

Grzegorz ANDRZEJEWSKI, Wojciech ZAJĄC

INSTYTUT INFORMATYKI I ELEKTRONIKI, UNIwersYTET ZIELONOGÓRSKI, ul. Licealna 9, 65-417 Zielona Góra

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

Dr inż. Grzegorz ANDRZEJEWSKI

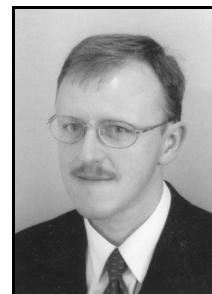
Pracownik Instytutu Informatyki i Elektroniki UZ. Absolwent Wydziału Elektrycznego Politechniki Poznańskiej. W roku 1995 uzyskał dyplom w specjalności elektronicznej aparatury i systemów pomiarowych. Stopień doktora z zakresu informatyki uzyskał na Wydziale Informatyki Politechniki Szczecińskiej w roku 2002. Jego zainteresowania naukowe ukierunkowane są na zagadnienia modelowania i syntezy systemów sterowania cyfrowego.



e-mail: G.Andrzejewski@iie.uz.zgora.pl

Dr inż. Wojciech ZAJĄC

Absolwent kierunku Informatyka i Metrologia na Wydziale Elektrycznym Wyższej Szkoły Inżynierskiej w Zielonej Górze. W roku 2000 uzyskał stopień doktora nauk technicznych na Wydziale Elektroniki Politechniki Wrocławskiej. Od roku 2001 kieruje pracami zespołu Zakładu Techniki Informatycznych Instytutu Informatyki i Elektroniki UZ. Zainteresowania naukowe autora obejmują zagadnienia modelowania systemów i algorytmów cyfrowego przetwarzania danych.



e-mail: W.Zajac@iie.uz.zgora.pl

Streszczenie

W artykule przedstawiono wyniki badań nad realizacją sprzętową modułu detekcji wzorców błędów transmisji obrazu stałego. Jest on częścią hybrydowego algorytmu maskowania błędów transmisji HECA. Opisano podstawy działania aparatu analitycznego, algorytm jego działania oraz realizację sprzętową na poziomie behawioralnym. Wyróżniono najistotniejsze bloki implementacyjne, zaprezentowano wyniki syntezy w środowisku Quartus II v.9.1 dla układu FPGA klasy Stratix III EP3SL70 oraz przedyskutowano uzyskane wyniki.

Słowa kluczowe: detekcja wzorców DCT, maskowanie błędów transmisji, przetwarzanie danych wizyjnych, specyfikacja zachowania, implementacja sprzętowa algorytmu w FPGA.

Hardware implementation of DCT error pattern detection module of hybrid error concealment algorithm HECA

Abstract

The paper presents a conception of hardware implementation of DCT pattern detection module of Hybrid Error Concealment Algorithm (HECA) [2]. The research is aimed at implementing a hardware version of the module, using possibilities of parallel operation in FPGA and optimizing the algorithm structure for hardware implementation and performance. Paragraph 1 gives introduction to digital image transmission error concealment. Paragraph 2 presents a structure and operation of the HECA algorithm. The dataflow is presented (Fig. 1.) and the implemented module is identified. Paragraph 3 deals with the mechanism of DCT error pattern occurring [3] and describes a method for detection of such patterns [2]. Paragraph 4 is focused on operation of the error pattern module of HECA in details. There are presented specific features of error patterns for a given DCT block size (Tab. 1) The erroneous block data example is shown in Fig. 2., while the error location storing example is presented in Fig. 3. The operation algorithm for error pattern detection is discussed and presented in Fig. 4. Paragraph 5 describes hardware implementation procedures. The implementation process is presented, the structure of hardware solution is shown (Figs. 5, 6, 7.) and discussed. The hardware resources consumption of the synthesis results is given in Tab. 2. Paragraph 6 contains the conclusion and directions for the future work. The research conclusions are that the DCT error pattern detection algorithm can be successfully implemented in FPGA with acceptable resources consumption. Such an implementation allows performing some of the algorithm elements in parallel, accelerating the operation. The problem is that the data amount tends to be high and it would be recommended to develop more effective notation to store such data in FPGA.

