Miron KŁOSOWSKI, Bogdan PANKIEWICZ, Marek WÓJCIKOWSKI

Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska

doi:10.15199/48.2015.09.25

Akcelerator transformacji DCT do kompresji obrazu w sensorach wizyjnych

Streszczenie. *W komunikacie przedstawiono konfigurowalny cyfrowy akcelerator transformacji DCT przeznaczony dla enkodera wideo standardu H.264. Akcelerator realizuje także odwrotną transformację DCT oraz kwantyzację i dekwantyzację. Akcelerator początkowo zaimplementowano w układzie FPGA. Został on pomyślnie zweryfikowany, a następnie zaimplementowany w układzie ASIC w technologii UMC 90 nm. Szczegółowe wyniki testów akceleratora ASIC zostały porównane z innymi dostępnymi w literaturze. Funkcjonalność akceleratora została szczegółowo opisana w komunikacie. System testujący został zoptymalizowany do współpracy z programem x.264 pracującym pod kontrolą systemu Linux i jest przeznaczony do sprzętowego wspierania kompresji wideo w standardzie HD. Ze względu na niewielki pobór mocy oraz małą powierzchnię rdzenia opisany akcelerator może łatwo zostać zintegrowany z sensorem wizyjnym.*

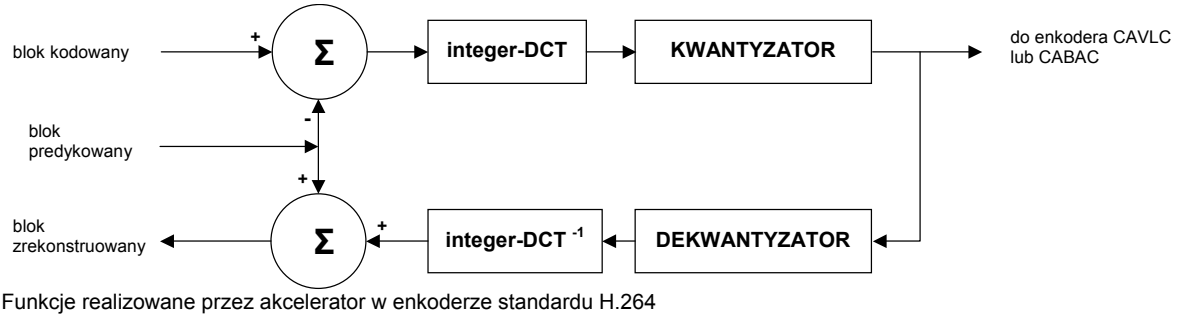
Abstract. *In the paper a customizable digital Discrete Cosine Transform accelerator for the H.264 video compression standard has been described. The accelerator also performs the inverse DCT, quantization and dequantization. The accelerator was initially implemented in the FPGA. It has been successfully verified, then implemented in an ASIC using the 90 nm UMC technology. Detailed test results of the accelerator ASIC were compared to other results available in the literature. Functionality of the accelerator has been described in detail in the paper. The testing system has been optimized for easy integration with the x.264 encoder software running under Linux OS and is devoted to accelerate HD video compression. Due to the low power consumption and a small area of the core described accelerator can be easily integrated with the video sensor. (**DCT transform accelerator for image compression in vision sensors**).*

Słowa kluczowe: kompresja wideo, H.264, DCT, kwantyzator, dekwantyzator, ASIC.
Keywords: video compression, H.264, DCT, quantizer, dequantizer, ASIC.

Wprowadzenie

Transformacja DCT (*Discrete Cosine Transform*) jest jednym z głównych algorytmów enkodera wideo standardu H.264. Ze względu na konieczność realizacji w enkoderze także bloku dekodera (dla prawidłowego działania predykcji) transformację tę wykonuje się dwukrotnie: najpierw w wersji prostej, potem w odwrotnej. Powoduje to, że moc obliczeniowa potrzebna do realizacji DCT stanowi znaczny procent mocy obliczeniowej całego enkodera. W związku z tym do implementacji prostej i odwrotnej DCT stosuje się często dedykowane sprzętowe moduły cyfrowe – akceleratory [1, 2].

