

Ernest JAMRO¹, Maciej WIELGOSZ¹, Witold CIOCH², Sławomir BIENIASZ³

¹AGH, KATEDRA ELEKTRONIKI, Al. Mickiewicza 30, 30-059 Kraków

²AGH, KATEDRA MACHANIKI I WIBROAKUSTYKI, Al. Mickiewicza 30, 30-059 Kraków

³AGH, KATEDRA INFORMATYKI, Al. Mickiewicza 30, 30-059 Kraków

Moduł wydajnego przetwarzania sygnałów dedykowany dla systemu wbudowanego opartego na układzie FPGA

Dr inż. Ernest JAMRO

Ukończył studia na AGH na kierunku Elektronika oraz na University of Huddersfield (UK) na kierunku Elektronika i Telekomunikacja. Obronił pracę doktorską w 2001 roku na AGH na wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki. Aktualnie jest adiunktem w Katedrze Elektroniki na AGH. Jego zainteresowania naukowe to sprzętowa akceleracja obliczeń, niskopoziomowe przetwarzanie obrazów, sieci neuronowe.



e-mail: jamro@agh.edu.pl

Dr inż. Maciej WIELGOSZ

Ukończył studia na AGH (2005), wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki na kierunku Elektronika i Telekomunikacja. Obronił pracę doktorską w 2010 roku. Obecnie jest pracownikiem Katedry Elektroniki AGH i bierze czynny udział w pracach badawczych realizowanych w zespole rekonfigurowalnych systemów obliczeniowych. Jego zainteresowania naukowe dotyczą sprzętowej akceleracji obliczeń, kompresji obrazu i sieci neuronowych.



e-mail: wielgosz@agh.edu.pl

Dr inż. Witold CIOCH

Ukończył studia na Wydziale Budowy Maszyn i Lotnictwa PRz, specjalność napędy lotnicze oraz na Wydziale Inżynierii Mechanicznej i Robotyki AGH, specjalność wibroakustyka. Pracę doktorską obronił w 2004. Obecnie pracuje jako adiunkt w Katedrze Mechaniki i Wibroakustyki AGH. Jego zainteresowania naukowe dotyczą diagnostyki technicznej i wibroakustyki. Zajmuje się zagadnieniami cyfrowego przetwarzania sygnałów, sieciami neuronowymi i analizą ryzyka eksploatacyjnego.



e-mail: cioch@agh.edu.pl

Dr inż. Sławomir BIENIASZ

Ukończył studia na kierunku Informatyka na AGH. Pracę doktorską obronił w 2006 roku na AGH na wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki. Aktualnie jest adiunktem w Katedrze Informatyki na AGH. Jego zainteresowania naukowe to agentowe modele symulacji zjawisk fizycznych, programowanie aspektowe, programowanie w środowisku systemu Unix (Linux).



e-mail: bieniasz@agh.edu.pl

Streszczenie

W niniejszym artykule opisano dedykowany moduł akceleracji obliczeń filtracji FIR (filtrów o skończonej odpowiedzi impulsowej) o nazwie *xsp_calc*. Moduł ten jest kompatybilny ze środowiskiem EDK (Embedded Development Kit) firmy Xilinx oraz magistralą PLB (Processor Local Bus). Na magistrali PLB niniejszy moduł jest urządzeniem typu master, oraz może wykonywać 8 operacji MACs (dodaj i akumuluj) na takt zegara. Dodatkowo moduł ten może obliczać wartość maksymalną, minimalną, średnią oraz skuteczną sygnału.

Słowa kluczowe: FPGA, systemy wbudowane, Procedura Liniowej Decymacji (PLD).

