

**Michał SKIWSKI**

INSTYTUT FIZYKI, WYDZIAŁ FIZYKI ASTRONOMII I INFORMATYKI STOSOWANEJ, UNIWERSYTET MIKOŁAJA KOPERNIKA,  
ul. Grudziądzka 5, 87-100 Toruń

**Cyfrowa filtracja sygnałów z wykorzystaniem układów FPGA**

Mgr inż. Michał SKIWSKI

Absolwent Wydziału Fizyki Astronomii i Informatyki Stosowanej Uniwersytetu Mikołaja Kopernika w Toruniu. W 2009 r ukończył Kierunek Fizyka Techniczna w stopniu magister inżynier. W 2012 roku rozpoczął studia doktoranckie na Wydziale Fizyki Astronomii i Informatyki Stosowanej Uniwersytetu Mikołaja Kopernika w Toruniu.



e-mail: msk@doktorant.umk.pl

**Streszczenie**

W artykule przedstawiono sposoby realizacji filtrów cyfrowych o skończonej odpowiedzi impulsowej z wykorzystaniem dedykowanych bloków w układzie FPGA. Proponowane rozwiązania dają nowe spojrzenie na tą gałąź systemów przetwarzania sygnałów ze względu na możliwość dokonywania filtracji sygnałów nawet w jednym cyklu zegara, co jest nieosiągalne dla systemów opartych na klasycznych procesorach DSP. Zaprezentowane zostało również kilka przykładów konstrukcji filtrów.

**Słowa kluczowe:** układy programowalne, FPGA, DSP48, filtry cyfrowe.

**Digital signal filtration using FPGA****Abstract**

This article shows how to implement the finite impulse response digital filters using dedicated FPGA blocks. This is a new approach to the digital signal processing because single FPGA chips can perform the requested operations much faster than traditional single processor, making them in parallel. The consequence of that fact is the ability to implement the algorithm nearly direct way to the programmable structure, as shown in the Fig. 2 (structure) and Fig. 5 (hardware counterpart). Several filter designs are presented both with full parallel processing and semi-parallel processing. Despite of many advantages in the proposed solution there are also some disadvantages, for example lack of possibility to operate floating-point arithmetic and the difficulty to create high order filters.

**Keywords:** programmable circuits, FPGA, DSP48, digital filters.

**1. Wprowadzenie**

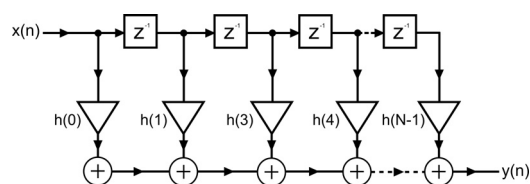
Filtry cyfrowe posiadają wiele zalet w stosunku do filtrów analogowych. Główną zaletą filtrów opartych na technikach cyfrowych jest możliwość uzyskania bardzo stromej charakterystyki amplitudowo-częstotliwościowej, o szerokości obszaru przejściowego nie do osiągnięcia w przypadku filtrów analogowych. Przy wykorzystaniu odpowiednich metod projektowania istnieje możliwość tworzenia filtrów cyfrowych o dokładnie liniowej fazie, nie do uzyskania w ich analogowych odpowiednikach. Szybkość działania filtrów cyfrowych silnie zależy od liczby współczynników, im ich więcej tym wymagany jest dłuższy czas na dokończenie pojedynczej operacji filtracji.

**1.1. Struktury filtrów o skończonej odpowiedzi impulsowej**

Podstawową strukturą filtrów o skończonej odpowiedzi impulsowej (FIR) jest tzw. postać bezpośrednia, która wynika z równania splotu (1).