W komunikacie przedstawiono konfigurowalny akcelerator transformacji DCT, odwrotnej transformacji DCT oraz kwantyzatora i dekwantyzatora, przeznaczony dla enkodera wideo standardu H.264 (transformacja prosta i odwrotna DCT została zaimplementowana w wersji „integer"). Akcelerator został wstępnie zaimplementowany w układzie FPGA VIRTEX6-VLX365T, a następnie w układzie ASIC w technologii UMC 90 nm. Do weryfikacji i testowania prototypu ASIC została wykorzystana zaawansowana platforma prototypowania [3]. Dzięki swojej konstrukcji platforma ta jest szczególnie przydatna do implementacji algorytmów kompresji obrazu wysokiej rozdzielczości oraz systemów widzenia maszynowego. Na platformie został zaimplementowany również system mikroprocesorowy oparty na mikroprocesorze Microblaze, pracujący pod kontrolą systemu operacyjnego Linux. Platforma została wyposażona w interfejsy pozwalające na podłączenie płytki z układem ASIC zawierającym moduł testowanego akceleratora. Do weryfikacji prototypu akceleratora został wykorzystany program enkodera x.264 [4]. Na rysunku 1 przedstawiono funkcje realizowane przez akcelerator w enkoderze standardu H.264.



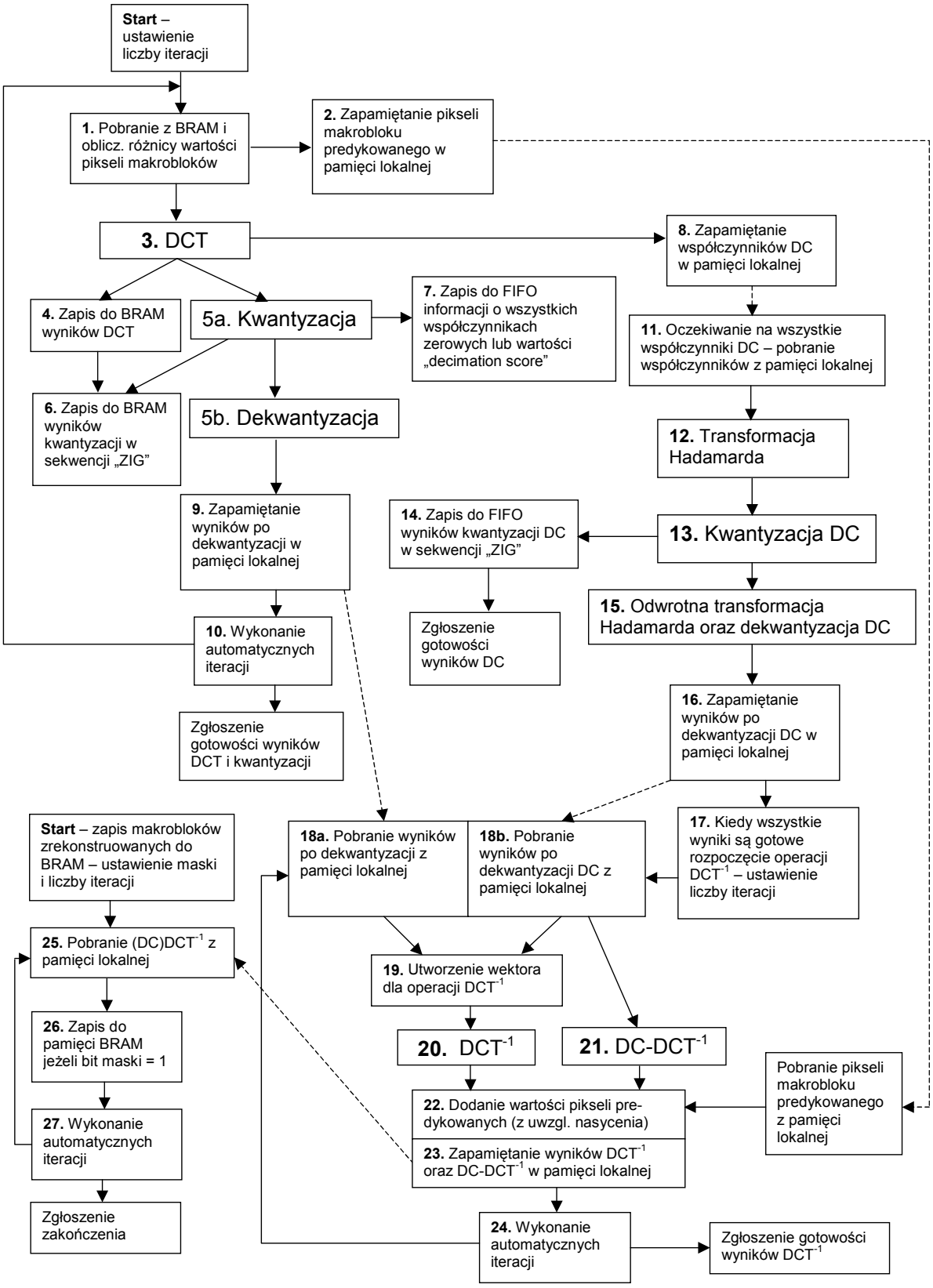
Rys. 1. Funkcje realizowane przez akcelerator w enkoderze standardu H.264

