Krzysztof KONOPKO

Politechnika Białostocka, Wydział Elektryczny, Białystok, Polska,

Zastosowanie procesorów masowo-równoległych w addytywnej syntezie sygnałów.

Streszczenie: *Moc obliczeniowa współczesnych procesorów graficznych GPU (ang. Graphics Processing Unit), stosujących architekturę masowo-równoległą, jest wykorzystywana w wielu dziedzinach inżynierii. Do obszarów stosowania GPU zaliczyć można między innymi: badania aerodynamiczne, symulowanie przepływu płynów, dyspersji cząsteczek czy efektów kolizji. W artykule przedstawiono zastosowanie procesorów masowo-równoległych w addytywnej syntezie sygnałów.*

Abstract*: The computational power of modern Graphics Processing Units (GPUs), using a massively parallel architecture, is used in many fields of engineering. The GPUs are used in variety of applications ranging from aerodynamic testing through a fluid flow simulation to a dispersion of particles and research on the effects of collisions. The paper presents the use of the massively parallel processors for additive synthesis.* ***The use of massively parallel processors in additive synthesis.***

Słowa kluczowe: synteza sygnałów, procesory masowo-równoległe, opencl.
Keywords: additive synthesis, massively parallel processors, opencl.

doi:10.12915/pe.2014.11.59

Wprowadzenie

Jednym z rozwiązań mających na celu zwiększenie wydajności obliczeniowej współczesnych procesorów jest wykorzystanie architektury układów graficznych GPU (ang. Graphics Processing Unit). Architektura ta zakłada realizację relatywnie prostych obliczeń na bardzo dużej liczbie danych. W przypadku obliczeń związanych z grafiką dotyczą one przede wszystkim przetwarzania wierzchołków wielokątów i nakładania tekstur. Dlatego w układach graficznych stosuje się bardzo dużo prostych jednostek obliczeniowych, które pracują równolegle.

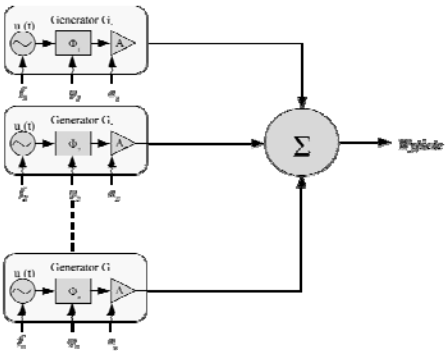
Ogromne możliwości przyśpieszenia obliczeń dzięki wykorzystaniu architektury procesorów masowo-równoległych skutkują publikowaniem coraz większej liczby prac poświęconych GPU w zagadnieniach nie związanych z przetwarzaniem grafiki [1, 2]. Niniejszy artykuł przedstawia wykorzystanie GPU w addytywnej syntezie sygnałów.

Metody syntezy sygnałów.

W zależności od sposobu generacji sygnałów algorytmy syntezy można podzielić na 4 grupy [3, 4]: metody przetwarzania zapisu, metody widmowe, metody abstrakcyjne oraz modelownie fizyczne. Przykładem algorytmu z pierwszej grupy są metody tablicowe polegające na generowaniu sygnału na podstawie rejestrowanych i przetwarzanych rzeczywistych przebiegach. Do drugiej grupy zalicza się metodę addytywną polegającą na syntezie sygnałów za pomocą elementarnych przebiegów oraz metodę subtraktywną polegającą na filtracji sygnałów szerokopasmowych. Do grupy algorytmów abstrakcyjnych zaliczyć można syntezę FM, metodę kształtowania fali polegającą na nieliniowym przetwarzaniu sygnałów harmonicznych oraz metody oparte na chaosie deterministycznym wykorzystywane między innymi do generacji przebiegów losowych. Do ostatniej grupy zaliczyć można modelowanie matematyczne, modelowanie falowodowe oraz metody komórkowe polegające na modelowaniu, najczęściej z zastosowaniem metod elementów skończonych, źródeł sygnału w postaci elementarnych cząstek masy (tzw. komórek) powiązanych ze sobą przy pomocy wiązań sprężystych.

