Maciej GOŁASZEWSKI, Tadeusz SONDEJ
WOJSKOWA AKADEMIA TECHNICZNA, WYDZIAŁ ELEKTRONIKI, INSTYTUT TELEKOMUNIKACJI

System dwuprocesorowy do sterowania licznikiem czasu w układzie SoC

Maciej GOŁASZEWSKI

Jest studentem V roku Wydziału Elektroniki Wojskowej Akademii Technicznej, gdzie studiuje na specjalności Systemy Cyfrowe w trybie indywidualnym. Jest również zastępcą kierownika Koła Naukowego Studentów Wydziału Elektronik WAT. Jego zainteresowania to zastosowanie wbudowanych systemów mikroprocesorowych w układach FPGA.

e-mail: golaszewski.maciej@gmail.com

Dr inż. Tadeusz SONDEJ

Ukończył studia na Wydziale Elektroniki WAT, obronił pracę doktorską w 2003 r. Jest adiunktem w Instytucie Telekomunikacji Wydziału Elektroniki WAT. Jego zainteresowania naukowe to zastosowanie mikroprocesorów w programowych metodach podwyższania dokładności pomiaru oraz systemy-on-chip.

e-mail: tsondej@wel.wat.edu.pl

Streszczenie

W artykule przedstawiono problematykę projektowania systemów wieloprocesorowych jako zintegrowanych systemów cyfrowych (SoC – ang. *System-on-Chip*). Opisano zaprojektowany system, składający się z dwóch procesorów programowych Nios II firmy *Altera* i precyzyjnego licznika czasu o rozdzielczości około 80 ps. Pierwszy procesor odpowiedzialny jest za komunikację systemu przez interfejs Ethernet z aplikacją uruchamianą na komputerze PC. Drugi procesor steruje licznikiem czasu oraz zajmuje się obliczeniami statystycznymi w czasie wykonywania próby pomiarowej. Wymiana danych pomiędzy procesorami realizowana jest za pomocą pamięci współdzielonej.

Słowa kluczowe: system wieloprocesorowy, FPGA, precyzyjny licznik czasu.

Dual processor system for precision time counter based on system-on-chip device

Abstract

This paper presents issues of designing and implementing FPGA-based multiprocessor systems. Practical example consists of two softcore processors Nios II from *Altera*. Developed system is designed for control and data processing of precision timer counter with 80 ps resolution. The first processor runs as server, providing communication and supervision of the system via Internet. The second processor controls timer counter and performs statistical computation. Shared memory from FPGA resources is used to interchange data between processors.

Keywords: multiprocessor system, FPGA, precision time counter.

1. Wprowadzenie

Współczesne układy programowalne charakteryzują się coraz to większą liczbą zasobów logicznych, umożliwiających implementowanie w jednym układzie scalonym kompletnych systemów cyfrowych (SoC). Układy SoC zawierają zwykle mikroprocesor (implementowany jako sprzętowy lub programowy), specjalizowane moduły cyfrowe oraz pamięć. Chociaż wydajność procesorów sprzętowych przeważnie jest większa od programowych [1], to jednak procesory programowe oferują znacznie większą elastyczność [2] i łatwość projektowania systemów wieloprocesorowych [3]. Dlatego czołowi producenci układów programowalnych oferują biblioteki modułów IP (ang. *Intellectual Property*), w skład których wchodzą również procesory programowe. Powszechność procesorów programowych oraz ogromne zasoby logiczne układów FPGA umożliwiają budowanie systemów wieloprocesorowych w jednym układzie scalonym [4]. Systemy z kilkoma procesorami radykalnie zwiększają moc obliczeniową [5]. Jeżeli cały system wieloprocesorowy można zaimplementować w jednym układzie scalonym to również zwiększa się jego niezawodność i wymiary zewnętrzne.

