

Krzysztof ARNOLD, Sławomir MICHALAKPOLITECHNIKA POZNAŃSKA, WYDZIAŁ ELEKTRONIKI I TELEKOMUNIKACJI,
ul. Polanka 3, 60-965 Poznań**Kontroler CPLD dla magistrali zewnętrznej mikrokomputera 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 naukowej i dydaktycznej zajmuje się zagadnieniami komputerowego wspomaganego projektowania i symulacji układów elektronicznych (ECAD), programowaniem układów mikroprocesorowych i układów programowalnych. Interesuje 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 popularnego modułu Raspberry Pi z wykorzystaniem programowalnych układów peryferyjnych (PPI). Wskazano na ograniczenia w zakresie rozszerzania portów równoległych, związane z liczbą dostępnych linii I/O Raspberry Pi. Zaproponowano rozwiązanie wykorzystujące zewnętrzny kontroler implementowany w strukturze CPLD, odpowiedzialny za adresowanie układów PPI, przyjmowanie zgłoszeń przerwania i wystawienie na magistralę wektora przerwania.

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

A CPLD controller for the external bus of a Raspberry Pi microcomputer**Abstract**

In this paper the possibility of increasing parallel inputs and outputs in a microprocessor system based on an educational microcomputer Raspberry Pi (Fig. 1) and a programmable peripheral interface (PPI) is presented. An example of multi-channel communication between the central processor unit, PPI devices and parallel interface modes for 82C55A PPI is described (Fig. 2). A system composed of a Raspberry Pi unit, a hardware CPLD controller and four parallel transmission devices is proposed (Fig. 3). The external address/interrupt controller is responsible for external PPI addressing – the word address is written with a confirmation signal /LOAD (Fig. 5), and for registering interrupts. The controller takes over the tasks of decoding and interrupts receiving, so consequently minimizes the time required by the Raspberry Pi for interruption of the current program, servicing of the peripheral units, and resumption of the interrupted program. The data bus can be implemented inside the CPLD, but also it is possible to use one of any external level translators with three-state output mode (OE signal). This 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 of the Raspberry Pi is 3.3 volts. In mixed (5V/3.3V/2.5V) systems, a controller can work with low power supply CPU. Use of this one programmable device gives us a chance for creating a flexible controller, which can work with different kind of 8-bit central units.

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

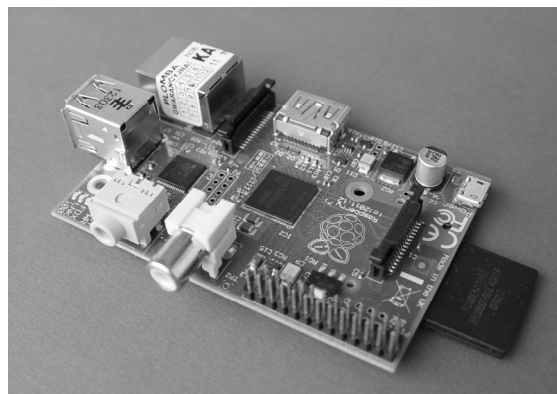
1. Wstęp

Raspberry Pi to niewielki, jednopłytkowy moduł komputerowy o wymiarach 86 x 56 mm i wadze 45 g., wielkością zbliżony do rozmiaru karty kredytowej. Opracowany został przez Raspberry Pi Foundation – brytyjską organizację charytatywną (wspieraną przez University of Cambridge Laboratory i Broadcom Corporation), której głównym celem jest promowanie nauczania technik komputerowych i programowania w szkołach. Zgodnie z tą myślą Raspberry Pi został zaprojektowany jako relatywnie tani komputer edukacyjny [1]. Obecny postęp technologiczny skłania do przypuszczenia, że idea szerokiego wykorzystywania małych, łatwo dostępnych i dysponujących coraz większą mocą obliczeniową

