

**Andrzej ABRAMOWSKI, Grzegorz PASTUSZAK**POLITECHNIKA WARSZAWSKA, WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH,  
ul. Nowowiejska 15/19, 00-665 Warszawa**Propozycja architektury dekodera arytmetycznego standardu H.264/AVC****Mgr inż. Andrzej ABRAMOWSKI**

Uzyskał stopień magistra inżyniera o specjalności telekomunikacja we wrześniu 2009 roku na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Obecnie pracuje w Instytucie Radioelektroniki Politechniki Warszawskiej na stanowisku asystenta naukowego. Od 2008 roku zajmuje się realizacją układów w technologii FPGA związanych z kompresją wideo.



e-mail: a.abramowski@ire.pw.edu.pl

**Dr inż. Grzegorz PASTUSZAK**

Uzyskał stopień magistra inżyniera o specjalności mikroelektronika w czerwcu 2001 roku a stopień doktora inżyniera o specjalności techniki multimedialne w czerwcu 2006 roku. Obecnie jest pracownikiem naukowym Instytutu Radioelektroniki Politechniki Warszawskiej. Jego obszar zainteresowania obejmuje: architektury i algorytmy VLSI, przetwarzanie obrazów/video/audio oraz kompresję, wydajne cyfrowe układy scalone.



e-mail: g.pastuszak@ire.pw.edu.pl

**Streszczenie**

Artykuł przedstawia architekturę binarnego dekodera arytmetycznego standardu H.264/AVC, zdolną do obsługi profilu High. Pozwala ona na dekodowanie w przybliżeniu jednego symbolu w czasie pojedynczego taktu sygnału zegarowego, głównie dzięki modyfikacji kolejności kroków algorytmu oraz wprowadzeniu potokowości w pętli sprzężenia zwrotnego. Architektura została opisana w języku VHDL, a analiza wyników syntezy wskazuje, że umożliwia ona obsługę sekwencji HDTV.

**Słowa kluczowe:** H.264/AVC, CABAC, dekodery entropijny, dekodery wideo.

**Context-adaptive binary arithmetic decoder architecture for H.264/AVC****Abstract**

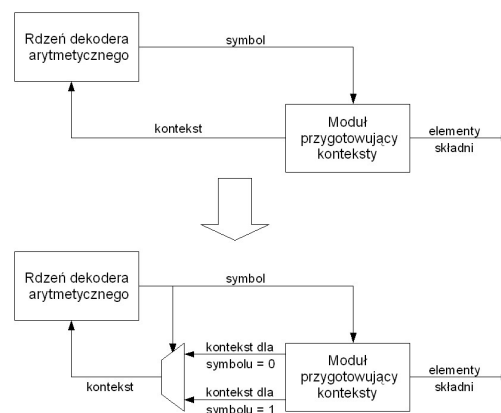
This paper presents a novel architecture of the H.264/AVC binary arithmetic decoder, which conforms to High Profile, including all chroma formats and Macroblock Adaptive Frame/Field coding (MBAFF). It is able to decode almost one symbol per clock cycle, while consuming very limited hardware resources. The main feature of the proposed solution is the parallelization of the feedback loop between the arithmetic decoder core and the context generator, achieved by calculation of two contexts for each possible case in advance and selection of the correct one immediately after the current symbol is available. Some smaller optimizations include the modification of the operation order on the critical path, and the projection of *mvd* values, allowing economizing the memory usage. The architecture was described in VHDL and realized using TSMC 0.13 $\mu$ m technology. Comparison of the synthesis results and the performance with previous works proves that the proposed architecture maintains the best trade-off between the speed of the video processing and the hardware utilization, while being able to process HDTV in real time.

