

Marek Kwiatkowski\*, Mariusz Kołton\*, Paweł Russek\*\*, Kazimierz Wiatr\*\*

## Softprocesor wizyjny z rekonfigurowalną listą instrukcji

### 1. Wprowadzenie

Pojęcie obróbki obrazu cyfrowego wiąże się z operacjami mającymi na celu zmianę jego właściwości. Na przykład są to następujące operacje: podkreślenie lub ukrycie niektórych szczegółów obrazu (tzw. filtracja), zmiana palety barw, zmiana rozdzielczości obrazu, zmniejszenie obszaru pamięci, jaki jest potrzebny do zapisania go na dysku itp. W praktyce operacje realizowane na danych obrazowych wymagają relatywnie dużych mocy obliczeniowych. Związane jest to z dużą ilością danych, które muszą być przetworzone. Przypadek taki w szczególności ma miejsce, kiedy dane wizyjne mają charakter sekwencyjny, tak jak dla sygnału pochodzącego z kamery. Dodatkowo sygnał wideo niejednokrotnie wymaga obróbki w reżimie czasu rzeczywistego, a więc czas przeznaczony na realizację określonych zadań jest ściśle limitowany. Dlatego przy wykonywaniu zadań związanych z obróbką obrazu, z konieczności sięga się do rozwiązań sprzętowych o charakterze dedykowanym [10]. Rozwiązania te dzięki specjalnej strukturze sprzętowej, która umożliwia czerpanie z technik przetwarzania równoległego i potokowego, cechuje duża wydajność obliczeniowa w odniesieniu do problemów, dla których były one projektowane.

Zaproponowane stanowisko do obróbki obrazu stanowi softprocesor wizyjny zrealizowany za pomocą rekonfigurowalnego układu FPGA. W odróżnieniu od klasycznych rozwiązań dedykowanych, których poważnym mankamentem jest mała uniwersalność polegająca na tym, że znajdują one zastosowanie jedynie przy realizacji jednego konkretnego algorytmu obliczeniowego [11, 4], zaproponowany softprocesor wizyjny jest architekturą specjalizowaną cechującą się jednak pewnymi właściwościami charakterystycznymi dla procesorów ogólnego zastosowania. Cechy te nadają softprocesorowi wizyjnemu właściwości rozwiązania uniwersalnego, w różnych obliczeniach przy zastosowaniu różnych algorytmów. Jest to możliwe, ponieważ zaproponowane rozwiązanie czerpie z idei obliczeń przy użyciu logiki rekonfigurowalnej [1]. Struktura takiej logiki może być łatwo zmieniana tak, aby w danej chwili dostosowana była do aktualnie realizowanego zadania obliczeniowego [3]. Ponieważ projektowanie struktury sprzętowej jest w ogólności zadaniem złożonym i czasochłonnym, softprocesor wizyjny zbudowany jest według określonego szablonu,

---

\* Katedra Elektroniki, Akademia Górniczo-Hutnicza w Krakowie

\*\* Katedra Elektroniki, ACK Cyfronet, Akademia Górniczo-Hutnicza w Krakowie

który ułatwia dostosowanie go do kolejnych, nowych zadań. Z drugiej strony narzucenie pewnego szkieletu architektury może się wydawać ograniczeniem pełnej swobody realizacji dedykowanego sprzętu. W praktyce okazuje się jednak, że algorytmy przetwarzania obrazów, do których został stworzony softprocesor, bardzo dobrze dają się efektywnie rozwiązywać za pomocą zaproponowanego szablonu hardwarowego.

## 2. Struktura procesora

Procesor ten został zaprojektowany tak, aby osiągnąć jak największą szybkość działania w algorytmach obróbki obrazu.

