

Grzegorz PASTUSZAK

POLITECHNIKA WARSZAWSKA, INSTYTUT RADIOELEKTRONIKI

## Architektura transformacji i kwantyzacji w wysoko-przepustowym koderze H.264/AVC opartym na zaawansowanym wyborze trybu kodowania

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 na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej. Od 2001 do 2002 roku był projektantem układów ASIC w firmach FFC (Tokio) i Fujitsu Devices (Yokohama). Jego obszar zainteresowania obejmuje: architektury i algorytmy VLSI, przetwarzanie obrazów/video/audio oraz kompresję, wydajne cyfrowe układy scalone.

e-mail: G.Pastuszak@ire.pw.edu.pl



### Streszczenie

Standard kompresji wideo H.264/AVC umożliwia uzyskanie wysokiej efektywności kompresji kosztem dużej złożoności obliczeniowej. Aby uzyskać możliwie wysoką efektywność architektura kodera opisana w tym artykule realizuje wybór trybu kodowania na bazie łącznej optymalizacji zniekształceń i stopnia kompresji. W szczególności założony przepływ danych pozwala na uzyskanie przepustowości 32 próbek/współczynników na takt zegara, co umożliwia sprawdzenie dużej ilości trybów kodowania przed wyborem najbardziej optymalnego. Architektura zawiera bloki transformacji, kwantyzacji, dekwantyzacji, odwrotnej transformacji, rekonstrukcji. Architektura obsługuje wszystkie rodzaje transformacji i formaty chrominancji wyspecyfikowane w profilu Wysokim przy użyciu tych samych zasobów sprzętowych. Przepustowość uzyskiwana w układach FPGA pozwala na wybór trybu na bazie analizy zniekształceń i stopnia kompresji dla wymagań HDTV.

**Słowa kluczowe:** FPGA, architektury sprzętowe, H.264/AVC, kompresja wideo.

### Transforms and Quantization in the High-Throughput H.264/AVC Encoder Based on Advanced Mode Selection

#### Abstract

The H.264/AVC standard allows for a high compression efficiency at the cost of computational complexity. To achieve the efficiency as high as possible, the architecture proposed in the paper supports the mode selection based on the rate-distortion optimization. In particular, the dataflow assumes throughput of 32 samples/coefficient per clock cycle, on average, allowing a lot of compression options to be checked. Moreover, the architecture supports all transform sizes specified for High Profile using the same hardware resources. Synthesis results show that the design can work at 100 MHz for FPGA Stratix II devices.

**Keywords:** FPGA, hardware architecture, H.264/AVC, video compression.

### 1. Wprowadzenie

Standard H.264/AVC [1] umożliwia znaczny wzrost efektywności kompresji w stosunku do poprzedników. Aby wykorzystać tę przewagę, koder powinien znaleźć optymalny zestaw opcji kompresji dla każdego makrobloku a nawet dla jego partycji (subbloków i bloków). W rzeczywistości sprawdzenie wszystkich kombinacji opcji jest niepraktyczne ze względu na bardzo dużą ilość obliczeń. Z tego względu kodery implementują uproszczone wersje procedury optymalizacji. Ponieważ uproszczenie oznacza niższą efektywność kompresji, głównym zadaniem projektowym jest zminimalizowanie strat przy utrzymaniu rozsądnej złożoności obliczeniowej.