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (1)$$

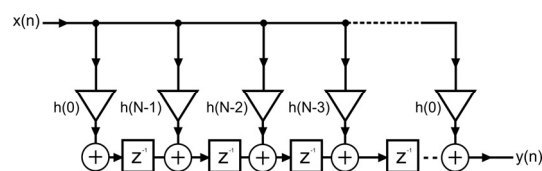
Struktura przedstawiona na rys. 1 odpowiada bezpośrednio uporządkowaniu dodawań i mnożeń związanych zależnością (1).



Rys. 1. Postać bezpośrednia filtra FIR

Fig. 1. FIR filter direct form

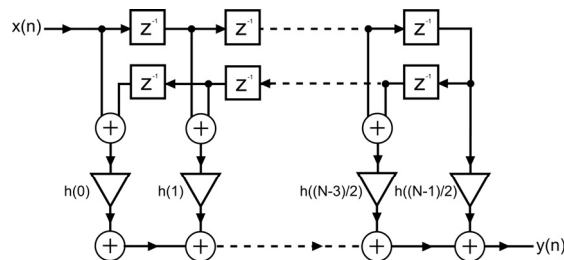
Istnieje jeszcze wiele innych organizacji obliczeń, które teoretycznie są równoważne strukturze bezpośredniej. Do tego typu sieci należy postać odwrócona (rys. 2). Modyfikacją jest tu przeniesienie bloków opóźniających pomiędzy bloki sumujące. Inne jest również rozdysponowanie wyrazów odpowiedzi impulsowej [1]. Ta forma dzięki wyeliminowaniu konieczności jednoczesnego zsumowania wyników operacji mnożenia jest idealnym rozwiązaniem dla sprzętowej implementacji w układzie FPGA.



Rys. 2. Odwrócona postać bezpośrednia filtra FIR

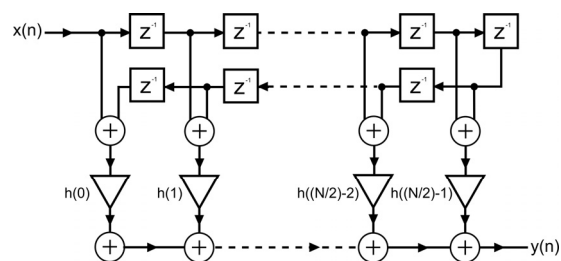
Fig. 2. Reverse FIR filter direct form

Pewnym ułatwieniem jest istnienie sieci symetrycznej wymagającej tylko połowy (dla  $N$  parzystego) bądź  $(N-1)/2$  (dla  $N$  nieparzystego) mnożeń. Tego typu strukturę dla  $N$  parzystego zilustrowano na rys. 3, a dla  $N$  nieparzystego na rys. 4. Podobnie jak w strukturze bezpośredniej możliwe są do otrzymania ich postacie odwrócone [1].



Rys. 3. Postać bezpośrednia symetrycznego filtra FIR parzystego rzędu

Fig. 3. Even order symmetric FIR filter direct form



Rys. 4. Postać bezpośrednia symetrycznego filtra FIR nieparzystego rzędu

Fig. 4. Odd order symmetric FIR filter direct form

## 2. Rozwiązania konstrukcyjne

Szeroko stosowanymi układami do obróbki sygnałów są procesory DSP. Cechą charakterystyczną procesora sygnałowego jest obecność między innymi jednostki mnożąco-akumulującej, która w przypadku klasycznego procesora sygnałowego jest w stanie przeprowadzić operację mnożenia z akumulacją w jednym taktie zegara. Jak można zauważyć dla przykładowego filtra o skończonej odpowiedzi impulsowej, przypuśćmy, o 256 współczynnikach, konieczne jest przeprowadzenie 256 operacji mnożenia z akumulacją, aby uzyskać wymagany wynik. Przekłada się to na 256 cykli zegara dla wyprodukowania jednej próbki wyjściowej.