Ze względu na charakter tego rodzaju operacji, wyposażono procesor w zestaw instrukcji zawierający rozkazy, które są specjalizowane pod kątem operacji obrazowych (m.in. dodanie instrukcji, tzw. „motylka FFT”). Odmową zaletą procesora jest fakt, że wykonanie każdego rozkazu zajmuje taką samą liczbę taktów zegara, możemy więc mówić tutaj o architekturze typu RISC. W odróżnieniu od innych architektur mikroprocesorowych, nie tylko kolejne instrukcje są rozpoczynane w każdym taktie zegara (*throughput*), ale także czas potrzebny na wykonanie pojedynczego rozkazu (*latency*) to także jeden takt. W procesorach sygnałowych efekt zapisywania wyników kolejnych rozkazów co takt zegara systemowego został osiągnięty poprzez zastosowanie mechanizmu kolejowania zadań (*pipelining*). W softprocesorze wizyjnym efekt ten został osiągnięty dzięki zastosowaniu architektury harwardzkiej (kod programu i dane znajdują się w oddzielnych blokach pamięci). Ponadto każdy rozkaz może korzystać z czterech danych (dwóch liczb 16-bitowych oraz dwóch 8-bitowych współczynników), wyniki natomiast mogą zostać zapisane w postaci dwóch liczb 16-bitowych.

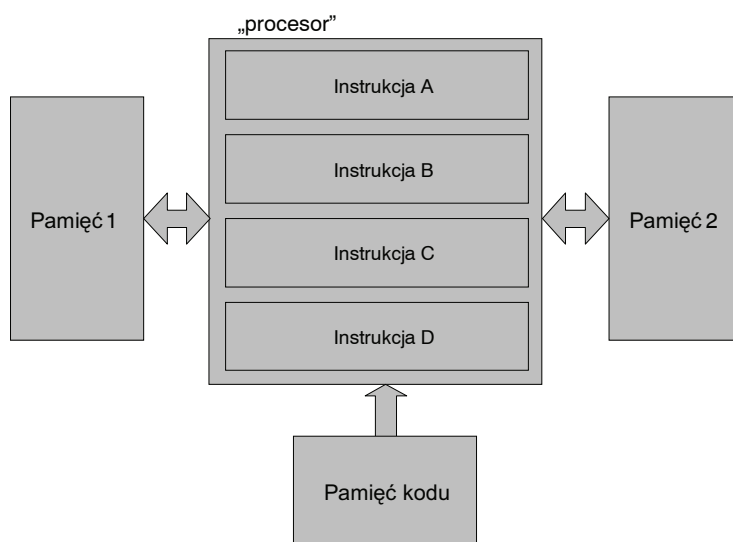
Aby uniknąć strat czasu na multipleksowanie magistral danych, poprzez które pobierane są argumenty rozkazu oraz zapisywane są wyniki, wykorzystano dwa bloki pamięci danych. W trakcie wykonywania zestawu instrukcji dane są pobierane z jednego bloku pamięci, a wyniki obliczeń zapisywane są w drugim bloku pamięci. Dzięki temu przełączanie magistral danych nie jest potrzebne w każdej instrukcji, co znacznie przyspiesza pracę procesora. Kolejnym pomysłem na skrócenie czasu obliczeń jest zrezygnowanie z jednostki odpowiedzialnej za obliczanie adresów pamięci (z powodu korzystania jednocześnie aż z czterech niezależnych argumentów układ taki musiałby być dość mocno rozbudowany). Adresy do pamięci danych są zatem podawane w kodzie maszynowym rozkazu i nie muszą być obliczane. Nad poprawnością adresów oraz wstawianiem operacji NOP podczas przełączania bloków pamięci i zmianie „kierunku” obliczeń czuwa napisany dla procesora kompilator.

Procesor posiada wbudowaną operację „motylka FFT”, która przebiega również w jednym taktie zegara, co pozwala mówić bardziej o sprzętowej, a nie o programowej obróbce obrazu.

Innym istotnym aspektem architektury softprocesora wizyjnego jest możliwość rekonfiguracji układu. Wprawdzie za pomocą istniejącej już listy instrukcji, oprócz 2D DCT, można przeprowadzać także inne operacje obrazowe (np. kwantyzację wektorową), to architektura procesora pozwala na dodanie innych rozkazów potrzebnych do wykonania in-

nych algorytmów. Musi jednak zostać zachowany warunek obliczenia wyniku takiej dodatkowej instrukcji w jednym takcie zegara.