Dedicated module for digital signal processing and FPGA-based embedded system

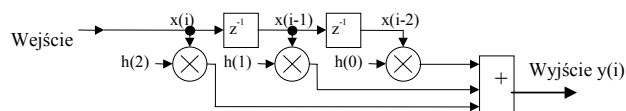
Abstract

In this paper a dedicated module compatible with PLB (Processor Local Bus) and EDK (Embedded Development Kit) provided by Xilinx is described. This module accelerates FIR (Finite Impulse Response) operations as well as average value and RMS (Root Mean Square) calculations. This module was employed in Programmable Unit for Diagnostics (PUD) [4, 5] and for Procedure of Linear Decimation (PLD) [6, 7]. For PLD the decimation ratio depends on the rotary machinery angular speed, and thus number of FIR filter nodes changes from 20 to 2000. Consequently, no standard FIR filter architecture for FPGA can be efficiently employed. Furthermore, the dedicated module presented in Fig. 2 was designed. This module is a master on PLB bus therefore it can perform input/output data transfer independently of the processor MicroBlaze. The processor just initialize calculation process by writing proper data to the selected control registers. This module can perform up to 8 MACs (Multiply and Accumulate) operations per clock cycle, sufficiently for the presented system and comparable with the computation power of a DSP (Digital Signal Processor). The implementation results presented in Tab. 1 illustrate that the presented module requires roughly twice the resources of the MicroBlaze and can speed up FIR calculation process roughly 20 times in comparison to the MicroBlaze.

Keywords: FPGA, embedded systems, Procedure of Linear Decimation (PLD).

1. Wstęp

Układy FPGA w porównaniu z procesorami ogólnego przeznaczenia CPU czy też procesorami sygnałowymi DSP charakteryzują się znaczącą mocą obliczeniową w wybranych zastosowaniach. Dotyczy to szczególnie przetwarzania sygnałów w postaci np. filtrowania danych pochodzących z przetworników analogowo-cyfrowych. Na przykład układ Virtex-7 posiada 3600 modułów DSP (ang. DSP Slices) i potrafi wykonać nawet 5 TMACS (5×10^{12} operacji mnoż i dodaj na sekundę) [1]. Architektura typowego układu filtrującego została przedstawiona na rys. 1. Firma Xilinx umożliwia automatyczną generację modułów najczęściej stosowanych filtrów [2]. Dokładne studium literaturowe dotyczące implementacji filtrów FIR w układach FPGA zostało przedstawione w [3].



Rys. 1. Architektura typowego filtra FIR
Fig. 1. Architecture of a typical FIR filter

Wspomniana moc obliczeniowa układów FPGA jest zdecydowanie zbyt wysoka dla typowego systemu wbudowanego. W referencyjnym systemie wbudowanym o nazwie Programowalne Urządzenie Diagnostyczne PUD [4, 5] wspomniana moc obliczeniowa zdecydowanie przewyższa wymagania. Liczba rejestrowanych kanałów wynosi 16, natomiast częstotliwość próbkowania jest rzędu 500kS/s na kanał. Urządzenie PUD jest w pierwszym rzędzie wykorzystywane do diagnozowania maszyn rotacyjnych w stanach niestacjonarnych. Podstawową operacją wykonywaną z wykorzystaniem urządzenia PUD jest autorska Procedura Liniowej Decymacji (PLD) [6, 7], która w pierwszym przybliżeniu polega na takim próbkowaniu sygnałów wejściowych, aby liczba próbek na jeden obrót maszyny rotacyjnej była stała. W konsekwencji wymagany jest zmienne współczynnik decymacji dla zmiennych obrotów maszyny. Dlatego rząd i parametry filtru antyaliasingowego również muszą

ulegać zmianie. Zmienność parametrów powoduje że rząd filtru FIR może być od 20 do nawet 2000. Co więcej w urządzeniu PUD może być stosowana np. stała decymacja sygnału wejściowego, tak aby ograniczyć liczbę koniecznych do zapisania próbek (w przypadku sygnałów wolnozmiennych), czy też mogą być stosowane filtry kaskadowe, które wydają się być najlepszym rozwiązaniem w przypadku dużego współczynnika decymacji nie będącego liczbą pierwszą. Wspomniane wymagania oraz skomplikowana ścieżka danych (wstępne filtrowanie, filtry kaskadowe, PLD) powodują, że użycie standardowego układu filtrującego zostało wykluczone. Dodatkowo należy zauważyć, że w standardowym układzie filtrującym typu FIR przy dużym współczynniku decymacji większość próbek wyjściowych jest odrzucana, czyli większość wykonywanych obliczeń jest bezużyteczna. W zaproponowanym układzie obliczenia prowadzone są tylko dla próbek wyjściowych, które są dalej wykorzystywane. Warto zauważyć, że dla filtrów IIR (o nieskończonej odpowiedzi impulsowej) obliczenia muszą być prowadzone dla wszystkich próbek (również tych odrzucanych). Ponadto filtry IIR są mało stabilne numerycznie dla dużych współczynników decymacji – wymagają obliczeń prowadzonych z dużą dokładnością bitową [8] – dużo większą niż dla filtrów FIR. Dlatego, w niniejszym projekcie korzystniejsze jest wykorzystanie filtrów FIR zamiast IIR.

