

Agnieszka DĄBROWSKA^{1,2}, Kazimierz WIATR^{1,2}

¹ AKADEMIA GÓRNICZO-HUTNICZA

² ACK CYFRONET AGH

Implementacja elementów bezstratnej kompresji standardu MPEG-2 w układach FPGA

Mgr inż. Agnieszka DĄBROWSKA

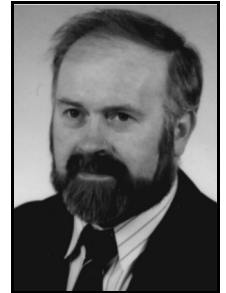
Ukończyła studia na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Jest asystentem w Katedrze Elektroniki AGH. Jej zainteresowania naukowe to kompresja obrazu, systemy czasu rzeczywistego, układy programowalne, układy rekonfigurowane.



e-mail: adabrow@agh.edu.pl

Prof. dr hab. inż. Kazimierz WIATR

Studia AGH Kraków (1980), dr nauk technicznych (1987), dr habilitowany (1999) i profesor (2002). Profesor zwyczajny na Akademii Górniczo-Hutniczej oraz Dyrektor Akademickiego Centrum Komputerowego Cyfronet AGH. Prowadzone prace badawcze dotyczą komputerowego sterowania procesami, systemów wizyjnych, systemów wieloprocesorowych, układów programowalnych, rekonfigurowalnych systemów obliczeniowych i sprzętowych metod akceleracji obliczeń.



e-mail: wiatr@agh.edu.pl

Streszczenie

Jednym z głównych elementów kompresji obrazów ruchomych oraz nieruchomych jest kodowanie entropijne. Typowy koder składa się z bloku uporządkowania pikseli, bloku kodowania ciągów oraz bloku kodera ze zmienną długością słowa. W artykule zostały przedstawione parametry zaimplementowanych metod kompresji bezstratnej: kodera/dekodera entropijnego oraz części składowych.

Słowa kluczowe: Zig-Zag, RLC, VLC, FPGA.

Implementation of lossless compression elements of MPEG-2 standard in FPGA

Abstract

An entropy coding is one of basic elements of video and still image compression. Typical entropy coder consists of pixels ordering block, run length coding block and Variable Length Coding block. Paper presents parameters lossless compression methods implementation: encoder/decoder and components part.

Keywords: Zig-Zag, RLC, VLC, FPGA.

1. Wstęp

W standardzie MPEG-2 zostały określone algorytmy, jakie mogą zostać użyte w procesie kompresji oraz dekompresji sygnału wizyjnego. Algorytmy te można podzielić na dwie grupy: algorytmy stratne oraz bezstratne. W skład pierwszej grupy wchodzi dyskretna transformacja kosinusowa, estymacja oraz kompensacja ruchu. Do drugiej grupy zalicza się kodowanie ciągów oraz kodowanie ze zmienną długością słowa.

2. Kodowanie bezstratne w MPEG-2

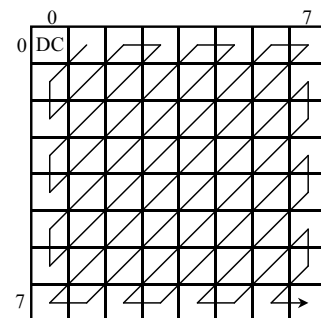
W standardzie MPEG-2 pierwszym etapem części bezstratnego kodowania jest blok uporządkowania współczynników otrzymanych w wyniku dyskretnego transformacji kosinusowej DCT. Jedną z podstawowych metod uporządkowania jest metoda Zig-Zag przedstawiona na rysunku 1. Obecna w standardzie JPEG, podobnie jak w MPEG-1, metoda Zig-Zag przestała być metodą optymalną, zatem w standardzie MPEG-2 zdefiniowano tak zwaną alternatywną metodę uporządkowania (rys. 2), chociaż jest stosowana również metoda Zig-Zag. W wyniku uporządkowania współczynników otrzymuje się ciąg 64 elementów. W ciągu tym występują współczynniki o wartościach zerowych pomiędzy współczynnikami niezerowymi.

Drugim etapem części bezstratnego kodowania w MPEG-2 jest kodowanie ciągów *RLC* (ang. *Run Length Coding*). W ogólnym przypadku *RLC* wyznacza parę liczb. Pierwsza z nich określa ile razy pojawił się dany symbol. Druga z liczb określa, jaki to symbol.

