Krzysztof ARNOLD, Sławomir MICHALAK
POLITECHNIKA POZNAŃSKA, WYDZIAŁ ELEKTRONIKI I TELEKOMUNIKACJI,
ul. Polanka 3, 60-965 Poznań

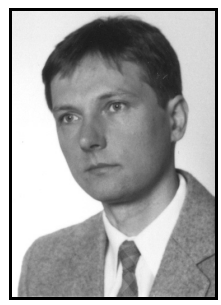
Implementacja kontrolera mikroprocesorowych układów transmisji równoległej w strukturach CPLD

Dr inż. Krzysztof ARNOLD

Absolwent Wydziału Elektroniki Politechniki Gdańskiej. Pracuje jako adiunkt w Katedrze Systemów Telekomunikacyjnych i Optoelektroniki na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. W pracy naukowej zajmuje się problemami pomiarów charakterystyk i parametrów sygnałów stochastycznych, tematyką akwizycji danych w systemach pomiarowych oraz zagadnieniami projektowania, diagnostyki i rozwoju mikroprocesorowych systemów pomiarowych.

e-mail: karnold@et.put.poznan.pl

Dr inż. Sławomir MICHALAK



Pracuje jako adiunkt w Katedrze Systemów Telekomunikacyjnych i Optoelektroniki na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. W pracy naukowo-dydaktycznej zajmuje się zagadnieniami komputerowego wspomagania projektowania, symulacji układów elektronicznych, programowaniem układów mikroprocesorowych i układów programowalnych. Zajmuje się tematyką pozyskiwania informacji z inteligentnych czujników pomiarowych.

e-mail: michalak@et.put.poznan.pl

Streszczenie

W pracy przedstawiono możliwości zwiększania liczby wejść i wyjść równoległych w systemach mikroprocesorowych z wykorzystaniem programowalnych układów peryferyjnych. Wskazano ograniczenia w zakresie rozszerzania portów równoległych i zaproponowano rozwiązanie problemu multi-liniowej komunikacji mikrokontrolerów z otoczeniem przez sterowanie zewnętrznych specjalizowanych układów peryferyjnych z poziomu kontrolera CPLD, odpowiedzialnego za dekodowanie adresów wejścia/wyjścia i przyjmowanie zgłoszeń przerwań.

Słowa kluczowe: systemy mikroprocesorowe, transmisja równoległa, układy CPLD, dekodowanie adresów, kontroler przerwań.

Implementation of a hardware controller of microprocessor PPI devices in CPLD structures

Abstract

In this paper the possibility of increasing parallel inputs and outputs in microprocessor systems with a programmable peripheral interface (PPI) is presented. The requirements and restrictions associated with expanding parallel ports for microprocessors with internal bus and microprocessors with external access memory are described. The basic system with a central processor unit and parallel transmission device(s) is described (Fig. 1) and parallel interface modes for 82C55A PPI are shown (Figs. 2, 3). An example of multi-channel communication between a microcontroller and external units, with hardware CPLD controller and PPI devices, is given. The controller is responsible for input/output address decoding and interrupts receiving (Fig. 4). The external address/interrupt controller minimizes the time required by the microcomputer for interruption of the current program, servicing of the peripheral units, and resumption of the interrupted program. The basic requirements for programmable devices working as controllers in input/output parallel integrated subsystems are shown. The controller was implemented in one of XC9500XL family devices (Tab. 1). For each device from this family the I/Os are fully 5V (CMOS, TTL) tolerant even though the core power supply is 3.3 volts. In mixed (5V/3.3V/2.5V) systems a controller can work with low power supply microprocessors (Fig. 5). Use of this one programmable device gives us a chance for creating a flexible controller (Fig. 6) which can work with different kinds of 8-bit central units.

Keywords: microprocessor systems, parallel transmission, CPLD, address decoding, interrupt controller.

1. Wstęp

Projektowanie systemów cyfrowych w znacznym stopniu opiera się obecnie na wykorzystaniu mikroprocesorów lub układów programowalnych, rozróżnianych w kontekście odmiennej architektury i specyfiki działania. Nie stanowi to przeszkody w poprawie parametrów użytkowych systemów, w których mikroprocesory i układy programowalne mogą efektywnie współpracować, wypełniając właściwe dla siebie zadania.