**Keywords:** H.264/AVC, CABAC, entropy decoder, video decoder.

**1. Wstęp**

Standard H.264/AVC jest jedną z najnowszych propozycji w dziedzinie kodowania sygnałów wideo, zapewniającą wysoki stopień kompresji przy bardzo dobrej jakości obrazu. Jedną z najważniejszych innowacji jest możliwość wykorzystania binarnego kodowania arytmetycznego, co pozwala zwiększyć efektywność kompresji o ok. 15%. Tak duży zysk wydaje się predestynować koder arytmetyczny do wykorzystania przy obsłudze sekwencji HDTV, jednak znaczącym problemem okazuje się wysoka złożoność obliczeniowa algorytmu.

Efektom opisanego problemu jest szereg propozycji [3-5], bazujących głównie na wykorzystaniu statystyki występowania elementów składni w strumieniu, co pozwala na dekodowanie kilku symboli na takt sygnału zegarowego dla najczęściej pojawiających się układów elementów. Alternatywne podejście zostało przedstawione w [2] i rozbudowane w niniejszej pracy.



Rys. 1. Rozbicie pętli sprzężenia zwrotnego  
Fig. 1. Optimization of the central feedback loop

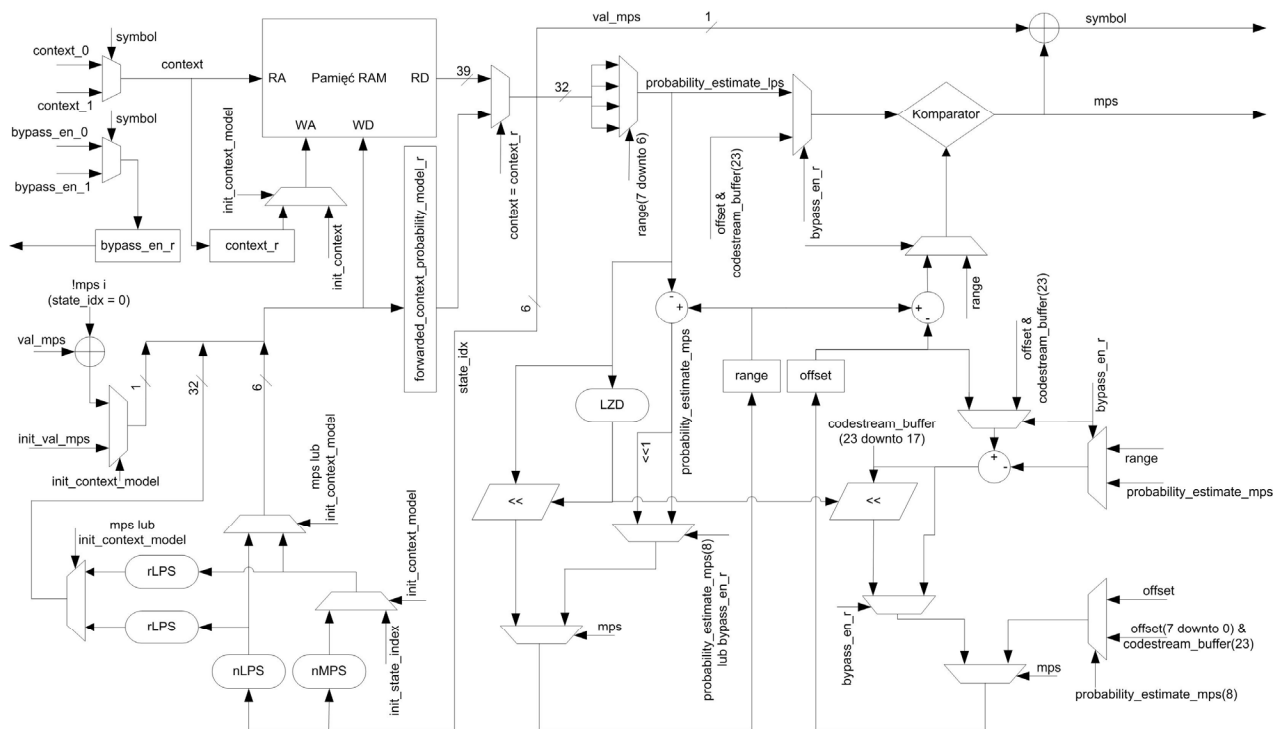
Dzięki przygotowaniu zestawu dwóch kontekstów, dla obu możliwych wartości symbolu, zaproponowana architektura jest w stanie przygotować symbol w czasie pojedynczego taktu sygnału zegarowego, niezależnie od typu obecnie dekodowanego elementu składni. Jednocześnie wykorzystuje ona bardzo ograniczoną ilość zasobów sprzętowych.

