

## Szeregowy interfejs komunikacyjny dla układów FPGA serii Virtex

Dr inż. Adam ŁUCZAK

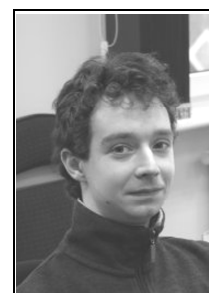
(ur. 1972) tytuł magistra uzyskał w 1997 na Politechnice Poznańskiej. Od roku 1997 prowadził badania w zespole przetwarzania obrazów. W roku 2001 uzyskał tytuł doktora. Od roku 2002 jest adiunktem w Katedrze Telekomunikacji Multimedialnej i Mikroelektroniki. Główna działalność badawcza związana jest ze sprzętową implementacją algorytmów kompresji i przetwarzania sekwencji wizyjnych w programowalnych układach FPGA. Obecnie głównym nurtem zainteresowań są systemy przetwarzania rozproszonego



e-mail: aluczak@multimedia.edu.pl

Mgr inż. Maciej KURC

Ukończył studia na Politechnice Poznańskiej, wydziale Elektroniki i Telekomunikacji, kierunku Elektronika i Telekomunikacja w 2008r. Jest doktorantem na wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. Jego zainteresowania to elektronika, przetwarzanie obrazów, implementacja algorytmów przy pomocy układów FPGA.



e-mail: mkurc@multimedia.edu.pl

Mgr inż. Jakub SIAST

Tytuł magistra uzyskał w 2009r. na Politechnice Poznańskiej na wydziale Elektroniki i Telekomunikacji na kierunku Elektroniki i Telekomunikacji. Obecnie jest doktorantem w Katedrze Telekomunikacji Multimedialnej i Mikroelektroniki. Głównie zainteresowania to sieci NoC oraz układy FPGA.



e-mail: jsiastr@multimedia.edu.pl

### Streszczenie

Artykuł przedstawia implementację sprzętową szeregowego interfejsu komunikacyjnego dla układów FPGA firmy Xilinx z serii Virtex. Rozwiązanie opiera się na wbudowanych w układy tej serii moduły SERDES i jest dedykowane dla zastosowań wymagających dużych przepływności. Interfejs charakteryzuje się skalowalnością, oraz możliwością pracy w osobnej domenie częstotliwościowej. Proponowane moduły zostały przetestowane w symulacjach, oraz w układzie sprzętowym.

**Słowa kluczowe:** SERDES, FPGA, komunikacja między układami

### Serial interface for Virtex series FPGA devices

#### Abstract

Following article presents hardware implementation of serial communication interface for Xilinx Virtex series programmable devices. Proposed solution is based on embedded SERDES modules of these devices and shows practical realization of fast interface for multimedia purposes, where high bitrates are required. Interface is scalable and has ability to operate in separate clock frequency domain which allows flexible modification of its parameters according to project requirements. Architecture of receiver and transmitter is presented in part 2. It also describe way of dividing transmitted data into transition flits, method of ensuring synchronization and gives theoretical throughput of developed link.

Proposed modules were tested in simulations and in hardware device, what is presented in part 3. Synthesis results for different FPGA's are presented in tables 1 and 2. Next application model is described incorporating video camera as a source of transmitted signal and starter boards with Virtex FPGAs as processing devices. The interface is used for connecting boards. It's proper work is proved in visual observation of transmitted and processed video data presented on LCD displays mounted on 2 system's starter boards. False rate level for transmitted data was also computed and are presented in table 3.

**Keywords:** SERDES, FPGA, inter-chip communication

### 1. Wstęp

Artykuł przedstawia implementację szeregowego interfejsu komunikacyjnego dla układów Virtex firmy Xilinx, która jest jednym z czołowych liderów w branży FPGA. Celem tej pracy było stworzenie interfejsu mogącego obsługiwać strumienie danych o różnych prędkościach bitowych, w tym również nieskompresowane strumienie wizyjne, w coraz bardziej popularnej wysokiej rozdzielczości, których przykładowa prędkość to 0,6 Gbps<sup>1</sup>. W układach Virtex dostępne są specjalizowane moduły Rocket IO, które pozwalają na transmisję z prędkością do 8,5Gbps, jednak występują one jedynie w małej liczbie w specjalizowanych seriach (seria T dla układów Virtex4[1] i seria LX dla Virtex5 [2]), a ich obecność podnosi koszt układu nawet o ponad 50%[3]. Alternat-