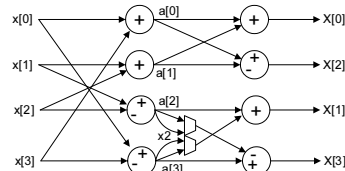
Kodery sprzętowe zwykle zakładają, że minimalizacja błędów predykcji, wyrażonego jako Suma Modułów Różnic (SAD) lub Suma Modułów Różnic Transformacji (SADT), zapewnia najlepszą drogę do maksymalizacji efektywności. To założenie jest

zwykle prawdziwe kiedy różnica w SAD odpowiadająca dwóm opcjom jest znacząca, ale może zawieść kiedy ta różnica jest mała. Najlepsza miara optymalności bierze pod uwagę długość strumienia i zniekształcenia (błąd rekonstrukcji) odpowiadające danym opcjom. Oszacowanie tych wielkości oraz ich ważenie przy użyciu optymalizacji Lagrange'a zapewnia najlepszą selekcję. Ponieważ długości strumienia oraz zniekształcenia obliczone są w oparciu o odpowiednio skwantowane indeksy i zrekonstruowane próbki, koder powinien zawierać moduły transformacji i (de)kwantyzacji o dużej przepustowości, aby sprawdzić jak największą liczbę kombinacji. Dodatkowo większa przepustowość skraca czas potrzebny do uzyskania predykcji wewnątrzramkowej (Intra) dla kolejnych makrobloków, subbloków (8x8) lub bloków (4x4).

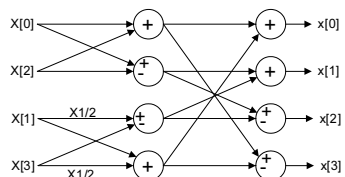
Szereg architektur sprzętowych opracowano tylko dla transformaty 4x4 [2-6] lub 8x8 [7]. Rekonfigurowalna architektura opisana w [8] może obsługiwać tylko transformaty odwrotne. W [9] zaproponowana architektura jest w stanie zmienić połączenia pomiędzy jednostkami arytmetycznymi, aby obsługiwać wszystkie transformacje proste i odwrotne. Ponieważ wszystkie te architektury mają małe przepustowości, nie mogą one być użyte w zaawansowanym wyborze opcji i parametrów kompresji. Artykuł ten opisuje architekturę mogącą przetwarzać 32 próbki/współczynniki na cykl zegara. W szczególności architektura integruje w ścieżce przetwarzania następujące moduły: prostej transformacji, kwantyzacji, dekwantyzacji, odwrotnej transformacji i rekonstrukcji próbek. Moduły prostej i odwrotnej transformacji zaprojektowane są tak aby implementować wszystkie rozmiary transformaty zdefiniowane w standardzie przy wykorzystaniu tych samych zasobów sprzętowych z minimalną ilością dodatkowych elementów.

### 2. Architektura rekonfigurowanej transformacji

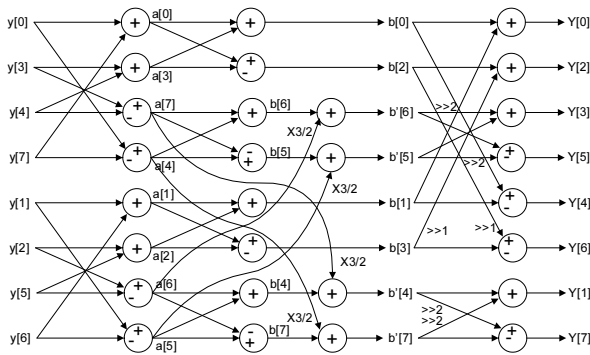
Najlepszym sposobem na zaimplementowanie transformacji jest użycie jej zdekomponowanej struktury. Taka zdekomponowana struktura przedstawiona jest na rys. 1 i 2 dla bloków 4x4 oraz na rys. 3 i 4 dla bloków 8x8. Prosta transformacja 4x4 (rys. 1) obsługuje zarówno przybliżone DCT jak również transformację Hadamard.



