POLITECHNIKA WARSZAWSKA
Wydział Elektroniki i Technik Informacyjnych
Instytut Telekomunikacji

PRACA DYPLOMOWA MAGISTERSKA
(TNRoman 18 kapitaliki wyśrodkowane)

Piotr Studencki
(TNRoman 14 normal wyśrodkowane)

Tytuł bardzo ważnej pracy dyplomowej v17012008
(TNRoman 14 bold wyśrodkowane)

Jeżeli DR, PROF to piszemy:

Praca wykonana pod kierunkiem
dra inż. Jana Nizinnego

Jeżeli MGR to piszemy:

Opiekun naukowy
mgr inż. Stefan Pasibrzuch

……
Ocena pracy

……
Podpis Przewodniczącego Komisji

Warszawa, 2012

Implementation of very important things (Title font TNRoman 14 Bold)

Abstract (Font TNRoman 14 Bold)

Implementation of …. (Font TNRoman 12 Normal).
This thesis includes the design and testing procedure of …. .

Streszczenie

ABC jest standardem kodowania mowy przy 1,6 kbit/s. Wykorzystuje on całkiem starą zasadę. Podstawową cechą wyróżniająca go spośród innych projektów jest wspaniała jakość transmisji. Wykorzystywany jest on do łączności między zielonymi pojazdami.

Praca zawiera projekt oraz opis próby realizacji kodeka w układzie reprogramowalnym FPGA wraz z procedurą sprawdzania poprawności algorytmicznej. Dokument ten składa się z trzech części. W pierwszej omówiono podstawy kodowania (rozdziały 3, 4 i 5). Druga część zawiera opis funkcjonalny standardu 1800-123 z podziałem na bloki funkcjonalne (rozdział 6). Trzecia przestawia sposób realizacji oraz testowania kodeka w układzie (rozdziały 7 i 8).

Życiorys

Urodziłem się 31 lutego 1992 roku w Pułtusku. W 1989 roku rozpocząłem naukę w X Liceum Ogólnokształcącym im. „C.K. Dezertera”, gdzie uczęszczałem do klasy o profilu autorskim. Po uzyskaniu świadectwa dojrzałości w 2001 roku, rozpocząłem studia na Politechnice Warszawskiej na Wydziale Elektroniki i Technik Informacyjnych.

Piotr Studencki

Spis treści

1. Wstęp[1]

Uwaga: Dokument ten nie jest wzorcem merytorycznym, ma jedynie pomóc w poprawnej edycji pracy dyplomowej. Reguły przedstawione poniżej zostały zebrane na podstawie doświadczenia autora i zgodnie z najlepszymi intencjami pozwalają na przygotowanie pracy dyplomowej odpowiadającej ogólnie przyjętym standardom.

W pracy stosujemy stronę bierną oraz piszemy w 3 osobie np. „W rozdziale autor przedstawił" lub „W rozdziale przedstawiono". Przygotowywany tekst jest dokumentem, staramy się zatem unikać beletrystyki i żargonu.

Cały tekst pisany jest czcionką Times New Roman 12 pkt., wyjustowany, z akapitami o wcięciu **1,25** cm, interlinia ustawiona na **wielokrotnie** o wartości **1,3**. Brak dodatkowych odstępów między akapitami. Tytuł rozdziału pisany czcionką TNRoman 14 bold. Cały tekst ma marginesy 2 cm, natomiast lewy ma 2,5 cm.

Bardzo szybki rozwój sieci teleinformatycznych w ostatnich latach stwarza nowe możliwości komunikowania się.

Obecnie projektowane oraz konstruowane urządzenia końcowe oraz bramy VoIP realizowane są głównie z wykorzystaniem procesorów DSP. Szczególny nacisk jak pokazano w [1, 2, 7].

Dotychczasowe próby realizacji kodeków mowy w FPGA (ang. *Field Programmable Gate Array*) ograniczały się. Pierwsze wystąpienie akronimu rozwijamy w nawiasie czcionką pochyłą, kolejne wystąpienia tego akronimu już nie rozwijamy. Jednakże zauważalny w ostatnich latach ogromny rozwój układów reprogramowalnych stwarza możliwość realizacji pełnego kodeka mowy. Układ ten musiałby spełniać wymagania jakościowe oraz wnosić akceptowalnie małe opóźnienie ze względu na działanie w czasie rzeczywistym. Próba realizacji kodera mowy w FPGA zakończona sukcesem otwierałaby nowe możliwości w projektowaniu urządzeń VoIP m.in. umieszczenie części sterującej oraz przetwarzającej mowę w jednym układzie, łatwiejszą synchronizację, minimalizację rozmiarów fizycznych oraz stwarzałaby możliwość realizacji całego urządzenia w technologii ASIC.

