

Daniel MAJEWSKI

POLITECHNIKA WARSZAWSKA, INSTYTUT RADIOELEKTRONIKI

Architektura dekodera wideo MPEG-2 dla wymagań HDTV

Mgr inż. Daniel MAJEWSKI

Pracownik w dziale R&D u jednego z większych operatorów telekomunikacyjnych na stanowisku: Specjalista w dziale usług multimedialnych. Ukończone studia magisterskie na Politechnice Warszawskiej w Instytucie Radioelektroniki. Udział w pracach nad transkoderm MPEG-2/H.264 z wyżej wymienionym instytutem. Uprawiana dziedzina: Implementacja algorytmów cyfrowego przetwarzania danych w układach cyfrowych.



e-mail: perskryt@wp.pl

Streszczenie

Wdrożenie standardu kompresji wideo H.264/AVC wiąże się z koniecznością zapewnienia konwersji informacji ze standardu MPEG-2. W ramach prac nad sprzętowym transkoderm H.264/AVC na MPEG-2 został wykonany dekodery MPEG-2, spełniający wymagania dla standardu HDTV. Dekoder realizuje dekompresję strumienia wideo, w wyniku czego zrekonstruowane są kolejne ramki zakodowanej sekwencji. Ze względu na wysokie wymagania na przepustowość, architektura stosuje blokową organizację dostępu do pamięci wewnętrznej. Wykorzystując tryb sekwencyjnego dostępu ciągle do pamięci dynamicznej dla bloków 8x8 uzyskana została duża wydajność transferu danych. W szczególności oznacza to rezygnację z sekwencyjnego zapisu kolejnych linii obrazu. Wyniki syntezy i analizy czasowej wykonanej w programie Quartus II pokazują, że cały układ może pracować przy częstotliwości 133 MHz w oparciu o układy FPGA Stratix II.

Słowa kluczowe: FPGA, kompresja wideo, MPEG-2, architektura sprzętowa.

Hardware architecture of HDTV MPEG-2 decoder

Abstract

The use of the H.264/AVC video compression standard involves the need for the data conversion from MPEG-2. Within the work on the MPEG-2/H.264 hardware transcoder, the MPEG-2 HDTV video decoder has been developed. The decoder decompresses video streams and reconstruct successive frames. Owing to the requirements on the high throughput, the architecture accesses the external memory in the block fashion. Using the burst mode for 8x8 blocks, a high throughput has been achieved. Particularly, sequential line-by-line access to the memory is avoided. The synthesis results show that the decoder can work at the 133 MHz clock on the FPGA Stratix II platform.

Keywords: FPGA, video compression, MPEG-2, hardware architecture.

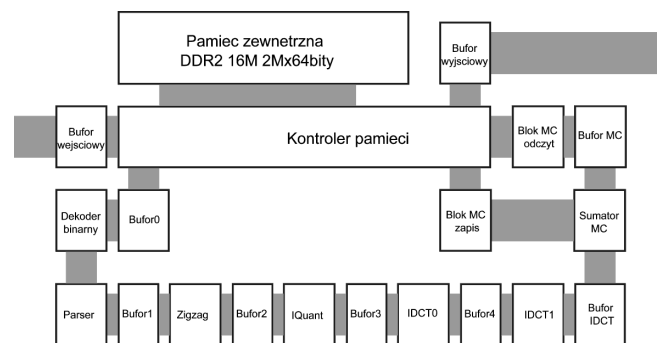
1. Wprowadzenie

Standard kompresji wideo MPEG-2 [1] został opracowany prawie piętnaście lat temu. Do tej pory stał się bardzo popularnym standardem kompresji. Jest powszechnie używany w telewizji cyfrowej i archiwizacji wideo. Mimo pojawiania się nowych standardów kompresji wideo, takich jak H.264/AVC kompresja MPEG-2 pozostanie obecna jeszcze przez pewien okres czasu. Wielu producentów telewizyjnych nie posiada tak dużych środków umożliwiających całkowitą wymianę sprzętu kompatybilnego z nowym standardem kompresji. Rozwiązaniem w tej sytuacji może być zastosowanie specjalnych urządzeń umożliwiających konwersję starszego standardu MPEG-2 na standard H.264/AVC. Urządzenia takie nazywane są transkoderami MPEG2/H.264. Składają się one z dekodera MPEG2 oraz kodera H.264. W pierwszej kolejności skompresowany strumień wideo MPEG-2 jest podawany na dekodery, który dekoduje go do postaci nieskompresowanych ramek obrazu wideo. Następnie ramki w tej postaci trafiają do kodera, który koduje je do standardu H.264/AVC.

