

Sławomir Cichoń\*

## **Implementacja w układzie reprogramowalnym operacji kwantyzacji odwrotnej na potrzeby wewnątrzramkowej dekompresji wideo**

### **1. Wprowadzenie**

Kwantyzacja danych wideo występuje w różnych odmianach, niemal w każdym algorytmie kompresji obrazów, wśród których należy wymienić przede wszystkim MPEG, JPEG, DV, DVCPRO HD. Ma ona na celu redukcję mniej istotnych danych, jednocześnie zwiększając stopień kompresji danych, w następującym po kwantyzacji procesie kodowania o zmiennej długości słowa. W niniejszej publikacji opisany zostanie szczegółowo proces skalowania wstępnego i kwantyzacji w wybranych standardach kompresji wewnątrzramkowej, ze szczególnym uwzględnieniem standardu [3]. Przedstawiona zostanie również architektura potokowa realizująca operację kwantyzacji odwrotnej, która może być wykorzystana jako jednostka przetwarzająca sprzętowego dekodera realizowanego przez układ reprogramowalny. Względem ogólnego schematu kwantyzacji odwrotnej dokonano dwóch modyfikacji, mających na celu zmniejszenie czasu przetwarzania pojedynczego bloku danych. Sposób kwantyzacji danych stosowany w kompresji DV o standardowej rozdzielczości opisany jest częściowo w pracach [1, 7]. W pracy [6] zasygnalizowano sprzętową implementację operacji kwantyzacji odwrotnej dla cyfrowego wideo. Przykłady sprzętowej architektury dla procesu kwantyzacji i kwantyzacji odwrotnej dla kompresji MPEG prezentują prace [5, 8].

### **2. Kwantyzacja w standardzie cyfrowego wideo (IEC 61834-2, IEC 61834-3, SMPTE 370M)**

Kwantyzacja jest operacją, w której ważne współczynniki, uzyskane przez transformatę DCT, są redukowane do reprezentacji 9-bitowej w jednym etapie bądź dwóch

---

\* Doktorant, Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

etapach. Dla ustalenia uwagi przedstawione rysunki opisują sposób kwantyzacji zgodny z IEC 61834-2, zwany DV SD (*Digital Video Standard Definition*). Etapy te to:

- skalowanie wstępne (nieobecne w SMPTE 370M),
- kwantyzacja.

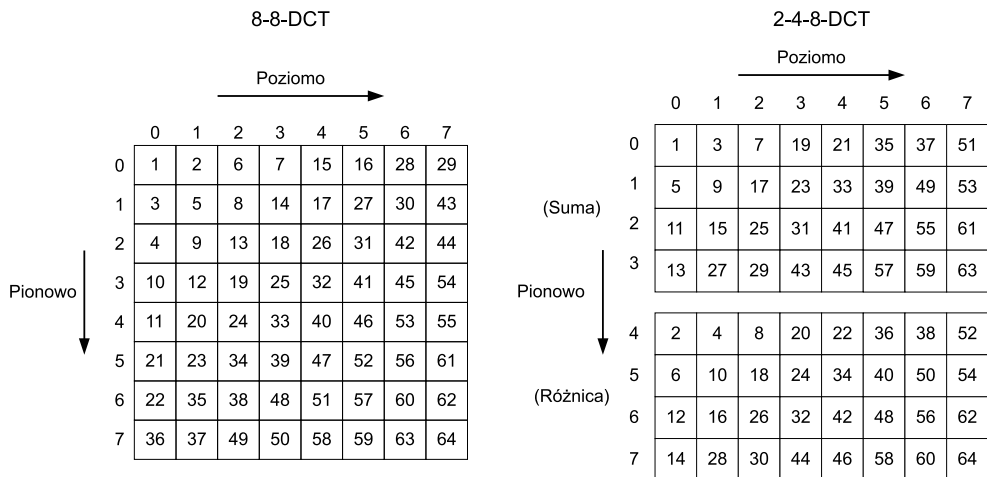
Oba te etapy są zależne od numeru współczynnika przypisanego podczas tzw. skanowania ziggag (rys. 1) oraz od parametrów: numeru kwantowania i klasy kwantyzacji. Wyjściem transformacji DCT (*Discrete Cosine Transform*) są współczynnik DC, oraz 63 współczynniki AC w następującej reprezentacji bitowej:

DC (9 bitów): b8 b7 b6 b5 b4 b3 b2 b1 b0

w kodzie U2 (od  $-255$  do  $+255$ )

AC (10 bitów): s b8 b7 b6 b5 b4 b3 b2 b1 b0

1 bit znaku + 9 bitów wartości bezwzględnej (od  $-511$  do  $+511$ ).