Dotyczy to również problemu komunikacji jednostki centralnej systemu mikroprocesorowego z kilku lub nawet kilkunastu urządzeniami zewnętrznymi, zwłaszcza w przypadku prowadzenia wielokanałowej transmisji równoległej. Konieczność taka pojawia się podczas sterowania polami odczytowymi, klawiaturą i przetwornikami c/a, a także przyjmowania informacji z przetworników a/c, zadajników stanów logicznych i terminali adresowych. Problem rośnie, jeśli transfer danych ma przebiegać z dużą szybkością, zgodnie z protokołem transmisji z potwierdzeniem lub z możliwie małym obciążeniem czasu jednostki centralnej.

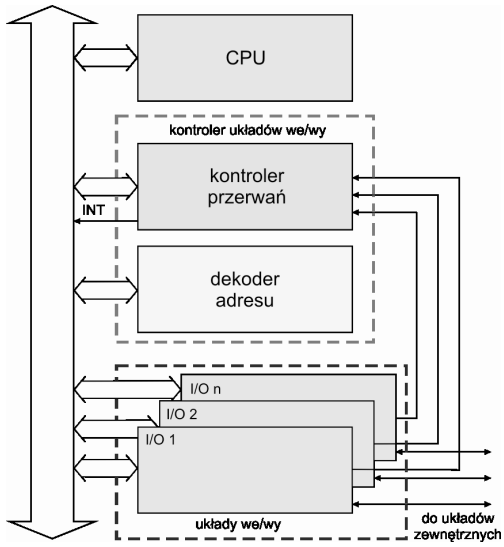
Podstawowym wyróżnikiem możliwości komunikacyjnych mikroprocesorów w trybie transmisji równoległej jest dostępność portów wejścia/wyjścia. W często wykorzystywanych 8-bitowych mikroprocesorach RISC, takich jak ATmega16A, jest ona zwykle ograniczona do czterech portów (32 linie wejścia/wyjścia) [1]. To zbyt mało, by bezpośrednio sprostać wymaganiom komunikacji wielokanałowej. Rozbudowane układy, do których należą ATmega128A [2] i ATmega2560 [3], dysponują większą liczbą portów (odpowiednio siedem portów/53 linie i aż jedenaście portów/86 linii). Nawet one jednak nie obsługują sprzętowo transmisji równoległej z potwierdzeniem. Mechanizm ten funkcjonuje natomiast w specjalizowanych układach PPI (*Programmable Peripheral Interface*) [4], pozwalając na zwiększenie liczby portów i zwolnienie mikroprocesora z zadań obsługi protokołów transmisji. Wiodące układy peryferyjne, w tym PPI, zaimplementowano w strukturach PLD [5, 6], wbudowano jako podsystemy niektórych układów VLSI i stworzono dla nich narzędzia symulacyjne [7], ale magistralowa współpraca typowego mikrokontrolera z układem PPI nadal wymaga użycia dodatkowej logiki sterującej. Stopień złożoności takiego rozwiązania można znakomicie zmniejszyć, implementując nie tylko dekoder adresu, ale i optymalizowany sterownik przerwań w układzie programowalnym.

2. Mikroprocesorowe podsystemy wejścia -wyjścia z układami PPI

Układy transmisji równoległej PPI współpracują z typową magistralą systemową. Warunki jej tworzenia zależą od możliwości dostępu mikroprocesora do zewnętrznych układów peryferyjnych i pamięci danych. Mikrokontrolery z zamkniętą magistralą własną wymagają przydziału portów do magistrali zewnętrznej i programowego wsparcia zewnętrznych cykli zapisu i odczytu. Mikroprocesory wyposażone w funkcję dostępu posiadają natomiast dedykowane porty magistralowe i sprzętowe wsparcie operacji zapisu i odczytu zewnętrznych lokacji adresowych. W obu przypadkach dołączanie układów PPI do magistrali systemowej wymaga dekodowania ich adresów, zapewnienia cykli zapisu i odczytu oraz sprzętowej kontroli przerwań.

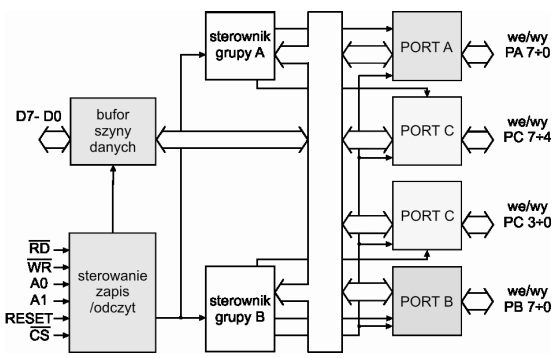
Za wybór układu PPI odpowiada systemowy lub lokalny dekoder adresu. Zaadresowany i zaprogramowany układ PPI przejmuje obsługę transmisji równoległej, a o jej zakończeniu może poin-

formować jednostkę centralną (CPU) za pomocą przerwań. Kontroler przerwań ma za zadanie przyjmowanie zgłoszeń, określanie ich poziomu oraz generację przerwania dla CPU i wystawienie wektora zgłoszenia o najwyższym priorytecie (rys. 1).