Synteza addytywna

Jedną z najstarszych metod generacji sygnałów jest synteza addytywna. Polega ona na generacji sygnału za pomocą elementarnych przebiegów sinusoidalnych zgodnie z algorytmem pokazanym na rysunku 1.



Rys. 1. Realizacja addytywnej syntezy sygnałów

Aby wygenerować sygnał metodą addytywną należy określić dla każdego generatora funkcje opisujące zmienność amplitudy oraz kąta fazowego. Właśnie konieczność określenia parametrów ogromnej liczby składowych stanowi jedną z podstawowych wad tej metody. Dlatego parametry do syntezy uzyskuje się najczęściej analizując sygnał rzeczywisty. W tym przypadku synteza (a właściwie resynteza) przebiega na podstawie analizy dynamicznego widma sygnału. Dwie najczęściej stosowane metody analizy widma to metoda wokodera fazowego oraz metoda McAulay'a-Quatieri'ego (MQ) [5]. Metoda wokodera fazowego zakłada, że sygnał poddawany analizie, a później resyntezie jest określony następującą zależnością (1):

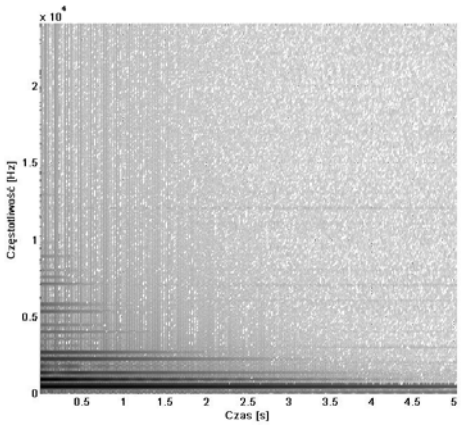
$$x(n)=\sum_{k=0}^{M} A_k(n)\sin\left(nT\left[2\pi f_0+2\pi\Delta f_k(n)\right]\right) \tag{1}$$

gdzie: $x(n)$ jest syntezowanym sygnałem, T okresem próbkowania, M liczbą harmonicznych, A_k amplitudą k-tej harmonicznej, f_k odchyłką częstotliwości k-tej harmonicznej, a f_0 częstotliwością składowej podstawowej.

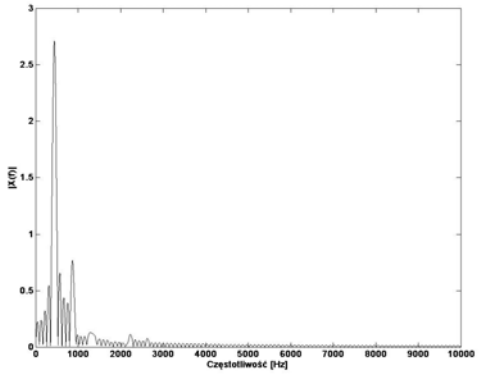
W procesie analizy dokonuje się ekstrakcji składowych widmowych oraz wyznacza częstotliwości poszczególnych harmonicznych z użyciem metody dopasowania długości okna analizy do okresu analizowanego sygnału (stąd metoda ta bywa czasem nazywana „metodą strojonego okna czasowego"). Następnie na bazie uzyskanych danych przeprowadza się resyntezę. Podstawową wadą metody wokodera fazowego jest brak możliwości syntezy często wykorzystywanych sygnałów nieharmonicznych [6]. Metoda McAulay'a-Quatieri'ego działa w oparciu o wyszukiwanie i interpolację lokalnych maksimów widma dynamicznego. W procesie analizy wykorzystuje się szybką transformatę