Rys. 1. Kolejność wyjścia współczynników DCT – tzw. ziggag

Taki sposób ustawienia powoduje, że współczynniki o niskich indeksach ziggag niosą zazwyczaj więcej informacji o obrazie, natomiast współczynniki o wysokich indeksach ( $> 40$ ) zazwyczaj mają wartości zero lub bliskie zero. Jest to również zgodne z własnością transformat częstotliwościowych, do jakich należy DCT. W standardzie SMPTE 370M, nie ma trybu 2-4-8-DCT, stąd ma zastosowanie tylko kolejność 8-8-DCT.

Każdy blok DCT jest zaklasyfikowany w zależności od wartości bezwzględnej największego współczynnika AC w bloku, oraz typu bloku DCT (Y – luminancja, Cr – chrominancja czerwona, Cb – chrominancja niebieska), do jednej z czterech klas.

Wartości zawarte w tabeli 1 zaczerpnięto z dokumentu normatywnego [3]. Dla standardu IEC 61834-3 [4], tabela ta wygląda podobnie, różne są wartości progowe przynależności do danej klasy kwantyzacji. Przedziały wartości bezwzględnych współczynników nie są jednak jednoznacznie zdefiniowane przez standard. Algorytm kodujący może wykazywać w pewnym stopniu dowolność, gdyż dla procesu odwrotnego nie przedziały są istotne, a wartości klas kwantyzacji.

**Tabela 1**  
Przykład przydzielania wartości klasy kwantyzacji

	Maksymalna wartość bezwzględna współczynników AC			
	od 0 do 11	od 12 do 23	od 24 do 35	> 35
Y	0	1	2	3
Cr	1	2	3	3
Cb	2	3	3	3

Klasyfikacja współczynników DCT do jednej ze stref, w zależności od ich indeksu (rys. 1), pokazana jest na rysunku 2. Klasyfikacja ta jest zdefiniowana dla każdego z bloków DCT.

8-8-DCT		2-4-8-DCT	
Poziomo →		Poziomo →	
	0 1 2 3 4 5 6 7		0 1 2 3 4 5 6 7
0	DC 0 0 1 1 1 2 2	0	DC 0 1 1 1 2 2 3
1	0 0 1 1 1 2 2 2	1	0 1 1 2 2 2 3 3
2	0 1 1 1 2 2 2 3	2	1 1 2 2 2 3 3 3
3	1 1 1 2 2 2 3 3	3	1 2 2 2 3 3 3 3
4	1 1 2 2 2 3 3 3	4	0 0 1 1 2 2 2 3
5	1 2 2 2 3 3 3 3	5	0 1 1 2 2 2 3 3
6	2 2 2 3 3 3 3 3	6	1 1 2 2 2 3 3 3
7	2 2 3 3 3 3 3 3	7	1 2 2 3 3 3 3 3

(Suma)

Pionowo ↓

(Różnica)

**Rys. 2.** Schemat przypisania współczynnikom DCT stref kwantyzacji

Strefa „0” zawiera współczynniki AC najbliższe współczynnikowi DC pod względem częstotliwości, niosące najistotniejszą część informacji o obrazie. Podobnie jak miało to miejsce przy indeksacji współczynników AC, przy skanowaniu ziggag, im wyższy

przypisany numer strefy kwantyzacji, tym znaczenie należących do niej współczynników AC, na całość zdekompresowanej ramki, jest mniejsze. Każdej ze stref przyporządkowana zostaje liczba, odpowiadająca przesunięciu bitowemu w prawo wartości współczynnika AC:

$$SHIFT(i, j, k) = F(QNO(i), qclass(j), k) \quad (1)$$

gdzie:

- $i$  – indeks makrobloku w segmencie wizji (0...4),
- $j$  – indeks bloku DCT w obrębie makrobloku (0...5),
- $k$  – numer strefy kwantyzacji (0...3),
- $QNO(i)$  – liczba kwantyzacji makrobloku,
- $qclass(j)$  – klasa kwantyzacji bloku DCT.