Keywords: DCT error detection, error concealment, visual data processing, behaviour specification, hardware implementation in FPGA.

1. Wprowadzenie

Cyfrowe przetwarzanie danych od szeregu lat zyskuje na popularności [3]. Zwiększa się liczba dziedzin, w jakich tego rodzaju

techniki są stosowane, co z kolei stwarza 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).

Koncepcja maskowania błędów opiera się na stosowaniu metod tzw. maskowania zakłóceń w miejsce tradycyjnie wykorzystywanych metod ich korekcji. 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 obciążony dodatkowymi danymi nadmiarującymi 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.

Szczegółowy opis założeń i działanie algorytmu HECA opisano w [2]. W niniejszym artykule przedstawione zostaną podstawowe informacje o algorytmie, 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 [4, 5].

2. Hybrydowy algorytm korekcji błędów transmisji obrazu stałego HECA

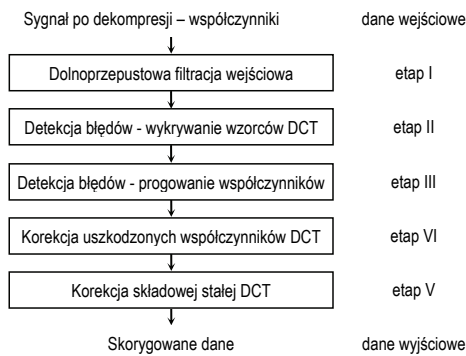
Algorytm HECA został zaprojektowany jako część toru odtwarzania sygnału w urządzeniu dekodującym. Dane wejściowe do algorytmu mają postać sygnału wstępnie odtworzonego i zdekompresowanego przed operacją odwrotnej dekorelacji (współczynników DCT). Posiada on budowę modułową, przepływ sygnału przedstawiono 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.

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 i dekoder sygnału ośmiopunktowej transformacji DCT do zrealizowania operacji prostej i odwrotnej dekorrelacji sygnału.

Prezentowana praca dotyczy sprzętowej realizacji bloku detekcji wzorców błędów DCT – etapu II. W pracy [1] przedstawione zostały wyniki prac nad sprzętową implementacją etapu I algorytmu.



Rys. 1. Etapy przetwarzania sygnału w algorytmie HECA
Fig. 1. Signal processing scheme in HECA algorithm

3. Aparat detekcji wzorców błędów DCT

Algorytm HECA przeznaczony jest do przetwarzania cyfrowych danych obrazu stałego, dekorrelowanych z wykorzystaniem transformaty DCT. Stosowanie tej transformaty obciążone jest potencjalną możliwością wystąpienia, podczas odtwarzania obrazu z postaci częstotliwościowej do przestrzennej, zakłóceń o charakterze zdeterminowanym równaniami filtrów pasmowych transformaty [3].

$$X'_{ij} = DCT_{ij}^T * Y_{ij} * DCT_{ij} = \begin{cases} \frac{1}{\sqrt{2^K}} * y_{mn} * \frac{1}{\sqrt{2^L}} & \text{dla } 1 \leq i \leq 2^K, 1 \leq j \leq 2^L, m=1, n=1 \\ \frac{1}{\sqrt{2^K}} * y_{mn} * \frac{1}{\sqrt{2^L}} \cos \left[\frac{(2j-1)(n-1)\pi}{2^{L+1}} \right] & \text{dla } 1 \leq i \leq 2^K, 1 \leq j \leq 2^L, m=1, 1 < n \leq 2^L \\ \frac{1}{\sqrt{2^K}} \cos \left[\frac{(2i-1)(m-1)\pi}{2^{K+1}} \right] * y_{mn} * \frac{1}{\sqrt{2^L}} & \text{dla } 1 \leq i \leq 2^K, 1 \leq j \leq 2^L, 1 < m \leq 2^K, n=1 \\ \frac{1}{\sqrt{2^K}} \cos \left[\frac{(2i-1)(m-1)\pi}{2^{K+1}} \right] * y_{mn} * \frac{1}{\sqrt{2^L}} \cos \left[\frac{(2j-1)(n-1)\pi}{2^{L+1}} \right] & \text{dla } 1 \leq i \leq 2^K, 1 \leq j \leq 2^L, 1 < m \leq 2^K, 1 < n \leq 2^L \end{cases} = f_m(i) * y_{mn} * f_n(j), \quad (1)$$

gdzie:

- $2^K, 2^L$ - rozmiary macierzy bazowych transformaty DCT,
- m, n - indeks wiersza i kolumny macierzy współczynników,

f_m, f_n - m -ta i n -ta funkcja bazowa transformaty,
 y_{mn} - współczynnik wzbudzący filtry.