tywą dla tych transceiverów są sprzętowe serializery i deserializery (w skrócie SERDES), które są standardowym wyposażeniem we wszystkich układach Virtex 4 i 5. Pojedynczy kanał zbudowany w oparciu o moduł SRERDES pozwala na przesłanie danych z prędkości około 0,7 Gbps, czyli mniejszej niż uzyskiwanej za pomocą RocketIO. Jednak niewątpliwą zaletą modułów SERDES jest ich duża liczba przypadająca na pojedynczy układ (znajdują się w każdym bloku wejścia/wyjścia wymienionych układów). Pozwala to na wprowadzenie skalowalności interfejsu, poprzez wykorzystanie różnej liczby modułów SERDES w zależności od potrzeb konkretnego systemu. Pomimo bogatej dokumentacji i dużej liczby not aplikacyjnych, jakie udostępnia firma Xilinx nie można odnaleźć w nich opisu skalowalnego interfejsu o potencjalnie dużej prędkości transmisji i przeznaczonego dla wszystkich układów Virtex z serii 4 i 5. W punkcie 2 opisane zostały szczegóły opracowanego przez nas wariantu takiego interfejsu, a punkt 3 przedstawia jego parametry rzeczywiste, które uzyskał w fizycznej realizacji, jako część systemu przetwarzania obrazu.

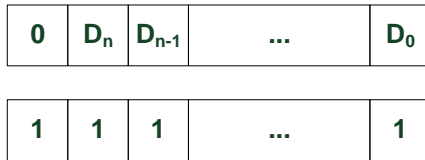
### 2. Architektura

Łącze transmisyjne składa się z modułu nadawczego oraz odbiorczego. Każdy z nich zawiera pojedynczą część odpowiedzialną za synchronizację oraz N części odpowiedzialnych za serializację oraz transmisję danych przez pojedynczy kanał. Zarówno moduł nadawczy jak i odbiorczy zostały wyposażone w kolejki FIFO pełniące role buforów oraz zapewniające synchronizację danych pomiędzy różnymi domenami częstotliwościowymi. Jest to niezbędne w sytuacji, gdy moduły te są taktowane innym zegarem niż reszta matrycy FPGA. Ze względu na brak funkcji odzyskiwania zegara ze strumienia danych w modułach SERDES, sygnał zegarowy jest przesyłany do odbiornika przez dedykowany kanał.

Moduł nadawczy dokonuje podziału słów danych, które mają zostać przesłane na krótsze tzw. słowa transmisyjne, o długości akceptowanej przez moduł SERDES pojedynczego kanału. Każde słowo dzielone jest na mniejsze fragmenty nadawane równolegle przez kolejne kanały. Podział następuje także w czasie (fragment słowa dzielony jest na jeszcze mniejsze części nadawane kolejno po sobie w ramach jednego kanału). Podczas szeregowej transmisji danych krytyczne jest zachowanie synchronizacji pomiędzy nadajnikiem i odbiornikiem. Odbiornik musi dysponować infor-

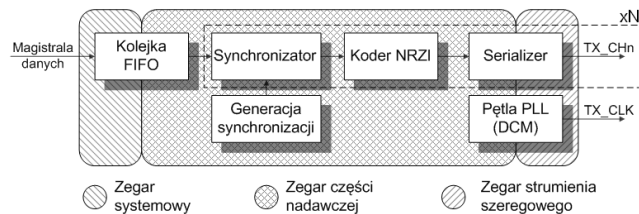
<sup>1</sup> Strumień w rozdzielczości 1920x1080 z prędkością 25 obrazów na sekundę i o schemacie próbkowania chrominancji 4:2:0.

macją, o położeniu granicy pomiędzy kolejnymi słowami w strumieniu bitowym. Informacja ta jest przesyłana za pomocą specjalnego słowa synchronizacyjnego, wstawianego co określony odstęp czasu pomiędzy słowami danych. Ułożenie bitów w słowie synchronizacyjnym jest unikalne i nie może wystąpić w żadnym słowie danych. Zastosowaną realizację kodowania danych i synchronizacji przedstawia rysunek 1.