W literaturze można spotkać wiele opracowań poszczególnych modułów kodeka MPEG-2 [5]. Jednak mało miejsca jest poświęcone problemowi ich integracji w systemie. W niniejszym artykule szczególna uwaga zwrócona została na optymalizację czasową dekodera. Prezentowany dekodery MPEG-2 został zaprojektowany do współpracy z koderem H.264/AVC w ramach opracowywanego transkodera sprzętowego. Jednym z podstawowych wymagań na dekodery była kompatybilność ze strumieniami HDTV (MP@HL).

2. Architektura dekodera

Zaproponowana architektura dekodera została przedstawiona na rys. 1.



Rys. 1. Diagram blokowy dekodera
Fig. 1. Block diagram of the decoder

Dekoder składa się z następujących bloków:

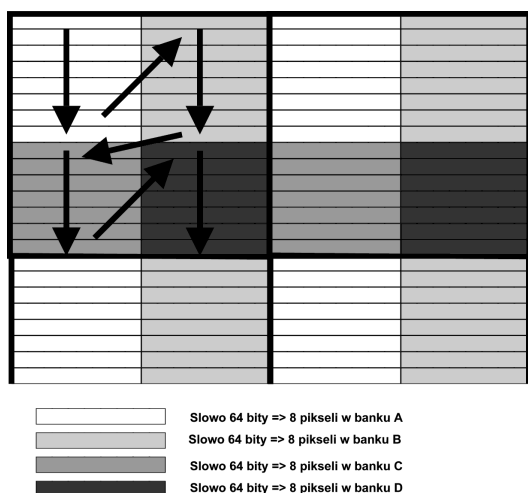
- *Bufor wejściowy* pobiera kolejne bajty z zewnętrznego źródła i formuje w osiem ośmiobajtowych słów, które zapisywane są do zewnętrznej pamięci.
- *Bufor0* - Zadaniem tego bufora jest dostarczanie kolejnych porcji danych do dekodera binarnego. Dwa wymienione bufony razem z odpowiednim obszarem w pamięci zewnętrznej stanowią bufor VBV dekodera.
- *Dekoder binarny* na zlecenie „parsera” pobiera kolejne symbole ze strumienia wideo z modułu *Bufor0*. Zgodnie z normą mogą to być symbole o zmiennej lub stałej długości. Dodatkowo moduł ten potrafi odnaleźć następną sekwencję synchronizującą. Główną cechą tego modułu jest to, że dekoduje on jeden symbol na takt zegara, co jest wymagane przy strumieniach wideo o przepływności MP@HL.
- Moduł *Bufor1* pobiera kolejne elementy bloku, a następnie udostępnia modułowi Zig-zag. Bufor ten posiada oddzielne porty do pisania i czytania i dwie przestrzenie pamięci. Dzięki temu możliwy jest równoczesny zapis i odczyt z dwóch różnych bloków.
- Moduł *Zig-zag* umiejscawia współczynniki DCT w odpowiednie miejsca bloku 8x8. Przychodzące dane mogą być ustawione na dwa sposoby dostępne w standardzie.
- Bufory *Bufor2*, *bufor3*, *bufor4* mają budowę podobną do *bufor1*.
- Dekwantyzator skaluje współczynniki transformaty DCT według współczynnika skali i maczyzy skalowania. Blok ten posiada możliwość użycia własnych maczyzy współczynników.
- Moduły *IDCT0* i *IDCT1* wykonują jednowymiarową odwrotną transformatę IDCT. W połączeniu z modułem *bufor4* składają się na transformatę dwuwymiarową.
- Moduł *Bufor_IDCT* przechowuje zdekodowany makroblok.