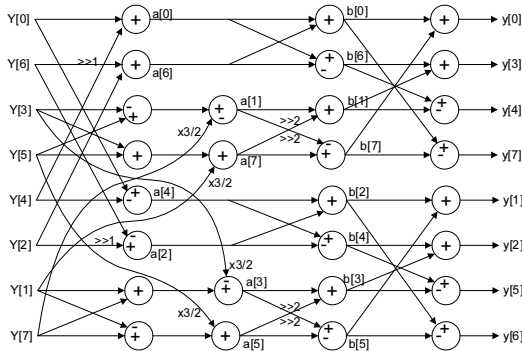
Rys. 1. Diagram transformacji prostej dla bloków 4x4  
Fig. 1. Diagram of the forward transform for 4x4 blocks



Rys. 2. Diagram transformacji odwrotnej dla bloków 4x4  
Fig. 2. Diagram of the inverse transform for 4x4 blocks



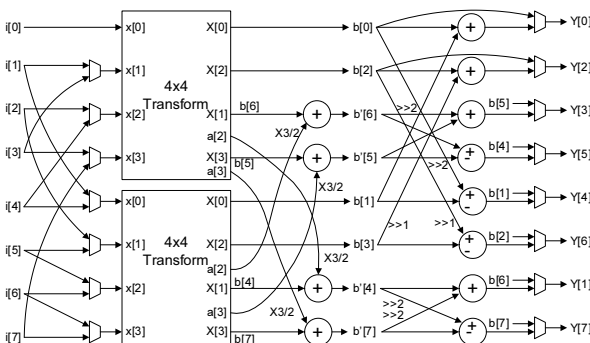
Rys. 3. Diagram transformacji prostej dla bloków 8x8  
Fig. 3. Diagram of the forward transform for 8x8 blocks



Rys. 4. Diagram transformacji odwrotnej dla bloków 8x8  
Fig. 4. Diagram of the inverse transform for 8x8 blocks

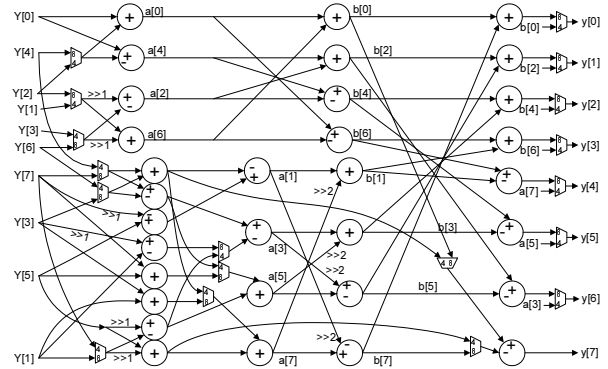
W szczególności dodatkowe multiplexery umożliwiają rekonfigurację sieci połączeń. Transformacje dla bloków 8x8 są bardziej złożone. Składają się one z czterech stopni przetwarzania w odróżnieniu od dwóch użytych w przypadku kiedy transformowane są bloki 4x4.

Gdy architektura nie obsługuje różnych typów transformacji, koder może użyć oddzielnych modułów, z których każdy obsługuje jeden typ transformacji. Do utrzymania regularności ścieżki danych dla prostej lub odwrotnej transformacji, powinno się użyć dwóch modułów dla transformaty cztero-punktowej (bloki 4x4) oraz jeden dla transformaty ośmio-punktowej (bloki 8x8). Wybór pomiędzy tymi dwoma typami transformacji jest dokonywany przy pomocy multiplexerów umieszczonych na końcu ścieżki przetwarzania danych. Taki projekt jest nieefektywny z punktu widzenia zasobów sprzętowych ponieważ tylko jedna gałąź wykorzystywana jest w danej chwili czasowej. Z tego względu efektywne rozwiązanie powinno cechować się użyciem tych samych zasobów z możliwie małym nadmiarem.



Rys. 5. Diagram transformacji prostej dla bloków 4x4 i 8x8 z współdzieleniem zasobów  
Fig. 5. Diagram of the forward transform for 4x4 and 8x8 blocks with resource sharing

W przypadku transformacji prostej można zauważyć, że graf sygnałowy dwóch pierwszych stopni transformaty ośmio-punktowej przypomina ten dla transformaty cztero-punktowej. Z tego względu transformacja cztero-punktowa może być mapowana na ośmio-punktową, co jest przedstawione na rys. 5. Jednakże porządek wejść musi być zmieniony ze względu na ich różne przypisanie dla obu typów transformacji. W konsekwencji siedem nowych multiplexerów dodanych jest do układu. Poza tym osiem multiplexerów wyjściowych jest koniecznych tak ja w przypadku oddzielnej implementacji obu transformacji do wyboru odpowiedniego wyniku.

