

Grzegorz BRZUCHALSKI, Grzegorz PASTUSZAK
POLITECHNIKA WARSZAWSKA, INSTYTUT RADIOELEKTORNIKI
ul. Nowowiejska 15/19, 00-665, Warszawa

Sprzętowa implementacja modułu transformaty kodeka MPEG-4 AAC

Mgr inż. Grzegorz BRZUCHALSKI

Uzyskał stopień magistra inżyniera o specjalności telekomunikacja w marcu 2009 roku na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej. Obecnie pracuje w Instytucie Radioelektroniki Politechniki Warszawskiej na stanowisku asystenta naukowego. Od 2008 roku zajmuje się realizacją układów w technologii FPGA do kompresji dźwięku.



e-mail: g.brzuchalski@ire.pw.edu.pl

Dr inż. Grzegorz PASTUSZAK

Uzyskał stopień magistra inżyniera o specjalności mikroelektronika w czerwcu 2001 roku a stopień doktora inżyniera o specjalności techniki multimedialne w czerwcu 2006 roku. Obecnie jest pracownikiem naukowym Instytutu Radioelektroniki Politechniki Warszawskiej. Jego obszar zainteresowania obejmuje: architektury i algorytmy VLSI, przetwarzanie obrazów/wideo/audio oraz kompresję, wydajne cyfrowe układy scalone.



e-mail: g.pastuszak@ire.pw.edu.pl

Streszczenie

Artykuł przedstawia architekturę modułu transformacji kodeka audio zgodnego ze standardem MPEG-4 AAC. Moduł oblicza MDCT, IMDCT, FFT, IFFT (wraz z wszystkimi kształtami) oraz wykorzystując te same zasoby sprzętowe, wybiera najlepiej pasującą do widma sygnału długość okna. Projektowany układ został zsyntezowany w układach Altera Stratix II i zweryfikowany z kodekiem FAAC/FAAD. Wyniki pokazują, że architektura może działać z częstotliwością 100 Mhz i obsługiwać do 93 kanałów dla częstotliwości próbkowania 44.1 kHz.

Słowa kluczowe: AAC, MDCT, IMDCT, kodowanie audio, algorytm wyboru długości okna, AAC, FPGA.

Hardware implementation of the transformation module of the MPEG-4 AAC codec

Abstract

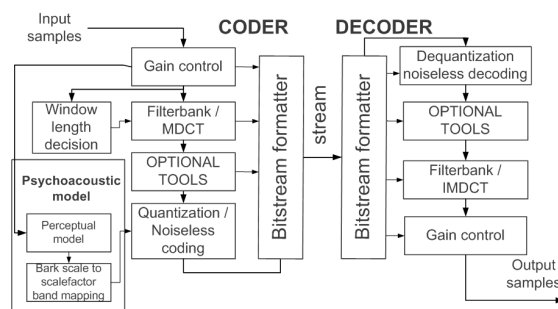
This paper presents the architecture of the audio codec transform module designed in conformity with the MPEG-4 AAC standard. The module provides forward and inverse transform computation with all defined shapes and sizes. In particular, the module computes MDCT, IMDCT, FFT, and IFFT transforms using the same hardware resources. All window shapes and FFT coefficients are stored in ROM submodules. The module automatically chooses the most suitable for current signal transform window length with reference to the signal spectrum and its energy. The algorithm is based on Free Advanced Audio Coder (FAAC) window length decision algorithm. Additionally, it supports Low Delay profile (with shorter windows). The designed module is synthesized into Altera Stratix II devices and verified with the reference to the FAAC/ Free Advanced Audio Decoder (FAAD) codec. The implementation results show that the architecture can work at 100 MHz, and it can support up to 93 channels of 44.1 kHz audio coding.

Keywords: MDCT, IMDCT, window length decision, AAC, FPGA.