**2. Algorytm działania dekodera arytmetycznego dla standardu H.264/AVC**

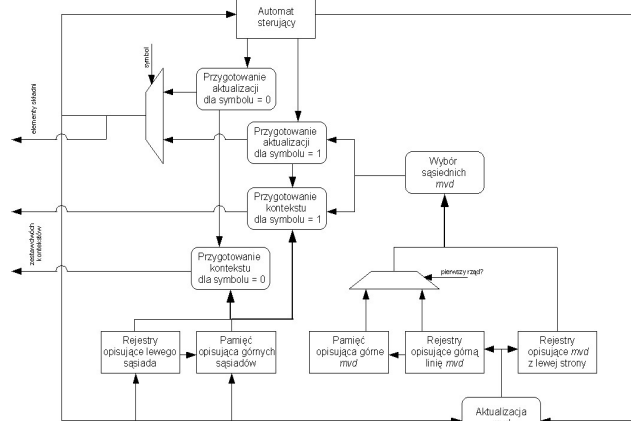
W standardzie H.264/AVC przyjęto, że w momencie rozpoczęcia dekodowania nowego plastra zostaną przypisane wartości domyślne zarówno modelom prawdopodobieństwa dla poszczególnych kontekstów, jak i kluczowym elementom rdzenia dekodera arytmetycznego (rejestrów *range*, opisującym górną granicę przedziału prawdopodobieństwa, oraz rejestrów *offset*, zawierającym obecną wartość słowa kodowego).

Właściwy cykl pracy dekodera obejmuje przygotowanie kontekstu, wybór odpowiadającego mu modelu prawdopodobieństwa, zdekodowanie pojedynczego symbolu na podstawie parametrów rdzenia, aktualizację wartości rejestrów *range* i *offset*, wraz z modelem prawdopodobieństwa, oraz sprawdzenie binaryzacji, złożonej z uzyskanych do tej pory symboli. Jeżeli otrzymana binaryzacja jest poprawna (zawiera się w zbiorze dopuszczalnych wartości) to dekodery kończy pracę dla danego elementu składni.

Model prawdopodobieństwa dla pojedynczego kontekstu składa się z indeksu stanu, bardziej prawdopodobnej wartości symbolu oraz przedziału, opisującego prawdopodobieństwo przyjęcia przez symbol wartości mniej prawdopodobnej. Wybór prawidłowej wartości symbolu dokonywany jest poprzez porównanie rejestru *offset* z różnicą rejestru *range* i opisanego przedziału.



Rys. 2. Rdzeń dekodera arytmetycznego  
Fig. 2. Arithmetic decoder core



Rys. 3. Moduł przygotowujący konteksty  
Fig. 3. Context generator

Po podjęciu decyzji rejestry z rdzenia dekodera arytmetycznego są poddawane renormalizacji, czyli usunięciu zer na najbardziej znaczących pozycjach, poprzez przesunięcie młodszych bitów i uzupełnienie jedynkami rejestru *range* oraz danymi ze strumienia rejestru *offset*. Równolegle aktualizowany jest model prawdopodobieństwa, tak aby uwzględnił podjętą decyzję.