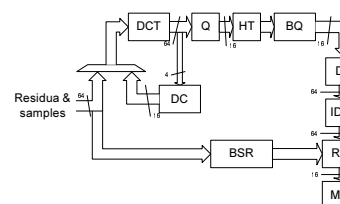


Rys. 6. Diagram transformacji odwrotnej dla bloków 4x4 i 8x8 z współdzieleniem zasobów  
Fig. 6. Diagram of the inverse transform for 4x4 and 8x8 blocks with resource sharing

W przeciwieństwie do transformacji prostej, odwrotna nie ma wspólnej części w grafie. W szczególności połączenia pomiędzy elementami arytmetycznymi w dwóch ostatnich stopniach zawierają stałe przesunięcia bitowe w dół, a rozmieszczenie sumatorów i subtraktorów jest zamienione. Z tych względów mapowanie transformacji cztero-punktowej na dwa ostatnie stopnie transformacji ośmio-punktowej wymagałoby znacznej liczby multiplexerów do rekonfiguracji połączeń. Okazuje się, że lepszym rozwiązaniem jest mapowanie na stopnie początkowe, co przedstawiono na rys. 6. By wypuklić strukturę transformacji cztero-punktowej, ośmio-punktowa jest przeorganizowana. Po pierwsze, porządek węzłów wejściowych, wyjściowych i wewnętrznych jest zmodyfikowany. Po drugie, porządek operacji dodawania i odejmowania jest zmieniony (łącznie z dodawaniem realizującym mnożenie przez 3/2). Po trzecie, wykorzystany jest jeden subtraktor w ostatnim stopniu przy wyjściu 7. Połączenia pomiędzy elementami arytmetycznymi są rekonfigurowane jeżeli typ transformacji jest zmieniony. W konsekwencji użytych jest 10 dodatkowych multiplexerów w porównaniu do przypadku oddzielnej implementacji obu typów transformacji.

### 3. Potok

Proponowana architektura zakłada 16-bitową dokładność operacji arytmetycznych i wczytuje subbloki 8x8 rezyduów. Subbloki wejściowe pobierane są co drugi cykl zegara. Stąd przepustowość wynosi 32 próbki/współczynniki na cykl zegarowy. Wszystkie moduły zorganizowane są w potok, co pokazuje rys. 7.



Rys. 7. Przepływ danych pomiędzy poszczególnymi modułami  
Fig. 7. Dataflow between all modules

Oprócz próbek/współczynników w potoku przenoszone są pewne dane kontrolne takie jak indeks subbloku, parametr kwantyzacji, typ transformaty i znacznik chromy.

Moduł transformacji przeprowadza jedno-wymiarową operację na całym subbloku w jednym cyklu zegara. Następnie wynik jest transponowany i poddawany tej samej operacji transformacji w następnym cyklu zegarowym, aby otrzymać transformatę 2D. Przetwarzanie równoległe wymaga użycia ośmiu podjednostek transformacji (opisanych w poprzedniej sekcji), z których każda operuje na jednym wierszu/kolumnie. Przy czym ten sam koncept stosowany jest zarówno do transformaty prostej (DCT) i odwrotnej (IDCT). Dla trybu chromy lub lummy 16x16 współczynniki DC zapisywane są w rejestrach (DC) i są one podawane na wejście transformaty prostej po przetworzeniu wszystkich zwykłych bloków 4x4. Ponieważ transformata Hadamard stosowana jest do bloków DC przed dekwantyzacją, specjalny moduł (HT) użyty jest do tej operacji pomiędzy kwantyzatorem i buforem pamięci. Moduł ten działa na podwójnej częstotliwości. Podobnie jak w przypadku innych transformat dwa cykle zegarowe potrzebne są do przeprowadzenia operacji 2D. Bloki inne niż DC są tylko przepuszczane bez modyfikacji.