Rys. 1. Schemat blokowy podsystemu wejścia/wyjścia
Fig. 1. Block diagram of the I/O subsystem

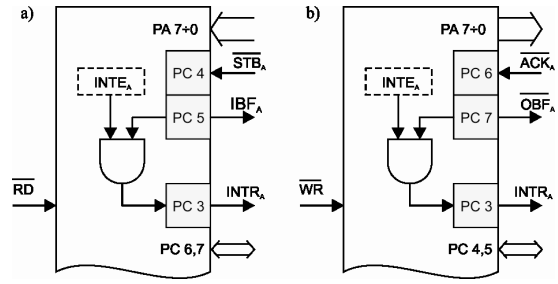
Większość podstawowych 8-bitowych mikroprocesorów ma kilka wejść przerwań zewnętrznych. Rozbudowa podsystemu wejścia/wyjścia z układami PPI wymaga wówczas zastosowania zewnętrznego kontrolera przerwań. Rozszerzony podsystem PPI przynosi jednak wymierne korzyści. Standardowe układy 82C55A PPI [8, 9] są wyposażone w trzy porty PA, PB, PC (rys. 2) i mogą pracować w kilku trybach. Najczęściej wykorzystywany tryb 0 odnosi się do transmisji danych bez potwierdzenia.



Rys. 2. Architektura układu 82C55A PPI
Fig. 2. Hardware architecture of the 82C55A PPI

Znacznie większe możliwości zapewniają tryby 1 i 2. W trybie 1 możliwe jest prowadzenie 8-bitowej transmisji równoległej z potwierdzeniem w portach PA i PB, w zaprogramowanym wcześniej kierunku. W trybie 2 port PA jest zdolny do prowadzenia dwukierunkowej transmisji z potwierdzeniem, podczas gdy port PB może pracować w trybie 0 lub 1. Tak w trybie 1, jak i 2, linie portu PC są wykorzystywane jako sterujące.

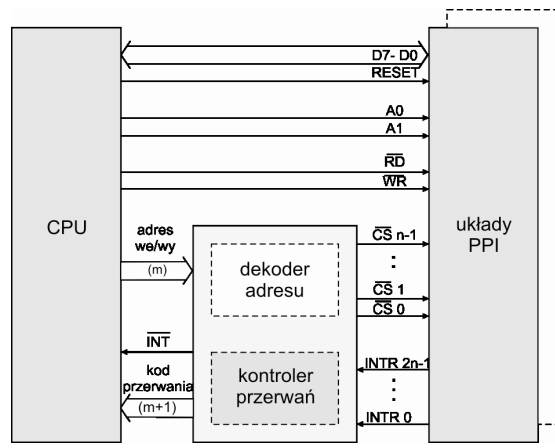
Port zaprogramowany w trybie 1 jako wejściowy informuje jednostkę centralną przerwaniem INTR o skompletowaniu znaku przychodzącego w buforze odbiorczym (rys. 3a). W przypadku konfiguracji portu jako wyjścia przerwanie INTR jest generowane po wysłaniu znaku (rys. 3b). Oba przerwania są kasowane sprzętowo, a realizacja protokołu transmisji z potwierdzeniem jest w całości zapewniona przez układ PPI [8, 9]. Połączenie funkcji dekodera adresowego i kontrolera przerwań w obrębie jednej struktury programowalnej może uprościć organizację podsystemu wejścia/wyjścia i pozwolić tym samym na efektywne wykorzystanie zalet układów PPI.



Rys. 3. Schemat funkcjonalny portu A w trybie 1: a) wejście, b) wyjście
Fig. 3. Functional diagram of Port A in Mode 1: a) input, b) output

3. Koncepcja realizacji reprogramowalnego kontrolera układów PPI

Jednostka centralna komunikuje się z układami PPI przez magistralę danych D7-D0 (rys.4). Bity adresu A1 i A0 służą do wyboru rejestrów wewnętrznych PPI, natomiast m-bitowa starsza część adresu jest dekodowana na 1 z n sygnałów wyboru układu PPI ($n=2^m$). W trybach 1 i 2 każdy z n układów PPI może generować dwa przerwania INTR (od portów PA i PB). Zintegrowany kontroler adresu i przerwań powinien więc przyjmować 2n zgłoszeń INTR i generować przerwanie dla CPU, wraz z towarzyszącym wektorem aktualnie najważniejszego zgłoszenia (rys. 4). Wskazane jest zapewnienie obsługi n≥4 układów PPI. Kontroler powinien też współpracować z otoczeniem zasilanym napięciem 5V.