1. Wstęp

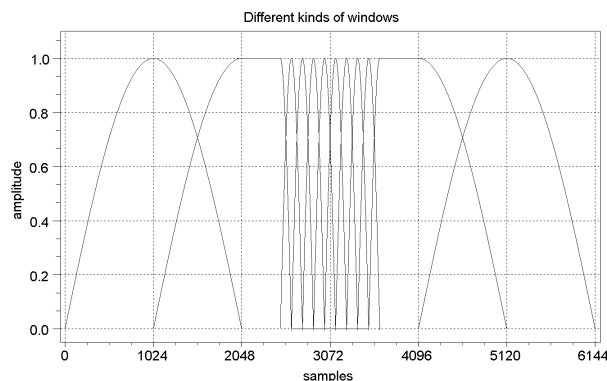
MPEG-4 Advanced Audio Coding jest najnowszym standardem stratnej kompresji dźwięku. Kodeki AAC w porównaniu z kodekami MP3 pozwalają na lepszą kompresję przy tej samej jakości dźwięku lub lepszą jakość dźwięku przy tej samej przepływności bitowej. Ogólny schemat kodeka AAC przedstawiono na rys. 1. Na początku, moduł kontroli wzmocnienia (gain control). W kolejnym kroku, moduł banku filtrów (filterbank) transformuje dane z dziedziny czasu do dziedziny częstotliwości. Ten proces odbywa się na grupie kolejnych próbek zwanej oknem. Moduł wyboru długości okna wybiera długość okna w zależności od energii widma. Opcjonalne narzędzia opisane w standardzie zwiększają efektywność kompresji, jednak w niniejszej realizacji nie zostały użyte. Kwantyzacja może być zrealizowana jako nieliniowa lub wektorowa. W tym kroku analiza psychoakustyczna ma za zadanie osiągnięcie najlepszej jakości dźwięku, dla zadanej przepływności bitowej, po stronie dekodera. Na końcu moduł formowania

strumienia (bitstream formatter) koduje skwantowane współczynniki za pomocą kodów Huffmana do elementarnego strumienia bitowego. Przebieg danych w dekodrze jest podobny do przebiegu w koderze, ale moduły są ułożone w odwrotnej kolejności. W dekodrze nie ma modułu modelu psychoakustycznego i kroków optymalizacyjnych, ponieważ wszystkie decyzje są podejmowane po stronie kodera.



Rys. 1. Diagram blokowy kodeka MPEG-4 AAC
Fig. 1. MPEG-4 AAC Block diagram

W profilu MAIN, standard definiuje cztery typy okien dla ramki o 2048 próbkach: długie (LW) 2048 próbek, serię ośmiu okien po 256 próbek (SW), okno przejściowe z długiego na krótkie (LS) i okno przejściowe z krótkiego na długie (SL) (rys. 2). Ponadto profil Low-Delay (LD) definiuje okno o 1024 próbkach. W standardzie są zdefiniowane dwa kształty okien: sinusowe (SINE) i Keiser-Bessel (KBD). W rezultacie mamy osiem rodzajów okien. Dla uniknięcia efektu blokowego kolejne okna nakładają się na siebie w połowie (rys. 2), a po transformacji bierze się tylko pierwszą połowę współczynników.



Rys. 2. Różne długości okien w standardzie AAC
Fig. 2. Different window length in the AAC standard

Zaproponowana architektura oblicza MDCT/IMDCT (Zmodyfikowaną Dyskretną Transformację Cosinusową/ Odwrotną Zmodyfikowaną Transformację Cosinusową) dla wszystkich okien w profilach Main i Low-Delay. Ponadto architektura zawiera moduł wyznaczający długość okna używany w programowej realizacji - Free Advanced Audio Coder (FAAC). Cechy te nie są spotykane w innych realizacjach.