komputerów osobistych będzie rozwijana [2]. Zadania edukacyjne obejmują jednak szereg różnorodnych zagadnień, a dotychczasowa droga rozwoju komputerów PC dowiodła już ich przydatności w wielu dziedzinach. Raspberry Pi może być w przyszłości wykorzystywany obok innych narzędzi [3] nie tylko do nauki programowania, ale i do sterowania procesami czy dokonywania pomiarów. Jego z założenia niewielkie rozmiary nie pozwalają jednak na dowolne rozmieszczanie złączy. W chwili obecnej Raspberry Pi ma tylko 17 typowych linii I/O, co ogranicza obszar jego potencjalnych zastosowań [4]. Przy wykorzystaniu tych linii można jednak znacznie zwiększyć funkcjonalność komputerów Raspberry Pi, zapewniając rozszerzenie równoległych portów I/O, przyłączanych jako opcjonalny moduł towarzyszący z układami PPI (*Peripheral Parallel Interface*). Rozwiązanie takie będzie efektywne przy zaimplementowaniu sterownika układów PPI w strukturze programowalnej [5].

2. Właściwości komputera Raspberry Pi

Moduł oparty jest na układzie Broadcom BCM2835 typ SoC. Ogólnie pojęciem SoC (*ang. System-on-a-chip*) określa się struktury zawierające kompletny system mikroprocesorowy, a więc jednostkę centralną CPU, bloki pamięci FLASH lub ROM, EEPROM i RAM, układy cyfrowe i analogowe, a także przetworniki analogowo-cyfrowe i cyfrowo-analogowe oraz interfejsy szeregowe. Układy SoC mogą zawierać kilka jednostek centralnych (MPSoC - *ang. Multiprocessor System-on-Chip*). Przykładem jednostek typu SoC są układy oparte na procesorach o architekturze ARM. W odróżnieniu od mikrokontrolerów, układy SoC są wyposażone w jednostkę centralną o stosunkowo dużej mocy obliczeniowej, umożliwiając uruchamianie systemów operacyjnych (np. Linux, Android). Widok modułu Raspberry Pi (Model B) przedstawiono na rys. 1.



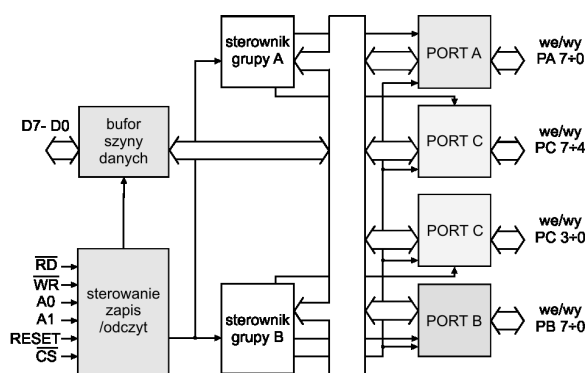
Rys. 1. Moduł komputerowy Raspberry Pi (Model B)
Fig. 1. The computer module Raspberry Pi (Model B)

Układ Broadcom BCM2835 zawiera 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 kart pamięci SD. Na karcie pamięci instalowany jest system operacyjny, jest ona również używana do przechowywania danych (spełnia zatem rolę „twardego dysku”). Urządzenie wyposażono w jeden port USB (Model A), lub w dwa porty USB oraz kartę 10/100 Ethernet (Model B). Porty USB pozwalają na dołączenie urządzeń zewnętrznych, np. klawiatury, myszy, pamięci FLASH, dysków twardek, itp. Karta Ethernet umożliwia dołączenie do internetu. Urządzenie posiada 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*). Są to bezpośrednio wyprowadzenia układu BCM2835, co nie jest najlepszym rozwiązaniem w zastosowaniach dydaktycznych. Oprócz typowej roli portów I/O niektóre linie posiadają alternatywne funkcje (np. SPI, I²C, UART). Dostosowane są do napięcia 3,3 V, zatem ich współpraca z układami TTL lub CMOS, zasilanymi napięciem 5 V, wymaga zastosowania dodatkowych układów dopasowujących. Rolę taką może pełnić również odpowiednio dobrany układ reprogramowalny (np. układy CPLD rodziny XC9500 Xilinx), który oprócz swej podstawowej funkcji (struktura reprogramowalna, w której implementowany jest dany projekt), dodatkowo umożliwia współpracę z układami o różnych standardach poziomów napięć i w pewnym stopniu zabezpiecza przed uszkodzeniem porty I/O jednostki centralnej (układ Broadcom BCM2835).