- Moduł *MC* pobiera bloki reprezentujące makrobloki z ramek referencyjnych. Dodatkowo moduł ten wykonuje interpolację pikseli potrzebną przy rozdzielczości półpikselowej oraz uśrednia makrobloki przy predykcji dwukierunkowej (Backward i Forward) lub predykcji Dual Prime. Zadaniem tego bloku jest również zapisanie ostatecznie zdekodowanego makrobloku do zewnętrznej pamięci.
- Bufor *Bufor_MC* służy do przechowywania makrobloków referencyjnych przed dodaniem ich do zdekodowanych błędów predykcji.
- Moduł *Sumator_MC* sumuje makroblok referencyjny z zdekodowanym błędem predykcji.
- *Kontroler pamięci* obsługuje wszystkie moduły odwołujące się do pamięci zewnętrznej.

3. Architektura wybranych bloków

3.1. Pamięć zewnętrzna

Kompresja MPEG-2 osiąga stosunkowo wysoki stopień kompresji dzięki zastosowaniu predykcji międzyramkowej INTER. Zastosowanie tej predykcji pociąga za sobą konieczność przechowywania w pamięci wcześniej zdekodowanych ramek. Wielkość przechowywanych ramek wymusza użycie zewnętrznej pamięci RAM. Na dzień dzisiejszy najbardziej popularną i dostępną pamięcią RAM jest pamięć DDR lub DDR2. Pamięci tego typu różnią się istotnie od tradycyjnych pamięci SRAM, ponieważ dostęp do wybranej komórki nie jest swobodny i natychmiastowy. Pamięci DDR mają złożony sposób adresowania. Składają się one z oddzielnych banków, z których każdy zawiera określoną ilość wierszy. Każdy wiersz składa się z komórek Podstawowym problemem przy pracy z pamięcią DDR jest konieczność aktywowania wiersza, w którym znajduje się wybrana komórka pamięci. W praktyce problem ten objawia się tym, że przy zapisie ośmiu komórek pamięci potrzeba najpierw poświęcić kilka taktów zegara na aktywowanie danego wiersza. Kolejnym problemem występującym w tych pamięciach jest konieczność zapisu do kilku komórek pamięci. W praktyce odwoływać się trzeba do czterech bądź ośmiu komórek. Narzuca to pewnego rodzaju gradacje w dostępie do ramek obrazu przechowywanych w pamięci zewnętrznej. Rozwiązaniem w tej sytuacji jest specjalna organizacja pamięci [3], [4] pozwalająca na eliminację taktów zegara potrzebnych na aktywację wiersza i wykorzystująca zapis pakietowy „burst” ośmiu komórek.



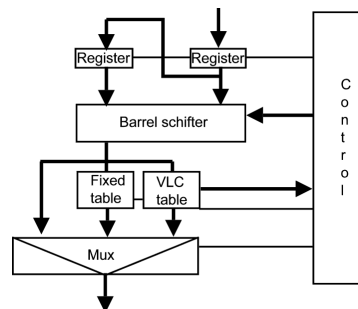
Rys. 2. Organizacja pamięci obrazu
Fig. 2. Picture memory arrangement

W opracowanym dekodерze wykorzystano pamięć o szerokości 64 bitów. Dzięki temu odczyt całego bloku zajmuje osiem taktów zegara. Kolejne osiem „pikseli” w przestrzeni adresowej znajduje

się w następnej linii. Dodatkowo każdy blok w makrobloku jest zapisywany w innym banku pamięci. Zabieg ten pozwala odczytywać kolejne bloki w makrobloku zmieniając banki pamięci bez straty cykli na aktywację wierszy w bankach.

3.2. Dekoder binarny

Dekoder binarny został zaprojektowany w sposób umożliwiający dekodowanie jednego symbolu w jednym takcie zegara. Dzięki temu uzyskano możliwość dekodowania strumieni o przepływności wykorzystywanej w HDTV.