2. Zaproponowane rozwiązanie

2.1. Założenia wstępne

W konsekwencji powyższych rozważań zaproponowano rozwiązanie, które jest stosowane zarówno w układach FPGA jak i w procesorach DSP. Pierwszym założeniem jest to, że do projektowania układu FPGA użyto pakietu Embedded Development Kit (EDK) firmy Xilinx (aktualna wersja 13.4). Dzięki temu możliwe stało się łatwe łączenie niezależnych modułów poprzez standard magistrali PLB (Processor Local Bus), użycie soft-procesora MicroBlaze dostarczanego w ramach niniejszego pakietu, łatwego dostępu do pamięci zewnętrznej SDRAM i innych modułów. Niestety moc obliczeniowa samego procesora MicroBlaze jest niewystarczająca.

W konsekwencji zaprojektowano własny moduł obliczeniowy o nazwie *xsp_calc*, który został zaprojektowany przy użyciu języka opisu sprzętu VHDL i jest kompatybilny z magistralą PLB oraz środowiskiem EDK. Moduł ten jest urządzeniem typu master na magistrali PLB, czyli sam inicjalizuje transfer danych podobnie do modułu DMA (Direct Memory Access). Moduł DMA, służy do transmisji blokowej danych pomiędzy różnymi lokalizacjami pamięci niezależnie od działań procesora. Podobnie jak moduł DMA, moduł *xsp_calc* jest inicjalizowany (konfigurowany) przez procesor MicroBlaze, natomiast samo przetwarzanie danych odbywa się niezależnie od procesora. W przeciwieństwie do modułu DMA, moduł *xsp_calc* nie tyle kopiuje dane, co wykonuje dodatkowe operacje, głównie filtrowania FIR.

Architektura zaprojektowanego filtru FIR, przypomina architekturę procesora DSP, czyli dane wejściowe i współczynniki filtru FIR są pobierane z pamięci zewnętrznej. Aby przyspieszyć proces obliczeń raz pobrane współczynniki są zapisywane w pamięci wewnętrznej omawianego modułu. Aby ścieżka danych wejściowych nie stanowiła wąskiego gardła, dane te również muszą być wewnętrznie pamiętane (jakby cache'owane). Ze względu na uporządkowany dostęp do pamięci danych wejściowych, nie jest konieczne stosowanie zaawansowanej logiki pamięci cache, wystarczy zwykła pamięć lokalna BRAM (wbudowana pamięć blokowa) oraz odpowiedni automat stanu sterujący zapisem i odczytem tej pamięci.

2.2. Architektura modułu filtrującego

Moduł filtrujący FIR wykonuje następującą operację:

$$y(j) = SAT\left(\sum_{i=0}^{N-1} x(j \cdot D + i) \cdot w(i)\right) \quad (1)$$

gdzie: $y(j)$ – j -ty element danej wyjściowej, *SAT*- moduł dokonujący operacji nasycenia w kodzie uzupełnień do dwóch do szerokości 16-bitów,

$$SAT(x) = \begin{cases} 2^{15} - 1 & \text{dla } x \geq 2^{15} \\ x & \text{dla } x < 2^{15} \text{ i } x \geq -2^{15} \\ -2^{15} & \text{dla } x < -2^{15} \end{cases}$$