2. Architektura

2.1. Transformata MDCT

Transformata MDCT wyraża się poniższą zależnością:

$$X_k = \sum_{n=0}^{2N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} + \frac{N}{2} \right) \left(k + \frac{1}{2} \right) \right] \quad (1)$$

Może to być przekształcone w następujący sposób:

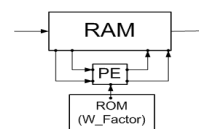
$$X_k = \sum_{n=0}^{N-1} x_n w_n \cos \left[\frac{2\pi}{N} (n + n_0) \left(k + \frac{1}{2} \right) \right] = \operatorname{Re} \left\{ \sum_{n=0}^{N-1} x_n w_n e^{-j \frac{2\pi}{N} (n + n_0) \left(k + \frac{1}{2} \right)} \right\} = \operatorname{Re} \left\{ e^{-j \frac{2\pi}{N} n_0 \left(k + \frac{1}{2} \right)} \sum_{n=0}^{N-1} x_n w_n e^{-j \frac{2\pi}{N} n \left(k + \frac{1}{2} \right)} e^{-j \frac{2\pi}{N} k n} \right\} \quad (2)$$

Jak można zauważyć MDCT może być policzone jako Dyskretna Transformata Fouriera (DFT) z krokami rotacji wstępnej (pre-twiddle) i końcowej (post-twiddle). Dekompozycja MDCT na rotacje i FFT (rys. 3) pozwala na obliczenia z użyciem tylko jednego bloku obliczającego - processing element (PE). Ponadto, blok PE pozwala na obliczenia transformaty odwrotnej. W praktyce Szybka Transformata Fouriera (FFT) jest użyta do obliczeń DFT. Jej schemat jest przedstawiony na rys. 5 (przykład dla $N = 16$). Teoretycznie 11254 (11×1024) bloki PE są potrzebne do obliczenia transformaty dla długiego okna bez iteracji. Takie podejście jest nieefektywne ze względu na dość dużą ilość zasobów sprzętowych. Dla typowych zastosowań przetwarzania audio wystarczy jeden blok PE połączony wraz z pamięcią RAM, co pokazano na rys. 4. To podejście wymienia zapotrzebowanie na zasoby sprzętowe na czas obliczeń. Zakładając ciągły proces i zoptymalizowany dostęp do pamięci architektura o jednej jednostce PE jest wystarczająca do obliczeń transformaty dla sygnału wielokanałowego. Pojedynczy moduł PE wykonuje operację zwaną „motylkiem”, która polega na obliczaniu DFT dla $N=2$. Schemat PE jest pokazany na rys. 6, gdzie zaokrąglone prostokąty reprezentują obliczenia zespolone.

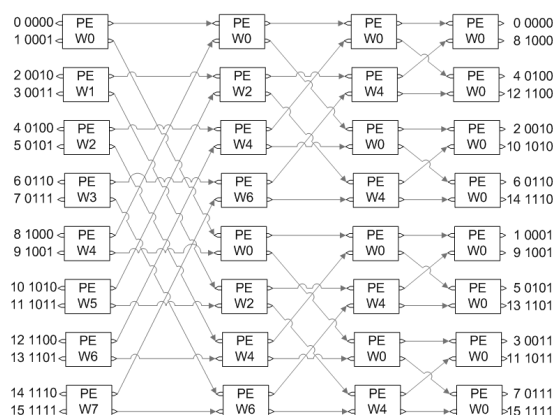
Projektowana architektura wymaga dużej ilości stałych współczynników dla obliczania transformat. Gdy współczynniki są przechowywane w oddzielnych modułach dla każdej długości/kształtu/ typu okna, to potrzebne jest około 30 modułów ROM. Taka ilość modułów pamięci jest niewygodna w technologii ASIC z powodu trudnego rozłożenia pamięci na topografii i odpowiednich połączeń. Zalecana jest zatem minimalizacja liczby modułów ROM. Ponadto, połączenie pamięci powoduje uproszczenie warstwy generującej adresy odczytu z pamięci oraz warstwy multipleksującej odczytane wyniki z pamięci. Po połączeniu odpowiednich pamięci ROM, zaproponowana architektura potrzebuje tylko dwóch modułów ROM, które mają taką samą pojemność bitową co we wcześniejszej realizacji.