Budowa i działanie akceleratora

Uproszczony schemat działania akceleratora przedstawiono na rysunku 2. Poniżej omówiono szczegółowo poszczególne etapy:

1. Pobranie z zewnętrznej pamięci (BRAM) makrobloku kodowanego i makrobloku predykowanego. Równocześnie obliczana jest różnica pomiędzy wartościami pikseli tych makrobloków. Operacja ta realizowana jest równolegle na czterech pikselach w jednym cyklu zegara.
2. Zachowanie makrobloku predykowanego w pamięci lokalnej akceleratora w celu późniejszego wykorzystania podczas obliczania makrobloku zrekonstruowanego.
3. Obliczenie DCT dla jednego bloku 4x4 złożonego z obliczonych wcześniej różnic wartości pikseli. Operacja DCT wykonywana jest równolegle na całym bloku 4x4 pikseli (16 bajtów) w 4 fazach (czas trwania: 4 cykle zegara).
4. Rezultaty DCT są zapisywane do pamięci zewnętrznej BRAM (operacja ta może być wyłączona – nie zawsze jest potrzebna). Operacja zapisu trwa 8 cykli zegara (w jednym cyklu zegara zapisywane są 2 współczynniki). Równocześnie wyniki DCT są wprowadzane do modułu kwantyzacji.