Fouriera (FFT). Otrzymane w ten sposób widma chwilowe poddaje się procesowi wyszukiwania lokalnych maksimów. Uzyskane w ten sposób dane są podstawą do procesu resyntezy. W metodzie MQ, w odróżnieniu od metody wokodera fazowego nie ma konieczności dopasowania długości okna analizy do długości okresu sygnału. Również częstotliwości odpowiadające maksimom lokalnym nie muszą spełniać warunku harmoniczności co skutkuje możliwością syntezy sygnałów nieharmonicznych. Podstawową wadą wynikającą z wykorzystywania w metodzie MQ krótkookresowej transformaty Fouriera jest zależność rozdzielczości analizy częstotliwościowej i czasowej sygnału wpływająca na „rozmycie" widma i skutkująca trudnością z precyzyjnym i automatycznym wyznaczaniem maksimów.

Na rysunku 2 pokazana została charakterystyka czasowo-częstotliwościowa wyznaczona dla sygnału zarejestrowanego z gitary elektrycznej (Fender Squirt Bullet Star) z częstotliwością próbkowania 48 kHz i rozdzielczością 24 bitów. Zaś na rysunku 3 przedstawiono krótkookresową transformatę Fouriera otrzymaną na podstawie fragmentu zarejestrowanego sygnału złożonego z 512 próbek pobranych po pierwszej sekundzie jego trwania.



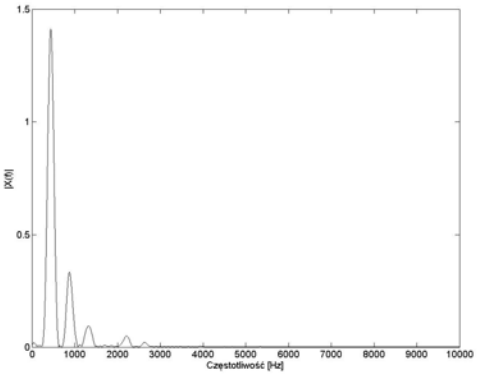
Rys. 2. Spektrogram uzyskany dla sygnału zarejestrowanego z gitary elektrycznej Fender Squirt Bullet z częstotliwością próbkowania 48 kHz i rozdzielczością 24 bitów



Rys. 3. Transformata Fouriera wyznaczona na podstawie fragmentu sygnału (zarejestrowanego z gitary elektrycznej Fender Squirt Bullet z częstotliwością próbkowania 48 kHz i rozdzielczością 24 bitów) złożonego z 512 próbek pobranych po pierwszej sekundzie jego trwania

Na wykresie 3 wyraźnie widać nie tylko „rozmycie" widma spowodowane krótkim czasem trwania okna analizy, ale także wpływ jego charakterystyki na wynikowe widmo. Zmniejszenie tego wpływu uzyskuje się poprzez stosowanie odpowiednio dobranych funkcji okna. Na rysunku 4 przedstawiono charakterystykę amplitudową tego samego sygnału poddanego okienkowaniu z zastosowaniem funkcji Hamminga (2).

(2) $$w(n) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi n}{N}\right)$$



Rys. 4. Transformata Fouriera wyznaczona na podstawie fragmentu sygnału (zarejestrowanego z gitary elektrycznej Fender Squirt Bullet z częstotliwością próbkowania 48 kHz i rozdzielczością 24 bitów) złożonego z 512 próbek pobranych po pierwszej sekundzie jego trwania poddanego okienkowaniu z zastosowaniem funkcji Hamminga.

Dalszą poprawę precyzji analizy można osiągnąć zastępując FFT metodą modelowania parametrycznego [7]. W tym przypadku estymacja charakterystyk widmowych odbywa się dwuetapowo. W kroku pierwszym wyznaczana jest transmitancja takiego układu liniowego, który po pobudzeniu białym szumem generuje na wyjściu analizowany sygnał. W kroku drugim na podstawie wyznaczonej transmitancji estymuje się charakterystyki widmowe. W zależności od przyjętej metody modelowania wyróżnia się metodę AR (Autoregressive), MA (Moving Average) lub ARMA. Model ARMA jest opisany za pomocą następującego równania różnicowego:

(3) $$x[n] = -\sum_{k=1}^{p} a[k]x[n-k] + \sum_{k=0}^{q} b[k]u[n-k]$$

gdzie: x jest analizowanym sygnałem, a u losowym procesem o rozkładzie normalnym.

