POLSKA AKADEMIA NAUK
KOMITET ELEKTRONIKI I TELEKOMUNIKACJI

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI

ELECTRONICS AND TELECOMMUNICATIONS QUARTERLY

TOM 52 — ZESZYT 4

WARSZAWA 2006

R
Kwar

1. P
W
Pocz
w jęz
przyg
zakw
ich w
Każd
(np. c
wzor
angie
Artyk

2. P
P
Wpro
artyk
przez
wysc
przez
dział
Kwa
tykuł
dużą
w sp
W p
publ
publ
W u
decy
za p
znac
opłat
nych
War
Nr
i Te

WAŻNY KOMUNIKAT DLA AUTORÓW

Rada Redakcyjna na swoim posiedzeniu w dniu 18 stycznia 2006 r. zaleciła redakcji Kwartalnika Elektroniki i Telekomunikacji zrealizowanie następujących ustaleń:

1. PUBLIKOWANIE NADSYŁANYCH ARTYKUŁÓW WYŁĄCZNIE W JĘZYKU ANGIELSKIM

Począwszy od zeszytu 1'2007 wszystkie artykuły będą publikowane w Kwartalniku w języku angielskim, a tym samym artykuły nadsyłane już obecnie do redakcji muszą być przygotowane przez autorów w tym języku. Artykuły już nadesłane do redakcji i wstępnie zakwalifikowane do publikacji w roku 2007 będą zwrócone autorom w celu opracowania ich w wersji angielskiej.

Każdy artykuł przygotowany w języku angielskim powinien być uzupełniony obszernym (np. dwustronicowym) streszczeniem w języku polskim z przytoczeniem najistotniejsszych wzorów oraz powołaniem się na ważne tabele i rysunki. Ponadto należy do tekstu angielskiego dołączyć polskie odpowiedniki tytułów tabel i podpisów pod rysunkami. Artykuły te będą recenzowane, a także weryfikowane pod względem językowym.

2. POKRYWANIE KOSZTÓW WYDAWNICZYCH PUBLIKACJI PRZEZ AUTORÓW

Wprowadza się, począwszy od zeszytu 1'2007, zasadę odpłatności za publikowanie artykułu w Kwartalniku, przy założeniu działalności redakcji non profit, to jest pokrywania przez autorów lub instytucje ich zatrudniające, kosztów wydawniczych w podstawowej wysokości 760 zł za arkusz wydawniczy. Uzyskana w ten sposób kwota będzie przeznaczana na uzupełnienie ograniczonych środków otrzymywanych z PAN na działalność wydawniczą, a w szczególności na zwiększenie objętości kolejnych zeszytów Kwartalnika oraz na weryfikację językową publikowanych w języku angielskim artykułów. Zwiększenie objętości poszczególnych zeszytów jest konieczne ze względu na dużą liczbę nadsyłanych artykułów, których publikowanie w obecnej objętości zeszytów, w sposób nie akceptowalny wydłuża czas ich ukazania się w druku.

W przypadku, wyrażonej w formie pisemnej, prośby autorów o przyspieszenie ich publikacji w stosunku do kolejności wynikającej z terminu ich nadesłania, opłata za publikację artykułu wynosić będzie 1500 zł za arkusz wydawniczy.

W uzasadnionych przypadkach, przedstawionych pisemnie, redakcja może podjąć decyzję częściowego lub całkowitego zwolnienia autorów z podstawowej odpłatności za publikację. Redakcja Kwartalnika wyraża nadzieję, że autorzy artykułów przeznaczonych do opublikowania w roku 2006, podejmą starania wniesienia powyższych opłat już w roku bieżącym. Dotyczy to w szczególności opracowań realizowanych w ramach „grantów". Wpłat należy dokonywać przelewem bankowym na konto Warszawskiej Drukarni Naukowej PAN, BPH PBK S.A. w Krakowie VIII O/Warszawa Nr 1060 0076 0000 4010 3000 0 510 z **dopiskiem dla Kwartalnika Elektroniki i Telekomunikacji.**

REDAKCJA

RADA REDAKCYJNA

Przewodniczący
Prof. dr hab. inż. STEFAN HAHN
czł. rzecz. PAN

Członkowie
prof. dr hab. inż. DANIEL JÓZEF BEM — czł. koresp. PAN, prof. dr hab. inż. MICHAŁ BIAŁKO — czł. rzecz. PAN, prof. dr hab. inż. MAREK DOMAŃSKI, prof. dr hab. inż. ANDRZEJ HAŁAŚ, prof. dr hab. inż. JÓZEF MODELSKI, prof. dr inż. JERZY OSIOWSKI, prof. dr hab. inż. EDWARD SĘDEK, prof. dr hab. inż. MICHAŁ TADEUSIEWICZ, prof. dr hab. inż. WIESŁAW WOLIŃSKI — czł. koresp. PAN, prof. dr inż. MARIAN ZIENTALSKI

REDAKCJA
Redaktor Naczelny
prof. dr hab. inż. WIESŁAW WOLIŃSKI

Zastępca Redaktora Naczelnego
doc. dr inż. KRYSTYN PLEWKO

Sekretarz Odpowiedzialny
mgr ELŻBIETA SZCZEPANIAK

ADRES REDAKCJI
00-665 Warszawa, ul. Nowowiejska 15/19 Politechnika, pok. 470
Instytut Telekomunikacji, Gmach im. prof. JANUSZA GROSZKOWSKIEGO

Dyżury Redakcji: środa i piątki, godz. 14–16
tel. (022) 234 77 37

Telefony domowe: Redaktora Naczelnego: 812 17 65
Zast. Red. Naczelnego: 826 83 41
Sekretarza Odpowiedzialnego: 251 02 01

Ark. wyd. 20,0 Ark. druk. 16,5	Podpisano do druku w październiku 2006 r.
Papier offset. kl. III 80 g. B-1	Druk ukończono w listopadzie 2006 r.

Skład, druk i oprawa: Warszawska Drukarnia Naukowa PAN
00-656 Warszawa, ul. Śniadeckich 8
Tel./fax: 628-87-77

Szanowni Autorzy

„Kwartalnik Elektroniki i Telekomunikacji" — Electronics and Telecommunications Quartery jest kontynuatorem tradycji powstałego 52 lata temu kwartalnika pt. „Rozprawy Elektrotechniczne".

Kwartalnik jest czasopismem Komitetu Elektroniki i Telekomunikacji Polskij Akademii Nauk. Wydawany jest przez Warszawską Drukarnię Naukową PAN. Kwartalnik jest czasopismem naukowym, na którego łamach są publikowane artykuły i komunikaty prezentujące wyniki oryginalnych prac teoretycznych i doświadczalnych, a także przeglądowych. Związane są one z szeroko rozumianymi dziedzinami współczesnej elektroniki, telekomunikacji, mikroelektroniki, oproelektroniki, radiotechniki i elektroniki medycznej.

Autorami publikacji są wybitni naukowcy, znani specjaliści o wieloletnim doświadczeniu, a także młodzi badacze — głównie doktoranci.

Artykuły charakteryzują się oryginalnym ujęciem zagadnienia, interesującymi wynikami badań, krytyczną oceną teorii lub metod, omówieniem aktualnego stanu, lub postępu danej gałęzi techniki oraz omówieniem perspektyw rozwojowych. Sposób pisania matematycznej części artykułów zgodny jest z wytycznymi IEC (International Electronics Commision) oraz ISO (International Organization of Standarization).

Wszystkie publikowane w Kwartalniku artykuły są recenzowane przez znanych krajowych specjalistów, co zapewnia że publikacje te są uznawane jako autorski dorobek naukowy. Opublikowane w kwartalniku wyników prac naukowych zrealizowanych w ramach „GRANTów" Komitetu Badań Naukowych spełnia więc jeden z wymogów stawianych tym pracom.

Czasopismo dociera do wszystkich zajmujących się elektroniką i telekomunikacją krajowych ośrodków naukowych oraz technicznych, a także szeregu instytucji zagranicznych. Jest ponadto prenumerowane przez liczne grono specjalistów i biblioteki.

Każdy Autor otrzymuje bezpłatnie 20 egzemplarzy nadbitek swojego artykułu, co ułatwia przesłanie go do indywidualnych wybranych przez Autora osób i instytucji w kraju lub za granicą. Ułatwia to dodatkowo fakt, że w Kwartalniku są publikowane artykuły w języku angielskim.

Nadesłane do redakcji artykuły są publikowane w terminie około pół roku, w przypadku sprawnej współpracy Autora z Redakcją. Wytyczne dla Autorów dotyczące formy publikacji są zamieszczone w zeszytach Kwartalnika, można je także otrzymać w siedzibie Redakcji.

Artykuły można dostarczać osobiście, lub pocztą pod adresem zamieszczanym na stronie redakcyjnej w każdym zeszycie.

Redakcja

W. P
E. K
D. K
M. C
T. M

R. J.
K. M
A. D

T. C

A Dy

Z. N
Z. N

Infor
Rocz
Lista

W. P
E. K
D. K
M. C
T. M

R.J.
K. M
A. D

T. C
A Dy

Z. N
Z. N

Infor
Year
The

SPIS TREŚCI

W. Pogribny, M. Drechny: Różnicowa DCT w zagadnieniach przetwarzania sygnałów 493
E. Kornatowski: Uniwersalna miara oceny jakości obrazów cyfrowych 511
D. Kania: Dekompozycja wierszowa w syntezie logicznej przeznaczonej dla struktur matrycowych . 521
M. Gajer: Zastosowanie binaryzacji okresów zadań w systemach wieloprocesorowych 545
T. Molik, A. Bartoszewicz: Dyskretne algorytmy sterowania szybkością nadawania danych w połączeniowych sieciach teleinformatycznych 563
R. J. Katulski, J. Sadowski: Impulsowy system ultraszerokopasmowy (UWB) 591
K. M. Noga: Statystyki drugiego rzędu kanału Rayleigha oraz Nakagamiego 617
A. Dąbrowska, K. Wiatr: Modyfikacja algorytmów transformacji FDCT dla potrzeb kompresji obrazów implementowanej w układach FPGA 629
T. Czarski: Modelowanie systemu sterowania nadprzewodzącej wnęki rezonansowej akceleratora liniowego 643
A Dybko, R. Graczyk, K. T. Poźniak, R. S. Romaniuk: Modularny system fotoniczny z programowalną warstwą sterowania i komunikacji w układzie FPGA 655
Z. Nagórny, A. Kos: Projektowanie topografii systemów VLSI. Cz. 3. Metody analityczne 669
Z. Nagórny, A. Kos: Projektowanie topografii systemów VLSI. Cz. 4. Symulowane wyżarzanie, sieci neuronowe 697
Informacje dla Autorów (w jęz. polskim) 729
Roczny spis terści 735
Lista recenzentów 741

CONTENTS

W. Pogribny, M. Drechny: Differential DCT in signal processing issues 493
E. Kornatowski: A universal digital image quality index 511
D. Kania: Row decomposition in logic synthesis for CPLDS 521
M. Gajer: The application of binarisation of tasks periods in multiprocessor systems 545
T. Molik, A. Bartoszewicz: Discrete time flow control algorithms for connection oriented communication networks 563
R.J. Katulski, J. Sadowski: Ultra wide band (UWB) impulse radio 591
K. M. Noga: Csecond order statistics for the Rayleigh and Nakagami channel 617
A. Dąbrowska, K. Wiatr: Transform FDCT algorithm modified for image compression implemented in FPGAS structures 629
T. Czarski: Control system modeling for superconducting accelerator 643
A Dybko, R. Graczyk, K. T. Poźniak, R. S. Romaniuk: FPGA based modular photonic system with programmable control and communication layer 655
Z. Nagórny, A. Kos: The sedign of the VLSI circuit layout. Part 3. Analytical methods 669
Z. Nagórny, A. Kos: The design of the VLSI circuit layout. Part 4. Simulated annealing, neural networks 697
Information for the Authors (w jęz. ang.) 733
Year contents of Electronics and Telecommunications Quarterly 735
The List of the reviewers 741

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 493–509

RÓŻNICOWA DCT W ZAGADNIENIACH PRZETWARZANIA SYGNAŁÓW

WŁODZIMIERZ POGRIBNY, MARCIN DRECHNY

Akademia Techniczno-Rolnicza w Bydgoszczy
Wydział Telekomunikacji i Elektrotechniki
al. Prof. S. Kaliskiego 7, 85-796 Bydgoszcz
pohry@atr.bydgoszcz.pl,
Marcin.Drechny@atr.bydgoszcz.pl

Otrzymano 2005.10.20
Autoryzowano 2005.02.16

Dyskretna transformata kosinusowa (DCT) jest szeroko stosowana do analizy i kompresji sygnałów. Klasyczne algorytmy DCT bazują na obliczeniach wykonywanych w wielobitowym formacie PCM (Pulse Code Modulation). Ograniczeniem zastosowania formatu PCM w DCT jest użycie wielobitowych operacji mnożenia, co często utrudnia jego wykorzystanie w systemach czasu rzeczywistego. Z drugiej strony, użycie niektórych niskobitowych rodzajów modulacji delta (DM) zapewnia taką samą dokładność obliczeń jak w przypadku użycia wielobitowego formatu PCM, lecz takie podejście nie zostało wystarczająco zbadane.

Celem pracy było opracowanie i zbadanie efektywnych różnicowych algorytmów i struktur procesorów DCT pracujących w czasie rzeczywistym.

W pracy tej przedstawiono nowe szybkie różnicowe algorytmy DCT. Szczególną uwagę skupiono na możliwości wyeliminowania w zaproponowanych algorytmach czasochłonnych operacji mnożenia i zastąpienia ich szybkimi operacjami bitowego przesunięcia SHIFT. Zastosowanie przesunięć umożliwiają kroki kwantowania będące potęgą liczby 2.

Do zbadania dokładności zaproponowanych algorytmów wykorzystano sygnały szumu różowego oraz mowy. Zaproponowane zostały szybkie i efektywne struktury procesorów specjalizowanych realizujące opracowane algorytmy. Wyniki symulacji komputerowych potwierdziły efektywność opracowanych podejść.

Słowa kluczowe: dyskretna transformata kosinusowa, modulacja delta, DPCM

1. WPROWADZENIE

Sygnały używane w systemach teleinformatycznych poddawane są analizie i przetwarzaniu w dziedzinach czasu i częstotliwości w celu uzyskania zawartych w nich informacji jak i w celu kompresji danych. Do transformacji sygnału z jednej dziedziny do drugiej najczęściej używa się dyskretnej transformaty Fouriera (DFT). Przez wiele lat udoskonalano algorytmy obliczeniowe transformaty DFT [1], czego wynikiem są jej szybkie odmiany FFT [2] (Fast Fourier Transform). W roku 1974 Rao, Ahmed i Natarajan opublikowali pracę [3], w której zaproponowali praktyczne podejście do użycia jednowymiarowej dyskretnej transformaty kosinusowej w przetwarzaniu sygnału do dziedziny częstotliwości w celu otrzymania odpowiedniego widma. Ponieważ DCT jest pochodną DFT przy założeniu, że sygnał kodowany poddawany przekształceniu jest funkcją parzystą z symetrią pierwszego rodzaju [1], [4] do wyznaczenia składowych widma DCT można użyć zmodyfikowanych algorytmów FFT [5]. Specyficzne własności DCT takie jak:

— obliczenia wykonywane wyłącznie w dziedzinie liczb rzeczywistych,
— większa dokładność przetwarzania sygnału niż w DFT biorąc pod uwagę wpływ efektów granicznych,
— skoncentrowanie w zdecydowanie większym stopniu niż w DFT widma sygnału na początkowych składowych (współczynnikach) transformaty pozwalające na ich dalszą kompresję,
— użycie tych samych funkcji bazowych do DCT jak i odwrotnej DCT (IDCT),
— przesunięcie fazowe wszystkich składowych częstotliwościowych o kącie zerowym,

powodują, że transformata ta jest szeroko stosowana w systemach teleinformatycznych [5]. Główne zastosowanie DCT to kompresja [6] sygnałów jedno i dwuwymiarowych, na której opierają się standardy JPEG (Joint Photographic Experts Group), MPEG (Moving Picture Experts Group) oraz MP3 (MPEG 1 Layer 3).

Zwykle wyznaczenie współczynników transformaty DCT wykonywane jest w wielobitowym formacie PCM. Zastosowanie tej modulacji zapewnia dużą dokładność obliczeń oraz szerokie pasmo przetwarzania. Użycie formatu PCM wymaga wykonywania wielobitowych operacji w szczególności mnożenia, co zmniejsza ekonomiczność i szybkość przetwarzania układów. Dlatego też, wykorzystanie wielobitowego formatu PCM do przetwarzania sygnałów szerokopasmowych w czasie rzeczywistym jest ograniczone.

Alternatywą dla zastosowania formatu PCM w DCT są formaty modulacji delta takie jak DPCM (Differential Pulse Code Modulation) czy ADM (Adaptive Delta Modulation). Przy odpowiednio dobranych parametrach DM [7], [8] można uzyskać dokładność obliczeń taką jak dla formatu PCM a także przy zastosowaniu odpowiednich algorytmów obliczeniowych zwiększyć szybkość działania lub uprościć obliczenia. Jednym z pierwszych sposobów uproszczenia operacji matematycznych dla filtracji cyfrowej za pomocą formatu różnicowego ∇PCM z wykorzystaniem rekursji jest praca z 1975 roku [9]. Autorzy przedstawili w niej koncepcję uproszczenia operacji przez

zastosowanie różnic otrzymywanych na podstawie odejmowania od bieżących próbek PCM różnic poprzednich próbek skalowanych z uwzględnieniem współczynników rekursji. Wymienione różnice zostały przyrównane do wartości, które są potęgą liczby 2 z całkowitymi wykładnikami. To pozwoliło na uproszczenie operacji mnożenia. Skutkuje to zwiększeniem szybkości działania natomiast dokładność takiej filtracji nie jest wystarczająca.

W niniejszej pracy zaproponowano użycie różnicowych formatów DM-PCM do obliczeń DCT w szczególności DPCM-PCM, ADM-PCM oraz zmodyfikowanego formatu DPCM (MDPCM-PCM), w którym kroki kwantowania są potęgami liczby 2, co eliminuje wszystkie czasochłonne operacje mnożenia i zastępuje je szybkimi operacjami przesunięcia bitowego SHIFT.

Celem pracy było opracowanie i zbadanie efektywnych różnicowych algorytmów i struktur procesorów DCT pracujących w czasie rzeczywistym.

2. ALGORYTMY DCT

2.1. KLASYCZNY ALGORYTM DCT W FORMACIE PCM-PCM

Dyskretna transformata DCT w postaci klasycznej [3] i [5] przedstawiona jest za pomocą poniższego wyrażenia:

$$X(m) = k_m \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi m (2n+1)}{2N}\right) \tag{1}$$

w którym, $X(m)$ — składowa częstotliwościowa widma o numerze m, N — ilość danych wejściowych, $m, n = \overline{0, N-1}$, natomiast k_m opisywane jest wyrażeniem:

$$k_m = \begin{cases} \dfrac{\sqrt{2}}{N}, dla\ m = 0 \\ \dfrac{2}{N}, dla\ m = \overline{1, N-1} \end{cases}$$

Dane wejściowe, których ilość N jest równa liczbie M wyznaczonych współczynników DCT są przedstawione w wielobitowym formacie PCM-PCM. Taki zapis formatu oznacza, że zarówno pierwsza próbka danych wejściowych jak i wszystkie następne są zakodowane w tym samym kodzie PCM. Algorytm w tej postaci wymaga $N \cdot M$ (czyli N^2 dla $M = N$) wielobitowych operacji mnożenia nie uwzględniając mnożenia czynnika sum przez współczynnik k_m i utworzenie czynników cos() oraz $N \cdot M$ (N^2) sumowań. Do zastosowania różnic w DCT należy odpowiednio przekształcić wyrażenie (1).

2.2. NOWY RÓŻNICOWY ALGORYTM DCT

Do utworzenia różnicowego algorytmu skorzystano z różnic wzrastających ∇PCM opisanych jako:

$$\nabla z_n = z_n - z_{n-1} \tag{2}$$

gdzie, ∇z_n — różnica wzrastająca n w formacie ∇PCM, z_n — próbka n sygnału w formacie PCM, z_{n-1} — próbka $n-1$ sygnału w formacie PCM. Przy różnicach ∇PCM-PCM odtworzenie sygnału przy założeniu warunku początkowego $\forall z_n = 0$, gdy $n < 0$ jest następujące:

$$z_n = z_{n-1} + \nabla z_n \tag{3}$$

$$z_n = \sum_{i=0}^{n} \nabla z_i = z_0 + \sum_{i=1}^{n} \nabla z_i,\ z_0 = z_{-1} + \nabla z_0 = \nabla z_0 \tag{4}$$

Zapis ∇PCM-PCM opisuje format, w którym pierwsza próbka z_0 jest przedstawiona w kodzie wielobitowego PCM natomiast reszta próbek jest przedstawiona w niskobitowym różnicowym kodzie ∇PCM.
Podstawiając wyrażenie (4) do (1) otrzymujemy transformatę w formacie różnicowym w następującej formie:

$$X(m) = k_m \sum_{n=0}^{N-1} \sum_{r=0}^{n} \nabla x_r \cos\left(\frac{\pi m(2n+1)}{2N}\right) = k_m \sum_{n=0}^{N-1} \left(x_0 + \sum_{r=1}^{n} \nabla x_r\right) \cos\left(\frac{\pi m(2n+1)}{2N}\right) \tag{5}$$

skąd

$$X(m) = k_m \sum_{n=0}^{N-1} x_0 \cos\left(\frac{\pi m(2n+1)}{2N}\right) + k_m \sum_{n=0}^{N-1} \sum_{r=1}^{n} \nabla x_r \cos\left(\frac{\pi m(2n+1)}{2N}\right) \tag{6}$$

Powyższy algorytm (5) składa się z $\frac{3N+N^3}{2}$ operacji mnożenia (nie uwzględniając mnożenia przez współczynnik k_m) różnic sygnału i funkcji kosinus oraz $\frac{3N+N^3}{2}$ sumowań nie uwzględniających sumowania podczas tworzenia czynników cos(). Utworzenie nowego szybkiego algorytmu wymaga transformowania sum różnic sygnału i sum kosinusów na płaszczyźnie, podobnie jak to zostało zrobione dla DFT w [10]. Otrzymujemy w ten sposób szybki różnicowy algorytm DCT:

$$X(m) = k_m \sum_{r=0}^{N-1} \nabla x_r \sum_{n=r}^{N-1} \cos\left(\frac{\pi m(2n+1)}{2N}\right) \tag{7}$$

Sumy kosinusów z wyrażenia (7) mogą zostać wcześniej obliczone i zapamiętane w pamięci procesora specjalizowanego DCT. Wtedy obliczenia według algorytmu (7)

wymagają N^2 operacji mnożenia (nie uwzględniając mnożenia przez współczynnik k_m) z bitowością różnic ∇PCM i sum kosinusów z bitowością PCM oraz N^2 sumowań. Otrzymane wyrażenie (7) może być podstawą szybkiego algorytmu FDCT (Fast DCT) z wykorzystaniem podejścia według Cooleya-Tukeya dzieląc zbiór danych wejściowych na odpowiednie podzbiory [2]. W ten sposób zmniejszona zostaje ilość operacji mnożenia w przybliżeniu do $N \log_2 N$.

Ponieważ format różnicowy ∇PCM-PCM nie wykorzystuje predykcji do kodowania sygnału, może to być źródłem dużych błędów kwantowania szczególnie dla krótkich słów kodowych. Błędy te można w dużej części wyeliminować stosując kody DM takie jak np. DPCM. Różnice ∇PCM są równoważne kodom DM wtedy, gdy nie występuje przeciążenie kodera [7]. Zamieniając różnice ∇PCM na kroki DM ($s_r^{(x)}$) w algorytmie (7) uzyskujemy szybki algorytm różnicowy w formacie DM-PCM:

$$X(m) = k_m \sum_{r=0}^{N-1} s_r^{(x)} \sum_{n=r}^{N-1} \cos\left(\frac{\pi m(2n+1)}{2N}\right) \tag{8}$$

Powyższa postać algorytmu umożliwia zastosowanie kroków kwantowania $s_r^{(x)}$, które są naturalnym wykładnikiem podstawy 2, co umożliwia całkowite wyeliminowanie czasochłonnych operacji mnożenia i zastąpienie ich szybkimi operacjami przesunięcia SHIFT.

Stosując DM do kodowania jak i przetwarzania sygnału trzeba odpowiednio wyznaczyć parametry tej modulacji takie jak częstotliwość próbkowania, rozmiar kroku kwantowania oraz minimalizować błędy szumu kwantyzacji i błędy przeciążenia kodera. Wszystkie te parametry oraz sposoby przejścia z różnic ∇PCM do kroków kwantowania $s_r^{(x)}$ zostały dokładnie opisane w pracach [7] i [8].

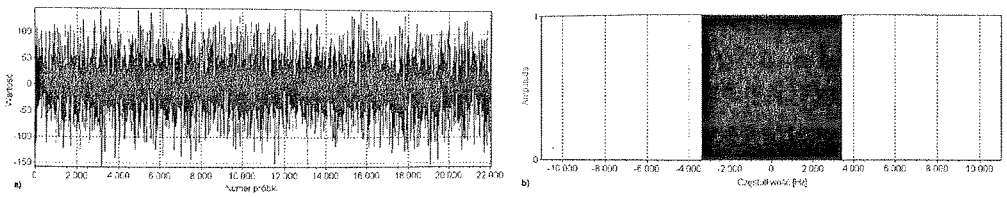
3. WIDMA SYGNAŁÓW OTRZYMANE Z WYKORZYSTANIEM RÓŻNICOWEGO DCT

3.1. WIDMO DCT SYGNAŁÓW PRZETWARZANYCH W FORMACIE PCM-PCM

3.1.1. Przetwarzanie szumu

Poniżej przedstawiono przykładowy sygnał szumu różowego jako ograniczony do częstotliwości 3400Hz szum biały wg. Rice'a [11] utworzony w 16-bitowym formacie PCM-PCM o częstotliwości próbkowania 22050Hz. Widma DFT i DCT sygnału z rysunku 1a zostały przedstawione na rysunkach 1b i 3a.

Jak widać na rysunkach 1b i 3a widmo DCT różni się od amplitudowego widma DFT przede wszystkim brakiem symetrii względem osi OY oraz występowaniem składowych częstotliwościowych zarówno ujemnych jak i dodatnich. Różnice pomiędzy widmami dotyczą także ich kształtów to znaczy w przeciwieństwie do DCT, kształt widma DFT



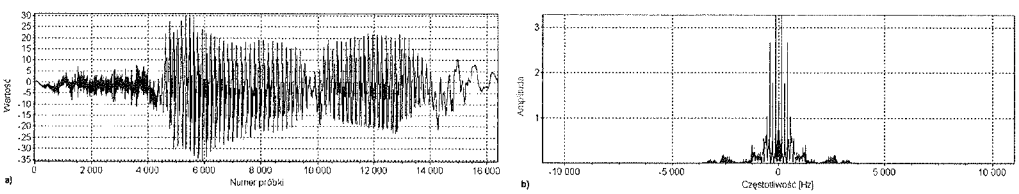
Rys. 1. a) Przebieg czasowy przykładowej realizacji szumu różowego o częstotliwości próbkowania 22050Hz, b) amplitudowe widmo DFT szumu różowego przedstawione w formacie PCM-PCM

Fig. 1. a) Example of pink noise realisation with the use of sampling rate 22050Hz, b) amplitude of DFT spectrum of the pink noise in PCM-PCM format

szumu różowego jest podobny do prostokąta a każda składowa częstotliwościowa ma jednakową amplitudę. Powyższe rysunki nie przedstawiają widm fazowych DFT i DCT. Widmo fazowe DCT jest liniowe o kącie 0° i nie jest wymagane jego przedstawianie.

3.1.2. Przetwarzanie sygnału mowy

Widmo DCT sygnału mowy różni się od widma DCT szumu przede wszystkim występowaniem silnie dominujących początkowych składowych widma. Rysunek 2a przedstawia przebieg czasowy słowa „szary" w 16-bitowym formacie PCM-PCM o częstotliwości próbkowania 22050Hz, natomiast rysunki 2b i 4a przedstawiają odpowiednio widmo amplitudowe DFT i widmo DCT tego sygnału.



Rys. 2. a) Przebieg czasowy słowa „szary" w formacie 16-bitowego PCM-PCM o częstotliwości próbkowania 22050Hz, b) amplitudowe widmo DFT słowa „szary" w formacie PCM-PCM

Fig. 2. a) Signal of the Polish word „szary" in 16-bit PCM-PCM format with the use of sampling rate 22050Hz, b) amplitude of DFT spectrum of the word „szary" in PCM-PCM format

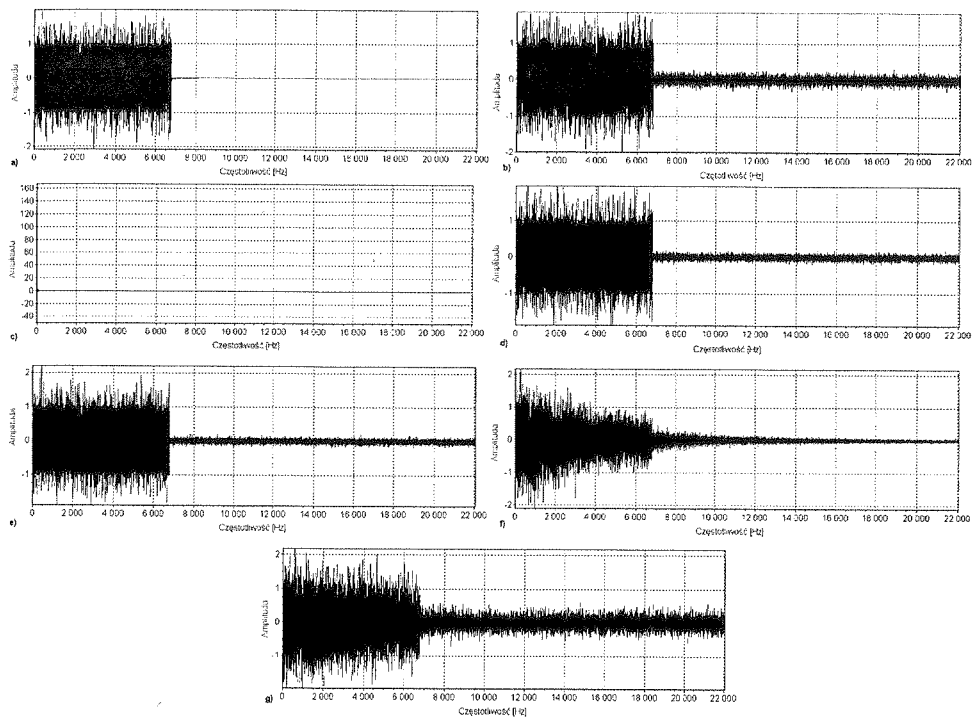
3.2. WIDMO DCT SYGNAŁÓW PRZETWARZANYCH W FORMATACH NISKOBITOWYCH

Przetwarzanie DCT sygnałów w niskobitowych formatach różnicowych powoduje powstawanie lub wzrost wyższych składowych harmonicznych. Termin wyższe składowe harmoniczne w tym kontekście odnosi się do składowych częstotliwościowych DCT

znajdujących się powyżej podwójnej częstotliwości granicznej sygnału. Podobna sytuacja występuje podczas przetwarzania sygnałów w niskobitowym formacie PCM-PCM. Jest to spowodowane szumami kwantyzacji niskobitowego przedstawienia sygnału.

3.2.1. Przetwarzanie szumu

Rysunki 3b–3g przedstawiają kształty widm DCT szumu różowego z rysunku 3a przetwarzanego odpowiednio w 3 bitowych formatach: PCM-PCM, ∇PCM-PCM, PCM-PCM, MDPCM-PCM oraz jednobitowych formatach ADM-PCM i LDM-PCM.



Rys. 3. Widmo DCT szumu różowego o częstotliwości próbkowania 22050Hz przetwarzanego w a) 16-bitowym formacie PCM-PCM, b) 3-bitowym formacie PCM-PCM, c) 3-bitowym formacie ∇PCM-PCM, d) 3-bitowym formacie DPCM-PCM, e) 3-bitowym formacie MPCM-PCM, f) 1-bitowym formacie ADM-PCM, g) 1-bitowym formacie LDM-PCM

Fig. 3. DCT spectrum of the pink noise with the use of sampling rate 22050Hz, processed in a)16-bit PCM- PCMformat, b) 3-bit PCM-PCM format, c) 3-bit ∇PCM-PCM format, d) 3-bit DPCM-PCM format, e) 3-bit MPCM-PCM format, f) 1-bit ADM-PCM format, g) 1-bit LDM-PCM format

Widma DCT szumu różowego przedstawione na rysunkach 3b, 3d i 3e są zbliżone do siebie kształtem i są podobne do widma szumu przetwarzanego w wielobitowym formacie PCM-PCM (rysunek 3a). Różnice między widmami DCT przedstawionymi

na rysunku 3 wynikają z zawartych w nich szumów kwantyzacji i błędów przeciążenia kodera, co wiąże się ze sposobem kodowania przetwarzanego sygnału. Odzwierciedla się to w uzyskanej podczas symulacji wartości współczynnika SNR (Signal to Noise Ratio), który określa stosunek sygnału do szumu w nim zawartego a opisany jest wzorem:

$$SNR = 10\log_{10} P_X/P_A \tag{9}$$

gdzie, $P_X = \frac{1}{N}\sum_{i=0}^{N-1} x_i^2$, $P_A = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - \hat{x}_i)^2$, x_i — próbka i sygnału wejściowego w formacie PCM-PCM, $\hat{x}_i$ — próbka i sygnału odtworzonego (wyjściowego) w formacie PCM-PCM, N — ilość próbek sygnału.

I tak, przetwarzając szum w niskobitowych formatach PCM-PCM, DPCM-PCM oraz MDPCM-PCM otrzymano większą wartość współczynnika SNR niż w przypadku zastosowania jednobitowych modulacji LDM-PCM i ADM-PCM oraz różnicowego formatu ∇PCM-PCM w którym nie występuje predykcja sygnału.

Naturalnie, zwiększenie częstotliwości próbkowania modulacji jednobitowych i modulacji ∇PCM może znacznie zwiększyć wartość współczynnika SNR oraz poprawić kształt widma.

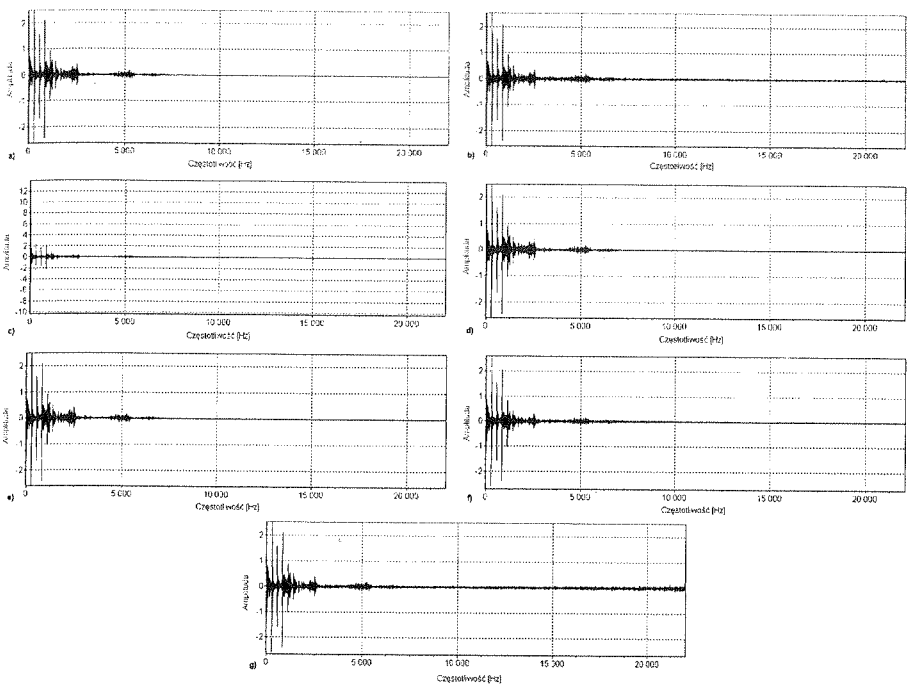
Najmniejsze zniekształcenia wśród przetwarzanych sygnałów przedstawionych na powyższych rysunkach (największa wartość współczynnika SNR) ma widmo sygnału w formacie MDPCM-PCM z rysunku 3e i wynosi 17,60dB przy częstotliwości próbkowania 22050Hz.

We wszystkich przedstawionych powyżej widmach DCT, wyższe harmoniczne licząc od częstotliwości 6800Hz mają znacznie mniejszą wartość amplitudy niż składowe częstotliwościowe do 6799Hz a wynika to z ograniczenia częstotliwościowego szumu podczas jego tworzenia [11].

3.3. PRZETWARZANIE SYGNAŁU MOWY

Poniższe rysunki (4a-4g) przedstawiają kształty widm DCT słowa „szary" przetwarzanego w 16-bitowym formacie PCM-PCM, 3-bitowym formacie PCM-PCM oraz 3 i 1-bitowych formatach różnicowych. Częstotliwość próbkowania przetwarzanego sygnału wynosiła 22050Hz.

Podczas przetwarzania DCT sygnału mowy występuje duże zróżnicowanie mocy składowych sygnału, przez co jego widmo jest nierównomierne i skoncentrowane w niewielkiej ilości składowych niskoczęstotliwościowych. Podobnie jak podczas przetwarzania szumu różowego, najbardziej zniekształcone widmo DCT przetwarzanego słowa otrzymano dla formatu ∇PCM-PCM. Natomiast najbardziej zbliżone do oryginalnego kształtu widma DCT (rysunek 4a) jest widmo słowa przetwarzanego w 3-bitowym formacie MDPCM-PCM, którego wartość współczynnika SNR wyniosła 24,92dB przy częstotliwości próbkowania 22050Hz. W przeciwieństwie do formatów PCM-PCM, DPCM-PCM, MDPCM-PCM czy ADM-PCM przetwarzanie w jednobitowym formacie



Rys. 4. Widmo DCT słowa „szary" o częstotliwości próbkowania 22050Hz przetwarzanego w a) 16-bitowym formacie PCM-PCM, b) 3-bitowym formacie PCM-PCM, c) 3-bitowym formacie ∇PCM-PCM, d) 3-bitowym formacie DPCM-PCM, e) 3-bitowym formacie MPCM-PCM, f) 1-bitowym formacie ADM-PCM, g) 1-bitowym formacie LDM-PCM

Fig. 4. DCT spectrum of Polish word „szary" with the use of sampling rate 22050Hz, processed in a)16-bit PCM- PCMformat, b) 3-bit PCM-PCM format, c) 3-bit ∇PCM-PCM format, d) 3-bit DPCM-PCM format, e) 3-bit MPCM-PCM format, f) 1-bit ADM-PCM format, g) 1-bit LDM-PCM format

LDM-PCM powoduje wzrost wartości amplitud wyższych składowych harmonicznych w pobliżu częstotliwości próbkowania 22050Hz.

Współczynnik SNR, który wyznaczony był po rekonstrukcji sygnału (IDCT) przez porównanie sygnału odtworzonego, z wzorcowym (oryginalnym) sygnałem zakodowanym w 16-bitowym PCM formacie jest jednoznacznie powiązany ze zniekształceniem widma DCT. Im widmo jest bardziej odkształcone tym wartość współczynnika SNR jest mniejsza.

4. DOKŁADNOŚĆ OBLICZENIOWA ZAPROPONOWANYCH ALGORYTMÓW

Dokładność zaproponowanych algorytmów została zbadana symulacyjnie przy pomocy napisanego w języku Builder C++ programu, w którym przetwarzano sygnały losowe, którymi były szum różowy i sygnały mowy. Szum różowy został utworzo-

ny z 3400 składowych częstotliwościowych o takich samych amplitudach i losowych przesunięciach fazowych. Natomiast sygnałami mowy były polskie słowa wybrane na podstawie podziału na fonemy w oparciu o pozycję [12] oraz na podstawie częstości ich występowania w języku polskim [13]. Słowami tymi były: *rok, rak, lis, dom, praca, ryba, szary, mama, tak.*

Sposób przeprowadzenia badań polegał na przekodowaniu sygnałów testowych (szum różowy i sygnały mowy) z formatu 16-bitowego PCM-PCM, w którym były zakodowane na odpowiedni niskobitowy format różnicowy lub niskobitowy format PCM-PCM. Następnie odpowiednio zakodowany sygnał poddawano różnicowej transformacie DCT, w wyniku której otrzymano widmo DCT tego sygnału. Ostatnim elementem badań było wykonanie odwrotnej transformaty DCT oraz obliczenie współczynnika SNR, który został wybrany do obiektywnej oceny dokładności przetwarzania, porównując otrzymany sygnał po IDCT z sygnałem oryginalnym w formacie 16-bitowego PCM-PCM.

4.1. PRZETWARZANIE SZUMU

W tym paragrafie przedstawione są wyniki badań symulacyjnych dokładności zaproponowanych algorytmów na przykładzie szumu różowego. W tabeli 1 przedstawione są wartości SNR dla szumu różowego przetwarzanego w sześciu formatach dla czterech częstotliwości próbkowania i pięciu różnych długości słów kodowych.

Najmniejsze wartości współczynnika SNR podczas przetwarzania szumu różowego otrzymujemy dla sygnału przetwarzanego w formacie ∇PCM-PCM dla słów kodowych o długościach od 2 do 5 bitów. Niska wartość współczynnika SNR w tym przypadku jest związana z brakiem predykcji podczas kodowania sygnału i powstawaniem dużych błędów przeciążenia kodera. Wyjątkiem jest częstotliwość próbkowania 44100Hz, dla której współczynnik SNR osiągnął wartość 5,63dB. W ten sposób zwiększając częstotliwość próbkowania oraz ilość bitów w formacie ∇PCM-PCM można uzyskać wzrost wartości współczynnika SNR przez znaczne zmniejszenie błędów przeciążenia.

Przetwarzanie w niskobitowym formacie PCM-PCM jest korzystne do częstotliwości próbkowania 11025Hz, ponieważ w przypadku użycia wyższych częstotliwości próbkowania większe wartości SNR dla słów kodowych o tej samej liczbie bitów osiągają formaty DPCM-PCM oraz MDPCM-PCM. Wartość współczynnika SNR podczas przetwarzania szumu w formacie PCM-PCM zmienia się w niewielkim zakresie, gdy zmieniamy częstotliwość próbkowania sygnału.

Dla formatów jednobitowych LDM-PCM i ADM-PCM uzyskujemy niskie wartości SNR, które są wyższe tylko od wartości SNR dla formatu ∇PCM-PCM. Formaty te wymagają wysokiej częstotliwości próbkowania.

Z badań przedstawionych w tym paragrafie wynika, że najlepsze efekty osiągamy podczas przetwarzania szumu w formacie DPCM-PCM.

4.2. PRZETWARZANIE SYGNAŁU MOWY

Podobnie jak dla szumu różowego, podczas symulacji komputerowych wyznaczono wartości współczynników SNR dla sygnałów mowy przetwarzanych w niskobitowych formatach różnicowych i w niskobitowym formacie PCM-PCM. Wyniki symulacji przedstawia tabela 2.

Porównując wartości SNR przedstawione w tabeli 2 zauważyć można, że największe wartości współczynnika SNR dla zadanych parametrów przetwarzania (częstotliwość próbkowania i liczba bitów słowa kodowego) uzyskaliśmy dla słów przetworzonych w formacie DPCM-PCM oraz MDPCM-PCM. W przypadku przetwarzania sygnałów mowy w formacie PCM-PCM wartość współczynnika SNR jest znacznie mniejsza niż dla tych samych sygnałów o tej samej bitowości przetwarzanych w formatach DPCM-PCM czy MDPCM-PCM. Co więcej, stosując format MDPCM-PCM dla słów kodowych 3 i 4-bitowych uzyskujemy większe wartości współczynnika SNR niż dla formatu DPCM-PCM.