3. Układy PPI w systemie magistralowym

Układy transmisji równoległej 82C55A PPI (rys. 2) komunikują się z jednostką centralną za pośrednictwem 8-bitowej magistrali, a z urządzeniami zewnętrznymi przez 8-bitowe porty PA, PB i PC, które mogą pracować w jednym z trzech trybów pracy [7, 8].



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

Tryb 0 obejmuje nadawanie lub odbiór danych bez potwierdzenia. W trybie 1 możliwe jest prowadzenie 8-bitowej transmisji równoległej z potwierdzeniem w portach PA i PB, w kierunku zaprogramowanym wcześniej dla każdego portu. W trybie 2 port PA prowadzi dwukierunkową transmisję z potwierdzeniem, podczas gdy PB może pracować w trybie 0 lub 1. W trybach 1 i 2 linie portu PC są wykorzystywane do sterowania transmisją. Port wyjściowy układu 82C55A PPI generuje przerwanie INTR do jednostki centralnej po wysłaniu znaku i otrzymaniu potwierdzenia, a port wejściowy po przyjęciu znaku i potwierdzeniu odbioru. Oba przerwania są kasowane sprzętowo, a realizacja protokołu transmisji z potwierdzeniem jest w całości zapewniona przez układ 82C55A PPI.

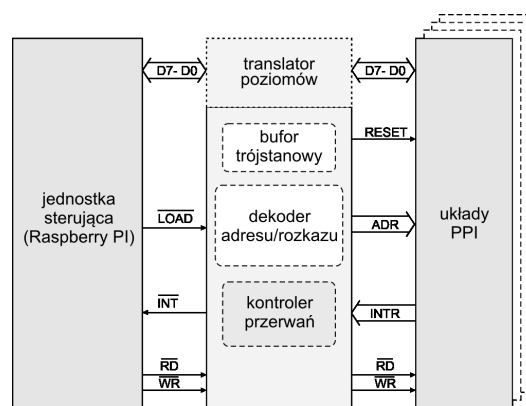
Współpraca modułu Raspberry PI z układem peryferyjnym może odbywać się z pomocą linii I/O, którym przypisano rolę

równoległej, 8-bitowej magistrali danych, a także kilku linii sterujących. Rozbudowa podsystemu PPI (współpraca Raspberry PI z kilkoma układami peryferyjnymi) bez angażowania kolejnych linii I/O wymaga zastosowania zewnętrznego kontrolera, który przejmie zadania adresowania układów 82C55A PPI i przyjmowania zgłoszeń przerwania. Odciąża to komputer, ale przede wszystkim zmniejsza liczbę angażowanych linii I/O jednostki centralnej stosownie do jej możliwości.

4. Koncepcja kontrolera współpracującego z modułem Raspberry PI

Komputer Raspberry PI komunikuje się jako jednostka centralna systemu z kontrolerem i układami 82C55A PPI przez dwukierunkową, 8-bitową multipleksowaną magistralę adresu i danych (rys. 3). Na magistrali tej wystawiane jest słowo adresowe, wpisywane przez komputer do kontrolera. Umożliwia ono wprowadzenie układów PPI w stan początkowy (RESET) i wybór obsługiwanego układu PPI oraz jego rejestrów wewnętrznych (linie ADR). Zapisem słowa adresowego steruje sygnał /LOAD. Słowa programujące i bajty danych D7-D0 są przekazywane do wybranego układu PPI za pośrednictwem translatora poziomów napięć. Dwukierunkowy translator, sterowany przez sygnały zezwolenia (/OE) lub wyboru kierunku (DIR), dostosowuje również napięciowe poziomy logiczne bloku PPI do wymagań jednostki Raspberry PI w cyklu odczytu danych lub słowa statusowego. W cyklach zapisu i odczytu układów peryferyjnych PPI aktywne są odpowiednio sygnały sterujące /WR i /RD.