Zwiększenie długości filtra prowadzi do coraz większej liczby cykli zegara niezbędnych do dokończenia zadanej operacji. Stanowi to poważne ograniczenie dla częstotliwości nadsyłania kolejnych próbek sygnału. Aby zapewnić optymalną szybkość przetwarzania konieczne jest zwiększanie częstotliwości zegara taktującego procesor DSP. Tego typu rozwiązanie z wielu względów jest dość kłopotliwe i stopniowo coraz trudniejsze do zrealizowania.

Układy FPGA dzięki swej architekturze są w stanie doprowadzić do zrównoleżenia wielu procesów. W efekcie możliwe jest do zrealizowania współbieżnie wielu operacji mnożenia bądź sumowania.

Kolejnym etapem rozwoju układów FPGA w dziedzinie aplikacji DSP jest zastosowanie w strukturze programowalnej wyspecjalizowanych bloków przeznaczonych do szybkiego wykonywania operacji arytmetycznych. Dla układów firmy Xilinx bloki te noszą nazwę DSP48 i to na nich została oparta architektura prezentowanych filtrów.

### 2.1. Podstawowe informacje o układach FPGA

Układy FPGA stanowią zbiór identycznych bloków o określonej budowie, równomiernie rozmieszczonych w strukturze układu, które można łączyć między sobą za pośrednictwem specjalnej matrycy połączeń. Układy FPGA mogą liczyć od kilkudziesięciu do kilkuset tysięcy i więcej takich bloków.

Producenci układów FPGA stosują różne rozwiązania dla danego układu. Cechą wspólną jest ogólny zarys komórki logicznej (każdy producent stosuje swoje nazwy) w skład, której wchodzi przerzutnik oraz tablica wartości funkcji (tzw. Look-up table) za pomocą, której można zrealizować dowolną funkcję logiczną. Tablica wartości funkcji w istocie rzeczy jest czterowejściową pamięcią RAM [2, 3].

Matryca połączeń stanowi rodzaj pamięci. Istnieje kilka technologii wytwarzania matrycy połączeń. Najbardziej popularne technologie to: SRAM oraz Flash. Pamięci tego typu pozwalają na wielokrotne przeprogramowanie danej struktury. Największą wadą matryc połączeń na bazie pamięci statycznej jest utrata informacji o kodzie konfiguracyjnym przy braku napięcia zasilania. Wolne od tego problemu są matryce na bazie pamięci Flash. Oprócz wymienionych powyżej technologii istnieją też inne, z czego najbardziej specyficzna jest technologia antifuse. W przypadku tego typu matryc połączeń, informacja o konfiguracji jest zapisywana na stałe bez możliwości przeprogramowania. Trakty połączeniowe w tym przypadku stanowią sieć bezpieczników, które uzyskują możliwość przewodzenia sygnałów po ich wcześniejszym przepaleniu [4].

### 2.2. Bloki DSP48

Bloki DSP48 stanowią wyspecjalizowane jednostki przeznaczone do wykonywania operacji arytmetycznych takich jak: mnożenie, dodawanie, odejmowanie, akumulacja, mnożenia z akumulacją, mnożenia z dodawaniem bądź odejmowaniem. Oprócz operacji arytmetycznych elementy te mogą pełnić funkcje: multipleksera, licznika oraz komparatora. Dzięki swej architekturze bloki DSP48 dają możliwość realizowania złożonych operacji matematycznych typu arytmetyka zespolona czy też (co jest ich

głównym celem) tworzenie aplikacji DSP w tym, przede wszystkim, filtrów cyfrowych.

Charakterystyczną cechą bloków DSP48 jest możliwość łączenia ich w kaskadę. Rozwiązanie to jest bardzo wygodne między innymi przy tworzeniu filtrów o skończonej odpowiedzi impulsowej.

Komórki DSP48 we wnętrzu układu programowalnego zorganizowane są w kolumnach, liczba kolumn jak również liczba samych DSP48 w kolumnie zależna jest od typu układu. Tego typu organizacja zapewnia, iż połączenia pomiędzy blokami w danej kolumnie są jak najkrótsze [5].