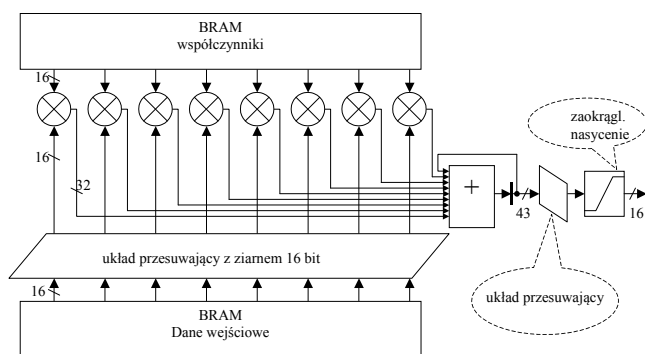
$x(i)$ – i -ta dana wejściowa, N - rząd filtru, D - współczynnik decymacji, dla $D=1$ uzyskuje się zwykłą filtrację, $w(i)$ – i -ty współczynnik filtru FIR.

Moduł filtrujący FIR składa się z:

- modułu arytmetycznego zdolnego wykonać 8 operacji MAC (dodaj i akumuluj) na takt zegara (dane wejściowe i współczynniki filtru FIR są 16-bitowe).
- lokalnej pamięci współczynników o szerokości magistrali danych $8 \times 16 \text{ bitów} = 128 \text{ bitów}$. Wielkość tej pamięci jest konfigurowalna, wartość domyślna 2048 współczynników.
- lokalnej pamięci danych wejściowych o szerokości magistrali danych 128-bitów i wielkości $2048 \times 16 \text{ bitów}$.
- układu przesuwanego dane wejściowe pochodzące z pamięci lokalnej BRAM o wielokrotność 16-bit, czyli rozdzielczość bitową danej wejściowej. Magistrala danych pamięci BRAM ma szerokość $8 \times 16 \text{ bitów} = 128 \text{ bity}$, a odczyt z tej pamięci musi być dokonywany z dokładnością 16-bitów. Warto zwrócić uwagę, że niniejszy układ wymaga dodatkowych rejestrów a pierwszy odczyt z tej pamięci może wymagać dwóch taktów zegara kiedy odczytywane 128-bitowe słowo jest współdzielone w pamięci przez dwa sąsiednie słowa 128-bitowe,
- układu przesuwanego wynik akumulacji – układ ten jest stosowany w celu dodatkowego mnożenia wyniku akumulacji przez: 0,5, 1, 2, 4. Umożliwia to dodatkowe mnożenie danej wejściowej w przypadku kiedy dana wejściowa ma niewielką amplitudę. Warto podkreślić że mnożenie to odbywa się przed procesem zaokrąglania i redukcji szerokości bitowej wyniku, dlatego w przypadku decymacji możliwe jest uzyskanie dodatkowej rozdzielczości bitowej danej wejściowej w porównaniu z rzeczywistą rozdzielczością bitową danej wejściowej. Na przykład jeśli dana wejściowa rejestrowana przez 16-bitowy przetwornik analogowo-cyfrowy zawiera się w połowie zakresu pomiarowego to rzeczywista rozdzielczość bitowa danej wejściowej wynosi tylko 15-bitów. Można jednak zwiększyć tę rozdzielczość dla odpowiednio dużego współczynnika decymacji do 16-bitów. Podobna metoda nadpróbkowania jest stosowana np. w przetwornikach analogowo-cyfrowych w celu zwiększenia rozdzielczości bitowej.
- moduł zaokrąglania i nasycenia – moduł ten stosowany jest w celu zaokrąglania i nasycenia danej wyjściowej. Nasycenie zabezpiecza przed otrzymaniem całkowicie błędnego wyniku filtracji w przypadku kiedy wynik akumulacji i przesunięcia zapisany w kodzie uzupełnień do dwóch przekracza 16-bitową szerokość.

W konsekwencji moc obliczeniowa wspomnianego modułu przy częstotliwości 50MHz (wyższa częstotliwość pracy systemu nie była testowana) wynosi 400MMACS. Jest to wystarczająca moc obliczeniowa wymagana w opisywanym systemie. Warto zwrócić uwagę, że wspomnianą moc obliczeniową można otrzymać używając procesora DSP. Jednakże użyto tutaj układ FPGA ze względu na konieczność rejestracji wielu kanałów, pojedynczy procesor DSP raczej nie byłby w stanie poradzić sobie z obsługą tak wielu przetworników analogowo-cyfrowych przy tej częstotliwości próbkowania.