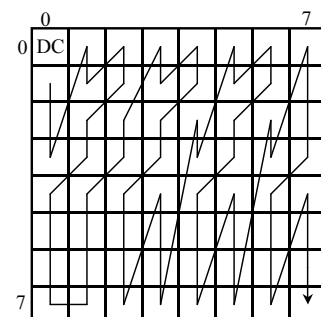
Przykładowo:

AAAAAAAAAAAAAAAABBBBBB $\Rightarrow$ (15,A), (5,B)

W MPEG-2 kodowanie *RLC* polega na wyznaczeniu pary liczb (*run*, *level*).



Rys. 1. Działanie metody Zig-Zag dla bloku o rozmiarze 8x8 pikseli
Fig. 1. A working of Zig-Zag method for 8x8 pixels block size



Rys. 2. Alternatywna metoda uporządkowania współczynników transformacji DCT w standardzie MPEG-2
Fig. 2. An alternative method of DCT transformation coefficients ordering in MPEG-2 standard

Pierwsza z pary określa ile współczynników o wartościach równych zero poprzedza współczynnik niezerowy, którego amplitudę określa parametr *level*.

W standardzie ISO/IEC 13818 zostały ściśle określone tablice trzy tablice – B14, B15 oraz B16, w których zawarte są słowa kodowe dla par *run-level* otrzymanych w wyniku kodowania *RLC*.

Trzecim elementem bezstratnego kodowania w MPEG-2 jest kodowanie ze zmienną długością słowa *VLC*, oparte na kodowaniu Huffmana.

W standardzie ISO/IEC 13818 zostały ściśle określone trzy tablice – B14, B15 oraz B16, w których zawarte są słowa kodowe dla par *run-level* otrzymanych w wyniku kodowania *RLC*. To, z której tabeli będzie korzystał koder/dekoder *VLC* (ang. *Variable Length Coding*) będzie zależało od kombinacji dwóch sygnałów sterujących: *intra_vlc_format* oraz *macroblock_intra*. Zależności pomiędzy tymi dwoma sygnałami a wyborem odpowiedniej tablicy zawarte są w tabeli 1.

Jednak nie wszystkie możliwe kombinacje są określone przy w tablicach B-14 i B-15. Jeśli dana para (*run*, *level*) nie występuje w tablicy zdeterminowanej przez *intra_vlc_format* i *macroblock_intra* to sekwencja wyjściowa *VLC* składa się z 6-bitowej „ucieczki”, 6-bitowego kodu reprezentującego *run* oraz 12-bitowego kodu określającego *level*. Kody dla nieokreślonych par (*run*, *level*) zawiera tablica B-16.

Tab. 1. Zależności między sygnałami sterującymi a wyborem tablicy VLC
Tab. 1 A dependency among control signals and VLC table selection

	intra_vlc_format	
	0	1
Bloki intra <i>macroblock_intra</i> = 1	B-14	B-15
Bloki inter <i>macroblock_intra</i> = 0	B-14	B-14

Tab. 2. Tablica B-16 – część dotycząca *run*
Tab. 2 A table B-16 – run's part

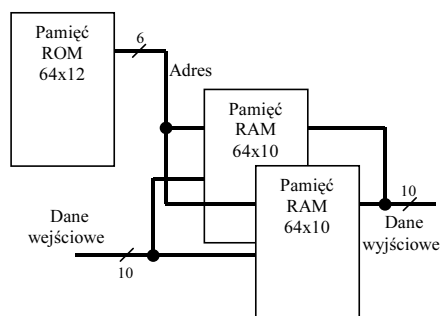
Kod	run
0000 00	0
0000 01	1
0000 10	2
...	...
1111 10	62
1111 11	63

Tab. 3. Tablica B-16 – część dotycząca *level*
Tab. 3 A table B-16 – level's part

Kod	level
1000 0000 0001	-2047
1000 0000 0010	-2046
...	...
1111 1111 1111	-1
0000 0000 0000	zabronione
0000 0000 0001	+1
...	...
0111 1111 1110	+2046
0111 1111 1111	+2047

