

Krzysztof ARNOLD, Sławomir MICHALAK

POLITECHNIKA POZNAŃSKA, WYDZIAŁ ELEKTRONIKI I TELEKOMUNIKACJI,
ul. Polanka 3, 60-965 Poznań

Implementacja kontrolera SPI/PPI w układzie CPLD dla zastosowań z mikrokomputerem Raspberry Pi

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 wspomaganie 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 wskazano potrzebę rozszerzenia portów równoległych w systemach z komputerem Raspberry Pi. Przedstawiono możliwości programowalnych układów peryferyjnych (PPI). Zaproponowano rozwiązanie problemu komunikacji komputera Raspberry z układami peryferyjnymi, polegające na wprowadzeniu kontrolera układów PPI, wyposażonego w interfejs SPI. Przedstawiono koncepcję integracji modułu SPI, dekodera adresów układów PPI i enkodera przerwań w strukturze CPLD. Zaprezentowano wyniki implementacji logiki kontrolera w układzie CPLD.

Słowa kluczowe: Raspberry Pi, podsystem transmisji równoległej, kontroler magistrali, układy CPLD.

Implementation of the SPI/PPI controller in a CPLD structure for the systems with a Raspberry Pi microcomputer

Abstract

In this paper the necessity of increasing parallel I/O lines in a microprocessor system based on the Raspberry Pi, a very popular small educational microcomputer, is described. The possibility of increasing parallel inputs and outputs with a programmable peripheral interface (PPI), and also hardware architecture of 82C55A is presented (Fig. 1). The cooperation rules between the central unit and PPI devices are given. The general block diagrams of the system with the Raspberry Pi and the implemented PPI controller, with parallel and serial (SPI) interface, are shown and discussed (Fig. 2). An effective solution of communication between Raspberry Pi and 82C55A PPI devices, using an external programmable controller with a simplified SPI, is suggested. The SPI/PPI controller takes over the duties of controlling the modes of 82C55A PPI devices, relieves the central unit and decreases the usage of I/O lines. The instruction sequences are composed of two bytes, the higher one includes codes for creating control signals for the controller and read/write cycles for 82C55A devices, the lower includes the data. The address and data words are written with the confirmation signal /STR (Fig. 3). The controller is responsible for PPI addressing and also takes over the tasks of decoding and receiving interrupts. This version of a controller was implemented in the XC9572XL (Xilinx) device, so we had additionally I/Os fully 5V (CMOS, TTL) tolerant, even though the core power supply of the Raspberry Pi and its I/O lines were 3.3 volts. The Behavioral and Post-Fit Simulations results are shown (Figs. 4,5 and 6).

Keywords: Raspberry Pi, PPI subsystem, bus controller, CPLD.

1. Wstęp

Mikrokomputer Raspberry Pi został opracowany przez brytyjską organizację Raspberry Pi Foundation, wspieraną przez University of Cambridge Laboratory i Broadcom Corporation. Głównym celem fundacji jest promowanie nauczania technik komputerowych i programowania w szkołach. Zgodnie z tym przesłaniem Raspberry Pi powstał jako relatywnie tani edukacyjny komputer osobisty [1] i zdążył już zyskać bardzo dużą popularność. Stało się to możliwe

dzięki połączeniu małych rozmiarów jednopłytkowego modułu komputera ze znaczną mocą obliczeniową. Te właściwości Raspberry Pi sprawiają, że idea szerokiego wykorzystywania małych, łatwo dostępnych i coraz lepiej przetwarzających dane komputerów osobistych będzie sukcesywnie rozwijana [2]. Podobnie jak komputery PC, Raspberry Pi może być wykorzystywany nie tylko do nauki programowania, ale i do sterowania procesami oraz dokonywania pomiarów. Jego z założenia niewielkie rozmiary zapewniają mobilność, nie pozwalają jednak na dowolne rozszerzanie wbudowanych zasobów i rozmieszczanie złącz. Z tego względu pojawiają się już moduły towarzyszące, które zwiększają funkcjonalne możliwości mikrokomputera [3]. Komunikacja pomiędzy nimi a modułem Raspberry Pi ma więc istotne znaczenie tak dla zamierzonych, jak i pojawiających się perspektyw rozwojowych projektu.