Norma [3] nie opisuje sposobu określania wartości  $QNO(i)$ . Wartości  $SHIFT(i, j, k)$  przedstawia tabela 2. Analogiczna tabela dla standardu DV HD [4] zawiera inne wartości. W przypadku SMPTE 370M, ze względu na brak podziału na strefy, tabela jest prostsza, jednak wartości przez które należy podzielić współczynniki, nie są wielokrotnościami liczby 2, a co za tym idzie, wymagają implementacji operacji mnożenia w czasie kwantyzacji odwrotnej.

**Tabela 2**  
Przyporządkowanie przesunięcia bitowego dla kwantyzacji

	<i>qclass(j)</i>				Numer strefy <i>k</i>			
	0	1	2	3	0	1	2	3
Liczba przedziałów kwantyzacji $QNO(i)$	15				1	1	1	1
	14				1	1	1	1
	13				1	1	1	1
	12	15			1	1	1	1
	11	14			1	1	1	1
	10	13		15	1	1	1	1
	9	12	15	14	1	1	1	1
	8	11	14	13	1	1	1	2
	7	10	13	12	1	1	2	2
	6	9	12	11	1	1	2	2
	5	8	11	10	1	2	2	4
	4	7	10	9	1	2	2	4
	3	6	9	8	2	2	4	4
	2	5	8	7	2	2	4	4
	1	4	7	6	2	4	4	8
	0	3	6	5	2	4	4	8
		2	5	4	4	4	8	8
	1	4	3	4	4	8	8	
	0	3	2	4	8	8	16	
		2	1	4	8	8	16	
		1	0	8	8	16	16	
		0		8	8	16	16	

Niezależnie od numeru kwantowania  $QNO(i)$ , najmniej redukowane są współczynniki AC przynależne do strefy „0”. Dla przykładu, jeśli klasa kwantyzacji bloku  $qclass(j)$  wynosi 3, a liczba kwantyzacji  $QNO(i)$  makrobloku wynosi 10, wówczas wartości przedziałów kwantyzacji dla kolejnych stref wynoszą odpowiednio: 1, 2, 2, 4. Warto podkreślić, że jest to operacja, w której następuje bezpowrotna utrata danych, konieczna dla osiągnięcia żądanego współczynnika kompresji, który dla SD DV wynosi ok. 5:1.

Zanim nastąpi właściwa kwantyzacja, przeprowadzane jest skalowanie wstępne, niewystępujące w standardzie SMPTE 370M (DVCPRO HD). Dotyczy ono tylko współczynników AC, i wykonywane jest w następujący sposób na reprezentacji binarnej z bitem znaku:

- dla wartości klas kwantyzacji  $qclass(j)$  0, 1, 2
  - wejście: s b8 b7 b6 b5 b4 b3 b2 b1 b0
  - wyjście: s b7 b6 b5 b4 b3 b2 b1 b0;
- dla klasy  $qclass(j)$  wynoszącej 3
  - wejście: s b8 b7 b6 b5 b4 b3 b2 b1 b0
  - wyjście: s b8 b7 b6 b5 b4 b3 b2 b1

gdzie s jest bitem znaku.

### 3. Implementacja kwantyzacji odwrotnej

Proces kwantyzacji odwrotnej wymaga przeprowadzenia dwóch etapów:

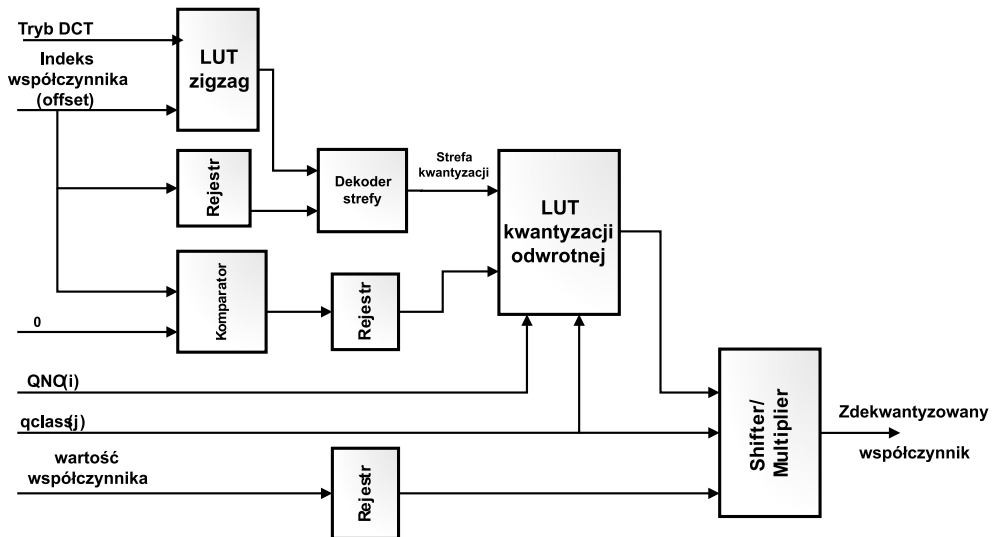
- kwantyzacji odwrotnej, czyli wyznaczenia wartości przesunięcia bitowego w prawo (lub mnożenia w zależności od rodzaju kompresji), zgodnej z tabelą 2,
- odwrotnego skalowania wstępnego, opcjonalnie, w zależności od wartości klasy kwantyzacji bloku, polegającego na rozszerzeniu reprezentacji bitowej, a niektórych przypadkach także mnożenia przez 2.

Schemat blokowy sprzętowego układu realizującego kwantyzację odwrotną przedstawia rysunek 3.

Głównymi blokami na schemacie przedstawionym na rysunku 3 są:

- LUT zigzag, zrealizowany jako tablica dwuwymiarowa, która jest odzwierciedleniem rysunku 1, zawierająca mapowanie zigzag, dla obydwu trybów DCT (8-8, 2-4-8).
- Dekoder strefy, implementujący rysunek 2, określający przynależność współczynnika AC do określonej strefy.
- LUT kwantyzacji odwrotnej, zrealizowany, podobnie jak blok LUT zigzag, w postaci tablicy dwuwymiarowej, służący do wyznaczenia przesunięcia bitowego, zgodnie z tabelą 2. W przypadku HD DV i DVCPRO HD służy on do wyznaczenia mnożnika dla zdekodowanego współczynnika podlegającego operacji kwantyzacji odwrotnej.

- Rejestry, pełniące roli linii opóźniających, ze względu na architekturę potokową realizowanego układu.
- Shifter/Multiplier, realizujący operację przesuwania bitowego w lewo (mnożenia przez wielokrotność liczby 2). Rezultat operacji zapisywany jest w pamięci blokowej układu FPGA, która jest elementem synchronizującym poszczególne elementy przetwarzające (*Processing Unit*) architektury potokowej realizującej całość dekodowania (dekompresji) sygnału wideo. W przypadku kompresji DV HD i DVCPRO HD, blok ten zastąpiony jest blokiem mnożącym, w którym mnożnik wybierany za pomocą poprzedzającej operacji LUT i zarazem jest mnożnikiem stałym KCM (*Constant Coefficient Multiplier*) [11].