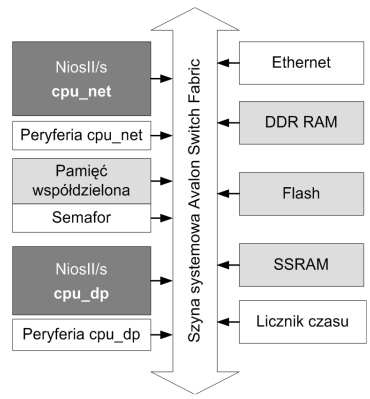
W niniejszym artykule omówiono zaprojektowany układ SoC zawierający dwa procesory programowe Nios II oraz precyzyjny licznik czasu. Jeden z procesorów zajmuje się sterowaniem licznikiem czasu i przetwarzaniem danych w czasie rzeczywistym, natomiast drugi realizuje komunikację z siecią Internet.

2. Projekt systemu dwuprocesorowego z licznikiem czasu

2.1. Ogólna idea działania systemu

Zaproponowany system przeznaczony jest do sterowania układem precyzyjnego licznika czasu i przetwarzania danych pomiarowych. Założeniem projektu było również opracowanie układu przy pomocy którego można by dokonywać zdalnych pomiarów oraz przeprowadzać jak największą liczbę obliczeń w układzie programowalnym. Dla serii pomiarowej istotnym parametrem jest oszacowanie wyniku pomiaru odpowiednią statystyką. Prostym estymatorem jest średnia arytmetyczna z próby. Bardziej zaawansowane estymatory, które potrafią odrzucić część zakłóconych pomiarów wymagają większych mocy obliczeniowych [6]. W celu skrócenia czasu przetwarzania danych pomiarowych zdecydowano się na rozdzielenie funkcji pomiarowo-obliczeniowych od funkcji komunikacyjnych. W tym celu opracowano system dwuprocesorowy, w którym jeden procesor (cpu_net) odpowiedzialny jest za komunikację systemu z zewnętrzną aplikacją sterującą a drugi steruje procesem dokonywania pomiaru i przeprowadza obliczenia statystyczne (cpu_dp).

Głównymi elementami zaprojektowanego systemu są procesory programowe Nios II oraz precyzyjny licznik czasu. Na rys. 1 zamieszczono schemat zaprojektowanego systemu dwuprocesorowego.



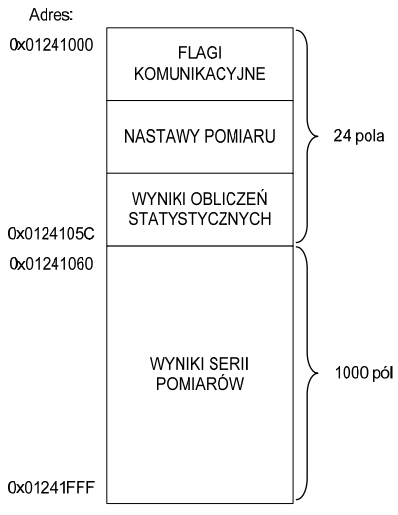
Rys. 1. Schemat zaprojektowanego systemu dwuprocesorowego
Fig. 1. Dual processor system block diagram

Procesory Nios II charakteryzują się 32-bitową architekturą typu RISC o harwardzkiej architekturze pamięci. Z dostępnych 3 wersji rdzenia wybrano wersję standardową charakteryzującą się korzystnym stosunkiem zajmowanych zasobów do wydajności. Wykorzystane z biblioteki standardowej procesora Nios II układy peryferyjne oraz interfejsy pamięci zostały połączone szyną systemową Avalon Switch Fabric.

Ze względu na wielkość dostępnej pamięci na płytce rozwojowej z układem Stratix II zdecydowano się na rozdzielenie pamięci programu oraz danych obu procesorów. Procesor komunikacyjny (cpu_net) wykorzystywał pamięć DDR SDRAM o pojemności 32 MB a procesor obliczeniowy (cpu_dp), ze względu na krótszy czas dostępu, wykorzystywał 2 MB pamięci SSRAM. Dodatkowo w dostępnej pamięci Flash o pojemności 16MB przechowywane były bootloadery kodu dla obu procesorów. Pamięć Flash wykorzystywana jest również jako pamięć konfiguracji układ FPGA. Przy włączeniu zasilania następuje procedura konfiguracji układu programowalnego, po której bootloadery kodu obu procesorów przenoszą zapisany kod programu do odpowiedniej pamięci RAM.

