

Sprzętowa implementacja algorytmu detekcji wzorców błędów DCT w hybrydowym algorytmie maskowania błędów transmisji obrazu stałego HECA

Grzegorz Andrzejewski, Wojciech Zajac

Streszczenie: W artykule przedstawiono działanie jednego z etapów hybrydowego algorytmu maskowania błędów transmisji obrazu stałego HECA - stopnia detekcji wzorców błędów DCT, oraz jego realizację sprzętową na poziomie behawioralnym. Wyróżniono dwa najistotniejsze bloki implementacyjne oraz zaprezentowano wyniki syntezy w środowisku Quartus II v.9.1 dla układu FPGA klasy Stratix III EP3SL70.

Słowa kluczowe: algorytm maskowania błędów transmisji, przetwarzanie danych wizyjnych, specyfikacja zachowania, implementacja sprzętowa algorytmu w FPGA

1. WPROWADZENIE

Cyfrowe przetwarzanie danych od szeregu lat zyskuje na popularności [7]. Zwiększa się liczba dziedzin, w jakich tego rodzaju techniki są stosowane, co z kolei wytwarza potrzebę opracowywania nowych metod, dedykowanych poszczególnym rozwiązaniom oraz dostosowywania technik istniejących.

Jednym ze skuteczniejszych sposobów szybkiego osiągnięcia zamierzonej sprawności poszczególnych mechanizmów przetwarzania danych jest stosowanie techniki hybrydowego łączenia różnych rozwiązań.

Jednym z przykładów tego rodzaju działań było opracowanie algorytmu HECA (ang. Hybrid Error Concealment Algorithm), hybrydowego algorytmu maskowania błędów transmisji monochromatycznego obrazu stałego, przetwarzanego z wykorzystaniem dyskretnej transformacji kosinusowej (ang. DCT, Discrete Cosine Transform). Szczegółowy opis założeń i działanie algorytmu opisano w [1, 2, 4, 5]. W niniejszym artykule przedstawione zostaną podstawowe informacje o algorytmie HECA, jego struktura oraz dokładniejsze informacje o jego etapie II, który jest podstawą do opracowania koncepcji mikrosystemu sprzętowego.

Sprzętowa realizacja algorytmu tego rodzaju przede wszystkim istotnie zwiększa zakres potencjalnego stosowania tego mechanizmu, dodatkowo pozwala na uzyskanie szeregu korzyści co do kosztu numerycznego i czasu realizacji operacji [8, 9, 10].

2. HYBRYDOWY ALGORYTM KOREKCJI BŁĘDÓW TRANSMISJI OBRAZU STAŁEGO HECA

Koncepcja jego działania opiera się na stosowaniu metod tzw. maskowania zakłóceń w miejsce tradycyjnie wykorzystywanych metod korekcji zakłóceń. Rozwiązania tej klasy posiadają szereg cech korzystnych z punktu widzenia konstrukcji systemów transmisyjnych:

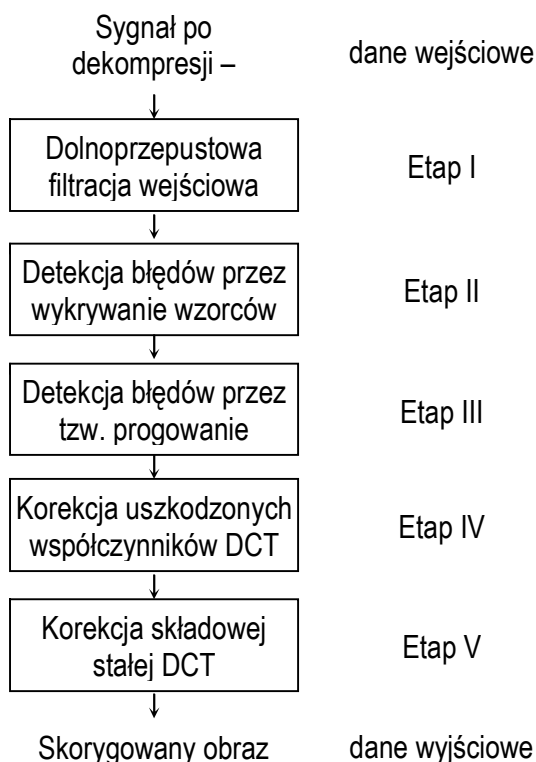
- nie jest wymagane utrzymywanie zwrotnego kanału transmisyjnego (sprzężenie zwrotne od odbiornika do nadajnika),
- kanał transmisyjny nie musi być obciążany dodatkowymi danymi nadmiarowymi sygnał (dane autokorekcyjne); ta cecha jest szczególnie istotna, gdyż dla sprawnego działania technik korekcji w tradycyjnych algorytmach autokorygujących wymagana ilość danych dodatkowych sięga 25% sygnału transmitowanego,
- nie ma konieczności ingerencji w strukturę ani sposób transmisji w istniejących systemach transmisyjnych.