Obecnie Raspberry Pi dysponuje tylko 17 typowymi liniami I/O, co w oczywisty sposób ogranicza obszar jego potencjalnych zastosowań [4]. Funkcjonalność komputerów Raspberry można jednak znacznie zwiększyć, przyłączając do nich opcjonalnie programowalne mikroprocesorowe układy transmisji równoległej PPI (*Peripheral Parallel Interface*). Architektura układów 82C55A PPI została opracowana pod kątem w pełni sprzętowego wsparcia komunikacji systemu z urządzeniami zewnętrznymi i odciążenia jednostki centralnej. Pod tym względem układy PPI nie znajdują jeszcze odpowiedników wśród modułów towarzyszących, dedykowanych Raspberry Pi. Wprowadzenie podsystemu PPI będzie efektywne przy zaimplementowaniu kontrolera układów 82C55A w strukturze programowalnej [5]. Ważne jest przy tym rozważenie wariantu implementacji, zakładającego minimalne zaangażowanie linii I/O Raspberry Pi w komunikację z kontrolerem podsystemu PPI.

2. Konfiguracje komputerów Raspberry Pi

Moduł Raspberry Pi wykorzystuje układ Broadcom BCM2835 typu SoC (*ang. System-on-a-chip*), który zawiera kompletny system mikroprocesorowy. W skład systemu wchodzi procesor ARM1176JZF-S taktowany zegarem 700 MHz, układ graficzny VideoCore IV GPU oraz 256 MB (Model A) lub 512 MB (Model B) pamięci RAM [6].

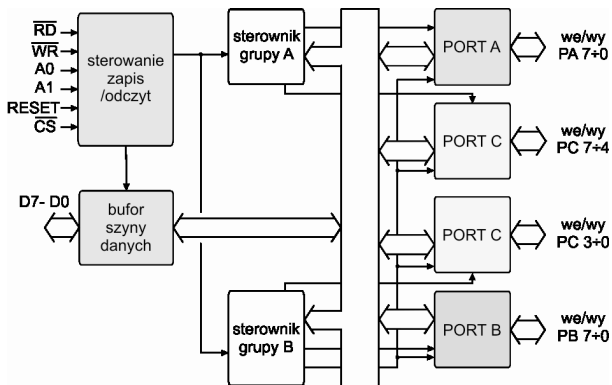
Raspberry Pi posiada złącze dla karty pamięci SD, na której instalowany jest system operacyjny i przechowywane są dane. Urządzenie wyposażono w jeden port USB (Model A) lub w dwa porty USB i kartę 10/100 Ethernet (Model B). Porty USB pozwalają na dołączenie urządzeń zewnętrznych, w tym klawiatury, myszy, pamięci FLASH i dysków twardej. Karta Ethernet umożliwia dołączenie do internetu. Komputer ma wyjście sygnału wideo HDMI oraz Composite Video, a także wyjście sygnału audio. Raspberry Pi pracuje pod jedną z wersji systemu operacyjnego Linux (np. Debian GNU, Fedora, Arch Linux). Moduł wyposażony jest w 17 linii GPIO (*General Purpose Input/Output*),

dostosowanych do napięcia 3,3V. Są to wyprowadzenia układu BCM2835, wymagające dopasowania poziomów napięć przy współpracy z układami o zasilaniu 5V, często spotykanymi w zastosowaniach dydaktycznych. Układ BCM 2835 nie jest zabezpieczony od strony złącza GPIO. Większość linii posiada funkcje alternatywne (np. SPI, I²C, UART).

3. Rozszerzenie portów równoległych

Komputer Raspberry PI dysponuje niewielką liczbą linii I/O, które ponadto wymagają programowej obsługi. Znacznie większe możliwości mają specjalizowane układy transmisji równoległej PPI, wykonywane w technologii CMOS [7] i implementowane w strukturach PLD [8], przy czym wersje PLD nie zapewniają kompatybilności z pierwowzorem.

Układ 82C55A PPI (rys. 1) wnosi do systemu trzy 8-bitowe porty PA, PB i PC, zapewniające transfer danych do lub z urządzenia zewnętrznego bez potwierdzenia (tryb 0), albo dwa porty PA i PB, obsługujące sprzętowo, z wykorzystaniem portu PC jako linii sterujących, transmisję równoległą z potwierdzeniem w zaprogramowanych kierunkach (tryb 1). Ponadto port PA może prowadzić dwukierunkową transmisję half-duplex z potwierdzeniem (tryb 2), podczas gdy PB pracuje w trybie 0 lub 1 [7, 8].



Rys. 1. Architektura układu 82C55A PPI
Fig. 1. Hardware architecture of the 82C55A PPI

W trybach 1 i 2 port wyjściowy może generować przerwanie do jednostki centralnej po wysłaniu znaku i otrzymaniu potwierdzenia, a port wejściowy po przyjęciu znaku i potwierdzeniu odbioru.