Dla sygnału u w postaci białego szumu o wariancji σ^2 gęstość mocy modelowanego sygnału opisana jest następującą zależnością:

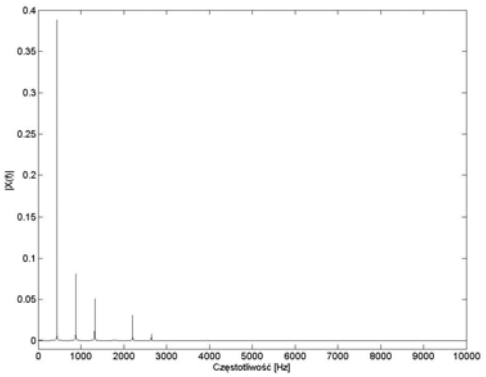
(4) $$P_{ARMA}(f) = \sigma^2 \left|\frac{B(f)}{A(f)}\right|^2$$

Model ARMA dobrze nadaje się do analizy sygnałów o złożonej strukturze widmowej, model AR jest efektywny dla sygnałów o widmie prążkowym, zaś model MA dla sygnałów szerokopasmowych. Na wykresie 5 przedstawiona jest charakterystyka widmowa otrzymana na podstawie modelu AR.

Zastosowanie parametrycznego modelowania pozwala zachować wysoką rozdzielczość analizy częstotliwościowej przy stosunkowo krótkich oknach czasowych. Wadą metod parametrycznych jest konieczność posiadania pewnej wiedzy a-priori dotyczącej modelowanych sygnałów. Na przykład algorytmy wykorzystujące modele AR i ARMA dobrze sprawdzają się w analizie sygnałów, których widma zmieniają się niezbyt szybko.

Oczywiście etap analizy może odbywać się na podstawie pobranych próbek sygnału i nie musi być realizowany w czasie rzeczywistym. Resynteza związana z generacją wymusza zazwyczaj realizację w czasie rzeczywistym. Przy dość skomplikowanej strukturze widma syntezowanego sygnału wymaga to stosowania znacznych

mocy obliczeniowych. Dodatkowo przebieg syntezowany często poddawany jest przetwarzaniu, umożliwiającemu realizację różnych efektów czy symulacji różnych środowisk, co jeszcze bardziej wpływa na zwiększanie wymaganych mocy obliczeniowych. Zapewnienie realizacji syntezy i przetwarzania w czasie rzeczywistym może być zrealizowane z zastosowaniem współczesnych procesorów GPU.



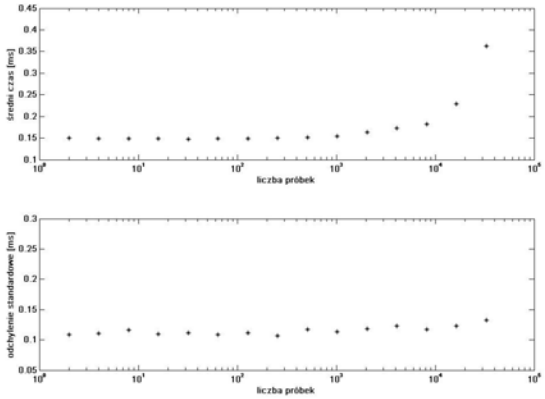
Rys. 5. Charakterystyka widmowa wyznaczona z zastosowaniem modelu AR dla fragmentu sygnału (zarejestrowanego z gitary elektrycznej Fender Squirt Bullet z częstotliwością próbkowania 48 kHz i rozdzielczością 24 bitów) złożonego z 512 próbek pobranych po pierwszej sekundzie jego trwania