[1] Dokument powstał na podstawie opracowania dr inż. Pawła Tomaszewicza. Oryginał jest dostępny pod adresem: „http://tomaszewicz.zpt.tele.pw.edu.pl/node/206"

2. Cel pracy

Założeniem pracy jest projekt oraz realizacja kodera i dekodera 1800-123 w strukturze układu reprogramowalnego wykonanego w technologii FPGA.

Na etapy projektu składać się będzie opracowanie schematów blokowych poszczególnych elementów układu oraz opis ich funkcjonalności.

W części realizacyjnej zostanie podjęta próba wykonania oraz optymalizacji działania kodeka na układzie reprogramowalnym. Do tego celu zostanie wykorzystane środowisko programistyczne oraz język opisu sprzętu SuperHDL. W celu weryfikacji poprawnego działania zrealizowanych urządzeń zostaną one poddane testom z wykorzystaniem wektorów testowych ITU-T.

3. Poziom pierwszy

3.1. Poziom drugi

Przykładowy tekst. Używany powszechnie w telefonii cyfrowej format PCM powstaje przez filtrację sygnału mowy do pasma o szerokości 3,1 kHz w zakresie od 300 do 3400 Hz. Jest to wystarczające pasmo dla mowy ludzkiej, gdyż ucho ludzkie wykazuje swoją największą czułość właśnie w tym zakresie. Przy próbkowaniu takiego sygnały z częstotliwością 8 kHz oraz wartościowości 8 bitów na próbkę daje to strumień o przepływności 64 kbit/s. Taki format sygnału okazuje się jednak nieodpowiedni do wykorzystania w przesyłaniu mowy z wykorzystaniem sieci IP.

Podstawową przeszkodą jest stosunkowo duża przepływność binarna. Dodatkowymi czynnikami dyskwalifikującymi ten format są duże wymagania pamięciowe urządzeń przetwarzających. Dlatego też, aby wyeliminować te niedoskonałości formatu PCM stosuje się odpowiednie kodowanie. Polega ono głównie na znalezieniu redundancji w sygnale. Tak przetworzony sygnał mowy nie będzie idealnie odpowiadać sygnałowi oryginalnemu – głównie ze względu na procesy próbkowania, kwantyzacji oraz kodowania. Jednakże odchylenia te są na tyle małe, że nie powodują dużej różnicy w subiektywnym odbiorze.

3.2. Poziom drugi

3.2.1. Poziom trzeci

Poziom drugi i trzeci piszemy na tej samej stronie (nie zaczynamy od nowej strony). Jeżeli jest potrzeba to wprowadzamy zamiast poziomu trzeciego, wyróżnienie tekstu.

Zamiast poziomu czwartego tytuł akapitu

I tutaj tekst poziomu czwartego.

4. Liniowa predykcja

4.1. Wstęp

Metoda liniowej predykcji polega na wyznaczeniu przybliżonej wartości próbki o indeksie n na podstawie M poprzednich próbek. Ma szczególne zastosowanie w procesie kodowania mowy ze względu na zjawisko silnej korelacji sąsiadujących ze sobą próbek w sygnale mowy. Wszystkie zmienne i symbole piszemy czcionką pochyłą.