Warto podkreślić, że w urządzeniu PUD, układ FPGA komunikuje się z procesorem ARM (OMAP3530) poprzez łącze SPI o przepustowości 2MB/s [9]. Pomimo zaprojektowania własnego modułu kompatybilnego ze środowiskiem EDK [9] standard SPI stanowi wąskie gardło opisywanego systemu. Dlatego wszelkie podstawowe przetwarzanie danych powinno odbywać się w układzie FPGA.



Rys. 2. Schemat blokowy części obliczeniowej
Fig. 2. Block diagram of the computation unit

2.3. Obliczanie podstawowych parametrów sygnału

Równoległe z filtrowaniem, moduł *xsp_calc* umożliwia obliczanie podstawowych parametrów statystycznych sygnału wejściowego (przed filtrowaniem) lub wyjściowego (po filtrowaniu). Do podstawowych obliczanych parametrów należą:

- 1) Wartość minimalna.
- 2) Wartość maksymalna.
- 3) Wartość średnia – niniejszy moduł oblicza sumę wszystkich danych wejściowych, obliczenie wartości średniej wymaga podzielenia sumy przez liczbę sumowanych elementów. Operacja dzielenia jest wykonywana tylko raz dla wszystkich zsumowanych danych wejściowych. W konsekwencji szybkość wykonywania operacji dzielenia nie jest kluczowa i jest wykonywana na soft-procesorze MicroBlaze.
- 4) Wartość skuteczna – w rzeczywistości dokonywane jest sumowanie danych wejściowych podniesionych do kwadratu $\sum x_i^2$.

Podobnie jak dla obliczenia wartości średniej operacje dzielenia i pierwiastkowania wykonuje soft-procesor MicroBlaze. Obliczenie poniższych parametrów sygnału nie zajmuje znaczących zasobów układu FPGA, a w wielu przypadkach zdecydowania pomaga analizować badane sygnały. Na przykład obliczenie wartości maksymalnej i minimalnej może być stosowane w celu odpowiedniego ustawienia zakresu pomiarowego (ustawieniu wartości wzmocnienia przedwzmacniacza, czy też przesunięcia bitowego wyniku filtracji FIR). Wartość średnia może być stosowana w celu obliczenia (odcinka) składowej stałej sygnału. Warto zauważyć że wzmacniacz wejściowy w układzie analogowym przed przetwornikiem analogowo-cyfrowym odcina składową stałą. Niemniej napięcie niezrównoważenia wzmacniacza może być w prosty sposób skorygowane w procesie kalibracji. Wartość skuteczna jest jednym z podstawowych parametrów sygnału wejściowego i sama w sobie może już świadczyć o uszkodzeniu maszyny rotującej.

2.4. Wyniki Implementacji

Wyniki implementacji w układzie FPGA Spartan3 XC3S1500 FG456 zostały przedstawione w tab. 1. Moduł obliczeniowy *xsp_calc* zajmuje około 2 razy więcej zasobów od procesora MicroBlaze (procesora bez pamięci cache). Natomiast jego szczytowa moc obliczeniowa jest co najmniej 20 razy większa od mocy obliczeniowej procesora MicroBlaze. Zakłada się, że procesor MicroBlaze potrzebuje co najmniej 1 takt zegara na wykonanie pojedynczej operacji mnożenia, dodawania i porównania (wynika to z analizy kodu assemblera tego procesora). Zakłada się, że moduł *xsp_calc* wykonuje 8 operacji mnożenia i dodawania (filtr FIR), 2 operacje porównania (wartość minimalna / maksymalna), 2 operacje dodawania (wartość średnia i skuteczna) oraz jedną operację mnożenia (obliczanie kwadratu dla wartości skutecznej) na takt zegara.