### 2.3. Budowa filtrów

Tworzenie filtrów o skończonej odpowiedzi impulsowej za pomocą bloków DSP jest stosunkowo prostym procesem. Komórki DSP48 dzięki swej architekturze są w stanie tworzyć struktury dające możliwość filtrowania w jednym taktie zegara.

Filtry o skończonej odpowiedzi impulsowej zostały zrealizowane w dwóch konfiguracjach:

- równoległej:
  - transponowanej,
  - wykorzystującej symetrię współczynników,
- szeregowo-równoległej.

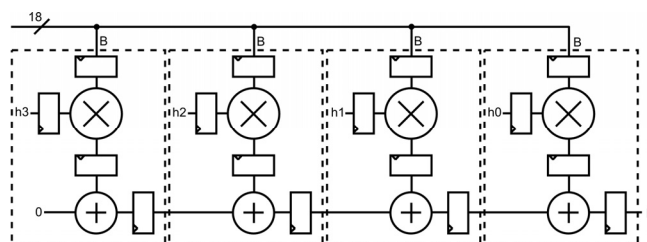
Konfiguracja równoległa stanowi przykład możliwości układów programowalnych, czyli współbieżne przeprowadzanie wielu procesów. Jednakże posiada ona istotne ograniczenie co do maksymalnej długości budowanego filtra. Dla równoległych filtrów FIR niesymetrycznych jeden blok DSP zarezerwowany jest dla jednego współczynnika. Oznacza to, że filtr o  $N$  współczynnikach zużywa  $N$  komórek DSP48. Wykorzystując symetrię współczynników konieczna jest tylko połowa tej liczby bloków DSP, aby zrealizować filtr o tej samej długości co transponowany.

Konfiguracja szeregowo-równoległa mimo tego, że w odróżnieniu od powyższych struktur nie jest w stanie wykonywać operacji filtracji w jednym taktie zegara ma sporo do zaoferowania. W przypadku tej konstrukcji długość filtra nie jest ograniczona liczbą komórek DSP48. Ma ona możliwość na ograniczonej ich liczbie tworzenia filtrów o skończonej odpowiedzi impulsowej, o dość znacznej długości.

### 2.4. Filtr o strukturze transponowanej

Filtr transponowany (rys. 5) stanowi najprostszą konstrukcję z tych poddanych realizacji. Filtr ten odpowiada odwróconej formie bezpośredniej typu I zaprezentowanej na rys. 2. Nie potrzebuje on żadnej dodatkowej logiki sterującej, a co za tym idzie nie ulegają zmianie główne zasoby układu programowalnego.

Na wejścia wszystkich bloków podawana jest ta sama próbka sygnału. Do każdego bloku przypisany jest odpowiedni współczynnik. Poprzez specjalne porty, wyjście jednego bloku podłączone jest do wejścia drugiego bloku itd.



Rys. 5. Schemat połączeń filtra FIR transponowanego  
Fig. 5. Transposed FIR filter connection diagram

Proces filtracji polega na tym, iż w pierwszym cyklu zegara próbka wejściowa zapisywana jest w rejestrze wejściowym każdego z bloków oraz równoległe przemnażana przez zadane współczynniki. Dalej, wynik mnożenia zapisywany jest w rejestrze pośrednim umieszczonym pomiędzy jednostką mnożącą, a sumu-

jąca i sumowany z zawartością rejestru wyjściowego z sąsiedniego bloku w łańcuchu. Jednocześnie kolejna próbka sygnału jest zapisywana w rejestrze wejściowym każdej z komórek DSP48. W trzecim cyklu wynik mnożenia z dodawaniem umieszczany jest w rejestrze wyjściowym każdego z bloków DSP48. Kolejna próbka sygnału umieszczana jest w rejestrach wejściowych, wynik mnożenia poprzedniej próbki zapisywany jest w rejestrze pośredniczącym i sumowany z zawartością rejestru wyjściowego poprzedzającego DSP48. W kolejnych cyklach następuje powtórzenie opisanej sekwencji [5].