Rys. 4. Ogólny schemat blokowy podsystemu PPI z implementowanym kontrolerem adresu i przerwań
Fig. 4. General block diagram of the PPI subsystem with implemented address/interrupt controller

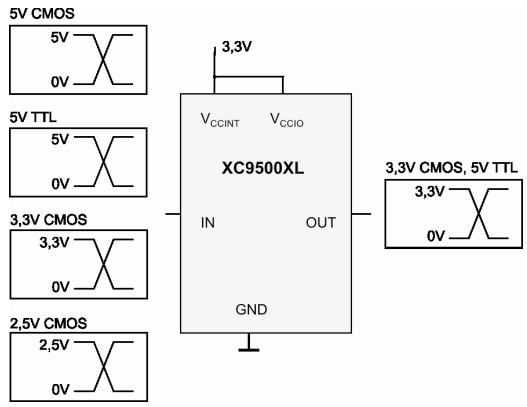
4. Implementacja sprzętowa kontrolera

Strukturę kontrolera adresu/przerwań zaimplementowano w układzie reprogramowalnym rodziny XC9500XL [10]. Do opisu działania układu i programowania wykorzystano język Verilog oraz środowisko Xilinx ISE 11 [11]. Zarówno układ dekodera adresu, jak i układ enkodera priorytetowego realizowane są jako stosunkowo proste struktury kombinacyjne, zatem przedstawione rozwiązanie zajmuje relatywnie niewiele zasobów wewnętrznych i może z powodzeniem zostać zaimplementowane również w innych, niezbyt złożonych strukturach GAL, PLD lub CPLD. Wybór konkretnego układu programowalnego jest natomiast zdeterminowany przez liczbę dostępnych wyprowadzeń, niezbędnych do obsługi wymaganej liczby układów PPI (przerwań). Docelowo kontroler zaprogramowano w układzie XC9572XL [12], ale może on zostać zaimplementowany w układzie tej samej rodziny o mniejszych zasobach. W tabeli 1 przedstawiono porównanie zajętości zasobów dla realizacji w układach XC9572XL i XC9536XL.

Tab. 1. Porównanie zajętości zasobów układów XC9572/36XL
Tab. 1. Comparison of resources for XC9572/36XL devices

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
XC9572XL				
12/72 (17%)	17/360 (5%)	0/72 (0%)	23/34 (68%)	18/216 (9%)
XC9536XL				
12/36 (34%)	17/180 (10%)	0/36 (0%)	23/34 (68%)	14/108 (13%)

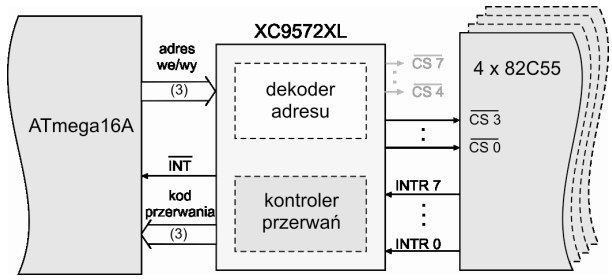
Napięciowe poziomy logiczne układów rodziny XC9500XL są kompatybilne względem kilku technologii. W podstawowej konfiguracji układ zasilany jest napięciem 3,3V, ale od strony wejść i wyjść może pracować z typowymi układami TTL i CMOS (5V), a także z układami niskonapięciowymi CMOS (3,3V lub 2,5V), bez konieczności stosowania dodatkowych układów dopasowujących wartości poziomów logicznych (rys. 5) [10].



Rys. 5. Kompatybilność poziomów logicznych układów XC9500XL [10]
Fig. 5. Compatibility of logic levels for XC9500XL devices [10]