Aby porównać zasoby zajmowane przez inne moduły podano również zasoby zajmowane przez moduł *mpmc* – Multi-Port Memory Controller [10], czyli kontroler pamięci zewnętrznej SDRAM. Jak widać moduł obliczeniowy *xsp_calc* zajmuje porównywalne zasoby układu FPGA. W ostatnich dwóch wierszach tab. 1 umieszczono zasoby zajmowane przez cały system (system) oraz dostępne zasoby użytego układu FPGA XC3S1500.

Tab. 1. Wyniki Implementacji
Tab. 1. Implementation results

Moduł	FF	LUT	BRAM	MUL18x18
<i>xsp_calc</i>	1589	2235	4	8
MicroBlaze	891	1362	0	3
MPMC	1555	1501	5	0
System	7972	10410	30	11
XC3S1500	29952	29952	32	32

3. Wnioski końcowe

W niniejszej pracy przedstawiony własny moduł obliczeniowy o nazwie *xsp_calc*, który został zaprojektowany przy użyciu języka opisu sprzętu VHDL i jest kompatybilny z magistralą PLB oraz środowiskiem EDK. Rząd filtra jest proporcjonalny do współczynnika decymacji. W konsekwencji jeżeli obliczenia będą wykonywane tylko dla użytecznych próbek wyjściowych, całkowita moc obliczeniowa pozostaje mniej więcej na stałym poziomie. Dla filtrów FIR rzędu 2000 i liczbie kanałów 16, częstotliwości próbkowania 250kS/s, kluczowa nie jest szybkość przetwarzania ale zasoby pamięci zajmowane przez taki filtr: $2000 \times 16 \text{ bitów} = 4 \text{ kB}$, dla szesnastu kanałów całkowita pamięć danych wynosi 64kB. Aby dostęp do pamięci zewnętrznej nie stanowił wąskiego gardła omawianego systemu, zdecydowano się na zastosowanie dodatkowej buforowej pamięci danych (pełniącej funkcję podobną do pamięci cache w CPU). Zdecydowano się również na przetwarzanie tylko pojedynczego kanału w jednym czasie aby zminimalizować potrzebne zasoby pamięciowe – liczbę przetwarzanych danych w jednym czasie.

Podsumowując opisywany moduł działa podobnie jak procesor sygnałowy i stanowi wygodną alternatywę dla użycie zewnętrznego procesora DSP.

Praca finansowana ze środków Ministerstwa Nauki i Szkolnictwa Wyższego.

4. Literatura

- [1] Xilinx Inc.: 7 Series FPGA overview, DS180 (v1.8) Sep. 13, 2011.
- [2] Xilinx Inc.: IP LogiCORE FIR Compiler v5.0, DS534 March 1, 2011.
- [3] Jamro E: Parameterised automated generation of convolvers implemented in FPGAs, Ph.D. Thesis, AGH University of Science and Technology, Kraków, Poland, June 2001.
- [4] Wielgosz M., Jamro E., Cioch W., Bieniasz S.: System wbudowany oparty na procesorze ARM oraz układzie FPGA, PAK 2011 nr 08, s. 877-879.
- [5] Jamro E., Wielgosz M., Bieniasz S., Cioch W.: FPGA – ARM Heterogeneous System for High Speed Signal Analysis, Solid State Phenomena Vol. 180 (2012) pp 207-213.
- [6] Adamczyk J., Krzyworzeka P., Cioch W., Jamro E.: Monitoring of Nonstationary States in Rotating Machinery, WITE Państwowy Instytut Badawczy - Radom, Kraków 2006.
- [7] Gardulski J.: The application of a linear decimation procedure In the diagnostics of shock absorbers in passenger vehicles, Zagadnienia Eksploatacji Maszyn, Zeszyt 4 (152), 2007.
- [8] Dąbrowski D., Cioch W., Jamro E.: Digital filters parameters selection for hardware implementation in programmable devices. Diagnostyka 2008 nr 4 s. 129-132.
- [9] Jamro E., Wielgosz M., Cioch W., Bieniasz S.: Efektywna komunikacja ARM-FPGA z użyciem interfejsu SPI, PAK vol. 57, nr 8/2011, pp. 874-876.
- [10] Xilinx Inc, LogiCORE IP Multi-Port Memory Controller (MPMC) (v6.03.a), DS643 March 1, 2011.