Wskazane cechy stawiają algorytmy maskowania błędów w bardzo interesującym świetle.

Algorytm HECA opracowany został dla systemu transmisyjnego przy następujących założeniach: przetwarzany jest obraz monochromatyczny, złożony z $3 \leq M \leq 128 \times 3 \leq N \leq 128$ bloków danych o rozmiarach 8×8 pikseli, dozwolona wartość piksela leży w przedziale $0 \leq p \leq 255$; ostatnim założeniem jest stosowanie przez koder sygnału ośmiopunktowej transformacji DCT do dekorrelacji sygnału.

Algorytm HECA posiada budowę modułową, przepływ sygnału przedstawiono na rys. 1. Kolejne operacje realizowane są przez poszczególne bloki algorytmu:

- wejściowy filtr dolnoprzepustowy,
- blok detekcji wzorców błędów,
- blok detekcji uszkodzonych współczynników DCT,
- blok korekcji (maskowania) błędów,
- wyjściowy filtr wygładzający.



Rys. 1. Przepływ sygnału w algorytmie HECA

3. IMPLEMENTOWANY FRAGMENT ALGORYTMU

Przedmiotem niniejszego opracowania jest implementacja drugiego etapu przetwarzania sygnału w algorytmie, jest to etap odpowiedzialny za detekcję niepoprawnych wartości współczynników DCT, niosących informacje o obrazie przekształconym do postaci opisanej w dziedzinie częstotliwościowej. Detekcja oparta jest na analizie wstępnie odtworzonego obrazu pod kątem wyszukiwania tzw. wzorców błędów, charakterystycznych dla zastosowanej transformacji kosinusowej. Zagadnienie powstawania oraz detekcji wzorców DCT szerzej omówiono w [2, 6].

W wyniku działania tego etapu powstaje tzw. mapa błędów, czyli macierz, przechowująca informacje o lokalizacji współczynników DCT, uznanych za uszkodzone.

Mechanizm detekcji błędów w postaci wzorców DCT bazuje na sekwencyjnym sprawdzaniu faktu istnienia specyficznych rozkładów ekstremów w blokach danych analizowanego obrazu. Obraz jest wstępnie odtwarzany do postaci przestrzennej, złożonej z $M \times N$ bloków danych o rozmiarze 8×8 komórek na każdy blok. Na podstawie badań opisanych w [4, 5], ustalono, że do detekcji wzorców błędów DCT można wykorzystać analizę pierwszego wiersza i pierwszej kolumny każdego bloku danych. Ustalono osiem specyficznych rozkładów ekstremów i noszą one dalej nazwę wzorców (Tabela 1).

Tabela 1.
Tabela rozkładu ekstremów wzorców błędów DCT

nr wzorca	Typ ekstremum	Liczba ekstremów	nr komórki							
			1	2	3	4	5	6	7	8
1	min	0	-	-	-	-	-	-	-	-
	max	0	-	-	-	-	-	-	-	-
2	min	1	-	-	-	-	-	-	-	x
	max	1	x	-	-	-	-	-	-	-
3	min	2	-	-	-	x	x	-	-	-
	max	2	x	-	-	-	-	-	-	x
4	min	2	-	-	x	-	-	-	-	x
	max	2	x	-	-	-	-	x	-	-
5	min	4	-	x	x	-	-	x	x	-
	max	4	x	-	-	x	x	-	-	x
6	min	3	-	x	-	-	x	-	-	x
	max	3	x	-	-	x	-	-	x	-
7	min	4	-	x	-	x	x	-	x	-
	max	4	x	-	x	-	-	x	-	x
8	min	4	-	x	-	x	-	x	-	x
	max	4	x	-	x	-	x	-	x	-