5. Moduły kwantyzacji (5a) i dekwantyzacji (5b) uruchamiane są z chwilą otrzymania wyników DCT. Wyniki te są otrzymane równolegle dla wszystkich 16 współczynników bloku 4x4. Kwantyzator przetwarza współczynniki potokowo startując od ostatniego. Potok kwantyzatora ma dwa poziomy i po ich wypełnieniu uruchamiany jest potok dekwantyzatora (także dwupoziomowy). Kwantyzator i dekwantyzator obliczają jeden współczynnik w czasie jednego cyklu zegara. Kwantyzator pobiera automatycznie z pamięci współczynnik kwantyzacji, zależny od aktualnego parametru kompresji (i_qp) oraz indeksu współczynnika. BIAS dodawany do kwantyzowanej wartości jest również uwzględniany automatycznie (obsługiwane są dwa rodzaje BIASu: dla bloku interframe i dla bloku intraframe).
6. Rezultaty kwantyzacji są wyprowadzane w trakcie pracy kwantyzatora (po zmianie kolejności współczynników według reguły „ZIG") do zewnętrznej pamięci BRAM (po zakończeniu zapisu DCT do BRAM). Równocześnie realizowane jest obliczanie „decimation score" dla przetwarzanego bloku 4x4 – przydatne do dalszej optymalizacji kodowania makrobloku przez oprogramowanie. Na podstawie liczby punktów „decimation score" można podjąć decyzję o wyzerowaniu współczynników.
7. Do wewnętrznej kolejki FIFO (dostępnej z poziomu rejestrów) zapisana zostaje informacja o obliczonym „decimation score" lub informacja o wystąpieniu po kwantyzacji bloku 4x4 wszystkich współczynników zerowych. Informacje te są odczytywane w późniejszym etapie przez oprogramowanie.
8. Współczynniki o indeksie 0 (tzw. współczynniki DC) po transformacji DCT poszczególnych bloków 4x4 są zapamiętywane w pamięci lokalnej, w celu ewentualnego późniejszego uruchomienia dla nich specjalnej ścieżki przetwarzania. Ścieżka ta obejmuje transformację Hadamarda, kwantyzację i dekwantyzację w wersji DC oraz odwrotną transformację Hadamarda. Współczynniki DC po transformacji Hadamarda i kwantyzacji zostają umieszczone w kolejce FIFO, skąd mogą zostać pobrane przez oprogramowanie.
9. Wyniki dekwantyzacji są przekazywane do pamięci lokalnej w celu późniejszego przetworzenia przez blok odwrotnej transformacji DCT. Wyniki dekwantyzacji nie są dostępne na wyjściu akceleratora.
10. Proces obliczania DCT, kwantyzacji i dekwantyzacji może zostać automatycznie powtórzony dla pozostałych bloków 4x4, z których jest zbudowany makroblok 16x16.
11. Oczekiwanie na wszystkie współczynniki DC. Jeżeli zebrano 16 współczynników DC i ścieżka DC jest włączona w rejestrze konfiguracji – następuje przejście do etapu 12.
12. Transformacja Hadamarda wykonywana jest dla wszystkich współczynników DC jednocześnie (4 cykle zegara).
13. Kwantyzacja DC wykonywana jest na kolejnych wynikach poprzedniego etapu – jeden współczynnik w jednym cyklu zegara. Parametry: BIAS oraz „quant_mf" dla kwantyzacji DC znajdują się w rejestrach konfiguracyjnych. Równocześnie obliczana jest liczba niezerowych współczynników po kwantyzacji DC.
14. Wyniki kwantyzacji DC są zapisywane do pamięci FIFO w kolejności „ZIG". Zakończenie obliczeń kwantyzatora DC sygnalizowane jest w rejestrze i przerwaniem. Po pierwszym odczycie z FIFO zwracana jest wartość licznika niezerowych współczynników. Jeżeli występują niezerowe współczynniki, to można je otrzymać poprzez kolejne odczyty rejestru FIFO (po dwa współczynniki przy każdym odczycie).
15. Równocześnie z kwantyzatorem uruchamiana jest odwrotna transformacja Hadamarda oraz dekwantyzator DC.
16. Wyniki z dekwantyzatora DC są przechowywane w pamięci lokalnej. Zakończenie dekwantyzacji sygnalizuje gotowość danych dla procesu wykonującego odwrotną DCT. Dodatkowo jest możliwe odczytanie współczynników po dekwantyzacji DC poprzez odczyt z rejestru FIFO (po dwa współczynniki przy jednym odczycie).
17. Oczekiwanie na zebranie wszystkich współczynników po dekwantyzacji ze wszystkich zaprogramowanych bloków 4x4. Oczekuje się także na wyniki dekwantyzacji DC, jeżeli są potrzebne (są wstawiane do bloków 4x4 na pozycję o indeksie 0).
18. Pobranie wyników po dekwantyzacji (18a) i ewentualnie dekwantyzacji DC (18b) z pamięci lokalnej.
19. Utworzenie wektora dla transformacji odwrotnej DCT.
20. Wykonanie odwrotnej DCT dla wszystkich współczynników. Proces odwrotnej DCT uruchamia się zawsze i dla wszystkich bloków 4x4. Proces ten wykonuje się w czasie, kiedy oprogramowanie odczytuje z FIFO „decimation score" bloków 4x4 po kwantyzacji. Oprogramowanie decyduje o dalszym przetwarzaniu bloku 4x4. Kiedy oprogramowanie zakończy pracę, odwrotna DCT jest już gotowa i czeka w pamięci akceleratora na odczyt.
21. Równoczesne wykonanie odwrotnej DCT tylko dla współczynników DC.
22. Transformacja odwrotna jest obliczana i od razu dodawana jest do niej wartość piksela predykowanego z nasyceniem na wartościach granicznych piksela (0 i 255). Piksel predykowany został zapamiętany w trakcie pobierania danych na początku procesu DCT.
23. Odwrotna DCT jest obliczana i zapamiętana w pamięci lokalnej równocześnie w dwóch wersjach: ze wszystkich współczynników oraz wyłącznie ze współczynników DC (zwróconych przez ścieżkę przetwarzania dla współczynników DC).
24. Proces obliczania odwrotnej DCT może być automatycznie powtórzony dla pozostałych bloków 4x4, z których jest zbudowany makroblok 16x16. Kiedy odwrotna DCT jest gotowa, można rozpocząć procedurę jej zapisu do zewnętrznej pamięci bloków zrekonstruowanych. Gotowość wyników odwrotnej DCT można zweryfikować za pomocą rejestru statusu lub przerwania.
25. Operację zapisu do zewnętrznej pamięci bloków zrekonstruowanych (BRAM) zaczyna zapis do rejestru konfiguracyjnego, w tym momencie można wybrać czy potrzebne są wyniki odwrotnej DCT czy odwrotnej DC-DCT (odwrotna DCT z bloku złożonego wyłącznie ze współczynników DC).
26. Stan bitów rejestru maski odwrotnej DCT (wygenerowanej przez oprogramowanie) oznacza które bloki 4x4 mają być zapisane (16 bitów maski – po jeden bit na blok). Zapis pikseli do BRAM odbywa się po 4 piksele w jednym cyklu zegara (autoinkrementacja adresów dopasowana jest do oprogramowania x.264). Liczba zapisywanych bloków 4x4 jest programowalna, ale nie może przekraczać 16.
27. Operacja zapisu odwrotnej DCT może być automatycznie powtórzona dla pozostałych bloków 4x4, z których jest zbudowany makroblok 16x16. Moment zakończenia operacji zapisu wyników odwrotnej DCT do pamięci zewnętrznej można określić za pomocą odczytu rejestru statusu lub poprzez wykrycie przerwania.