2.2. Komunikacja pomiędzy procesorami

Opracowany system dwuprocesorowy wykorzystuje pamięć współdzieloną do wymiany informacji pomiędzy rdzeniami. W celu zapewnienia jak najkrótszego czasu dostępu użyto bloków pamięci RAM z zasobów układu FPGA o łącznej pojemności 4kB. Pamięć współdzielona dostępna jest w przestrzeni adresowej obydwu procesorów, w związku z czym jednoczesna próba zapisu do tej samej komórki pamięci przez obydwa procesory może spowodować niespójność danych. Aby wyeliminować jednoczesny dostęp do pamięci i zapewnić tym samym spójność danych, do tej pamięci został przypisany układ sprzętowego semafora – tzw. mutex. Układ ten nie zapewnia jednak samoistnie spójności danych, gdyż zapis do pamięci możliwy jest w dowolnej chwili. Odpowiednio napisane programy obydwu procesorów muszą uwzględniać proces ustawienia semafora przed dostępem do pamięci współdzielonej. Organizacja pamięci współdzielonej została przedstawiona na rys. 2.



Rys. 2. Organizacja wspólnego segmentu pamięci systemu dwuprocesorowego
Fig. 2. Dual processor system memory map

Aby uzyskać dostęp do pamięci współdzielonej procesor musi najpierw sprawdzić czy semafor jest ustawiony, czyli zajęty przez drugi procesor. Jeżeli jest on ustawiony to oczekuje na jego zwolnienie, w przeciwnym przypadku sam go ustawia. Gdy programy obydwu procesorów sprawdzają zajętość semafora przed dostępem do pamięci spójność danych zostaje zachowana. Pamięć współdzielona została podzielona na 1024 pola typu całkowitoliczbowego – integer. Pierwsze 24 pola zostały zarezerwowane do przesyłania flag sterujących procesorem podrzędnym, przekazywania ustawień pomiaru oraz do przechowywania wyników obliczeń statystycznych serii pomiarów. W pozostałych 1000 polach zapisywane są kolejne wyniki serii pomiarowej.

2.3. Precyzyjny licznik czasu

Precyzyjny licznik czasu działa w oparciu o metodę licznikową z wykorzystaniem zegara wielofazowego. W układzie Stratix II dostępnych jest 6 pętli PLL. Jedna z nich pracuje jako syntezer częstotliwości 400 MHz. Sygnał zegarowy 400 MHz podawany jest następnie na 4 pętle PLL, które działają jako przesuwniki fazy. Zespół pętli PLL generuje 16 sygnałów zegarowych przesuniętych względem siebie o 156,25 ps. Każdy z tych sygnałów podawany jest na dwa układy szeregowego licznika 16-bitowego, z których jeden reaguje na zbocze narastające a drugi na zbocze opadające sygnału zegarowego. Po zakończeniu pomiaru wyniki z 32 liczników są sumowane i przekazywane do systemu dwuprocesorowego. Rozdzielczość tak zaprojektowanego układu wynosi ok. 80 ps a maksymalny zakres pomiarowy to 164 μs.

Licznik czasu został dołączony do szyny systemowej Avalon Switch Fabric jako układ podrzędny – Avalon Slave. W systemie zbudowanym w oparciu o magistralę Avalon tylko układy nadrzędne – Avalon Master, mogą dokonywać operacji zapisu lub odczytu na innych układach systemu. Moduł połączenia z szyną Avalon zawiera rejestry, przez które procesor cpu_dp steruje układem licznika czasu.

2.4. Projekt oprogramowania

Procesor Nios II oznaczony jako cpu_net odpowiedzialny jest za komunikację systemu z aplikacją sterującą uruchamianą na zdalnym komputerze. Jako medium komunikacji wybrano sieć Internet oraz stos protokołów TCP/IP. Rozwiązanie takie pozwala na umieszczenie układu na odległym, bezobsługowym stanowisku pomiarowym.