Rys. 3. Schemat blokowy układu realizującego kwantyzację odwrotną

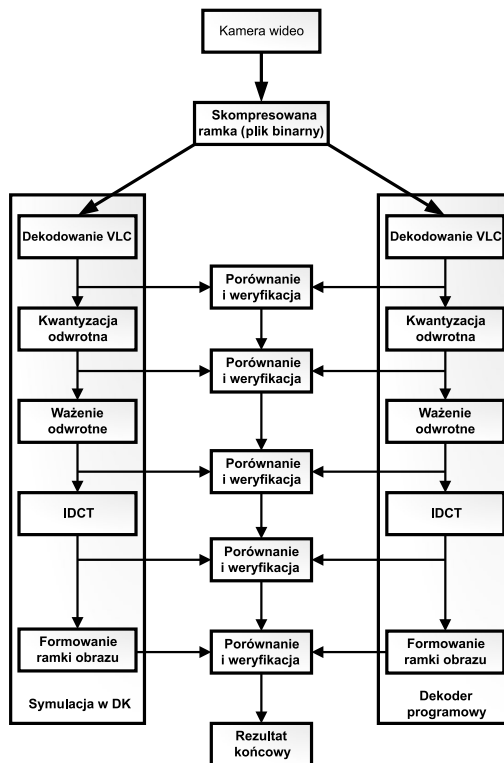
Powyższa operacja teoretycznie powinna być realizowana dla wszystkich 64 współczynników DCT. W większości przypadków charakterystyka danych w blokach DCT jest taka, że nie wszystkie współczynniki są niezerowe, przeciwnie, szczególnym i zarazem rzadkim przypadkiem jest sytuacja, w której wszystkie współczynniki AC mają niezerowe wartości. W związku z tym, dla skrócenia czasu realizacji operacji kwantyzacji odwrotnej, w poprzedzającej operacji dekodowania VLC (*Variable Length Coding*) [1, 3, 9], zapamiętywany jest indeks ostatniego, zdekodowanego, niezerowego współczynnika AC dla danego bloku DCT. Drugą wprowadzoną modyfikacją, która bezpośrednio wiąże się z wcześniejszą, jest wprowadzenie w architekturze kompletnego dekodera DV, operacji zerowania zawartości pamięci, w której zapisywane są współczynniki zdekwantyzowane. Wynika to z prostej przyczyny, że ilość niezerowych współczynników AC zazwyczaj różni się pomiędzy poszczególnymi blokami. Zatem nie wystarczy tylko nadpisanie poprzednio zapisanych danych, w przypadku kiedy poddawany kwantyzacji odwrotnej blok danych zawiera mniej istotnych współczynników AC niż ten, którego dane znajdowały się w pa-

mięci wcześniej. Ponieważ jest to operacja bezkontekstowa, tzn. taka, która dla uzyskania rezultatu nie potrzebuje znajomości otoczenia (innych współczynników AC), oraz korzysta z tablic LUT, dlatego jest z całą pewnością operacją szybką. Algorytm został zaimplementowany przy użyciu języka Handel-C [2], który w naturalny sposób umożliwia opis algorytmu w architekturze realizowanej potokowo, cechującej się największą wydajnością, kosztem wprowadzenia opóźnień między wejściem a wyjściem układu. Algorytm ten stanowi element w opracowywanym przez autora dekodерze wewnątrzramkowym realizowanym w układzie FPGA.

## 4. Rezultaty

Ze względu na fakt, że proponowana architektura operacji kwantyzacji odwrotnej stanowi fragment większego projektu dekodera wewnątrzramkowego, a także ze względu na wykorzystanie języka Handel-C do jej implementacji, weryfikacja opierała się na symulacji w środowisku DK Design Suite. Jest to tzw. zintegrowane środowisko deweloperskie (*Integrated Development Environment*), które oprócz edytora tekstów i menedżera projektów umożliwia symulację i debugowanie projektu w języku Handel-C.

Sposób weryfikacji funkcjonalnej realizowanego algorytmu przedstawia rysunek 4.



Rys. 4. Metodyka weryfikacji funkcjonalnej implementowanego algorytmu

Każda aplikacja sprzętowa wymaga weryfikacji funkcjonalnej poprawności realizacji algorytmu. Często stosowana, w szczególności do kodeków wideo jest metoda tzw. „czarnej skrzynki” (*blackbox testing*), w których sprawdzana jest jedynie poprawność danych na wyjściu systemu testowanego, uzyskanych w wyniku stymulacji układu danymi wejściowymi. Ponieważ cały schemat dekompresji składa się z kilku etapów [1, 3, 4, 9], stąd weryfikacji może podlegać również każdy z nich.

Dane referencyjne zostały utworzone przy użyciu dekodera programowego Quasar [10]. Rezultaty uzyskane na wyjściu każdego z etapów zostały zapisane w postaci plików tekstowych i wykorzystane do porównania z rezultatami uzyskanymi podczas symulacji.

Oprócz weryfikacji funkcjonalnej, główną zaletą symulatora Handel-C, w środowisku DK, jest możliwość określenia liczby cykli zegara potrzebnych implementacji do realizacji zadanego algorytmu. Wynika to z podstawowej idei języka, że każda podstawowa instrukcja wykonywana jest w jednym cyklu zegara. Zegar ten może być utożsamiany z zewnętrznym sygnałem zegarowym, ewentualnie z jego podzieloną częstotliwością. Z tego względu symulator nie dostarcza informacji o bezwzględnej ilości czasu potrzebnej do realizacji algorytmu, jedynie liczbę cykli zegara. Taka informacja na etapie symulacji pomaga w określeniu maksymalnej częstotliwości taktowania układu FPGA potrzebnej do spełnienia wymagań czasu rzeczywistego, w naszym przypadku dekodowania 25 klatek/s.