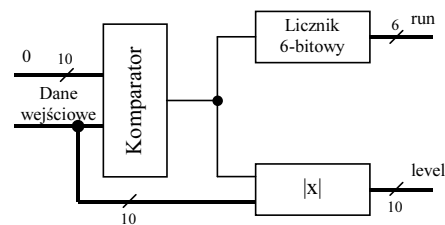
3. Implementacja

Pierwszym zaimplementowanym elementem kompresji bezstratnej standardu MPEG-2 jest układ porządkujący współczynniki otrzymane z dyskretniej transformacji kosinusowej. Schemat blokowy układu przedstawia rysunek 3. Układ składa się z trzech bloków pamięci. Pierwszy z bloków pamięci – ROM – o wielkości 64x12 bitów zawiera dwie tablice, z których jedna odpowiada za uporządkowanie zgodne z metodą Zig-Zag, a druga tablica jest zgodna z metodą alternatywnego uporządkowania. Tak zaimplementowana pamięć ROM generuje adresy dla cykli odczytu dwóch bloków pamięci RAM 64x10 bitów. Pamięci te działają naprzemiennie. Jeżeli w danym momencie jeden z bloków jest w stanie odczytu to do drugiego z bloków zapisywane są sekwencyjnie współczynniki otrzymane z dwuwymiarowej dyskretniej transformacji kosinusowej. Takie działanie umożliwia ciągły odczyt danych, co byłoby niemożliwe w przypadku zastosowania pojedynczego bloku pamięci RAM. Przy takim rozwiązaniu odczyt byłby przerywany na 64 takty zegara w celu zapisu nowych 64 współczynników, co oznaczałoby dwukrotnie mniejszą wydajność przetwarzania. Dane są wpisywane do pamięci pod kolejne adresy, natomiast kolejność odczytu jest uwarunkowana sekwencją adresów generowanych przez pamięć ROM. Implementacja takiego bloku porządkowania współczynników 2D-DCT w układzie XC2VP100(-6)FF1704 firmy Xilinx wymaga 23 bloków SLICE (z 44096 możliwych). Na taką zajętość składa się 26 przerzutników Flip-Flop (z 88192) oraz 40 bloków LUT (z 88192). Ponadto wykorzystane zostaje 3 bloki BRAM. Zaimplementowany układ może współpracować z maksymalną częstotliwością wynoszącą 413,394MHz.



Rys. 3. Schemat bloku uporządkowania współczynników 2D-DCT
Fig. 3. A block scheme of 2D-DCT coefficients ordering

Drugim zaimplementowanym blokiem toru kompresji bezstratnej standardu MPEG-2 jest koder *RLC*. Schemat blokowy przedstawia rysunek 4.



Rys. 4. Schemat blokowy kodera *RLC*
Fig. 4. A block scheme of RLC encoder

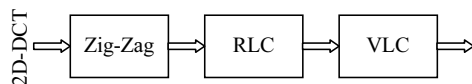
Układ kodera *RLC* składa się z trzech bloków. Pierwszy z nich odpowiada za wykrywanie czy dana wejściowa ma wartość zerową. Jeżeli na wejściu pojawi się współczynnik o zerowej wartości zostaje uruchomiony licznik liczący w zakresie od 0 do 63. Każdy pojawiający się kolejno współczynnik zerowy powoduje zwiększenie

szenie stanu licznika o 1. Jeśli jednak komparator wykryje współczynnik niezerowy to uruchamiany jest trzeci blok. Wyznacza on moduł wartości przychodzącego współczynnika. Jako najstarszy bit danej wyjściowej przepisywany jest najstarszy bit danej wejściowej. Nowe dane wyjściowe w zaimplementowanym koderze pojawiają się dopiero po wykryciu współczynnika o niezerowej wartości. Implementacja kodera *RLC* w układzie XC2VP100 zajmuje 14 bloków SLICE, na co składa się 12 przerzutników typu Flip-Flop oraz 24 bloki LUT. Koder może pracować z maksymalną częstotliwością 437,828MHz.

Kolejnym zaimplementowanym blokiem toru kompresji bezstratnej jest koder o zmiennej długości słowa *VLC*. Układ koder wykorzystującego określone przez specyfikację ISO/IEC 13818 tablice zajmuje 596 bloków SLICE w tym 28 przerzutników Flip-Flop oraz 1076 bloków LUT. Koder zaimplementowany w układzie XC2VP100(-6)FF1704 firmy Xilinx może pracować z częstotliwością 193,949MHz.

Schemat końcowy zaimplementowanego koder entropijnego standardu MPEG-2 przedstawia rysunek 5. Daną wejściową koder jest pojedynczy współczynnik dwuwymiarowej dyskretnej transformacji kosinusowej. Daną wyjściową stanowi kod o maksymalnej wielkości 24 bitów dla pary wyznaczonych liczb (*run*, *level*).