Do wymiany informacji pomiędzy procesorem cpu_net i aplikacją sterującą opracowano własny protokół, który wykorzystuje protokół TCP jako warstwę transportową. Do obsługi stosu protokołów TPC/IP posłużono się biblioteką NicheStack opracowaną przez firmę *InterNiche Technologies Inc*. Oprogramowanie procesora cpu_net implementuje protokół komunikacyjny ze zdalnym hostem – aplikacją sterującą. Aplikacja sterująca może wymusić wykonanie pojedynczego pomiaru, serii pomiarów od 2 do 1000 próbek, przeprowadzenia procedur kalibracyjnych oraz dokonać ustawienia obliczania wybranych statystyk próby pomiarowej. Procesor cpu_net po otrzymaniu odpowiednich komunikatów zleca procesorowi cpu_dp odpowiednie zadania. Po przeprowadzeniu pomiaru wysyła wyniki do aplikacji sterującej. Może to być wynik pomiaru pojedynczego lub serii pomiarów albo też wszystkie zebrane próbki wraz z policzonymi wartościami statystycznymi serii pomiarów.

Procesor cpu_dp w trakcie wykonywania pomiarów zwalnia okresowo dostęp do pamięci współdzielonej, aby procesor cpu_net mógł odczytać liczbę aktualnie pobranych pomiarów. W trakcie wykonywania próby pomiarowej procesor cpu_dp wyznacza aktualne wartości parametrów statystycznych próby takich jak na przykład wartości ekstremalne czy sumę wyników pomiaru. Po zakończeniu pomiaru obliczane są pozostałe wartości, które nie mogły być wyznaczone w trakcie pomiarów – np. przy wyznaczaniu odchylenia standardowego niezbędne jest wcześniejsze obliczenie wartości średniej, które otrzymywane jest dopiero na końcu pomiaru.

2.5. Aplikacja komputerowa

Do sterowania zaprojektowanym systemem przez sieć Internet została opracowana aplikacja dla komputera PC w języku Java. Rozwiązanie to pozwala na uniezależnienie się od platformy sprzętowo-programowej. Aplikacja ta implementuje protokół sterowania system do precyzyjnego pomiaru czasu oraz pozwala na przeprowadzenie zdalnych pomiarów, odczytania wyników oraz graficznego zobrazowania otrzymanych pomiarów. W projekcie zastosowano zbiór klas języka Java – JFreeChart do rysowania histogramu wyników pomiaru odcinku czasu.

3. Implementacja systemu w układzie FPGA Stratix II

Zaprezentowany system został zaprojektowany przy użyciu narzędzi firmy *Altera* oraz przetestowany w układzie programowalnym Stratix II EP2S60F672, dostępnym na płytce rozwojowej Nios II Development Kit Stratix II Edition. Wybór układu podyktowany był ilością pętli PLL dostępnych w układzie umożliwiającą wytworzenie 16. faz sygnału zegarowego dla potrzeb precyzyjnego licznika czasu. Dodatkowo wielkość dostępnych zasobów logicznych układu FPGA umożliwia implementację rozbudowanych systemów mikroprocesorowych.

W tabeli 1 przedstawiono wykorzystanie zasobów układu programowalnego przez zaprojektowany system.

Tab. 1. Wykorzystanie zasobów układu FPGA Stratix II
Tab. 1. Stratix II FPGA resource utilization

Zasób	System 2uP	System 2uP + Licznik Czasu	Dostępne w Stratix II
ALUT	5 120	6 364	48 352
Rejestry	3 392	4 228	48 352
Bloki DSP	16	16	288
Pamięć (bity)	126 464	126 464	2 544 192
Pętle PLL	1	5	6

System dwuprocesorowy stanowi około 80% zasobów zajmowanych przez cały projekt. Dużo większa zajętość zasobów pełnego projektu w porównaniu do miejsca zajmowanego przez układ samego licznika czasu zrekompensowana jest przez możliwość przetwarzania wyników po stronie układu FPGA. Pozwala to na przesyłanie do aplikacji sterującej przetworzonych wyników pomiarów.

Jednakże całkowite wykorzystanie układu programowalnego Stratix II jest stosunkowo niewielkie – około 19% zasobów logicznych i mniej niż 5% wbudowanej pamięci RAM. Możliwe jest więc przeniesienie systemu do mniejszego układu programowalnego co powinno zmniejszyć koszty rozwiązania. Można również wykorzystać więcej wbudowanej pamięci aby wyeliminować jedną lub obydwie pamięci zewnętrzne.