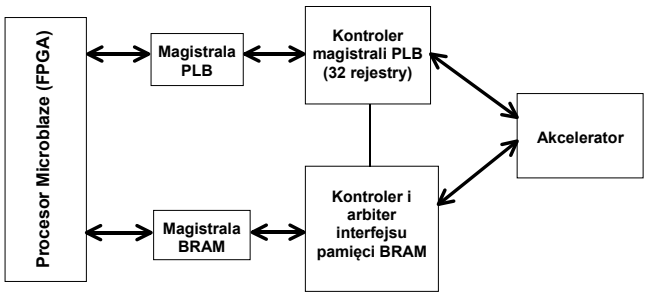


Rys. 2. Uproszczony schemat działania akceleratora

Implementacja i pomiary akceleratora

Do implementacji akceleratora wybrano proces CMOS 90nm firmy UMC w wersji L90N Mixed-Mode/RF – 1P9M2T1F wraz z bibliotekami firmy Faraday. Produkcja układu została przeprowadzona w ramach projektu Europractice przez organizację IMEC z Belgii. Synteza i implementacja układu została wykonana z wykorzystaniem oprogramowania Cadence. W syntezie wykorzystano funkcje redukcji poboru mocy poprzez bramkowanie sygnału zegara. Założona powierzchnia układu scalonego wynosi 1874,76 µm x 1874,76 µm (Mini@Sic IMEC). Układ ASIC zawiera kilka projektów cyfrowych, które ze względu na ograniczoną liczbę wyprowadzeń, muszą współdzielić jedną magistralę I/O. Opisywany akcelerator zajmuje około

70% rdzenia ASICa (pow. rdzenia: 1239125 μm^2, całkowita pow. układu: 3514725 μm^2). Kompresowane makrobloki wprowadzane są do akceleratora poprzez współdzieloną pamięć (BRAM). Konfiguracja akceleratora, jego uruchomienie oraz odczyt niektórych rezultatów odbywa się poprzez rejestry dostępne za pośrednictwem magistrali PLB. Podczas wprowadzania i wyprowadzania danych przez BRAM lub PLB częstotliwość zegara jest ograniczona do 100 MHz (ze względu na zegar procesora Microblaze), podczas obliczeń wewnętrznych częstotliwość zegara akceleratora może osiągać 400 MHz. Przy zastosowaniu całkowicie sprzętowego interfejsu akceleratora można by zlikwidować spowolnienie zegara, związane z wprowadzaniem danych, uruchomieniem i reakcją na przerwanie. Na rysunku 3 przedstawiono schemat blokowy systemu testowania akceleratora.