Kwantyzacja (Q) i dekwantyzacja (DQ) prowadzone są równoległe na 32 współczynnikach w każdym cyklu zegara. W ten sposób, dwa cykle zegara potrzebne są do przetworzenia jednego subbloku. W każdym cyklu zegarowym, multiplexer w module kwantyzacji wybiera połowę współczynników z subbloku i dokonuje kwantyzacji. Kwantowane współczynniki zapisywane są w buforze pamięci przy podwójnej częstotliwości zegara, dlatego jeden blok 4x4 zapisywany jest w każdym cyklu zegara. Takie bloki czytane są również przy podwojonym zegarze i zapisywane do rejestru 8x4 taktowanego zegarem podstawowym. Dla trybu chromy lub lummy 16x16 odwrotnie transformowany blok DC czytany jest w pierwszej z bufora pamięci, a jego skwantowane współczynniki są kolejno dystrybuowane do zwykłych bloków w następujących cyklach. Po przeczytaniu skwantowane bloki wystawiane są na wejście modułu odpowiedzialnego za estymację długości strumienia binarnego. Po dekwantyzacji współczynniki zapisywane są do rejestrów w kolejnych cyklach zegara, aby odtworzyć subblok. Moduły kwantyzacji i dekwantyzacji zawierają po 32 identyczne i pracujące równoległe podjednostki. Każda podjednostka zawiera układ mnożący z następującym po nim sumatorem (zaokrąglenie) i przesuwniku bitowym. Multiplikatory i wartości przesunięć otrzymywane są przy użyciu logiki kombinacyjnej (tablice LUT) na bazie parametru kwantyzacji  $Q_p$  i typu transformaty.

#### 4. Wyniki implementacji

Zaprojektowane architektury opisane zostały przy pomocy języka RTL VHDL i zweryfikowane funkcjonalnie w symulacji względem modelu referencyjnego JM (Joint Model) organizacji JVT (Joint Video Team), wersja 11. Z punktu widzenia przepustowości prezentowana architektura znacznie przewyższa inne opisane w literaturze. Całość układu może wspierać wybór trybu oparty o analizę RD dla rozdzielczości HDTV. Przykładowo zegar 100 MHz umożliwia alokację średnio 408 cykli na jeden makroblok dla sekwencji wideo 1080x1920 30 Hz. Oznacza to, że można przeanalizować 51 trybów bloków 16x16 dla każdego makrobloku. Przy czym rozmiar ramki i częstotliwość odświeżania mogą być wymieniane na liczbę sprawdzonych trybów kompresji.

Architektura jest zsyntezowana dla technologii FPGA i ASIC, a wyniki podano w tabeli 1. Jak można zauważyć duża przepustowość zwiększa wymagania na ilość zasobów sprzętowych. W ogólności ograniczenia na częstotliwość zegara pochodzą układów kwantyzacji, który zawiera mnożenie. Tabela 2 zawiera wyniki syntezy dla różnych konfiguracji modułów transformacji. Opcje konfiguracji pozwalają na wybór pomiędzy typami transformacji 4x4, 8x8 lub oboma. Wyniki syntezy dowodzą, że łączna implementacja prowadzi do oszczędności zasobów. Układ używa

pamięci dwu-portowych dla skwantowanych współczynników (BQ) oraz danych oryginalnych (BSR).