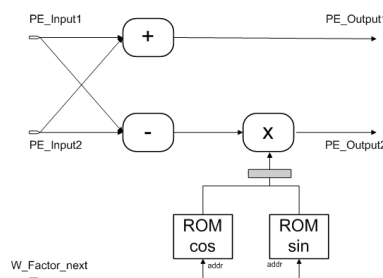
Rys. 3. Schemat algorytmu obliczeń MDCT
Fig. 3. MDCT computation algorithm scheme



Rys. 4. Koncepcja użycia jednego PE
Fig. 4. Concept of single PE usage



Rys. 5. Diagram blokowy FFT dla $N=16$
Fig. 5. FFT Block diagram for $N=16$



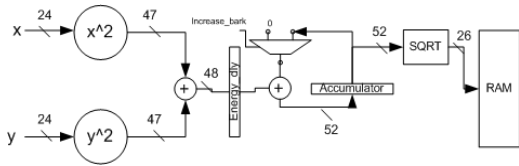
Rys. 6. Architektura modułu „motylka”
Fig. 6. Architecture of “butterfly” module

Większość współczynników przechowywanych w ROM to wartości zespolone, które trafiają na wejście PE w tym samym cyklu zegarowym. Aby odczytać dwie wartości w jednym cyklu zegarowym pierwszy moduł ma podwojoną długość słowa (2×19 bitów), a drugi jest taktowany zegarem o dwa razy większej częstotliwości.

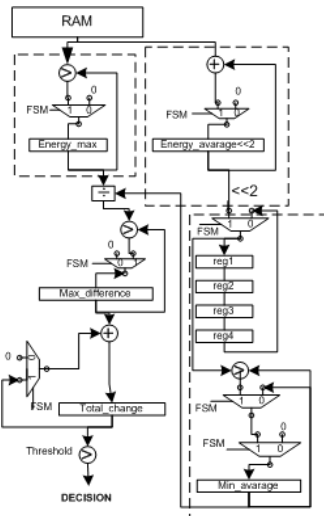
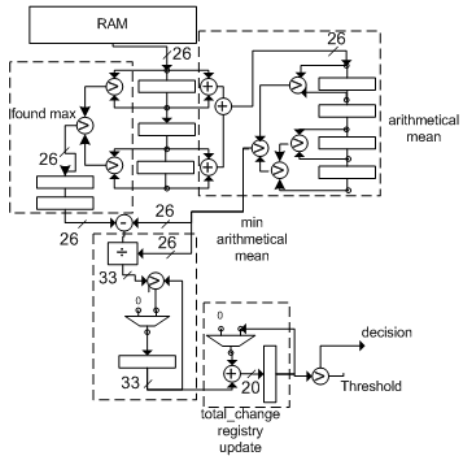
2.2. Moduł wyboru długości okna dla MDCT

Moduł wyboru długości okna nie jest aktywny dla trybu LD, ponieważ w tym trybie występuje tylko jedna długość okna. Dla zrealizowania algorytmu decyzji posłużono się algorytmem z realizacji programowej FAAC. Algorytm bazuje na analizie zmian energii w pasmach dla trzech kolejnych okien (obecnego, poprzedniego i następnego). Proces obliczeniowy składa się z dwóch części odpowiedzialnych za akumulację energii w pasmach i porównanie jej celem podjęcia decyzji. Architektura pierwszej części przedstawiona jest na rys. 7. Składowe współczynniki FFT są podniesione do kwadratu i zsumowane dając energię. Dla każdego pasma energia jest akumulowana i zapisywana w pamięci. Druga część algorytmu wyboru okna bazuje na analizie energetycznej. Na początku algorytm znajduje, z czterech kolejnych pasm, pasmo o największej energii a następnie i oblicza średnią arytmetyczną energii tych pasm. Następnie algorytm znajduje najmniejszą średnią spośród pięciu kolejnych obliczonych średnich. Znaleziona maksymalna energia jest dzielona przez minimalną średnią. Wynik dzielenia jest akumulowany. Powyższe operacje są powtarzane dla wszystkich pasm. Na koniec wartość

akumulatora jest porównywana z progiem i podjęta zostaje odpowiednia decyzja.