Procesory graficzne (GPU) jako przykład architektury masowo-równoległej

Początkowo układy graficzne funkcjonowały jako elementy systemów komputerowych o sztywno określonych funkcjach. Jednak wzrost wymagań związanych z przetwarzaniem grafiki doprowadził do przedstawienia przez firmę 3dfx w roku 1996 pierwszego akceleratora grafiki 3D. Jednak jako pierwsza terminu GPU użyła firma NVIDIA wprowadzając w roku 1999 na rynek kartę graficzną GeForce 256. Karta ta wyposażona była między innymi w sprzętowe jednostki T&L (ang. Transform and Lighting). Od tego momentu rozwój GPU znacząco przyśpieszył. W roku 2001 wprowadzono programowalne jednostki cieniujące umożliwiające wyliczanie koloru pikseli (ang. Pixel Shader) oraz transformacje położenia wierzchołków z przestrzeni 3D na współrzędne 2D (ang. Vertex Shader). Jednostki te zostały zunifikowane we wprowadzonych w 2006 roku procesorach graficznych. Dzięki unifikacji jednostek oraz zwiększeniu ich liczby znaczenie poszerzyły się możliwości wykorzystania procesorów graficznych w obliczeniach ogólnego przeznaczenia. Najnowsze procesory graficzne wykorzystują tysiące tzw. procesorów strumieniowych SP (ang. stream processors) (np. Radeon R9 290X posiada 2816, a GeForce GTX 780Ti - 2880 procesorów strumieniowych) umożliwiając realizację obliczeń zmiennoprzecinkowych przekraczających 5000 GFLOPS. Drugą bardzo ważną cechą współczesnych kart graficznych, wpływających na wzrost ich wykorzystywania, jest duża przepustowość pamięci wynosząca np. dla procesora Radeon R9 290X 320 GB/s, a dla procesora GeForce GTX 780Ti 336 GB/s.

Addytywna synteza sygnałów z zastosowaniem GPU

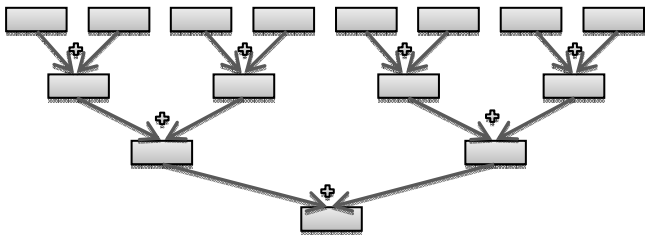
Zastosowanie GPU w addytywnej syntezie oprócz podstawowych zalet związanych z cyfrowym przetwarzaniem sygnałów posiada również pewne wady. Zaliczyć do nich można wprowadzanie zmiennego opóźnienie (tzw. latencja) związanego z realizacją obliczeń w wielozadaniowym systemie komputerowym. Wzrost opóźnienia i jego zmienność uwidacznia się szczególnie w przypadku stosowania pojedynczej karty graficznej, która dodatkowo wykorzystywana jest do przetwarzania i wizualizacji grafiki. Na rysunku 6 przedstawiono czasy wymagane do transmisji danych z wewnętrznej pamięci karty graficznej zmierzone w środowisku OpenCL [8] (*ang.* Open Computing Language) dla karty graficznej Asus EAH5830 DirectCU/2DIS/1GD5 wyposażonej w procesor Radeon HD 5830.



Rys. 6. Średnie czasy (rysunek górny) wymagane do transmisji danych z wewnętrznej pamięci karty graficznej oraz odchylenie standardowe od średniego czasu transmisji w zależności od liczby przesłanych próbek zmierzone na podstawie 100000 transmisji zrealizowanych w środowisku OpenCL dla karty graficznej Asus EAH5830 DirectCU/2DIS/1GD5 wyposażonej w procesor Radeon HD 5830

Na podstawie dokonanych pomiarów można stwierdzić, że średnie opóźnienie transmisji danych w zakresie od kilku do około 1000 próbek (przy próbkowaniu z częstotliwością 48 kHz odpowiada to czasowi równemu 20,8 ms) wynosi około 0,15 ms. Na uwagę zasługuje jednak dosyć duża jego zmienność (odchylenie standardowe od średniego opóźnienia wynosi około 0,1 ms).