Poniżej użyty jest wzór, wyśrodkowany, numer wyrównany do prawej, numeracja jest (X.Y) gdzie X oznacz numer rozdziału głównego, Y kolejny numer w ramach tego rozdziału:

$$\tilde{x}[n] = \sum_{i=1}^{M} a_i x[n-i] \tag{4.1}$$

gdzie:

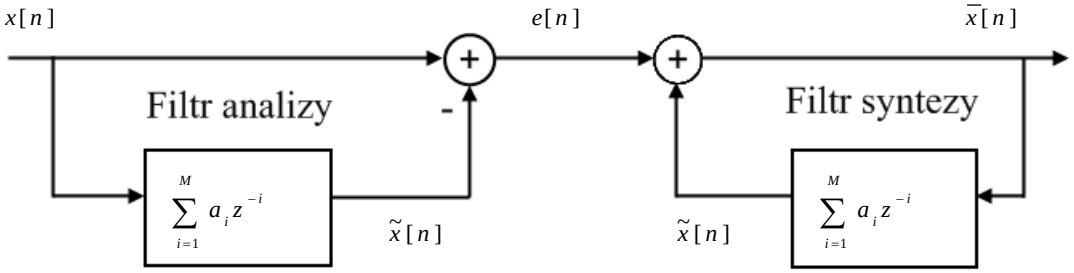
M – rząd filtru

$x[n]$ – wejściowy sygnał

$\tilde{x}[n]$ – sygnał predykcji

a_i – współczynniki filtru predykcji

to problem liniowej predykcji sprowadza się do prawidłowego określenia współczynników a_i. W tym celu należy zminimalizować błąd średniokwadratowy pomiędzy sygnałami wejściowym $x[n]$ i predykcji $\tilde{x}[n]$. Rysunek 4.1 przedstawia filtr analizy oraz filtr syntezy. Podpis jest pod rysunkiem, wyśrodkowany, czcionka pochylona, numer X.Y gdzie X to numer głównego rozdziału a Y to kolejny numer rysunku w ramach głównego rozdziału. Cały rysunek wyśrodkowany.



Rys. 4.1. Filtr analizy i filtr syntezy

Najczęściej stosowanym algorytmem w kodowaniu mowy pozwalającym na wyznaczenie optymalnych współczynników predykcji jest procedura Levinsona – Durbina (szczegółowo opisana w rozdziale 6.3.5).

7.7. Parametry zrealizowanych urządzeń

Tabela 7.1 przedstawia podstawowe parametry zrealizowanych urządzeń. Opis tabeli nad tabelą, numeracja jak dla rysunków, czcionka prosta, tekst wyśrodkowany. Cała tabela wyśrodkowana.

Tabela 7.1. Parametry zrealizowanych urządzeń

	Moduł podstawowy		**Moduł testowy**	
	Koder	Dekoder	Koder	**Dekoder**
ALUT[25]	30 058 (62%)	18 596 (38%)	30 282 (63%)	**18 888 (39%)**
Rejestry	21 144 (44%)	13 307 (28%)	21 207 (44%)	**13 342 (28%)**
Pamięć	97 488 bitów (4%)	45 088 bitów (2%)	135 376 bitów (5%)	**115 744 bitów (5%)**
Bloki DSP	94 (33%)	42 (19%)	94 (33%)	**42 (19%)**
f_{max}	62,58 MHz	67,39 MHz	66,24 MHz	**68,31 MHz**
t_{min}	**15,979 ns**	**14,838 ns**	**15,096 ns**	**14,639 ns**

7.8. Cechy i właściwości rozwiązania

Głównymi zaletami rozwiązania są: (a) szybkość wykonywania operacji, (b) maksymalne wykorzystanie dostępnych zasobów, (c) niewielki koszt energetyczny. W przypadku list składających się z niewielkiej liczby, mało rozbudowanych pozycji, można je umieścić kolejno w akapicie, po dwukropku, wyróżniając pozycje małą literą ujętą w nawiasy.

Wśród cech opisywanego oprogramowania można wyróżnić: cechy zaletami rozwiązania są:

- Możliwość wprowadzania zmian bezpośrednio z poziomu graficznego interfejsu użytkownika lub za pomocą plików konfiguracyjnych.
- System automatycznego wykonywania kopii zapasowych. Został on rozbudowany o funkcje odzyskiwania danych ze sprawdzeniem ich integralności.

W przypadku list składających się z rozbudowanych pozycji, korzystamy z punktorów w postaci kropki. Unikamy numerowania list ze względu na wprowadzenie niespójności z numeracja rozdziałów. Pozycje listy są w tym przypadku zdaniami, zaczynają się zatem dużą literą i kończą kropką.