Oprócz standardowego trybu pracy dekodera przewidziane są jeszcze tryby *bypass* i *decode\_terminate*. Ten pierwszy występuje dla przypadków, gdy prawdopodobieństwa dla obu możliwych wartości symbolu są sobie równe. Drugi jest wykorzystywany dla ostatniego symbolu obsługiwanego przez dekodery arytmetyczny.

Podstawowym problemem w zaprojektowaniu efektywnej architektury dekodera arytmetycznego jest silna zależność pomiędzy kolejnymi krokami algorytmu. Dopiero po przygotowaniu kontekstu możliwy staje się wybór modelu prawdopodobieństwa i zdekodowanie kolejnego symbolu. Jednakże w typowej sytuacji wyznaczenie kontekstu wymaga znajomości wartości uprzednio przygotowanego symbolu. W ten sposób powstaje pętla sprzężenia zwrotnego.

### 3. Ogólny zarys architektury

W standardzie H.264/AVC wykorzystano binarny koder arytmetyczny, co oznacza, że każdy symbol może przyjmować jedną z dwóch dostępnych wartości. Dzięki temu możliwe jest przygotowanie niezależnych zestawów danych, w tym kontekstów i wybranie właściwego w momencie podjęcia decyzji, co obrazuje rys. 1.

Proponowana architektura składa się z trzech, niezależnych modułów. Moduł inicjalizujący modele prawdopodobieństwa ma za zadanie jedynie pomóc w przywróceniu domyślnego stanu dekodera arytmetycznego na początku każdego plastra. Rdzeń umożliwia dekodowanie kolejnych symboli, podczas gdy moduł przygotowujący konteksty zajmuje się obliczaniem kontekstów, dekodowaniem oryginalnych wartości elementów składni oraz zarządzaniem pracą całego dekodera arytmetycznego.

### 4. Rdzeń dekodera arytmetycznego

Głównym celem, postawionym przy projektowaniu architektury rdzenia dekodera arytmetycznego, była zdolność do zdekodowania symbolu w czasie pojedynczego taktu sygnału zegarowego, niezależnie od wybranego trybu pracy. Dodatkowo, aby ograniczyć zużycie zasobów sprzętowych, wszystkie tryby pracy wykorzystują te same elementy logiczne. Dla przypadku *decode\_terminate* wiąże się to z przygotowaniem odpowiedniego modelu prawdopodobieństwa dla kontekstu 276, podczas gdy do obsługi trybu *bypass* wystarczy dodanie zestawu multiplexerów.

Podstawą zaproponowanej architektury rdzenia jest dwuportowa pamięć RAM, przechowująca wartości wszystkich 1024 modeli prawdopodobieństwa. Każdy z nich znajduje się pod adresem odpowiadającym przyporządkowanemu mu kontekstowi.

Zdekodowany symbol determinuje kontekst podawany na wejście adresowe pamięci RAM, dzięki czemu na wyjściu pojawia się indeks stanu, bardziej prawdopodobna wartość symbolu oraz zbiór czterech przedziałów, odpowiadających prawdopodobieństwu przyjęcia przez symbol wartości mniej prawdopodobnej. Z tego zbioru wybrany zostaje jeden, *probability\_estimate\_lps*, na podstawie najbardziej znaczących bitów rejestru *range*. Wybór właściwej wartości symbolu w standardzie H.264/AVC, jak już zostało wspomniane, rozstrzyga nierówność:

$$\text{offset} \geq \text{range} - \text{probability\_estimate\_lps} \quad (1)$$

Jednocześnie określa ono ścieżkę krytyczną rdzenia, a przez to całego dekodera arytmetycznego. Jednak proste przekształcenie:

$$\text{offset} - \text{range} \geq \text{probabilisty\_estimate\_lps} \quad (2)$$

$$\text{range} - \text{offset} \leq \text{probability\_estimate\_lps} \quad (3)$$

pozwała usunąć subtraktor ze ścieżki z dwoma multiplekserami, prowadząc do bardziej zbalansowanego czasowo rozmieszczenia elementów logicznych. Podjęta decyzja wyznacza wartości symbolu.