Poglądowy schemat procesora wizyjnego przedstawia rysunek 1. Jego struktura składa się z procesora, pamięci kodu i dwóch pamięci danych (pamięć 1, pamięć 2). Procesor umożliwia realizację zaimplementowanych sprzętowo instrukcji (instrukcja A, instrukcja B, ...). Jego struktura umożliwia modyfikację dostępnych rozkazów, co pozwala użytkownikowi na definiowanie własnych operacji arytmetycznych. Zestaw tych instrukcji jest uzależniony od rodzaju zadania, które jest przez procesor realizowane w danej chwili. Sprzętowe wykonanie wybranych i charakterystycznych dla danego algorytmu instrukcji pozwala na ich bardzo efektywne i szybkie obliczanie (w praktyce jest to pojedynczy tak zegara). Zestaw instrukcji jest jedyną częścią architektury, która musi być zmodyfikowana przy zmianie realizowanego algorytmu. Dzięki temu implementacja nowego algorytmu jest w znacznym stopniu uproszczona. Sekwencję wykonywania instrukcji wyznacza kod programu zapisany w pamięci kodu.



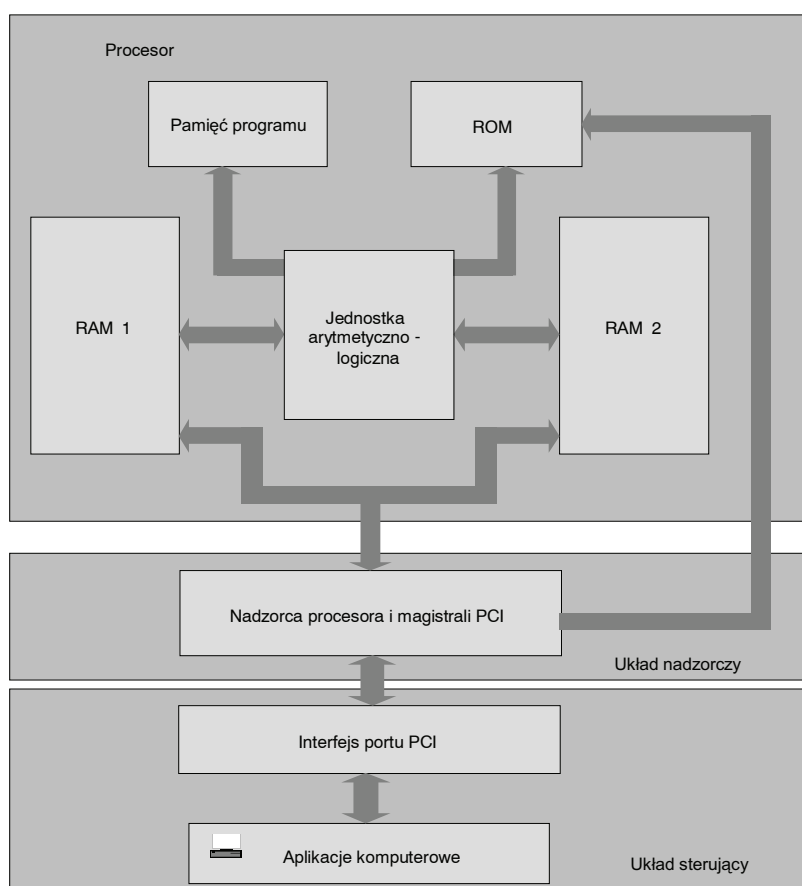
Rys. 1. Schemat blokowy procesora wizyjnego

Bardzo ważnym elementem zaproponowanego softprocesora jest to, że zawiera on dwa oddzielne banki pamięci danych. W trakcie pracy dane przepisywane są tam i z powrotem z jednej pamięci do drugiej, ulegając stopniowo operacjom obliczeniowym. Takie rozwiązanie pozwala na równoczesny zapis wyniku i odczyt nowych danych w tym samym takcie zegara. Podczas jednego taktu zegara dane są odczytywane z pamięci, następuje wykonanie rozkazu oraz zapis wyników do drugiej pamięci. Takie „przemiatanie” danych znacznie przyspiesza kilkakrotne wykonanie tego samego rozkazu na różnych zestawach danych, co jest szczególnie przydatne w operacjach na macierzach. Zastosowanie dwuportowych pamięci pozwala na definiowanie wieloargumentowych rozkazów. Takie rozwiązania są wyjątkowo przydatne w procesach obróbki obrazu.