4. Podsumowanie

Współczesne układy FPGA i specjalne oprogramowanie projektowe umożliwiają budowanie zaawansowanych, zintegrowanych systemów cyfrowych. Szczególne miejsce zajmują układy wieloprocesorowe, które wyznaczają kierunek rozwoju przyszłych systemów mikroprocesorowych. Projektowanie systemów wieloprocesorowych ułatwiają procesory programowe (MicroBlaze, Nios II), które realizowane są bezpośrednio z zasobów logicznych układu FPGA. Jednak istotnym problemem takich systemów jest korzystanie z pamięci, komunikacja pomiędzy procesorami oraz podział zadań na poszczególne procesory. Korzystanie z jednej wspólnej pamięci dla danych i programu jest możliwe ale wydłuża się wówczas czas dostępu do pamięci co spowalnia wykonywanie się programu. Lepszym rozwiązaniem jest przydzielenie oddzielnej pamięci dla każdego procesora i korzystanie tylko z pamięci współdzielonej do wzajemnej wymiany informacji. Takie rozwiązanie wymaga stosowania semafora sprzętowego. Sama obsługa semafora (lub semaforów) zajmuje też pewien czas, dlatego wymiana informacji pomiędzy procesorami powinna występować znacznie rzadziej niż czas potrzebny na obsługę semafora. W przeciwnym przypadku stosowanie systemu wieloprocesorowego może nawet zmniejszyć moc obliczeniową całego systemu. Istotnym czynnikiem jest również podział zadań dla procesorów. Zadania te powinny być w sposób czytelny rozdzielone, np. tak jak w zaproponowanym systemie dwuprocesorowym, gdzie jeden procesor zajmuje się sterowaniem i przetwarzaniem danych z precyzyjnego licznika czasu a drugi realizuje zdalną komunikację on-line przez sieć Internet.

Projekt zrealizowanego systemu dwuprocesorowego do sterowania precyzyjnym licznikiem czasu pokazuje, że z powodzeniem można budować zaawansowane układy SoC, bazując na jednym układzie FPGA. Dla licznika czasu uzyskano rozdzielczość ok. 80 ps a błąd pomiaru (odchylenie standardowe) ok. 100 ps. Jest to wynik klasyfikujący wykonany licznik do czołowych rozwiązań światowych. Zastosowanie dedykowanego procesora do przetwarzania danych z licznika pozwala na znaczące przyśpieszenie obliczeń a przez to skrócenie czasu martwego urządzenia. Ponadto wszystkie obliczenia związane z obliczaniem wartości oczekiwanej (na podstawie średniej lub zaawansowanych statystyk, np. silnych estymatorów) mogą być realizowane już w czasie trwania próby pomiarowej i bez konieczności przesyłania jednostkowych wyników do aplikacji sterującej.

5. Literatura

[1] T. Sondej, L. Zagoździński, R. Pełka: Porównanie wydajności sprzętowego i programowego procesora w układzie FPGA Xilinx Virtex-4, Pomiary Automatyka Kontrola, nr 7bis, 2006, s. 20-22

[2] P.Yiannacouras, J.Rose, J.G.Steffan, The microarchitecture of FPGA-Based Soft Processors, Conference on Compilers, Architecture and Synthesis for Embedded Systems, (2005) pp. 202-212

[3] A. A. Jerraya, W. Wolf : Multiprocessor Systems-on-Chips, Morgan Kaufman, 2005.

[4] M.Hübner, K.Paulsson, J.Becker, Parallel and Flexible Multiprocessor System-On-Chip for Adaptive Automotive Applications based on Xilinx MicroBlaze Soft-Cores, Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, (2005) 149a - 149a

[5] P. James-Roxby, P. Schumaher, C. Ross: A Single Program Multiple Data Parallel Processing Platform for FPGAs, Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2004.

[6] T. Sondej, R. Pełka, A. Poniecki: Optimized data processing in precision laser rangefinder with embedded microcontroller, Metrology and Measurement Systems; vol. X, p. 271-286, no. 3/2003.

Artykuł recenzowany