Dla przykładu, przetwarzając sygnały mowy o częstotliwości próbkowania 44100Hz dla 4-bitowych słów kodowych różnica osiągniętego uśrednionego współczynnika SNR pomiędzy formatem MDPCM-PCM i formatem DPCM-PCM wyniosła 3,51dB natomiast między formatem MDPCM-PCM a PCM-PCM aż 23,01dB. Zatem uproszczenie obliczeń przez eliminację operacji mnożenia na rzecz operacji przesunięcia bitowego, co zwiększa szybkość ich wykonywania umożliwia wzrost wartości współczynnika SNR. Jak widać z porównania tabel 1 i 2, wartości SNR podczas przetwarzania sygnałów mowy są istotnie większe od wartości SNR uzyskanych przy przetwarzaniu szumu różowego. Wyjaśnić to można istnieniem znacznie większej korelacji pomiędzy fonemami w sygnale mowy niż korelacji pomiędzy składowymi częstotliwościowymi w krótkim sygnale szumu.

Symulacje dla sygnałów mowy przeprowadzono także pod kątem przepływności danych dla wybranych sposobów kodowania, długości słów kodowych i częstotliwości przetwarzania.

Tabela 3 podzielona została na dwie części ze względu na szybkość przepływu danych. Dodatkowo każdą z części tabeli podzielono na podzbiory wyznaczone przez długości słów kodowych. W ten sposób zauważyć można, że dla przepływności 48kb/s jednobitowy format ADM-PCM przewyższa dokładnością przetwarzania 2, 3 i 4-bitowy format PCM-PCM oraz dwubitowy format DPCM-PCM a i dorównuje 3-bitowym formatom DPCM-PCM i MDPCM-PCM. Podobna sytuacja ma miejsce dla przepływności 64kb/s. Jak pokazują wyniki badań przedstawione w tabeli 2 i 3, format DPCM-PCM jest formatem najdokładniejszym z punktu widzenia przetwarzania DCT sygnałów mowy, dla słów kodowych dłuższych niż 4 bitowe. Z drugiej strony szybki MDPCM-PCM jest dokładniejszy dla 3 i 4-bitowych słów kodowych. Cechy wysokiej dokładności formatu DPCM-PCM oraz dużej szybkości obliczeń wykonywanych w formacie MDPCM-PCM można wykorzystać i utworzyć syntezowany format DPCM-PCM (SDPCM-PCM), któ-

Tabela 1

Porównanie SNR dla szumu różowego przetwarzanego w formacie PCM-PCM i formatach różnicowych

The comparison of the SNR of pink noise processed in PCM-PCM and differential formatst

Sposób przedstawienia sygnału	Liczba bitów	Częstotliwość próbkowania			
		8000 Hz	11025 Hz	22050 Hz	44100 Hz
		SNR [dB]			
PCM	2	7,22	7,29	7,14	7,17
VPCM		-0,42	-3,73	-3,16	-0,06
DPCM		4,82	6,37	10,00	14,52
MDPCM		4,82	6,37	10,00	14,52
PCM	3	13,29	13,38	13,22	13,26
VPCM		-0,42	-3,73	-3,16	-0,07
DPCM		11,11	12,91	17,30	22,24
MDPCM		11,01	12,91	17,60	23,00
PCM	4	18,81	19,03	18,87	18,90
VPCM		-0,47	-3,73	-3,16	-0,13
DPCM		16,85	18,65	23,43	28,41
MDPCM		12,09	13,99	19,13	24,96
PCM	5	24,23	24,52	24,29	24,42
VPCM		0,05	-2,72	1,64	5,63
DPCM		22,49	24,18	29,11	34,12
MDPCM		12,10	13,99	19,13	24,97
LDM	1	2,14	3,12	5,80	9,92
ADM	1	0,59	1,21	4,49	9,18

Tabela 2

Porównanie uśrednionych wartości SNR dla słów przetwarzanych w formacie PCM-PCM i formatach różnicowych

The comparison of mean SNR coefficients for some words processed in PCM-PCM and differential formatst

Sposób przedstawienia sygnału	Liczba bitów	Częstotliwość próbkowania			
		8000 Hz	11025 Hz	22050 Hz	44100 Hz
		SNR [dB]			
PCM	2	6,55	6,34	6,38	6,37
VPCM		0,20	0,06	0,31	0,31
DPCM		11,50	12,91	16,74	21,05
MDPCM		11,50	12,91	16,74	21,05
PCM	3	11,64	11,14	11,19	11,19
VPCM		0,20	0,06	1,77	0,30
DPCM		17,52	19,29	23,94	28,81
MDPCM		18,82	20,81	25,78	30,69
PCM	4	17,02	15,68	15,67	15,67
VPCM		0,20	1,04	1,98	1,75
DPCM		23,07	25,03	29,93	35,17
MDPCM		23,69	26,90	32,83	38,68
PCM	5	21,62	20,73	20,69	20,70
VPCM		4,16	4,10	4,26	4,76
DPCM		28,38	30,50	35,81	41,22
MDPCM		23,73	26,95	32,88	38,74
LDM	1	7,01	8,07	11,92	16,09
ADM	1	8,69	10,90	16,25	21,73

Tabela 3

Porównanie wybranych sposobów kodowania sygnałów mowy dla przepływności 48, 60, 64 i 66kb/s

The comparison of selected ways of coding voice signals for bit rate 48, 60, 64, 66kb/s

Sposób kodowania (częstotliwość próbkowania – ilość bitów - przepływność)	SNR [dB]
LDM (48kHz – 1 bit – 48kb/s)	16,09
ADM (48kHz – 1 bit – 48kb/s)	21,73
PCM (24kHz – 2 bity – 48kb/s)	6,42
DPCM, MDPCM (24kHz – 2 bity – 48kb/s)	12,66
PCM (16kHz – 3 bity – 48kb/s)	11,67
DPCM (16kHz – 3 bity – 48kb/s)	22,43
MDPCM (16kHz – 3 bity – 48kb/s)	24,18
PCM (12kHz – 4 bity – 48kb/s)	15,95
DCM (12kHz – 4 bity – 48kb/s)	26,00
MDPCM (12kHz – 4 bity – 48kb/s)	27,57
PCM (8kHz – 6 bity – 48kb/s)	27,63
DPCM (8kHz – 6 bitów – 48kb/s)	33,49
MDPCM (8kHz – 6 bitów – 48kb/s)	23,68
LDM (64kHz – 1 bit – 64kb/s)	19,39
ADM (64kHz – 1 bit – 64kb/s)	25,09
PCM (32kHz – 2 bity – 64kb/s)	6,60
DPCM, MDPCM (32kHz – 2 bity – 64kb/s)	19,70
PCM (22kHz – 3 bity – 66kb/s)	11,19
DPCM (22kHz – 3 bity – 66kb/s)	23,94
MDPCM (22kHz – 3 bity – 66kb/s)	25,78
PCM (16kHz – 4 bity – 64kb/s)	16,63
DPCM (16kHz – 4 bity – 64kb/s)	28,68
MDPCM (16kHz – 4 bity – 64kb/s)	30,16
PCM (12kHz – 5 bitów – 60kb/s)	21,02
DPCM (12kHz – 5 bitów – 60kb/s)	31,48
MDPCM (12kHz – 5 bitów – 60kb/s)	27,60
PCM (11kHz – 6 bitów – 66kb/s)	26,67
DPCM (11kHz – 6 bitów – 66kb/s)	36,16
MDPCM (11kHz – 6 bitów – 66kb/s)	26,90
PCM (8kHz – 8 bitów – 64kb/s)	39,41
DPCM (8kHz – 8 bitów – 64kb/s)	44,68
MDPCM (8kHz – 8 bitów – 64kb/s)	23,68

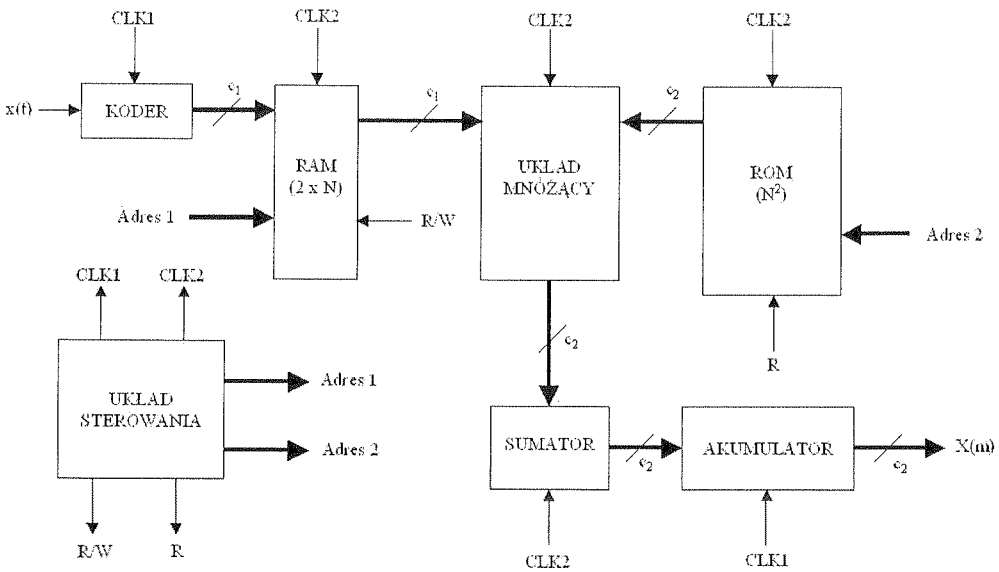
ry umożliwi uproszczenie budowy procesorów specjalizowanych pracujących w czasie rzeczywistym.

5. STRUKTURY PROCESORÓW SPECJALIZOWANYCH

Wysoka dokładność obliczeniowa różnicowych algorytmów DCT dla formatów DPCM-PCM i MDPCM-PCM przedstawiona w punkcie 4 sprzyjała opracowaniu struktur specjalizowanych procesorów DCT dla tychże formatów. Poniżej zostały przedstawione dwie przykładowe struktury procesorów specjalizowanych opartych na wyrażeniu (8).

5.1. PODSTAWOWA STRUKTURA SZYBKIEGO PROCESORA DCT

Schemat blokowy specjalizowanego procesora DCT dla danych wejściowych zakodowanych w formacie DPCM-PCM lub MDPCM-PCM przedstawia rysunek 5. Procesor



Rys. 5. Schemat blokowy specjalizowanego procesora DCT dla formatu DPCM-PCM lub MDPCM-PCM

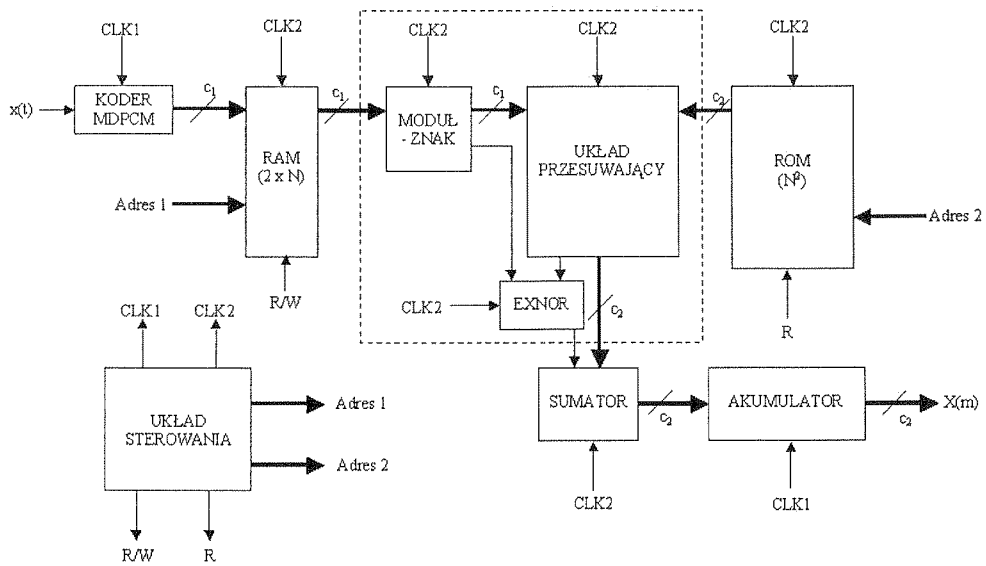
Fig. 5. Block diagram of specialized DCT processor for DPCM-PCM or MDPCM-PCM format

specjalizowany do realizacji różnicowego algorytmu DCT (8) przedstawiony na rysunku 5 posiada dwie pamięci: RAM i ROM. Pamięć ROM służy do przechowywania wcześniej obliczonych sum kosinusów z wyrażenia 7 lub 8 (rozdział 2.2). Zapamiętanie obliczonych wyrażeń w pamięci ROM znacznie skraca czas wyznaczenia współczynników DCT. Pamięć RAM o pojemności 2N komórek o odpowiedniej bitowości zależnej od wybranego formatu spełnia dwie funkcje. Po pierwsze zapamiętuje aktualnie kodowane i przesyłane z KODERA różnicowego DM kroki kwantyzacji $\left\{s_r^{(x)}\right\}_{r=0}^{N-1}$, których do wyznaczenia różnicowego DCT jest potrzebnych N. Drugą funkcją pamięci RAM jest przesyłanie kolejno N próbek do UKŁADU MNOŻENIA, które zostały zapamiętane w poprzednim cyklu próbkowania i kodowania. UKŁAD MNOŻENIA oblicza iloczyn odpowiednich sum kosinusów z pamięci ROM oraz odpowiednich kroków DM z pamięci RAM, które to iloczyny są odpowiednio sumowane (SUMATOR) i akumulowane (AKUMULATOR) tworząc sygnał wyjściowy $\{X(m)\}_{m=0}^{N-1}$, którym są składowe widma DCT. UKŁAD STEROWANIA synchronizuje działanie KODERA oraz odpowiedni zapis i odczyt próbek do i z pamięci RAM a także odczyt sum kosinusów z pamięci

ROM. Dlatego też, sygnał sterujący CLK2 ma N razy większą częstotliwość od sygnału sterującego CLK1. Przykładowe bitowości szyny danych c1 wynoszą 3, 4 lub 5 bitów natomiast bitowość szyny danych c2 zależy od przyjętej dokładności obliczeniowej i może wynosić np. 16 lub 32 bity.

5.2. STRUKTURA SZYBKIEGO PROCESORA RÓŻNICOWEGO DCT Z OPERACJĄ SHIFT

Procesor specjalizowany dla różnicowego DCT z rysunku 5 ma prostą strukturę, lecz posiada ograniczenie szybkości działania wynikające z użycia UKŁADU MNOŻENIA. Ograniczenie to można zniwelować zastępując mnożenie szybkimi operacjami przesunięcia bitowego SHIFT. Jest to możliwe, gdy kroki kwantowania są proporcjonalne do potęgi liczby 2 czyli zastosowany jest format MDPCM. Taką sytuację przedstawia schemat blokowy specjalizowanego procesora DCT z rysunku 6.



Rys. 6. Schemat blokowy specjalizowanego procesora DCT dla formatu MDPCM-PCM z operacją przesunięcia SHIFT

Fig. 6. Block diagram of specialized DCT processor for MDPCM-PCM format with SHIFT operation

Struktura procesora specjalizowanego DCT dla formatu MDPCM-PCM (rysunek 6) jest podobna do struktury procesora specjalizowanego z rysunku 5. Różnice pomiędzy nimi zaznaczono linią przerywaną. UKŁAD MNOŻENIA z rysunku 20 zastąpiono UKŁADEM PRZESUWAJĄCYM, układem wydzielającym z kroków MDPCM ich znaku i modułu (MODUŁ-ZNAK) oraz sumatorem modulo 2 (EXNOR). Kroki MDPCM przesyłane są z pamięci RAM do układu ZNAK-MODUŁ. Każdy moduł określa liczbę przesunięć wielobitowych sum kosinusów przesyłanych z pamięci ROM.

Znak kroku kwantowania razem ze znakiem wyniku przesunięcia jest sumowany modulo 2 w układzie EXNOR. Wyniki przesunięć są sumowane (SUMATOR) ze znakiem wyznaczonym przez układ EXNOR i akumulowane (AKUMULATOR).
Przedstawiona powyżej struktura pozwala na szybkie i dokładne wykonywanie obliczeń różnicowego DCT bez czasochłonnych operacji wielobitowego mnożenia, co jest szczególnie istotne dla procesorów czasu rzeczywistego zbudowanych w oparciu o pola programowalne np. FPGA (Field Programmable Gate Array).

5.3. WYNIKI

W pracy zaprezentowano wykorzystanie różnicowych formatów dyskretnego przetwarzania kosinusowego sygnałów losowych powiązanych z obliczeniami na krótkich słowach kodowych. Zbadano możliwości szybkich i dokładnych obliczeń DCT na krótkich różnicowych słowach kodowych oraz zaproponowano dwie struktury procesorów specjalizowanych DCT, zorientowanych na pracę w trybie czasu rzeczywistego.
Wyniki badań przedstawione w tabelach 2 i 3 pokazują, że stosując te same warunki przetwarzania (częstotliwość próbkowania i długość słowa kodowego) sygnały mowy przetwarzane w formatacie DPCM-PCM osiągają znacznie wyższe wartości SNR od słów przetwarzanych w formacie PCM-PCM. Modyfikując format DPCM tak, aby kroki kwantowania były proporcjonalne do potęgi liczby 2 (MDPCM) eliminujemy z obliczeń wszystkie operacje mnożenia i zastępujemy je szybkimi operacjami przesunięcia bitowego SHIFT. Pozwala to na zwiększenie szybkości działania algorytmów przy zachowaniu wysokiej rozdzielczości obliczeń. Jednocześnie, przetwarzanie sygnałów mowy z wykorzystaniem MDPCM kodów do 4 bitów pozwala na znaczne zwiększenie współczynnika SNR w porównaniu do przetwarzania mowy w kodzie DPCM.
Powyżej przedstawione algorytmy są perspektywiczne i mogą być rozwijane w różnych kierunkach, w szczególności do realizacji na polach obliczeniowych takich jak np. FPGA.

6. BIBLIOGRAFIA

1. R.N. Bracewell: *The Fourier Transform and its Applications. Singapore*, McGraw Hill, 2000.
2. J.W. Cooley, J.W. Tukey: *An algorithm for the machine calculation of complex Fourier series*, Mathematics of Computation, vol. 19, 1965, pp. 297-301.
3. N. Ahmed, T. Natarajan, K.R. Rao: *Discrete Cosine Transform*, IEEE Transaction on Computers, vol. 23, January 1974, pp. 90-93
4. L.P. Yaroslavsky: *Digital Picture Processing — an introduction*, Berlin Heidelberg, Springer-Verlang 1985, pp. 99-110.
5. K.R. Rao, P. Yip: *Discrete Cosine Transform — algorithms, advantages, applications*, San Diego, Academic Press, 1990.
6. K. Sayood: *Introduction to Data Compression*, San Francisco, Morgan Kaufman Publishers, 2000.
7. W. Pogribny, A. Rajs: *Kodowanie i odtwarzanie sygnałów losowych z wykorzystaniem ekonomicznych modulacji różnicowych*, Kwartalnik Elektroniki i Telekomunikacji, tom 48 zeszyt 1, ss. 7-25.

8. W. Pogribny, A. Rajs: *Dokładność algorytmów modulacji delta z krokami kwantowania będącymi potęgą liczby 2*, Krajowe Sympozjum Telekomunikacji, Bydgoszcz, 1999, ss. 44-50.
9. P.J. van Grewen, W.F.G. Mecklenbräuker, N.A.M. Verhoeckx, F.A.M. Snijders, H.A. van Essen: *A New Type of Digital Filter for Data Transmission*, IEEE Transaction on Communications, vol. 23 no. 2, February 1975, pp. 222-234.
10. W. Pogribny: *Delta Modulacja w cyfrowym przetwarzaniu sygnałów*, Moskwa, Radio i Swiaź, 1990 (w języku rosyjskim).
11. S. O. Rice: *Mathematical analysis of random noise*, Bell System Technical Journal, vol. 23, no 3, 1944, pp. 282-232.
12. R. Tadeusiewicz: *Sygnał mowy*, Warszawa, Wydawnictwa Komunikacji i Łączności, 1987.
13. W. Jassem, D. Gembiak: *Subiektywne prawdopodobieństwo wyrazów polskich*, Poznań, Państwowe Wydawnictwo Naukowe, 1980

W. POGRIBNY, M. DRECHNY

DIFFERENTIAL DCT IN SIGNAL PROCESSING ISSUES

Summary

Discrete Cosine Transform (DCT) is widely used in analysis and signal compression. The classical DCT algorithms are based on calculations in multibit PCM (Pulse Code Modulation) format. The restriction of using PCM format in DCT is the application of multibit multiplication operations, which often limits the usage of PCM in real time systems. On the other hand, using some low bit kinds of delta modulation (DM) ensures the same calculation accuracy as in case of the multibit PCM format, however this approach has not been examined sufficiently yet.

The purpose of the work was working out and studying the effective differential DCT algorithms and processors structures working in real time.

In this work are presented new fast differential DCT algorithms. The particular attention was paid to the possibility of eliminating the multiplication operations in the proposed algorithms and replacing them by fast SHIFT operations. The application of shifts is done on the basis of the quantization steps, which are the power of number 2.

To study the accuracy of proposed algorithms pink noise and voice signals were used. The authors proposed fast and effective structures of specialized processors realizing the worked out algorithms. The results of computer simulations confirm the efficiency of the presented approach.

Keywords: discrete cosine transform, delta modulation, DPCM

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 511–520

Uniwersalna miara oceny jakości obrazów cyfrowych

EUGENIUSZ KORNATOWSKI

Katedra Przetwarzania Sygnałów i Inżynierii Multimedialnej,
Politechnika Szczecińska, Wydział Elektryczny,
ul. 26 Kwietnia 10, 71-126 Szczecin,
e-mail: korn@ps.pl

Otrzymano 2005.06.20
Autoryzowano 2005.01.10

W artykule przedstawiono definicję obiektywnej miary jakości odwzorowania sygnału wejściowego na wyjście dowolnego algorytmu DSP, a w szczególności nieliniowych filtrów cyfrowych. Współczynnik probabilistycznej wierności odwzorowania (tzw. współczynnik PIF — opracowany przez autora i omówiony we wcześniejszych publikacjach) bazuje na funkcji wiążącej dystrybuantę odpowiedzi algorytmu z dystrybuantą wymuszenia. W swojej pierwotnej definicji kryterium to nie uwzględniało pewnych zaburzeń struktury obrazu. Niniejsza praca zawiera wyniki badań nad możliwościami zwiększenia efektywności współczynnika PIF.

Słowa kluczowe: filtracja cyfrowa, cyfrowe przetwarzanie sygnałów

1. WSTĘP

Dynamiczny rozwój zarówno sprzętu obliczeniowego, jak i opracowywane w wielu ośrodkach naukowych skomplikowane, ale skuteczne algorytmy przetwarzania sygnału wymuszają stosowanie obiektywnych i zgodnych z odczuciami subiektywnymi kryteriów oceny jakości przetwarzania. Problem jest bardzo istotny z bardzo wielu powodów. Obiektywne kryterium oceny jakości sygnału umożliwia bowiem porównanie wielu algorytmów i wybór konkretnego, odpowiedniego w przypadku określonego zagadnienia. Aspekt ten ma znaczenie w przypadku doboru algorytmu w zagadnieniach regeneracji sygnału np. usuwaniu zakłóceń. W procesie doboru stopnia kompresji obiektywna ocena jakości sygnału skompresowanego jest także oczywista. Konstruowanie wielu algorytmów przetwarzania (filtracji) adaptacyjnego także wymaga zastosowania określonego kryterium, którego optymalizacja będzie podstawą działania algorytmu. Oczy-

wiste jest także, że matematycznie zdefiniowane i opisane kryterium musi być zgodne z odczuciami subiektywnymi w tych zagadnieniach, gdzie przetwarzaniu podlega dźwięk lub obraz.

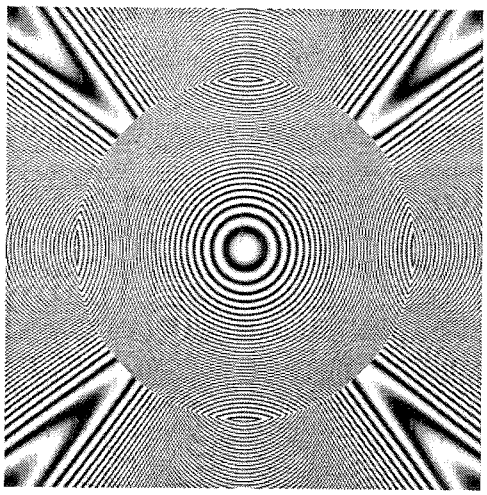
Praca ta poświęcona jest problemom wykorzystania opracowanego przez autora kryterium PIF [10] z uwzględnieniem pewnych jego modyfikacji. Rozważania dotyczą problemu oceny jakości obrazu o reprezentacji w postaci odcieni szarości. Wnioski płynące z przedstawionych dalej rozważań można odnieść zarówno do zagadnień dotyczących przetwarzania dźwięku (sygnału 1-D), a także (po pewnych modyfikacjach) obrazów kolorowych.

2. OCENA JAKOŚCI ALGORYTMÓW DSP

Pod pojęciem „jakość przetwarzania" zwykle rozumie się co najmniej dwa parametry: zachowanie pierwotnej struktury sygnału i stopień tłumienia zakłóceń. W tej pracy rozpatrywany jest tylko pierwszy z nich, bowiem — zdaniem autora — jest trudniejszy do zdefiniowania i budzi wśród autorów prac z zakresu DSP wiele kontrowersji [2]. Dowodem tego jest mnogość miar jakości: AD (średnia różnica), MD (maksymalna różnica — maximum difference), IF (wierność obrazu — image fidelity), MSE (błąd średniokwadratowy — mean square error), PMSE (szczytowy błąd średniokwadratowy — peak mean square error), NAE (znormalizowany błąd bezwzględny — normalized absolute error), NMSE (znormalizowany błąd średniokwadratowy — normalized mean square error) [3] itd.

Cytowane miary jakości nie wyczerpują wszystkich możliwości, ale po analizie już tych kilku można bez trudu zauważyć, że wartość liczbowa wskaźników zależy od treści obrazu testowego. Ta wada jest niezwykle istotna, ponieważ pozwala wyłącznie na analizę porównawczą z wykorzystaniem (we wszystkich eksperymentach) tego samego obrazu testowego. Chcąc uczynić z powyższych zależności wzory opisujące uniwersalne miary jakości przetwarzania obrazu, należałoby zdefiniować obraz testowy. Rzadkością jest stosowanie w eksperymentach obrazu testowego, który próbowano upowszechnić w latach osiemdziesiątych. Definicja tego obrazu podana jest w [1], a obraz wygenerowany na mocy tej definicji pokazano na rys. 1.

Ten syntetyczny obraz, co do swojej struktury, jest na tyle skomplikowany, iż z łatwością umożliwia ocenę zniekształceń wnoszonych przez badany algorytm. Niestety, nie jest stosowany powszechnie i badania prowadzone przez autorów różnych prac są trudne do skonfrontowania. W większości przypadków stosowane są własne obrazy testowe, często fotografie. Stosowanie wielu miar jakości (lub tylko wybranych), w połączeniu z pełną dowolnością w wyborze sygnału testowego, może prowadzić do niejednoznaczności, a wręcz błędów. Tę tezę ilustrują wyniki eksperymentu , który autor opisał w [10, *Przykład 1*] . W celu porównania jakości przetwarzania użyto w przykładzie dwóch fotografii, poddając je filtracji dwoma filtrami medianowymi tak dobranymi, iż łatwo było przewidzieć efekt końcowy: jeden z filtrów zdecydowanie degradował obraz przetwarzany, drugi z kolei tylko w minimalny sposób zaburzał struk-



Rys. 1. Obraz testowy o definicji pokazanej w [1]

Fig. 1. Testing image with definition shown by [1]

turę obrazu. Obserwując wartości obliczonych w eksperymencie miar jakości trudno było wyciągnąć wiarygodne wnioski, nie mówiąc o ich zbieżności z odczuciami subiektywnymi. Największą ułomnością przeprowadzonych testów było to, że wynik zależał od rodzaju sygnału wejściowego — z wyłączeniem oczywiście kryterium PIF, bowiem z założenia nie zależy ono od rodzaju przetwarzanego obrazu, a jedynie od właściwości probabilistycznych algorytmu. Jak pokazano w [10] PIF odzwierciedla także wrażenia subiektywne.

Przykład 1.

Określono jakość obrazu z rys.1 poddanego kompresji algorytmem JPEG. W eksperymencie wykorzystano następujące kryteria jakości [3,10]:
— średnia różnica, **AD**
— maksymalna różnica (maximum difference), **MD**
— wierność obrazu (image fidelity), **IF**
— błąd średniokwadratowy (mean square error), **MSE**
— szczytowy błąd średniokwadratowy (peak mean square error), **PMSE**
— znormalizowany błąd bezwzględny (normalized absolute error), **NAE**
— znormalizowany błąd średniokwadratowy (normalized mean square error), **NMSE**
— probabilistyczny wskaźnik jakości, **PIF** (Probabilistic Image Fidelity).

Obraz testowy z rys. 1 o rozmiarach 301x301 pikseli o 256. poziomach szarości został skompresowany algorytmem JPEG (oryginał w formacie BMP). Pierwszy z obrazów — dalej oznaczany *A* — skompresowano do 50% (plik o rozmiarach dwukrotnie mniejszych niż oryginał), drugi obraz, *B* — 20% (plik o rozmiarach pięciokrotnie mniejszych niż oryginał). Wyniki testów zebrano w tabeli 1.

Tabela 1

Wyniki testów algorytmu JPEG. Wyróżnione wiersze zawierają dane zgodne z rzeczywistością

Algorithm JPEG test results. Marked lines contain data compatible with reality

	Obraz A (jakość lepsza)	Obraz B (jakość gorsza)	Optymalna wartość wskaźnika jakości	Wnioski
AD	$5{,}97 \cdot 10^{-2}$	$0{,}83 \cdot 10^{-2}$	0	B znacznie lepszy niż A
MD	255	255	min	A = B
IF	-247,37	-96,338	1	B znacznie lepszy niż A
MSE	23927	23891	min	B lepszy niż A
PMSE	$5{,}65 \cdot 10^{-6}$	$5{,}65 \cdot 10^{-6}$	min	A=B
NAE	$5{,}3755 \cdot 10^{-3}$	$5{,}3757 \cdot 10^{-3}$	min	A lepszy niż B
NMSE	248,37	97,338	min	B lepszy niż A
PIF[1]	0,99729	0,99719	1	A lepszy niż B

[1] Wartość współczynnika PIF zawiera się w przedziale od 0 do 1 dla obrazów zupełnie się różniących i identycznych odpowiednio.

Wnioski płynące z przykładu 1 są zbieżne z uwagami wyrażonymi wcześniej. W tym eksperymencie należało się spodziewać, że jakość obrazu A będzie lepsza niż B. Istotnie tak jest, co potwierdza również ocena subiektywna (rys.2.).

Autor sugeruje w tej pracy, jak i w innych artykułach [9,10], że kryterium PIF w bardzo dużym stopniu odzwierciedla wrażenia subiektywne i nie zależy od rodzaju przetwarzanego obrazu, a zatem ma charakter uniwersalnego wskaźnika jakości.

3. ZWIĘKSZENIE „CZUŁOŚCI" PROBABILISTYCZNEJ MIARY JAKOŚCI PIF

Współczynnik PIF, posiadający niewątpliwe zalety, wykazuje pewną, aczkolwiek niezbyt oczywistą wadę. Koncepcja tego kryterium opiera się na analizie cech probabilistycznych sygnału na wyjściu i wejściu algorytmu. Dokładniej rzecz ujmując, wyliczając PIF, bada się w jakim stopniu wielomian

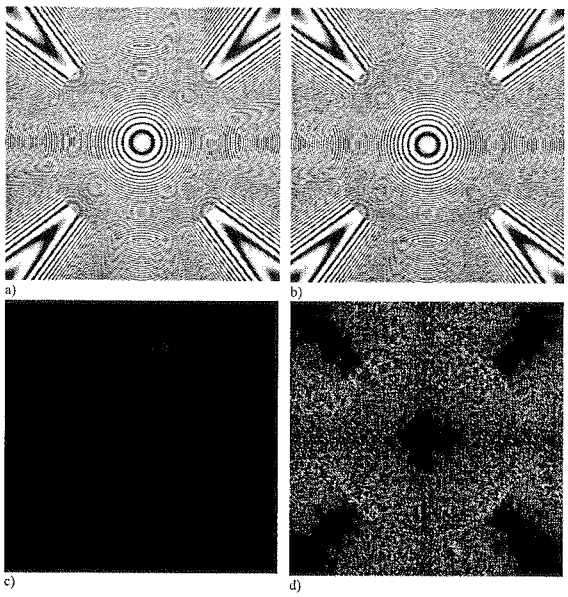
$$G(F) = \sum_{i=0}^{N} c_i \cdot F^i \tag{1}$$

różni się od linii prostej, tj. od

$$G(F) = F \tag{2}$$

gdzie G(x) i F(x) oznaczają dystrybuantę rozkładu (funkcji) prawdopodobieństwa sygnału na wyjściu i wejściu algorytmu odpowiednio, natomiast x — zmienna losowa (dyskretna), w przypadku przetwarzania obrazu poziom szarości: liczba całkowita $x \in \langle 0, 255 \rangle$ dla reprezentacji 8-bitowej. Jak wykazano w [9] postać wielomianu (1) nie zależy od funkcji prawdopodobieństwa sygnału wejściowego.

Podając dla konkretnego algorytmu wielomian (1) można określić w jakim stopniu algorytm ten będzie zaburzał strukturę sygnału wejściowego, co w postaci odpowied-



Rys. 2. Wyniki przetwarzania z przykładu 1: a) obraz skompresowany dwukrotnie, b) obraz po kompresji pięciokrotnej; c) różnice między oryginałem z rys.1 a obrazem z rys. 2 a, d) różnice między oryginałem z rys.1 a obrazem z rys. 2 b

Fig. 2. Results of processing from 1 example: a) image twice compressed, b) image compressed five times; c) differences between original from fig.1 and image from fig.2 a, d) differences between original from fig.1 and image from fig.2 b

niego współczynnika (miary) jakości oddaje, wspomniany wyżej, PIF (Probabilistic Image Fidelity) [10]:

$$PIF = 1 - const \cdot \int_0^1 [G(F) - F]^2 \, dF \tag{3}$$

Współczynnik *const* w oryginalnej postaci wzoru (3) ma wartość 12. Wartość tę wyliczono przy założeniu, że dla algorytmu o najgorszych właściwościach (PIF=0) spełniona jest następująca zależność:

$$G(F) = \begin{cases} 0 \; dla \; 0 \leqslant F < 0,5 \\ 1 \; dla \; 0,5 \leqslant F \leqslant 1 \end{cases} \tag{4}$$

Przypadek taki ma miejsce w przypadku rozpatrywania ważonych filtrów medianowych i o takim jest mowa w [10].

Żądając natomiast, aby PIF był równy zero dla trywialnego filtru zerującego obraz wyjściowy, dla którego G(F) = 1 przy $0 \leq F \leq 1$, należy przyjąć stałą *const* = 3.

Autor wykazał w [10], że miarę PIF można wyznaczyć eksperymentalnie, nie zawsze bowiem możliwe jest analityczne obliczenie współczynników wielomianu (1).

Chcąc wykazać pewien zakres „nieczułości" miary PIF proponuje się wykonanie następującego eksperymentu:

Przykład 2.

Działanie pewnego algorytmu polega na wprowadzeniu w przetwarzanym obrazie efektu specjalnego typu „wir". Oryginał obrazu i wynik przetwarzania pokazano na rys. 3. Określając dla rozpatrywanego algorytmu współczynnik PIF okazało się, iż wynosi on dokładnie 1. Należało się tego spodziewać, bowiem badany algorytm tak skonstruowano, że nie zmienia funkcji prawdopodobieństwa sygnału wejściowego. W efekcie również dystrybuanta na wejściu algorytmu jest równa dystrybuancie wyjściowej, a zatem G(F)=F, czyli na mocy (3) PIF=1.

Proponuje się zatem „uczulenie" miary PIF na tego typu zniekształcenia przez wprowadzenie następującej modyfikacji:

$$PIF^{+} = \frac{Q+1}{2} \cdot PIF \tag{5}$$

lub

$$PIF^{+} = \varepsilon \cdot PIF \tag{5a}$$

Definicja nowego czynnika Q jest następująca:

Definicja 1.

Dla dwóch obrazów $X = \{x_{i,j} | i = 1,2,\ldots,N; j = 1,2,\ldots,M\}$ i $Y = \{y_{i,j} | i = 1,2,\ldots,N; j = 1,2,\ldots,M\}$ wejściowego i wyjściowego odpowiednio, definiuje się czynnik Q następująco:

$$Q = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y} \tag{6}$$

gdzie:

$$\sigma_{xy} = \frac{1}{N-1} \cdot \frac{1}{M-1} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} (x_{i,j} - \overline{x}) \cdot (y_{i,j} - \overline{y}) \tag{7}$$

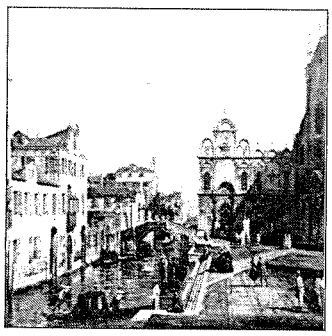
$$\sigma_x^2 = \frac{1}{N-1} \cdot \frac{1}{M-1} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} (x_{i,j} - \overline{x})^2 \tag{8}$$

$$\sigma_y^2 = \frac{1}{N-1} \cdot \frac{1}{M-1} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} (y_{i,j} - \overline{y})^2 \tag{9}$$

$$\overline{x} = \frac{1}{N} \cdot \frac{1}{M} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j} \tag{10}$$

$$\overline{y} = \frac{1}{N} \cdot \frac{1}{M} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j} \tag{11}$$

a)



b)

Rys. 3. Obraz oryginalny a) i wynik przetwarzania b) algorytmem badanym w przykładzie 2

Fig. 3. Original image a) and result of processing b) with algorithm researched in 2 example

Nietrudno zauważyć, że współczynnik Q określa korelację między obrazami [1]. Ponieważ Q może przyjmować wartości od –1 do 1, a nadal żądamy (nawiązując do założeń sformułowanych w [10]), aby współczynnik określający jakość przetwarzania zmieniał się w zakresie od 0 do 1, stąd we wzorze (5) przed czynnikiem PIF występuje $\frac{Q+1}{2}$, a nie Q. Obecnie dla algorytmu przetwarzania i obrazów z przykładu 2 wartość miary PIF+=0,8857, co można uznać za zgodne z oceną subiektywną. Wykonując podobne obliczenia dla przykładu 1 współczynnik PIF^+ wynosi: dla obrazu A 0.99408, dla obrazu B 0.98462. Miara PIF^+ skupia w postaci jednej wartości liczbowej jakość odwzorowania histogramu (jasność, kontrast — PIF) oraz precyzję odwzorowania treści

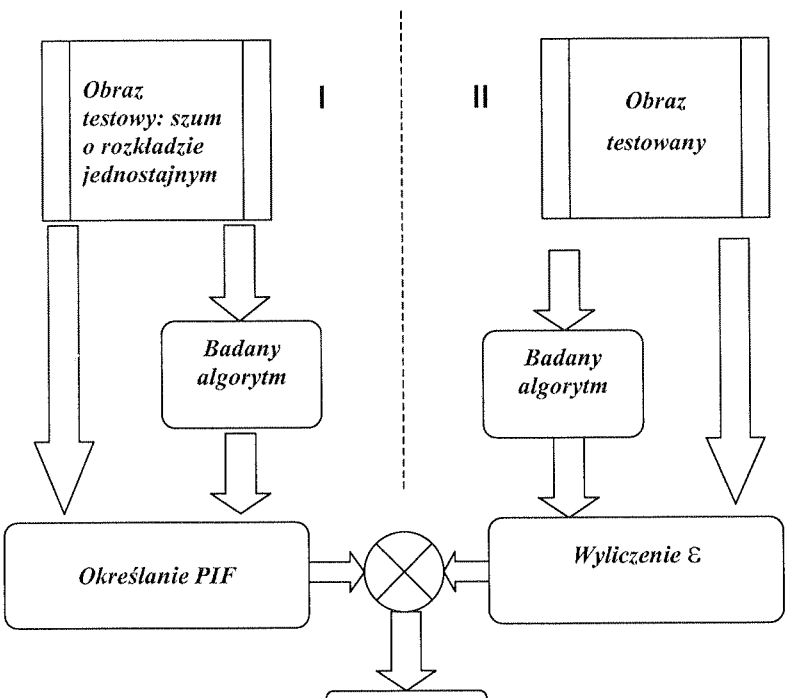
[1] Funkcja licząca Q dostępna jest Matlabie pod nazwą corr2

obrazu (czynnik $(Q + 1)/2$). Chcąc zwiększyć czytelność miary PIF^+ proponuje się następującą jej prezentację:

— w przykładzie 1:
dla obrazu A: $PIF^+ = \varepsilon \cdot PIF = 0.99678 \cdot 0.99729 = 0.99408$,
dla obrazu B: $PIF^+ = \varepsilon \cdot PIF = 0.98739 \cdot 0.99719 = 0.98462$,
— w przykładzie 2: $PIF^+ = \varepsilon \cdot PIF = 0.8857 \cdot 0 = 0.8857$.

Sugerowany sposób zapisu w sposób jednoznaczny określa, w jakim stopniu testowany algorytm zniekształca histogram (niezależnie od rodzaju obrazu) i na ile dokładnie została odwzorowana struktura badanego obrazu.

Wyliczanie współczynnika PIF^+ odbywa się zatem dwuetapowo: etap I to wyliczenie współczynnika PIF [10], gdzie obrazem testowym jest szum o rozkładzie jednostajnym, etap II — obliczenia związane z określeniem ε (5, 5a) dla obrazu, którego jakość ma być porównana z oryginałem. Proces ten pokazano na rys. 4.



Rys. 4. Schemat blokowy algorytmu liczenia współczynnika PIF^+

Fig. 4. Block diagram algorithm of calculating PIF^+ coefficient

4. WNIOSKI

Celem autora było zdefiniowanie uniwersalnej miary szacowania jakości obrazów cyfrowych. Rozważania przedstawione w niniejszej pracy oparto o definicję wcześniej opracowanej przez autora miary PIF. W artykule wskazano na pewien zakres zagadnień, w których współczynnik PIF wykazuje tylko niewielką czułość. Jak wykazały badania, Probabilistic Image Fidelity to kryterium bardzo dobrze oddające ocenę subiektywną w przypadku badania jakości algorytmów DSP (i obrazów przez nie generowanych), które zmieniają funkcję prawdopodobieństwa sygnału wejściowego. Autor wykazał m.in. w [10], że PIF jest godnym polecenia narzędziem w przypadku szacowania jakości algorytmów filtracji nieliniowej. Z rozważań zawartych w tej pracy wynika natomiast, iż dla obrazów generowanych przez algorytmy, które funkcji prawdopodobieństwa nie zmieniają (przykład 2), „czułość" współczynnika PIF jest niedostateczna. W związku z tym uzupełniono definicję PIF o dodatkowy czynnik, który rozszerza zakres zastosowań tej miary.