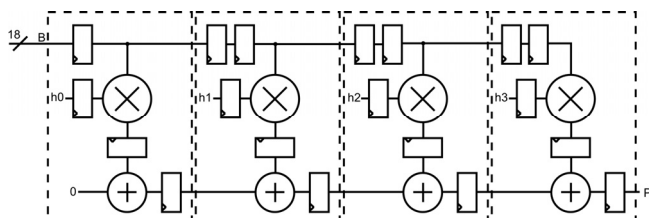
Z racji występowania w każdym bloku po trzy rejestry do przechowywania poszczególnych wartości z każdego etapu, sygnał wyjściowy, z każdego bloku, jest opóźniony o trzy cykle w stosunku do sygnału wejściowego. O ile rejestr za jednostką sumującą jest niezbędny do prawidłowego działania filtra to pozostałe już nie. Wynik końcowy dla pojedynczej próbki wejściowej jest opóźniony tylko o tyle cykli ile wynosi opóźnienie pojedynczej komórki DSP 48 (trzy cykle zegara).

## 2.5. Filtr o strukturze kaskadowej

Filtr, o skończonej odpowiedzi impulsowej, o strukturze kaskadowej nie został poddany praktycznej realizacji. Główną przyczyną tego stanu rzeczy był fakt, że dla niskich częstotliwości próbkowania sygnału samo działanie tego filtra poza większym opóźnieniem (między próbką wejściową a wyjściową) nie różniłoby się od struktury transponowanej.

Konstrukcja kaskadowa zaczyna mieć przewagę nad konstrukcją transponowaną, gdy mamy do czynienia z dość dużymi częstotliwościami nadsyłania kolejnych próbek sygnału. Przyczyną tej sytuacji jest sposób dostarczania próbek sygnału do poszczególnych bloków DSP. W filtrze transponowanym do wszystkich komórek DSP48 dostarczana jest ta sama próbka. O ile przy niższych częstotliwościach opóźnienia w ich dostarczaniu do poszczególnych wejść nie są aż tak istotne (nie istnieje ryzyko, że próbka sygnału pojawi się na wejściu najbardziej odległego bloku po odpowiadającym jej zboczu narastającym zegara – nie zostanie zapisana na czas w rejestrze wejściowym). Przy wysokich częstotliwościach pracy może dojść do sytuacji, że sygnał wejściowy może nie dotrzeć na czas do wszystkich elementów filtra.

Struktura kaskadowa wolna jest od tego typu problemów, o ile liczba współczynników nie przekracza liczby DSP48 w kolumnie. To ograniczenie wynika z tego, iż w układzie FPGA nie istnieje bezpośrednie połączenie pomiędzy kolumnami bloków DSP48, oczywiście samo połączenie jest możliwe jednakże byłoby ono bardzo długie i wprowadzałoby dość duże opóźnienia w przesyłanych sygnałach prowadząc do ograniczenia częstotliwości pracy samego filtra. W efekcie nie pozwala to na tworzenie filtrów o wysokim rzędzie. Schemat powyższej konstrukcji został zamieszczony na rysunku 6.



Rys. 6. Schemat połączeń filtra FIR kaskadowego  
Fig. 6. Systolic FIR filter connection diagram

Podstawową cechą odróżniającą strukturę kaskadową od transponowanej jest obecność dodatkowego rejestru przesuwającego. Oprócz rejestru w torze sumacyjnym (dolnego), występuje jeszcze rejestr w obwodzie wejściowym (górne połączenia między blokami). Jest on odpowiednikiem rejestru przesuwającego stosowanego w klasycznych realizacjach filtrów (szeregowych). Zwrócić należy uwagę na jego specyficzną budowę, o ile w standardowych roz-