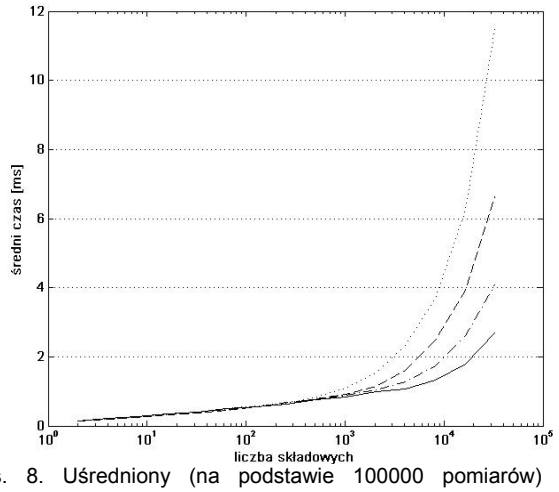
Drugim czynnikiem zwiększającym opóźnienie jest sama procedura syntezy. Zgodnie z algorytmem pokazanym na rysunku 1 przebiega ona dwuetapowo. W pierwszym kroku wyznaczane są chwilowe wartości wszystkich składowych, zaś w drugim następuje ich sumowanie. Pierwszy etap umożliwia bardzo dobre wykorzystanie architektury GPU, w której możliwe jest równoległe wyznaczanie wartości bardzo wielu (setek, a nawet tysięcy) składowych. Zrównoleglenie etapu sumowania może przebiegać np. z zastosowaniem pokazanej na rysunku 7 prostej procedury redukcji umożliwiającej nawet kilkudziesięciokrotne zredukowanie czasu trwania obliczeń.



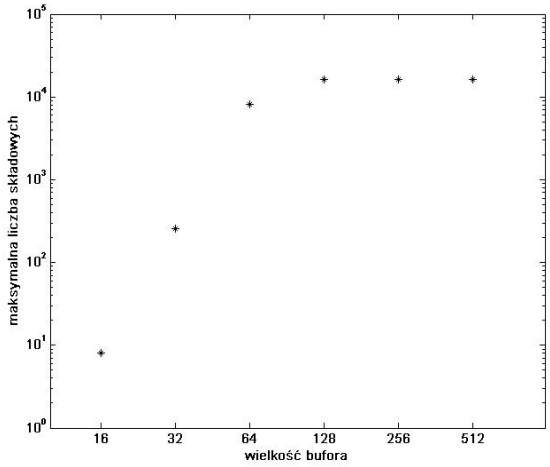
Rys. 7. Algorytm redukcji umożliwiający wykorzystanie przetwarzania równoległego w procedurze sumowania danych

Na rysunku 8 pokazano średni czas realizacji procedury syntezy (zmierzony w środowisku OpenCL dla karty graficznej Asus EAH5830) w zależności od liczby składowych. Zaś na rysunku 9 przedstawiono maksymalną liczbę składowych, dla sygnału generowanego z częstotliwością próbkowania równą 48 kHz, w zależności od wielkości bufora. Z analizy otrzymanych danych wynika, że

przy wielkości bufora złożonego z 32 próbek (przy próbkowaniu z częstotliwością 48 kHz odpowiada to czasowi równemu 0,67 ms) możliwa jest generacja sygnału złożonego z ponad 200 składowych, dla bufora złożonego z 64 próbek (przy próbkowaniu z częstotliwością 48 kHz odpowiada to czasowi równemu 1,33 ms) możliwa jest synteza sygnału złożonego z ponad 8000 składowych, a dla bufora złożonego z ponad 128 próbek (przy próbkowaniu z częstotliwością 48 kHz odpowiada to czasowi równemu 2,67 ms) możliwa jest synteza sygnału złożonego z ponad 16000 składowych.