7.9. Algorytm działania

Na poniższym wydruku przedstawiono fragment rozwiązania dotyczącego głównej pętli decyzyjnej w module zliczania danych. Fragmenty kodu programu powinny zostać zapisane czcionką Courier New 10pt. Nie numerujemy w/w, unikamy nieczytelnych i zbyt długich fragmentów. Obowiązuje zasada - im krócej tym lepiej. W praktyce, kod przekraczający stronę jest już nieczytelny (sic!). Należy pamiętać o rzetelnym skomentowaniu kodu w tekście. Niedopuszczalny jest komentarz skopiowany wraz z wydrukiem. Należy rozważyć możliwość zastosowania tzw. pseudo-kodu, co zawsze znacznie poprawia czytelność tekstu.

```
dates_count = ${#datetab[*]}
echo ${#datetab[@]}
((dates_count = ${#datetab[@]} / 3))
echo dates_count
```

10. Bibliografia

[1] Tadeusiewicz R., Korohoda P.: Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków 1997

[2] Wiatr K.: Sprzętowe implementacje algorytmów przetwarzania obrazów w systemach wizyjnych czasu rzeczywistego, Uczelniane Wydawnictwa Naukowo – Dydaktyczne AGH, Kraków 2002

[3] Woźnicki J.: Podstawowe techniki w dziedzinie przetwarzania obrazu, Wydawnictwa Komunikacji i Łączności, Warszawa 1996

[4] Watkins C. D., Sadun A., Marenka S.: Nowoczesne metody przetwarzania obrazu, Wydawnictwa Naukowo – Techniczne, Warszawa 1995

[5] Fiok A.: Telewizja – podstawy ogólne, Wydawnictwa Komunikacji i Łączności, Warszawa 1991

[6] Vega-Rodríguez M., Sánchez-Pérez J.: Gómez-Pulido J., AN FPGA-BASED IMPLEMENTATION FOR MEDIAN FILTER MEETING THE REAL-TIME REQUIREMENTS OF AUTOMATED VISUAL INSPECTION SYSTEMS, Proceedings of the 10th Mediterranean Conference on Control and Automation – MED2002, http://med.ee.nd.edu/MED10/pdf/329.pdf, stan na 14.08.2007

[7] Fisher B., Perkins S., Walker A., Wolfart E.: Hypermedia Image Processing Reference, Department of Artificial Intelligence, University of Edinburgh, Edinburgh 1994, http://www.cee.hw.ac.uk/hipr/html/hipr_top.html, stan na 15.08.2007

[8] Nain N., Laxmi V., Jain A. K., Agarwal R.: MORPHOLOGICAL EDGE DETECTION AND CORNER DETECTION ALGORITHM USING CHAIN-ENCODING, Malaviya National Institute of Technology, Jaipur 2006, http://ww1.ucmss.com/books/LFS/CSREA2006/IPC4042.pdf, stan na 12.09.2007

Ponieważ prawidłowe sformatowanie bibliografii, czyli takie, które zapewni zgodność ze standardem, jest pracochłonne, można posłużyć się dedykowanym oprogramowaniem. Przykładem w/w jest program na licencji *freeware* o nazwie BiblioExpress, który można pobrać ze strony: „http://www.softpedia.com/get/Others/Home-Education/BiblioExpress.shtml". Aplikacja tworzy automatycznie listę pozycji literaturowych w formacie MS-Word.

Dodatek A
Opis zawartości dołączonej płyty CD

Dołączona płyta CD zawiera:

- Zestaw testowy przeznaczony dla kodera/dekodera
  - *.v – pliki poszczególnych modułów kodera/dekodera (Verilog)
  - *.mif – pliki inicjacyjne pamięci ROM modułów kodera/dekodera
  - ldcelp_encoder_tester.qar – archiwum projektu Quartus II zawierające zestaw testowy kodera
  - ldcelp_decoder_tester.qar – archiwum projektu Quartus II zawierające zestaw testowy dekodera
  - EncoderUSBReader.java – moduł programowy testera kodera odczytujący dane z portu USB i zapisujący je do pliku (Java)
  - DecoderUSBReader.java – moduł programowy testera dekodera odczytujący dane z portu USB i zapisujący je
  - do pliku (Java)
  - jd2xx.jar – biblioteka procedur komunikacji z układem FT245BM (Java)
- Zestaw wektorów testowych wraz ze stanami wewnętrznymi
- Elektroniczną wersję pracy dyplomowej magisterskiej