Rys. 7. Architektura części akumulacyjnej algorytmu WLD
 Fig. 7. Architecture of accumulation part of the window-length decision



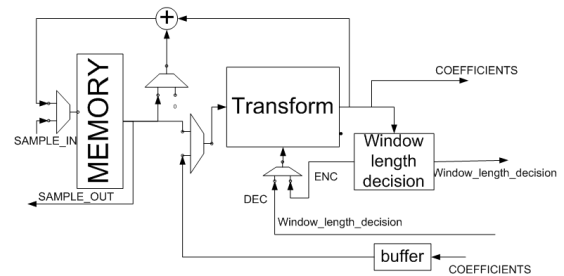
Rys. 7. Schemat bezpośredniej implementacji algorytmu WLD oraz zaprojektowany układ algorytmu WLD
 Fig. 7. Schema of the direct implementation of the WLD algorithm and its realization

Algorytm wyboru długości okna (WLD – Window Length Decision) jest zmodyfikowany aby zużywał mniej zasobów w realizacji sprzętowej. W szczególności, podukład obliczający pierwiastek kwadratowy jest przesunięty do pierwszej części algorytmu (tak jak na rys. 7). Z tego powodu bezpośredni pomiar energii nie jest możliwy w dalszej analizie. Okazuje się, że taka zmiana algorytmu znacząco zmniejsza zapotrzebowanie na zasoby sprzętowe (w szczególności szerokość bitowa pamięci RAM zmniejsza się dwukrotnie), dając bardzo podobne wyniki względem oryginalnego algorytmu (ok 1% różnic w decyzjach). Bezpośrednie mapowanie omawianego algorytmu (jego części drugiej) na układ cyfrowy jest pokazane na rys. 7. Taka realizacja zużywa pięć sumatorów/ subtraktorów i dziewięć komparatorów, które właściwie są subtraktorami. Poprzez transformację takiego układu według schematu pokazanego na rys. 7, zużycie sumatorów i komparatorów zostało ograniczone do sześciu. Liczba rejestrów

została ograniczona z 11 do 9. Choć przetransformowany układ wymaga więcej taktów zegara do podjęcia decyzji, to nie ma to znaczenia dla opóźnienia całości układu, ponieważ opóźnienie całego układu wyboru okna jest zależne od jednostki dzielącej.

2.3. Przepływ danych

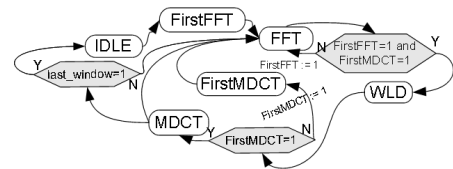
Na rys. 8 jest widoczny moduł transformacji. Znajdują się tu dwie ścieżki przepływu danych związanych z kodowaniem i dekodowaniem. Ścieżka kodowania zaczyna się od źródła sygnału audio (interfejs I2S lub przetwornik A/C). Próbkę są buforowane w pamięci używanej do synchronizacji źródła dźwięku i modułu obliczającego transformatę. Ponadto, pamięć pozwala na wykonanie nakładania się okien. Kontroler wybiera odpowiednie próbki i podaje je do modułu transformacji. Bazujący na FFT moduł wyboru długości okna WLD inicjuje obliczenia MDCT, a wyniki transformaty – współczynniki częstotliwościowe trafiają do następnego modułu kodera.