Bezpośrednia obsługa jednego układu 82C55A wymaga zaangażowania 14 (bez przyjmowania przerw) lub 16 linii I/O jednostki Raspberry PI (rys. 1). Sprzętowe i programowe obciążenie Raspberry PI wymaga więc wprowadzenia kontrolera, odpowiedzialnego za sterowanie układów 82C55A oraz wyznaczanie wektorów przerw w ramach przyjętej hierarchii zgłoszeń.

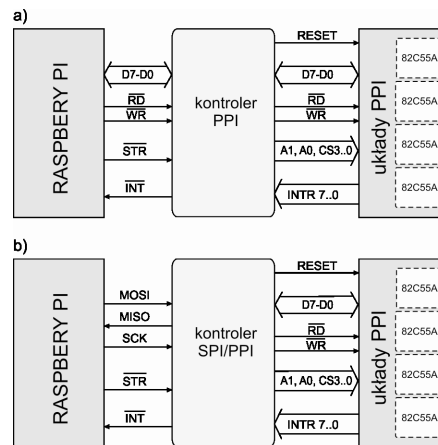
4. Możliwości komunikacji Raspberry PI z kontrolerem układów PPI

Komputer Raspberry PI może komunikować się z kontrolerem podsystemu PPI z wykorzystaniem linii I/O jako interfejsu równoległego lub szeregowego. Interfejs równoległy obejmuje dwukierunkową, multiplexowaną magistralę D7-D0 oraz sygnały odczytu /RD, zapisu /WR, strobu /STR i przerwania /INT [5]. Pozwala to na dołączenie czterech układów 82C55A do Raspberry PI, ale zajmuje 12 linii I/O komputera (rys. 2a).

Alternatywne funkcje linii I/O mikrokomputera Raspberry PI umożliwiają dostęp do interfejsów szeregowych SPI, I²C i UART. Ponieważ układ 82C55A może pracować w cyklach maszynowych, wyznaczanych przez kontroler z częstotliwością 2MHz [7], konieczny jest wybór szybkiego interfejsu SPI [9], który w mikrokomputerach Raspberry PI jest taktowany sygnałem uzyskanym z zegara 250 MHz, teoretycznie ze współczynnikiem podziału programowanym od wartości minimalnej równej 2. Należy jednak

liczyć się z przerwami w transmisji, powodowanymi przez działanie systemu operacyjnego Linux.

Architektura kontrolera SPI/PPI może opierać się na rozwiązaniu, które opracowano dla mikroprocesorów AVR [10]. Kontroler musi jednak przejmować wszystkie sygnały, pełniąc rolę translatora poziomów napięć pomiędzy Raspberry PI a układami 82C55A. Wykorzystanie linii MOSI, MISO i SCK interfejsu SPI może zmniejszyć zajętość portu I/O Raspberry PI do 5 linii (rys. 2b).

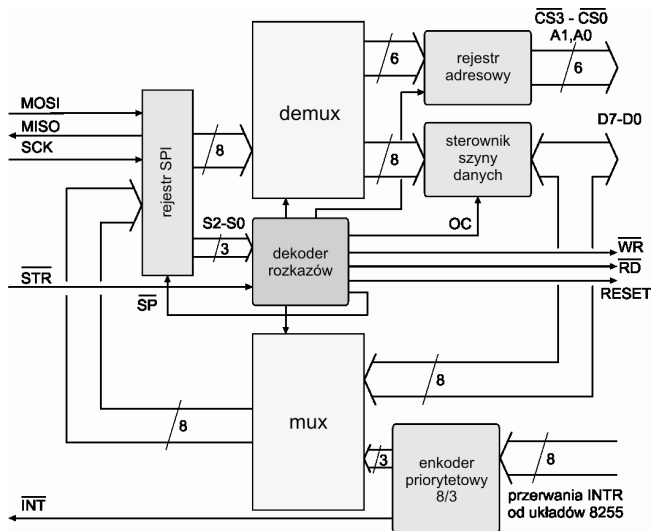


Rys. 2. Schemat blokowy systemu z komputerem Raspberry PI i implementowanym kontrolerem podsystemu PPI:
a) z interfejsem równoległym, b) z interfejsem SPI

Fig. 2. General block diagram of the microcomputer system with the Raspberry PI and the implemented PPI controller:
a) with parallel interface, b) with SPI

5. Implementacja sprzętowa kontrolera SPI/PPI w układzie CPLD

Proponowane rozwiązanie kontrolera podsystemu PPI zawiera rejestr nadajnika/odbiornika SPI, pracujący w trybie *slave*, gdy urządzeniem *master* jest nadrzędny Raspberry PI (rys. 3). Bajty polecenia i danych wprowadzane są przez wejście MOSI, w takt sygnału SCK. Bajty danych i kod przerwania o najwyższym priorytecie są wyprowadzane przez wyjście MISO. Zmianę stanu podsystemu PPI sygnalizuje komputerowi przerwanie /INT.