Renormalizację rejestrów należy oddzielnie rozpatrzyć dla obu możliwych przypadków. Gdy wybrana została wartość mniej prawdopodobną, kluczową rolę odgrywa jednostka LZD (ang. Leading Zeros Detector), pozwalająca określić liczbę zer na najstarszych bitach sygnału podanego na wejście, czyli wymagane przesunięcie bitowe. Dla przypadku bardziej prawdopodobnego przesunięcie bitowe nie występuje lub wynosi jeden, czyli wystarczy zwykle sprawdzenie najstarszego bitu zaktualizowanej wartości rejestru *range*. W obu przypadkach renormalizacja nie wymaga marnowania dodatkowych taktów sygnału zegarowego.

Tablice nMPS i nLPS, zrealizowane w formie LUT (ang. Lookup Table), umożliwiają wyznaczenie nowego indeksu stanu dla przypadku bardziej i mniej prawdopodobnego. Na tej podstawie, z tablicy rLPS, można otrzymać zbiór czterech przedziałów, opisujących prawdopodobieństwo przyjęcia przez symbol wartości mniej prawdopodobnej. Po podjęciu decyzji, na podstawie równania 3, właściwe dane podawane są na wejście pamięci RAM.

## 5. Moduł przygotowujący konteksty

Moduł przygotowujący konteksty jest odpowiedzialny za pracę całego dekodera arytmetycznego. Podstawą jego działania jest automat, którego stany odpowiadają poszczególnym elementom składniowym. W każdym taktie sygnału zegarowego wyznaczane są dwa zestawy danych, dla obu możliwych wartości symbolu, zawierające sygnały sterujące poszczególnymi modułami, kontekst, flagę określającą wystąpienie trybu *bypass*, uzyskane elementy składni oraz uaktualniony stan modułu z wszystkimi niezbędnymi danymi na temat obecnego i sąsiednich makrobloków.

Sam kontekst dla danego symbolu zależy od szeregu parametrów, z których najważniejsze to indeks symbolu, obecny element składni i wartości elementów opisujących sąsiednie partycje. Najwięcej problemów sprawia przechowywanie danych na temat sąsiadów. Proponowane rozwiązanie składa się z pamięci RAM, zawierającej informacje na temat linii makrobloków powyżej, oraz zestawu rejestrów dla lewego sąsiada. Jeżeli moduł obsługuje tryb MBAFF, operujący na parach makrobloków, oba te elementy są zdublowane, odpowiednio dla górnej i dolnej linii.

Szczególny przypadek występuje dla kodowanych predykcijnie wektorów ruchu *mvd*, gdzie do przygotowania kontekstu niezbędna jest znajomość dokładnej wartości *mvd* dla sąsiedniej partycji. Pojedynczy takt sygnału zegarowego zabiera samo wyznaczenie położenia poszczególnych *mvd* w ramach makrobloku. Do przechowywania danych referencyjnych wyodrębniono oddzielną pamięć RAM i zbiór rejestrów. Jednocześnie wykorzystano operację rzutowania, zgodnie z tabelą 1, nie wpływającą na wartość przygotowanego kontekstu, a pozwalającą na drobne ograniczenie wykorzystywanych zasobów.

Tab. 1. Rzutowanie wartości *mvd*

Tab. 1. Projection of *mvd* value

wartość <i>mvd</i>	$\geq 65$	63 - 64	61 - 62	$\leq 60$
wartość przechowywana	63	62	61	oryginalna wartość

## 6. Wyniki syntezy i efektywność działania dekodera arytmetycznego

Proponowana architektura została opisana w języku VHDL i zweryfikowana w oparciu o oprogramowanie referencyjne JM16.