Zatem, definiując PIF^+ uzyskano kryterium o właściwościach, które umożliwiają jego stosowanie w następujących przypadkach:

a) obraz wyjściowy analizowanego algorytmu zmienia geometrię względem wejściowego przy zachowaniu funkcji prawdopodobieństwa (histogramu)
b) zmienia się funkcja prawdopodobieństwa (histogram) przy zachowaniu geometrii
c) jednocześnie zachodzi a) i b)

Miara PIF^+ nie jest „czuła" w przypadkach, gdy analizowane obrazy są liniowo zależne, tzn. $y_{i,j} = a \cdot x_{i,j} + b$ dla wszystkich $i = 1,2,\ldots,N$, $j = 1,2,\ldots,M$ (y i x elementy macierzy reprezentujących obraz wyjściowy i wejściowy, a i b stałe, $a > 0$). Trudno jednak tę właściwość uznać za wadę, bowiem tego rodzaju zniekształcenia to przesunięcie wzajemne obrazów. Oceniając subiektywnie przesunięte obrazy (zwykle bardzo niewiele) nie odnosi się wrażenia pogorszenia jakości.

5. BIBLIOGRAFIA

1. T. Thong: *Digital image processing test patterns*, IEEE Trans. Acoust., Seech Signal Processing, vol. ASSP — 31, Jun. 1983.
2. E.J. Coyle, J.H. Lin: *Stack filters and the mean absolute error criterion*, IEEE Transactions on Acoustic, Speech and Signal Processing, vol. ASSP-36, nr 4, sierpień 1988, pp. 1244–1254.
3. J. Woźnicki: *Podstawowe techniki przetwarzania obrazu*, WKŁ Warszawa 1996.
4. O. Yli-Harja, J. Astola, Y. Neuvo: *Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation*, IEEE Transactions on Acoustic, Speech and Signal Processing, vol. ASSP-39, nr 2, luty 1991, pp. 395–410.
5. J. Purczyński, E. Kornatowski: *Wybrane problemy projektowania cyfrowych filtrów medianowych*, XX SPETO'97, 21–24 maj 1997, Tom 2, ss. 465–469.
6. B.I. Justusson: *Median filtering: Statistical properties*, Topics in Applied Physics. Two-Dimensional Digital Signal Processing II, T. S. Huang, Ed. Berlin: Springer, 1981.

7. T. S. Huang: *Two — Dimensional Digital Signal Processing II. Transforms and Median Filters*, Springer Verlag, 1981.
8. E. Kornatowski: *Optymalizacja złożonych algorytmów nieliniowych filtrów cyfrowych*, IV Konferencja Naukowo — Techniczna ZKwE 1999, ss. 43–46.
9. E. Kornatowski: *Problemy oceny jakości odwzorowania sygnałów*, Materiały XXIII Międzynarodowej Konferencji z Podstaw Elektrotechniki i Teorii Obwodów SPETO-2000, pod patronatem PAN, Gliwice- Ustroń,24–27 maj 2000, ss. 415–418.
10. E. Kornatowski: *Probabilistyczna miara wierności odwzorowania sygnału*, Kwartalnik Elektroniki i Telekomunikacji, Polska Akademia Nauk, PWN Warszawa 1999, zeszyt 3–4, ss. 297–309.

E. KORNATOWSKI

A UNIVERSAL DIGITAL IMAGE QUALITY INDEX

Summary

In an article is proposed a new universal objective quality index, which is easy to calculate and applicable to various image processing applications. There is described a new definition of input-to-output signal projection fidelity coefficient for any DSP algorithm, particularly for non-linear digital filter. The probabilistic fidelity coefficient PIF (worked out by the author and presented in previous publications [10]) is based on a function that assigns an algorithm output with input distributions — (3). In its basic definition this criterion did not take into account some disorders in the structure of the image. This paper contains results of the research into possibilities of enlarging effectiveness of PIF coefficient.

New PIF coefficient is described by (5) formula. Calculating PIF coefficient is possible by going through two steps:

1. calculating PIF coefficient [10], where uniform noise is a testing image,
2. calculating connected with defining ? (5, 5a) for an image, which quality should be comparable with original. The process is shown on fig. 4.

Defining PIF we get criteria with attributes which make using this coefficient possible in following situations:

a) analyzing algorithm output image changes geometry towards input one with probability density function (histogram) remaining unchanged,
b) probability density function (histogram) changes with geometry remaining unchanged,
c) a) and b) happens at the same time

Keywords: digital filters, digital signal processing

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 521–543

Dekompozycja wierszowa w syntezie logicznej przeznaczonej dla struktur matrycowych*

DARIUSZ KANIA

Politechnika Śląska, Instytut Elektroniki,
ul. Akademicka 16, 44-100 Gliwice
e-mail: dkania@polsl.pl

Otrzymano 2005.06.29
Autoryzowano 2005.12.07

Jądrem większości układów matrycowych (ang. CPLD — Complex Programmable Logic Devices) jest blok logiczny typu PAL. Celem artykułu jest przedstawienie nowej metody dekompozycji dedykowanej dla struktur matrycowych typu PAL. Zaproponowana metoda jest alternatywą dla metody klasycznej opartej na dwupoziomowej minimalizacji.

Istota metody polega na sekwencyjnym wyszukiwaniu odpowiedniej dekompozycji wierszowej. Poszczególne kroki procesu syntezy zostały dostosowane do zasobów logicznych struktur matrycowych typu PAL. Dostosowanie dekompozycji do zasobów logicznych scharakteryzowanych przez blok logiczny typu PAL doprowadziło do znaczącej poprawy efektywności syntezy w porównaniu z metodą klasyczną.

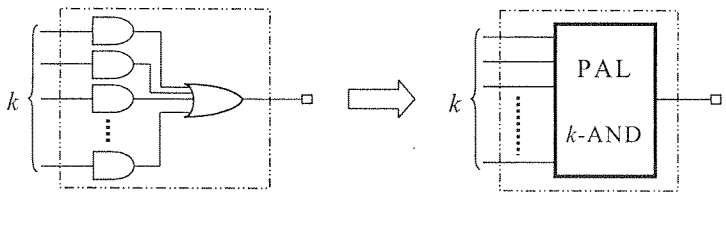
Wyniki eksperymentów pokazują, że proponowana metoda syntezy prowadzi do znaczącej redukcji powierzchni wykorzystywanego układu w porównaniu z metodą klasyczną, szczególnie dla struktur matrycowych składających się z bloków logicznych typu PAL zawierających 4 lub 8 iloczynów.

Słowa kluczowe: synteza logiczna, dekompozycja, podział, układy matrycowe typu PAL

1. WPROWADZENIE

Bloki logiczne zawarte w większości układów matrycowych (ang. CPLD — Complex Programmable Logic Devices) wykorzystują strukturę typu PAL (rys.1). Tego typu układy zwane będą układami matrycowymi typu PAL w odróżnieniu od drugiej mało popularnej grupy układów matrycowych typu PLA.

* Pracę wykonano w ramach projektu badawczego 3T11B04027 finansowanego przez MNiI



Rys. 1. Struktura bloku logicznego typu PAL zawierającego *k*-iloczynów

Fig. 1. Structure of PAL-based logic block consisting of *k* terms

Klasyczna metoda syntezy układów realizowanych w strukturach matrycowych typu PAL obejmuje dwupoziomową minimalizację wykonywaną dla każdej funkcji oddzielnie, po której następuje etap realizacji zminimalizowanych funkcji na blokach logicznych typu PAL zawierających k-iloczynów. W przypadku realizacji zminimalizowanych funkcji, będących sumą p implikantów zachodzi potrzeba wykorzystywania większej liczby bloków. Klasyczna ekspansja iloczynów polega na wprowadzaniu sprzężeń zwrotnych zwiększających czas propagacji sygnału od wejść do wyjść.

Celem niniejszego artykułu jest przedstawienie alternatywnej metody realizacji funkcji w strukturach matrycowych typu PAL wykorzystującej tzw. dekompozycję wierszową.

Dekompozycja odgrywa niezwykle istotną rolę w nowoczesnej syntezie logicznej. Jest ona przede wszystkim wykorzystywana w syntezie przeznaczonej dla struktur FPGA typu tablicowego. Pierwsze algorytmy syntezy MIS-PGA, Chortle, Asyl przeznaczone dla struktur bramkowych zostały również wykorzystane w procesie syntezy, przeznaczonej dla układów FPGA typu tablicowego. Algorytmy te wykorzystują między innymi dekompozycję Roth-Karpa, technikę wyłączania tzw. rdzeni (ang. kernel extraction), faktoryzację wyrażeń boolowskich. Wspomagane są one różnorodnymi procedurami grupowania, leksykograficznego porządkowania zmiennych, programowaniem dynamicznym itp. Bardzo często poszczególne kroki dekompozycji polegają na iteracyjnym podziale sieci logicznej, odpowiadającej wyrażeniu booleowskiemu, powstającemu w wyniku dwupoziomowej minimalizacji. Podział sieci logicznej związany jest z odpowiednim wyborem węzłów, gałęzi itd. W kilku kolejnych latach powstało szereg rozwinięć pierwotnych koncepcji, przeznaczonych przede wszystkim dla struktur FPGA typu tablicowego lub multiplekserowego. Równolegle powstawały i rozwijały się metody syntezy, których podstawą jest klasyczny model dekompozycji funkcjonalnej, zaproponowany przez Ashenhursta [2], a rozszerzony przez Curtisa [5]. Model ten stanowi fundament szeregu algorytmów dekompozycji, które wykorzystują często różnorodne strategie kolorowania grafów [6, 8, 10], transformacji funkcji [10], łączenia klasycznej dekompozycji z procedurami podziału wyjść itd. Wydaje się, że jedną z zalet tych metod syntezy jest uwzględnianie stanów nieokreśloności w początkowym etapie syntezy, najczęściej związanym z dekompozycją.

W ostatnich latach dużym zainteresowaniem cieszą się metody syntezy wykorzystujące binarne diagramy decyzyjne (ang. Binary Decision Diagram). W literaturze

pojawiło się szereg prac dotyczących różnorodnych odmian binarnych diagramów decyzyjnych. Związane są one przede wszystkim z efektywnością reprezentacji funkcji, porządkowaniem zmiennych, faktoryzacją wyrażeń booleowskich oraz elementami dekompozycji, bazującymi na odpowiednich „cięciach" różnych form binarnych diagramów decyzyjnych.

Ciągły rozwój różnorodnych modeli dekompozycji prowadzi do powstawania coraz to lepszych algorytmów syntezy, przedstawionych w bardzo licznych pracach. Mimo burzliwego i wielotorowego rozwoju tematyki związanej z dekompozycją optymalny sposób podziału układu cyfrowego na bloki logiczne zawarte w strukturach programowalnych na obecnym poziomie wiedzy ciągle nie jest jednak znany. Najczęściej dekompozycja wykorzystywana jest w syntezie przeznaczonej dla układów FPGA, stanowiąc nieodzowny element syntezy umożliwiający podział projektowanego układu na konfigurowalne bloki logiczne. Znacznie rzadziej jest wykorzystywana w syntezie logicznej, przeznaczonej dla innych typów układów programowalnych. Znane są algorytmy syntezy wykorzystujące dekompozycję, przeznaczone dla matryc PLA [4]. W niektórych metodach syntezy bezpośrednio zaadaptowano algorytmy dekompozycji, opracowane dla układów FPGA typu tablicowego. Cechą charakterystyczną tego typu algorytmów jest występujący w nich proces odpowiedniego kodowania wyjść oraz wejść, znacząco wpływający na minimalizację liczby iloczynów wykorzystywanych w blokach uzyskiwanych w wyniku dekompozycji. Problemy odpowiedniego kodowania wejść i wyjść są szeroko dyskutowane w powiązaniu z zagadnieniami kodowania stanów wewnętrznych automatów sekwencyjnych [1]. Między innymi związane są one bezpośrednio z symbolicznym kodowaniem stanów wewnętrznych, teorią dychotomii, minimalizacją funkcji wielowartościowych, analizą relacji dominacji wyjść itp. [1].

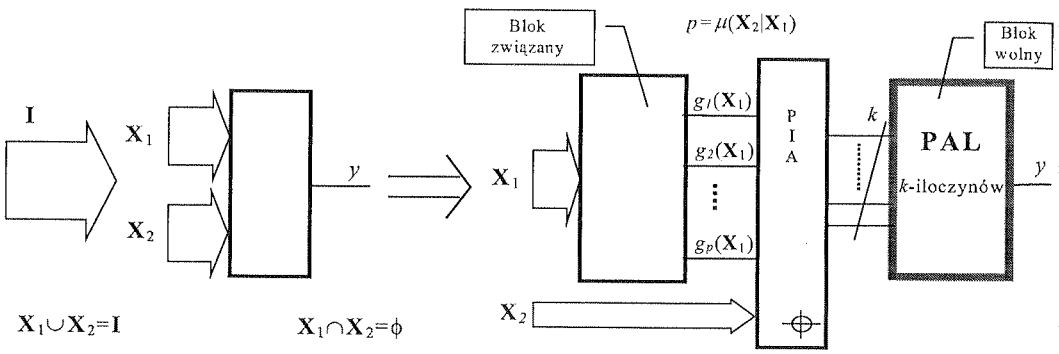
Zwykle istota dekompozycji funkcji sprowadza się do odpowiedniego podziału projektowanego układu logicznego na podukłady o zadanej liczbie wejść i wyjść (realizacja układu cyfrowego w strukturach FPGA typu tablicowego zawierającego konfigurowalne bloki logiczne). Dekompozycja może być również wykorzystywana w innych przypadkach, np. do podziału projektu na części, prowadząc do wielopoziomowej realizacji w postaci sieci prostych układów PLD [6, 9].

Bloki logiczne typu PAL zawarte w strukturach CPLD mają znacznie większą liczbę wejść niż konfigurowalne bloki logiczne zawarte w strukturach FPGA lub proste układy PLD. Zwykle do każdego iloczynu wchodzącego w skład bloku logicznego typu PAL można dołączyć sygnał z dowolnego wejścia lub sprzężenia zwrotnego, co ogranicza znaczenie dekompozycji jako narzędzia stanowiącego podstawę podziału projektu na poszczególne bloki logiczne. Dodatkowo projektując układ cyfrowy implementowany w strukturach programowalnych dąży się zawsze do odpowiedniego doboru struktury, aby zapewnić realizowalność całego projektu w jednym układzie. Szeroka oferta struktur CPLD w większości przypadków umożliwia dobór jednej struktury do konkretnego zadania. Powoduje to dalsze ograniczenie znaczenia dekompozycji jako podstawy podziału projektu na poszczególne układy.

Głównym ograniczeniem bloków logicznych typu PAL jest liczba wielowejściowych iloczynów zawartych w jednym bloku. Fakt ten sprawia, że istota dekompozycji przeznaczonej dla struktur typu PAL sprowadza się do dwóch głównych zadań:

1. minimalizacji liczby wykorzystywanych bloków logicznych typu PAL;
2. dopasowania projektowanego układu do struktury bloków logicznych typu PAL.

Zaproponowany model dekompozycji (dekompozycja wierszowa) dotyczy bezpośrednio drugiego zagadnienia. Istota dekompozycji wierszowej (rys.2) sprowadza się do znalezienia takiego podziału układu (dekompozycji funkcji), który umożliwi realizację bloku wolnego na bloku logicznym typu PAL zawierającym założoną liczbę iloczynów.



Rys. 2. Istota dekompozycji wierszowej

Fig. 2. The essence of the row decomposition

2. PODSTAWY TEORETYCZNE

Funkcja $y = f(i_n, \ldots, i_2, i_1) = f(X_2, X_1)$ podlega dekompozycji prostej, tzn. $f(X_2, X_1) = F[g(X_1), X_2]$ wtedy i tylko wtedy, gdy złożoność kolumnowa matrycy podziałów (siatki Karnaugha) $\nu(X_2/X_1) \leq 2$ [2, 5]. Zbiory X_1 i X_2 nazywane są odpowiednio zbiorem związanym i wolnym, przy czym $X_1 \cup X_2 = \{i_n, \ldots, i_2, i_1\}$ oraz $X_1 \cap X_2 = \phi$.

Pierwotne twierdzenie o dekompozycji prostej zostało rozszerzone, stanowiąc podstawę dekompozycji funkcjonalnej zespołu funkcji.

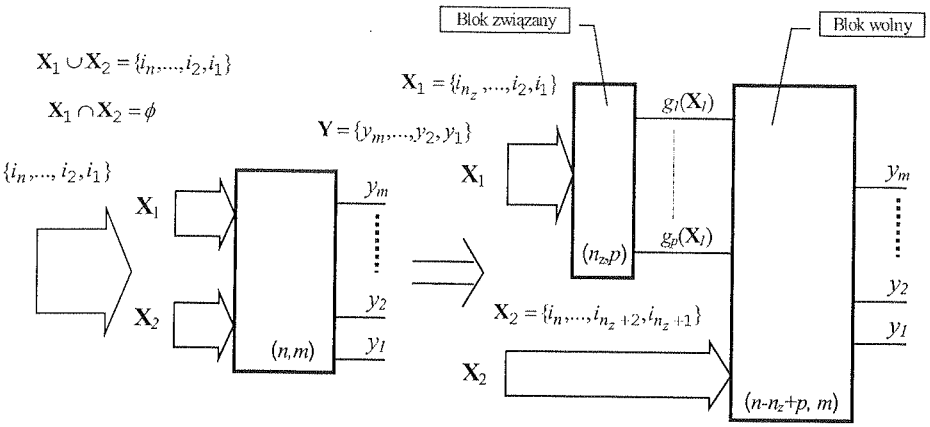
Funkcja $f : B^n \rightarrow B^m$ podlega dekompozycji wtedy i tylko wtedy, gdy złożoność kolumnowa matrycy podziałów (siatki Karnaugha) $\nu(\mathbf{X}_2|\mathbf{X}_1) \leq 2^p$, tzn.

$$\nu(\mathbf{X}_2|X_1) \leqslant 2^p \Leftrightarrow f(\mathbf{X}_2, X_1) = F\left[g_1(X_1), g_2(X_1), \ldots, g_p(X_1), (X_2)\right], \qquad (1)$$

przy czym $\mathbf{X}_1 \cap \mathbf{X}_2 = \{i_n, \ldots, i_2, i_1\}$ oraz $\mathbf{X}_1 \cup \mathbf{X}_2 = \phi$.

Wykorzystanie takiego modelu dekompozycji prowadzi do podziału wielowyjściowego bloku scharakteryzowanego uporządkowaną parą liczb (liczba wejść, liczba

wyjść), wynoszącą (n, m), na dwa inne o parametrach (n_z, p) i $(n - n_z + p, m)$. Podział układu odpowiadający dekompozycji zespołu funkcji przedstawiono na rys. 3.



Rys. 3. Rozłączna dekompozycja zespołu funkcji

Fig. 3. Disjoint decomposition of a multi-output function

Powyższy podział wykonywany jest najczęściej w celu dopasowania liczby wejść poszczególnych bloków do liczby wejść konfigurowanych bloków tablicowych występujących w strukturze programowalnej. Po wprowadzeniu pewnych dodatkowych elementów może prowadzić do minimalizacji sumarycznej liczby iloczynów pośrednio wpływając na minimalizację liczby bloków typu PAL [6, 7]. Cenne byłoby wplecenie w etap dekompozycji elementu dopasowania realizowanych funkcji do struktury bloków logicznych typu PAL. W celu przedstawienia sposobu poszukiwania takiej dekompozycji konieczne jest wprowadzenie niezbędnych określeń. Zróbmy to korzystając z prostego przykładu.

Przykład 1

Rozpatrzmy funkcję $f : B^6 \rightarrow B$, której siatka Karnaugha przedstawiona jest na rys. 4.

Wzorcem wiersza siatki Karnaugha nazywany wiersz opisany przez funkcję $g(\mathbf{X}_1)$, przyjmującą wartość jeden dla wektorów skojarzonych z kolumnami, dla których funkcja $f(\mathbf{X}_2, \mathbf{X}_1)$ przyjmuje wartość jeden. Przykładowo wzorzec pierwszego wiersza oznaczony literą A opisany jest przez funkcję $g_1(\mathbf{X}_1) = g_1(d, e, h) = \bar{d}\bar{h} + \bar{d}e + e\bar{h} + d\bar{e}h$.

Wierszem pełnym, oznaczanym przez symbol **1**, nazywamy wzorzec wiersza, dla którego $g(\mathbf{X}_1) = 1$. ***Wierszem pustym***, oznaczanym przez symbol **0**, nazywamy wzorzec wiersza, dla którego $g(\mathbf{X}_1) = 0$. ***Dopełnieniem wzorca wiersza*** A, oznaczanym jako $\overline{A}$, nazywamy wzorzec wiersza opisany przez funkcję $\overline{g}(\mathbf{X}_1)$. Analiza wierszy przykładowej siatki Karnaugha prowadzi do wyznaczenia wzorców wierszy, przyporządkowując poszczególnym grupom wierszy odpowiednie symbole, przedstawione na rys. 4.

X1={d,e,h}, X2={a,b,c}

abc \ deh	000	001	011	010	110	111	101	100	
000	1	0	1	1	1	0	1	0	A
001	1	0	1	0	1	1	1	0	B
011	0	1	0	0	0	1	0	1	$\overline{A}$
010	1	0	1	1	1	0	1	0	A
110	1	0	1	1	1	0	1	0	A
111	1	0	1	0	1	1	1	0	B
101	0	0	0	0	0	0	0	0	0
100	1	1	1	1	1	1	1	1	1

$A \rightarrow g_1(d,e,h) = \overline{d}\overline{h} + \overline{d}e + e\overline{h} + d\overline{e}h$

$B \rightarrow g_2(d,e,h) = eh + de + dh + \overline{d}\overline{e}\overline{h}$

Rys. 4. Siatka Karnaugha z symbolami rozróżniającymi poszczególne wzorce wierszy

Fig. 4. The Karnaugh map with symbols, which distinguish individual row patterns

W ogólnym przypadku zbiór wierszy można podzielić na kilka grup: wiersze puste, wiersze pełne oraz wiersze skojarzone z identycznym wzorcem wiersza lub jego dopełnieniem (np. A lub $\overline{A}$).

Złożonością wierszową siatki Karnaugha (matrycy podziałów) $\mu(\mathbf{X}_2|\mathbf{X}_1)$ nazywamy liczbę różnych grup wierszy, z wyłączeniem grup zawierających wiersze puste i wiersze pełne. Przykładowo dla funkcji przedstawionej na rys. 4 złożoność wierszowa $\mu(\mathbf{X}_2|\mathbf{X}_1) = 2$.

Jeżeli $\mu(\mathbf{X}_2|\mathbf{X}_1) = p$, to zbiór wierszy można podzielić na $p + 2$ grup:

- grupę zawierającą wiersze puste,
- grupę zawierającą wiersze pełne,
- p-grup zawierających wiersze skojarzone z jednym wzorcem wiersza lub jego dopełnieniem.

Każdej grupie wierszy można przyporządkować funkcję, której argumentami są zmienne zbioru wolnego. Niech

- $H_0(\mathbf{X}_2) = \begin{cases} 1, & \text{dla wierszy pełnych} \\ 1, & \text{dla pozostaych wierszy} \end{cases}$
- $H_i(\mathbf{X}_2) = \begin{cases} 1, & \text{dla wierszy, dla których wzorzec wiersza jest opisany przez funkcję } g_i(\mathbf{X}_1) \\ 1, & \text{dla pozostaych wierszy} \end{cases}$

gdzie $i = 1, 2, \ldots, p$

- $H_i'(\mathbf{X}_2) = \begin{cases} 1, & \text{dla wierszy, dla których wzorzec wiersza jest opisany przez funkcję } \overline{g}_i(\mathbf{X}_1) \\ 1, & \text{dla pozostaych wierszy} \end{cases}$

gdzie $i = 1, 2, \ldots, p$

Zgodnie z powyższymi oznaczeniami funkcja $f(\mathbf{X}_2, \mathbf{X}_1)$ może być przedstawiona w postaci :

$$f(\mathbf{X}_2, \mathbf{X}_1) = H_0(\mathbf{X}_2) + \sum_{i=1}^{p} \left[H_i(\mathbf{X}_2)\, g_i(\mathbf{X}_1) + H_i'(\mathbf{X}_2)\, \overline{g}_i(\mathbf{X}_1) \right]$$

Podsumowując powyższe rozważania można stwierdzić, że

$$\mu(\mathbf{X}_2|\mathbf{X}_1) = p \Rightarrow f(\mathbf{X}_2, \mathbf{X}_1) = H_0(\mathbf{X}_2) + \sum_{i=1}^{p} \left[H_i(\mathbf{X}_2)\, g_i(\mathbf{X}_1) + H_i'(\mathbf{X}_2)\, \overline{g}_i(\mathbf{X}_1) \right] \quad (2)$$

Przykład 2

Rozpatrzmy ponownie funkcję z przykładu 1. Wyznaczmy dla rozważanego podziału zmiennych ($\mathbf{X}_1 = \{d, e, h\}$, $\mathbf{X}_2 = \{a, b, c\}$) funkcje $H_i(\mathbf{X}_2)$. Funkcje te przyjmują następujące postacie:

- $H_0(\mathbf{X}_2) = a\bar{b}\bar{c}$
- $H_1(\mathbf{X}_2) = \bar{a}\bar{c} + b\bar{c}$; $H_1'(\mathbf{X}_2) = \bar{a}bc$
- $H_2(\mathbf{X}_2) = \bar{a}\bar{b}c + abc$; $H_2'(\mathbf{X}_2) = 0$

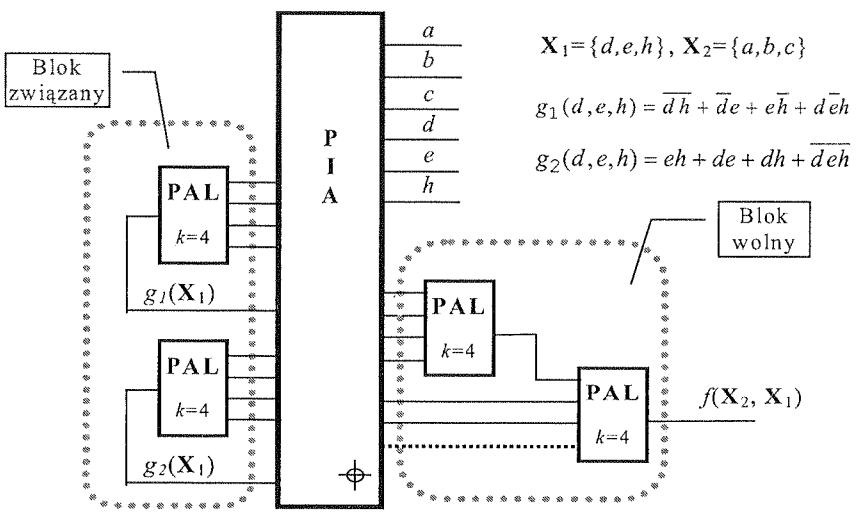
Znając funkcje $H_i(\mathbf{X}_2)$ oraz funkcje $g_i(\mathbf{X}_1)$ (rys. 4), możliwe jest, korzystając z zależności (2), przedstawienie funkcji $f(\mathbf{X}_2, \mathbf{X}_1)$ w następującej postaci:

$f(\mathbf{X}_2, \mathbf{X}_1) = H_0(\mathbf{X}_2) + H_1(\mathbf{X}_2)g_1(\mathbf{X}_1)\bar{g}_1(\mathbf{X}_1) + H_2(\mathbf{X}_2)g_2(\mathbf{X}_1) + H_2'(\mathbf{X}_2)\bar{g}_2(\mathbf{X}_1) = a\bar{b}\bar{c} +$
$+(\bar{a}\bar{c}+b\bar{c})(\overline{dh}+\bar{d}e+e\bar{h}+d\bar{e}\bar{h})+\bar{a}bc(\overline{\overline{dh}+\bar{d}e+e\bar{h}+d\bar{e}\bar{h}})+(\bar{a}\bar{b}c+abc)(eh+de+dh+\overline{d\bar{e}\bar{h}})$,

tzn.:

$$f(\mathbf{X}_2, \mathbf{X}_1) = a\bar{b}\bar{c} + (\bar{a}\bar{c} + b\bar{c})g_1(\mathbf{X}_1) + \bar{a}bc\bar{g}_1(\mathbf{X}_1) + (\bar{a}\bar{b}c + abc)g_2(\mathbf{X}_1)$$

Realizacja funkcji wykorzystująca bloki logiczne typu PAL przedstawiona jest na rys. 5.



$$f(\mathbf{X}_2, \mathbf{X}_1) = a\bar{b}\bar{c} + (\bar{a}\bar{c} + b\bar{c})g_1(\mathbf{X}_1) + \bar{a}bc\,\bar{g}_1(\mathbf{X}_1) + (\bar{a}\bar{b}c + abc)g_2(\mathbf{X}_1)$$

Rys. 5. Realizacja funkcji oparta na dekompozycji wierszowej

Fig. 5. Implementation of the function based on the row decomposition

3. DEKOMPOZYCJA WIERSZOWA

Istotą dekompozycji wierszowej jest poszukiwanie takiego podziału argumentów funkcji, który zapewni realizowalność bloku wolnego w jednym bloku logicznym typu PAL o zadanej liczbie iloczynów, zapewniając równocześnie minimalizację liczby wyjść bloku związanego. Ponieważ liczba wyjść bloku związanego jest równa złożoności wierszowej siatki Karnaugha ($p = \mu(\mathbf{X}_1|\mathbf{X}_2)$), stąd jednym z podstawowych problemów występujących w początkowym etapie poszukiwania odpowiedniej dekompozycji wierszowej jest problem wyznaczenia złożoności wierszowej dla analizowanych podziałów zmiennych wejściowych.

3.1. METODA WYZNACZANIA ZŁOŻONOŚCI WIERSZOWEJ

W niniejszym podrozdziale zostanie zaprezentowany algorytm wyznaczania złożoności wierszowej, wykorzystujący specyficzne kolorowanie wierzchołków grafu niezgodności i dopełnień wierszy. Konieczne jest wprowadzenie dodatkowych pojęć.

Parę komórek (i, j) należących do tej samej kolumny siatki Karnaugha nazywamy niezgodną, jeżeli wartości funkcji w nich zawarte wynoszą (1,0) lub (0,1). Jeżeli w zbiorze par komórek należących do dwóch wierszy wystąpi przynajmniej jedna para komórek niezgodnych, to takie wiersze nazywamy ***wierszami niezgodnymi***. Jeżeli w zbiorze par komórek należących do dwóch różnych wierszy nie występują pary (1,1) i (0,0), to mówimy, że jeden wiersz jest ***dopełnieniem*** drugiego.

Grafem niezgodności i dopełnień wierszy nazywamy graf $G(\mathbf{Y}, \mathbf{U})$, gdzie $\mathbf{Y}$ jest zbiorem wierzchołków odpowiadających wierszom, natomiast $\mathbf{U} = \mathbf{U}_n \cup \mathbf{U}_d$ zbiorem krawędzi, gdzie:

$\mathbf{U}_n$ — jest zbiorem krawędzi łączących wierzchołki odpowiadające wierszom niezgodnym z wyłączeniem wierszy, w których jeden jest dopełnieniem drugiego oraz wierszy pustych i pełnych,

$\mathbf{U}_d$ — jest zbiorem krawędzi łączących wierzchołki odpowiadające wierszom, w których jeden jest dopełnieniem drugiego z wyłączeniem wierszy pustych i pełnych.

Krawędzie należące do zbioru $\mathbf{U}_n$ będą oznaczane linią ciągłą, natomiast krawędzie należące do zbioru $\mathbf{U}_d$ linią przerywaną.

Przykład 3

Wyznaczmy graf niezgodności i dopełnień wierszy dla funkcji $f : B^6 \rightarrow B$ przedstawionej na rys. 6a. W pierwszej kolejności wykrywane są wiersze puste i pełne, które są eliminowane z dalszych rozważań. W przypadku analizowanej funkcji występują tylko dwa wiersze puste, oznaczone na rys. 6a) obok wiersza symbolem **0**.

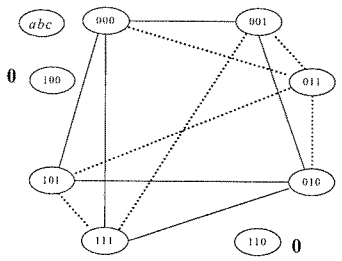
Analiza poszczególnych par wierszy pozwala utworzyć graf niezgodności i dopełnień wierszy przedstawiony na rys. 6b).

Zaproponowane w dalszej części kolorowanie wierzchołków grafu niezgodności i dopełnień wierszy pozwala wyznaczyć złożoność wierszową siatki Karnaugha

a) X1={d,e,h}, X2={a,b,c}

abc \ deh	000	001	011	010	110	111	101	100	
000	0	1	0	1	0	-	1	0	
001	1	-	1	0	-	0	1	0	
011	-	0	-	-	-	1	0	1	
010	0	1	0	1	0	-	-	0	
110	-	0	-	0	-	0	-	0	0
111	0	-	-	1	0	1	0	1	
101	-	1	1	0	1	0	-	-	
100	-	0	-	-	-	-	0	-	0

b)



Rys. 6. Siatka Karnaugha wraz z grafem niezgodności i dopełnień wierszy

Fig. 6. The Karnaugh map together with the graph of incompatibility and complementation of rows

$\mu(\mathbf{X}_2|\mathbf{X}_1)$, która jest równa liczbie kolorów użytych do rozróżnienia wszystkich wierzchołków grafu. Istota kolorowania polega na przypisaniu wierzchołkom grafu minimalnej liczby kolorów w taki sposób, aby każde dwa wierzchołki połączone ciągłą krawędzią miały inny kolor. Algorytm kolorowania grafu niezgodności i dopełnień wierszy składa się z dwóch etapów.

W pierwszym etapie poszczególnym wierzchołkom przyporządkowywane są wszystkie możliwe kolory dozwolone i kolory dopełnień.

Algorytm kolorowania wierzchołków oparty jest na sekwencyjnym wyborze wierzchołków, którym przypisywany jest ***kolor dozwolony*** (oznaczany dużą literą) lub ***kolor dopełnienia*** (oznaczany dużą literą z podkreśleniem u góry) w miarę możliwości taki, który jest już przypisany innemu wierzchołkowi. Po przypisaniu w i-tym kroku wybranemu wierzchołkowi koloru dozwolonego lub koloru dopełnienia, przypisywane są kolory zabronione (oznaczane małą literą) wszystkim wierzchołkom, połączonym krawędziami ciągłymi z wybranym wierzchołkiem i kolory dopełnienia wszystkim wierzchołkom, połączonym krawędziami przerywanymi z wybranym wierzchołkiem.

Wybór *i*-tego wierzchołka prowadzony jest zgodnie z rozpatrywanymi kolejno następującymi zasadami:

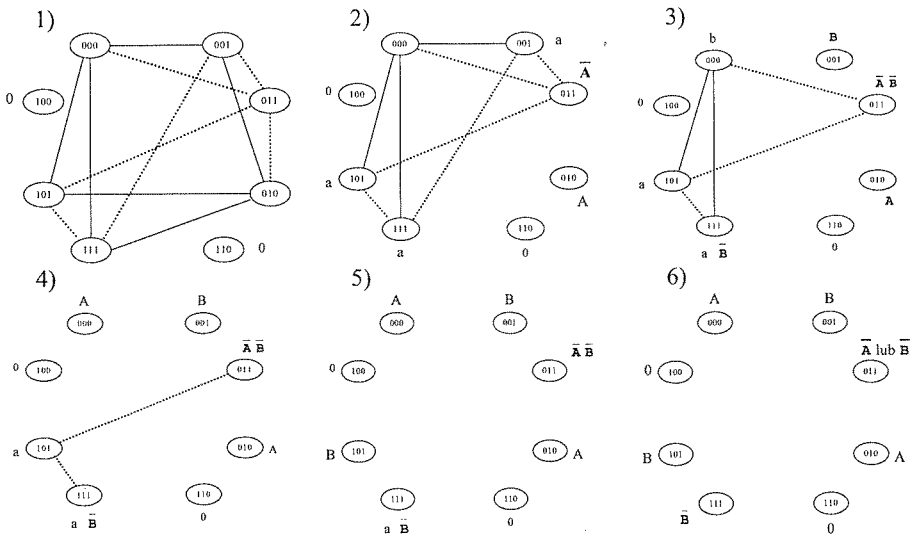
- wybierany jest wierzchołek z maksymalną liczbą kolorów zabronionych; przyporządkowywany jest mu kolor dozwolony (jeżeli jest to możliwe, to kolor ze zbioru już użytych kolorów);
- spośród wierzchołków z identyczną liczbą kolorów zabronionych wybierany jest ten, do którego dochodzi maksymalna liczba krawędzi;
- spośród wierzchołków z identyczną liczbą kolorów zabronionych i maksymalną liczbą krawędzi wybierany jest ten, który ma dodatkowo maksymalną liczbę kolorów dopełnień (jeżeli jest to możliwe, przyporządkowywany jest mu kolor dopełnienia);
- spośród wierzchołków z identyczną liczbą kolorów zabronionych, kolorów dopełnień i krawędzi wybierany jest ten, do którego dochodzi maksymalna liczba

krawędzi ciągłych.

Po wybraniu wierzchołka i przypisaniu odpowiednim wierzchołkom kolorów dozwolonych, kolorów dopełnień i kolorów zabronionych wykonywana jest redukcja grafu, polegająca na eliminacji krawędzi łączących wybrany wierzchołek z innymi wierzchołkami grafu, po czym wybierany jest kolejny (i+1) wierzchołek, tym razem po analizie grafu zredukowanego. W ostatnim kroku pierwszego etapu kolorowania, wierzchołkom, które mają kolory zabronione i kolory dopełnień, przypisywane są wszystkie możliwe, występujące na grafie kolory dozwolone i kolory dopełnień.

Przykład 4

Zastosujmy przedstawiony algorytm kolorowania wierzchołków grafu uzyskanego w przykładzie 3. Kolejne etapy kolorowania wierzchołków przedstawione są na rys.7.



Rys. 7. Kolejne etapy kolorowania wierzchołków grafu niezgodności i dopełnień wierszy

Fig. 7. Step-by-step colouring of the incompatibility and complementation of graph

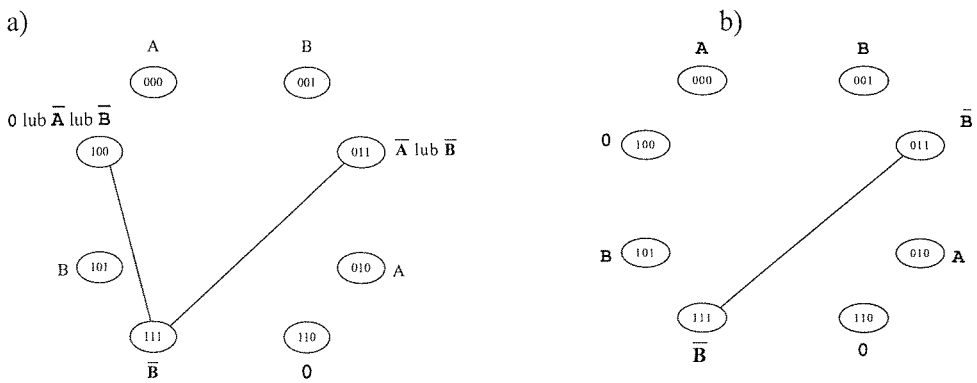
Celem drugiego etapu kolorowania grafu jest przypisanie jednego koloru (dozwolonego lub dopełnienia) wierzchołkom, które mają więcej niż jeden kolor — dozwolony lub dopełnienia. W tym celu tworzony jest graf zgodności kolorów, który powstaje przez połączenie wierzchołków mających więcej niż jeden kolor (dozwolony lub dopełnienia) z pozostałymi wierzchołkami z identycznym, jednym kolorem dozwolonym lub dopełnienia (tzn. wierzchołek z kolorami „B i C" łączy się z wierzchołkami mającymi tylko kolor „B" i tylko kolor „C"; wierzchołek z kolorami dopełnienia „$\overline{B}$ i $\overline{C}$" łączy się z wierzchołkami mającymi tylko kolor dopełnienia „$\overline{B}$" i tylko kolor „$\overline{C}$"). Dodatkowo

rozpatrywane są wszystkie wiersze pełne i puste, którym przypisywane są wszystkie możliwe kolory dozwolone i kolory dopełnienia.

Po utworzeniu grafu zgodności kolorów podlega on redukcji zgodnie z następującymi regułami:

- wybierany jest wierzchołek, który ma więcej niż jeden kolor (dozwolony lub dopełnienia),
- eliminowane są krawędzie łączące wybrany wierzchołek z wierzchołkami, których odległość kodowa jest większa o 1 od kodu wybranego wierzchołka.

Wynikiem tego etapu jest zredukowany graf zgodności kolorów, na podstawie którego przypisywane są kolory wierzchołkom, które po pierwszym etapie kolorowania miały większą liczbę kolorów dozwolonych i kolorów dopełnienia (rys. 8). Przypisanie identycznych kolorów wierzchołkom, dla których odległość kodowa wynosi 1, stwarza warunki do minimalizacji liczby iloczynów wykorzystywanych w bloku wolnym.



Rys. 8. Drugi etap kolorowania; a) graf zgodności kolorów, b) zredukowany graf zgodności kolorów z przypisanymi wierzchołkom kolorami

Fig. 8. The second step of the colouring process; a) graph of colour compatibility; b) reduced graph of colour compatibility with fixed colours assigned to the nodes

Zaproponowana metoda analizy grafu zgodności kolorów wplata w proces kolorowania element wpływający na minimalizację liczby iloczynów wykorzystywanych w bloku wolnym, powstającym w wyniku dekompozycji wierszowej. W ten sposób w jeden z pierwszych etapów dekompozycji wpleciony został element heurystycznej minimalizacji liczby użytych iloczynów.

3.2. WYBÓR I ELEMENTY OPTYMALIZACJI DEKOMPOZYCJI WIERSZOWEJ

Istota dekompozycji wierszowej polega na takim podziale układu, który zapewnia realizowalność bloku wolnego w jednym bloku logicznym typu PAL o zadanej liczbie

iloczynów. Nasuwa się pytanie, jak znaleźć odpowiednią dekompozycję wierszową, zakładając realizację na blokach logicznych typu PAL o zadanej liczbie iloczynów.

Efektywny podział układu pierwotnego na blok związany i wolny, będący wynikiem dekompozycji wierszowej, powinien spełniać dwa warunki:

- liczba iloczynów potrzebnych do realizacji bloku wolnego powinna być mniejsza lub równa zadanej liczbie iloczynów zawartych w bloku logicznym typu PAL,
- blok związany powinien być realizowany na minimalnej liczbie bloków logicznych typu PAL o zadanej liczbie iloczynów.

O ile pierwszy warunek można stosunkowo łatwo sprawdzić, poszukiwanie rozwiązania, dla którego blok związany realizowany jest na minimalnej liczbie bloków logicznych typu PAL o zadanej liczbie iloczynów, jest trudne i w ogólnym przypadku nierozwiązywalne.

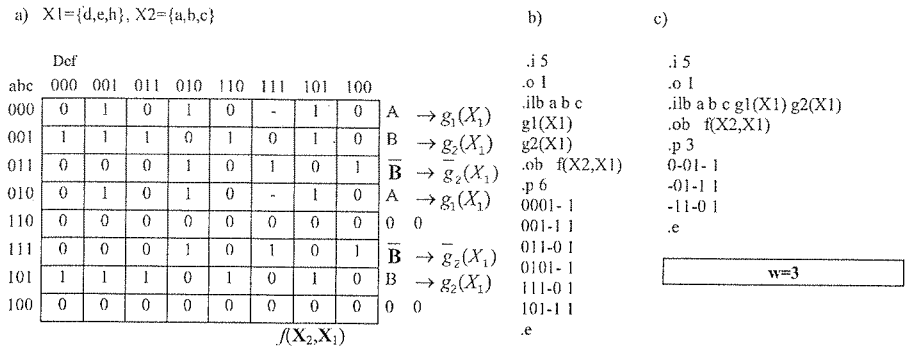
Dla danego podziału argumentów funkcja $f(\mathbf{X}_2, \mathbf{X}_1)$ może być przedstawiona w postaci: $f(X_2, X_1) = H_0 + \sum_{i=1}^{\mu(\mathbf{X}_2|\mathbf{X}_1)} \left[H_i(\mathbf{X}_2)\, g_i(\mathbf{X}_1) + H_i'(X_2)\, \bar{g}_i(\mathbf{X}_1)\right]$. Liczba iloczynów potrzebnych do realizacji bloku wolnego jest równa liczbie implikantów funkcji $f(\mathbf{X}_2, \mathbf{X}_1)$.