Niech dany będzie sygnał wejściowy DCT w postaci macierzy Y . Odwrotna transformacja macierzy danych Y ma postać określonego równaniem:

$$X' = DCT^T * Y * DCT, \quad (2)$$

gdzie:

- Y - macierz współczynników DCT,
- DCT - macierz bazowa transformaty,
- DCT^T - transponowana macierz bazowa transformaty,
- X' - macierz odtworzonych danych wyjściowych.

Wartości elementów X'_{ij} macierzy odtworzonych danych X' w funkcji wartości współczynników wzbudzących filtry określa równanie (1).

Z równania wynika, że funkcja danych macierzy X' ma postać złożoną z iloczynu m -tej funkcji bazowej DCT w funkcji współrzędnej wiersza i , wartości współczynnika y_{mn} i n -tej funkcji bazowej w funkcji współrzędnej kolumny j .

Ponieważ w obrębie pierwszej kolumny macierzy X' indeks j jest stały i równy 1, zachodzi:

$$X'_{i1} = f_m(i) * y_{mn} * f_n(1). \quad (3)$$

Ponieważ $f_n(1) = const$, dane w pierwszej kolumnie macierzy X' zależne są od iloczynu m -tej funkcji bazowej DCT w funkcji współrzędnej wiersza i oraz współczynnika C_1 :

$$X'_{i1} = f_m(i) * C_1, \quad \text{gdzie } C_1 = y_{mn} * f_n(1). \quad (4)$$

Oznacza to, że przebieg funkcji luminancji danych w pierwszej kolumnie macierzy bloku odtworzonych danych X' ma taką samą liczbę i położenie ekstremów co funkcja bazowa f_k jednowymiarowej transformaty DCT o indeksie k zgodnym z indeksem wiersza m współczynnika wzbudzącego. Analogiczne rozumowanie dotyczy przebiegu funkcji luminancji danych w pierwszym wierszu macierzy odtworzonych danych X' - ma ona taką samą liczbę i położenie ekstremów co funkcja bazowa f_k jednowymiarowej transformaty DCT o indeksie zgodnym z indeksem kolumny n współczynnika wzbudzącego.

W oparciu o informacje o przebiegu funkcji luminancji w pierwszej kolumnie i wierszu analizowanego bloku wstępnie odtworzonego obrazu można określić współrzędne współczynników, które wywołały wzorce błędów. Wyróżnialność wzorca w bloku zależy od treści zakłóconego sygnału (najwyraźniej widoczne dla odbiorcy i najłatwiejsze do wykrycia są wzorce zakłócające mało dynamiczne fragmenty obrazu) i przede wszystkim od wartości błędów (różnicy w wartości oryginalnej i zakłóconej uszkodzonego współczynnika). Jednak niezależnie od amplitudy zakłócenia, wzorzec przez nie generowany jest jednoznacznie związany z położeniem uszkodzonego współczynnika w macierzy danych częstotliwościowych

4. Implementowany moduł algorytmu HECA

Implementowany II etap algorytmu HECA opiera swoje działanie na wniosku sformułowanym powyżej. W wyniku działania tego etapu powstaje tzw. mapa błędów – macierz, przechowująca informacje o lokalizacji współczynników DCT, uznanych za uszkodzone.

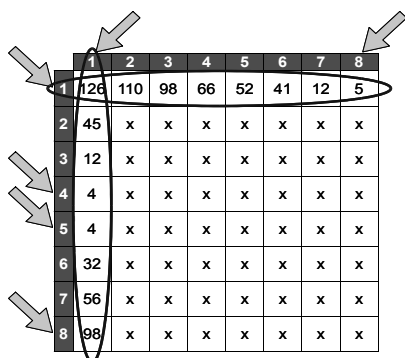
Obraz jest wstępnie odtwarzany do postaci przestrzennej, złożonej z $M \times N$ bloków danych o rozmiarze 8×8 komórek na każdy blok. W oparciu o równanie (4) ustalono osiem specyficznych rozkładów ekstremów wzorców błędów (tabela 1).