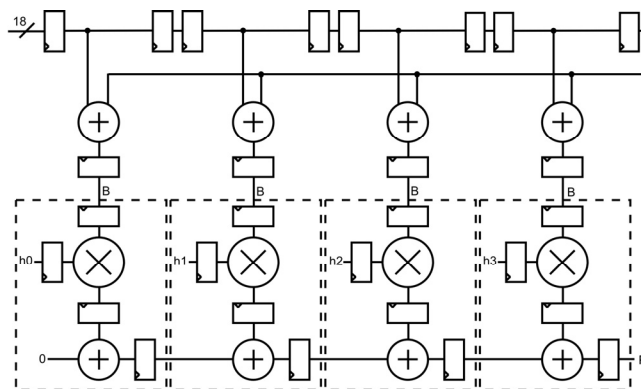
wiązaniach długość rejestru jest równa liczbie współczynników to w tym przypadku ta zależność nie jest spełniona. Inny jest również sposób rozdysponowania współczynników (współczynnik  $h_0$  znajduje się po lewej stronie, tak jak w strukturze bezpośredniej). Wszystkie bloki z wyłączeniem pierwszego wnoszą po dwa elementy opóźniające wchodzące w skład górnego rejestru przesuwającego [5].

## 2.6. Filtr o strukturze symetrycznej

Wzajemne połączenia bloków DSP48 dla filtra wykorzystującego symetrię współczynników są identyczne z połączeniem w przypadku filtra transponowanego. O ile w strukturze transponowanej do wszystkich wejść B podłączony był ten sam sygnał wejściowy to w strukturze symetrycznej konieczna jest dodatkowa logika zewnętrzna, która to stanowi układ pośredniczący pomiędzy próbkami sygnału a wejściami bloków DSP48.

Omawiany filtr (rys. 7) jest jedynym filtrem, w którym nie wszystkie operacje arytmetyczne są realizowane za pomocą komórek DSP48. Przyczyna tego jest prosta, co prawda ogranicza się o połowę liczbę mnożeń ale odbywa się to kosztem zwiększenia liczby bloków sumujących.

Realizacja tego typu filtra tylko i wyłącznie za pomocą bloków DSP48 miałyby się wówczas z celem. Sumaryczna liczba tych elementów użytych w projekcie wyniosła by tyle samo co dla konstrukcji równoległej - transponowanej. Zważywszy na to, że operacja dodawania jest mniej skomplikowana niż operacja mnożenia, korzystanie z innych modułów sumujących nie nastrocza zbyt wielkiego zużycia zasobów układu FPGA.



Rys. 7. Schemat połączeń filtra FIR symetrycznego  
Fig. 7. Symmetric FIR filter connection diagram

Dane dostarczane do zewnętrznych bloków sumujących są podawane w sposób charakterystyczny zarówno dla architektury transponowanej jak i kaskadowej. Pierwszym operandem jednostki sumującej jest dana z odpowiedniego miejsca rejestru przesuwającego. Drugim operandem jest wartość z ostatniej komórki rejestru przesuwającego, argument ten podawany jest na wejścia wszystkich wejściowych modułów sumujących.

Efekt końcowy jest wynikiem połączenia zasad działania architektury kaskadowej i transponowanej.

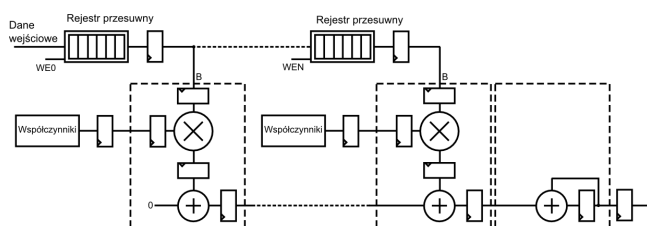
## 2.7. Filtr o strukturze szeregowo-równoległej

Konstrukcja szeregowo-równoległa stanowi kompromis pomiędzy szybkością działania, a dostępną liczbą komórek DSP48. Całość można wyobrazić sobie jako pracujące równoległe podukłady analogiczne do filtrów szeregowych.