Optymalizacja bloku związanego wymaga przyjęcia pewnych uproszczeń, umożliwiających szybki wybór odpowiedniej dekompozycji wierszowej. Wyznaczana w wyniku kolorowania grafu niezgodności i dopełnień wierszy złożoność wierszowa powiązana jest z minimalną liczbą wyjść bloku związanego. Jeżeli liczba implikantów potrzebnych do realizacji funkcji $g_i(\mathbf{X}_1)$ jest mniejsza lub równa k, to liczba niezbędnych bloków logicznych typu PAL, potrzebnych do realizacji bloku związanego, wynosi $\delta_z = \mu(\mathbf{X}_2|\mathbf{X}_1)$. Fakt ten sprawia, że w celu minimalizacji wartości współczynnika δ_z przyjęto heurystyczną zasadę zawężenia obszaru poszukiwania rozwiązania do rozwiązań, dla których $\mu(\mathbf{X}_2|\mathbf{X}_1) = \min$.

Przykład 5

Wyznaczmy liczbę iloczynów potrzebnych do realizacji bloku wolnego funkcji analizowanej w przykładzie 3. W wyniku kolorowania wierzchołków grafu niezgodności i dopełnień wierszy (przykład 4) poszczególnym wierszom zostały przydzielone kolory. Doprowadziło to do określenia wartości funkcji dla wektorów, dla których przyjmowała ona wartości nieokreślone. Wynik tego etapu przedstawiony jest na rys. 9a). W kolejnych krokach wyznacza się funkcję opisującą blok wolny (rys. 9b)) i po jej minimalizacji liczbę iloczynów potrzebnych do realizacji bloku wolnego oznaczoną na rysunku symbolem „w" (rys. 9c)).

Ponieważ relacja dopełnienia wierszy jest relacją symetryczną (tzn., jeżeli wiersz X jest dopełnieniem wiersza Y, to wiersz Y jest dopełnieniem wiersza X), to istnieje możliwość takiego przyporządkowania wierszom kolorów dozwolonych, które prowadzi do minimalizacji złożoności bloku związanego. Należy podkreślić fakt, że zamiana kolorów dozwolonych z kolorami dopełnień nie wpływa na złożoność bloku wolnego.

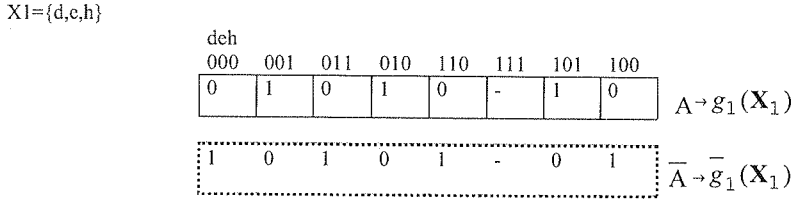


Rys. 9. Sposób wyznaczania liczby iloczynów potrzebnych do realizacji bloku wolnego; a) wynik kolorowania; b) pierwotny opis funkcji realizowanej w bloku wolnym; c) wynik minimalizacji funkcji

Fig. 9. The method of determining the number of AND gate of the free block; a) results of the colouring process; b) primary description of the function, which is implemented within the free block; c) results of the minimization process

Przykład 6

Rozpatrzmy wzorce wierszy z przykładu 5. Ponieważ funkcja $g_1(\mathbf{X}_1)$ jest sumą mniejszej liczby implikantów niż funkcja $\bar{g}_1(\mathbf{X}_1)$, stąd właściwe jest przyporządkowanie koloru A do wzorców wierszy opisanych przez funkcję $g_1(\mathbf{X}_1)$.



$$g_1(\mathbf{X}_1) = \bar{e}h + \bar{d}e\bar{h} \qquad \bar{g}_1(\mathbf{X}_1) = \bar{e}\bar{h} + eh + d\bar{h}$$

Inaczej wygląda sytuacja w przypadku wzorca B. Funkcja $g_2(\mathbf{X}_1)$ jest sumą większej liczby implikantów niż funkcja $\bar{g}_2(\mathbf{X}_1)$. W tej sytuacji korzystna jest zamiana przyporządkowania koloru dozwolonego z kolorem dopełnienia. Wpłynie to na minimalizację wykorzystywanych w bloku związanym iloczynów, nie wpływając na złożoność bloku wolnego.

Uwaga: Sposób przyporządkowania kolorów dozwolonych i kolorów dopełnień jest istotny tylko w przypadku wykorzystywania bloków logicznych typu PAL bez możliwości ustawiania aktywnego stanu wyjściowego.

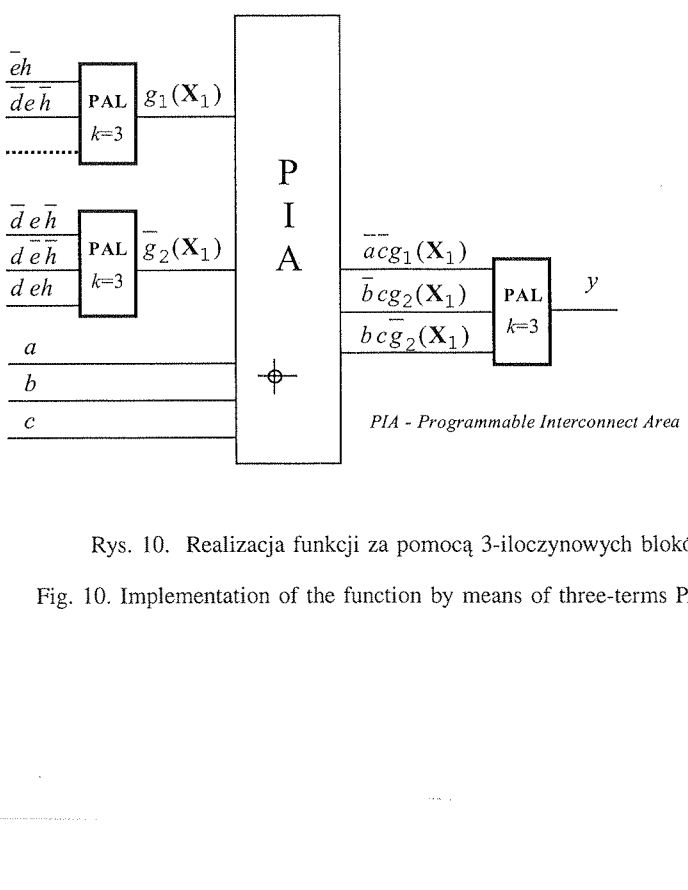
Efektem rozważań przedstawionych w przykładach 3–6 jest realizacja przykładowej funkcji na blokach logicznych typu PAL zawierających trzy iloczyny przedstawiona na rys. 10.

X1={d,e,h}

deh	000	001	011	010	110	111	101	100	
	1	1	1	0	1	0	1	0	$B \rightarrow g_2(\mathbf{X}_1)$
	0	0	0	1	0	1	0	1	$\overline{B} \rightarrow \overline{g}_2(\mathbf{X}_1)$

$$g_2(\mathbf{X}_1) = \overline{d}\,\overline{e} + \overline{d}\,h + \overline{e}h + de\overline{h} \qquad \overline{g}_2(\mathbf{X}_1) = \overline{d}e\overline{h} + deh + d\overline{e}\,\overline{h}$$

abc \ deh	000	001	011	010	110	111	101	100		
000	0	1	0	1	0	-	1	0	A	$\rightarrow g_1(X_1)$
001	1	1	1	0	1	0	1	0	$\overline{B}$	$\rightarrow g_2(X_1)$
011	0	0	0	1	0	1	0	1	B	$\rightarrow \overline{g}_2(X_1)$
010	0	1	0	1	0	-	1	0	A	$\rightarrow g_1(X_1)$
110	0	0	0	0	0	0	0	0	0	0
111	0	0	0	1	0	1	0	1	B	$\rightarrow \overline{g}_2(X_1)$
101	1	1	1	0	1	0	1	0	$\overline{B}$	$\rightarrow g_2(X_1)$
100	0	0	0	0	0	0	0	0	0	0



Rys. 10. Realizacja funkcji za pomocą 3-iloczynowych bloków typu PAL

Fig. 10. Implementation of the function by means of three-terms PAL-based blocks

4. ALGORYTM IMPLEMENTACJI FUNKCJI W STRUKTURACH TYPU PAL WYKORZYSTUJĄCY DEKOMPOZYCJĘ WIERSZOWĄ

Konsekwencją dekompozycji funkcji jest podział układu na dwa podukłady: związany i wolny. Podział ten prowadzi do ekspansji sumarycznej liczby wyjść. Każde dodatkowe wyjście pociąga za sobą użycie kolejnego bloku logicznego typu PAL. W wyniku podziału należy wykorzystać co najmniej $\mu(\mathbf{X}_2|\mathbf{X}_1)$ bloków logicznych typu PAL do realizacji bloku związanego. Fakt ten sprawia, że zastosowanie dekompozycji, jako elementu minimalizacji może być opłacalne tylko wtedy, gdy w klasycznym podejściu wykorzystanie sprzężeń zwrotnych prowadzi do użycia większej liczby bloków logicznych typu PAL. Minimalna liczba wyjść, równocześnie minimalna liczba bloków logicznych typu PAL, potrzebnych do realizacji układu po podziale, wynosi $\mu(\mathbf{X}_2|\mathbf{X}_1) + 1$ (co najmniej $\mu(\mathbf{X}_2|\mathbf{X}_1)$ bloków logicznych typu PAL potrzebnych jest do realizacji bloku związanego i dokładnie jeden blok do realizacji bloku wolnego). Rozpatrzmy realizację funkcji $f_i : B^n \to B$ na k-iloczynowych blokach PAL. Zminimalizowana postać funkcji $f_i : B^n \to B$, przedstawiona w postaci zbioru Δ_{f_i} implikantów, może być zrealizowana metodą klasyczną za pomocą δ_{f_i} bloków logicznych typu PAL, zawierających k-iloczynów, przy czym $\delta_{f_i} = \left\lceil \frac{\Delta_{f_i} - k}{k-1} \right\rceil + 1$.

Zastosowanie dekompozycji może prowadzić do ograniczenia liczby bloków logicznych typu PAL tylko wtedy, gdy:

$$\mu(\mathbf{X}_2|\mathbf{X}_1) + 1 < \delta_{f_i} = \left\lceil \frac{\Delta_{f_i} - k}{k-1} \right\rceil + 1 \tag{3}$$

Zależność 3 stanowi warunek konieczny ograniczenia liczby bloków logicznych typu PAL, który może eliminować pewne podziały z dalszych rozważań lub wręcz wskazywać na niemożliwość uzyskania lepszych rozwiązań od metody klasycznej. Jeżeli na przykład zminimalizowana postać funkcji $f_i : B^n \to B$ opisana jest przez zbiór mniej niż $2k$–implikantów, to stosowanie dekompozycji jest bezcelowe, gdyż na pewno nie ograniczy liczby bloków logicznych wykorzystywanych w metodzie klasycznej.

W procesie poszukiwania efektywnej dekompozycji wierszowej jednym z głównych elementów jest problem doboru odpowiedniego podziału argumentów funkcji.

Rozpatrzmy podział argumentów funkcji na rozłączne podzbiory $\mathbf{X}_1$ i $\mathbf{X}_2$. Realizowalność bloku wolnego w k-iloczynowym bloku PAL jest zapewniona dla każdego podziału, dla którego spełniona jest nierówność $2^{\bar{\bar{\mathbf{X}}}_2}$, tzn. $\bar{\bar{\mathbf{X}}} \le [\lg_2 k]$. Fakt ten sprawia, że poszukiwania podziału argumentów zapewniającego realizowalność bloku wolnego w bloku logicznym typu PAL o zadanej liczbie iloczynów są rozpoczynane od podziałów, dla których moc zbioru $\mathbf{X}_2$ spełnia warunek $\bar{\bar{\mathbf{X}}} = [\lg_2 k] + 1$. Przytoczony warunek doboru podziałów sprawia, że w pierwszym kroku analizowane są podziały, dla których może (nie musi) istnieć dekompozycja wierszowa prowadząca do realizacji bloku wolnego na k–iloczynowym bloku PAL. W zależności od uzyskanego wyniku pierwszego

kroku moc zbioru wolnego jest zmniejszana o jeden (zawsze istnieje rozwiązanie) lub zwiększana o jeden (poszukiwanie „lepszego" rozwiązania).

Algorytm dekompozycji wierszowej, prowadzący do efektywnego podziału układu na blok związany i wolny, realizowane na *k*–iloczynowych blokach typu PAL

1. Minimalizacja funkcji $f_i : B^n \rightarrow B^1$ (Espresso); wyznaczanie parametru $\delta_{f_i} = \left\lceil \frac{\Delta_{f_i} - k}{k-1} \right\rceil + 1$
2. Jeżeli $\Delta_{f_i} > 2k$, to realizacja funkcji $f_i : B^n \rightarrow B^1$ metodą klasyczną; w przeciwnym przypadku analiza podziałów argumentów funkcji i poszukiwanie odpowiedniej dekompozycji wierszowej, tzn.:
   - **A.** Rozpatrywane są podziały zbioru $\mathbf{I} = \{i_n, \ldots, i_2, i_1\}$ na podzbiory $\mathbf{X}_1 = \{i_{n_z}, \ldots, i_2, i_1\}$ oraz $\mathbf{X}_2 = \{i_n, \ldots, i_{n_z+2}, i_{n_z+1}\}$ takie, że $\overline{\overline{\mathbf{X}}} = [\lg_2 k] + 1$, gdzie k jest liczbą iloczynów zawartych w bloku logicznym typu PAL;
   - **B.** Dla poszczególnych podziałów wyznaczane są złożoności wierszowe (algorytm kolorowania grafu niezgodności i dopełnień wierszy) oraz liczby iloczynów potrzebnych do realizacji bloku wolnego;
   - **C.** Spośród analizowanych podziałów, dla których liczba iloczynów potrzebnych do realizacji bloku wolnego jest mniejsza lub równa k, jako rozwiązanie wybierany jest jeden podział, dla którego współczynnik złożoności wierszowej $\mu(\mathbf{X}_2|\mathbf{X}_1)$ przyjmuje wartość minimalną. Jeżeli znaleziono taki podział, to jest on zapamiętywany, po czym następuje przejście do punktu 2D. Jeżeli do tego kroku nie wybrano żadnego podziału, to następuje przejście do punktu 2E. Jeżeli nie wybrano żadnego podziału w aktualnym kroku, a wybrano podział w poprzednich krokach, to ostatnio wybrany podział jest poszukiwanym podziałem, po czym następuje przejście do punktu 2F;
   - **D.** Poszukiwany jest „lepszy podział", zapewniający realizowalność bloku wolnego w bloku logicznym typu PAL o zadanej liczbie iloczynów k, w kolejnej grupie podziałów, dla których $\overline{\overline{\mathbf{X}}}_2 := \overline{\overline{\mathbf{X}}} + 1$. Analizie poddawane są tylko wybrane podziały, dla których $\overline{\overline{\mathbf{X}}}_2 := \overline{\overline{\mathbf{X}}} + 1$, przy czym dodatkowa zmienna dodawana jest do zbiorów wolnych, dla których w poprzednim kroku (2C) liczba iloczynów potrzebnych do realizacji bloku wolnego była mniejsza lub równa k.
   - **E.** Jeżeli nie znaleziono żadnego podziału spełniającego warunek realizowalności w bloku wolnym, zawierającym k–iloczynów, to rozwiązanie wybierane jest w zbiorze podziałów, dla których $\overline{\overline{\mathbf{X}}}_2 = [\lg_2 k]$. Wybierane jest rozwiązanie, dla którego $\mu(\mathbf{X}_2|\mathbf{X}_1)$ przyjmuje wartość minimalną.
   - **F.** Jeżeli dla wybranego podziału zachodzi zależność $\mu(\mathbf{X}_2|\mathbf{X}_1) + 1 > \delta_{f_i}$, to analizowana funkcja realizowana jest metodą klasyczną.
3. Dla wybranego podziału poszukiwane jest efektywne przyporządkowanie kolorów dozwolonych i dopełnień, prowadzące do minimalizacji liczby iloczynów potrzeb-

nych do realizacji bloku związanego (przykład 6); przyporządkowanie jest poszukiwane z wykorzystaniem programu Espresso (Espresso–Dso_both).

4. Minimalizacja funkcji realizowanych w bloku związanym $g : B^{n_z} \rightarrow B^p$ (Espresso-Dso); wyznaczanie liczby bloków logicznych typu PAL, potrzebnych do realizacji układu po podziale; liczba ta oznaczana przez symbol δ_{zw} jest sumą liczby bloków potrzebnych do realizacji bloku związanego (δ_z) i wolnego ($\delta_w = 1$).

$$\delta_{zw} = \delta_z + 1 = \sum_{i=1}^{p} \left(\left\lceil \frac{z_i - k}{k-1} \right\rceil + 1 \right) + 1$$

gdzie z_i jest liczbą iloczynów wykorzystywanych przez i–te wyjście bloku związanego

5. Jeżeli $\delta_{f_i} \leq \delta_{zw}$, to realizacja funkcji metodą klasyczną; w przeciwnym przypadku kolejny etap dekompozycji - podział bloku związanego (p.2), realizowany dla każdej funkcji $g_i : B^{n_z} \rightarrow B^1$, gdzie $i = 1, 2, \ldots, \mu(\mathbf{X}_2|\mathbf{X}_1)$.

Przykład 7

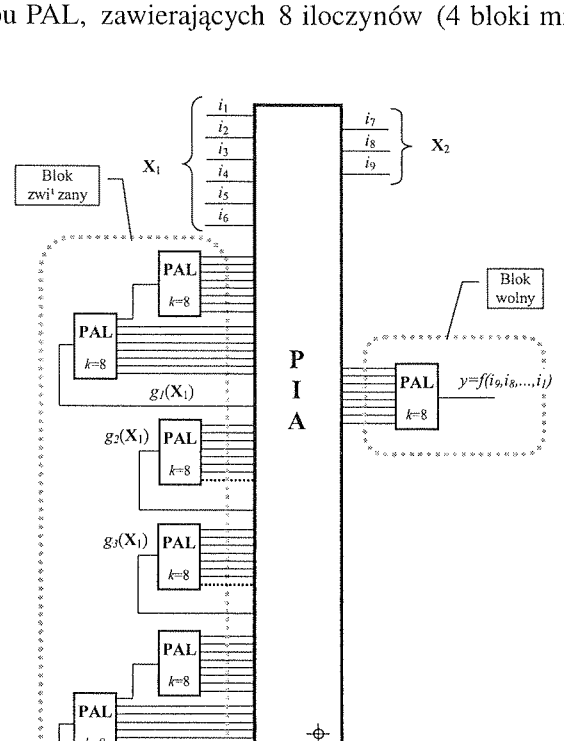
Rozpatrzmy realizację układu testowego $9sym$ ($f_i : B^9 \rightarrow B^1$) [3] korzystając z bloków logicznych typu PAL, zawierających 8 iloczynów ($k = 8$). W dalszej części przykładu przedstawione są kolejne etapy poszukiwania rozwiązania zgodne z algorytmem dekompozycji wierszowej.

- W wyniku minimalizacji (Espresso) uzyskujemy parametr $\Delta_{f_i} = 87$, który pozwala wyznaczyć liczbę bloków potrzebnych do realizacji klasycznej $\delta_{f_i} = \left\lceil \frac{\Delta_{f_i} - k}{k-1} \right\rceil + 1 = \lceil \frac{87-8}{7} \rceil + 1 = 13$.
- Ponieważ $\Delta_{f_i} > 2k$, stąd poszukiwana jest odpowiednia dekompozycja analizowanej funkcji. W pierwszej kolejności rozpatrywane są rozłączne podziały argumentów, dla których $\overline{\overline{\mathbf{X}}}_2 = [\lg_2 k] + 1 = [\lg_2 8] + 1 = 4$ oraz $\overline{\overline{\mathbf{X}}}_1 = 5$.
- Dla wszystkich analizowanych podziałów w wyniku kolorowania grafu niezgodności i dopełnień wierszy uzyskiwana jest złożoność wierszowa $\mu(\mathbf{X}_2|\mathbf{X}_1) = 4$.
- Dla wszystkich podziałów analizowanej grupy uzyskiwany w wyniku podziału blok wolny nie jest realizowalny w bloku logicznym typu PAL, zawierającym 8 iloczynów.
- W kolejnym kroku rozpatrywane są podziały, dla których $\overline{\overline{\mathbf{X}}}_2 = [\lg_2 k] + 1 = [\lg_2 8] = 3$ oraz $\overline{\overline{\mathbf{X}}}_1 = 6$.
- Dla wszystkich analizowanych podziałów w wyniku kolorowania grafu niezgodności i dopełnień wierszy uzyskiwana jest złożoność wierszowa $\mu(\mathbf{X}_2|\mathbf{X}_1) = 4$.
- Dla wszystkich podziałów analizowanej grupy uzyskiwany w wyniku podziału blok wolny jest realizowalny w bloku logicznym typu PAL, zawierającym 8 iloczynów.

- Nie występują żadne kolory dopełnień; minimalizacja funkcji realizowanych w bloku związanym (Espresso-Dso) wskazała, że lepiej wzorce wierszy skojarzyć z kolorami dopełnień (minimalizacja bloku związanego).
- W wyniku minimalizacji uzyskiwane są liczby iloczynów potrzebnych do realizacji bloku związanego $z_1 = 15$, $z_2 = 7$, $z_3 = 7$, $z_4 = 15$, co pozwala wyznaczyć liczbę bloków logicznych typu PAL, potrzebnych do realizacji układu po podziale:

$$\delta_{zw} = \delta_z + \delta_w = \sum_{i=1}^{4} \left(\left\lceil \frac{z_i - k}{k - 1} \right\rceil + 1 \right) + 1 = 7$$

- Wyszukiwany podział jest akceptowany, ponieważ $\delta_{zw} < \delta_{fi}$.
- Ponieważ liczby iloczynów potrzebnych do realizacji bloku związanego z_1 do z_4 są mniejsze od wartości $2k = 16$, stąd poszukiwania zostają zakończone. Ostateczne rozwiązanie, przedstawione na rys. 11 wykorzystuje tylko 7 bloków logicznych typu PAL, zawierających 8 iloczynów (4 bloki mniej niż metoda klasyczna).



Rys. 11. Wynik dekompozycji układu testowego 9*sym*

Fig. 11. Results of decomposition for the 9*sym* benchmark

5. WYNIKI EKSPERYMENTÓW

Przedstawioną w artykule metodę realizacji funkcji w strukturach typu PAL zaimplementowano w opracowanym w Instytucie Elektroniki Politechniki Śląskiej systemie PALDec (Palec — PAL decomposition) przeznaczonym do syntezy układów cyfrowych realizowanych w strukturach matrycowych typu PAL. Przeprowadzono szereg eksperymentów. Uzyskiwane wyniki porównano pod względem liczby wykorzystywanych bloków logicznych oraz liczby warstw z metodą klasyczną (Classical). Dla wielowyjściowych układów testowych syntezie poddawano każdą funkcję oddzielnie. Wyniki syntezy zawarte są w tab. 1. W pierwszej części tej tabeli przedstawione są wyniki syntezy układów testowych [3] metodą klasyczną (Classical). W kolumnach, w których występuje litera „B", podano liczbę wykorzystywanych k–iloczynowych bloków logicznych typu PAL, natomiast w kolumnach z literą „W" — liczbę warstw wynikowego układu. W drugiej części tabeli z nagłówkiem PALDec zawarte są wyniki syntezy uzyskiwane proponowaną metodą. Prezentowana metoda w poszczególnych etapach poszukiwania odpowiedniej dekompozycji zawiera elementy porównania z metodą klasyczną. Zapewnia to znalezienie rozwiązania wykorzystującego nie większą liczbę bloków logicznych niż liczba bloków uzyskiwana w metodzie klasycznej, nieznacznie (2-3%) wydłużając czas syntezy.

W zbiorze 440 porównywanych przypadków proponowana metoda (PALDec) doprowadziła do 141 (32%) rozwiązań wykorzystujących mniejszą liczbę bloków logicznych niż metoda klasyczna. Dla pewnych układów testowych uzyskano znaczącą redukcję wykorzystywanych bloków logicznych, np.: *rd*84 (k = 4; metoda klasyczna wykorzystuje około 10 razy więcej bloków logicznych niż proponowana metoda), *rd*73_*f*1 (k = 4; metoda klasyczna wykorzystuje 7 razy więcej bloków logicznych), 5*xp*1_*f*2, *clip*_*f*3, *f*51*m*_*f*1, *rd*53_*f*1, *sao*2_*f*3, *xor*5, *z*4*m*1_*f*2, *z*5*xp*1_*f*3 (k = 4; metoda klasyczna wykorzystuje około 2 razy więcej bloków logicznych). Znaczne różnice występują nie tylko w przypadku wykorzystywania bloków logicznych złożonych z niewielkiej liczby iloczynów.

Niestety z redukcją liczby bloków logicznych nie idzie w parze redukcja liczby warstw logicznych. Wśród analizowanych przypadków znalazło się 37 (8%) realizacji zawierających większą liczbę warstw logicznych w porównaniu z metodą klasyczną i tylko 9 (2%) rozwiązań zawierających mniejszą liczbę warstw logicznych. W pozostałych przypadkach uzyskiwano identyczną liczbę warstw logicznych.

Podsumowaniem pierwszej części eksperymentów mogą być wykresy 1 oraz 2. Wartości przedstawione na wykresie 1 wyznaczone zostały z zależności $\frac{\sum bloków_{(Classical)}}{\sum bloków_{(PALDec)}}$, gdzie $\sum bloków_{(Classical)}$ i $\sum bloków_{(PALDec)}$ oznaczają odpowiednie sumy liczby bloków, wykorzystywanych w poszczególnych metodach syntezy, przedstawione w tab. 1.

Podobnie wyznaczone zostały wartości przedstawione na wykresie 2 na podstawie zależności $\frac{\sum warstw_{(PALDec)}}{\sum warstw_{(Classical)}}$, gdzie $\sum warstw_{(Classical)}$ i $\sum warstw_{(PALDec)}$

Tabela 1

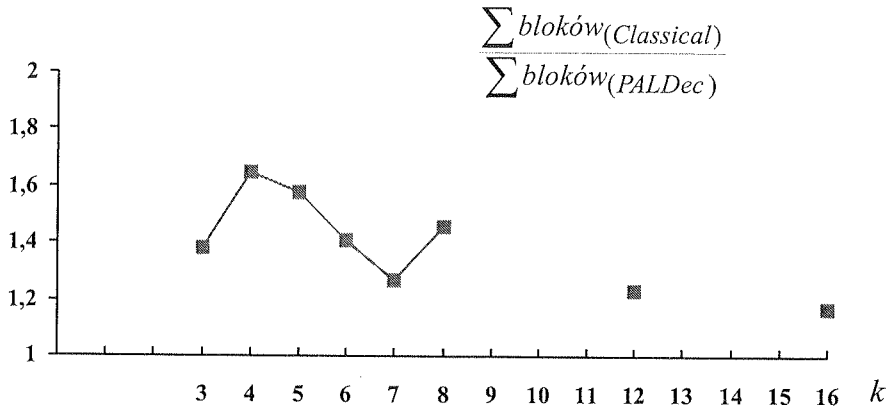
Porównanie klasycznej realizacji funkcji (Classical) z zaproponowaną metodą opartą na dekompozycji (PALDec)

				C L A S S I C A L																P A L D E C																	
				k= 3		k= 4		k= 5		k= 6		k= 7		k= 8		k= 12		k= 16		k= 3		k= 4		k= 5		k= 6		k= 7		k= 8		k= 12		k= 16	
Przykład	i	o	p	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W	B	W
f0			7	3	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	3	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1
f1			11	5	3	4	2	3	2	2	2	2	2	2	1	1	1	1	1	5	3	4	3	3	2	2	2	2	2	2	2	1	1	1	1	1	1
f2			18	9	3	6	3	5	2	4	2	3	2	3	2	2	2	2	2	5	3	3	2	3	2	3	2	3	2	2	2	2	2	2	2	2	2
f3			14	7	3	5	2	4	2	3	2	3	2	2	2	2	2	1	1	4	3	3	2	3	2	3	2	3	2	2	2	2	2	1	1	1	1
f4			10	5	3	3	2	3	2	2	2	2	2	2	2	1	1	1	1	3	3	3	3	2	2	2	2	2	2	2	2	1	1	1	1	1	1
f5			5	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
f6-f9				4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1
5xp1	7	10	75	35	3	26	3	22	2	18	2	16	2	15	2	12	2	11	2	26	3	21	3	18	2	17	2	16	2	14	2	12	2	11	2		
9sym	9	1	87	43	5	29	4	22	3	18	3	15	3	13	3	8	2	6	2	42	5	21	4	15	4	14	4	14	4	7	3	7	3	5	2		
9symml	9	1	87	42	5	28	4	21	3	17	3	14	3	12	3	8	2	6	2	42	5	21	4	15	4	14	4	14	4	7	3	7	3	5	2		
f0			21	10	3	7	3	5	2	4	2	4	2	3	2	2	2	2	2	10	3	7	3	5	2	4	2	4	2	3	2	2	2	2	2		
f1			31	15	4	10	3	8	3	6	3	5	2	5	2	3	2	2	2	15	4	10	3	8	3	6	3	5	3	5	2	3	2	2	2		
f2			42	21	4	14	3	11	3	9	3	7	2	6	2	4	2	3	2	21	4	14	3	11	3	9	3	7	2	6	2	4	2	3	2		
f3			34	17	4	11	3	9	3	7	3	6	2	5	2	3	2	2	2	9	4	6	3	5	3	5	3	4	2	4	2	3	2	3	2		
f4			20	10	3	7	3	5	2	4	2	4	2	3	2	2	2	2	2	10	3	7	3	5	2	4	2	4	2	3	2	2	2	2	2		
clip	9	5	167	73	4	49	3	38	3	30	3	26	2	22	2	14	2	11	2	65	4	44	3	34	3	28	3	24	3	21	2	14	2	12	2		
f0			23	11	3	8	3	6	2	5	2	4	2	4	2	2	2	2	2	6	4	5	3	5	3	3	2	3	2	3	2	2	2	2	2		
f1			18	9	3	6	3	5	2	4	2	3	2	3	2	2	2	2	2	5	3	3	2	3	2	3	2	3	2	3	2	2	2	2	2		
f2			14	7	3	5	2	4	2	3	2	3	2	2	2	2	2	1	1	4	2	3	2	2	2	2	2	2	2	2	2	2	2	2	1	1	
f3			10	5	3	3	2	3	2	2	2	2	2	2	2	1	1	1	1	3	3	3	3	2	2	2	2	2	2	2	2	1	1	1	1		
f4			5	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1		
f5-f7			1..3	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1		
f51m	8	8	76	37	3	27	3	22	2	18	2	16	2	15	2	11	2	10	2	23	4	19	3	16	3	14	2	14	2	14	2	11	2	10	2		
f0			5	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1		
f1			16	8	3	5	2	4	2	3	2	3	2	2	2	1	1	4	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1		
f2			10	5	3	4	2	3	2	2	2	2	2	2	2	1	1	1	1	5	3	3	2	3	2	2	2	2	2	2	2	2	1	1	1		
rd53	5	3	32	15	3	11	2	8	2	6	2	6	2	6	2	4	2	3	1	11	3	7	2	6	2	5	2	5	2	5	2	4	2	3	1		
f0			42	21	4	14	3	11	3	9	3	7	2	6	2	4	2	3	2	21	4	7	3	7	3	7	3	7	3	3	2	3	2	3	2		
f1			64	32	4	21	3	16	3	13	3	11	3	9	2	6	2	5	2	6	3	3	2	3	2	3	2	3	2	2	2	2	2	2	2		
f2			35	17	4	12	3	9	3	7	2	6	2	5	2	4	2	3	2	15	4	7	3	5	3	5	3	5	3	5	3	4	2	3	2		
rd73	7	3	141	70	4	47	3	36	3	29	3	24	3	20	2	14	2	11	2	42	5	17	3	15	3	15	3	15	3	10	3	9	2	8	2		
f0			84	42	4	28	3	21	3	17	2	14	2	12	2	8	2	6	2	40	4	11	4	7	3	7	3	7	3	5	3	3	2	3	2		
f1			128	64	5	43	4	32	4	26	3	22	3	19	3	12	2	9	2	7	3	4	3	4	3	4	3	4	3	3	3	3	3	2	2		
f2			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
f3			70	35	4	23	3	18	3	14	3	12	3	10	3	7	2	5	2	28	4	11	3	11	3	11	3	10	3	6	3	5	2	4	2		
rd84	8	4	256	142	5	95	4	72	4	58	3	49	3	42	3	28	2	21	2	76	4	27	4	23	3	23	3	22	3	15	3	12	3	10	2		
f0			10	5	3	3	2	3	2	2	2	2	2	2	2	1	1	1	1	5	3	3	2	3	2	2	2	2	2	2	2	1	1	1	1		
f1			20	10	3	7	3	5	2	4	2	4	2	3	2	2	2	2	2	10	3	7	3	5	2	4	2	4	2	3	2	2	2	2	2		
f2			22	11	3	7	3	6	2	5	2	4	2	3	2	2	2	2	2	8	3	7	3	6	3	4	2	4	2	3	2	2	2	2	2		
f3			21	10	3	7	3	5	2	4	2	4	2	3	2	2	2	2	2	5	3	4	3	4	3	3	2	3	2	3	2	2	2	2	2		
sao2	10	4	58	36	3	24	3	19	2	15	2	14	2	11	2	7	2	7	2	28	3	21	3	18	3	13	2	13	2	11	2	7	2	7	2		
xor5	5	1	16	8	2	5	2	4	2	3	2	3	2	3	2	2	2	1	1	4	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1		
f0			15	7	3	5	2	4	2	3	2	3	2	2	2	2	1	1	1	4	3	3	3	3	3	3	2	3	2	2	2	2	2	1	1		
f1			28	14	3	9	3	7	3	6	2	5	2	4	2	3	2	2	2	5	4	3	3	3	3	3	3	3	2	2	2	2	2	2	2		
f2			12	6	3	4	2	3	2	3	2	2	2	2	2	1	1	1	1	4	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1		
f3			4	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
z4ml	7	4	59	29	3	19	3	15	3	13	2	11	2	9	2	7	2	5	2	15	4	9	3	9	3	9	3	8	2	7	2	6	2	5	2		
f1			7	3	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	3	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1		
f2			11	5	3	4	2	3	2	2	2	2	2	2	1	1	1	1	1	5	3	4	2	3	2	2	2	2	2	2	2	1	1	1	1		
f3			22	11	3	7	3	6	2	5	2	4	2	3	2	2	2	2	2	5	3	3	2	3	2	3	2	3	2	2	2	2	2	2	2		
f4			14	7	3	5	2	4	2	3	2	3	2	2	2	2	2	1	1	4	3	3	2	3	2	3	2	3	2	2	2	2	2	1	1		
f5			10	5	3	3	2	3	2	2	2	2	2	2	2	1	1	1	1	3	3	3	3	2	2	2	2	2	2	2	2	1	1	1	1		
f6			5	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1		
f0;f7-f9				4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1
z5xp1	7	10	128	37	3	27	3	23	2	19	2	17	2	15	2	12	2	11	2	26	3	21	3	18	2	17	2	16	2	14	2	12	2	11	2		
z9sym	9	1	420	42	5	28	4	21	3	17	3	14	3	12	3	8	2	6	2	42	5	21	4	15	4	14	4	14	4	7	3	7	3	5	2		
Σ				609		415		323		26		225		195		135		109		442		251		204		18		177		134		110		93		24	
				48		41		34	1	32		31		30		26		24		50		41		38	5	36		35		31		30					

oznaczają odpowiednie sumy liczby warstw logicznych dla poszczególnych wartości k.

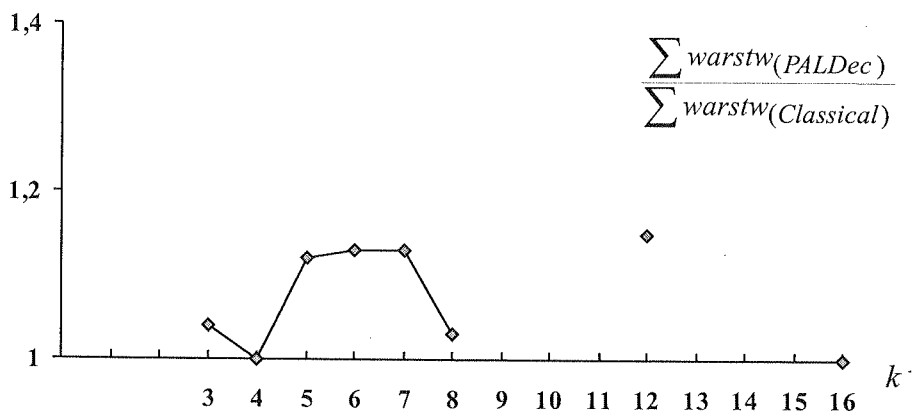
Wykres 1

Porównanie zaproponowanej metody syntezy (PALDec) z metodą klasyczną (Classical) pod względem liczby bloków (opis w tekście)



Wykres 2

Porównanie zaproponowanej metody syntezy (PALDec) z metodą klasyczną (Classical) pod względem liczby warstw logicznych (opis w tekście)



Analiza wyników pozwala stwierdzić, że uzyskana proponowaną metodą (PALDec) redukcja liczby bloków pociąga za sobą (w większości przypadków) ekspansję liczby warstw logicznych. Przedstawiona metoda jest szczególnie skuteczna w sytuacji, gdy

$k = 4$ lub $k = 8$. Występuje wtedy znaczne zmniejszenie liczby bloków przy porównywalnej liczbie warstw logicznych. Powyższy wniosek jest intuicyjnie wyczuwalny. Rozpatrywane podziały argumentów funkcji tworzą zawsze 2^i wierszy. Dla funkcji „słabo dekomponowalnych" często występują sytuacje, w których złożoność wierszowa jest równa liczbie wierszy siatki Karnaugha. Sytuacje takie są szczególnie korzystne w przypadku wykorzystywania 2^i– iloczynowych bloków logicznych typu PAL.

6. PODSUMOWANIE

W artykule przedstawiono nowy model dekompozycji — dekompozycję wierszową. Została ona opracowana dla potrzeb syntezy przeznaczonej dla struktur CPLD typu PAL. Polega ona na sekwencyjnym poszukiwaniu dekompozycji zapewniającej realizowalność bloku wolnego w bloku logicznym typu PAL zawierającym k–iloczynów. Proponowana metoda syntezy jest alternatywą do klasycznej metody opartej na dwupoziomowej minimalizacji poszczególnych funkcji.

Poszczególne etapy syntezy (sposób doboru elementów zbioru związanego, kolorowanie wierzchołków grafu niezgodności i dopełnień wierszy, przyporządkowanie kolorów dozwolonych i kolorów dopełnień itp.) są dostosowane do zasobów logicznych struktury typu PAL. Dostosowania elementów dekompozycji do zasobów logicznych scharakteryzowanych przez blok logiczny typu PAL, pozwala na znaczną poprawę efektywności syntezy pod względem powierzchni w stosunku do metody klasycznej. W porównaniu z innymi metodami syntezy bazującymi na dekompozycji, metoda ta jest szczególnie atrakcyjna, ponieważ prowadzi do minimalizacji liczby warstw uzyskiwanych rozwiązań.

Przedstawione w pracy wyniki eksperymentów dowodzą, że proponowana metoda syntezy wykorzystująca dekompozycję wierszową jest szczególnie atrakcyjna dla struktur CPLD składających się z bloków logicznych typu PAL zawierających 2^i iloczynów zarówno pod względem liczby warstw, jak i liczby wykorzystywanych bloków logicznych.

7. BIBLIOGRAFIA

1. P. Ashar, S. Devadas, A.R. Newton: *Sequential Logic Synthesis*, Kluwer Academic Publisher, Boston/London/Dotdrecht, 1992.
2. R.L. Ashenhurst: *The decomposition of switching functions*, Proceedings of an International Symposium on the Theory of Switching, April 1957.
3. Collaborative Benchmarking Laboratory, Department of Computer Science at North Carolina State University, http://www.cbl.ncsu/edu/.
4. M.J. Ciesielski, S. Yang, PLADE: *A two-stage PLA decomposition*, IEEE Transactions on Computer-Aided Design, Vol. 11, No. 8, August 1992, pp. 943-954.
5. H.A. Curtis: *The Design of switching Circuits*, D.van Nostrand Company, Inc., Princeton, New Jersey, Toronto, New York, 1962.

6. D. Kania: *Decomposition-based synthesis and its application in PAL– oriented technology mapping*, Proceedings of 26-th Euromicro Conference, IEEE Computer Society Press, Maastricht, 2000, pp. 138-145.
7. D. Kania, A. Milik, J. Kulisz: *Decomposition of Multiple– Output Functions for CPLDs*, Proceedings of 31-st Euromicro Conference, IEEE Computer Society Press, Porto, 2005, pp. 42-449.
8. R. Malvi, M. Perkowski, L. Józwiak: *Exact Graph Coloring for Functional Decomposition: Do we Need it?*, Proceedings of 3rd International Workshop on Boolean Problems, Freiberg University of Mining and Technology, Institute of Computer Science, September 17–18, 1998, pp. 1–10.
9. G. Saucier, P. Sicard, L. Bouchet: *Multi-level synthesis on PAL's*, Proc. European Design Automation Conference, Glasgow, March 1990, pp. 542-546.
10. W. Wan, M. Perkowski: *A new Approach to the Decomposition of Incompletely Specified Multi-Output Functions Based on Graph Coloring and Local Transformations and Its Applications to FPGA Mapping*, Proc. of EDAC'92, 1992, pp. 230-235.

D. KANIA

ROW DECOMPOSITION IN LOGIC SYNTHESIS FOR CPLDS

Summary

A PAL-based logic block is the core of the most CPLDs. The aim of the paper is to present a new method of decomposition dedicated for PAL-based CPLDs. The proposed approach is an alternative to the classical method based on two-level minimization.

The idea of the method consists in sequential search for appropriate row decomposition. Individual steps of the synthesis process are adapted to logical resources of PAL-based CPLDs. Adjusting decomposition to logical resources characteristic for a PAL-based logic block allows for significant improvement of synthesis effectiveness in relation to the classical approach. Results of experiments show that the proposed logic synthesis method leads to significant reduction of chip area in relation to the classical method, especially for CPLD structures consisting of PAL-based logic blocks with 4 or 8 product terms.

Keywords: logic synthesis, decomposition, partitioning, PAL-based CPLD

proc
wart
potę
rego
okre
ogra
syste
wart
proc
tech
oblic
lopr
Rate
lustr
syste

Słow

Zasto
transporc

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 545–561

Zastosowanie binaryzacji okresów zadań w systemach wieloprocesorowych

MIROSŁAW GAJER

Katedra Automatyki AGH,
al. Mickiewicza 30, 30-059 Kraków,
email: mgajer@ia.agh.edu.pl

Otrzymano 2005.11.29
Autoryzowano 2006.02.27

Artykuł dotyczy zastosowania techniki binaryzacji okresów zadań w systemach wieloprocesorowych. Binaryzacja okresów zadań jest techniką polegającą na takiej transformacji wartości okresów zadań periodycznych, aby były wielokrotnościami (będącymi naturalną potęga liczby 2) pewnego okresu bazowego r. Zatem w systemie o okresach binarnych szeregowaniu podlegają zadania o wartościach okresów r, 2r, 4r, 8r, 16r, 32r itd. Binaryzacja okresów zadań ułatwia znalezienie ich planu szeregowania, gwarantującego dotrzymanie ograniczeń czasowych przez wszystkie zadania. Binaryzacja jest stosowana powszechnie w systemach czasu rzeczywistego o ostrych ograniczeniach czasowych. Obecnie binaryzacji wartości okresów podlegają głównie zadania przeznaczone do realizacji w systemach jednoprocesorowych. Artykuł niniejszy stanowi propozycję rozszerzenia obszaru stosowalności techniki binaryzacji okresów zadań na systemy zbudowane z większej liczby jednostek obliczeniowych. Zgodnie z propozycją autora zastosowanie binaryzacji okresów zadań wieloprocesorowych stanowi klucz do zastosowania do ich szeregowania popularnego algorytmu Rate Monotonic Scheduling. Przedstawione w artykule rozważania teoretyczne zostały zilustrowane przykładem szeregowania zadań wieloprocesorowych o okresach binarnych dla systemu zbudowanego z czterech jednostek obliczeniowych.