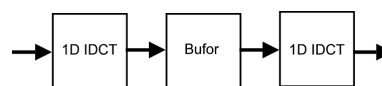


Rys. 3. Diagram dekodera binarnego
Fig. 3. Diagram of the binary decoder

Sercem dekodera jest przesuwnik bitowy (Barrel shifter) oraz tabela kodów (Fixed, VLC). Tabela porównuje kolejne ciągi bitów, odnajduje określone symbole i zarazem steruje przesuwnikiem za pomocą informacji o długości danego symbolu. Dekoder ma możliwość dekodowania kodów o stałej i zmiennej długości. Zaprojektowany dekodер binarny może pracować z zegarem 133MHz przy wykorzystaniu układów FPGA Stratix II.

3.3. Moduł odwrotnej transformaty IDCT

Moduł dwuwymiarowej odwrotnej transformaty IDCT zrealizowano za pomocą dwóch oddzielnych transformat jednowymiarowych i specjalnego bufora [2]. Blok pierwszej transformaty pobiera współczynniki z poprzedniego bufora w kolejności wierszowej. Blok drugiej transformaty pobiera współczynniki z przedstawionego bufora w kolejności kolumnowej. Przetworzenie bloku 8x8 współczynników zajmuje 64 taktów zegara.

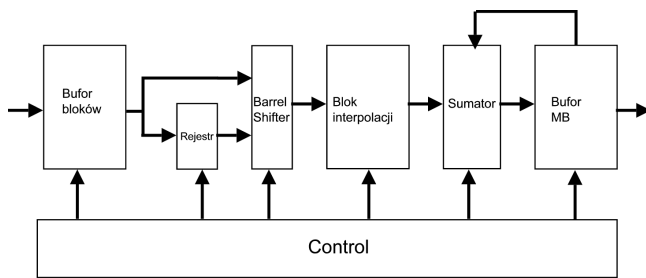


Rys. 4. Schemat bloku IDCT
Fig. 4. Diagram of the IDCT block

Pojedynczy moduł 1D IDCT składa się z rejestru przesuwnego na osiem współczynników, ośmiu rejestrów, ośmiu układów mnożących, drzewa sumatorów i pamięci współczynników. Dzięki zastosowaniu podwójnego bufora możliwe jest równoczesne zapisywanie nowej transformaty i odczytywanie starej. Moduł ten może być taktowany zegarem 133MHz występującym w systemie.

3.4. Moduł MC

Wspomniana wcześniej architektura pamięci ramek posiada określoną gradację. Powoduje to pewne problemy przy odczycie makrobloków referencyjnych, które mogą być wskazywane przez wektory ruchów o różnych długościach niedopasowanych do dostępnej gradacji. Z tego powodu konieczne jest przeczytanie wszystkich bloków zawierających choć część poszukiwanego bloku.

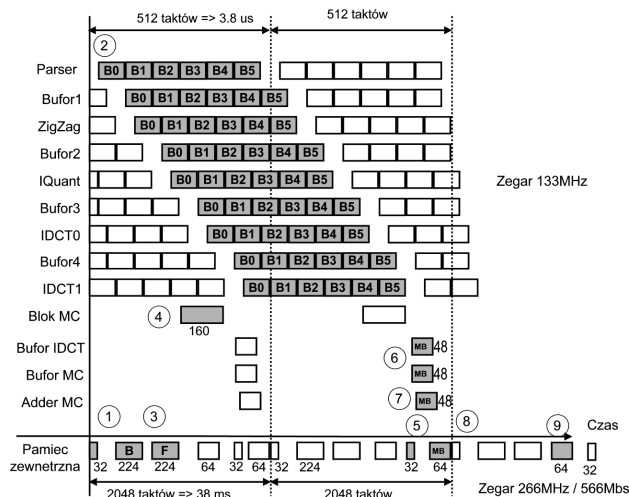


Rys. 5. Schemat blokowy układu kompensacji ruchu
Fig. 5. Block diagram of the motion compensation unit

Na rys. 5 przedstawiono moduł kompensacji ruchu (MC) potrafiący pobrać z dostarczonych danych interesujący makroblok. Bufor bloków zawiera wszystkie bloki, w których zawiera się poszukiwany MB. Z tego to bufora pobierane są sąsiednie bloki i podawane są na rejestr i przesuwnik próbek. Na wyjściu przesuwnika uzyskiwane są kolejne linie poszukiwanego bloku. W bloku interpolacji dokonywana jest interpolacja wartości pikseli dla rozdzielczości połowy piksela. Następnie uzyskiwany blok jest uśredniany z innym blokiem. Uśrednianie to jest potrzebne przy predykcji „dwukierunkowej” lub predykcji *Dual Prime*. Wynik tego etapu przetwarzania jest następnie zapisywany do bufora. Cały układ współpracuje z zegarem systemowym 133MHz.