Efektywność działania dekodera, zdefiniowaną jako średnią liczbę taktów sygnału zegarowego niezbędną do przetworzenia pojedynczego makrobloku, przedstawia tabela 2, zawierająca zestawienie dla przykładowych sekwencji, kodowanych z podpróbko- waniem chrominancji 4:2:0 i współczynnikiem kwantyzacji równym 24. Dekoder, przy częstotliwości 100 MHz, nie może poświęcić więcej niż 490 taktów sygnału zegarowego na pojedynczy makroblok, aby być w stanie obsłużyć sekwencję 1080i. Zgodnie więc z otrzymanymi wynikami, można stwierdzić że przedstawiona architektura pozwala na obsługę HDTV.

Synteza została przeprowadzona w oparciu o technologię TSMC 0,13  $\mu\text{m}$ . W porównaniu do dostępnych publikacji, (tabela 3), prezentowana architektura wykorzystuje bardzo mało zasobów sprzętowych, przy jednoczesnym zapewnieniu odpowiedniej szybkości działania.

Tab. 2. Liczba taktów sygnału zegarowego, niezbędna do zdekodowania makrobloku

Tab. 2. Number of cycles necessary to decode single macroblock

sekwencja	carphone	football	mobile	średnia
struktura ramek	IPPP	172	298	478
	IBBPBB	140	291	395

Tab. 3. Porównanie z innymi pracami

Tab. 3. Comparison with others Works

Praca	technologia	częstotliwość	liczba ramek	liczba symboli na takt sygnału zegarowego
proponowana	TSMC 0,13 $\mu\text{m}$	200 MHz	24308	0,99
[3]	UMC 90 nm	222 MHz	82445	1,96
[4]	0,18 $\mu\text{m}$	200 MHz	28956	1,27
[5]	TSMC 0,13 $\mu\text{m}$	137 MHz	40762	0,8

Podsumowując, prezentowane rozwiązanie zapewnia niskie zużycie zasobów sprzętowych przy zachowaniu wysokiej efektywności działania. Skrócenie ścieżki krytycznej oraz rozbitcie pętli sprzężenia zwrotnego pozwala na obsługę sekwencji nawet o rozdzielczościach rzędu 1080i. Jednocześnie opracowana architektura jest w pełni zgodna z profilem High standardu H.264/AVC, umożliwiając obsługę trybu MBAFF, czy dowolnego sposobu podpróbki chrominancji.

*Praca jest elementem projektu "Zintegrowany mobilny system wspomagający działania antyterrorystyczne i antykryzysowe - PROTEUS", realizowanego dzięki wsparciu finansowemu Europejskiego Funduszu Rozwoju Regionalnego w ramach 1. osi priorytetowej Programu Operacyjnego Innowacyjna Gospodarka, 2007-2013, Poddziałanie 1.1.2, projekt numer PO-IG.01.02.01-00-014/08-00.*

## 7. Literatura

- [1] ITU-T Recommendation H.264 and ISO/IEC 14496-10 MPEG-4 Part 10, Advanced Video Coding (AVC), Listopad 2007.
- [2] Eeckhaut H., Christiaens M., Strooband D., Nollet V.: Optimizing the critical loop in the H.264/AVC CABAC decoder. IEEE International Conference on Field Programmable Technology, strony 113-118, Grudzień 2006.
- [3] Lin P. C., Chuang T. D., Chen L. G.: A branch selection multi-symbol high throughput CABAC decoder architecture for H.264/AVC. IEEE International Symposium on Circuits and Systems, strony 365-368, Czerwiec 2009.
- [4] Shi B., Zheng W., Lee H. S., Li D. X., Zhang M.: Pipelined architecture design of H.264/AVC CABAC real-time decoding. IEEE International Conference on Circuits and Systems for Communications, strony 492-496, Maj 2008.
- [5] Chen J. W., Lin Y. L.: A high-performance hardwired CABAC decoder. IEEE International Conference on Acoustics, Speech and Signal Processing, wol. 2, strony 37-40, Kwiecień 2007.