Słowa kluczowe: binaryzacja okresów zadań, systemy wieloprocesorowe, zadani wieloprocesorowe, szeregowanie zadań

1. WPROWADZENIE

Zastosowanie komputerowych systemów czasu rzeczywistego w telekomunikacji, transporcie, lotnictwie oraz nowoczesnych instalacjach przemysłowych staje się coraz

bardziej powszechne. Systemy czasu rzeczywistego różnią się tym od systemów komputerowych ogólnego przeznaczenia, że program podzielony jest na zadania, z których każde obsługuje wybrane procesy przemysłowe, będące przedmiotem sterowania przez komputer [1]. Zwykle każde z takich zadań posiada ściśle określone ograniczenie czasowe, którego naruszenie jest niedopuszczalne, ponieważ grozi utratą kontroli nad sterowanym procesem, co w konsekwencji prowadzić może do powstania katastrofy, wielkich strat finansowych, utraty życia ludzi itp. [2]

W związku z powyższym systemy czasu rzeczywistego muszą być starannie projektowane i w ich przypadku należy wykazać, już na etapie projektu, na drodze dowodów formalnych, że ograniczenia czasowe zostaną zawsze dochowane dla każdego z zadań, we wszystkich możliwych scenariuszach pracy systemu, gdzie bierze się pod uwagę również interakcję systemu z otoczeniem [3].

Rozwój metod formalnych przewidzianych do analizy i projektowania systemów czasu rzeczywistego stał się bezpośrednią przyczyną opracowania teorii szeregowania zadań, której celem jest wyznaczenie momentów rozpoczęcia wykonywania poszczególnych zadań w taki sposób, aby zostały dochowane ograniczenia czasowe dla wszystkich zadań [4].

Jednym z najbardziej popularnych algorytmów szeregowania zadań jest algorytm *Rate Monotonic Scheduling*, którego podstawowe właściwości zostały opisane w kolejnym punkcie [18]. Alternatywę dla rozważanego algorytmu szeregowania zadań stanowią algorytmy dynamiczne, takie jak np. EDF (ang. *Earliest Deadline First*) lub LLF (ang. *Least Laxity First*). Są to algorytmy przewidziane do szeregowani zadań napływających dynamicznie do systemu i w porównaniu z algorytmem *Rate Monotonic Scheduling* odznaczają się zdecydowanie mniejszą pewnością i stabilnością uzyskiwanych za ich pośrednictwem planów realizacji zadań. W rozważanym przez autora przypadku szeregowania zbioru zadań o z góry znanych charakterystykach zdecydowanie lepszym okazuje się algorytm *Rate Monotonic Scheduling*, ponieważ daje on gwarancję na dochowanie ograniczeń czasowych przez każde z zadań.

2. RATE MONOTONIC SCHEDULING

W swej klasycznej postaci algorytm *Rate Monotonic Scheduling* przewidziany jest do szeregowania okresowych, niezależnych i wywłaszczalnych zadań jednoprocesorowych na pojedynczym procesorze. Działanie tego algorytmu oparte jest na priorytetach przydzielanych do poszczególnych zadań w ten sposób, że zadania o mniejszej wartości okresu, czyli aktywowane częściej, otrzymują wyższy priorytet [12]. W danej chwili spośród zadań będących w stanie gotowości wykonywane jest to zadanie, które ma najwyższy priorytet. Jeśli podczas wykonywania danego zadania w stan gotowości wejdzie jakieś inne zadanie o wyższym priorytecie, wówczas wykonywane aktualnie zadanie zostanie wywłaszczone a procesor zostanie przekazany zadaniu o wyższym priorytecie [13]. Wywłaszczone zadanie zostanie wznowione dopiero wtedy, gdy w

stanie gotowości nie będzie już żadnych innych zadań o wyższym od niego priorytecie [14].

Szeregowaniu podlega zawsze zbiór N zadań okresowych, niezależnych i wywłaszczalnych, przy czym dla każdego zadania podany jest czas jego realizacji C_i, wartość jego okresu T_i, na podstawie czego może zostać wyliczona wartość współczynnika wykorzystania czasu pracy procesora przez i–te zadanie, która wynosi C_i/T_i. Ponadto każde z zadań posiada ograniczenie czasowe, które równe jest jego okresowi, ponieważ każde zadanie musi zostać zakończone przed wejściem w stan gotowości kolejnego egzemplarza tego zadania [9].

Warunek konieczny na szeregowalność zbioru zadań wynika bezpośrednio z oczywistego faktu, polegającego na tym, że łączny stopień wykorzystania czasu pracy procesora przez wszystkie zadania nie może być większy niż 100% [17].

$$\sum_{i=1}^{N} \frac{C_i}{T_i} \leqslant 1 \tag{1}$$

Z kolei warunek wystarczający podany został w 1973 roku przez Liu i Laylanda i wyraża się wzorem

$$\sum_{i=1}^{N} \frac{C_i}{T_i} \leqslant N\left(2^{\frac{1}{N}} - 1\right) \tag{2}$$

Dla dużych wartości N prawa strona nierówności (2) jest zbieżna do wartości $ln2 = 0,69$. Zatem na mocy warunku wystarczającego każdy zbiór zadań, który wykorzystuje czas pracy procesora w nie więcej niż 69%, zawsze będzie szeregowalny [15].

W przypadku, gdy warunek dany nierównością (2) nie jest spełniony, a jest spełniony warunek konieczny dany nierównością (1), dany zbiór zadań równie dobrze może być szeregowalny bądź nie. Aby przekonać się o tym, czy dany zbiór zadań jest szeregowalny, należy rozważyć najgorszy z możliwych przypadków, tzn. taki, w którym wszystkie zadania podlegające szeregowaniu wchodzą jednocześnie w stan gotowości, i policzyć czasy zakończenia realizacji wszystkich zadań. Jeżeli każde z rozważanych zadań zostanie zakończone przed upływem swego ograniczenia czasowego, wówczas dany zbiór zadań jest szeregowalny. W przypadku przeciwnym zbiór zadań szeregowalny nie jest [16].

Czas zakończenia realizacji zadania obliczany jest za pośrednictwem procedury iteracyjnej, przy czym jako pierwsze przybliżenie brana jest suma czasów wykonywania danego zadania oraz wszystkich zadań o wyższych priorytetach, ponieważ zanim dane zadanie będzie mogło rozpocząć swoją realizację, każde z zadań o priorytetach wyższych musi zostać wykonane co najmniej jeden raz [19]. Zatem pierwsze przybliżenie czasu realizacji zadania dane jest wzorem

$$t_0 = \sum_{i=1}^{M} C_i \tag{3}$$

Kolejne przybliżenia czasu zakończenia wykonywania zadania obliczane są z zależności

$$t_{i+1} = \sum_{i=1}^{M} C_i \left\lceil \frac{t_i}{T_i} \right\rceil \tag{4}$$

Jako czas zakończenia wykonywania zadania przyjmuje się t_k, gdy spełniony jest warunek

$$t_{k+1} = t_k \tag{5}$$

W przypadku systemów czasu rzeczywistego często stosowanym zabiegiem jest tzw. binaryzacja okresów zadań, która polega na dokonaniu takiej transformacji okresów zadań, aby szeregowaniu podlegały wyłącznie zadania o tzw. okresie bazowym r oraz zadania o wielokrotnościach okresu bazowego, które są naturalną potęgą dwójki [7]. Zatem po przeprowadzeniu binaryzacji okresów zadań szeregowaniu podlegają wyłącznie zadania o okresach $r, 2r, 4r, 8r, 16r, \ldots, 2^N r$.

Binaryzacja przeprowadzana jest w celu zmniejszenia rozmiarów tzw. horyzontu czasowego systemu, którego wartość równa jest najmniejszej wspólnej wielokrotności okresów zadań. Horyzont czasowy jest zatem przedziałem czasu, po upływie którego cały scenariusz pracy systemu powtarza się na nowo. Po przeprowadzeniu binaryzacji rozmiar horyzontu czasowego równy jest wartości najdłuższego z okresów szeregowanych zadań i zwykle zostaje wydatnie skrócony [18]. Niestety ujemną stronę binaryzacji stanowi fakt polegający na tym, że zbiór zadań z okresami binarnymi wykorzystuje czas pracy procesora w nieco większym stopniu niż przed binaryzacją. W szczególności zbiór zadań, który był szeregowalny przed przeprowadzeniem binaryzacji, w wyniku binaryzacji może tę szeregowalność utracić [8].

Binaryzacja okresów zadań, w wyniku której zadanie o okresie T_i uzyskuje nową, binarną wartość okresu T_i^b, przeprowadzana jest zgodnie z następującym wzorem

$$T_i^b = r2^{\left\lfloor \log_2\left(\frac{T_i}{r}\right)\right\rfloor} \tag{6}$$

Ponieważ jako okres bazowy r wybrana może zostać dowolna dodatnia liczba rzeczywista, nie większa od najmniejszej wartości okresu szeregowanych zadań, w rezultacie tego istnieje nieskończenie wiele alternatywnych sposobów przeprowadzenia binaryzacji okresów zadań. Wybór rożnych wartości okresu bazowego r skutkował będzie różnym stopniem wzrostu wykorzystania czasu pracy procesora przez szeregowany zbiór zadań. Zatem spośród nieskończenie wielu możliwych sposobów binaryzacji okresów zadań należy wybrać ten, który prowadził będzie do jak najmniejszego wzrostu stopnia wykorzystania czasu pracy procesora [10]. Najlepszej binaryzacji poszukuje się wśród tzw. binaryzacji istotnych, tzn. takich, dla których wartość okresu bazowego r_i dana jest wzorem

$$r_i = \frac{T_i}{2^{\left\lfloor \log_2\left(\frac{T_i}{T_{MIN}}\right)\right\rfloor}} \quad \text{dla} \quad i = 1, 2, \ldots, N \tag{7}$$

We wzorze (7) T_{MIN} oznacza najmniejsza wartość okresu spośród okresów wszystkich zadań podlegających szeregowaniu, ale wyznaczonych przed przeprowadzeniem binaryzacji.

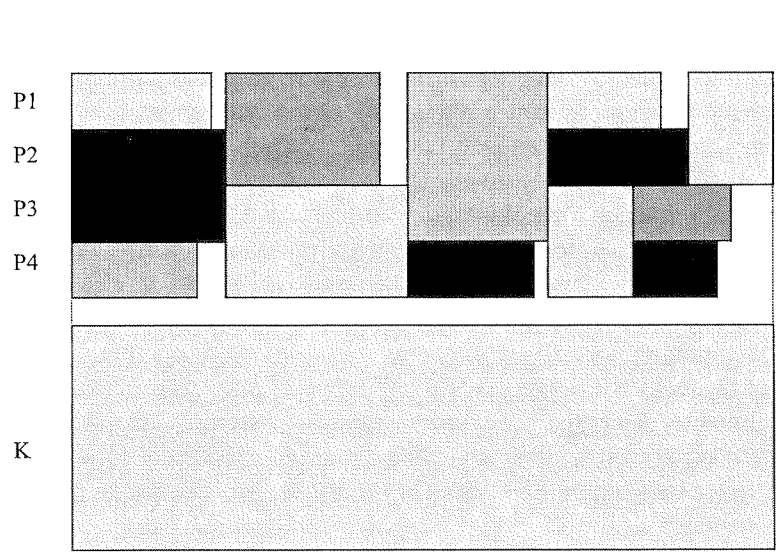
3. SZEREGOWANIE ZADAŃ W SYSTEMACH WIELOPROCESOROWYCH

Duże wymagania na moc obliczeniową, jakie są obecnie stawiane przed systemami czasu rzeczywistego, są bezpośrednim powodem stosowania rozwiązań wieloprocesorowych. Dokonanie podziału zadań na współpracujące podzadania, które zostają przydzielone do odrębnych procesorów, umożliwia skrócenie czasu realizacji zadań w stopniu gwarantującym dochowanie ich ograniczeń czasowych [5]. Innym powodem stosowania rozwiązań wieloprocesorowych jest konieczność zapewnienia odpowiedniej wielkości zasobów potrzebnych do realizacji zadania, których nie jest w stanie dostarczyć pojedynczy system mikroprocesorowy. Do zasobów, które można zwielokrotnić poprzez zastosowanie systemu równoległego złożonego z kilku procesorów, można zaliczyć między innymi pamięć operacyjną oraz układy wejścia i wyjścia (I/O).

Szeregowanie wieloprocesorowych zadań okresowych nie jest w ogólnym przypadku sprawą łatwą [11]. Spowodowane jest to tym, że zadania, do realizacji których wymagany jest co najmniej jeden wspólny procesor, blokują wzajemnie możliwość swego wykonywania do czasu, aż przynajmniej jedno z rozważanych zadań zostanie zakończone. Ponadto w przypadku występowania w systemie dużej liczby zadań wieloprocesorowych o różnych wartościach okresów trudno jest znaleźć taki plan ich uszeregowania, który zapewniałby efektywne wykorzystanie czasu pracy poszczególnych procesorów systemu [6].

Metoda szeregowania wieloprocesorowych zadań okresowych zaproponowana przez autora polega na dokonaniu binaryzacji okresów zadań, w celu zmniejszenia liczby różnych wartości okresów zadań występujących w szeregowanym systemie. Następnie w ramach poszczególnych okresów binarnych wieloprocesorowe i jednoprocesorowe zadania zostają połączone w tzw. superzadania, czyli zadania, do realizacji których wymagana jest jednoczesna dostępność wszystkich procesorów systemu. Po wykonaniu wymienionych czynności, szeregowaniu podlegają wyłącznie superzadania. Ponieważ każde z superzadań wymaga do swej realizacji identycznego zestawu procesorów, żadne dwa superzadania nie mogą być realizowane równocześnie. Zatem superzadania można potraktować jak zwyczajne zadania jednoprocesorowe, z tym że do realizacji takich zadań potrzebny jest procesor specjalnego typu, stanowiący klaster wieloprocesorowy. Rozwiązania wieloprocesorowe stają się obecnie coraz bardziej popularne, a ponadto wydaje się, że w najbliższej przyszłości rynek procesorów zostanie zdominowany przez układy wielordzeniowe. Właśnie procesory wielordzeniowe stanowią rozważane w artykule klastry wieloprocesorowe i głównie z myślą o nich rozważana w artykule modyfikacja algorytmu *Rate Monotonic Scheduling* została zaproponowana. Koszty układów wielordzeniowych są obecnie większe niż klasycznych procesorów jednordzeniowych, jednak ich ceny systematycznie spadają i zapewne w niezbyt odległej przyszłości tego

typu układy będą powszechnie stosowane. Ponadto należy nadmienić, że z punktu widzenia programu szeregującego superzadania wewnętrzna budowa rozważanego klastra nie jest istotna, bowiem poszczególne zadania wieloprocesorowe i jednoprocesorowe zostały już odpowiednio wcześniej połączone w superzadania. Zatem program szeregujący traktuje superzadania dokładnie tak, jak zadania jednoprocesorowe, a w takim wypadku może już zostać bezpośrednio zastosowany do ich szeregowania algorytm *Rate Monotonic Scheduling*. Sposób tworzenia superzadnia przeznaczonego do realizacji przez klaster czteroprocesorowy został przedstawiony na rys. 1. Rozważane superzadanie zostało utworzone z 6 zadań wieloprocesorowych i 7 jednoprocesorowych.



Rys. 1. Ilustracja sposobu tworzenia superzadania z 6 zadań wieloprocesorowych i 7 jednoprocesorowych

Fig. 1. The way of building a single supertask from 6 multiprocessor tasks and 7 uniprocessor tasks

Na rys. 1 przez P1, P2, P3 i P4 oznaczono poszczególne procesory systemu czteroprocesorowego, dla którego szeregowane są zadania wieloprocesorowe i jednoprocesorowe. Natomiast przez K oznaczono czteroprocesorowy klaster zbudowany z procesorów P1, P2, P3 i P4, którego przeznaczeniem jest wykonywanie superzadania utworzonego z 6 zadań wieloprocesorowych i 7 jednoprocesorowych.

Podstawowym problemem jest znalezienie takiego sposobu alokacji poszczególnych zadań wieloprocesorowych i jednoprocesorowych w ramach każdego superzadania, który byłby w stanie zapewnić jak najkrótszy czas realizacji rozważanych superzadań. Problem poszukiwania optymalnego sposobu alokacji zadań w obrębie superzadania, zapewniającego minimalizację czasu jego wykonywania należy do klasy problemów NP-trudnych, w związku z czym dla dużej liczby zadań nie jest możliwe

jego dokładne rozwiązanie. Należy zatem ograniczyć się do poszukiwania rozwiązania suboptymalnego, dającego umiarkowany czas realizacji superzadań, czyli taki, który jest możliwy do zaakceptowania przez algorytm szeregujący superzadania.

Ponadto należy jeszcze zauważyć, że różne sposoby przeprowadzenia binaryzacji okresów zadań, polegające na wyborze różnych wartości okresu bazowego r, prowadzą do różnej liczby zadań wieloprocesorowych i jednoprocesorowych, jakie są alokowane w ramach poszczególnych superzadań, co oczywiście posiada znaczący wpływ na czas realizacji superzadań i dodatkowo komplikuje problem znalezienia optymalnego sposobu alokacji i szeregowania zadań, który zapewniałby jak najmniejsze wykorzystanie czasu pracy procesorów systemu równoległego i gwarantował dochowanie ograniczeń czasowych przez wszystkie zadania.

W celu rozwiązania problemu alokacji i szeregowania zadań wieloprocesorowych i jednoprocesorowych dla systemu równoległego, autor zdecydował się na zastosowanie strategii ewolucyjnej.

4. IMPLEMENTACJA STRATEGII EWOLUCYJNEJ W ZAGADNIENIU SZEREGOWANIA ZADAŃ

W celu implementacji strategii ewolucyjnej należy najpierw zdefiniować sposób zakodowania rozwiązania na materiale genetycznym osobnika.

Pierwsza część materiału genetycznego osobnika służy do zakodowania sposobu wyboru binaryzacji okresów zadań. Jeśli w systemie podlegającym szeregowaniu występuje pewna liczba zadań o K różnych wartościach okresów, wówczas najlepszej binaryzacji należy poszukiwać wśród K binaryzacji istotnych. Rozważane K binaryzacje istotne mogą zostać uszeregowane i ponumerowane ze względu na wzrastająca wartość ich okresów bazowych r. Zatem sposób wyboru binaryzacji może zostać zakodowany na odcinku materiału genetycznego osobnika składającego się z N bitów, gdzie N jest taką liczbą, że 2^N jest najmniejszą naturalną potęgą dwójki większą bądź równą K. Jeśli liczbę binarną zakodowaną na N bitach materiału genetycznego oznaczyć przez L_B, wówczas jako odpowiednia binaryzacja jest wybierana ta, która posiada numer okresu bazowego r_i wyrażony wzorem

$$r_i = 1 + l_B \bmod K, \text{ gdzie } i = 1, 2, 3, \ldots, K \tag{8}$$

Kolejna część materiału genetycznego osobnika służy do zakodowania sposobu przydziału zadań do procesorów. Zaproponowany przez autora sposób kodowania pozwala na uwzględnienie zarówno zadań przeznaczonych dla dedykowanych procesorów, zadań dla procesorów całkowicie arbitralnych, jak również zadań przeznaczonych dla pewnego niepustego podzbioru zbioru procesorów tworzących system równoległy. Rozważany sposób kodowania sposobu przydziału zadań do procesorów zostanie zilustrowany na przykładzie przydziału wybranych zadań dwuprocesorowych w systemie równoległym złożonym z czterech procesorów P1, P2, P3 i P4.

Niech zatem dane będą okresowe zadania dwuprocesorowe o charakterystykach podanych w tab. 1. W tab. 1 zamieszczono w kolejnych kolumnach numer zadania dwuprocesorowego, okres zadania T oraz czasy realizacji dla każdej z sześciu możliwych kombinacji procesorów, np. czas realizacji zadania przez parę procesorów P1 i P2 oznaczono C12, a procesorów P2 i P4 jako C^{24}.

Tabela 1

Charakterystyki zadań dwuprocesorowych podlegających alokacji w systemie czteroprocesorowym

The characteristics of biprocessor tasks that are allocated in four processor system

zadanie	okres zadania T [ms]	czas realizacji C^{12} [ms]	czas realizacji C^{13} [ms]	czas realizacji C^{14} [ms]	czas realizacji C^{23} [ms]	czas realizacji C^{24} [ms]	czas realizacji C^{34} [ms]
z_1	124	23	-	20	-	-	-
z_2	345	14	12	15	17	16	11
z_3	276	-	-	-	56	54	51
z_4	521	36	-	-	-	-	-
z_5	639	45	42	-	-	44	41
z_6	483	-	33	32	36	-	37

Analizując zawartość tab. 1 widać, że dwuprocesorowe zadanie z_1 może zostać zrealizowane na procesorach P1 i P2 lub na procesorach P1 i P4. Inne kombinacje procesorów są dla tego zadania niedopuszczalne, np. z powodu nie dysponowania przez pozostałe procesory odpowiednimi zasobami wymaganymi do realizacji rozważanego zadania. Zatem zadanie z_1 stanowi przykład zadania przeznaczonego dla częściowo arbitralnych procesorów.

Z kolei dwuprocesorowe zadanie z_2 może zostać zrealizowane z wykorzystaniem dowolnej kombinacji dwóch spośród czterech dostępnych procesorów systemu równoległego P1, P2, P3 i P4, czyli przez procesory P1 i P2, P1 i P3, P1 i P4, P2 i P3, P2 i P4 lub P3 i P4. Zatem zadanie z_2 stanowi przykład zadania przeznaczonego dla całkowicie arbitralnych procesorów. Dla kontrastu zadanie z_4 jest przykładem zadania przeznaczonego dla dedykowanych procesorów P1 i P2, ponieważ nie ma innej możliwości realizacji tego zadania, jak właśnie za pośrednictwem tych procesorów. Pozostałe zadania z_3, z_5 i z_6 stanowią przykład zadań przeznaczonych dla częściowo arbitralnych procesorów, ponieważ w przypadku każdego z tych zadań istnieją co najmniej dwa alternatywne sposoby wyboru pary procesorów do realizacji danego zadania.

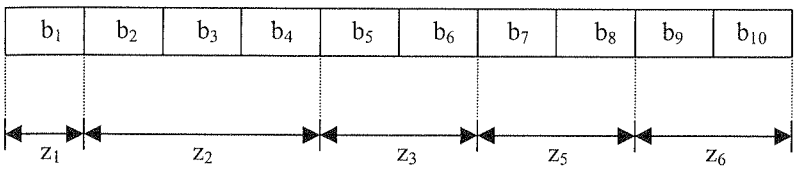
W ogólnym przypadku dla danego zadania istnieje S różnych sposobów przydziału do procesorów systemu równoległego, które można odpowiednio ponumerować od 1 do S, ze względu na wzrastające numery indeksów procesorów, tak jak np. kolumny w tab. 1. Do zakodowania sposobu przydziału zadania do procesorów wykorzystywanych jest N bitów, gdzie 2^N jest najmniejszą naturalną potęgą dwójki, większą bądź równą S. Wtedy liczba binarna L_B zakodowana na N bitach określa sposób wyboru numeru

zestawu procesorów potrzebnych do realizacji danego zadania, zgodnie z następującym wzorem

$$x = 1 + l_B \bmod S \tag{9}$$

We wzorze (9) x oznacza numer sposobu wyboru przydziału danego zadania do procesorów i przybiera wartości od 1 do S.

Postępując zgodnie z opisaną metodą do zakodowania sposobu przydziału zadań z tab. 1 do procesorów potrzeba 10 bitów, których znaczenie zostało przedstawione na rys. 2.



Rys. 2. Ilustracja metody kodowania sposobu przydziału zadań do procesorów

Fig. 2. The illustration of the way of coding of task allocation to processors

Z rys. 2 wynika, że bit b1 koduje sposób przydziału zadania z_1 do procesorów. Jeśli $b_1 = 0$, wówczas zadanie z_1 zostaje przydzielone do procesorów P1 i P2. Natomiast jeśli $b_1 = 1$, wówczas zadanie z_1 zostaje przydzielone do procesorów P1 i P4.

Z kolei w przypadku zadania z_2 możliwe są następujące kombinacje wartości bitów b_2, b_3 i b_4:

- $b_2b_3b_4 = 000$ — zadanie z_2 przydzielone zostaje do procesorów P1 i P2
- $b_2b_3b_4 = 001$ — zadanie z_2 przydzielone zostaje do procesorów P1 i P3
- $b_2b_3b_4 = 010$ — zadanie z_2 przydzielone zostaje do procesorów P1 i P4
- $b_2b_3b_4 = 011$ — zadanie z_2 przydzielone zostaje do procesorów P2 i P3
- $b_2b_3b_4 = 100$ — zadanie z_2 przydzielone zostaje do procesorów P2 i P4
- $b_2b_3b_4 = 101$ — zadanie z_2 przydzielone zostaje do procesorów P3 i P4
- $b_2b_3b_4 = 110$ — zadanie z_2 przydzielone zostaje do procesorów P1 i P2
- $b_2b_3b_4 = 111$ — zadanie z_2 przydzielone zostaje do procesorów P1 i P3

Natomiast do zakodowania sposobu przydziału zadania z_3 do procesorów wykorzystano bity b_5 i b_6, przy czym interpretacja kolejnych kombinacji wartości bitów jest następująca:

- $b_5b_6 = 00$ — zadanie z_3 przydzielone zostaje do procesorów P2 i P3
- $b_5b_6 = 01$ — zadanie z_3 przydzielone zostaje do procesorów P2 i P4
- $b_5b_6 = 10$ — zadanie z_3 przydzielone zostaje do procesorów P3 i P4
- $b_5b_6 = 11$ — zadanie z_3 przydzielone zostaje do procesorów P2 i P3

Podobnie w przypadku zadania z_5 następujące kombinacje wartości bitów określają sposób przydziału tego zadania do procesorów:

- $b_7b_8 = 00$ — zadanie z_5 przydzielone zostaje do procesorów P1 i P2
- $b_7b_8 = 01$ — zadanie z_5 przydzielone zostaje do procesorów P1 i P3

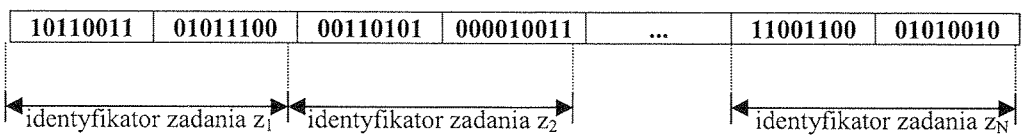
- $b_7b_8 = 10$ — zadanie z_5 przydzielone zostaje do procesorów P2 i P4
- $b_7b_8 = 11$ — zadanie z_5 przydzielone zostaje do procesorów P3 i P4

Natomiast w przypadku zadania z_6 następujące kombinacje wartości bitów określają sposób przydziału rozważanego zadania do poszczególnych procesorów:

- $b_9b_{10} = 00$ — zadanie z_6 przydzielone zostaje do procesorów P1 i P3
- $b_9b_{10} = 01$ — zadanie z_6 przydzielone zostaje do procesorów P1 i P4
- $b_9b_{10} = 10$ — zadanie z_6 przydzielone zostaje do procesorów P2 i P3
- $b_9b_{10} = 11$ — zadanie z_6 przydzielone zostaje do procesorów P3 i P4

Należy zauważyć, że materiał genetyczny osobnika w ogóle nie został wykorzystany do zakodowania sposobu przydziału do procesorów zadania z_4, ponieważ jest to zadanie przeznaczone dla procesorów dedykowanych i w takim przypadku przydział zadania do procesorów jest określony w sposób jednoznaczny.

Z kolei trzecia część materiału genetycznego osobnika służy do zakodowania sposobu szeregowania zadań wieloprocesorowych i jednoprocesorowych w ramach poszczególnych superzadań. W tym celu dla każdego zadania przewidziano po dwa bajty materiału genetycznego, na których zakodowany zostaje tzw. identyfikator zadania (ang. task ID), będący w przypadku każdego osobnika dwubajtową liczbą losową. Fragment materiału genetycznego z zakodowanymi identyfikatorami poszczególnych zadań został przedstawiony na rys. 3.



Rys. 3. Materiał genetyczny osobnika z zakodowanymi identyfikatorami poszczególnych zadań podlegających szeregowaniu

Fig. 3. The genetic material with coded identifications of the tasks that are scheduled

W przypadku, gdy szeregowaniu podlega N zadań, do zakodowania wartości ich identyfikatorów potrzeba 2N bajtów materiału genetycznego. Każdy identyfikator zadania jest liczbą losową z przedziału od 0 do 32767.

W ramach pojedynczego superzadania poszczególne zadania szeregowane są w ten sposób, że ustawione zostają w kolejności wynikającej ze wzrastających wartości ich identyfikatorów. Proces szeregowania zadań zostanie zilustrowany na następującym przykładzie. Niech dany będzie zbiór 12 zadań o identycznych wartościach okresu (oczywiście po wcześniejszej binaryzacji), które zostały przydzielone do procesorów w sposób przedstawiony w tab. 2. W kolejnych kolumnach tab. 2 zamieszczono numer zadania, wartość identyfikatora zadania, czas realizacji zadania, liczbę procesorów potrzebnych do realizacji danego zadania oraz zaznaczono znakiem „x" te z procesorów, do których dane zadanie zostało przydzielone.

Pewien wyjątek stanowią zadania, które do swej realizacji wymagają jednoczesnej dostępności wszystkich procesorów systemu, np. w systemie równoległym zbudowanym z czterech procesorów będą to zadania czteroprocesorowe. W przypadku takich zadań nie jest im przydzielany żaden identyfikator, ponieważ zadania te zawsze należy umieścić na początku każdego planu szeregowania zadań i nie mają one w związku z tym żadnego wpływu na jakość uzyskiwanych rozwiązań, gdyż problem o kombinatorycznej złożoności obliczeniowej występuje dopiero w przypadku poszukiwania planu szeregowania pozostałych zadań, gdzie o efektywności rozwiązania decyduje właśnie kolejność ułożenia poszczególnych zadań.

W przypadku zbioru zadań o charakterystykach zebranych w tab. 2, zadaniem czteroprocesorowym — a więc wymagającym jednoczesnej dostępności wszystkich czterech procesorów systemu równoległego — jest zadanie z_1. Zadaniu z_1 nie został zatem przydzielony żaden identyfikator i w związku z tym zadanie to zostaje umieszczone zawsze na początku każdego planu szeregowania rozważanego zbioru zadań.

Tabela 2

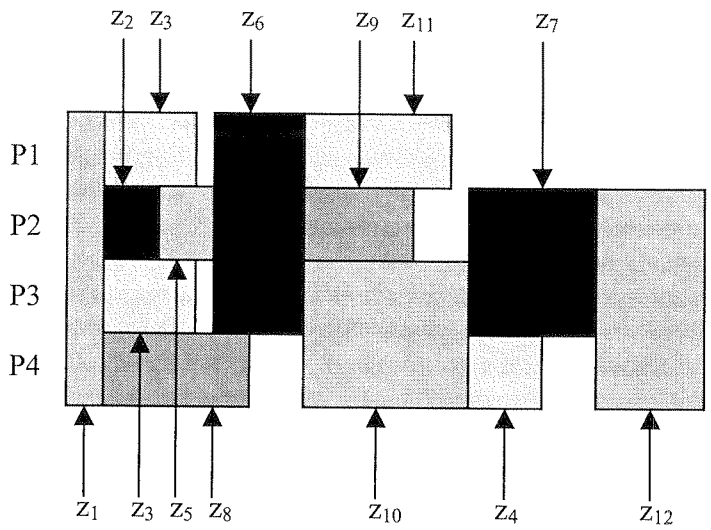
Charakterystyki zadań podlegających szeregowaniu w ramach pojedynczego superzadania

The characteristics of tasks that are scheduled as a single supertask

numer zadania	wartość identyfikatora zadania	czas realizacji zadania C [ms]	liczba procesorów wymaganych do realizacji zadania	P1	P2	P3	P4
z_1	-	2	4	x	x	x	x
z_2	35	3	1	-	x	-	-
z_3	543	5	2	x	-	x	-
z_4	22765	4	1	-	-	-	x
z_5	1290	3	1	-	x	-	-
z_6	1290	5	3	x	x	x	-
z_7	27634	7	2	-	x	x	-
z_8	9082	8	1	-	-	-	x
z_9	14987	6	1	-	x	-	-
z_{10}	20384	9	2	-	-	x	x
z_{11}	28921	8	1	x	-	-	-
z_{12}	30072	6	3	-	x	x	x

Sposób szeregowania zadań o charakterystykach z tab. 2 został zilustrowany na rys. 4.

Analizując rys. 4 widać, że całkowity czas realizacji planu szeregowania 12 zadań równy jest sumie czasów wykonywania zadań z_1, z_2, z_5, z_6, z_{10}, z_7 i z_{12}, a zatem wynosi 35 ms. Nie jest to z pewnością plan optymalny i wybór innego sposobu przydziału zadań do procesorów oraz kolejności ułożenia zadań mogłyby w rezultacie dać krótszy czas realizacji rozważanego zbioru zadań. Znalezienie planów alokacji i szeregowania



Rys. 4. Ilustracja planu szeregowania zadań dla czterech procesorów

Fig. 4. The illustration of task schedule for four processors

zadań o jak najkrótszym czasie realizacji jest właśnie zadaniem, którego rozwiązanie jest celem strategii ewolucyjnej.

Jak już zostało powiedziane na samym początku planu szeregowania zadań zostaje umieszczone zadanie czteroprocesorowe z_1, ponieważ jest to zadanie wykorzystujące wszystkie procesory systemu równoległego. Następnie brane jest zadanie z_2, które ma najmniejsza wartość identyfikatora równą 35. Następnie brane jest zadanie z_3 o kolejnej, najmniejszej co do wielkości wartości identyfikatora równej 543. Następną najmniejszą wartością identyfikatora jest 1290. Taką wartość identyfikatora posiadają dwa zadania z_5 i z_6. Przypadek, w którym co najmniej dwa różne zadania mają identyczne wartości identyfikatorów jest niezwykle rzadki, choć oczywiście może wystąpić w praktyce. Wówczas jako pierwsze brane jest to zadanie, które posiada mniejszy numer porządkowy, czyli zadanie z_5, a dopiero później z_6. Kolejna najmniejsza wartość identyfikatora wynosi 9082, a zatem jako następne do planu szeregowania dołączane jest zadanie z_8. Następnie brane są kolejno zadania z_9 (wartość identyfikatora równa 14987), z_{10} (wartość identyfikatora równa 20384), z_4 (wartość identyfikatora równa 22765), z_7 (wartość identyfikatora równa 27634), z_{11} (wartość identyfikatora równa 28921) i jako ostatnie zadanie z12 (wartość identyfikatora równa 30072). Kolejną, niezwykle istotną kwestią związaną z implementacją strategii ewolucyjnych jest zdefiniowanie postaci tzw. funkcji dopasowania, która pozwoli na wzajemne porównanie rozwiązań reprezentowanych

przez materiały genetyczne dwóch różnych osobników i wybranie, podczas realizacji operacji selekcji turniejowej, osobnika przedstawiającego sobą lepsze rozwiązanie.

Dla potrzeb realizacji genetycznej operacji selekcji, autor zdefiniował dwie funkcje dopasowania f_1 i f_2, które wykorzystywane są w dwóch różnych, wzajemnie wykluczających się przypadkach.

W przypadku pierwszym co najmniej jedno z szeregowanych superzadań narusza swoje ograniczenie czasowe. W takim wypadku rozwiązanie reprezentowane przez materiał genetyczny danego osobnika jest niedopuszczalne, ponieważ ograniczenia czasowe mają być dochowane dla wszystkich zadań. W takim wypadku funkcji f_1 zostaje przypisana wartość zero ($f_1 = 0$), natomiast funkcji f_2 zostaje przypisana wartość, będąca sumą odcinków czasowych, o jakie poszczególne superzadania przekraczają swoje ograniczenia czasowe.

W przypadku drugim wszystkie superzadania dochowują swoich ograniczeń czasowych, zatem funkcji f_2 zostaje przypisana wartość zero ($f_2 = 0$). Natomiast funkcji f_1 przypisana zostaje wartość równa sumarycznemu stopniowi wykorzystania klastra wieloprocesorowego przez realizowane na nim superzadania.

W procesie selekcji turniejowej porównywane są ze sobą zawsze dwa osobniki, w związku z czym należy rozpatrzyć trzy różne, możliwe do wystąpienia przypadki.

W przypadku pierwszym porównywane są ze sobą dwa osobniki, dla których $f_1 = 0$. Zatem oba osobniki reprezentują sobą rozwiązania, które są niedopuszczalne, ponieważ w ich przypadku co najmniej jedno z szeregowanych superzadań narusza ograniczenie czasowe. Jednak operacja selekcji turniejowej musi zostać mimo wszystko zrealizowana i osobnik reprezentujący gorsze rozwiązanie musi zostać usunięty z populacji. Jako osobnik gorszy, przegrywający w turnieju, wybierany jest ten, w przypadku którego wartość funkcji f_2 jest większa, co oznacza, że w rozwiązaniu reprezentowanym przez materiał genetyczny tego osobnika, łączny czas przekroczeń ograniczeń czasowych przez poszczególne superzadania jest większy.

Z kolei w przypadku drugim porównywane są ze sobą dwa osobniki, dla których $f_2 = 0$, co oznacza, że oba reprezentują sobą rozwiązania w pełni dopuszczalne, tzn. takie, w przypadku których wszystkie superzadania dochowują swoich ograniczeń czasowych. Jednak jako rozwiązanie lepsze wybierane jest to, w przypadku którego funkcja f_1 ma mniejszą wartość, co oznacza, że klaster wieloprocesorowych wykorzystywany jest w mniejszym stopniu, a co za tym idzie większy jest margines zapasu czasu pracy wieloprocesorowej jednostki obliczeniowej, który może zostać wykorzystany, np. przez występujące dodatkowo w systemie zadania sporadyczne. Fakt ten ma również znaczenie w przypadku dłuższego niż przewiduje specyfikacja czasu wykonywania któregoś z superzadań, powstałego w przypadku reakcji na nietypowy zestaw danych wejściowych bądź nietypowy scenariusz interakcji systemu z otoczeniem. W takim wypadku dysponowanie przez wieloprocesorowa jednostkę obliczeniową dodatkowym nadmiarem wolnego czasu pracy może uchronić dane superzadanie przed naruszeniem jego ograniczenia czasowego.

Wreszcie w przypadku trzecim porównywane są ze sobą dwa rozwiązania, takie że w przypadku pierwszego $f_1 = 0$, natomiast w przypadku drugiego $f_2 = 0$. W takim przypadku, jako rozwiązanie lepsze zostanie wskazane zawsze rozwiązanie drugie ($f_2 = 0$), ponieważ każde rozwiązanie, w przypadku którego są dochowane ograniczenia czasowe dla wszystkich superzadań jest lepsze od każdego innego rozwiązania, w przypadku którego choć jedno z superzadań narusza swoje ograniczenie czasowe ($f_1 = 0$).

5. ZAKOŃCZENIE

Cykliczna realizacja wielu tysięcy operacji genetycznych, takich jak krzyżowanie, mutacja i selekcja prowadzi w efekcie do systematycznego wzrostu jakości rozwiązań reprezentowanych przez poszczególne osobniki populacji i pozwala na ostateczne wyłonienie rozwiązania najlepszego, gwarantującego, po pierwsze, dochowanie ograniczeń czasowych dla wszystkich superzadań i, po drugie, umiarkowany stopień wykorzystania czasu pracy wieloprocesorowej jednostki obliczeniowej.

Ponadto istotną zaletą zaproponowanej przez autora metody szeregowania jest jej duża uniwersalność. W praktyce znanych jest niewiele wszechstronnych algorytmów szeregowania zadań. Wręcz przeciwnie, istnieją oddzielne algorytmy przewidziane do szeregowania zadań jednoprocesorowych i wieloprocesorowych, a w przypadku tych drugich występują odrębne algorytmy przeznaczone do szeregowania zadań dla procesorów dedykowanych i procesorów arbitralnych. Natomiast algorytm przedstawiony w niniejszej pracy uwzględnia wszystkie wymienione przypadki, ponieważ może zostać wykorzystany do szeregowania zadań wieloprocesorowych, jak i jednoprocesorowych, a także do zadań przeznaczonych dla procesorów arbitralnych i dedykowanych. Co więcej opracowany przez autora algorytm pozwala na uwzględnienie wszelkich przypadków pośrednich, mieszczących się pomiędzy dwoma skrajnymi biegunami zadań przeznaczonych dla procesorów dedykowanych i arbitralnych, bowiem możliwe jest szeregowania za jego pomocą również zadań przeznaczonych do szeregowania jedynie na pewnym ograniczonym podzbiorze zbioru procesorów systemu równoległego — jest to przypadek zadań o częściowej arbitralności przydziału procesora.

W przyszłości należy oczekiwać, że zapotrzebowanie na moc obliczeniową, zgłaszane ze strony aplikacji związanych z obszarem przemysłowych zastosowań w dziedzinie automatyki i sterowania, będzie nieustannie wzrastać. Ponadto wydaje się, że możliwości podnoszenia mocy obliczeniowej pojedynczych procesorów poprzez podnoszenie częstotliwości zegara taktującego bliskie już są barier technologicznych wynikających z ziarnistej struktury budowy materii. Na przykład w technologii 30nm wymiary przyrządów półprzewodnikowych równe są długości odcinka, na którym ustawiono w rządku około 300 atomów krzemu. Wszystko to powoduje, że zbliżanie się do kresu możliwości wzrostu mocy obliczeniowej pojedynczych procesorów wymusza intensywne badania w zakresie rozwiązań wieloprocesorowych i obliczeń równoległych.

Niestety, już samo zrównoleglenie algorytmu nie jest sprawą prostą, a ponadto istnieje wiele zagadnień, dla których rozwiązujące je algorytmy z wielkim trudem poddają się paralelizacji. Co więcej, uzyskawszy już równoległą postać algorytmu rozwiązującego dane zagadnienie obliczeniowe, projektant staje przed kolejnym wyzwaniem, jakim jest opracowanie odpowiedniego planu przydziału zadań do jednostek obliczeniowych, a następnie poszukiwanie schematów efektywnego szeregowania rozważanych zadań w ramach poszczególnych procesorów.

Ponieważ w systemach automatyki przemysłowej, czyli w komputerowych systemach sterujących występują przede wszystkim zadania o charakterystykach cyklicznych, autor skupił swą uwagę przed wszystkim na szeregowaniu zadań periodycznych, aczkolwiek zadania aperiodyczne również mogą zostać łatwo sprowadzone do postaci zadań cyklicznych, stosując technikę cyklicznego serwera zadań sporadycznych.

Ponadto w literaturze przedmiotu dotyczącej problematyki szeregowania zadań podawanych jest niewiele przykładów algorytmów szeregujących, które charakteryzowałyby się dużą uniwersalnością i mogłyby być zastosowane do różnych typów zadań. W tym kontekscie ważne jest, że przedstawione w artykule, zaproponowane przez autora rozwiązanie odznacza się dużym stopniem uniwersalności, ponieważ pozwala na poszukiwanie efektywnych planów szeregowania dla zadań przeznaczonych do realizacji na dowolnej liczbie procesorów i to zarówno w przypadku zadań przeznaczonych dla procesorów dedykowanych, jak i arbitralnych.