Rys. 2 ilustruje działanie mechanizmu detekcji wzorców na przykładowych danych bloku. Analiza pierwszego wiersza i pierwszej kolumny bloku (zaznaczone owalami) wskazuje eks-

tema specyficzne: dla wiersza 1 jest to rozkład zgodny ze wzorcem nr 2, a dla kolumny 1 rozkład zgodny jest ze wzorcem nr 3.

Tab. 1. Układ ekstremów wzorców błędów DCT
Tab. 1. Localisation of extremes in DCT error patterns

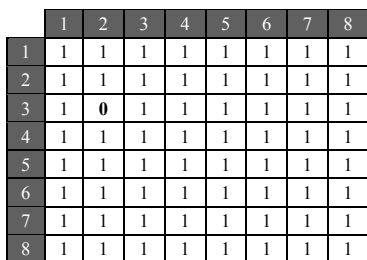
nr wzorca	Typ ekstremum	Liczba ekstremów	nr komórki							
			1	2	3	4	5	6	7	8
1	min	0	-	-	-	-	-	-	-	-
	max	0	-	-	-	-	-	-	-	-
2	min	1	-	-	-	-	-	-	-	x
	max	1	x	-	-	-	-	-	-	-
3	min	2	-	-	-	x	x	-	-	-
	max	2	x	-	-	-	-	-	-	x
4	min	2	-	-	x	-	-	-	-	x
	max	2	x	-	-	-	-	x	-	-
5	min	4	-	x	x	-	-	-	x	x
	max	4	x	-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



Rys. 2. Przykład zakłóconego bloku danych i rozkładu ekstremów wzorca błędów
Fig. 2. An example of erroneous block data and error pattern extremes locations

Lokalizacja wykrytych błędów przechowywana jest w tzw. macierzy błędów E o rozmiarze zgodnym z rozmiarem z macierzy danych 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.

Rys. 3 ilustruje sposób zaznaczenia lokalizacji przykładowego błędu, przedstawionego na rys. 2 w mapie błędów.