5. Wyniki eksperymentów

Testy funkcjonalne układów logicznych, zaimplementowanych w strukturze XC9572XL, przeprowadzono w środowisku sprzętowym składającym się z mikrokontrolera ATmega16A, czterech układów transmisji równoległej 82C55A PPI i sprzętowego symulatora ośmiu urządzeń zewnętrznych. Mikrokontroler ATmega16A i układ XC9572XL programowano w trybie ISP, przy wsparciu z poziomu dwóch komputerów klasy PC. Implementacja testowanej wersji kontrolera adresu i przerwań objęła logikę dekodera adresowego, wystawiającego sygnały wyboru ośmiu układów peryferyjnych oraz logikę enkodera priorytetowego, przyjmującego i hierarchizującego 8 zgłoszeń przerwań. Pozwoliło to na pełną obsługę czterech dołączonych do magistrali systemowej jednostek PPI, z uwzględnieniem przerwań zgłaszanych przez komplet portów PA i PB. Pozostałe dostępne sygnały wyboru układów peryferyjnych (rys. 6) można dedykować czterem dodatkowym jednostkom PPI, wykorzystywanym do pracy w trybie 0 lub innym układom wejścia/wyjścia. Program testowy napisano w języku asemblera. Mikrokontroler inicjował cztery układy PPI w trybie 1 wyjściowym, z odblokowaniem przerwań od wszystkich portów, a następnie rozsyłał dane do rejestrów nadawczych w kanałach transmisji. Symulator urządzeń zewnętrznych potwierdzał przyjęcie danych w sekwencjach o różnej kolejności i odstępach czasowych. Pozwoliło to na weryfikację poprawności działania dekodera adresu i enkodera priorytetowego CPLD przy obsłudze pojedynczego przerwania, kilku przerwań przychodzących jednocześnie i zagnieżdżaniu zgłoszeń przerwań. Wyniki testów potwierdziły prawidłowe funkcjonowanie kontrolera zaimplementowanego w strukturze XC9572XL.



Rys. 6. Schemat blokowy testowanego podsystemu PPI z kontrolerem adresu i przerwań zaimplementowanym w układzie XC9572XL
Fig. 6. Block diagram of the tested PPI subsystem with address/interrupt controller implementation in XC9572XL circuit

6. Podsumowanie

Połączenie zadań dekodera adresowego i sprzętowego kontrolera przerwań w obrębie jednego układu programowalnego zmniejsza stopień komplikacji układowej magistralowych systemów mikroprocesorowych. Pozwala to w szczególności na efektywne wykorzystanie podsystemów z układami PPI, przez zwiększenie liczby kanałów transmisji równoległej i jednoczesne odciążenie jednostki centralnej. W przypadku N kanałów transmisji w układach 82C55A jednostka centralna jest zobowiązana do przyjmowania jednego przerwania zewnętrznego i wektora przerwań o długości $\log_2 N$ bitów. Oznacza to przykładowo możliwość korzystania z 16 kanałów, zdolnych do prowadzenia transmisji równoległej z potwierdzeniem, w ramach obsługi jednego przerwania z przyporządkowanym 4-bitowym wektorem. Każdy z kanałów dostosowuje się przy tym indywidualnie do szybkości transmisji urządzenia zewnętrznego.

Warto zauważyć, że implementacja dekodera adresowego i kontrolera przerwań w strukturze CPLD zapewnia współpracę nie tylko układów PPI, ale w ogólności podsystemów wejścia/wyjścia, z większością standardowych 8-bitowych mikrokontrolerów, wyposażonych w kilka portów.

7. Literatura

[1] ATmega16A. 8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash. Atmel Corporation 2009.
[2] ATmega128A. 8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash. Atmel Corporation 2011.
[3] ATmega640/1280/1281/2560/2561. 8-bit ATmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash. Preliminary. Atmel Corporation 2011.
[4] Ray A.K., Bhurchandi K.M.: Advanced Microprocessors and Peripherals. Tata McGraw-Hill, New Delhi 2007.
[5] Microperipheral MegaCore Function. Data Book, Altera 1997.
[6] D8255 Programmable Peripheral Interface v1.00. Digital Core Design, http://dcd.pl
[7] Pradeep Kumar Jaisal, Anant G. Kulkarni, Srikant B. Burje: Design and Simulation of 8255 Programmable Peripheral Interface Adapter using VHDL. International Journal of Computer Science and Technology, IJCST Vol.2, Issue 1, March 2011.
[8] 82C55A Data Sheet FN2969.10. Intersil 2006.
[9] WS 82C55A CMOS Programmable Peripheral Interface. Wing Shing Computer Components Corporation 2008.
[10] XC9500XL High-Performance CPLD Family Data Sheet. Product Specification. Xilinx 2009.
[11] Hajduk Z.: Wprowadzenie do języka Verilog. BTC, Legionowo 2009.
[12] XC9572XL High Performance CPLD. Product Specification. Xilinx 2007.

otrzymano / received: 26.04.2012
przyjęto do druku / accepted: 01.06.2012 *artykuł recenzowany / revised paper*