W pracy został rozważony przypadek wieloprocesorowego systemu równoległego o topologii połączeń typu każdy z każdym, tzn. takiego, w przypadku którego istnieje bezpośrednie łącze komunikacyjne umożliwiające wymianę danych pomiędzy dowolnie wybraną parą procesorów. Jednak uzyskane wyniki mogą zostać stosunkowo łatwo przeniesione na przypadki innych topologii systemów wieloprocesorowych o niepełnej sieci połączeń, takich jak np. hipersześcian, krata dwuwymiarowa, walec czy torus.

6. BIBLIOGRAFIA

1. T. Szmuc, G. Motet: *Specyfikacja i projektowanie oprogramowania czasu rzeczywistego*, Krakowskie Centrum Informatyki Stosowanej, Kraków, 1998.
2. J. Werewka, T. Szmuc: *Analiza i projektowanie systemów czasu rzeczywistego o różnym stopniu rozproszenia*, Polskie Towarzystwo Informatyczne — Oddział Małopolski, Kraków, 2001.
3. T. Szmuc: *Zaawansowane metody tworzenia oprogramowania czasu rzeczywistego*, Krakowskie Centrum Informatyki Stosowanej, Kraków, 1998.
4. T. Szmuc:*Modele i metody inżynierii oprogramowania systemów czasu rzeczywistego*, AGH — Uczelniane Wydawnictwa Naukowo-Dydaktyczne, Kraków, 2001.
5. J. Miloško, E. Kotov: *Algorithms, software and hardware of parallel computers*, VEDA — Publishing House of the Slovak Academy of Sciences, Bratislava 1984.
6. O. Kwon, K.Y. Chwa: *Approximation algorithms for general parallel task scheduling*, Information Processing Letters 81 (2002), pp. 143–150.
7. A. Czajka, J. Nawrocki: *Szeregowanie zadań o okresach binarnych w systemach silnie uwarunkowanych czasowo*, I Krajowa Konferencja Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim, Kraków, 1997, ss. 669—676.

8. J. Nawrocki, A. Czajka: *Binaryzacja okresów zadań cyklicznych*, VII Konferencja Systemy Czasu Rzeczywistego, Kraków, 2000, ss. 41–51.
9. J. Werewka: *Zagadnienia alokacji zadań w rozproszonych systemach czasu rzeczywistego*, Elektrotechnika, Tom 14, Zeszyt 4, 1995, ss. 479–488.
10. C. Hsueh, K.J. Lin: *Scheduling real-time systems with end-to-end timing constraints using the distributed pinwheel model*, IEEE Transaction on Computers, vol. 50, no. 1, 2001, ss. 51–68.
11. O. H. Kwon, K.Y. Chwa: *Approximation algorithms for general parallel task scheduling*, Information Processing Letters 81, 2002, ss. 143–150.
12. D. Ye, G. Zhang: *On-line scheduling with extendable working time on a small number of machines*, Information Processing Letters 85, 2003, ss. 171–177.
13. J. H. Kim, K. Y. Chwa: *Online deadline scheduling on faster machines*, Information Processing Letters 85, 2003, ss. 31–37.
14. K. G. Shin, P. Ramanathan: *Real-time computing: A new discipline of computer science and engineering*, Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 6–24.
15. L. Liu, J.W. Layland: *Scheduling algorithms for multiprogramming in a hard real time environment*, Journal on Assoc. Comput. Mach., vol. 20, no. 1, 1973, pp. 46–61.
16. T. Stoyenko, P. Baker: *Real-time schedulability-analyzable mechanizms in Ada9X*, Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 95–107.
17. K. Ramanathan, J.A. Stankovic: *Scheduling algorithms and operating systems support for real-time systems*, Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 55–67.
18. A. Zalewski: *What every engineer needs to know about rate monotonic scheduling: A tutorial*, Odessa, TX, 1995, pp. 321–335.
19. S. Tanenbaum: *Rozproszone systemy operacyjne*, PWN, Warszawa, 1997.

M. GAJER

THE APPLICATION OF BINARISATION OF TASKS PERIODS IN MULTIPROCESSOR SYSTEMS

Summary

Nowadays the real-time systems find application in many branches of industry, science and transport. The most important factor in the real-time systems is the time of execution of tasks. The results that are correct but delivered with the violation of task deadline are useless. In the case of hard real-time systems the violation of task deadline can lead to a catastrophe or even loss of human life. This is the reason why the task scheduling theory developed strongly over the last years.

In the computer systems with hard real-time constraints the most popular task scheduling algorithm is Rate Monotonic Scheduling (RMS). In RMS all tasks are assigned priorities. The rule is the shorter is task period the higher priority the task obtains. In a given moment the task that is read for execution and has the highest priority is executed. If any other task with higher priority arrives the task that is executed is pre-empted and the task that came is then executed.

Together with RMS binarisation is often used. Binarisation is a technique that transforms the periods of tasks in the way that only tasks with harmonic values of periods exist. Up till now binarisation was used only for a uniprocessor systems. This author proposed a new solution in which binarisation is used for multiprocessor systems as well.

Scheduling and allocation of multiprocessor tasks in a multiprocessor system is NP- complete optimisation problem. This is the reason why the optimal solution for scheduling of multiprocessor task can

not be found in a reasonable time. In such case only suboptimal solutions can be found. This author proposed evolutionary algorithm as a solution to multiprocessor tasks allocation and scheduling.

The way in which the order of execution of task is coded on the genetic material was illustrated on the example of scheduling the tasks for four processor systems. The genetic material also allows to code the way in which the proper binarisation is chosen.

There are very few tasks scheduling algorithms that could be used for a general class of tasks. There are different scheduling algorithms for uniprocessor task and multiprocessor task and for tasks that are executed on arbitrary or dedicated processors. The algorithm of tasks scheduling and allocation that was proposed by this author is a general one, because it can be used for uniprocessor and multiprocessor tasks and for dedicated, arbitrary and partially arbitrary processors.

Keywords: binarisation of periods of tasks, multiprocessor systems, multiprocessor tasks, tasks scheduling

DYSK

nada
zani
gory
śred
zwią
niu
prze
właś
dost

Słow

Dynar
leinformat
i bardziej
zań techn
nych. Dlat

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 563–589

DYSKRETNE ALGORYTMY STEROWANIA SZYBKOŚCIĄ NADAWANIA DANYCH W POŁĄCZENIOWYCH SIECIACH TELEINFORMATYCZNYCH

TOMASZ MOLIK, ANDRZEJ BARTOSZEWICZ

Instytut Automatyki Politechniki Łódzkiej,
ul. Stefanowskiego 18/22, 90-924 Łódź
Email:andpbart@p.lodz.pl

Otrzymano 2005.10.14
Autoryzowano 2005.12.08

W artykule zaprezentowano dwa nowe dyskretne algorytmy sterowania szybkością nadawania danych w połączeniowych sieciach teleinformatycznych. Proponowane rozwiązania wykorzystują predyktor Smitha oraz dyskretny regulator proporcjonalny. Jeden z algorytmów uwzględnia także informację o stanie sieci dostarczaną przez sprzężenie bezpośrednie od dostępnego pasma, co przyczynia się do zmniejszenia wahań czasu opóźnienia związanego z kolejkowaniem danych w węźle wąskiego gardła. Wykazano, że przy spełnieniu odpowiednich warunków przedstawione algorytmy pozwalają całkowicie wyeliminować przeciążenia sieci oraz zapewniają pełne wykorzystanie łącza. Algorytmy zachowują swoje właściwości zarówno w stanie ustalonym sieci, jak i wtedy gdy liczba nadajników oraz dostępne pasmo mogą ulegać zmianie.

Słowa kluczowe: kontrola przeciążeń, sieci ATM, predyktor Smitha, sterowanie obiektów z opóźnieniem

1. WSTĘP

Dynamicznie rosnące zapotrzebowanie na szybkie przesyłanie danych kanałami teleinformatycznymi, wzrost liczby użytkowników sieci rozległych oraz ich coraz większe i bardziej zróżnicowane wymagania stwarzają potrzebę poszukiwania nowych rozwiązań technicznych, jak również doskonalenia już istniejących technik transmisji danych. Dlatego w ostatnim czasie dużego znaczenia nabrały prace nad udoskonalaniem

mechanizmów wykorzystywanych do przesyłania danych w połączeniowych sieciach teleinformatycznych.

Działanie takich sieci opiera się na zintegrowaniu technik komutacji łączy oraz komutacji pakietów i dzięki temu, że ma zalety obu tych sposobów transmisji danych, stało się atrakcyjną techniką przesyłania informacji. W związku z tym, w dalszej części artykułu rozważane będzie zagadnienie sterowania przepływem danych w połączeniowych sieciach teleinformatycznych. Analizowane będzie działanie sieci w takim trybie – jak na przykład tryb ABR (ang. *Available Bit Rate*) w sieci ATM (ang. *Asynchronous Transfer Mode*) — który pozwala wykorzystać mechanizm sprzężenia zwrotnego do regulacji szybkości nadawania danych przez źródła. Mechanizmy tego typu mogłyby także znaleźć zastosowanie w sieciach MPLS (ang. *Multiprotocol Label Switching*).

Problem kontroli przeciążeń w szerokopasmowych sieciach połączeniowych jest obecnie tematem wielu projektów badawczych i publikacji. W ostatnich latach przedstawiono algorytmy znacznie poprawiające mechanizmy sterowania przepływem danych. Pierwsze przeznaczone do tego celu algorytmy używały pojedynczego bitu ustawianego w zależności od stanu sieci w przesyłanych komórkach ATM. Skuteczność mechanizmów zapobiegających powstawaniu przeciążeń, których działanie oparte było na pojedynczym bicie okazała się jednak niewystarczająca. Dlatego wkrótce zaproponowano nowy schemat, który do przesyłania informacji zwrotnej o stanie sieci wykorzystywał specjalnie w tym celu skonstruowane komórki sterujące RM (ang. *Resource Management*). Komórki RM generowane były przez nadajnik co stałą liczbę N (zazwyczaj $N = 32$) komórek z danymi i zawierały wielobitowe pole ER (ang. *Explicit Rate*) przeznaczone do przenoszenia informacji o oczekiwanej szybkości nadawania źródła. Opis wcześniej proponowanych mechanizmów sterowania ruchem w sieci, takich jak BECN, DECbit, EFCI, PRCA, EPRCA, OSU, ERICA, zaprezentował Jain w pracy [7]. Opracowany przez tego samego autora algorytm ERICA [9] (ang. *Explicit Rate Indication for Congestion Avoidance*), a następnie jego ulepszona wersja ERICA+ [8], stały się przedmiotem wielu badań i publikacji oraz zostały zaakceptowane przez ATM Forum i włączone do oficjalnej dokumentacji zarządzania ruchem w sieci ATM [18]. Algorytm ERICA wyznacza szybkość nadawania danych na podstawie liczby zestawionych połączeń i stopnia obciążenia sieci. Nie wymaga skomplikowanych obliczeń, dzięki czemu jest prosty i tani w implementacji. Algorytm ten zapewnia dobre warunki pracy układu, a prowadzone badania symulacyjne pokazywały, że przeważnie (chociaż nie zawsze) umożliwia on szybkie osiągnięcie stanu, w którym spełnione są kryteria sprawiedliwego, w sensie *max-min fairness*, podziału pasma. Dokładny opis działania i właściwości algorytmu ERICA, opracowany na podstawie zebranych wyników analiz i badań prezentowanych w różnych publikacjach, przedstawili Kalyanaraman i in. w pracy [10].

Ciekawych spostrzeżeń na temat niekorzystnego efektu synchronizacji szybkości nadawania występującego w niektórych algorytmach bazujących na analizie długości kolejki w węźle dokonali Jae Yong Lee i in. w artykule [12]. Przedstawili oni, na przykładzie algorytmu EPRCA, rozwiązanie tego problemu, które polegało na wprowadze-

niu schematu probabilistycznej detekcji przeciążeń. Imer i in. w pracy [5] szczegółowo przedstawili teoretyczne podejście do projektowania regulatorów szybkości nadawania danych w sieci ATM. Autorzy rozważali między innymi takie aspekty kontroli przeciążeń jak model przekazywania danych sterujących, kryteria oceny efektywności działania algorytmów, budowa i działanie przełącznika ATM oraz sposób podziału pasma między poszczególne obwody. Proponowane w pracy [5] nowe algorytmy sterowania, wykorzystując metody stochastyczne, wyznaczają szybkość nadawania poprzez minimalizację odpowiedniej funkcji celu. W artykule [11] Laberteaux i in. zaproponowali zastosowanie algorytmów adaptacyjnych do sterowania prędkością nadawania danych. Autorzy pokazali, że ich strategie umożliwiają lepsze wykorzystanie zasobów sieciowych i szybszą zbieżność układu do stanu ustalonego niż inne algorytmy wcześniej opisywane w literaturze. Inne interesujące podejście do problemu kontroli przeciążeń w połączeniowych sieciach teleinformatycznych przedstawili Quet i in. w pracy [17]. Rozwiązali oni problem sterowania ruchem w węźle wąskiego gardła obsługującym wiele połączeń przez minimalizację odpowiedniej normy H∞. Wykazali, że zaproponowany sposób wyznaczania szybkości nadawania danych przez źródła zapewnia stabilność układu nawet wtedy, gdy opóźnienia w poszczególnych obwodach wirtualnych nie są z góry znane i mogą ulegać zmianom w czasie pracy układu. Do regulacji szybkości nadawania proponowano także klasyczne regulatory PD [13] i PID [3] oraz rozmyte regulatory PID [16]. Do sterowania ruchem w sieci były również stosowane sztuczne sieci neuronowe. Jagannathan i Talluri w artykule [6] pokazali, że ich regulator wykorzystujący taką sieć, może zapewnić stabilność układu i pożądaną jakość usług QoS (*Quality of Sernice*).

Ze względu na istotne opóźnienia występujące w sieci, w pracach [1, 2, 4, 14, 15], proponowano zastosowanie w układzie regulacji predyktora Smitha. W artykule [14] Mascolo rozważał problem sterowania szybkością nadawania danych w pojedynczym obwodzie wirtualnym. W zaproponowanym algorytmie szybkość nadawania wyznaczana jest na podstawie porównania wartości zadanej długości kolejki w węźle stanowiącym wąskie gardło sieci z aktualną długością tej kolejki i liczbą komórek „w locie", czyli tych komórek dla których szybkość nadawania została już wyznaczona, ale które — ze względu na duże czasy propagacji sygnału w sieci — nie dotarły jeszcze do węzła wąskiego gardła. Przedstawione rozwiązanie umożliwia całkowitą eliminację niebezpieczeństwa przepełnienia bufora wąskiego gardła, gwarantuje stuprocentowe wykorzystanie łącza oraz zapewnia szybką zbieżność długości kolejki do stanu ustalonego. Ten sam autor w artykule [15] proponuje zastosowanie predykatora Smitha w układzie zawierającym wiele obwodów wirtualnych. Dalsze badania nad sterowaniem szybkością nadawania danych w trybie ABR z zastosowaniem predyktora Smitha prowadzili Gomez-Stern i in. W pracy [4] zaproponowali oni algorytm wykorzystujący regulator PI, który pozwala zmniejszyć średnią długość kolejki w buforze wąskiego gardła i jej wrażliwość na zmiany dostępnego pasma. Z kolei w artykule [1] zaproponowano nieliniowe algorytmy regulacji prędkości nadawania danych w połączeniowych sieciach teleinformatycznych, które eliminują utratę danych i zapewniają

pełne wykorzystanie łącza stanowiącego wąskie gardło sieci nawet wtedy, gdy czasy propagacji w poszczególnych obwodach wirtualnych nie są dokładnie znane, a jedynie mogą być oszacowane w pewnym przybliżeniu. Przedstawione w artykule [1] algorytmy zapewniają omówione właściwości sieci przy wykorzystaniu mniejszych zasobów niż podobne strategie sterowania wcześniej opisane w literaturze. Ponieważ współczesne układy regulacji realizowane są zazwyczaj jako układy dyskretne, w artykule [2] zaproponowano podobne rozwiązanie jak we wcześniej omówionych pracach, jednak realizowane z uwzględnieniem niezerowego okresu dyskretyzacji sygnału sterującego. Pokazano, że także w przypadku użycia regulatora dyskretnego zastosowanie predyktora Smitha pozwala wyeliminować niebezpieczeństwo utraty danych i zapewnia pełne wykorzystanie dostępnego pasma. Należy jednak zauważyć, że proponowane dotychczas algorytmy [1, 2, 4, 14, 15] działają prawidłowo tylko wtedy, gdy liczba obwodów wirtualnych jest stała i z góry znana. Ponadto wszystkie omawiane strategie sterowania zapewniają wyłącznie równy podział dostępnego pasma, co z praktycznego punktu widzenia, stanowi ich istotną wadę.

W przeciwieństwie do omówionych strategii sterowania, w tym artykule zaproponowano dwa układy regulacji szybkości nadawania danych w połączeniowej sieci teleinformatycznej, w której liczba źródeł może ulegać zmianie. Wykazano, że zaproponowane algorytmy nie powodują przepełnienia bufora wąskiego gardła, a przy spełnieniu odpowiednich, przedstawionych w rozdziałach 3 i 4 warunków, zapewniają pełne wykorzystanie dostępnego pasma. Proponowane rozwiązania umożliwiają dowolny — a więc także sprawiedliwy w sensie *max-min fairness* — podział dostępnego pasma, co jest ich istotną zaletą w porównaniu z algorytmami wcześniej proponowanymi w literaturze.

2. MODEL SIECI

Sieć rozważana w tym artykule składa się z nadajników, węzłów pośredniczących i odbiorników. Każdy z węzłów pośredniczących ma bufor ograniczonej pojemności, w którym przechowuje dane oczekujące na przesłanie do następnego węzła lub do odbiornika (jeśli dany węzeł jest ostatnim węzłem na ścieżce przesyłu danych). Z każdym nadajnikiem związany jest pojedynczy obwód wirtualny, którego parametry określane są podczas zestawiania połączenia. O tym, czy dane połączenie może być zrealizowane decydują odpowiednie procedury kontroli dołączeń. Liczba połączeń może być zmienna i jej przewidzenie a priori nie jest możliwe. Przyjęto, że tylko jeden z węzłów jest węzłem wąskiego gardła. Sieć przekazuje do wszystkich źródeł informację zwrotną w specjalnie do tego celu przeznaczonych jednostkach sterujących (dla sieci ATM są to komórki RM), które mają priorytet przed innymi jednostkami danych. Oznacza to, że przy przesyłaniu jednostek sterujących nie występują opóźnienia związane z oczekiwaniem na wysłanie. Zatem, dla każdego źródła czas *RTT* (ang. *Round Trip Time*) pełnego obiegu jednostki sterującej jest stały. Przyjęto założenie, że ten czas dla każdego źródła jest wielokrotnością niezerowego okresu próbkowania układu T.

Jednostki sterujące generowane są przez nadajnik w odstępach czasu T. Dla j-tego źródła, czas RTT_j można wyrazić jako sumę czasu T_{fj} przejścia jednostki sterującej od j-tego nadajnika do węzła wąskiego gardła i czasu T_{bj} stanowiącego czas przejścia tej jednostki od węzła wąskiego gardła do odbiornika i z powrotem do nadajnika o numerze j.

W dalszych rozważaniach długość kolejki danych w buforze wąskiego gardła w chwili t oznaczona została przez $x(t)$, natomiast wartość zadana tego sygnału przez x_d. Ponieważ przed nawiązaniem połączeń bufor wąskiego gardła jest pusty, więc $x(t < 0) = 0$. Symbolem $a(t)$ oznaczono łączną szybkość generowaną przez algorytm w chwili t, natomiast symbolem $a_j(t)$ szybkość z jaką nadaje j-te źródło w chwili t. Szybkość $a(t)$ generowana jest w dyskretnych chwilach czasu kT, gdzie $k = 0, 1, 2, \ldots$. Założono, że dla j-tego połączenia szybkość nadawania $a_j(t < T_{bj}) = 0$. Ponieważ szybkość a_j z jaką nadaje j-te źródło jest wyznaczana w węźle wąskiego gardła, a następnie z opóźnieniem T_{bj} przekazywana do źródła, więc $a_j(t) = \beta_j(kT)a(kT)$, gdzie k jest największą liczbą całkowitą większą od lub równą zeru, taką że $kT \leq t - T_{bj}$, a $\beta_j(kT)$ określa część szybkości $a(kT)$ przydzieloną dla j-tego źródła w chwili kT. Łączna szybkość $a(kT)$ dzielona jest w taki sposób, że

$$\sum_{j=1}^{n(kT)} \beta_j(kT) = 1 \tag{1}$$

gdzie $n(kT)$ oznacza liczbę nadajników branych pod uwagę podczas podziału dostępnego pasma w chwili kT. Zależność (1) stanowi jedyny warunek dotyczący podziału łącznej szybkości narzucony przez algorytm sterowania. W związku z tym, możliwe jest zastosowanie różnego rodzaju metod podziału pasma, na przykład uwzględniających minimalną szybkość nadawania określoną parametrem MCR (ang. *Minimum Cell Rate*). Najprostszy sposób podziału polega na równym podziale szybkości między wszystkie obsługiwane w danej chwili obwody. Przy takim podziale — który stosowano między innymi w artykułach [1, 2, 4, 14, 15] — dla każdej chwili kT mamy $\underset{j=1,\ldots,n(kT)}{\forall} \beta_j(kT) = 1/n(kT)$.

W dalszej części artykułu przez $d(t)$ oznaczono dostępne pasmo wąskiego gardła w chwili t, natomiast wykorzystywaną część dostępnego pasma wąskiego gardła w chwili t oznaczono symbolem $h(t)$. Maksymalne dostępne pasmo wąskiego gardła wynosi $d_{\max}$. Jeśli w chwili t bufor węzła zawiera dane oczekujące na wysłanie, to $h(t) = d(t)$. W przeciwnym razie, tj. gdy bufor węzła jest pusty, szybkość $h(t)$ z jaką dane opuszczają węzeł jest równa szybkości ich wpływania do węzła. Zatem

$$\underset{t>0}{\forall} 0 \leq h(t) \leq d(t) \leq d_{\max} \tag{2}$$

Symbolem $A(kT)$ oznaczono ilość danych „w locie" w chwili kT, czyli sumę danych, dla których do chwili kT wyznaczona została szybkość z jaką zostały lub zostaną nadane i które nie dotarły jeszcze do bufora wąskiego gardła. Wprowadzamy dodatkowo

funkcję dwóch zmiennych $B(\tau, \alpha)$, która określa ilość danych, które wpłynęły do bufora wąskiego gardła w czasie od τ do $\tau + \alpha$.

Definiujemy sposób włączania się nowego źródła do układu. Oznaczmy to źródło indeksem j. Przyjmijmy, że w chwili t_{j0} źródło j wysyła pierwszą jednostkę sterującą, zgłaszając w ten sposób chęć włączenia się do układu. Definiujemy liczbę k_{j0} jako najmniejszą liczbę całkowitą, taką że $k_{j0}T \geq t_{j0} + T_{fj}$. Dla czasu $t \in [t_{j0}, k_{jo}T + T_{bj})$, źródło o indeksie j nie nadaje jednostek z danymi, lecz jedynie jednostki sterujące. Po czasie $t_{j0} + T_{fj}$ węzeł wąskiego gardła zacznie otrzymywać jednostki sterujące pochodzące od j-tego nadajnika, w wyniku czego dowie się o istnieniu dodatkowego źródła danych. Od chwili $t_{j0} + T_{fj}$ algorytm będzie uwzględniał dodatkowy nadajnik w procesie generowania szybkości. Ponieważ zaproponowane algorytmy wyznaczają szybkości w dyskretnych chwilach czasu kT, to pierwsza wartość szybkości, uwzględniająca nowo dołączone źródło, wyznaczona zostanie w chwili $t = k_{j0}T$. Pierwsze jednostki z danymi pochodzące od źródła o numerze j dotrą do węzła wąskiego gardła w chwili $k_{j0}T + RTT_j$.

Następnie definiujemy sposób odłączania się źródła od układu. Przyjmujemy dodatkowe założenie dotyczące konstrukcji jednostek sterujących. Każde źródło uczestniczące w wymianie informacji wysyłając jednostkę sterującą zapisuje w niej ilość danych, które ma jeszcze do wysłania. Weźmy pod uwagę j-te źródło. Węzeł wąskiego gardła dysponując informacją o generowanych wcześniej szybkościach dla j-tego źródła oraz informacją o ilości danych pozostałych do wysłania może sprawdzać na bieżąco, czy przydzielanie pasma dla rozważanego źródła jest zasadne. Jeśli w chwili wyznaczania szybkości $t = k_{j\beta}T$ węzeł wąskiego gardła stwierdzi, że j-te źródło do chwili $k_{j\beta}T + T_{bj}$ wyśle wszystkie dane, które chce nadać, to uzna że źródło to należy odłączyć. W takim wypadku od chwili $k_{j\beta}T$ pasmo będzie rozdzielane pomiędzy pozostałe źródła, czyli inaczej mówiąc źródło o indeksie j nie będzie już brane pod uwagę podczas podziału dostępnego pasma.

W dalszej części artykułu zaproponowane zostaną dwa nowe algorytmy regulacji szybkości nadawania danych w opisanej sieci i udowodnione zostaną istotne właściwości przedstawionych rozwiązań. Prezentowane algorytmy mogą działać także wtedy, gdy liczba źródeł w sieci nie jest stała. Jest to ważna zaleta tych algorytmów, w porównaniu z podobnymi rozwiązaniami wcześniej proponowanymi w literaturze [1, 2, 4, 14, 15], ponieważ w praktyce liczba nadajników uczestniczących w przesyłaniu informacji zmienia się i nie może być z góry określona.

3. DYSKRETNY REGULATOR PROPORCJONALNY

Szybkość nadawania danych wyznaczana jest w dyskretnych chwilach czasu $t = kT$, gdzie $k = 0, 1, 2, \ldots$, w następujący sposób:

a) dla czasu $t \in [0, k_0T)$, gdzie k_0 jest liczbą całkowitą większą od lub równą zeru, taką że k_0T jest pierwszą chwilą, w której spełniony jest warunek

$$K[x_d - x(k_0T) - A(k_0T)] \leq d_{\max} \tag{3}$$

szybkość wyznaczana przez algorytm jest równa $d_{\max}$;

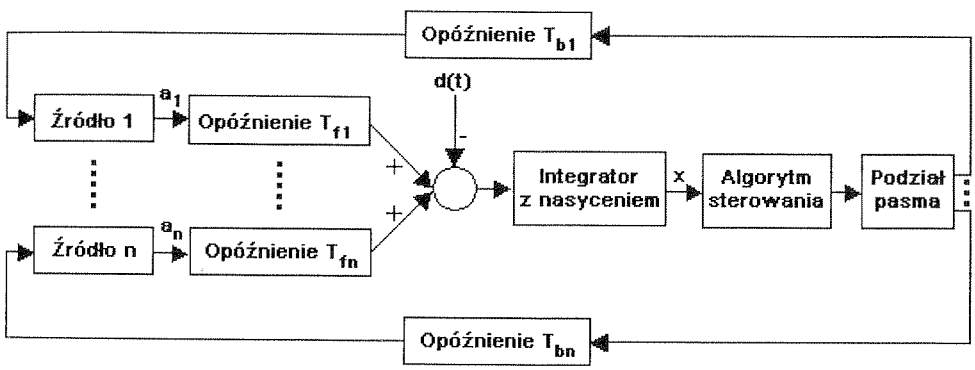
b) dla czasu $t \geq k_0T$ szybkość wyznaczana jest z równania

$$a(kT) = K[x_d - x(kT) - A(kT)] \tag{4}$$

przy czym współczynnik wzmocnienia $K \in (0, T^{-1})$.

Z nierówności (3) wynika, że wartość k_0 zależy między innymi od wykorzystywanej części dostępnego pasma. Ponieważ nie jest ono z góry znane, więc także wartość k_0 nie jest ustalona a priori, lecz wyznaczana *on-line* w węźle wąskiego gardła, który dysponując wszystkimi potrzebnymi informacjami sprawdza warunek (3). Może się tak zdarzyć, że warunek (3) nie zostanie nigdy spełniony. W takim wypadku szybkość nadawania danych wyznaczana przez algorytm jest zawsze równa $d_{\max}$.

Schemat sieci wykorzystującej przedstawiony algorytm został pokazany na rysunku 1.



Rys. 1. Układ regulacji

Fig. 1. Control system

W dalszej części rozdziału właściwości zaproponowanego algorytmu zostaną sformułowane w postaci czterech twierdzeń. Pierwsze dwa twierdzenia pokazują, że generowana przez algorytm szybkość nadawania danych jest ograniczona. Z kolei twierdzenie 3 podaje warunek pełnego wykorzystania pasma, a twierdzenie 4 pokazuje, że stosując zaproponowany algorytm całkowicie eliminuje się gubienie jednostek danych spowodowane przepełnieniem bufora wąskiego gardła.

Twierdzenie 1

Generowana przez algorytm szybkość nadawania danych jest zawsze nieujemna.

Dowód

Z definicji algorytmu bezpośrednio wynika, że dla $t \in [0, k_0T)$ generowana szybkość nadawania danych jest nieujemna. Na początku pokażemy, że szybkość wygenerowana w chwili k_0T jest nieujemna. Ponieważ ta szybkość wyznaczana jest na podstawie zależności (4), więc

$$\begin{aligned} a(k_0T) &= K\,[x_d - x(k_0T) - A(k_0T)] = \\ &= Kx_d - K\left[x(k_0T - T) + B(k_0T - T, T) - \int_{k_0T-T}^{k_0T} h(\tau)\,d\tau\right] + \\ &- K\left[A(k_0T - T) + B(k_0T - T, T) + \int_{k_0T-T}^{k_0T} a(\tau)\,d\tau\right] = \\ &= K\,[x_d - x(k_0T - T) + A(k_0T - T, T)] + K\int_{k_0T-T}^{k_0T} h(\tau)\,d\tau - KTd_{\max} \end{aligned} \tag{5}$$

Ponieważ k_0 jest najmniejszą liczbą całkowitą, dla której spełniony jest warunek (3), więc

$$K[x_d - x(k_0T - T) - A(k_0T - T)] > d_{\max} \tag{6}$$

Wykorzystując nierówność (6) i zależność (5) otrzymujemy

$$a(k_0T) > d_{\max} + K\int_{k_0T-T}^{k_0T} h(\tau)\,d\tau - KTd_{\max} \geqslant (1 - KT)\,d_{\max} \geq 0 \tag{7}$$

W ten sposób pokazaliśmy, że $a(k_0T) \geq 0$. Weźmy teraz pod uwagę dowolną chwilę $kT \geq k_0T$, gdzie $k = 0, 1, 2, \ldots$. Niech $a(kT) \geq 0$. Pokażemy, że również $a(kT + T) \geq 0$. Wartość szybkości wygenerowanej w chwili kT zapisujemy z definicji algorytmu jako

$$a(t) = a(kT) = K[x_d - x(kT) - A(kT)] \tag{8}$$

Korzystając z założeń (1) i (4) szybkość generowaną w chwili $kT + T$ możemy zapisać

$$\begin{aligned} a(kT + T) &= \\ &= K\left\{x_d - x(kT + T) - \left[A(kT) - B(kT, T) + \sum_{j=1}^{n(kT)} \beta_j(kT) \int_{kT}^{kT+T} a(\tau)\,d\tau\right]\right\} \\ &= K\,\{x_d - x(kT + T) - [A(kT) - B(kT, T) + Ta(kT)]\} \end{aligned} \tag{9}$$

Długość kolejki danych w buforze wąskiego gardła w chwili $kT + T$ można zapisać jako

$$x(kT + T) = x(kT) + B(kT, T) - \int_{kT}^{kT+T} h(\tau)\, d\tau \tag{10}$$

Z zależności (9) i (10) otrzymujemy

$$\begin{aligned} a(kT + T) &= \\ &= K\{x_d - x(kT) - B(kT, T) + \int_{kT}^{kT+T} h(\tau)\, d\tau - [A(kT) - B(kT, T) + T_a(kT)]\} = \\ &= K[x_d - x(kT) - A(kT) + \int_{kT}^{kT+T} h(\tau)\, d\tau - Ta(kT)] \end{aligned} \tag{11}$$

Wykorzystując równanie (8), zależność (11) zapisujemy w następującej postaci

$$\begin{aligned} a(kT + T) &= K[x_d - x(kT) - A(kT)] + K[\int_{kT}^{kT+T} h(\tau)\, d\tau - Ta(kT)] = \\ &= a(kT) + K \int_{kT}^{kT+T} h(\tau)\, d\tau - KTa(kT) \end{aligned} \tag{12}$$

a zatem — biorąc pod uwagę, że $a(kT) \geq 0$ oraz uwzględniając nierówność (2) — otrzymujemy

$$a(kT + T) = (1 - KT)a(kT) + K \int_{kT}^{kT+T} h(\tau)\, d\tau \geq 0 \tag{13}$$

co kończy dowód twierdzenia.

Twierdzenie 2

Zaproponowany algorytm nie generuje szybkości nadawania danych większych niż $d_{\max}$.

Dowód

Z warunków (3) i (4) wynika, że szybkość nadawania danych generowana dla czasu $t \in [0, k_0 T)$ wynosi $d_{\max}$, a w chwili $t = k_0 T$ jest nie większa niż $d_{\max}$. Niech w pewnej chwili kT, gdzie $k = 0, 1, 2, \ldots$ oraz $k \geq k_0$, $a(kT) \leq d_{\max}$. Pokażemy, że w tej sytuacji także $a(kT + T) \leq d_{\max}$.

Korzystając z zależności (13) otrzymujemy

$$a(kT+T) = (1-KT)a(kT) + K\int_{kT}^{kT+T} h(\tau)\,d\tau \leq \leq (1-KT)d_{\max} + KTd_{\max} = d_{\max} \tag{14}$$

co dowodzi prawdziwości twierdzenia 2. Udowodnione twierdzenia pokazują, że generowana przez algorytm szybkość nadawania danych jest zawsze nieujemna i ograniczona z góry, co stanowi warunek konieczny praktycznej implementacji każdej strategii sterowania szybkością nadawania danych w sieci. Kolejne twierdzenie podaje warunki pełnego wykorzystania pasma i dlatego jest istotne z punktu widzenia racjonalnej eksploatacji sieci.

Niech $RTT_{\max}$ będzie największą całkowitą wielokrotnością T spełniającą warunek

$$RTT_{\max} < \frac{x_d}{d_{\max}} - \frac{1}{K} \tag{15}$$

Twierdzenie 3

Jeżeli czas RTT pełnego obiegu jednostki sterującej żadnego źródła uczestniczącego w wymianie informacji nie jest większy niż $RTT_{\max}$, to dla $t \geq k_0 T$ długość kolejki danych w buforze wąskiego gardła jest zawsze ściśle dodatnia.

Dowód

Niech, zgodnie z założeniami twierdzenia, dla każdego j

$$RTT_j \leq RTT_{\max} \tag{16}$$

Długość kolejki danych w buforze wąskiego gardła w dowolnej chwili t wyraża się zależnością

$$x(t) = x(kT) + B(kT,\delta) - \int_{kT}^{kT+\delta} h(\tau)\,d\tau \tag{17}$$

gdzie k jest największą liczbą całkowitą, taką że $kT \leq t$, a δ jest liczbą rzeczywistą z przedziału $[0, T)$, taką że $t = kT + \delta$. Niech $t \geq k_0 T$. Z twierdzenia 2 wiadomo, że proponowany algorytm nie generuje szybkości nadawania większych niż $d_{\max}$. Korzystając z tej właściwości mamy

$$x(kT) \geq x_d - A(kT) - K^{-1} d_{\max} \tag{18}$$

gdzie $k \geq k_0$. Z zależności (17) i (18) otrzymujemy

$$x(t) \geq x_d - A(kT) - K^{-1} d_{\max} + B(kT,\delta) - \int_{kT}^{kT+\delta} h(\tau)\,d\tau \tag{19}$$

Wprowadzamy funkcję $C(t) = A(kT) - B(kT, \delta)$, która określa ilość danych „w locie" w chwili kT pomniejszoną o ilość danych, które wpłynęły do bufora wąskiego gardła w czasie $t \in [kT, kT + \delta)$. Wyrażenie $C(t)$ możemy zapisać w postaci

$$C(t) = \sum_{i=1}^{p} C_i(t) \tag{20}$$

gdzie p jest taką liczbą całkowitą, że $pT = RTT_{\max}$, natomiast $C_i(t)$ oznacza ilość danych „w locie" w przedziale czasu $t \in [kT - iT,\ kT - (i-1)T)$. Wartość $C_i(t)$ określa wyrażenie

$$C_i(t) = \sum_{j \in Q_i(kT-iT)} \beta_j(kT - iT) \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau + \sum_{j \in R_i(kT-iT)} \beta_j(kT - iT) \int_{kT-iT+\delta}^{kT-(i-1)T} a(\tau)d\tau \tag{21}$$

gdzie:

- $Q_i(kT–iT)$ — zbiór źródeł, dla których $RTT > iT$ i które były brane pod uwagę podczas podziału pasma w chwili $kT–iT$;
- $R_i(kT–iT)$ — zbiór źródeł, dla których $RTT = iT$ i które były brane pod uwagę podczas podziału pasma w chwili $kT–iT$.

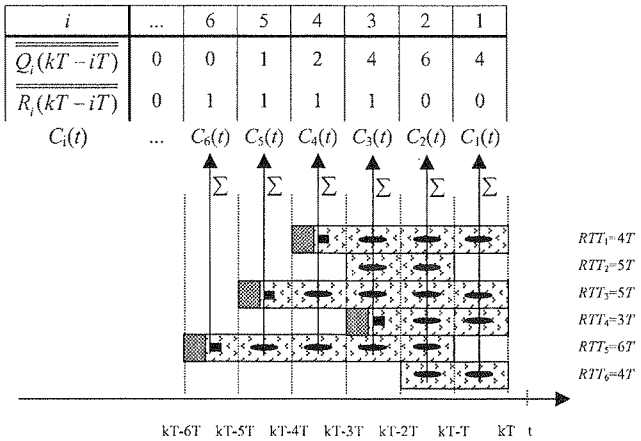
Wprowadzone definicje funkcji $C_i(t)$ oraz $C(t)$ ilustruje rysunek 2. Ma on charakter poglądowy i przedstawia graficzną reprezentację wartości wyrażeń $C_i(t)$ w przypadku, gdy w wymianie informacji w czasie $t \in [kT - RTT_{\max}, kT)$ uczestniczy, lub uczestniczyło, sześć nadajników. Rysunek odzwierciedla rozłożenie danych „w locie" w czasie, lecz nie pokazuje dokładnej ilości tych danych. W tabeli zaznaczono liczność zbiorów Q_i oraz R_i w poszczególnych przedziałach czasu. Rysunek ilustruje przykładową sytuację, w której:

- Źródło o numerze 2 było brane pod uwagę podczas podziału pasma w czasie $t \in [kT - 3T,\ kT - T)$. Oznacza to, że w chwili $kT–T$ źródło zostało odłączone od układu; dane z tego źródła nie wpływają do bufora w czasie $t \in [kT,\ kT + \delta)$.
- Źródło o numerze 6 zostało dołączone w chwili $t = kT–2T$. Dane z tego źródła nie wpływają do bufora w czasie $t \in [kT,\ kT + \delta)$.
- W czasie $t \in [kT,\ kT + \delta)$ do bufora wąskiego gardła wpływają dane ze źródeł 1, 3, 4 i 5.

Zauważmy, że z definicji zbiorów Q_i i R_i wynikają zależności

$$\bigvee_{i=1,..,p} \overline{\overline{Q_i(kT - iT)}} + \overline{\overline{R_i(kT - iT)}} \le n(kT - iT) \tag{22}$$

$$Q_i(kT - iT) \cap R_i(kT - iT) = \emptyset \tag{23}$$



różnica ilość danych „w locie" w chwili kT i tych danych, które wpłynęły do bufora w czasie $t \in [kT, kT+\delta)$;

ilość danych, które wpłynęły do bufora wąskiego gardła w czasie $t \in [kT, kT+\delta)$;

źródło, dla którego $RTT > iT$ i które było brane pod uwagę podczas podziału pasma w chwili $kT - iT$;

źródło, dla którego $RTT = iT$ i które było brane pod uwagę podczas podziału pasma w chwili $kT - iT$.

Rys. 2. Graficzna reprezentacja wyrażeń $C(t)$ oraz $C_i(t)$

Fig. 2. Graphical representation of expressions $C(t)$ and $C_i(t)$

Dla $i = 1, \ldots, p-1$ dokonujemy oszacowania

$$C_i(t) \le \sum_{j \in Q_i(kT-iT)} \beta_j(kT-iT) \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau + \sum_{j \in R_i(kT-iT)} \beta_j(kT-iT) \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau =$$

$$= \sum_{j \in Q_i(kT-iT) \cup R_i(kT-iT)} \beta_j(kT-iT) \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau \tag{24}$$

Wykorzystując równanie (1) i nierówność (22) otrzymujemy

$$C_i(t) \le \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau \tag{25}$$

Zauważmy dalej, że gdy $i = p$, to $\overline{\overline{Q_1(kT-iT)}} = 0$, dla każdej chwili czasu kT. Zatem dla $i = p$

$$C_i(t) = \sum_{j \in R_i(kT-iT)} \beta_j(kT-iT) \int_{kT-iT+\delta}^{kT-(i-1)T} a(\tau)d\tau \le \int_{kT-iT+\delta}^{kT-(i-1)T} a(\tau)d\tau \tag{26}$$

Wykorzystując zależności (25) i (26) możemy oszacować wartość $C(t)$ w następujący sposób

$$C(t) = \sum_{i=1}^{p} C_i(t) \leq \sum_{i=1}^{p-1} \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau + \int_{kT-pT+\delta}^{kT-(p-1)T} a(\tau)d\tau =$$
$$= \int_{kT-pT+\delta}^{kT} a(\tau)d\tau \leqslant (pT - \delta)d_{\max} = (RTT_{\max} - \delta)d_{\max} \tag{27}$$

Biorąc pod uwagę nierówność (19) i zależność (27) możemy napisać

$$x(t) \geq x_d - [A(kT) - B(kT, \delta)] - K^{-1}d_{\max} - \int_{kT}^{kT+\delta} h(\tau)d\tau \geq$$
$$\geq x_d - (RTT_{\max} - \delta)d_{\max} - K^{-1}d_{\max} - \delta d_{\max} = x_d - (RTT_{\max} + K^{-1})d_{\max} \tag{28}$$

Uwzględniając warunek (15), który można przedstawić w następującej postaci

$$x_d > \left(RTT_{\max} + K^{-1}\right)d_{\max} \tag{29}$$

z nierówności (28) wnioskujemy, że $x(t) > 0$, co kończy dowód twierdzenia.

Kolejne twierdzenie pokazuje, że zaproponowana strategia sterowania eliminuje niebezpieczeństwo zagubienia informacji na skutek przepełnienia bufora wąskiego gardła sieci. Ta właściwość jest szczególnie istotna w przypadku gdy przesyłane dane wykorzystywane są przez aplikacje wrażliwe na utratę danych.

