacja, WKiŁ, Warszawa, 1983.
[2] Michel G.: Programmable Logic Controllers, Architecture and Applications. John Wiley & Sons, West Sussex, England, 1990.
[3] Aramaki N., Shimokawa Y., Kuno S., Saitoh T., Hashimoto H.: A new Architecture for High-performance Programmable Logic Controller. Proceedings of the IECON'97 23rd International Conference on Industrial Electronics, Control and Instrumentation, IEEE vol. 1, str. 187-190, New York, USA, 1997.
[4] Chmiel M., Hrynkiewicz E.: Remarks on Parallel Bit-Byte CPU structures of Programmable Logic Controllers. In: Design of Embedded Control Systems, Section V, (Adamski M.A., Karatkevich A., Węgrzyn M). str. 231-242, Springer Science + Business Media, Inc., 2005.
[5] Chmiel M., Hrynkiewicz E., Milik A.: Concurrent operation of the processors in Bit-Byte CPU of a PLC. Preprints of the IFAC World Congress, Prague, Czech Republic, July 3-8, 2005.
[6] Donandt J.: Improving response time of Programmable Logic Controllers by use of a Boolean coprocessor. IEEE Comput. Soc. Press. 4:167-169, Washington, DC, USA, 1989.
[7] Virtex-4 FPGA User Guide, UG070, version 2.6. www.xilinx.com, Xilinx, 2008, USA.
[8] ML401/ML402/ML403 Evaluation Platform User Guide, UG080 version 2.5. www.xilinx.com, Xilinx, USA, 2006.
[9] Koo K., Rho G.S., Kwon W.H., Park J., Chang N.: Architectural Design of an RISC Processor for Programmable Logic Controllers. Journal of Systems Architecture, vol. 44, nr 5, Feb. 1998, str. 311-325. Publisher: Elsevier, Netherlands, 1998.
[10] Chmiel M., Hrynkiewicz E.: Improving of Concurrent Operation of the Bit-Byte PLC CPU. International Conference on PDS, str.15-20, Brno, Czech Republic, February 14-17, 2006.
[11] Chmiel M., Hrynkiewicz E.: Fast Operating Bit-Byte PLC. Preprints of the 17th IFAC World Congress (on DVD-ROM), Seoul, Korea, July 6-11, str. 14810-14815, 2008.
[12] Chmiel M.: On Reducing PLC Response Time. Bulletin of the Polish Academy of Sciences. Technical Sciences, vol.56, nr.3, str.229-238, 2008.
[13] Berger H.: Automatic with STEP7 in STL and SCL – SIMATIC S7 300/400 Programmable Controllers. Siemens AG, Germany, 2001.
[14] Simatic S7-200 Programmable Controller System Manual ed. 04/2002, Siemens AG, 2002.

otrzymano / received: 15.10.2010
przyjęto do druku / accepted: 01.12.2010 *artykuł recenzowany*