Układy PPI komunikują się z urządzeniami zewnętrznymi, przejmując całkowicie odpowiedzialność za realizację protokołów transmisji z potwierdzeniem. Kontroler jest informowany o przebiegu transmisji za pośrednictwem przerwania INTR i na tej podstawie generuje przerwanie /INT do komputera oraz wystawia na żądanie wektor przerwania na magistralę danych.

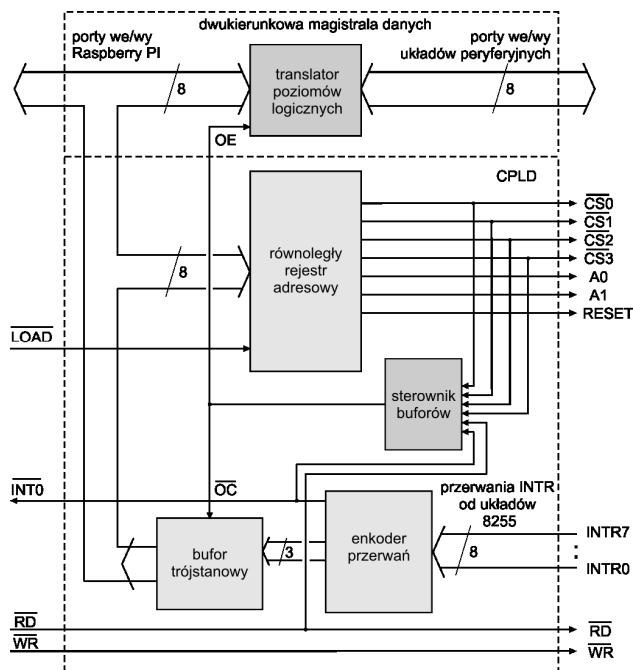


Rys. 3. Schemat blokowy systemu z komputerem Raspberry PI i implementowanym kontrolerem układów PPI

Fig. 3. General block diagram of the microcomputer system with Raspberry PI and implemented PPI controller

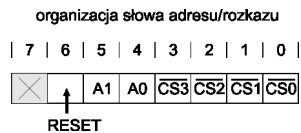
5. Implementacja sprzętowa kontrolera w strukturze CPLD

W proponowanym układzie kontrolera (rys. 4) w strukturze CPLD zaimplementowano rejestr słowa adresowego, priorytetowy enkoder przerwania z wyjściowym buforem trójstanowym i sterownik bufora, tworząc dwukierunkową magistralę danych. W układach CPLD można również zintegrować translator napięciowych poziomów logicznych. Podczas zapisu słowa adresowego wykorzystuje się 8, a podczas odczytu wektora przerwania 3 bity magistrali. Adres układu peryferyjnego wystawiany jest przez Raspberry PI na magistralę i zatraskiwany w rejestrze CPLD opadającym z boczem sygnału /LOAD.



Rys. 4. Architektura kontrolera adresu i przerwań
Fig. 4. Architecture of the address/interrupt controller

Słowo adresowe (rys. 5) zawiera oprócz bitów wyboru układu 82C55A (/CS3-/CS0) i adresu portu (A1, A0) także bit zerowania RESET, aktywny w stanie wysokim. Dla układów 82C55A sygnał RESET jest nadrzędny w stosunku do pozostałych sygnałów.



Rys. 5. Format słowa adresowego
Fig. 5. The address word format

Priorytetowy enkoder przerwań przyjmuje zgłoszenia, generuje przerwanie dla jednostki Raspberry PI i wystawia na jej żądanie wektor zgłoszenia o najwyższym priorytecie. Zgłoszenie przez układy peryferyjne przerwania sygnalizowane jest na wyjściu kontrolera niskim poziomem logicznego sygnału /INT. Adresy ośmiu przerwań kodowane są z zachowaniem priorytetu na trzech bitach. Odczyt adresu przerwania jest poprzedzony wystawieniem na magistrali słowa zawierającego „1” logiczne dla sygnałów /CS. Wprowadza to stan wysokiej impedancji na liniach danych wszystkich układów peryferyjnych. Jednocześnie sygnały /CS, /INT i /RD sterują przez układ kombinacyjny otwarciem buforów trójstanowych od strony Raspberry PI (sygnał /OC), umożliwiając odczyt wektora przerwań przez komputer.