Twierdzenie 4

Zastosowanie proponowanego algorytmu w opisanej sieci zapewnia, że długość kolejki danych w buforze wąskiego gardła nigdy nie przekroczy wartości sygnału odniesienia xd, czyli

$$\forall_{t \geqslant 0} \; x(t) \leq x_d \tag{30}$$

Dowód

W pierwszej kolejności pokażemy, że twierdzenie jest prawdziwe dla $t \in [0, k_0T)$. Niech $t \in [0, k_0T)$ i niech k będzie największą liczbą całkowitą, taką że $kT \leq t$, a δ liczbą rzeczywistą z przedziału $[0, T)$, taką że $t = kT + \delta$. Z definicji algorytmu wynika, że dla chwili $kT \in [0, k_0T)$ warunek (3) nie jest spełniony, a zatem

$$K[x_d - x(kT) - A(kT)] > d_{\max} \tag{31}$$

Przekształcając nierówność (31) otrzymujemy

$$x(kT) < x_d - d_{\max}K^{-1} - A(kT) \tag{32}$$

Wykorzystując zależności (17) oraz (32) mamy

$$x(t) < x_d - d_{\max}K^{-1} - A(kT) + B(kT,\delta) - \int_{kT}^{kT+\delta} h(\tau)d\tau \tag{33}$$

Ponieważ $A(kT)$ oznacza ilość danych „w locie" w chwili kT, natomiast $B(kT,\delta)$ część tych danych, które wpłynęły do bufora wąskiego gardła w czasie $t \in [kT, kT+\delta)$ i $\delta \in [0,T)$ więc $B(kT,\delta) \leq A(kT)$ i z nierówności (33) mamy

$$x(t) < x_d - d_{\max}K^{-1} - \int_{kT}^{kT+\delta} h(\tau)d\tau < x_d \tag{34}$$

Zatem z nierówności (34) wynika, że twierdzenie jest prawdziwe dla $t \in [0, k_0T)$.

Niech teraz $t \geq k_0T$. Wartość czasu t możemy zapisać jako $t = kT + \delta$, gdzie $k \geq k_0$ i $\delta \in [0,T)$. Z twierdzenia 1 wiadomo, że

$$a(kT) = K[x_d - x(kT) - A(kT)] \geq 0 \tag{35}$$

Z tej zależności otrzymujemy

$$x(kT) \leq x_d - A(kT) \tag{36}$$

Wykorzystując nierówność (36) w równaniu (17) otrzymujemy

$$x(t) \leq x_d - [A(kT) - B(kT,\delta)] - \int_{kT}^{kT+\delta} h(\tau)d\tau = x_d - C(t) - \int_{kT}^{kT+\delta} h(\tau)d\tau \tag{37}$$

Z definicji wyrażeń $C_i(t)$ wynika, że są one nieujemne, tak więc $C(t)$ jako suma nieujemnych składników również jest nieujemna, a zatem z zależności (37) wynika, że

$$x(t) \leqslant x_d - C(t) - \int_{kT}^{kT+\delta} h(\tau)d\tau \leq x_d \tag{38}$$

Stąd bezpośrednio wynika teza twierdzenia, mówiąca że długość kolejki danych w buforze wąskiego gardła nigdy nie przekroczy wartości x_d.

4. DYSKRETNY REGULATOR PROPORCJONALNY ZE SPRZĘŻENIEM BEZPOŚREDNIM OD DOSTĘPNEGO PASMA

W tym rozdziale proponujemy zmodyfikowany, nieliniowy regulator prędkości nadawania danych, który wykorzystuje nie tylko sprzężenie zwrotne od długości kolejki

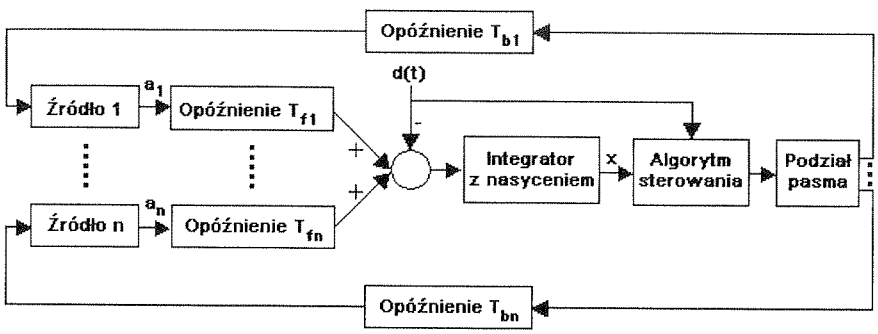
danych w buforze wąskiego gardła, ale także kompensację zmian dostępnego pasma w torze głównym układu sterowania. Algorytm wyznacza szybkość nadawania danych w dyskretnych chwilach $t = kT$, gdzie $k = 0, 1, 2, \ldots$. Niech funkcja

$$W(kT) = K\left[x_d - x(kT) - A(kT) + \varepsilon d(kT)\sum_{j=1}^{n(kT)} \beta_j(kT) RTT_j\right] \tag{39}$$

gdzie $K > 0$ jest współczynnikiem wzmocnienia. Stała $\varepsilon \geq 0$ określa wpływ sprzężenia bezpośredniego od dostępnego pasma na wyznaczaną szybkość nadawania danych. Wartość szybkości w chwilach kT wyznaczana jest na podstawie zależności

$$a(t) = a(kT) = \begin{cases} 0 & gdy \quad W(kT) < 0 \\ W(kT) & gdy \quad 0 \leq W(kT) \leq a_{\max} \\ a_{\max} & gdy \quad W(kT) > a_{\max} \end{cases} \tag{40}$$

Symbolem $a_{\max}$ oznaczono maksymalną dopuszczalną szybkość generowaną przez algorytm. Z zależności (40) bezpośrednio wynika, że szybkość nadawania danych generowana przez algorytm jest zawsze nieujemna i ograniczona z góry. Schemat sieci wykorzystującej tego typu regulację został pokazany na rysunku 3.



Rys. 3. Układ regulacji ze sprzężeniem bezpośrednim od dostępnego pasma

Fig. 3. Control system with feed-forward compensation

Zostaną teraz przedstawione dwa twierdzenia, które pokazują istotne właściwości układu. Pierwsze z nich określa warunki jakie muszą być spełnione, aby wyeliminować niebezpieczeństwo utraty danych na skutek przepełnienia bufora wąskiego gardła, natomiast drugie podaje warunek pełnego wykorzystania łącza.

Twierdzenie 5

Niech RTT_s będzie dodatnią liczbą rzeczywistą. Jeśli dla dowolnego $k = 0, 1, 2, \ldots$ spełniony jest warunek

$$\sum_{j=1}^{n(kT)} \beta_j(kT) RTT_j \leqslant RTT_s \tag{41}$$

to dla każdej chwili t długość kolejki danych w buforze wąskiego gardła spełnia nierówność

$$x(t) < x_d + \varepsilon d_{\max} RTT_s + Ta_{\max} \tag{42}$$

Warto zauważyć, że ani liczba połączeń obsługiwanych przez węzeł, ani czasy RTT tych połączeń nie są z góry znane. Dlatego, za przestrzeganie warunku (41) odpowiadają odpowiednie procedury decydujące o przyjęciu lub odrzuceniu połączenia w trakcie procesu włączania nowego nadajnika do układu.

Dowód

Weźmy pod uwagę dowolną chwilę czasu $t \geq 0$. Niech k będzie największą liczbą całkowitą, większą od lub równą zeru, taką że $kT \leq t$. Można zatem zapisać $t = kT + \delta$, gdzie δ jest liczbą rzeczywistą z przedziału $[0, T)$. Rozważać będziemy dwa przypadki: pierwszy gdy $W(kT) < 0$ i drugi gdy $W(kT) \geq 0$.

Przypadek 1: Analizujemy sytuację gdy $W(kT) < 0$. Znajdźmy chwilę $k_1 T < t$, w której po raz ostatni funkcja W przyjmowała wartość dodatnią. Zauważmy, że

$$W(0) = K\left[x_d + \varepsilon d(0) \sum_{j=1}^{n(0)} \beta_j(0) RTT_j\right] > 0 \tag{43}$$

a zatem chwila $k_1 T$ istnieje. Ponadto prawdziwa jest zależność $k_1 T < kT \leq t$. W chwili $k_1 T$

$$W(k_1 T) = K\left[x_d - x(k_1 T) - A(k_1 T) + \varepsilon d(k_1 T) \sum_{j=1}^{n(k_1 T)} \beta_j(k_1 T) RTT_j\right] > 0 \tag{44}$$

Przekształcając powyższe wyrażenie otrzymujemy

$$x(k_1 T) < x_d - A(k_1 T) + \varepsilon d(k_1 T) \sum_{j=1}^{n(k_1 T)} \beta_j(k_1 T) RTT_j \tag{45}$$

Z kolei długość kolejki w buforze wąskiego gardła w chwili t możemy wyrazić następująco

$$\begin{aligned} x(t) &= x(k_1 T) + B(k_1 T, t - k_1 T) - \int_{k_1 T}^{t} h(\tau) d\tau < \\ &< x_d - A(k_1 T) + \varepsilon d(k_1 T) \sum_{j=1}^{n(k_1 T)} \beta_j(k_1 T) RTT_j + B(k_1 T, t - k_1 T) - \int_{k_1 T}^{t} h(\tau) d\tau \leqslant \\ &\leqslant x_d + \varepsilon d(k_1 T) \sum_{j=1}^{n(k_1 T)} \beta_j(k_1 T) RTT_j + [B(k_1 T, t - k_1 T) - A(k_1 T)] \end{aligned} \tag{46}$$

W wyrażeniu (46) wartość składnika $B(k_1T, t-k_1T)-A(k_1T)$ określa różnicę ilości danych, które wpłynęły do bufora wąskiego gardła w przedziale czasu od k_1T do t i danych „w locie" w chwili k_1T. Poprzez dane „w locie" rozumiemy wszystkie dane, dla których wyznaczona została szybkość z jaką zostały lub zostaną nadane i które nie dotarły jeszcze do bufora wąskiego gardła. Zauważmy, że ilość danych, które wpłyną do bufora w czasie od k_1T do t nie może być większa niż suma ilości danych „w locie" w chwili k_1T i ilości tych danych, które zostaną nadane w czasie od k_1T do t. Ponieważ w chwili k_1T algorytm po raz ostatni wyznaczył niezerową szybkość, więc w czasie od $k_1T + T$ do t generowana szybkość jest równa zeru. A zatem

$$B(k_1T, t-k_1T) - A(k_1T) \leqslant \int_{k_1T}^{t} a(\tau)d\tau = \int_{k_1T}^{k_1T+T} a(\tau)d\tau \leqslant Ta_{\max} \tag{47}$$

Wykorzystując, w nierówności (46), zależność (47) otrzymujemy

$$x(t) < x_d + \varepsilon d(k_1T) \sum_{j=1}^{n(k_1T)} \beta_j(k_1T)RTT_j + Ta_{\max} \leqslant x_d + \varepsilon d_{\max}RTT_s + Ta_{\max} \tag{48}$$

co kończy pierwszą część dowodu.
Przypadek 2: Analizujemy teraz sytuację gdy $W(kT) \geq 0$, a zatem możemy zapisać

$$W(kT) = K\left[x_d - x(kT) - A(kT) + \varepsilon d(kT) \sum_{j=1}^{n(kT)} \beta_j(kT)RTT_j\right] \geqslant 0 \tag{49}$$

Stąd otrzymujemy

$$x(kT) \leq x_d - A(kT) + \varepsilon d(kT) \sum_{j=1}^{n(kT)} \beta_j(kT)RTT_j \tag{50}$$

Długość kolejki w buforze wąskiego gardła w chwili $t = kT + \delta$

$$\begin{aligned} x(t) &= x(kT) + B(kT, \delta) - \int_{kT}^{kT+\delta} h(\tau)d\tau \leq \\ &\leq x_d - A(kT) + \varepsilon d(kT) \sum_{j=1}^{n(kT)} \beta_j(kT)RTT_j + B(kT, \delta) - \int_{kT}^{kT+\delta} h(\tau)d\tau \end{aligned} \tag{51}$$

Ponieważ dla każdego połączenia czas RTT jest nie mniejszy niż T i δ należy do przedziału $[0, T)$, więc ilość danych, które wpłyną do bufora wąskiego gardła w czasie

od kT do $kT+\delta$ jest mniejsza niż ilość danych „w locie" określona w chwili kT. Zatem $A(kT) - B(kT,\ \delta) > 0$ i z zależności (51) otrzymujemy

$$x(t) < x_d + \varepsilon d_{\max} RTT_s - \int_{kT}^{kT+\delta} h(\tau)d\tau \leq x_d + \varepsilon d_{\max} RTT_s < \\ < x_d + \varepsilon d_{\max} RTT_s + Ta_{\max} \tag{52}$$

co kończy dowód twierdzenia.

Udowodnione twierdzenie pokazuje, że jeżeli istnieją procedury kontroli dołączanych źródeł gwarantujące spełnienie w każdej chwili warunku (41), to dane nigdy nie będą tracone. Drugą pożądaną cechą rozważanego układu sterowania, jest zapewnienie możliwie jak najpełniejszego wykorzystania łącza. Kolejne twierdzenie podaje warunek, którego spełnienie gwarantuje stuprocentowe wykorzystanie dostępnego pasma.

Twierdzenie 6

Jeżeli maksymalna szybkość nadawania danych $a_{\max} > d_{\max}$ i wartość sygnału zadanego spełnia nierówność

$$x_d > \left(RTT_{\max} + K^{-1}\right) a_{\max} + Td_{\max} \tag{53}$$

to istnieje taka chwila

$$k_2 T \in \left[\frac{x_d - K^{-1}a_{\max}}{a_{\max}}, \frac{x_d - d_{\max}RTT_{\min} - K^{-1}a_{\max}}{a_{\max} - d_{\max}}\right] \tag{54}$$

gdzie k_2 jest dodatnią liczbą całkowitą, że długość kolejki danych w buforze wąskiego gardła dla $t \geq k_2 T$ jest zawsze większa od zera. Symbolami $RTT_{\min}$ i $RTT_{\max}$ oznaczono odpowiednio najmniejszą i największą dopuszczalną wartość RTT.

Dowód

Definiujemy funkcję czasu φ w następujący sposób

$$\varphi(kT) = x(kT) + A(kT) \tag{55}$$

Wartość funkcji φ stanowi sumę ilości danych „w locie" oraz tych, które tworzą kolejkę w buforze wąskiego gardła. Tę funkcję można przedstawić następująco

$$\varphi(kT) = x(kT) + A(kT) = x(0) + B(0, kT) - \int_0^{kT} h(\tau)d\tau + A(kT) \tag{56}$$

Zauważmy, że składnik $B(0,kT) + A(kT)$ określa ilość danych, dla których od chwili 0 do kT wyznaczona została szybkość z jaką zostały lub zostaną nadane. A zatem $B(0,kT) + A(kT) = \int_0^{kT} a(\tau)d\tau$ i z równania (56) mamy

$$\varphi(kT) = \int_0^{kT} a(\tau)d\tau - \int_0^{kT} h(\tau)d\tau \tag{57}$$

Dla $k = 0$

$$\varphi(kT) = \varphi(0) = 0 < x_d - K^{-1}a_{\max} \tag{58}$$

Z kolei jeżeli dla dowolnego k spełniona jest nierówność $\varphi(kT) < x_d - K^{-1}a_{\max}$, to

$$\begin{aligned} W(kT) &= K\left[x_d - \varphi(kT) + \varepsilon d(kT)\sum_{j=1}^{n(kT)} \beta_j(kT)RTT_j\right] > \\ &> K\left[K^{-1}a_{\max} + \varepsilon d(kT)\sum_{j=1}^{n(kT)} \beta_j(kT)RTT_j\right] \geqslant a_{\max} \end{aligned} \tag{59}$$

Z zależności (40) wynika, że w tej sytuacji $a(kT) = a_{\max}$. Wobec tego, uwzględniając założenie $a_{\max} > d_{\max}$ i zależność (57), można stwierdzić, że wartość funkcji φ rośnie co najmniej w tempie równym różnicy $a_{\max} - d_{\max}$. Zatem istnieje taka skończona chwila k_2T, w której po raz pierwszy zostanie spełniony warunek

$$\varphi(kT) \geq x_d - K^{-1}a_{\max} \tag{60}$$

Zostanie teraz określony przedział czasu, do którego należy chwila k_2T. W tym celu należy sprawdzić kiedy po raz pierwszy warunek (60) zostanie spełniony. Korzystając z zależności (57) można zauważyć, że dla chwili k_2T warunek (60) będzie spełniony, gdy

$$\int_0^{k_2T} a(\tau)d\tau - \int_0^{k_2T} h(\tau)d\tau \geq x_d - K^{-1}a_{\max} \tag{61}$$

Biorąc pod uwagę to, że szybkość wyznaczana dla $t < k_2T$ była równa $a_{\max}$, z nierówności (61) otrzymuje się

$$k_2Ta_{\max} - \int_0^{k_2T} h(\tau)d\tau \geq x_d - K^{-1}a_{\max} \tag{62}$$

Najwcześniej, powyższy warunek może zostać spełniony, gdy wykorzystane pasmo w przedziale od 0 do k_2T jest równe zeru. Stąd wynika lewy kraniec przedziału (54). Z

kolei warunek ten zostanie spełniony najpóźniej wtedy, gdy wykorzystane pasmo będzie największe. Biorąc pod uwagę zależność (2) i uwzględniając to, że dla $t < RTT_{\min}$ wykorzystane pasmo $h(t) = 0$, można oszacować kiedy najpóźniej zostanie spełniony warunek (60), a tym samym wyznaczyć prawy kraniec przedziału (54).

Pokażemy teraz, że dla każdej chwili $kT > k_2T$ warunek (60) jest spełniony. W tym celu bierzemy pod uwagę dowolną chwilę $kT > k_2T$ i rozważamy dwa przypadki: pierwszy gdy $W(kT) < a_{\max}$ i drugi gdy $W(kT) \geq a_{\max}$. W pierwszym przypadku z zależności (39) i (55) otrzymuje się następującą nierówność

$$W(kT) = K\left[x_d - \varphi(kT) + \varepsilon d(kT)\sum_{j=1}^{n(kT)} \beta_j(kT) RTT_j\right] < a_{\max} \tag{63}$$

z której wynika

$$\varphi(kT) > x_d + \varepsilon d(kT)\sum_{j=1}^{n(kT)} \beta_j(kT) RTT_j - K^{-1}a_{\max} \geq x_d - K^{-1}a_{\max} \tag{64}$$

co oznacza, że w rozważanym przypadku, dla $kT > k_2T$, warunek (60) jest rzeczywiście spełniony.

Rozważmy teraz przypadek drugi gdy $W(kT) \geq a_{\max}$. W tej sytuacji, aby pokazać, że warunek (60) jest spełniony dla każdego $kT > k_2T$, można zastosować zasadę indukcji matematycznej. Z wcześniejszych rozważań wiadomo, że istnieje chwila k_2T, dla której nierówność (60) jest spełniona. Przyjmiemy teraz, że dla pewnej chwili $kT \geq k_2T$ analizowany warunek jest prawdziwy i pokażemy, że z tego wynika również prawdziwość tego warunku dla chwili $kT + T$. Biorąc pod uwagę równanie (55) można napisać

$$\begin{aligned} &\varphi(kT + T) = x(kT + T) + A(kT + T) = \\ &= x(kT) + A(kT) - \int_{kT}^{kT+T} h(\tau)d\tau + \int_{kT}^{kT+T} a(\tau)d\tau \geq \\ &\geqslant x_d - K^{-1}a_{\max} - Td_{\max} + Ta(kT) \end{aligned} \tag{65}$$

Ponieważ $W(kT) \geq a_{\max}$, więc $a(kT) = a_{\max}$. Zatem, uwzględniając nierówność $a_{\max} > d_{\max}$, otrzymuje się

$$\varphi(kT + T) \geqslant x_d - K^{-1}a_{\max} - Td_{\max} + Ta_{\max} > x_d - K^{-1}a_{\max} \tag{66}$$

Z nierówności (64) i (66) wynika, że dla $kT > k_2T$, warunek (60) jest zawsze spełniony.

W dalszej części dowodu wykażemy, że jeśli spełniona jest nierówność (60), to dla dowolnej chwili $t \geq k_2T$ długość kolejki w buforze wąskiego gardła jest zawsze większa od zera.

Niech $t = kT + \delta \geq k_2T$, gdzie $\delta \in [0, T)$. Z zależności (55) i (60) wynika, że

$$x(kT) = \varphi(kT) - A(kT) \geq x_d - K^{-1}a_{\max} - A(kT) \tag{67}$$

Długość kolejki danych w buforze wąskiego gardła w chwili $t = kT + \delta$ wynosi

$$x(t) = x(kT) + B(kT, \delta) - \int_{kT}^{kT+\delta} h(\tau)d\tau \geq$$
$$\geq x_d - K^{-1}a_{\max} - A(kT) - \int_{kT}^{kT+\delta} h(\tau)d\tau \tag{68}$$

Dokonamy teraz oszacowania wartości składnika $A(kT)$. W tym celu zapisujemy wartość $A(kT)$ w postaci sumy

$$A(kT) = \sum_{i=1}^{p} \sum_{j \in Q_i(kT-iT)} \beta_j(kT - iT) \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau \tag{69}$$

gdzie p jest liczbą całkowitą, taką że $pT = RTT_{\max}$, natomiast $Q_i(kT-iT)$ jest zbiorem źródeł, dla których czas RTT jest większy od lub równy iT i które były brane pod uwagę podczas podziału pasma w chwili $kT-iT$. Zauważmy, że $\forall_{i=1,\ldots,p} \overline{Q_i(kT - iT)} \leqslant n(kT - iT)$. A zatem możemy dokonać oszacowania

$$A(kT) \leq \sum_{i=1}^{p} \int_{kT-iT}^{kT-(i-1)T} a(\tau)d\tau = \int_{kT-pT}^{kT} a(\tau)d\tau \leq pTa_{\max} = RTT_{\max}a_{\max} \tag{70}$$

Wykorzystując, w równaniu (68), zależność (70) otrzymujemy

$$x(t) \geq x_d - (RTT_{\max} + K^{-1})a_{\max} - Td_{\max} \tag{71}$$

Z powyższej zależności bezpośrednio wynika, że jeżeli spełniona jest nierówność (53) to długość kolejki danych w buforze wąskiego gardła w dowolnej chwili $t \geq k_2T$ jest zawsze większa od zera, co kończy dowód twierdzenia.

Udowodnione twierdzenie jest istotne ponieważ pokazuje jaki warunek musi być spełniony, aby zagwarantować pełne wykorzystanie dostępnego pasma.

5. PRZYKŁAD SYMULACYJNY

Właściwości algorytmów zaproponowanych w rozdziałach 3 i 4 zostały zweryfikowane symulacyjnie przy użyciu pakietu Matlab-Simulink. Badano zachowanie układu, w którego skład wchodzi sześć nadajników. Tabela 1 pokazuje kiedy poszczególne źródła danych są aktywne oraz kiedy pozostają wyłączone, tzn. nie są brane pod uwagę w

Tabela 1

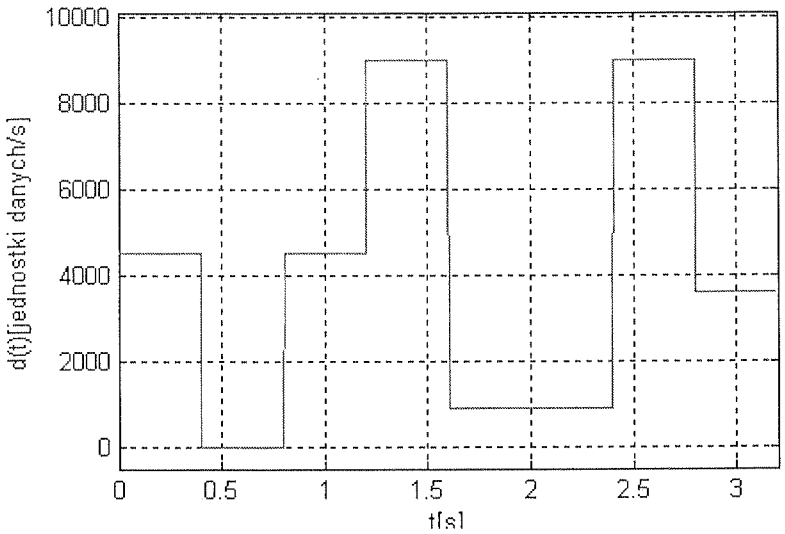
Aktywność nadajników w poszczególnych przedziałach czasu;
[A] — nadajnik aktywny, [–] — nadajnik wyłączony

Source activity: [A] — source on, [–] — source off

	Czas [s]							
	[0,0.4)	[0.4,0.8)	[0.8,1.2)	[1.2,1.6)	[1.6,2.0)	[2.0,2.4)	[2.4,2.8)	[2.8,3.2)
Nadajnik 1	–	A	–	–	A	A	–	–
Nadajnik 2	A	–	–	A	A	A	–	–
Nadajnik 3	A	A	A	–	A	A	A	A
Nadajnik 4	A	–	A	–	A	A	–	–
Nadajnik 5	A	–	A	–	A	A	–	–
Nadajnik 6	A	–	A	–	A	A	–	–

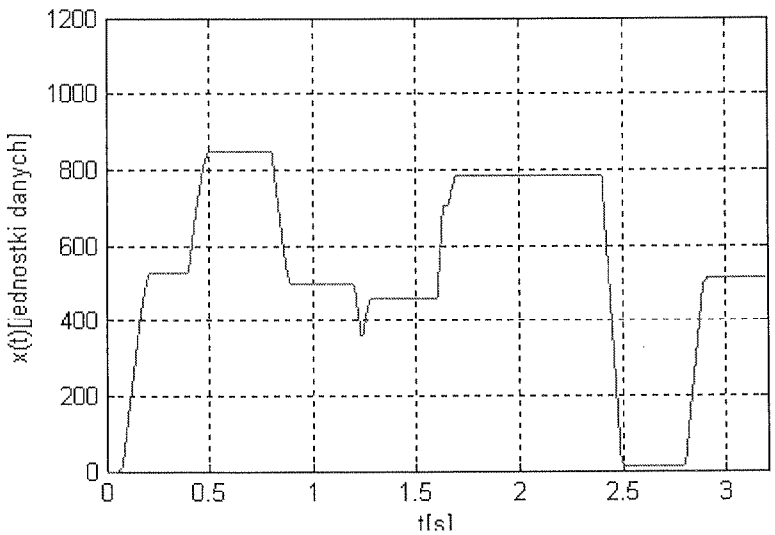
trakcie wyznaczania szybkości nadawania danych przez badane algorytmy. Z każdym nadajnikiem związany jest pojedynczy obwód wirtualny. Czasy RTT pełnego obiegu jednostek sterujących w kolejnych obwodach są równe 60 ms, 30 ms, 80 ms, 50 ms, 60 ms i 70 ms, a okres dyskretyzacji wynosi 10 ms.

Maksymalne dostępne pasmo wąskiego gardła wynosi $d_{\max} = 9000$ jednostek danych na sekundę, natomiast maksymalna łączna szybkość wyznaczana przez algorytm z kompensacją dostępnego pasma jest równa $a_{\max} = 10000$ jednostek na sekundę. Zależność dostępnego pasma od czasu przedstawia rysunek 4. W obu algorytmach współczynnik wzmocnienia wynosi $K = 75\ \mathrm{s}^{-1}$. W pierwszej kolejności symulowano działanie układu z regulatorem proporcjonalnym, który opisano w rozdziale 3. Uwzględniając poszczególne czasy RTT i korzystając z twierdzenia 3, wyznaczono wartość sygnału odniesienia x_d, dla której pasmo będzie zawsze w pełni wykorzystane. W rozważanym przypadku wartość ta powinna wynosić co najmniej 840 jednostek. Rysunek 5 przedstawia przebieg długości kolejki w buforze wąskiego gardła dla rozważanej sytuacji gdy $x_d = 850$ jednostek. Z wykresu widać, że istotnie nie następuje przepełnienie bufora, a zarazem długość kolejki nigdy nie maleje do zera, co jest równoznaczne z pełnym wykorzystaniem dostępnego pasma. Przy tych samych założeniach przeprowadzono także symulację działania algorytmu z kompensacją bezpośrednią dostępnego pasma. Przyjęto współczynnik kompensacji $\varepsilon = 1,25$. Aby spełnione zostały założenia twierdzenia 6, wartość zadaną długości kolejki danych wybrano jako $x_d = 1024$ jednostki. Przebieg długości kolejki w buforze wąskiego gardła uzyskany podczas symulacji przedstawia rysunek 6. Z wykresu wyraźnie widać, że w stanie ustalonym długość kolejki w buforze wąskiego gardła tylko nieznacznie różni się od wartości zadanej tego sygnału. Szczególny przypadek obserwujemy w przedziale czasu od 0,4 do 0,8 sekundy, gdzie wartość $x(t)$ ustala się na stałym wysokim poziomie. Jest to spowodowane spadkiem dostępnego pasma do zera, skutkiem czego żadne jednostki danych nie mogą opuścić bufora i tym samym długość kolejki nie zmienia się. Mniejsze odchylenia od wartości zadanej zostały uzyskane kosztem wzrostu wymaganej pojemności bufora, który musi być większy niż bufor użyty przy zastosowaniu algorytmu bez kompensacji bezpośredniej. Na podstawie twierdzenia 5 można stwierdzić, że dane na pewno nie będą gubione, gdy pojemność bufora będzie wynosiła 2024 jednostki.



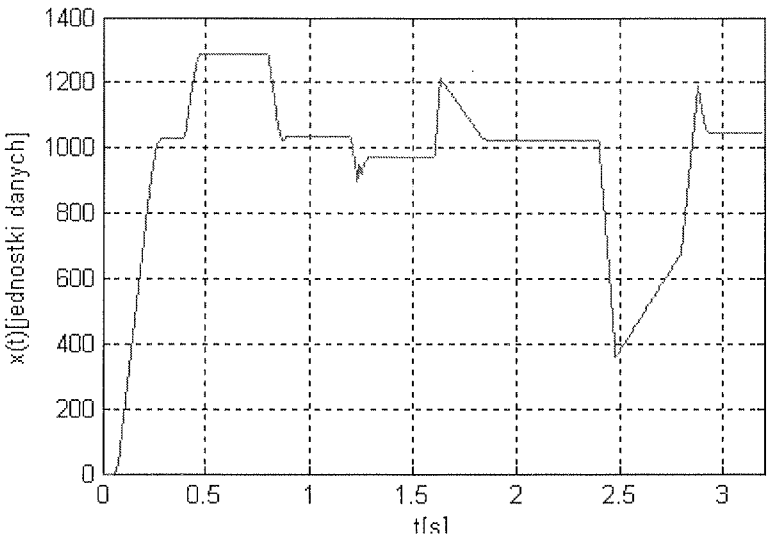
Rys. 4. Dostępne pasmo

Fig. 4. Available bandwidth



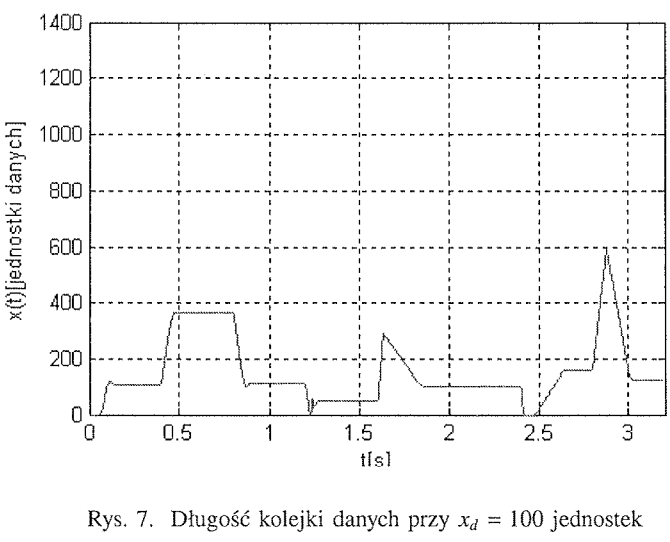
Rys. 5. Długość kolejki danych

Fig. 5. Queue length



Rys. 6. Długość kolejki danych w układzie regulacji ze sprzężeniem bezpośrednim od dostępnego pasma

Fig. 6. Queue length in the system with feed-forward compensation



Rys. 7. Długość kolejki danych przy x_d = 100 jednostek

Fig. 7. Queue length when x_d = 100 pdu

Z rysunku 6 wynika, że w symulowanym przypadku długość kolejki nie jest tak duża, ale należy pamiętać, że wartość graniczna wyznaczona w twierdzeniu 5 odnosi się do wszystkich możliwych sytuacji wynikających ze zmienności liczby obwodów i dostępnego pasma.

Interesujące wyniki działania algorytmu z kompensacją dostępnego pasma można zaobserwować po zmniejszeniu wartości sygnału odniesienia do $x_d = 100$ jednostek. Rysunek 7 przedstawia otrzymany przebieg zmian długości kolejki. Porównując rysunki 5, 6 i 7 można zauważyć, że przy zmienionej wartości x_d dla algorytmu z kompensacją dostępnego pasma znacząco zmniejszyła się średnia długość kolejki danych w buforze. Oznacza to zmniejszenie średniego czasu opóźnienia transmisji wynikającego z kolejkowania danych w buforze węzła. Jest to szczególnie widoczne w przedziałach czasu, w których długość kolejki ustala się na poziomie bliskim sygnałowi odniesienia. Ponieważ wybrana wartość sygnału odniesienia nie spełnia warunków twierdzenia 6, więc pasmo nie jest w pełni wykorzystane. Warto jednak zauważyć, że wykorzystanie dostępnego pasma ulega tylko niewielkiemu pogorszeniu, a więc zmniejszenie czasu transmisji i jego zmienności uzyskuje się kosztem jedynie nieznacznie gorszego wykorzystania zasobów. Jak widać, dobierając odpowiednie wartości parametrów x_d, K i ε można skutecznie kształtować długość kolejki danych w buforze wąskiego gardła, a tym samym wpływać na jakość usług w sieci.

6. PODSUMOWANIE

W artykule zaproponowano nowe dyskretne algorytmy sterowania szybkością nadawania danych w połączeniowej sieci teleinformatycznej, w której liczba źródeł może ulegać zmianie. Przedstawione algorytmy łączą w sobie predyktor Smitha oraz dyskretny regulator proporcjonalny. Drugi z przedstawionych regulatorów, oprócz informacji o stanie sieci dostarczanej w pętli sprzężenia zwrotnego, wykorzystuje także informację o dostępnym paśmie dostarczaną w trybie feed-forward przez sprzężenie bezpośrednie. Wpływ dostępnego pasma na wyznaczaną szybkość nadawania danych może być regulowany przez odpowiedni dobór współczynnika kompensacji ε w algorytmie. Zaproponowane algorytmy pozwalają skutecznie unikać przeciążenia sieci, a tym samym eliminują potrzebę retransmisji danych, oraz zapewniają pełne wykorzystanie łącza. Wykazano, że szybkości generowane przez algorytmy są zawsze nieujemne i ograniczone, co jest warunkiem koniecznym sprzętowej implementacji regulatorów. Należy zwrócić uwagę, że algorytm wykorzystujący sprzężenie bezpośrednie – w stanie ustalonym – charakteryzuje się, mniejszym wpływem dostępnego pasma na długość kolejki danych w buforze wąskiego gardła, niż podobne algorytmy wcześniej proponowane w literaturze. Dzięki temu umożliwia on uzyskanie lepszej jakości usług w rozważanej sieci.

Praca naukowa finansowana ze środków Komitetu Badań Naukowych w latach 2004 – 2006 jako projekt badawczy nr 3T11C 043 26.

7. BIBLIOGRAFIA

1. A. Bartoszewicz: *Nonlinear flow control strategies for connection- oriented communication networks*, Proceedings of the IEE — Part D Control Theory and Applications, 2006, Vol. 153, No. 1, ss. 21-28.
2. A. Bartoszewicz, T. Molik: *ABR traffic control over multi- source single-bottleneck ATM networks*, Journal of Applied Mathematics and Computer Science, 2004, Vol. 14, s. 101–109.
3. F. Blanchini, R. Lo Cigno, R. Tempo: *Robust rate control for integrated services packet networks*, IEEE/ACM Transactions on Networking, 2002, Vol. 10, Issue 5, s. 644–652.
4. F. Gómez-Stern, J. Fornés, F. Rubio: *Dead-time compensation for ABR traffic control over ATM networks*, Control Engineering Practice, 2002, Vol. 10, s. 481–491.
5. O. Imer, S. Compans, T. Basar, R. Srikant: *Available bit rate congestion control in ATM networks*, IEEE Control Systems Magazine, February 2001, s. 38–56.
6. S. Jagannathan, J. Talluri: *Predictive congestion control of ATM networks: multiple sources/single buffer scenario*, Automatica, 2002, Vol. 38, s. 815–820.
7. R. Jain: *Congestion control and traffic management in ATM networks: recent advances and a survey*, Computer Networks and ISDN Systems, 1996, Vol. 28, s. 1723–1738.
8. R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, F. Lu: *ERICA+: Extensions to the ERICA switch algorithm*, Tech. Rep. AF-TM 95-1145R1, 1995.
9. R. Jain, S. Kalyanaraman, R. Viswanathen: *A sample switch algorithm*, Tech. Rep. AF-TM 95-0178R1, 1995.
10. S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, B. Vandalore: *The ERICA switch algorithm for ABR traffic management in ATM networks*, IEEE/ACM Transactions On Networking, 2000, Vol. 8, No. 1, s. 87–98.
11. K.P. Laberteaux, C.E. Rohrs, P.J. Antsaklis: *A practical controller for explicit rate congestion control*, IEEE Transactions on Automatic Control, 2002, Vol. 47, Issue 6, s. 960–978.
12. J.Y. Lee, B.G. Kim, E.B. Lee: *Design of new ABR congestion control algorithm using probabilistic congestion detection in ATM networks*, Electronics Letters, 2000, Vol. 36, Issue 2, s. 178-179.
13. I. Lengliz, F. Kamoun: *A rate-based flow control method for ABR service in ATM networks*, Computer Networks, 2000, Vol. 34, s. 129–138.
14. S. Mascolo: *Congestion control in high-speed communication networks using the Smith principle*, Automatica, 1999, Vol. 35, s. 1921–1935.
15. S. Mascolo: *Smith's principle for congestion control in high-speed data networks*, IEEE Transactions on Automatic Control, 2000, Vol. 45, s. 358–364.
16. D.H. Sun, Q.H. Zhang, Z.C. Mu: *Single parametric fuzzy adaptive PID control and robustness analysis based on the queue size of network node*, Proceedings of 2004 International Conference on Machine Learning and Cybernetics, 2004, Vol. 1, s. 397-400.
17. P. Quet, B. Ataslar, A. Iftar, H. Özbay, S. Kalyanaraman, T. Kang: *Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays*, Automatica, 2002, Vol. 38, s. 917–928.
18. ATM Forum: *Traffic Management Specification Version 4.0*, 1996.

T. MOLIK, A. BARTOSZEWICZ

DISCRETE TIME FLOW CONTROL ALGORITHMS FOR CONNECTION ORIENTED COMMUNICATION NETWORKS

Summary

The paper presents two new discrete time flow control algorithms for connection oriented communication networks. The algorithms effectively combine the Smith predictor with a discrete time, proportional

controller. One of the strategies proposed in the paper uses also the feed-forward compensation of the time varying available bandwidth. This helps reduce transport delay and its variance, consequently ensuring improved quality of service (QoS). It has been proved that the algorithms eliminate the risk of the bottleneck buffer overflow and assure full bandwidth utilisation. The algorithms proposed in the paper maintain these desirable properties not only in the steady state, but also when the number of data sources and the available bandwidth may change during the network operation. The algorithms can support an arbitrary bandwidth allocation, which is their important advantage when compared with other strategies proposed in the past. The favourable properties of the proposed control algorithms are strictly proved and verified by means of a simulation example.

Keywords: congestion control, ATM networks, Smith predictor, time delay systems

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 591–615

Impulsowy system ultraszerokopasmowy (UWB)

RYSZARD J. KATULSKI, JAROSŁAW SADOWSKI

Politechnika Gdańska,
Wydział Elektroniki, Telekomunikacji i Informatyki,
Katedra Systemów i Sieci Radiokomunikacyjnych,
ul. Narutowicza 11/12, 80-952 Gdańsk
e-mail: rjkat@eti.pg.gda.pl, jarod@kastor.ds.pg.gda.pl

Otrzymano 2005.09.05
Autoryzowano 2005.11.28

W pracy przedstawiono zasadę działania systemu radia impulsowego dla potrzeb transmisji danych o dużych przepływnościach, głównie dla potrzeb bezprzewodowych sieci osobistych (WPAN). Opisano sygnał radiowy oraz budowę i działanie układów odbiorczych w tym systemie. Poruszono problem kształtowania widma ultraszerokopasmowego sygnału impulsowego. Scharakteryzowano znane sposoby kształtowania tego widma, wskazując na ich ograniczenia. Zaproponowano modyfikacje tych metod. Omówiono wyniki wykonanych w tym zakresie badań symulacyjnych oraz sformułowano wynikające z tego wnioski użytkowe.

Słowa kluczowe: systemy ultraszerokopasmowe (UWB), radio impulsowe (IR), bezprzewodowe sieci osobiste (WPAN), kształtowanie widma, badania symulacyjne

1. WSTĘP

W ostatnim czasie ma miejsce duże zainteresowanie kanałami radiowymi o dużych przepływnościach, z przeznaczeniem do transmisji danych w bezprzewodowych sieciach osobistych o małych zasięgach ograniczonych do obszaru pojedynczego pomieszczenia — tzw. pikosieciach. W zastosowaniach tych bierze się pod uwagę technikę radia impulsowego IR (*Impulse Radio*), przy użyciu którego, tworzona jest koncepcja rozwiązania radiowego systemu ultraszerokopasmowego UWB (*Ultra Wide Band*) [1, 2].

Przesyłanie informacji przy użyciu bardzo szerokiego widma częstotliwościowego pozwala na uzyskanie dużych szybkości transmisji, na co szczególnie jest zapotrzebowanie w sytuacjach przesyłania strumieni danych multimedialnych.

Ważną właściwością sygnału UWB jest jego niska gęstość widmowa mocy, co pozwala na eksploatację na tym samym obszarze i przy użyciu tych samych zasobów widmowych jednocześnie systemu UWB oraz istniejących radiowych systemów wąskopasmowych, w których są stosowane sygnały o dużo większych poziomach energetycznych. Ma to duże znaczenie praktyczne z punktu widzenia wymagań międzysystemowej kompatybilności elektromagnetycznej, których spełnienie w określonych warunkach eksploatacyjnych można uzyskać na drodze odpowiednio ukształtowanego widma sygnału UWB.

Mając to na uwadze, w wielu ośrodkach naukowych na całym świecie prowadzone są intensywne badania zmierzające do opracowania efektywnych metod kształtowania widma impulsowego sygnału UWB poprzez odpowiedni dobór kształtu impulsu oraz poprzez odpowiedni dobór właściwości statystycznych ciągów rozpraszających stosowanych do tworzenia tego sygnału, co w szczególności zmierza do eliminacji wybranych częstotliwości w jego widmie. W rezultacie tego można w znaczącym stopniu poprawić warunki kompatybilnej pracy systemu UWB z innymi systemami, przez co rozumie się ograniczanie zjawiska ich zakłócania przez składniki widma sygnału UWB oraz zwiększanie odporności odbiorników UWB na zakłócanie sygnałami – zwłaszcza od systemów wąskopasmowych.

Najbardziej znane metody eliminacji składowych o określonych częstotliwościach w widmie sygnału UWB zostały opracowane przez L. Piazzo i J. Romme [3, 4]. Jednakże charakteryzują się one istotnymi ograniczeniami. Autorzy artykułu opracowali modyfikacje tych metod, które zbadali na drodze symulacji komputerowych. Wyniki tych badań wskazują na ich przydatność praktyczną, o czym będzie mowa w drugiej części pracy.

2. TECHNIKA RADIA IMPULSOWEGO

Formalnie rzecz ujmując, system telekomunikacyjny uważa się za ultraszerokopasmowy jeżeli bezwzględna szerokość widma sygnału transmisyjnego w tym systemie wynosi co najmniej 500 MHz lub względna (odniesiona do jego częstotliwości środkowej) szerokość pasma tego sygnału wynosi przynajmniej 20% [5].