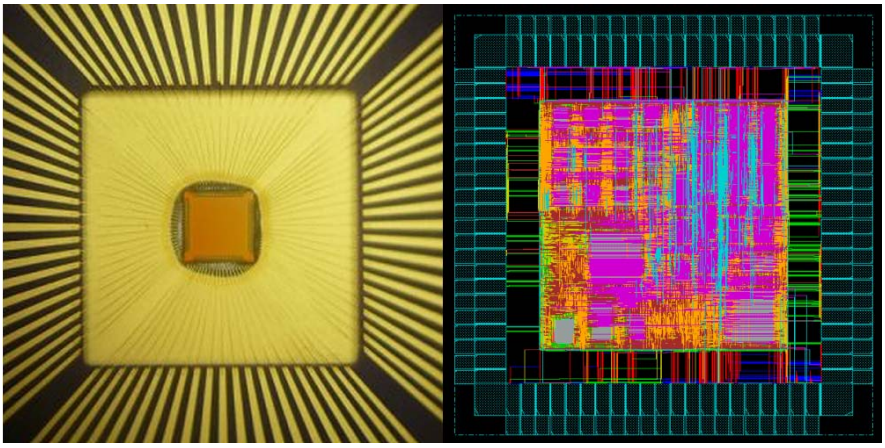
Rys. 3. Schemat blokowy systemu testowania akceleratora

Akcelerator ASIC został uruchomiony na płytce prototypowej współpracującej z wcześniej opracowanym systemem prototypowym opartym na układzie FPGA Virtex6. Wyniki pomiarów akceleratora zebrano w tabeli 1.

Tabela 1. Wyniki pomiarów akceleratora

Częstotliwość zegara [MHz]	Wydajność dla makrobloków 16x16 [makrobloki/s]	Wydajność dla ramek 1024x768 [ramki/s]	Pobór mocy rdzenia akceleratora (zasil.1V) [mW]
100	138000	45	15
200	276000	90	21
400	552000	180	30

W literaturze [1] przedstawiono przykład implementacji sprzętowej DCT i kwantyzacji (bez dekwantyzacji i odwrotnej DCT). Osiągnięto tam częstotliwość zegara 208 MHz, co dało 50 ramek na sekundę dla obrazu o wymiarach 1024x768 pikseli.
Na rysunku 4 przedstawiono zdjęcie wykonanego układu scalonego oraz wygląd topografii układu akceleratora z programu Encounter.



Rys. 4. Zdjęcie układu scalonego akceleratora oraz jego topografia z programu Encounter

Podsumowanie

Przedstawiona sprzętowa implementacja akceleratora DCT, odwrotnej DCT, kwantyzacji i dekwantyzacji za pomocą cyfrowego układu ASIC została zweryfikowana pozytywnie. Wydajność obliczeniowa oraz pobór mocy są porównywalne lub lepsze od innych opublikowanych propozycji implementacji. Dzięki łatwości integracji z oprogramowaniem x.264 opracowany ASIC może zostać wykorzystany do natychmiastowego wdrożenia, może także zostać dodany do układu sensora wizyjnego, gdzie uzyska jeszcze większą wydajność dzięki zredukowaniu ograniczeń komunikacyjnych.

Praca była częściowo finansowana z grantu NCBiR nr O R00 0046 09 oraz z grantu NCN nr DEC-2011/03/B/ST7/03547.

LITERATURA

[1] Keshaveni N., Ramachandran S., Gurumurthy K.S., Design and Implementation of Integer Transform and Quantization Processor for H.264 Encoder on FPGA, *International Conference on Advances in Computing, Control & Telecommunication Technologies,* 2009. ACT '09. 646-649.
[2] Kordasiewicz R.C., Shirani S., ASIC and FPGA implementations of H.264 DCT and quantization blocks, *IEEE International Conference on Image Processing,* ICIP 2005, Volume: 3; Page(s): III - 1020-3
[3] Kłosowski M., Wireless intelligent audio-video surveillance prototyping system, *Przegląd Elektrotechniczny*, nr 10 (2013), 97-99, 2013
[4] x.264 encoder, http://www.videolan.org/x264.html

***Autorzy**: dr inż. Miron Kłosowski, Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. Narutowicza 11/12, 80-233 Gdańsk, E-mail: mkl@ue.eti.pg.gda.pl; dr inż. Bogdan Pankiewicz, Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. Narutowicza 11/12, 80-233 Gdańsk; dr inż. Marek Wójcikowski, Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. Narutowicza 11/12, 80-233 Gdańsk, E-mail: wujek@ue.eti.pg.gda.pl.*