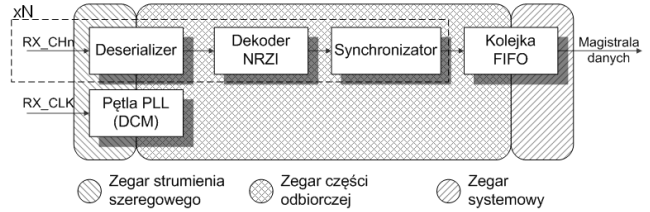
Rys. 1. Słowo danych (górne) i synchronizacji (dolne).  
Fig. 1. Data word (upper) and synchronization word (lower).

Słowa synchronizacyjne wstawiane są jednocześnie we wszystkich kanałach transmisyjnych. Ich funkcją, oprócz synchronizacji bitowej, jest przenoszenie informacji o aktywności łącza dla modułu odbiorczego. Przed wysłaniem danych przez kanał transmisyjny konieczne jest zastosowanie kodowania kanałowego. W prezentowanym rozwiązaniu zastosowano kod NRZI. Kod ten zapewnia usunięcie składowej stałej z sygnału, co poprawia znacznie jego właściwości transmisyjne przy większych odległościach pomiędzy nadajnikiem a odbiornikiem. Tak zakodowany sygnał poddawany jest serializacji we wbudowanym module SERDES a następnie jest wyprowadzany na zewnątrz układu FPGA. Ponieważ moduł SERDES wymaga taktowania docelowym zegarem strumienia szeregowego, do jego generacji zastosowana została pętla PLL (również w postaci wbudowanego modułu) pełniąca funkcję mnożnika częstotliwości. Rysunek 2 przedstawia schemat blokowy części nadawczej. Bloki funkcjonalne odpowiadające pojedynczemu kanałowi transmisyjnemu zaznaczone zostały linią przerywaną.



Rys. 2. Schemat blokowy części nadawczej.  
Fig. 2. Block schematic of the transmitter part

Moduł odbiorczy składa się z  $N$  części obsługujących pojedyncze kanały transmisyjne oraz pojedynczej części zapewniającej synchronizację strumienia bitowych i detekcję błędów. Strumień bitowy każdego kanału trafia na wejście modułu SERDES, który dokonuje jego zrównoleglenia na słowa o ustalonej długości. Następnie, słowa te przetwarzane są przez dekodery NRZI. Przed scaleniem słów transmisyjnych ze wszystkich kanałów odtwarzana jest ich granica podziału. Ta operacja odbywa się w synchronizatorze i opiera się na wykryciu wcześniej opisanego słowa synchronizacji. Na wyjściu synchronizatorów w module odbiorczym podział strumienia bitowego na słowa transmisyjne jest taki sam jak na wejściu synchronizatorów w module nadawczym. Przed zapisaniem do wyjściowej kolejki FIFO następuje konkatencja słów transmisyjnych ze wszystkich kanałów (z wyjść ich synchronizatorów), co kończy operację odbioru pierwotnego słowa danych. Wszystkie kanały składowe łącza odbiorczego pracują synchronicznie na wspólnym zegarze, który pochodzi z dedykowanego kanału zegarowego. Przed wejściem do układu sygnał zegarowy jest "czyszczony" przez moduł pętli PLL ze wszystkich zakłóceń, jakie mogły powstać na drodze nadajnik - odbiornik. Schemat blokowy części odbiorczej przedstawia rysunek 3.



Rys. 3. Schemat blokowy części odbiorczej.  
Fig. 3. Block schematic of the receiver part

Przebiegłość łącza może zostać policzona zgodnie ze wzorem 1.

$$B = f_{clk} \cdot M \cdot R \cdot N [\text{bit} / \text{s}], \quad (1)$$

gdzie  $f_{clk}$  to częstotliwość zegara taktującego układ nadajnika,  $M$  to mnożnik dla częstotliwości wyjściowej (współczynnik serializacji będący parametrem modułu SERDES i zawierający się w granicach od 2 do 5 dla trybu DDR i 2 do 8 dla SDR [1][4]),  $R$  określa czy dane są nadawane w konwencji SDR, czy DDR i jest równy odpowiednio 1, lub 2. Zmienna  $N$  określa liczbę kanałów użytą do zbudowania łącza.