4. Analiza czasowa

Cały dekodery został zweryfikowany w symulacjach w celu sprawdzenia czy cały przepływ danych jest poprawny. Układ dekodera zawiera jeden główny zegar systemowy 133MHz, który steruje większością układu. Dodatkowo przy układach współpracujących z pamięcią zewnętrzną występuje zegar 266MHz. Na rys. 6 przedstawiono przepływ danych podczas dekodowania jednego makrobloku.



Rys. 6. Analiza czasowa
Fig. 6. Timing of the decoder

Przedstawiony wykres składa się z dwóch głównych części. Pierwsza (górna i większa) dotyczy części dekodera pracującego z zegarem 133 MHz, a druga (dolna) przedstawia przepływ danych do pamięci zewnętrznej przy zegarze 266 MHz. Dla parametrów obrazu HDTV jeden makroblok powinien być zdekodowany w czasie 3.8 us czyli 512 taktów zegara 133 MHz. Na wykresie podano również ilość taktów zegara dla poszczególnych etapów dekodowania. Proces dekodowania makrobloku przebiega następująco:

1. Na samym początku zapelniany jest bufor ze strumieniem video.
 2. W tym momencie dekodery binarny zaczyna przetwarzać strumień. Pobiera on wektory ruchu potrzebne w module kompensacji ruchu MC oraz zaczyna dekodować wszystkie bloki w makrobloku.
 3. Moduł kompensacji ruchu MC pobiera wszystkie dane potrzebne do odtworzenia makrobloków referencyjnych.
 4. Moduł kompensacji ruchu MC pobiera dane z kontrolera pamięci i wykonuje interpolację pikseli oraz uśrednianie bloków dla predykcji dwukierunkowej lub predykcji *Dual Prime*.
 5. Odświeżenie pamięci zewnętrznej.
 6. Odczytanie makrobloku z błędem predykcji z bufora IDCT oraz makrobloku referencyjnego z bufora MC.
 7. Sumowanie powyższych makrobloków.
 8. Zapisanie w pełni zdekodowanego makrobloku do pamięci zewnętrznej.
 9. Pobranie makrobloku do bufora wyjściowego
- W zaprojektowanym dekodery wykorzystano stały dostęp do pamięci zewnętrznej przez wszystkie moduły dekodera. Rozwiązanie takie umożliwia zaplanowanie wszystkich odczytów i zapisów do pamięci zewnętrznej. Pozwala ono wyeliminować w dużym stopniu nieefektywne odwołania do pamięci.

5. Wyniki syntezy i podsumowanie

Tab. 1. Wyniki syntezy
Tab. 1. Synthesis results

Moduł	Stratix II [ALUT]
Dekoder binarny	1327
IDCT	1275 + 32DSP
MC	1493
Kontroler pamięci	1145
Zegar [MHz]	133

Zaproponowane rozwiązania i rozkład przepływów pozwalają na przetwarzanie strumieni video o rozdzielczości HDTV.

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

6. Literatura

- [1] ISO/IEC 13818-2: 1995 MPEG video standard, ITU-T H.262 Recommendation.
- [2] Xilinx xapp611 Application Note „Video Decompression Using IDCT, 2007.
- [3] Jui-Hua Li. Optimization of Queueing Performance and Design Variabls in a Single-Bus Shared-Memory System – with Application to MPEG-2 Video Decoder System, SANTA CLARA UNIVERSITY 2002.
- [4] Nien-Tsu Wang. Processing and Storage Models for MPEG-2 Main Level and High Level Video Decoding — A Block-Level Pipeline Approach., SANTA CLARA UNIVERSITY 2004
- [5] Agnieszka Dąbrowska, Implementacja kodeka MPEG-2 w układach FPGA. Pomiary, Automatyka, Kontrola, ISSN 0032-4140 ,2007.