Rys. 8. Uśredniony (na podstawie 100000 pomiarów) czas realizacji procedury syntezy sygnału (zmierzony w środowisku OpenCL dla karty graficznej Asus EAH5830) w zależności od wielkości wyjściowego bufora danych (linia ciągła - 64 próbki, linia złożona z kresek i kropek - 128 próbek, linia kreskowana - 256 próbek i linia kropkowana - 512 próbek)



Rys. 9. Maksymalna (wyznaczona na podstawie 100000 pomiarów) liczba składowych, dla sygnału generowanego z częstotliwością próbkowania równą 48 kHz, w zależności od wielkości wyjściowego bufora danych (zmierzona w środowisku OpenCL, dla karty graficznej Asus EAH5830)

Podsumowanie

Ogromne możliwości przyśpieszenia obliczeń dzięki wykorzystaniu architektury procesorów masowo-równoległych powodują coraz częstsze stosowanie GPU zarówno w aplikacjach inżynieryjnych jak i badaniach naukowych. Moc obliczeniowa i elastyczność narzędzi programistycznych sprzyjają również wykorzystaniu współczesnych procesorów graficznych w implementacji algorytmów wymagających realizacji w czasie rzeczywistym. Przykładem takiej aplikacji jest addytywna synteza sygnałów.

Jak pokazano w artykule zastosowanie GPU umożliwia realizację syntezy przebiegu złożonego z wielu tysięcy składowych przy jednocześnie wprowadzanym niewielkim (na poziomie pojedynczych milisekund) opóźnieniu. Dodatkowo w artykule zaproponowano modyfikację standardowej procedury analizy sygnałów poprzez zastąpienie szybkiej transformaty Fouriera algorytmami parametrycznej estymacji widma. Modyfikacja ta umożliwia zwiększenie rozdzielczości częstotliwościowej analizy przy braku konieczności zwiększania szerokości okna w dziedzinie czasu.

LITERATURA

[1] Stakhiv P., Strubytska I., Kozak Y. „Parallelization of calculations using GPU in optimization approach for macromodels construction”, Przegląd Elektrotechniczny, 03a/2012 Str. 7-9.

[2] Dąbrowski A., Pawłowski P., Stankiewicz M., Misiorek F. „Fast and accurate digital signal processing realized with GPU technology”, Przegląd Elektrotechniczny, 06/2012 Str. 47-50.

[3] Czyżewski A.: „Dźwięk cyfrowy : wybrane zagadnienia teoretyczne, technologia, zastosowania”, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001.

[4] Kirn P.: „Real World Digital Audio”, Peachpit Press, Berkeley 2005.

[5] McAulay, R. J.; Quatieri, T. F.: "Speech analysis/synthesis based on a sinusoidal representation". IEEE Transactions on Acoustics, Speech, Signal Processing ASSP-34: 744–754., 1986.

[6] Pawłowski P., Portalski M., Portalska H., „Generacja wielotonów nieharmonicznych z wykorzystaniem procesorów sygnałowych”, Przegląd Elektrotechniczny, 10/2013 str. 122-125.

[7] Grishin Y., Konopko K.: “Metody estymacji widma w zagadnieniach wykrywania sygnałów na tle zakłóceń wąskopasmowych” XI Konferencji Naukowej : Sterowanie w radiolokacji i obiektach latających, Jelenia Góra 2000.

[8] Khronos Group, The OpenCL Specification Version 1.2, A. Munshi, ed. Khronos Group, 2012.

***Author**: dr inż. Krzysztof Konopko, Politechnika Białostocka, Wydział Elektryczny, ul. Wiejska 45D, 15-351 Białystok, e-mail: krzysiek@teleinfo.pb.edu.pl;*