### 3. Model aplikacyjny

Implementacja sprzętowa ma na celu zweryfikowanie właściwości zaproponowanego rozwiązania. Wykorzystano interfejsy SERDES taktowane zegarem  $f_{clk}$  o częstotliwości 50 MHz, z transmisją w systemie DDR. Teoretyczna przepływność takiego interfejsu może być obliczona zgodnie ze wzorem 1 i wynosi 0,5 Gbps/kanał, natomiast opisany we wstępie strumień wizyjny wysokiej rozdzielczości wymaga przepływności 0,6 Gbps. Dodatkowo przesłanie komend zastosowanego protokołu sieciowego powiększa ten strumień o około 15%, co łącznie daje strumień o prędkości bitowej 0,7 Gbps. Aby przepustowość interfejsu była większa niż prędkość strumienia użyto  $N = 2$  kanałów, co dało 1 Gbps przepływności łącza.

Wyniki syntezy opracowanych interfejsów zebrane zostały w tabelach 1 i 2. Syntezę przeprowadzono za pomocą środowiska Xilinx ISE 11.3, osobno dla modułu nadawczego i odbiorczego. Częstotliwość maksymalna przedstawiona w tabeli odnosi się do prędkości pracy modułów FPGA i nie uwzględnia ograniczenia maksymalnej częstotliwości sygnału, jaka jest akceptowana przez bloki wejścia/wyjścia matrycy FPGA.

Tab. 1. Wyniki syntezy modułu nadawczego  
Tab. 1. Synthesis results of the transmitter module

Układ FPGA	$N$	$f_{clk}$ [MHz] (maksymalna)	Liczba LUT	Liczba rejestrów
Virtex 4 XC4V45X35-12	1	214.9	252	189
	2	211.5	325	228
Virtex 5 XC5V55X35T-2	1	377.8	211	185
	2	316.8	275	224

Tab. 2. Wyniki syntezy modułu odbiorczego  
Tab. 2. Synthesis results of the receiver module

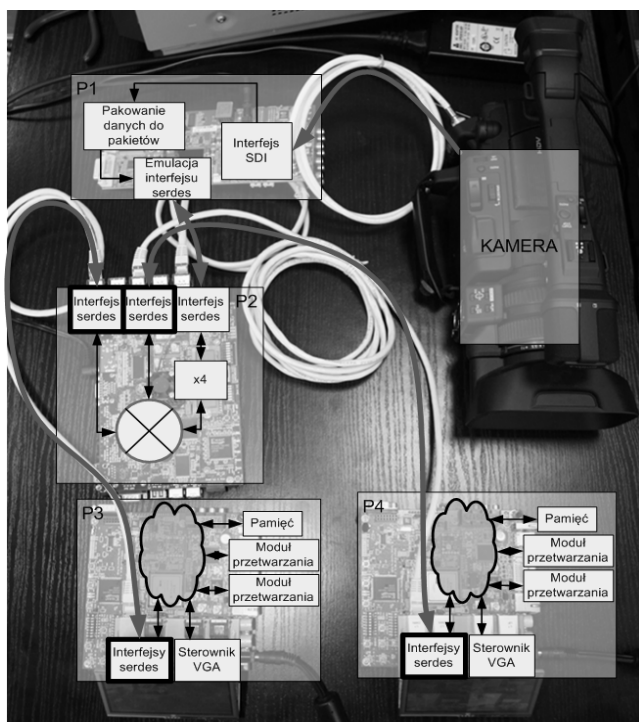
Układ FPGA	$N$	$f_{clk}$ [MHz] (maksymalna)	Liczba LUT	Liczba rejestrów
Virtex 4 XC4V45X35-12	1	237.2	210	116
	2	237.2	417	207
Virtex 5 XC5V55X35T-2	1	358.4	151	116
	2	358.4	301	207

Opracowane moduły transmisji zostały wykorzystane w połączeniu z siecią NoC [6]. Układ z wykorzystaniem 4 płyt z układami Virtex i z naniesionym logicznym schematem połączeń przedstawiony jest na rysunku 4.