Rys. 3. Architektura kontrolera SPI/PPI
Fig. 3. Architecture of the SPI/PPI controller

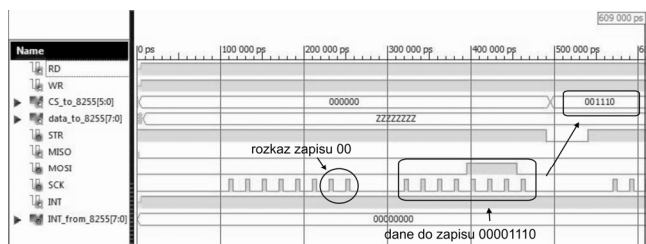
Dekoder rozkazów jest strobowanym układem kombinacyjnym, który steruje pozostałymi blokami w zależności od trybu pracy, określonego przez kod S2-0. W trybie zerowania sygnał strobowujący /STR powoduje aktywizację linii RESET i ustawienie podsys-

temu PPI w stan początkowy. W trybie adresowania demultiplekser kieruje bity CS3-0 (wybór układu 82C55A) oraz A1, A0 (programowanie lub wybór portu) do rejestru adresowego, sterującego 6-bitową magistralą adresową. W trybie zapisu danych demultiplekser kieruje bajt do sterownika szyny danych, sygnał /OC powoduje wystawienie bajtu na dwukierunkową magistralę D7-0, a na linii /WR jest generowany sygnał zapisu wybranego układu i rejestru wewnętrznego 82C55A. W trybie odczytu danych dekodery rozkazów wystawia sygnał /RD, multiplekser kieruje dane do rejestru SPI, a impuls /SP wymusza wpis równoległy, po którym dane mogą być przekazane do Raspberry PI.

Odczyt i zapis kolejnych danych przez Raspberry PI odbywa się w obsłudze przerwania /INT. Enkoder priorytetowy przyjmuje 8 przerwania INTR7-0, zgłaszanych przez układy 82C55A, koduje numer zgłoszenia o aktualnie najwyższym priorytecie (hierarchia rośnie od INTR0 do INTR7) i aktywizuje linię /INT. W odpowiedzi Raspberry PI programuje tryb odczytu numeru przerwania. Sygnał /STR inicjuje przekazanie przez multiplekser wektora przerwania i jego wpis równoległy do rejestru SPI. Po szeregowym odczytaniu zawartości rejestru SPI Raspberry PI identyfikuje zdarzenie, które trzeba obsłużyć w pierwszej kolejności.

6. Wyniki implementacji

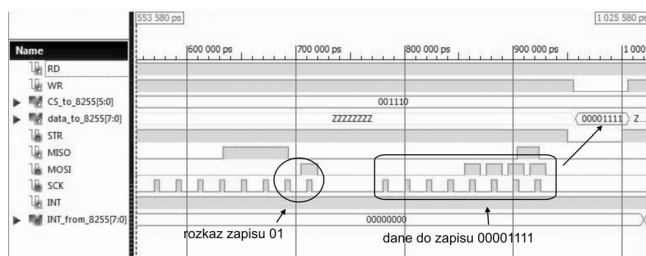
Strukturę kontrolera SPI/PPI zaimplementowano w układzie XC9572XL, należącym do rodziny XC9500 (Xilinx), która umożliwia współpracę z układami o różnych poziomach napięć [11]. Do opisu działania układu i do jego programowania wykorzystano język Verilog oraz środowisko Xilinx ISE 11. Implementowana struktura zajmuje około 25% zasobów wewnętrznych układu XC9572XL i wykorzystuje 30 spośród 34 pinów I/O.



Rys. 4. Przykład symulacji cyklu zapisu adresu dla implementacji kontrolera w strukturze XC9572XL

Fig. 4. An example of address write simulation for controller implementation using the XC9572XL device

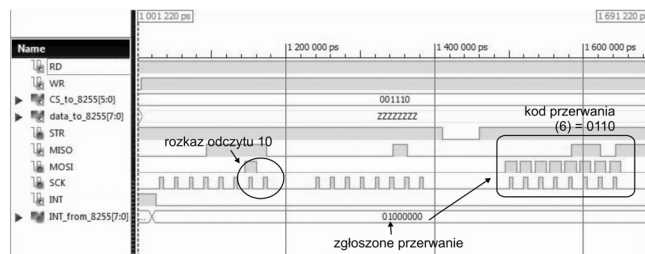
Działanie kontrolera zasymulowano (rys. 4,5,6) w środowisku Xilinx ISim 11 dla układu XC9572-5-VQ44 (*Behavioral Simulation i Post-Fit Simulation*), z wykorzystaniem języka Verilog.