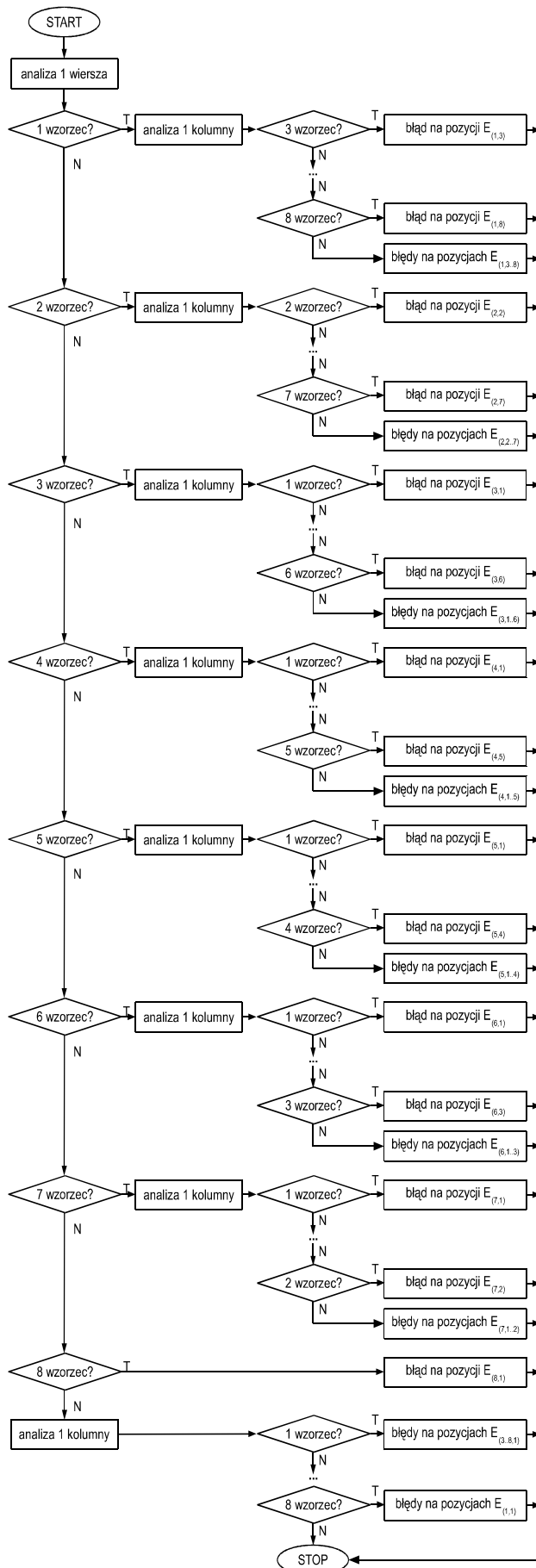
	1	2	3	4	5	6	7	8
1	126	110	98	66	52	41	12	5
2	45	x	x	x	x	x	x	x
3	12	x	x	x	x	x	x	x
4	4	x	x	x	x	x	x	x
5	4	x	x	x	x	x	x	x
6	32	x	x	x	x	x	x	x
7	56	x	x	x	x	x	x	x
8	98	x	x	x	x	x	x	x

Rys. 2. Przykład bloku danych i rozkładu ekstremów

Rysunek 2 ilustruje rozkład danych w pojedynczym bloku. Analizowane są pierwszy wiersz i pierwsza kolumna bloku (zaznaczone owalami). Miejsca wskazane strzałkami zawierają ekstrema specyficzne, przy czym dla wiersza 1 jest to rozkład zgodny ze wzorcem nr 2, a dla kolumny 1 rozkład zgodny ze wzorcem nr 3.

Dla potrzeb realizacji całości funkcji korekcji błędów transmisji obrazów HECA, błędy wykryte w blokach DCT przedstawiane są w macierzy błędów E, przy czym macierz ta ma rozmiar zgodny z macierzą DCT. Wartość domyślna każdej komórki macierzy E jest równa 1, co oznacza brak błędu. Wykrycie błędu powoduje wpisanie wartości 0 w komórki indeksowane numerami wzorców, przy czym w przypadku ogólnym wykrycie wzorca w wierszu determinuje indeks wiersza podbloku E odpowiadającego analizowanemu blokowi DCT, a wykrycie wzorca w kolumnie determinuje indeks kolumny podbloku E odpowiadającego analizowanemu blokowi DCT.

Szczegółowe działanie algorytmu realizującego wykrywanie rozkładów specyficznych, a co za tym idzie wykrywający współrzędne błędów w pojedynczym bloku macierzy DCT przedstawiono na rysunku 3.