Rezultaty uzyskanych podczas symulacji czasów realizacji operacji odwrotnej dla 27 kolejnych segmentów wizji, przedstawia tabela 3. Segment wizji to 5 makrobloków pobranych z różnych fragmentów ramki w celu uśrednienia strumienia wideo.

**Tabela 3**  
Czas wykonania kwantyzacji odwrotnej pojedynczego segmentu wizji  
(w cyklach Handel-C, SD DV)

Minimalny	524
Maksymalny	906
Średni	719
Maksymalny (kwantyzacja odwrotna wszystkich współczynników)	2040
Czas wykonania najdłuższego etapu (IDCT) w opisywanym dekodерze DV	2011

Tabela pokazuje, że średni czas realizacji operacji kwantyzacji odwrotnej dla pojedynczego segmentu wizji został zredukowany średnio 2,8-krotnie.

Powyższe wyniki wskazują, że sposób implementacji kwantyzacji odwrotnej, w której przetwarzane byłyby wszystkie współczynniki AC, byłby większy od uzyskanego przez autora czasu realizacji najdłuższej operacji w całym dekodерze DV. Zastosowanie opisanej modyfikacji pozwoliło skrócić czas dekodowania całego segmentu wizji, a co za tym idzie, także całej ramki.



## 5. Wnioski

W artykule opisano szczegółowo proces skalowania wstępnego i kwantyzacji w standardzie cyfrowego wideo, a także zarysowano różnice pomiędzy kilkoma najpopularniejszymi standardami kompresji wewnątrzramkowej. Przedstawiona została również architektura potokowa realizująca operację kwantyzacji odwrotnej, która może być wykorzystana jako jednostka przetwarzająca sprzętowego dekodera wideo, realizowanego przez układ reprogramowalny. Zastosowanie opisanej modyfikacji pozwoliło skrócić czas dekodowania całego segmentu wizji, a co za tym idzie także całej ramki. Opis algorytmu w języku Handel-C [2] uprościł znacząco jego implementację, jednocześnie kod źródłowy jest czytelniejszy w porównaniu do typowych języków opisu sprzętu, jak VHDL, czy Verilog. Jako moduł, opisana jednostka przetwarzająca (*Processing Unit*), może być z powodzeniem wykorzystana w kompletnym dekodерze cyfrowego wideo.

## Literatura

- [1] *DVCAM Format Overview*, Sony Corporation, 2000.
- [2] Celoxica: *Handel-C Language Reference Manual*, Celoxica Ltd., 2005.
- [3] CENELEC: *Norma europejska EN 61834-2, Format SD dla systemów 525-60 i 625-50*, październik 1998.
- [4] CENELEC: *Norma europejska EN 61834-3, Format HD dla systemów 1125-60 i 1250-50*, grudzień 2002.
- [5] Dąbrowska A, Wiatr K., *Implementacja procesu kwantyzacji w strukturach FPGA dla potrzeb kompresji obrazu*. Automatyka (półrocznik AGH), t. 9, z. 3, 2005.
- [6] Gorgoń M., Cichoń S., Pac M., *Real-time Handel-C based implementation of DV decoder*. Proc. 2005 International Conference on Field Programmable Logic and Applications (FPL), IEEE 05EX1155, IEEE, Piscataway, New York, USA 2005, 130–135.
- [7] Kwiecień P., *Implementacja w układach FPGA modułów sprzętowych obsługujących cyfrową transmisję obrazu*. Katedra Automatyki AGH, Kraków, 2001.
- [8] Sima M. et al., *Inverse Quantization on FPGA-augmented TriMedia*. Delft University, 2002.
- [9] SMPTE: *SMPTE 370M-2006 for Television – Data Structure for DV-based Audio, Data and Compressed Video at 100 Mb/s 1080/60i, 1080/50i, 720/60p, 720/50p*. April 2006.
- [10] Quasar DV Codec: <http://sourceforge.net/projects/libdv/files/>.
- [11] Wiatr K., Jamro E., *Constant Coefficient Multiplication in FPGA Structures*. Proc. of the 26th Euromicro Conference on Digital Systems Design, IEEE Computer Society, 2000, 415–420.