Rys. 5. Przykład symulacji cyklu zapisu danych dla implementacji kontrolera w strukturze XC9572XL

Fig. 5. An example of data write simulation for controller implementation using the XC9572XL device

Testy funkcjonalne bloków logicznych kontrolera, zaimplementowanych w strukturze XC9572XL, przeprowadzono w środowisku składającym się z komputera Raspberry PI, oscyloskopu cyfrowego oraz zadajnika i monitora stanów logicznych, zastępujących podsystem PPI. Pozwoliło to na potwierdzenie poprawności działania kontrolera w trybie pracy krokowej, z możliwością obserwacji zagnieżdżenia przerwania priorytetowych.



Rys. 6. Przykład symulacji cyklu odczytu wektora przerwania dla implementacji kontrolera w strukturze XC9572XL

Fig. 6. An example of interrupt vector read simulation for controller implementation using the XC9572XL device

7. Podsumowanie

Właściwości komputerów Raspberry PI otwierają im drogę nie tylko do zastosowań edukacyjnych, ale również pomiarowych i kontrolno – sterujących. Ograniczeniem w tym względzie jest mała liczba linii I/O komputera. Dołączanie układów PPI do systemu pozwala na zwiększenie liczby portów i jednocześnie zapewnienie obsługi protokołów transmisji z potwierdzeniem przez hardware, przy czym każdy z kanałów dostosowuje się indywidualnie do szybkości urządzenia zewnętrznego.

Bezpośrednia obsługa jednego układu 82C55A PPI wymaga jednak zaangażowania nie mniej niż 14 z 17 dostępnych linii I/O Raspberry PI, a do obsługi czterech układów potrzeba aż 25 linii. Wprowadzenie kontrolera CPLD z komunikacją równoległą pozwala na dołączenie czterech układów 82C55A przy wykorzystaniu 12 linii I/O komputera, a proponowany kontroler z interfejsem SPI może obsługiwać te układy, a więc 8 portów prowadzących transmisję równoległą z potwierdzeniem lub 12 portów pracujących bez potwierdzenia, przy zmniejszeniu zaangażowania portu I/O Raspberry PI do 5 linii. Pozwala to na opcjonalne dołączenie do Raspberry PI kolejnych urządzeń, z zastrzeżeniem, że szybkość chwilowa ich obsługi zależy również od systemu operacyjnego komputera i może zmieniać się w czasie.

Praca realizowana w ramach tematu 08/83/DSPB/4709.

8. Literatura

- [1] Upton E., Halfacree G.: Meet the Raspberry Pi. Wiley, 2012.
- [2] Mitchell G.: The Raspberry Pi single-board computer will revolutionise computer science teaching. *Engineering Technology*, vol. 7, no. 3, p. 26, 2012.
- [3] Johnson, G.D., Krusienski, D.J.: A low-cost configurable multichannel cortical stimulator prototype. NER'2013, 6th International IEEE/EMBS Conference on Neural Engineering, November, 2013, p. 641-644.
- [4] Bizony P.: The Raspberry Pi single-board computer won't revolutionise computer science teaching. *Engineering Technology*, vol. 7, no. 3, p. 27, 2012.
- [5] Arnold K., Michalak S.: Kontroler CPLD dla magistrali zewnętrznej mikrokomputera Raspberry PI. *Pomiary Automatyka Kontrola*, vol. 59, nr 8/2013, s.766-768.
- [6] BCM 2835 ARM Peripheral. Product Specification, Broadcom Europe Ltd. 2012.
- [7] 82C55A Data Sheet FN2969.10. Intersil 2006.
- [8] Microperipheral MegaCore Function. Data Book, Altera 1997.
- [9] Jamro E., Wielgosz M., Cioch W., Bieniasz S.: Efektywna komunikacja ARM-FPGA z użyciem interfejsu SPI. *Pomiary Automatyka Kontrola*, vol. 57, nr 8/2011, s.874-876.
- [10] Arnold K., Michalak S.: Programowalny kontroler mikroprocesorowych układów transmisji równoległej z interfejsem SPI. *Pomiary Automatyka Kontrola*, vol.59, nr 8/2013, s.803-805.
- [11] XC9572XL High Performance CPLD. Product Specification. Xilinx 2007.