Istotną własnością procesora jest możliwość rekonfiguracji. Ponieważ został on zaprojektowany przy użyciu języka opisu sprzętu VHDL [8], istnieje możliwość jego implementacji na innych platformach rekonfigurowalnych oraz zmiana jego listy instrukcji, tak by był on wykorzystany w innych zastosowaniach.

### 3. Realizacja procesora

Softprocesor wizyjny wraz z układem nadzorczym jest zaimplementowany na płycie Spyder firmy X2D [9]. Płyta ta posiada m.in. układ FPGA Virtex XCV300 [12], bloki pamięci ROM, własny generator sygnału zegarowego, a także interfejs portu PCI. To właśnie dzięki temu interfejsowi możliwa jest komunikacja płyty z komputerem PC. Na rysunku 2 przedstawiony jest schemat ideowy systemu obróbki obrazu wraz z elementami składowymi platformy Spyder



Rys. 2. Schemat ideowy softprocesora i systemu obróbki obrazu

Dzieli się on na trzy podstawowe części – procesor, układ nadzorczy i układ sterujący. Jednostka arytmetyczno-logiczna procesora (ALU) połączona jest z czterema pamięciami: dwiema pamięciami RAM, pamięcią ROM i pamięcią kodu. Pamięci RAM są integralną częścią układu Virtex, programowo zostały skonfigurowane jako dwuportowe pamięci o 16-bitowej magistrali danych. Dzięki tak skonfigurowanym pamięciom możliwy jest równoczesny odczyt bądź zapis dwóch komórek. Pamięci RAM i pamięć ROM mogą być inicjalizowane z komputera PC przed każdym uruchomieniem procesora, natomiast pamięć kodu wyłącznie podczas implementacji całego systemu. Tak więc zmiana kodu programu dla softprocesora wymaga zmiany składnika „pamięć programu” na poziomie języka VHDL. Głównym składnikiem układu nadzorczego jest *supervisor*. Składnik ten analizuje instrukcje użytkownika bezpośrednio odebrane z układu sterującego. Jego funkcją jest kontrola zadań wykonywanych przez cały system obróbki obrazu. W przeciwieństwie do procesora i układu nadzorczego układ sterujący nie składa się wyłącznie ze składników zaimplementowanych w układzie Virtex. Łączy on w sobie także aplikacje języka C++, narzędzia obsługi płyty Spyder oraz mostek PCI firmy PLX znajdujący się na płycie. Za pomocą narzędzi softwarowych można sprawdzić poprawność instalacji urządzenia, a także skonfigurować układ FPGA. Komunikacja z systemem odbywa się przez port PCI przez wpisanie do układu nadzorczego instrukcji użytkownika. Odrębna aplikacja służy do przygotowania pamięci kodu poprzez przetłumaczenie rozkazów asemblera na język VHDL.

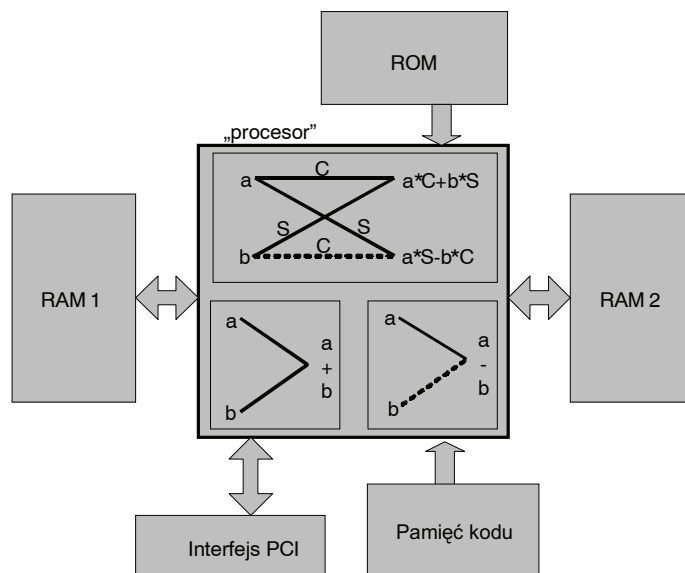
#### 4. Implementacja dyskretnej transformaty kosinusowej