W przypadku architektury szeregowo-równoległej (rys. 8) do każdej z komórek DSP48 skonfigurowanej jako układ mnożąco-sumujący, dołączony jest bufor danych w postaci rejestru przesuwającego na bazie tablic LUT oraz pamięć ROM. Wielkość rejestru i pamięci ROM (taka sama dla wszystkich podukładów)

uzależniona jest od liczby współczynników i od tego na ilu blokach DSP ma być zrealizowany filtr.

Konstrukcja szeregowo-równoległa jest rozwinięciem architektury kaskadowej. Sposób połączeń podukładów jest zbliżony do połączeń z jakimi mamy do czynienia w strukturze kaskadowej i również w tym przypadku wykorzystywane są dedykowane połączenia pomiędzy blokami. Rejestr występujący na wyjściu każdego bufora danych ma na celu synchronizację odczytu. Jego obecność wymusza występowanie zewnętrznego rejestru pomiędzy pamięcią współczynników a blokiem DSP.



Rys. 8. Schemat połączeń filtra FIR szeregowo-równoległego  
Fig. 8. Semi-parallel FIR filter connection diagram

Wykonanie wszystkich koniecznych operacji w celu wyliczenia jednej próbki wyjściowej, wymaga kilku dodatkowych cykli zegara. Liczba taktów zegara jest równa tzw. współczynnikowi podziału (ogólna liczba współczynników/liczba bloków DSP48).

Na końcu łańcucha umieszczony jest dodatkowy blok DSP48, jego zadaniem jest akumulacja wyników z poszczególnych cykli w celu uformowania ostatecznego rezultatu [5].

Ważnym elementem działania tej struktury jest konieczność resetowania akumulatora po ukończeniu wszystkich cykli pracy, tak aby pierwsza partia wyniku końcowego dla nowych danych wejściowych nie była zakłócona poprzednim wynikiem mieszczącym się w akumulatorze. Resetowanie można dokonać na dwa sposoby. Pierwszy sposób polega na wyczyszczeniu rejestru akumulatora poprzez standardowe wejście resetujące. Drugi sposób znacznie ciekawszy i wykorzystany w projekcie polega na zmianie, na czas jednego cyklu, trybu pracy dodatkowego bloku DSP z akumulacji na sumowanie (pierwszy operand stanowi pierwszą partię wyniku końcowego, drugi operand jest zerem). W takim przypadku aktualnie występująca wartość w akumulatorze zostaje zastąpiona wartością z pierwszego cyklu pracy.

## 2.8. Implementacje filtrów

Wszystkie rodzaje filtrów, poddane realizacji, zostały napisane w języku VHDL. W skład współczynników zaprojektowanego filtra prawie zawsze wchodzi wartości mniejsze od jedności. Bloki DSP48 są w stanie operować tylko i wyłącznie na liczbach całkowitych zapisanych w kodzie uzupełnień do dwójki. Najprostszym a jednocześnie najbardziej efektywnym sposobem przekonwertowania wyliczonych współczynników na wartości akceptowalne przez bloki DSP jest przesunięcie bitowe.

## 3. Wnioski

W przypadku filtra transponowanego jedynymi danymi przekazywanymi do entity (jednostki deklaracji), oprócz współczynników, są takie parametry jak: długość filtra, o ile bitów przesunięte zostały wartości współczynników filtra oraz w którym miejscu występuje przejście z jednej kolumny do drugiej. Aby przejść z jednej kolumny do drugiej skorzystać trzeba z wejść i wyjść ogólnego dostępu i głównej matrycy połączeń. Punkt przejścia jest więc wysoce istotny, gdyż program musi wiedzieć w którym miejscu zastosować inny rodzaj połączenia. Informacja o przeszu-

nięciu bitowym jest konieczna w celu odpowiedniego wyskalowania wyniku końcowego operacji filtracji.