Źródłem danych dla całego systemu jest kamera. Płyta oznaczona jako P1 to Video Input/Output Daughter Card firmy Xilinx[7], która odpowiada za umieszczenie danych z kamery w pakietach wykorzystanej sieci. Ze względu na brak modułów SERDES, w układzie, w który jest wyposażona ta płyta konieczne było ich emulowanie i przesłanie danych z mniejszą od zakładanej prędkością bitową (strumień SD o prędkości 156 Mbps). Strumień o prędkości 0,6 Gbps został uzyskany przez czterokrotne zwielokrotnienie strumienia SD na płycie P2. Transmisja z opisaną prędkością odbywa się pomiędzy płytą oznaczoną jako P2 i wyposażoną w układ Virtex 4, a płytami P3 i P4 wyposażonymi w układy Virtex 5 (interfejsy oznaczone na rysunku 4 grubą obwódką).

Przesłany sygnał wizyjny jest przetwarzany i wyświetlany na ekranach znajdujących się na płytach P3 i P4 (rys. 5). Zaobserwowana poprawność wyświetlanych danych stanowi przesłankę do stwierdzenia o prawidłowym działaniu systemu.

Niezależnie przeprowadzona została analiza stopy błędów transmisji. W tym celu kolejne pakiety przed przesłaniem przez interfejs zostały ponumerowane i sprawdzona została ciągłość ich numeracji po dotarciu do odbiornika. Jednocześnie w odbiornikach sprawdzana była poprawność formatowania pakietu zgodnie z przyjętym protokołem. Jeżeli pakiet na skutek błędów transmisji nie był zgodny z protokołem to następowało jego usunięcie, co skutkowało brakiem ciągłości w strumieniu pakietów i odbijało się na stopie błędów. Prawdopodobieństwo wystąpienia błędu w pakiecie zależy od jego długości, od częstotliwości transmisji oraz medium transmisyjnego wpływającego na zaszumienie sygnału. Sygnał był przesyłany przez półtoręj metrową skrętkę kategorii 6.



Rys. 4. Zdjęcie oraz schemat systemu demonstracyjnego.  
Fig. 4. Photo and schematic of demonstration system.



Rys. 5. Wyświetlacze z przetworzonym strumieniem wizyjnym.  
Fig. 5. Displays with processed video.

W tabeli 3 zgromadzone są stopy błędów transmisji mierzonych, jako ilość usuniętych pakietów w stosunku całkowitej liczby pakietów w strumieniu, oraz uzyskane prędkości w pojedynczym kanale z uwzględnieniem częstotliwości taktowania modułu nadawczego i długości pakietu.

Tab. 3. Parametry opracowanego interfejsu  
Tab. 3. Developed interface parameters

$f_{clk}$ [MHz]	Długość pakietu [słowa 32-bitowe]	B/kanal [bit/s]	Stopa błędów
66	16	0,66 Gbps	$1,2 \cdot 10^{-5}$
66	32	0,66 Gbps	$2,3 \cdot 10^{-5}$
50	16	0,5 Gbps	$8,7 \cdot 10^{-6}$
50	32	0,5 Gbps	$1,6 \cdot 10^{-5}$
25	16	0,25 Gbps	$1,1 \cdot 10^{-6}$
25	32	0,25 Gbps	$1,3 \cdot 10^{-6}$

#### 4. Podsumowanie

W artykule została przedstawiona idea skalowalnego interfejsu o potencjalnie dużej prędkości transmisji, przeznaczonego dla wszystkich układów Virtex z serii 4 i 5. Interfejs ten stanowi prostą i tanią alternatywę dla modułów RocketIO, które znacznie podnoszą koszt wyposażonych w nie układów. Poprawność działania opracowanego rozwiązania została potwierdzona poprzez jego aplikację do systemu przetwarzania strumieni wizyjnych.

#### 5. Literatura

- [1] "Virtex-4 FPGA User Guide", Xilinx UG070 (v2.6), 12 styczeń 2008
- [2] "Virtex-5 Family Overview", Xilinx DS100 (v5.0), 6 Luty 2009
- [3] <http://em.avnet.com/>, 27 luty 2010
- [4] "Virtex-5 FPGA User Guide", Xilinx UG190 (v4.7), 1 Maj 2009
- [5] "Virtex-5 FPGA Packaging and Pinout Specification", Xilinx UG195 (v4.6) 5 Maj 2009
- [6] A. Łuczak, M.Kurc, M.Stępniewska, K. Wegner, "Platforma przetwarzania rozproszonego bazująca na sieci NoC" PAK (vol. 54), nr 06/2008, Warszawa
- [7] "Video Input/Output Daughter Card User Guide", Xilinx UG235 (v1.2.1), 31 październik 2007