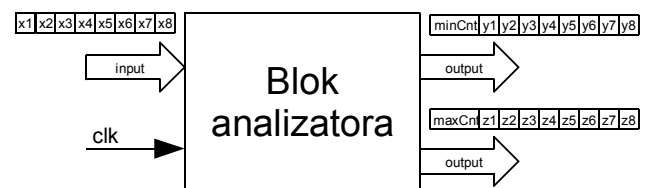


Rys. 3. Algorytm wykrywania rozkładów specyficznych w pojedynczym bloku macierzy współczynników DCT

4. REALIZACJA SPRZĘTOWA II ETAPU ALGORYTMU HECA

Tak przedstawiony algorytm implementować można na wiele sposobów. W niniejszym artykule przedstawiono kontynuację prac publikowanych m.in. w [3] dla realizacji sprzętowej, wykorzystującej programowalne matryce FPGA lub CPLD oraz języki opisu sprzętu klasy VHDL lub Verilog.

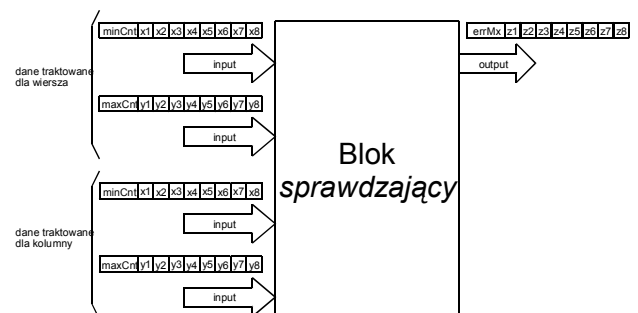
Realizację przygotowano z wykorzystaniem języka Verilog, jako bardziej elastycznego pod kątem implementacji obliczeniowych. Całość projektu podzielona została na dwa bloki: analizatora i sprawdzający. Blok analizatora stanowi podstawowy moduł odpowiedzialny za wykrywanie rozkładów ekstremów.



Rys. 4. Przepływ sygnału w bloku analizatora

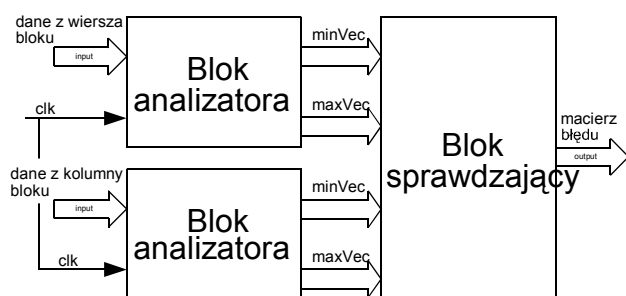
Na jego wejście podawany jest wektor liczb całkowitych z zakresu 0..255. Dane do niego zarówno mogą być pobrane z analizowanego wiersza jak i kolumny wybranego bloku danych. Blok wykonuje analizę i zwraca dwa wektory liczb całkowitych, z których pierwszy zawiera liczbę wykrytych minimów oraz ich rozkład, a drugi liczbę wykrytych maksimów oraz ich rozkład.

Blok sprawdzający przeprowadza równoległe sprawdzanie dla wiersza i kolumny analizowanego bloku danych warunków szczegółowych określonych w algorytmie z rysunku 3. Danymi wejściowymi dla bloku są odpowiednie wektory liczbowe zawierające informacje o rozkładach minimów i maksimów zarówno dla wiersza jak i dla kolumny. Wyjściem bloku jest wektor uporządkowany zgodnie ze strukturą pojedynczego bloku macierzy błędów E.



Rys. 5. Przepływ sygnału w bloku sprawdzającym

Ze względu na przyspieszenie obliczeń wykorzystano w strukturze ogólnej bloku detekcji błędów dwa bloki analizatora (współbieżne określanie rozkładów ekstremów) oraz jeden blok sprawdzający.



Rys. 6. Przepływ sygnału w bloku tetekcji

Tak przedstawiony blok detekcji błędów zrealizowany został z wykorzystaniem systemu Quartus II v.9.1. Do syntezy wybrano układ FPGA klasy Stratix III EP3SL70. Zajątość zasobów sprzętowych po syntezy systemu przedstawiono w tabeli 2.

Tabela 2.