Filtr ten do swego działania wymaga tylko bloków DSP48, żadna zewnętrzna logika nie jest wymagana.

Sposób przekazywania próbek wejściowych do bloków DSP (na wejścia wszystkich bloków podawana jest ta sama próbka) powoduje, że maksymalna częstotliwość pracy spada wraz z długością filtra.

Do każdego bloku DSP48 przypisany jest tylko jeden współczynnik, w efekcie nie jest konieczna żadna pamięć do ich przechowywania. Wejścia zarezerwowane dla współczynników podłączone są bezpośrednio do globalnej '1' logicznej i globalnego '0' logicznego. Układ połączeń odzwierciedla wartość danego współczynnika.

Zasadniczą część filtra o strukturze symetrycznej stanowi kod użyty podczas programowania filtra transponowanego. Jedyną różnicą jest występowanie jednostek sumujących nienależących do bloków DSP48 oraz zewnętrznego rejestru przesuwającego.

Z racji występowania rejestru przesuwającego, wykorzystującego przerzutniki D, filtr ten charakteryzuje się stosunkowo dużym zapotrzebowaniem na główne zasoby układu programowalnego. Konsekwencją występowania zewnętrznej logiki jest również mniejsza częstotliwość pracy w odniesieniu do filtra o strukturze transponowanej.

Stopień skomplikowania filtra o strukturze szeregowo-równoległej jest znacznie większy od stopnia skomplikowania filtra symetrycznego.

Współczynniki tego filtra, w odróżnieniu od pozostałych konstrukcji, nie są mapowane jako odpowiednie połączenia do globalnej '1' logicznej i globalnego '0' logicznego. Wartości wpisane jako stałe generyczne mapowane są do pamięci ROM tworzonej na bazie tablic LUT. Istnieje, oczywiście, możliwość zmapowania do dedykowanych pamięci typu RAM.

Filtr ten wymaga podania takich parametrów jak liczba bloków DSP48, na których ma być on zrealizowany, oraz wartości licznika przy której ma nastąpić „resetowanie” akumulatora.

Maksymalnie wykorzystanych może być tylko tyle komórek DSP48, ile liczy kolumna, podyktowane to jest tym, iż w przeciwnym przypadku spowodowałyby to znaczne skomplikowanie kodu programu. Dodatkowo pojawienie się bardzo długiego połączenia, które by wystąpiło w wyniku przejścia z jednej kolumny do drugiej, w znacznym stopniu wpłynęłoby na maksymalną częstotliwość pracy.

Długość tego typu filtra ograniczona jest w dużej mierze dokładnością zapisu współczynników. Dla wysokich rzędów filtra, część współczynników była tak mała, że niewystarczające okazało się zastosowanie przesunięcia bitowe.

Mimo swego skomplikowania i wykorzystywania komponentów pomocniczych takich jak: pamięć ROM i rejestry przesuwne, zużycie zasobów przez ten filtr porównywalne jest ze zużyciem zasobów przez filtr symetryczny.

## 4. Literatura

- [1] Oppenheim A. V., Shafer R. W.: Cyfrowe przetwarzanie sygnałów, WKŁ, Warszawa, 1979.
- [2] Dokumentacja techniczna, Virtex-4 FPGA User Guide, Xilinx.
- [3] Pasierbiński J., Zbysiński P.: Układy programowalne w praktyce, WKŁ, Warszawa 2001.
- [4] <http://elektronikab2b.pl/biznes/1315-fpga-od-poczatku-do-konca-czesc-pierwsza>, ElektronikaB2B, (15 listopada 2012)
- [5] Dokumentacja techniczna, XtremeDSP for Virtex-4FPGAs, Xilinx.

otrzymano / received: 27.03.2013

przyjęto do druku / accepted: 01.05.2013

artykuł recenzowany / revised paper