Rys. 8. Schemat blokowy modułu transformaty
 Fig. 8. Transform block diagram

Ścieżka dekodera zaczyna się od bufora który przechowuje zdekwantyzowane współczynniki IMDCT. Ten bufor jest potrzebny aby podawać współczynniki do modułu transformacji IMDCT w odpowiedniej kolejności. Dla drugiej części okna wyniki IMDCT są zapisywane bezpośrednio do pamięci. Próbkę z pierwszej części okna są dodawane do odpowiednich próbek drugiej części poprzedniego okna (odczytanych z pamięci). Taka operacja kończy składanie okien po stronie dekodera. Na koniec zrekonstruowane próbki są odczytywane z pamięci i podawane do odbiornika (np kontrolera I2S, przetwornika C/A).

Zaprojektowany układ oblicza MDCT, IMDCT, FFT, IFFT, dla wielu kanałów audio. Ponadto potrafi dokonać odpowiedniego wyboru długości okna. W profilu LD sterowanie jest prostsze, ponieważ tam są tylko transformacje MDCT i IMDCT. Odpowiednie przejścia między transformacjami i fakt, że obliczenia pierwszych okien są inne także są wzięte pod uwagę.



Rys. 9. Schemat automatu WLD
 Fig. 9. Window length decision finite state machine

Schemat automatu stanu (FSM) jest przedstawiony na rys. 9. Praca automatu rozpoczyna się w stanie IDLE. Kiedy odpowiednia ilość próbek zostanie zapisana w pamięci, to pierwsze FFT jest obliczane. Następnie dwie kolejne FFT są obliczane. Te trzy FFT są potrzebne do wyznaczenia długości pierwszego okna dla pierwszego MDCT. Na podjęcie decyzji w WLD potrzebne jest ok 100 cykli zegarowych. Następnym krokiem jest obliczenie pierwszego MDCT. Od tego miejsca nowe FFT, WLD i MDCT są cyklicznie obliczane dla kolejnych okien. Automat jest dużo prostszy dla dekodera, ponieważ długość okna jest odczytywana ze strumienia

binarnego i obliczane jest tylko IMDCT. Opisana powyżej kolejność operacji jest odpowiednia dla kodowania/ dekodowania dla jednokanałowego sygnału audio. Koder/ dekodery wielokanałowy działa podobnie do jednokanałowego. Jedyną różnicą jest fakt, że cykl operacji (FFT, WLD, MDCT) jest powtarzany dla każdego kanału przed obliczeniami następnego okna.

3. Porównanie wyników i wnioski

Prezentowana architektura jest zweryfikowana w symulacjach pod względem funkcjonalności i wydajności. Dla uzyskania danych weryfikacyjnych użyto oprogramowania Free Advanced Audio Coder/ Decoder (FAAC/FAAD). Kodeki audio zazwyczaj są realizowane na procesorach DSP. Znaleźliśmy nieliczne implementacje AAC projektowane w technologiach FPGA lub ASIC. Zaprojektowany układ potrzebuje 15360 cykli zegarowych dla MDCT lub IMDCT z użyciem kształtów okien i nakładaniem się okien. Zsyntezowana architektura wykorzystuje 3895 ALUT'ów (w tym 1433 to WLD). Maksymalna częstotliwość zegara układu wynosi 100 MHz dla układu FPGA Altera Stratix II. Układ potrafi obsłużyć wielokanałowe kodowanie/ dekodowanie AAC, np. dla częstotliwości zegara równej 100 MHz pozwala na obliczenie 93 kanałów w profilu Main (z WLD).

Tab. 1. Porównanie układów
Tab. 1. Design comparison

system	Fmax[MHz]	Cykle CLK
[4]	-	525k
[3]	-	15362
[1]	54,3	12288
[2]	-	795k
Nasz	100	15360

Większość rozwiązań używa FFT do obliczeń MDCT/IMDCT. Jest kilka algorytmów obliczeń FFT. Porównanie z innymi algorytmami MDCT/IMDCT zamieszczono w tab.1. Zaproponowany układ pod względem wydajności przewyższa pozostałe przedstawione w literaturze.