Wyniki syntezy sprzętowej systemu detekcji wzorców

Blok	Zasoby			
	Zajęte CALUTs	Zajęte DLRs	Dostępne	% zajętości
analizator	6 594	85	54 000	12
sprawdzający	265	36	54 000	1
detekcji	13 453	206	54 000	25

Zapotrzebowanie zasobów sprzętowych na realizację bloku detekcji błędów DCT dla rozpatrywanej platformy implementacyjnej jest na poziomie 13.5 tys. bloków CALUT (ang. Combinational Adaptive Look-Up Table) oraz ok. 200 rejestrów DLR (ang. Dedicated Logic Register), co stanowi ogółem ok. 25% całych zasobów układu Stratix III EP3SL70. Pozwala to na wykorzystanie pozostałych zasobów układu do realizacji innych bloków systemu detekcji błędów HECA.

5. PODSUMOWANIE

Uzyskane wyniki pozwalają stwierdzić, że złożony algorytm przetwarzania danych wizyjnych może być z powodzeniem implementowany sprzętowo. Specyfika zadania algorytmicznego pozwala na wydzielenie w nim części operacji do realizacji współbieżnej. Zajątość zasobów w przyjętym do wykorzystania układzie nie jest szczególnie wysoka, co daje możliwość realizacji kolejnych zadań w układzie.

Ze względu na potrzebę operowania na dużych ilościach danych (reprezentacja obrazu, reprezentacja DCT, etc) na etapie dalszych badań należy opracować efektywny sposób reprezentacji danych jako pamięci w układach klasy FPGA lub współpracy tychże układów z pamięciami zewnętrznymi.

LITERATURA

- [1] Kisilewicz J., Zajac W. "An Improved Method of Error Concealment in Digital Image Transmission." W: *Modelling and simulation of system. Proceedings of the conference*. Ostrava (Cz) 1999
- [2] Sielicki A., Zajac W. *Nowa koncepcja korekcji zakłóceń transmisji cyfrowych danych wizualnych*

z wykorzystaniem techniki maskowania błędów. *Pomiary, Automatyka, Kontrola* nr 2-3/2003, s. 26-28

- [3] Zajac W. Andrzejewski G. *Akceleracja obliczeń wejściowego stopnia filtrującego hybrydowego algorytmu maskowania błędów transmisji obrazu stałego*. Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne nr 6/2008, s. 734-736
- [4] Zajac W. "An Error Concealment Algorithm for Digital Image Transmission." W: *Proceedings of XIV International Symposium on Computer and Information Sciences*, Kusadasi Turcja, October 1999
- [5] Zajac W. "A hybrid error concealment algorithm for digital image transmission." *Proceedings of The Second International Workshop on Multidimensional (nD) Systems*. NDS-2000. Czocha Castle, Poland, June 27-30, 2000. Zielona Góra: Technical University Press 2000, p. 263-268
- [6] Zajac W. "Nieoptymalna technika dekorelacji w cyfrowym przetwarzaniu obrazu." W: *Informatyka - sztuka czy rzemiosło - KNWS'05: II Konferencja* Nukowa, Złotniki Lubañskie, Polska, 2005.- Zielona Góra, 2005
- [7] Clarke R. *Digital Compression of Still Image and Video*. Academic Press, London 1995
- [8] Saegusa, T.; Maruyama, T.; Yamaguchi, Y.; *How fast is an FPGA in image processing?* International Conference on Field Programmable Logic and Applications, 2008. FPL 2008.
- [9] Albo-Canals, J.; Villasante-Bembibre, J.A.; Riera-Babures, J.; Fernandez-Garcia, N.A.; Brea, V.M.; *An efficient FPGA implementation of a DT-CNN for small image gray-scale pre-processing*. European Conference on Circuit Theory and Design, 2009. ECCTD 2009.
- [10] McCurry, P.; Morgan, F.; Kilmartin, L.; *Xilinx FPGA implementation of an image classifier for object detection applications*, International Conference on Image Processing, 2001. Proceedings. 2001



dr inż. Grzegorz Andrzejewski
Uniwersytet Zielonogórski
Wydział Elektrotechniki, Informatyki i Telekomunikacji
Instytut Informatyki i Elektroniki
ul. Podgórna 50
65-246 Zielona Góra

tel.: 68 328 2599
e-mail: W.Zajac@iie.uz.zgora.pl



dr inż. Wojciech Zajac
Uniwersytet Zielonogórski
Wydział Elektrotechniki, Informatyki i Telekomunikacji
Instytut Informatyki i Elektroniki
ul. Podgórna 50
65-246 Zielona Góra

tel.: 68 328 2599
e-mail: W.Zajac@iie.uz.zgora.pl