W niniejszej pracy zastosowanie procesora przedstawiono na przykładzie obliczenia dyskretnej transformaty kosinusowej. Opracowano kod programu i zrealizowano sprzętowo dedykowane instrukcje procesora dla realizacji algorytmu DCT obrazu o rozmiarze  $8 \times 8$  pikseli za pomocą metody Chena [2]. Architektura procesora i jego instrukcje pozwalają na równoległe przeprowadzenie podstawowych operacji arytmetycznych takich jak mnożenie bądź dodawanie a także na wykonanie całej operacji „motylkowej” podczas jednego taktu zegara.

Ułamkowe współczynniki transformaty są wprowadzane do pamięci procesora jako 16-bitowe liczby 128 razy większe od rzeczywistych współczynników. Wynik mnożenia współczynników przez wartości pikseli jest automatycznie dzielony (bitowe przesunięcie w prawo) podczas tego samego taktu zegara, w którym została wykonana operacja mnożenia. Wykonanie całej dwuwymiarowej transformaty DCT obrazu o rozmiarach  $8 \times 8$  pikseli trwa przez 499 taktów zegara.

Ogólną konstrukcję softprocesora wraz z zaimplementowanymi rozkazami przedstawia rysunek 3.

Zastosowana tutaj architektura pozwala na szybkie wykonanie rozkazów mogących pobierać jednocześnie 4 argumenty (dwie 16-bitowe wartości pikseli przepływu danych w układzie: z pamięci A do pamięci B lub z pamięci B do pamięci A. Takie „przemiatanie” danych pozwala na przyspieszenie procesu obliczeń poprzez minimalizację opóźnień wynikających z przełączania pamięci w tryb zapisu/odczytu.



Rys. 3. Procesor wizyjny zaprogramowany do realizacji DCT

## 5. Dodatkowe narzędzia

Procesor wyposażony jest w dwa ułatwiające pracę projektanta narzędzia programowe. Obydwa: „spyder\_service.exe” i „makevhd.exe” są napisane w języku C++.

Aplikacja spyder\_service.exe komunikuje się z urządzeniem Spyder Virtex przez port PCI za pomocą pakietu funkcji dostarczonych przez producenta płyty. Dzięki temu użytkownik nie musi wnikać w struktury języka C++, pisać własnych aplikacji, aby sterować urządzeniem. Zasada działania programu opiera się na dwóch głównych krokach: kompilacji poleceń zawartych w pliku tekstowym i ich wykonaniu. Polecenia są składniowo podobne do rozkazów języka assembler, zawierają m.in. instrukcje zapisu i odczytu z pamięci softprocesora. W instrukcjach tych podane są nazwy plików, z których i do których będą odczytywane i zapisywane dane. Kompilator sprawdza zarówno składnię instrukcji, jak i zawartość podanych w nich plików. Zadaniem kompilatora jest także stwierdzenie, jaka ilość danych znajduje się w sprawdzanym pliku. Jeśli kompilacja zakończy się bezbłędnie następuje wykonanie instrukcji. Wszystkie wyniki i komunikaty są zapisywane do pliku.

## 6. Wyniki implementacji

Procesor wizyjny został zrealizowany za pomocą aplikacji Active-HDL 6.2 oraz Riviera 2005.08 firmy Aldec. Synteza i implementacja projektu została przeprowadzona przy użyciu narzędzia ISE 6.2 firmy Xilinx. Wykorzystanie zasobów układu FPGA Virtex XCV300BG432 po zaimplementowaniu projektu przedstawia tabela 1.