Podczas zapisu i odczytu danych z układów peryferyjnych wykorzystywane są sygnały /WR i /RD, generowane na liniach I/O modułu Raspberry PI.

6. Wyniki implementacji

Strukturę kontrolera adresu/przerwań zaimplementowano w układzie reprogramowalnym XC9536XL [9]. Do opisu działania układu i jego implementacji wykorzystano język Verilog oraz środowisko Xilinx ISE 11. Wersja kontrolera z układem XC9536XL angażuje większość wyprowadzeń (tab. 1) i nie obejmuje translatora, dopasowującego napięciowe poziomy logiczne mikrokomputera Raspberry PI i magistrali danych interfejsu PPI.

Dopasowanie napięciowe pozostałych sygnałów realizuje układ XC9536XL. Jest on zasilany napięciem 3,3V, podobnie jak moduł Raspberry PI, 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). Z tego względu linie /WR i /RD poprowadzono przez strukturę CPLD (rys. 4).

Tab. 1. Wykorzystanie zasobów układu XC9536XL
Tab. 1. Use of resources of XC9536 device

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
XC9572XL				
19/36 (53%)	41/180 (23%)	8/36 (23%)	30/34 (89%)	25/108 (24%)

Testy funkcjonalne kontrolera, zaimplementowanego w strukturze XC9536XL, przeprowadzono w środowisku sprzętowym składającym się z modułu Raspberry PI, towarzyszącego translatora 8-bitowej magistrali danych ST2378 i sprzętowego symulatora ośmiu urządzeń zewnętrznych. Programy testowe zrealizowano w językach C oraz Python.

7. Podsumowanie

Założenia sformułowane dla projektu Raspberry PI wskazują, że sprzętowe wsparcie tego modułu powinno odbywać się z zastosowaniem struktur programowalnych. Zaimplementowany układ kontrolera łączy funkcje dekodera adresów i priorytetowego kontrolera przerwań, pozwalając na dołączenie do komputera Raspberry PI i efektywne wykorzystanie zaawansowanych układów PPI, w różnych wersjach wykonania. W szczególności umożliwia to wyposażenie Raspberry PI w większą liczbę kanałów transmisji równoległej bez obciążenia jednostki centralnej. Reprogramowalna struktura kontrolera może być modyfikowana i rozwijana przy zwiększaniu liczby układów peryferyjnych. Zintegrowanie translatora poziomów magistrali danych w obrębie kontrolera pozwoli na współpracę mikrokomputera Raspberry PI i modułów PPI za pośrednictwem jednego chipu.

Warto zauważyć, że zaproponowana implementacja kontrolera w strukturze CPLD stwarza możliwość współpracy modułu Raspberry PI nie tylko z układami PPI, ale w ogólności z podsystemami wejścia/wyjścia.

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] Łazoryszczak M.: Mikroprocesor PicoBlaze na platformie CPLD w dydaktyce systemów wbudowanych. Pomiar Automatyka Kontrola, vol. 58, nr 7/2012, s.638-640.
- [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.: Implementacja kontrolera mikroprocesorowych układów transmisji równoległej w strukturach CPLD. Pomiar Automatyka Kontrola, vol.58, nr 7/2012, s.635-637.
- [6] BCM 2835 ARM Peripheral. Product Specification, Broadcom Europe Ltd. 2012.
- [7] 82C55A Data Sheet FN2969.10. Intersil 2006.
- [8] Zargari A., Combs M.S.: Construction, interfacing, and application of an 8255-based programmable peripheral interface card. Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Conference. Proceedings, pp.119-123, 2001.
- [9] XC9536XL High Performance CPLD. Product Specification. Xilinx 2007.

otrzymano / received: 21.05.2013

przyjęto do druku / accepted: 03.07.2013

artykuł recenzowany / revised paper