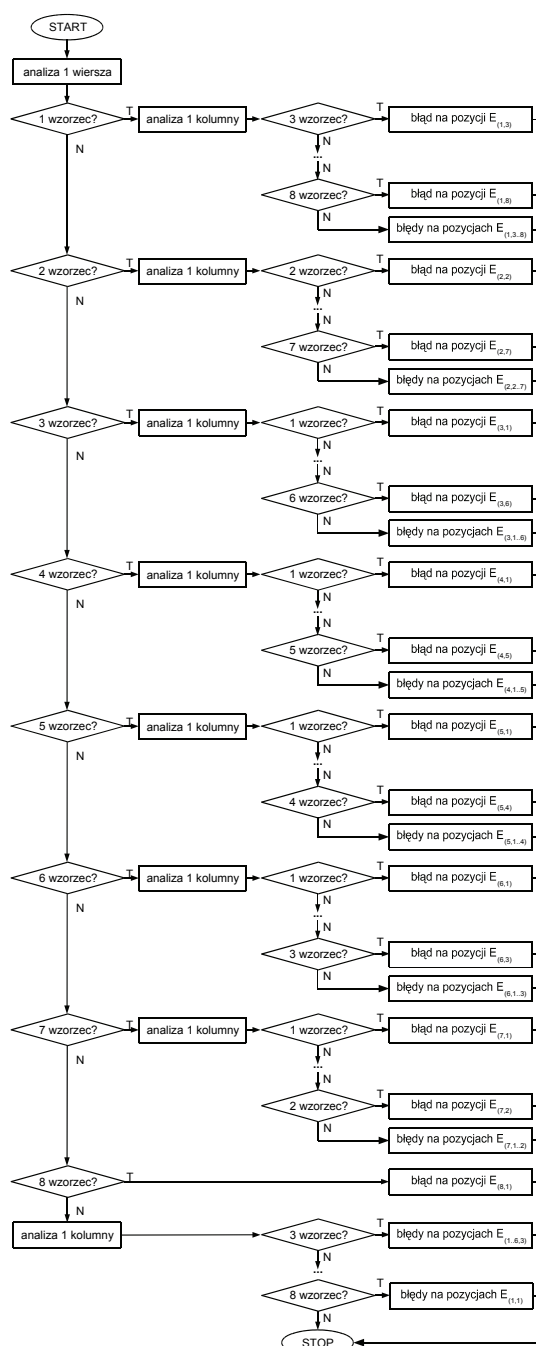


Rys. 3. Przykład błędu z rys. 2 zaznaczonego w mapie błędów
Fig. 3. Example error of Fig. 2 in error map matrix

Mapa błędów jest przekazywana później do kolejnego etapu (progowania współczynników), który dokonuje jej dalszych modyfikacji w miarę wykrywania kolejnych błędów. Po zakończeniu jego działania mapa stanowi podstawę do wykonania operacji maskowania błędów, wskazywanych przez mapę. Działanie algorytmu realizującego wykrywanie rozkładów specyficznych w pojedynczym bloku macierzy DCT przedstawiono na rys. 4.

Jego pierwszym krokiem jest pobranie pierwszego wiersza analizowanego bloku danych DCT i analizowanie go pod kątem wykrycia kolejnych wzorców błędów. Realizowane jest to przez porównanie liczby ekstremów z tabelą wzorców specyficznych.

W przypadku zlokalizowania i identyfikacji wzorca specyficznego w pierwszym wierszu bloku (co daje informacje o numerze wiersza danych, w której wystąpiło uszkodzenie danych) przeprowadzana jest analiza pierwszej kolumny danych, w celu zidentyfikowania wzorca specyficznego w kolumnie (współrzędnej kolumnowej uszkodzenia). Jeśli identyfikacja współrzędnej kolumnowej się nie powiedzie, algorytm zaznacza w mapie błędów wszystkie współczynniki w wierszu uznanym za błędny. W przypadku niewykrycia wzorca specyficznego w wierszu, algorytm dokonuje analizy pierwszej kolumny danych. Jeśli wykryte zostanie tam wzorec specyficzny, cała kolumna zaznaczana jest jako wymagająca korekcji.



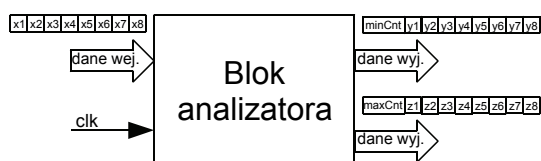
Rys. 4. Algorytm wykrywania specyficznych rozkładów danych w pojedynczym bloku macierzy współczynników DCT
Fig. 4. Algorithm of specific data pattern detection in a single DCT data block

W procesie optymalizacji funkcjonowania algorytmu wyeliminowano szereg operacji porównania. Eliminacja ma miejsce w dwóch przypadkach: w odniesieniu do współczynników DCT, które zostały wyzerowane w wyniku działania pierwszego etapu algorytmu HECA (dolnoprzepustowa filtracja wejściowa) oraz dla współczynników, których wzorce błędów dają przebiegi słabo wykrywalne na tym etapie: współczynniki y_{11} , y_{12} i y_{21} .

5. Realizacja sprzętowa etapu detekcji wzorców DCT

Tak przedstawiony algorytm może znaleźć różnorodne implementacje. W niniejszym artykule przedstawiono kontynuację prac publikowanych m.in. w [1] dla realizacji sprzętowej, wykorzystującej programowalne matryce FPGA lub CPLD oraz języki opisu sprzętu klasy VHDL lub Verilog.