**Tabela 1**  
Ilość wykorzystanych zasobów sprzętowych podczas implementacji softprocesora dla DCT

Rodzaj urządzenia	Zasoby wykorzystane	Zasoby dostępne	Procent wykorzystania
Flip Flops	744	6144	12
4 input LUTs	3786	6144	61
Slices	2147	3072	69
Block RAMs	2	16	12

Wykorzystano 69% dostępnych komórek logicznych *slice*, przy czym są one częścią składową prawie wszystkich komórek logicznych CLB. W skład CLB wchodzi 4 komórki *slice* i prawie w każdym CLB przynajmniej jedna *slice* jest użyta. Stanowi to poważne ograniczenie dla rozbudowy projektu. Próby wykorzystania większej liczby komórek ze względu na konieczność ich połączenia kończy się przekroczeniem możliwości układu pod względem wielkości zasobów i dopuszczalnych opóźnień czasowych.

Parametry czasowe projektu są przedstawione w tabeli 2. Interfejs portu PCI oraz układ Virtex są taktowane zegarem systemowym o częstotliwości 33MHz. Jednakże ze względu na ograniczenia układu i specyfikę projektu softprocesor pracuje z częstotliwością czterokrotnie niższą.

**Tabela 2**  
Parametry czasowe zaimplementowanego softprocesora dla DCT

Wymaganie czasowe	Żądane	Otrzymane
Okres zegara	20 ns	18 ns
Czas wyprzedzenia wejścia do zegara ( $t_{\text{setup}}$ )	15 ns	8 ns
Czas ustalania wyjścia ( $t_{\text{datavalid}}$ )	15 ns	14 ns

## 7. Możliwości rozbudowy

Softprocesor wraz z układem nadzorczym został zaimplementowany w układzie FPGA Virtex V300 BG432. Wielkość tego układu mocno ogranicza możliwości modernizacji procesora. W projekcie wykorzystuje się jedynie dwa spośród szesnastu bloków pamięci RAM. Zastosowanie układu o większej pojemności, a w szczególności o większej liczbie dostępnych ścieżek logicznych, pozwoliłoby na zwiększenie funkcjonalności procesora poprzez umieszczenie kodu w pamięci RAM. Umożliwiłoby to programowanie procesora po jego implementacji. Kod 64-bitowy można zapisać za pomocą portu PCI w dwóch dwuportowych pamięciach RAM. Jednakże w takim przypadku niezbędne byłyby dodatkowe połączenia oraz multiplexery pełniące rolę arbitrów magistral adresowych pamięci RAM.

Dodatkowe pamięci pozwoliłyby także na zwiększenie rozdzielczości obliczeń z 16 na 32 bity. Takie rozwiązanie wymaga jednak dodatkowych połączeń pamięci z układem nadzorczym i jednostką arytmetyczno-logiczną, dodatkowych multiplexerów, zmiany funkcji

arytmetycznych, a także zaimplementowania mechanizmu rozdzielającego wynik operacji arytmetycznej na poszczególne porty pamięci RAM. Oczywiście zapis 32-dwubitowej danej do pamięci RAM przez port PCI wymagałby co najmniej dwóch taktów zegara systemowego.

Softprocesor wykonuje wszystkie operacje arytmetyczne podczas jednego taktu zegara. Procesor jest jednak taktowany zegarem o częstotliwości czterokrotnie mniejszej od zegara systemowego. Zwiększenie tej częstotliwości mogłoby być wprowadzone poprzez zastosowanie obliczeń potokowych. Przy obecnej realizacji procesora jest to jednak zadanie skomplikowane. Procesor bowiem wykonuje szereg różnych operacji arytmetycznych zależnych od kodu maszynowego. Nieefektywne byłoby wprowadzenie tej samej liczby taktów zegara na operację dodawania i na o wiele bardziej skomplikowaną „operację motylkową”. Natomiast różna liczba taktów zegara dla różnych operacji spowodowałaby przestój potoku (wstrzymanie pobierania kodu maszynowego). Taka sytuacja nastąpiłaby w przypadku, gdy bieżąca operacja arytmetyczna wymagałaby większej liczby taktów zegara niż następna instrukcja. Potokowość operacji arytmetycznych może mieć zastosowanie w przypadku, gdy procesor wykonywałby dużą liczbę takich samych operacji z rzędu. Oczywiście w takim przypadku wymagany byłby kilkuktaktowy przestój potoku podczas zmiany kierunku przesyłu danych.