Tab. 1. Wyniki syntezy dla różnych technologii  
Tab. 1. Synthesis results for different technologies

Moduł	Stratix II [ALUT]	AMS 0.35 [bramki]	TOWER 0.18 [bramki]
DCT	5143	35347	55728
DC	260	1810	1797
Q	2180+32DSP	77668	104051
HT	949	4916	4761
BQ	831	6481	6702
DQ	2049+32DSP	55385	76112
IDCT	5370	35232	59395
REC+BSR	1495	12420	11539
All	18277+64DSP	229259	32085
Zegar [MHz]	100	79	76

Tab. 2. Wyniki syntezy dla różnych konfiguracji transformacji  
Tab. 2. Synthesis results for different configurations of transformations

Moduł & Konfiguracja	Stratix II (90 nm)		AMS 0.35µm	
	koszt [ALUT]	zegar [MHz]	koszt [bramki]	zegar [MHz]
DCT 4x4	2830	108	20209	98
DCT 8x8	4718	109	30030	83
DCT 4x4, 8x8	5143	100	35347	82
IDCT 4x4	2144	108	18494	99
IDCT 8x8	3744	109	28900	83
IDCT 4x4, 8x8	5258	100	35232	83

Proponowana architektura przewyższa inne z punktu widzenia przepustowości. Chociaż koszt zasobów jest duży, stosunek przepustowości do obszaru bramek jest podobny lub lepszy w porównaniu do innych projektów. Należy zauważyć, że projekty te, oprócz [9], nie wspierają transformaty 8x8 Profilu Wysokiego. Co więcej, nie obejmują one także ścieżki do obliczania transformaty hierarchicznej dla trybów chromy i lummy 16x16.

Prezentowana praca wykonana została w ramach projektu badawczego 3 T11D 014 30 finansowanego przez Ministra Nauki i Szkolnictwa Wyższego.

#### 5. Literatura

- [1] ITU-T Recommendation H.264 and ISO/IEC 14496-10 MPEG-4 Part 10, Advanced Video Coding (AVC), 2003.
- [2] H.-Y. Lin, Y.-C. Chao, C.-H. Chen, B.-D. Liu, and J.-F. Yang, "Combined 2-D transform and quantization architectures for H.264 video coders," ISCAS'2005, pp. 1802 - 1805 vol. 2, Kobe, Japan, 23-26 May, 2005.
- [3] R.C. Kordasiewicz and S. Shirani, "ASIC and FPGA implementations of H.264 DCT and quantization blocks," ICIP'2005, pp. III - 1020-3, vol. 3, Genova, Italy, 11-14 September 2005.
- [4] K.-H. Chen, J.-I. Guo, and J.-S. Wang, "A high-performance direct 2-D transform coding IP design for MPEG-4AVC/H.264," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 4, pp. 472 - 483, April 2006.
- [5] S. Lee and K. Cho, "Implementation of an AMBA-Compliant IP for H.264 Transform and Quantization," APCCAS'2006, pp. 1071 - 1074, 4-7 Dec. 2006.
- [6] Y.-F. Lin, X.-Y. Zeng, J. Chen, Z. Zhang, "A Low Cost design for Optimized IDCT and de-quantization of H.264," ICSICT'2006, pp. 1893 - 1895, Shanghai, China, 23-26 October 2006.
- [7] I. Amer, W. Badawy, and G. Jullien, "A high-performance hardware implementation of the H.264 simplified 8/spl times/8 transformation and quantization," ICASSP'2005, vol. 2, pp. ii/1137 - ii/1140, Philadelphia, PA, 18-23 March 2005.
- [8] M. Stepniwska and A. Luczak, "Reconfigurable architecture of AVC/H.264 Integer Transform," European Signal Processing Conference, EUSIPCO2006, Florence, Italy, 4-8 September 2006.
- [9] R. R. Osorio and J. D. Bruguera, "A Unified Architecture for H.264 Multiple Block-Size DCT with Fast and Low Cost Quantization," 9th IEEE EUROMICRO Conference on Digital System Design (DSD'2006), pp. 407-414, Dubrovnik, Croatia, August 2006.