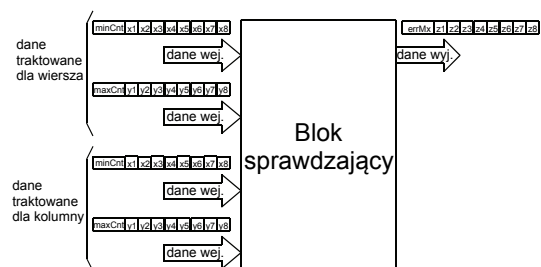
Realizację przygotowano z wykorzystaniem języka Verilog, jako bardziej elastycznego pod kątem implementacji obliczeniowych. Całość projektu podzielona została na dwa bloki: analizatora (rys. 5) i sprawdzający (rys. 6). Blok analizatora stanowi moduł odpowiedzialny za wykrywanie rozkładów ekstremów. Jego działanie polega na przeprowadzeniu identyfikacji liczby i lokalizacji ekstremów w danych wejściowych.



Rys. 5. Przepływ sygnału w bloku analizatora
Fig. 5. Signal flow in the analyser block

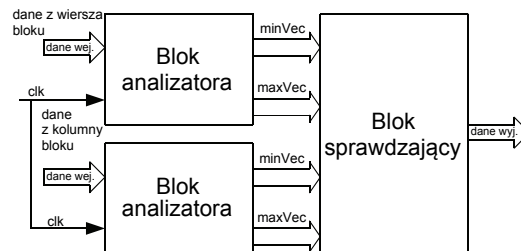
Na wejście analizatora podawany jest wektor liczb całkowitych z zakresu 0..255. Dane do niego zarówno mogą być pobrane z analizowanego wiersza jak i kolumny wybranego bloku danych. Blok wykonuje analizę i zwraca dwa wektory liczb całkowitych, z których pierwszy zawiera liczbę wykrytych minimów oraz ich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

Dane wyjściowe z bloku analizatora są kierowane na wejście bloku sprawdzającego celem porównania liczby i lokalizacji ekstremów w analizowanym bloku z ekstremami specyficznymi.



Rys. 6. Przepływ sygnału w bloku sprawdzającym
Fig. 6. Signal flow in the checker block

Blok sprawdzający przeprowadza równoległe sprawdzanie dla wiersza i kolumny analizowanego bloku danych warunków szczegółowych określonych w algorytmie z rys. 4. Danymi wejściowymi 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. Ze względu na przyspieszenie obliczeń wykorzystano w strukturze ogólnej bloku detekcji błędów dwa bloki analizatora (równoległe określanie rozkładów ekstremów) oraz jeden blok sprawdzający (rys. 7). Tak przedstawiony blok detekcji błędów zrealizowany został z wykorzystaniem systemu Quartus II v.9.1. Do syntezy wybrano układ FPGA klasy Stratix III EP3SL70.



Rys. 7. Przepływ sygnału w bloku detekcji
Fig. 7. Signal flow in the detector block

Tab. 2. Zajętość zasobów sprzętowych po syntezy systemu
Tab. 2. Resources consumption after the system synthesis

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 wynosi ok. 13.5 tys. bloków CALUT (ang. Combinational Adaptive Look-Up Table) oraz ok. 200 rejestrów DLR (ang. Dedicated Logic Register) (Tab. 2.), co stanowi ogółem ok. 25% całych zasobów użytego układu. Pozwala to na wykorzystanie pozostałych zasobów układu do realizacji innych bloków systemu.

6. Wnioski

Uzyskane wyniki pozwalają stwierdzić, że złożony algorytm przetwarzania danych wizyjnych może być z powodzeniem implementowany sprzętowo. Zajętość zasobów w przyjętym do wykorzystania układzie nie jest szczególnie wysoka, co daje możliwość realizacji kolejnych zadań w układzie.

Specyfika zadania algorytmicznego pozwala na wydzielenie w nim części operacji do realizacji równoległej, co znacząco wpływa na prędkość realizacji operacji, zwiększając zakres potencjalnego stosowania omawianego algorytmu.

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

7. Literatura

- [1] Zając 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.
- [2] Zając W.: An Error Concealment Algorithm for Digital Image Transmission. Proceedings of XIV International Symposium on Computer and Information Sciences, Kusadasi Turcja, October 1999.
- [3] Clarke R.: Digital Compression of Still Image and Video. Academic Press, London 1995.
- [4] 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.
- [5] McCurry P., Morgan, F., Kilmartin L.: Xilinx FPGA implementation of an image classifier for object detection applications, International Conference on Image Processing, 2001.