Zaimplementowany podobnie jak bloki składowe, w układzie XC2VP100(-6)1704 zajmuje 644 bloków SLICE, co stanowi 1% zasobów układu. Na takie wykorzystanie składa się 162 przerzutniki Flip-Flop oraz 1235 bloki LUT (1% wszystkich możliwych). Dodatkowo wykorzystane jest 3 bloki BRAM. Dane na wyjściu koder mogą pojawiać się co takt zegara jeśli nie zostaną wykryte współczynniki o wartości zerowej. Koder może pracować z maksymalną częstotliwością równą 139,005MHz.



Rys. 5. Uproszczony schemat blokowy koder entropijnego standardu MPEG-2
Fig. 5. A simplified block scheme of entropy encoder in MPEG-2 standard

Tab. 4. Parametry implementacji w układzie XC2VP100(-6)1704 firmy Xilinx
Tab. 4. Parameters of implementation in Xilinx XC2VP100(-6)1704

	Zig-Zag	RLC	VLC	Koder entropijny
SLICE	23	14	596	644
FF	26	12	29	162
LUT	40	24	1076	1235
BRAM	3	-	-	3
f(MHz)	413,394	437,828	193,949	139,005

Oprócz toru koder entropijnego, został zaimplementowany de-koder. Implementacja wykorzystuje 119 bloków SLICE, na co składa się 135 przerzutników Flip-Flop oraz 206 bloków LUT. Dodatkowo w układzie XC2VP100(-6)1704 wykorzystywane są 4 bloki pamięci BRAM. Implementacja pojedynczego bloku de-kodera ciągów *RLD* wymaga 35 bloków SLICE. Na taką liczbę bloków składa się 56 przerzutników Flip-Flop oraz 30 bloków LUT. Częstotliwość pracy dekodera ciągów wyniosła 233,236. W przypadku implementacji dekodera o zmiennej długości słowa kodowego wymagane jest 96 bloków SLICE (62 przerzutniki Flip-Flop oraz 169 bloki LUT). Ponad to wykorzystane zostały 323 bufory trójstanowe oraz jeden blok pamięci BRAM. Sam

zaimplementowany dekoderek *VLD* może pracować z częstotliwością 297,442MHz.

Tab. 5. Parametry implementacji w układzie XC2VP100(-6)1704 firmy Xilinx
Tab. 5. Parameters of implementation in Xilinx XC2VP100(-6)1704

	RLD	VLD	Dekoder entropijny
SLICE	35	96	119
FF	56	62	135
LUT	30	169	206
BRAM	-	-	4
f(MHz)	233,236	297,442	233,236

4. Podsumowanie

Zaimplementowany koder entropijny jest w stanie przetwarzać 62 ramki na sekundę obrazu o rozdzielczości 1920x1152. Rozdzielczość ta jest charakterystyczna dla profilu wysokiego standardu MPEG-2. Taka wydajność oznacza, że koder spełnia wymogi pracy w czasie rzeczywistym. Parametry dla innych rozdzielczości prezentuje tabela 6. Zaimplementowany dekoderek entropijny również spełnia wymogi pracy w czasie rzeczywistym. Przy rozdzielczości profilu wysokiego dekoderek może przetworzyć 107 ramek sekwencji obrazów ruchomych w ciągu sekundy.

Tab. 6. Wydajność zaimplementowanego koder oraz dekodera entropijnego
Tab. 6. A performance of implemented entropy encoder and decoder

Rozdzielczość obrazu	Liczba ramek na sekundę	
	koder	dekoder
800x600	289	492
1024x768	176	300
1000x1000	139	236
1920x1152	62	107
2000x2000	34	59

Praca naukowa finansowana ze środków budżetowych na naukę w latach 2005-2007 jako projekt badawczy.

5. Literatura

- [1] Bhaskaran V., Konstantinides K. – Image and Video compression standards. Algorithms Architectures.
- [2] Haskell B.G., Puri A., Netravali A.N. – Digital video: An introduction to MPEG-2.
- [3] ISO/IEC 13818-2: 1995
- [4] Netravali A.N., Haskell B.G. – Digital pictures. Representation, Compression and Standards.
- [5] Rabbani M., Jones P.W. – Digital Image Compression Techniques.
- [6] Rudberg M.K., Wanhammar L. – Implementation of a fast MPEG-2 compliant Huffman decoder.
- [7] Sudharsan S., Sinnathamby M.: Support for VariableLength Decode on Embedded Processors
- [8] Xilinx: xapp616, www.xilinx.com
- [9] Xilinx: xapp621, www.xilinx.com