Jak można zauważyć, przedstawiona architektura potrzebuje najmniejszą liczbę cykli zegarowych z wyjątkiem [3]. Jednakże, maksymalna częstotliwość zaproponowanego układu jest niemalże dwa razy większa. Zużycie pamięci jest większe niż w pozostałych architekturach co jest spowodowane rozszerzoną funkcjonalnością (FFT, IFFT, MDCT, IMDCT, kształty okien, tryb wielokanałowy i LD). Dla zredukowanej funkcjonalności zużycie pamięci byłoby podobne do pozostałych architektur.

Artykuł ten przedstawia architekturę transformacji MDCT/IMDCT, które mogą być także użyte jako FFT/ IFFT. MDCT jest obliczane wraz z wyborem długości okna dla sygnału wielokanałowego oraz trybu LD.

Praca jest elementem projektu "Zintegrowany mobilny system wspomagający działania antyterrorystyczne i antykrzysowe - PROTEUS", realizowanego dzięki wsparciu fin. Europ. Funduszu Rozwoju Reg. w ramach I. osi priorytetowej Programu Operacyjnego Innowacyjna Gospodarka, 2007 - 2013, Poddziałanie 1.1.2, projekt numer PO-IG.01.02.01-00-014/08-00.

4. Literatura

- [1] Du F., Du G., Song Y., Zhang D., Gao M.: An implementation of filterbank for mpeg-2 aac on fpga. In Anti-counterfeiting, 2008. ASID 2008. 2nd International Conference on Security and Identification.
- [2] Li L., Miao H., Li X., Guo D.: Efficient architectures of mdct/imdct implementation for mpeg audio codec. In Anti-counterfeiting, ASID 2009.3rd International Conference on Security, and Identification in Communication.
- [3] Tsai T.H., Liu C.N. (2009): Low-power system design for mpeg-2/4 aac audio decoder using pure asic approach. Trans. Cir. Sys. Part I, 56(1), 144-155.
- [4] Tsai T.H., Liu J.N.: Architecture design for mpeg-2 aac filterbank decoder using modified regressive method. In Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on, volume 3, III{3216 { III{3219 vol. 3.

otrzymano / received: 15.10.2010

przyjęto do druku / accepted: 01.12.2010

artykuł recenzowany

INFORMACJE

Nowa inicjatywa PAK

Na stronie internetowej Wydawnictwa PAK został utworzony dział: **Niepewność wyników pomiarów** w którym są zamieszczane aktualne informacje dotyczące problemów teoretycznych i praktycznych związanych z szacowaniem niepewności wyników pomiarów. W dziale znajdują się:

- aktualne informacje o publikacjach dotyczących niepewności wyników,
- informacje o przedsięwzięciach naukowo–technicznych i edukacyjnych, o tematyce związanej z niepewnością,
- dokumenty dotyczące niepewności,
- pytania do ekspertów (FAQs).

Zapraszamy:

- autorów opublikowanych prac dotyczących niepewności o nadsyłanie tekstów do zamieszczenia w tym dziale,
- organizatorów przedsięwzięć naukowo – technicznych lub edukacyjnych do nadsyłania informacji o imprezach planowanych lub odbytych,
- zainteresowanych zagadnieniami szczegółowymi do nadsyłania pytań do ekspertów.

Materiały mogą mieć formę plików lub linków do źródeł. Warunkiem zamieszczenia w tym dziale strony internetowej PAK materiałów lub linków jest przysłanie do redakcji PAK pocztą zwykłą zgody właściciela praw autorskich na takie rozpowszechnienie. Zamieszczanie i pobieranie materiałów i informacji w tym dziale strony internetowej jest bezpłatne. Redakcja PAK będzie nadzorować zawartość działu, ale za szczegółowe treści merytoryczne odpowiadają autorzy nadsyłanych materiałów.