## 8. Podsumowanie

Z założenia softprocesor wizyjny nie jest procesorem ogólnego przeznaczenia. Został on zaprojektowany z zamysłem efektywnej realizacji algorytmów obróbki obrazu. Pomimo stosunkowo niskiej częstotliwości zegara, z jakim pracuje softprocesor, wszystkie rozkazy wykonywane są w jednym takcie zegara. Ta właściwość oraz fakt, że lista instrukcji zawiera rozkazy pracujące jednocześnie na czterech argumentach wejściowych i zawierające w sobie wyniki czterech operacji mnożenia stanowią o szybkim wykonywaniu algorytmu. Efektywność działania i tak wysoka specjalizacja instrukcji były możliwe dzięki zastosowaniu dwóch oddzielnych, dwuportowych pamięci zawierających dane oraz pamięci współczynników. Pozwoliło to na jednoczesne pobieranie argumentów rozkazu i zapisywanie wyników poprzednio wykonanej instrukcji. Efektem jest obliczenie transformaty DCT macierzy  $8 \times 8$  zawierającej 16-bitowe dane w 499 taktach zegara. Dla porównania, przy zastosowaniu procesorów ogólnego zastosowania liczba wymaganych taktów przeznaczonych na to samo zadanie jest kilkaset razy większa (tab. 3).

**Tabela 3**  
Przybliżona liczba wykonanych taktów zegara poszczególnych procesorów podczas realizacji dwuwymiarowej transformaty kosinusowej

Procesor	Częstotliwość procesora [MHz]	Przybliżona liczba taktów zegara
Athlon 2600 +XP	2070	68000
Intel P4 3200	3200	102000
Intel Celeron 2,6 GHz	2600	130000



Lista rozkazów potrzebna do wykonania tego zadania stwarza możliwości zastosowania procesora także do innych operacji obrazowych. Dzięki rekonfigurowalności układu FPGA możliwa staje się ingerencja w zestaw instrukcji procesora. Przy spełnieniu założenia wykonania rozkazu w jednym taktie zegara użytkownik jest w stanie zdefiniować własną instrukcję. Oprócz DCT i IDCT softprocesor może być wykorzystywany w innych operacjach związanych z obróbką obrazu, na przykład: mnożeniu macierzy, konwersji kolorów, kwantyzacji wektorowej.

### Literatura

- [1] Barr M.: *A Reconfigurable Computing Primer*. Multimedia Systems Design, 1998
- [2] Chen W.H., Smith C.H., Fralick S.: *A fast computational algorithm for the discrete cosine transform*. IEEE Trans. Commun., vol. COM-25, Sept. 1977, 1004–1009
- [3] Barat F. et al.: *Low Power Coarse-Grained Reconfigurable Instruction Set Processor*. Proceedings of 13-th International Conference on Field Programmable Logic and Applications Conference, Portugal, 2003
- [4] Dąbrowska A., Wiatr K.: *Realizacja Algorytmu Chena i Loeffera transformacji FDCT w układach FPGA*. Automatyka Półrocznik AGH, Kraków, t. 8, z. 3, 2005
- [5] Kiryukhin G.; Celenk M.: *Implementation of 2D-DCT on XC4000 series FPGA using DFT-based DSFG and DA architectures*. Proceedings 2001 International Conference on vol. 3, 7–10, Oct. 2001, 302–305
- [6] Metzger P.: *Anatomia PC*. Helion, Gliwice 2001
- [7] Meyer-Baese U.: *Digital Signal Processing with Field Programmable Gate Arrays*. Springer, Berlin, 2001
- [8] Skahill K.: *Język VHDL*. Warszawa, WNT 2001
- [9] Spyder: [www.x2e.de](http://www.x2e.de).
- [10] Wiatr K.: *Sprzętowe implementacje algorytmów przetwarzania obrazów w systemach wizyjnych czasu rzeczywistego*. Kraków, UWND AGH 2002
- [11] Wiatr K.: *Akseleracja obliczeń w systemach wizyjnych*. Warszawa, WNT 2003
- [12] Xilinx, [www.xilinx.com](http://www.xilinx.com)