W opisywanej technice, jednym z możliwych sposobów spełnienia tych założeń jest zastosowanie ciągów impulsów o bardzo krótkim czasie trwania, rzędu nanosekund. Obok tego ważny jest także odpowiedni dobór kształtu tych impulsów. Przy czym do realizacji procesu zapisu informacji stosuje się wybrany rodzaj modulacji impulsowej, tzn.:

- modulację amplitudy impulsów PAM (*Pulse Amplitude Modulation*), lub
- modulację położenia impulsów PPM (*Pulse Position Modulation*).

Ponadto, mając na uwadze teleinformatyczne przeznaczenie opisywanego systemu, należy przyjąć, że w jego skład wchodzi wiele pojedynczych radiowych łączy UWB — np. pomiędzy serwerem i poszczególnymi stacjami roboczymi, co oznacza jednoczesną pracę na tym samym obszarze wielu nadajników i odbiorników radiowych.

2.1. OPIS SYGNAŁU RADIOWEGO

Dla potrzeb analitycznego opisu impulsowego sygnału radiowego przyjmijmy, że system UWB składa się z N_u jednocześnie pracujących łączy radiowych. Impulsowy sygnał wyjściowy $s_{nad,m}(t)$ z m-tego nadajnika $1 \geq m \geq n_u$, dla przypadku modulacji PAM, może być opisany poniższym wzorem:

$$s_{nad,m}(t) = \sum_{j=-\infty}^{\infty} \left(a_{m,j} \cdot w_{nad}\left(t - j \cdot T_f - c_{m,j} \cdot T_c\right)\right), \tag{1}$$

w którym: $a_{m,j}$ — j-ta próbka przebiegu/ciągu modulującego w nadajniku, $w_{nad}(t)$ — funkcja opisująca kształt pojedynczego impulsu, T_f — czas powtarzania impulsów, nazywany czasem trwania ramki, który może być kilka rzędów dłuższy od czasu trwania pojedynczego impulsu, T_c — czas trwania pojedynczego chipu, $c_{m,j}$ — wartość pseudoprzypadkowa, która wpływa na dodatkowe opóźnienie impulsu określone zależnością $c_{m,j} \cdot T_c$, przy czym $c_{m,j}$ jest liczbą całkowitą z przedziału $0 \geq c_{m,j} < N$, zaś $N_h \cdot T_c \geq T_f$ dobierane jest tak, aby zminimalizować możliwość kolizji przy jednoczesnym nadawaniu sygnałów przez kilka nadajników.

Natomiast w przypadku zastosowania modulacji PPM sygnał $s_{nad,m}(t)$ wyraża się w poniższy sposób:

$$s_{nad,m}(t) = \sum_{j=-\infty}^{\infty} w_{nad}\left(t - j \cdot T_f - c_{m,j} \cdot T_c - d_{m,j}\right), \tag{2}$$

gdzie $d_{m,j}$ oznacza kolejne dodatkowe przesunięcie j-tego impulsu zależne od wartości próbki sygnału modulującego w m-tym nadajniku.

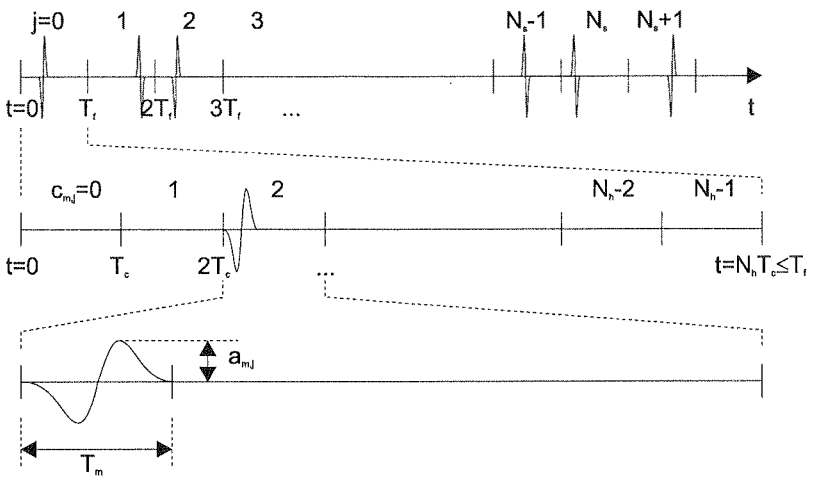
Z zapisów sygnału (2.1.1) oraz (2.1.2) wynika, że składa się on z ciągu impulsów określanych w literaturze mianem pulse train, przy czym położenie czasu transmisji j-tego impulsu z m-tego nadajnika jest określone zależnością:

a) w przypadku PAM: $-j \cdot T_f - c_{m,j} \cdot T_c$, natomiast
b) w przypadku PPM: $-j \cdot T_f - c_{m,j} \cdot T_c - d_{m,j}$.

Postać czasową takich sygnałów impulsowych pokazano na rys. 1 i 2.

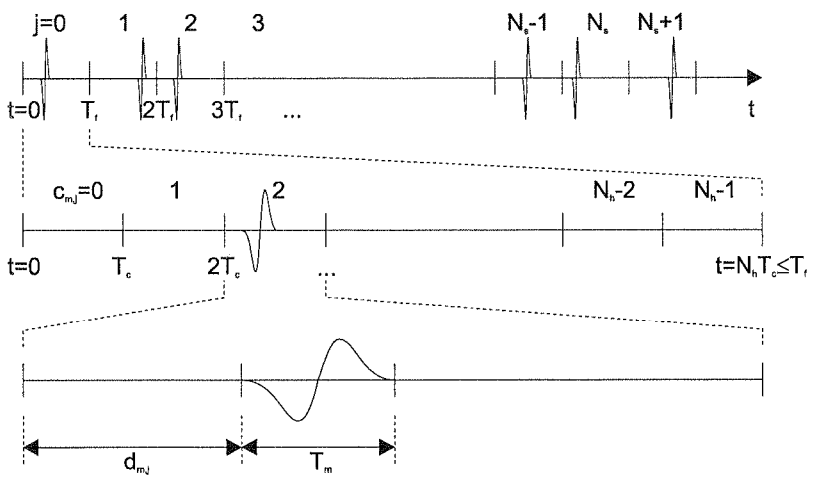
Należy podkreślić, że widmo pojedynczego impulsu determinuje widmo sygnału impulsowego UWB, przy czym najczęściej stosowane są impulsy, których kształty opisują różnego rzędu pochodne impulsu Gaussowskiego — np. dublet gaussowski opisany poniższą funkcją [6]:

$$w_{nad}(t) = (t - \tau) \cdot \exp(-2\pi(\frac{t - \tau}{\tau})^2), \tag{3}$$



Rys. 1. Postać czasowa sygnału impulsowego UWB z PAM dla binarnego sygnału modulującego

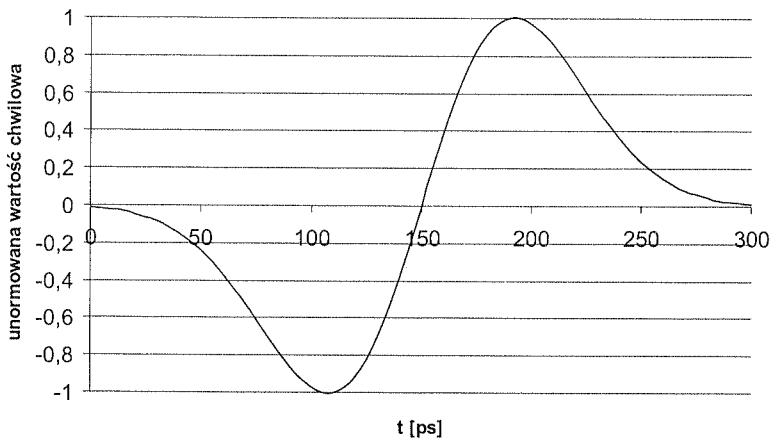
Fig. 1. Time form of the impulse UWB signal with PAM for binary modulation signal



Rys. 2. Postać czasowa sygnału impulsowego UWB z PPM dla binarnego sygnału modulującego

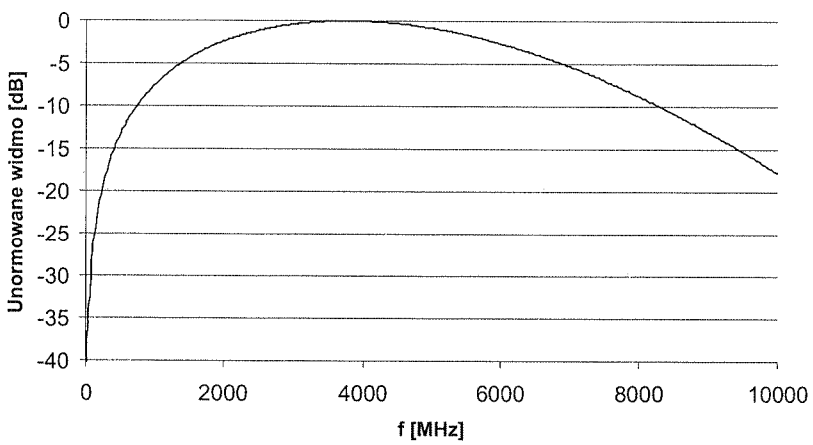
Fig. 2. Time form of the impulse UWB signal with PPM for binary modulation signal

którego widmo amplitudowe (–10 dB–we) dla $\tau = 150ps$ rozciąga się od 740 do 8300 MHz. Kształt tego impulsu oraz jego widmo amplitudowe pokazano na rys. 3 i 4. Inne stosowane w badaniach i eksperymentach kształty impulsów sygnału UWB są opisane w [7] i [8].



Rys. 3. Przykładowy kształt impulsu stosowanego w systemie UWB

Fig. 3. Example shape of the impulse signal for UWB system



Rys. 4. Widmo amplitudowe pojedynczego impulsu

Fig. 4. The single impulse spectrum

Stosownie do przyjętego na wstępie założenia, tzn., gdy w rozpatrywanym systemie UWB jest aktywnych jednocześnie N_u nadajników — źródeł sygnałów opisanych zależnością (1) albo (2), sygnał $r(t)$ odbierany na wejściu określonego odbiornika można wyrazić w następujący sposób:

$$r(t) = \sum_{m=1}^{N_u} s_{odb,m}(t - \tau_m) + n(t) \tag{4}$$

w którym: $s_{odb,m}(t)$ — impulsowy sygnału UWB odebrany z kolejnego m-tego nadajnika, przy czym jeden z tych sygnałów jest użyteczny, zaś pozostałe są zakłócające [9], τ_m — zmienna losowa opisująca rozbieżność wzorca czasu pomiędzy m-tym nadajnikiem i rozpatrywanym odbiornikiem, co wynika z czasu propagacji tego sygnału oraz ewentualnej niedokładności synchronizacji tych urządzeń, $n(t)$ — reprezentacja pozostałych sygnałów zakłócających i szumowych.

Na zakończenie tego podrozdziału należy dodać, że transmisja sygnału impulsowego wiąże się ze zmianą jego kształtu [10]. Z tego powodu sygnał $s_{odb,m}(t)$ odebrany z m-tego nadajnika modeluje się serią impulsów będących w najprostszym przybliżeniu pochodną kształtu impulsów nadanych. Zatem przykładowo dla przypadku impulsu nadawanego w postaci dubletu gaussowskiego (3), impuls odebrany $w_{odb}(t)$ wyraża się zależnością:

$$w_{odb}(t) = \left[1 - 4\pi\left(\frac{t-\tau}{\tau}\right)^2\right] \cdot \exp\left(-2\pi\left(\frac{t-\tau}{\tau}\right)^2\right). \tag{5}$$

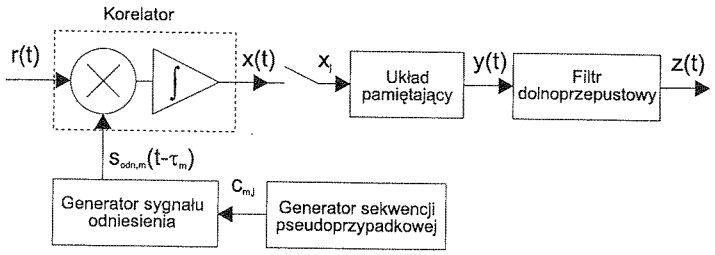
2.2. BUDOWA I DZIAŁANIE ODBIORNIKA

2.2.1. Sygnał źródłowy analogowy

Uproszczony schemat blokowy odbiornika UWB przeznaczonego do odbioru sygnału impulsowego opisanego zależnością (1) albo (2), w przypadku, gdy sygnał modulujący (źródłowy) ma charakter analogowy, przedstawiono na rys. 5, gdzie dla ułatwienia rozważań ograniczono się tylko do zobrazowania opisywanej specyfiki takiego odbioru. Przy czym, bez utraty ogólności tych rozważań można przyjąć, że odbiornik ten jest idealnie zsynchronizowany z sygnałem odbieranym przykładowo z m-tego nadajnika, co oznacza, że z tym nadajnikiem odbiornik ten tworzy rozpatrywane łącze radiowe, co zaś z kolei oznacza, że zarówno ciąg wartości $\{c_{m,j}\}$ we wzorach (1) albo (2) jak i zmienna τ_m we wzorze (4) są znane.

Przy takich założeniach sygnał odniesienia $s_{odn,m}(t)$ w korelatorze tego odbiornika ma postać:

$$s_{odn,m}(t - \tau_m) = \sum_{j=-\infty}^{\infty} w_{cor}\left(t - j \cdot T_f - c_{m,j} \cdot T_c - \tau_m\right). \tag{6}$$



Rys. 5. Uproszczony układ odbiornika UWB dla sygnału źródłowego analogowego

Fig. 5. Simple form of the UWB receiver for analog source signal

Jak widać sygnał ten ma postać analogiczną do sygnału nadawanego przez $m-ty$ nadajnik przy braku modulacji, przy czym $w_{cor}(t)$ opisuje wzorcowe impulsy odniesienia [11].

Zatem, na wyjściu korelatora w trakcie trwania j-tej ramki otrzymujemy wartość x_j, która wynika z poniższej zależności:

$$x_j = \int_{-\infty}^{\infty} \left[r(t) \cdot w_{cor}\left(t - j \cdot T_f - c_{m,j} \cdot T_c - \tau_m\right)\right] dt, \tag{7}$$

przy czym rzeczywisty zakres całkowania jest ograniczony do skończonego przedziału czasu, w którym $w_{cor}(t - j \cdot T_f - c_{m,j} \cdot T_c - \tau_m)$ jest niezerowe.

W rezultacie na wartość x_j składa się poniższa suma:

$$x_j = \sum_{i=1}^{i=m-1} s_{i,j} + s_{m,j} + \sum_{i=m+1}^{i=N_u} s_{i,j} + n_j, \tag{8}$$

w której:

$$\begin{aligned} s_{i,j} &= \int_{-\infty}^{\infty} a_{i,j} \cdot w_{odb}\left(t - jT_f - c_{i,j}T_c - \tau_i - d_{i,j}\right) \cdot w_{cor}\left(t - jT_f - c_{m,j}T_c - \tau_m\right) dt \\ &= a_{i,j} \cdot R_w\left(d_{i,j} + \left(c_{i,j} - c_{m,j}\right)T_c + \tau_i - \tau_m\right) \end{aligned} \tag{9}$$

jest składnikiem wartości x_j na wyjściu korelatora powiązanym z sygnałem odebranym w ogólności z i-tego nadajnika, przy czym $s_{m,j}$ (dla $i = m$) — składnik tej wartości powiązanym z sygnałem użytecznym odebranym z m-tego nadajnika, gdzie $R_w(\tau)$ jest funkcją korelacji wzajemnej pomiędzy pojedynczym impulsem odniesienia generowanym w odbiorniku $w_{cor}(t)$ i impulsem odebranym $w_{odb}(t)$, tzn.:

$$R_w(\tau) = \int_{-\infty}^{\infty} \left(w_{cor}(t + \tau) \cdot w_{odb}(t)\right) dt, \tag{10}$$

zaś składnik n_j wyrażony zależnością:

$$n_j = \int_{-\infty}^{\infty} \left(n(t) \cdot w_{cor}\left(t - j \cdot T_f - c_{m,j} \cdot T_c - \tau_m\right)\right) dt, \tag{11}$$

jest tą częścią składowej szumu i zakłóceń nieimpulsowych, która ma wpływ na odbiór sygnału z m-tego nadajnika.

W zależności od stosowanego rodzaju modulacji różne są kryteria doboru kształtu impulsów odniesienia $w_{cor}(t)$. W przypadku PAM, gdy we wzorze (9) nie występuje składnik $d_{i,j}$, impuls odniesienia powinien być kopią kształtu impulsu odbieranego. Wówczas informacja użyteczna $a_{m,j}$, zawarta w składniku $s_{m,j}$ we wzorze (8), jest dostępna wprost na wyjściu korelatora.

Z kolei w przypadku stosowania PPM najkorzystniejszy z punktu widzenia minimalizacji zniekształceń składowej użytecznej odbieranego sygnału jest taki kształt impulsu wzorcowego, który zapewnia uzyskanie nieparzystej funkcji korelacji wzajemnej $R_w(\tau)$, zdefiniowanej zależnością (10), która w jak największym stopniu spełnia poniższe przybliżenie:

$$R_w(\tau) \approx \dot{R}_w(0) \cdot \tau, \tag{12}$$

przy czym $\dot{R}_w(0)$ oznacza pierwszą pochodną. Zatem (12) stanowi przybliżenie pierwszego rzędu funkcji korelacji według rozwinięcia Maclaurina.

Przebieg przykładowej funkcji korelacji wzajemnej zilustrowano na rys. 6. Jeśli system odbiorczy pracuje w zakresie przybliżenia (12), co na rys. 6 zaznaczono linią przerywaną, to wówczas informacja z m-tego nadajnika jest dostępna jako jeden ze składników sygnału na wyjściu korelatora:

$$s_{m,j} = R_w(d_{m,j}) \approx \dot{R}_w(0) \cdot d_{m,j}. \tag{13}$$

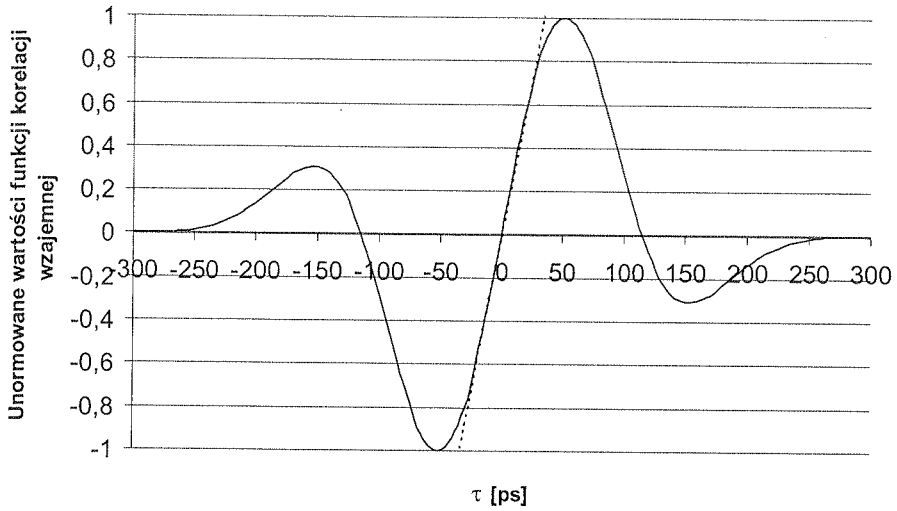
Zminimalizowanie lub teoretycznie nawet całkowite usunięcie niepożądanych składników w sygnale na wyjściu korelatora, tzn.: $a_{i,j}$ (w przypadku PAM) albo $d_{i,j}$ (w przypadku PPM), pochodzących od pozostałych pracujących w danej chwili nadajników (), jest możliwe przez zapewnienie ortogonalności ciągów pseudoprzypadkowych $\{c_{i,j}\}$ stosowanych w poszczególnych nadajnikach systemu, tak by dla $i \neq m$ zależność:

$$c_{i,j} \neq c_{m,j}, \tag{14}$$

była spełniona dla każdego j.

Aby z ciągu wartości x_j z wyjścia korelatora (por. (8)) otrzymać ciągły sygnał użyteczny, wymagane jest zastosowanie układu pamiętającego oraz filtru interpolacyjnego. Działanie układu pamiętającego może być od strony formalnej zamodelowane splotem ciągu wartości x_j z funkcją bramkującą, opisaną wzorem:

$$\bar{T}_f(t) = \begin{cases} 1, & dla \ \ |t| < T_f/2, \\ 0, & dla \ \ |t| \geqslant T_f/2. \end{cases} \tag{15}$$



Rys. 6. Przykładowa funkcja korelacji wzajemnej kształtu impulsu wzorcowego i odebranego przy modulacji PPM

Fig. 6. The exemplary cross-correlation function of template and received pulse in PPM modulation

W wyniku takiej operacji splatania na wyjściu filtru interpolacyjnego w postaci interpolatora schodkowego 1–rzędu otrzymujemy sygnał $y(t)$, który stanowi poniższą sumę:

$$y(t) = \sum_{j=-\infty}^{\infty} \left[x_j \cdot \overline{T_f}\left(t - j \cdot T_f - \gamma\right) \right], \tag{16}$$

gdzie

$$\gamma = \tau_m + T_f \tag{17}$$

W rezultacie sygnał $z(t)$ na wyjściu filtru interpolacyjnego zawiera składową użyteczną, teoretycznie będącą wierną rekonstrukcją przebiegu modulującego $a_m(t)$ otrzymanego z próbek $a_{m,j}$ zawartych wprost w ciągu wartości x_j, w przypadku PAM, albo rekonstrukcją przebiegu modulującego $d_m(t)$ z próbek $d_{m,j}$ w przypadku PPM.

2.2.2. *Sygnał źródłowy cyfrowy*

W impulsowym systemie UWB transmisja informacji cyfrowej odbywa się przy zastosowaniu ograniczonej do dwóch liczby możliwych wartości próbek $a_{m,j}$ ciągu modulującego przy modulacji PAM albo próbek $d_{m,j}$ ciągu modulującego przy modulacji PPM, co oznacza, że ciąg $\{a_{m,j}\}$w wyrażeniu (1) jest zbiorem wartości binarnych –1 albo 1, zaś ciąg $\{d_{m,j}\}$ w wyrażeniu (2) jest zbiorem wartości binarnych: 0 albo

δ, gdzie δ jest określonym odcinkiem czasu porównywalnym z T_m — zastosowana notacja odnosi się oczywiście do m-tego nadajnika.

Pamiętając o tym, że czas powtarzania impulsów T_f jest zazwyczaj wielokrotnie mniejszy od czasu nadawania kolejnych symboli informacji, należy mieć na uwadze, że pojedynczy symbol — w tym przypadku „0"' albo „1" — jest nadawany przy użyciu N_s kolejnych następujących po sobie impulsów. Stąd kolejne symbole danych są numerowane indeksem k ($0 \le k < N_b$, gdzie N_b oznacza liczbę przesyłanych symboli), zaś kolejne impulsy składające się na symbol są oznaczone indeksem n ($0 \le n < N_s$). Takie same znaczenie mają indeksy n i k przy kolejnych elementach ciągu pseudoprzypadkowego $\{c_{m,n,k}\}$, gdzie m w dalszym ciągu oznacza numer nadajnika.

Przyjmując, że nadawanie k-tego symbolu rozpoczyna się dla $n = 0$, odbiornik musi podjąć decyzję binarną w czasie $T_s = N_s \cdot T_f$, a więc dla $0 \le n < N_s$. Realizuje się to poprzez wybór bardziej prawdopodobnej z dwóch poniższych hipotez, różnych dla każdej z wymienionych metod modulacji. Tzn. przy PAM:

a) dla wartości binarnej „1"

$$r_k(t) = \sum_{n=0}^{N_s-1} w_{odb}\left(t - \tau_m - n \cdot T_f - c_{m,n,k} \cdot T_c\right) + n(t), \tag{18}$$

b) dla wartości binarnej „0"

$$r_k(t) = -1 \cdot \sum_{n=0}^{N_s-1} w_{odb}\left(t - \tau_m - n \cdot T_f - c_{m,n,k} \cdot T_c\right) + n(t), \tag{19}$$

Zaś przy PPM:

a) dla wartości binarnej „1"

$$r_k(t) = \sum_{n=0}^{N_s-1} w_{odb}\left(t - \tau_m - n \cdot T_f - c_{m,n,k} \cdot T_c - \delta\right) + n(t), \tag{20}$$

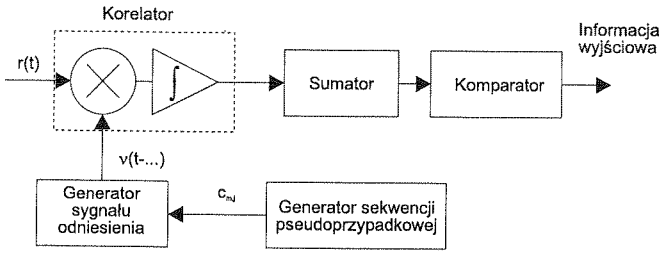
b dla wartości binarnej „0"'

$$r_k(t) = \sum_{n=0}^{N_s-1} w_{odb}\left(t - \tau_m - n \cdot T_f - c_{m,n,k} \cdot T_c\right) + n(t), \tag{21}$$

gdzie δ oznacza przesunięcie czasowe o które dodatkowo opóźniane jest nadawanie symbolu odpowiadającego wartości binarnej „1" — wyłącznie przy PPM. W powyższych wyrażeniach wszystkie inne sygnały $n(t)$, niepożądane z punktu widzenia m-tego nadajnika i odbiornika, a więc zakłócające sygnały impulsowe od innych nadajników oraz inne zakłócenia nie mające charakteru impulsowego i szum, zostały ujęte jako:

$$n(t) = \sum_{i=1}^{i=m-1} s_{odb,i}(t - \tau_i) + \sum_{i=m+1}^{i=N_u} s_{odb,i}(t - \tau_i) + n_s(t), \tag{22}$$

gdzie $S_{odb,i(t)}$ wyraża impulsowe sygnały zakłócające, natomiast $n_s(t)$ oznacza inne zakłócenia nieimpulsowe i szumy.



Rys. 7. Uproszczony układ odbiornika UWB dla sygnału źródłowego cyfrowego

Fig. 7. Simple form of the UWB receiver for digital source signal

W najprostszym przypadku odbioru sygnału tylko z jednego nadajnika, przyjmując kanał z addytywnym szumem gaussowskim, odbiór impulsowego sygnału UWB może być zrealizowany przy użyciu odbiornika korelacyjnego przedstawionego na rys. 7. Jednakże obecność innych impulsowych sygnałów zakłócających powoduje, że optymalną regułą odbioru jest detekcja łączna wszystkich odebranych sygnałów impulsowych. Jest to możliwe dla ograniczonej liczby tych sygnałów. W praktyce oznacza to co najwyżej kilkadziesiąt impulsowych łączy UWB pracujących jednocześnie w tym samym obszarze (pomieszczeniu). W obu skrajnych sytuacjach, tzn. w przypadku najprostszym oraz w przypadku dużej liczby jednocześnie pracujących łączy, decyzję b_k o wartości k-tego symbolu binarnego nadawanego przez m-ty nadajnik można zapisać w poniższy sposób:

$$b_k = \begin{cases} 0, \text{ dla } \sum_{n=0}^{n=N_s-1} \int_{\tau_k+n\cdot T_f}^{\tau_k+(n+1)\cdot T_f} r(t)\cdot v\left(t-\tau_m-n\cdot T_f-c_{m,n,k}\cdot T_c\right)\cdot dt > 0, \\ 1, \text{ dla } \sum_{n=0}^{n=N_s-1} \int_{\tau_k+n\cdot T_f}^{\tau_k+(n+1)\cdot T_f} r(t)\cdot v\left(t-\tau_m-n\cdot T_f-c_{m,n,k}\cdot T_c\right)\cdot dt \leqslant 0, \end{cases} \tag{23}$$

gdzie funkcja $v(t) = w_{odb,,0''}(t) - w_{odb,,1''}(t)$ wyraża różnicę pomiędzy wzorcami impulsów odebranych dla nadawanej informacji logicznej „0"'oraz „1". Przy PAM funkcja ta ma taki sam kształt jak wzorzec sygnału odebranego $v(t) = 2\cdot w_{odb}(t)$, zaś przy PPM jest to różnica dwóch impulsów przesuniętych w czasie, tzn. $v(t) = w_{odb}(t) - w_{odb}(t-\delta)$, przy czym $w_{odb}(t)$ jest niezerowe w przedziale czasu: $0 \leq t \leq T_m$, natomiast $v(t)$ jest niezerowe w przedziale czasu: $0 \leq t \leq T_m + \delta$.

2.3. KSZTAŁTOWANIE WIDMA SYGNAŁU

Jak to napisano na wstępie, użytecznie ważny jest dobór odpowiedniego kształtu widma impulsowego sygnału UWB, co przede wszystkim polega na wytłumianiu wybranych składników tego widma. Przy czym, niezależnie od rodzaju modulacji — dla przypomnienia pod uwagę brana jest modulacja impulsowa PAM i PPM — widmo sygnału UWB można kształtować poprzez odpowiednio dobrane właściwości ciągów rozpraszających. Znane metody eliminacji wybranych częstotliwości w widmie sygnału UWB [3, 4], które pokrótce zostaną opisane poniżej, nie pozwalają na kształtowanie tego widma poza wytłumianymi częstotliwościami, co w szczególności dotyczy wyrównywania stałości jego rozkładu w sąsiedztwie tych częstotliwości. Proponowane modyfikacje tych metod polegają na sposobie doboru pseudoprzypadkowych ciągów rozpraszających [12].

Przedmiotem dalszych rozważań są metody kształtowania widma impulsowego sygnału UWB z modulacją (kluczowaniem) amplitudy impulsów — PAM, przy czym dla formalności zapisu przyjęto, że każdy symbol o czasie trwania T_s jest nadawany jako seria N_s impulsów o kształcie opisanym funkcją $w_{nad}(t)$. Każdy z tych impulsów jest nadawany w oddzielnej ramce o czasie trwania: $T_f = T_s/N_s$, zaś każda ramka jest podzielona na N_h chipów o czasie trwania: $T_h = T_f/N_h$, przy czym o wyborze chipu, w którym jest nadawany impuls, decyduje pseudoprzypadkowy ciąg rozpraszający. Dla przejrzystości analizy wygodnie jest taką sekwencję pseudoprzypadkową interpretować jako ciąg N_s — elementowych wektorów kodowych. Wówczas *n-ty* element *k-tego* wektora oznaczony jako $c_{n,k}$, gdzie $(n = 0, 1, 2, \ldots, N_s - 1)$, przy czym $c_{n,k} \in \{0, 1, 2, \ldots, N_h - 1\}$, określa numer chipu, w którym jest nadawany *n-ty* impuls *k-tego* symbolu. W dalszej części pominięte zostały indeksy m oznaczające numer omawianego nadajnika, gdyż metody kształtowania widma sygnału mogą być stosowane w każdym nadajniku indywidualnie.

Przy założeniu liczby N_b nadawanych symboli, zapis czasowy $s_{nad}(t)$ opisywanego sygnału UWB ma postać:

$$s_{nad}(t) = \sum_{k=0}^{N_b-1} \left[a_k \cdot \sum_{n=0}^{N_s-1} w_{nad}\left(t - c_{n,k} \cdot T_c - n \cdot T_f - k \cdot T_s\right) \right], \tag{24}$$

w której $a_k \in \{-1, 1\}$ wyraża dane binarne modulujące (kluczujące) amplitudę nadawanych impulsów. Jeśli prawdopodobieństwo występowania obu wartości a_k jest jednakowe, to widmo gęstości mocy $P_u(f)$ sygnału UWB może być opisane wzorem [4, 13]:

$$P_u(f) = |W_{nad}(f)|^2 \cdot P_c(f) \tag{25}$$

w której $W_{nad}(f)$ oznacza transformatę Fouriera pojedynczego impulsu $w_{nad}(t)$, zaś $P_c(f)$ odpowiada za wpływ na widmo $P_u(f)$ właściwości ciągu pseudoprzypadkowego.

Tzn.:

$$P_c(f) = \sum_{k=0}^{N_b-1} |T_k(f)|^2, \tag{26}$$

gdzie

$$T_k(f) = \sum_{n=0}^{N_s-1} \exp\left(-j2\pi \cdot f \cdot \left(c_{n,k} \cdot T_c + n \cdot T_f + k \cdot T_s\right)\right). \tag{27}$$

Powyższe zależności pokazują, że jest możliwe kształtowanie widma sygnału UWB poprzez dobór właściwości ciągu rozpraszającego $c_{n,k}$.

W dalszych rozważaniach i przeprowadzonych symulacjach przyjęto także, że przebieg widma $W_{nad}(f)$ pojedynczego impulsu jest stały, tzn.: $W_{nad}(f) = 1$, co odpowiada stosowaniu teoretycznych impulsów o przebiegu czasowym $w_{nad}(t)$ opisanym funkcją:

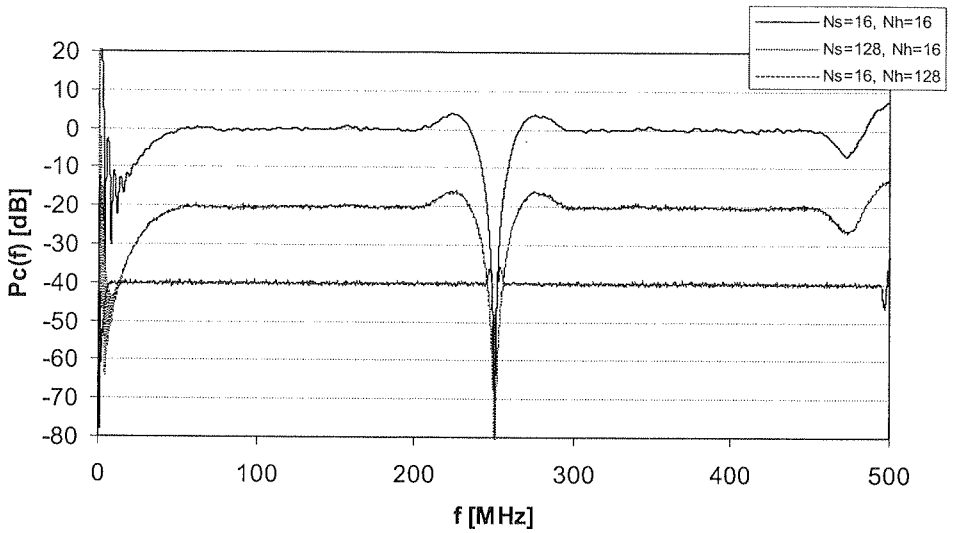
$$w_{nad}(t) = \frac{\sin\left(\frac{\pi \cdot t}{T_c}\right)}{\frac{\pi \cdot t}{T_c}}. \tag{28}$$

Przy takim założeniu symulowane widma sygnałów UWB wynikają wyłącznie z właściwości stosowanych ciągów rozpraszających.

Aby ułatwić porównanie otrzymanych wyników badań symulacyjnych z rezultatami prac zawartymi w [3], przyjęto, że czas T_c trwania pojedynczego chipu wynosi $1ns$ ($T_c = 1ns$). Z tego powodu widmo badanych sygnałów (bez powtórzeń) zawarte jest w paśmie od 0 do 500MHz i tylko taki zakres widma przedstawiony został na wszystkich wykresach. Opisywane metody kształtowania widma i wynikające z badań wnioski pozostają słuszne przy zmianie czasu T_c, jednakże wówczas widmo powinno być przeskalowane w dziedzinie częstotliwości bez zmiany jego kształtu.

2.4. WYTŁUMIANIE WYBRANEJ CZĘSTOTLIWOŚCI

Analizowane metody kształtowania widma sygnału radia impulsowego opierają się na doborze wartości poszczególnych elementów stosowanych ciągów pseudoprzypadkowych. Przy ustalonych parametrach transmisji takich jak liczba N_s ramek w każdym symbolu oraz liczba N_h chipów w ramce, ciąg pseudoprzypadkowy stosowany przy generacji każdego symbolu może przyjmować tylko jedną z możliwej liczby $N_s \cdot N_h$ jego postaci. Skończona liczba możliwych realizacji ciągu rozpraszającego powoduje, że również skończony jest zbiór częstotliwości możliwych do całkowitego wyeliminowania w widmie sygnału impulsowego. Z tego względu inne są reguły doboru wartości elementów ciągu $c_{n,k}$, przeznaczonych do eliminacji określonych częstotliwości, spełniających pewne szczególne zależności pomiędzy T_c, N_s i N_h ujęte w opisie metod: pierwszej i drugiej, a inne przy wytłumianiu w widmie sygnału dowolnej częstotliwości nie spełniającej tych zależności, gdy uzyskanie całkowitej eliminacji składnika widma jest niemożliwe, co opisano w metodzie trzeciej.



Rys. 8. Widma impulsowych sygnałów UWB otrzymane przy użyciu algorytmu pierwszego

Fig. 8. UWB impulse signal spectrum obtained by using the first algorithm

2.4.1. Metoda pierwsza

Pierwsza z zaproponowanych przez Piazzo i Romme'a metod dotyczących wytłumiania ściśle określonej częstotliwości $f_0 = 1/(K \cdot T_c)$ z widma sygnału UWB, polega na traktowaniu N_s składników sumy (3.4) tego widma jako wektorów K — wartościowego sygnału PSK (Phase Shift Keying). Tzn., że składowa widma o częstotliwości f_0 może być zapisana w poniższy sposób:

$$T_k(f_0) = \sum_{n=0}^{N_s-1} e^{-j \cdot \frac{2\pi}{K} \cdot (c_{n,k} + n \cdot N_h + k \cdot N_h \cdot N_s)}. \tag{29}$$

Przy założeniu, że K jest naturalną liczbą parzystą, każdy punkt konstelacji PSK ma swój odpowiednik o przeciwnej fazie. Grupując zatem parami składniki sumy (27) i dobierając różnicę faz równą $\pm\pi rad$ pomiędzy składnikami każdej pary, można całkowicie wyeliminować składową o częstotliwości f_0 z omawianego widma.

Zaproponowany w [3] algorytm przewiduje regularność tworzenia ciągu rozpraszającego, tzn. elementy ciągu $\{c_{2n,k}\}$ są losowane z zakresu od 0 do $N_h - 1$, natomiast elementy ciągu $\{c_{2n+1,k}\}$ oblicza się tak, aby spełniony był poniższy warunek:

$$[c_{2n+1,k}]_K = \left[c_{2n,k} - N_h - \frac{K}{2} \right]_K, \tag{30}$$

w którym $[\,]_K$ oznacza resztę z dzielenia przez K.

Przykładowe widma sygnałów UWB otrzymane na drodze symulacji komputerowych, z których wyeliminowano składową o wybranej częstotliwości $f_0 = 250MHz$ (teoretycznie, składowa o tej częstotliwości powinna zostać całkowicie usunięta z widma sygnału), przedstawiono na rys. 8, przy czym dla przejrzystości porównania w symulacjach tych przyjęto następujące zestawy wartości parametrów N_s i N_h: ($N_s = 16$, $N_h = 16$), ($N_s = 128$, $N_h = 16$) oraz ($N_s = 16$, $N_h = 128$). Ponadto, dla ułatwienia odróżnienia poszczególnych widm zastosowano zróżnicowanie poziomów energetycznych sygnałów w taki sposób, by uzyskane wartości średnie mocy widma wynosiły 0, –20 i –40 dB. Zróżnicowanie to nie ma wpływu na kształt przebiegów ani też na wielkość uzyskiwanych tłumień, odnoszonych zawsze do wartości średniej widma mocy.

2.4.2. Metoda druga

Podobnie jak poprzednio, metoda ta służy do wyeliminowania z widma sygnału UWB składowej o częstotliwości $f_0 = 1/(K \cdot T_c)$, przy czym $2 \leq K \leq N_h$. Jednakże w tym przypadku zakłada się, że N_s jest całkowitą wielokrotnością K. Wówczas N_s wektorów w sumie (3.6) należy do konstelacji K — wartościowej modulacji PSK. Jak wiadomo, suma wszystkich wektorów konstelacji PSK jest równa zero, zatem można to wykorzystać do eliminacji składowej o wybranej częstotliwości poprzez łączenie N_s wektorów w grupy o liczności K i zapewnienie, że w każdej z tych grup każdy wektor konstelacji jest użyty tylko jeden raz.

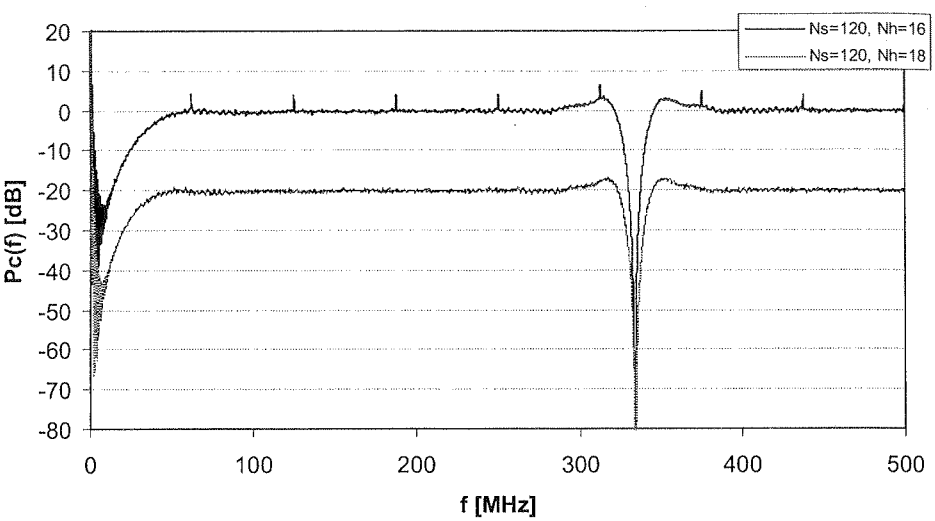
Metoda realizująca powyższą koncepcję polega na wygenerowaniu wektora $[p_0, \ldots, p_{K-1}]$, będącego losową permutacją liczb od 0 do $K-1$, dla każdej grupy K elementów ciągu $\{c_{n,k}\}$ i takim doborze wartości tych elementów, by spełniony był poniższy warunek:

$$[c_{n,k} + n \cdot N_h]_K = p_n, \tag{31}$$

dla $n = 0, \ldots, K-1$. Przy czym N_h powinno być całkowitą wielokrotnością K, co wynika z faktu, że jeśli $\frac{N_h}{K}$ nie jest liczbą całkowitą, to wówczas elementy ciągu $\{c_{n,k}\}$ mogą przyjmować wartości z zakresu tylko od 0 do $K \cdot (N_h/K) - 1$. Pominięcie wartości $\{c_{n,k}\}$ z zakresu od $N_h - [N_h]_K$ do $N_h - 1$ pociąga za sobą utworzenie w widmie sygnału niepożądanych prążków o częstotliwościach $m/(N_h \cdot T_c)$, gdzie $m = 0, \ldots, N_h - 1$. Efekt ten można zaobserwować na rys. 9 dla $N_h = 16$.

W skrótowym ujęciu, algorytm postępowania przy realizacji tej metody przedstawia się następująco:

- dla każdej grupy K elementów sumy (3.6) wygenerować permutację p_n,
- dla każdego wektora w grupie wylosować liczbę całkowitą s z przedziału $0 \leq s \leq (N_h/K) - 1$, przy czym (N_h/K) oznacza dzielenie całkowite,
- obliczyć $c_{n,k} = [p_n - n \cdot N_h]_K + s \cdot K$.



Rys. 9. Widma impulsowych sygnałów UWB otrzymane przy użyciu algorytmu drugiego

Fig. 9. UWB impulse signal spectrum obtained by using the second algorithm

Przykładowe widma sygnałów UWB, otrzymane na drodze symulacji komputerowych, z których wyeliminowano składową o wybranej częstotliwości $f_0 = 333MHz$, przedstawiono na rys. 9, przy czym w symulacjach tych przyjęto następujące wartości parametrów: $K = 3$ i $N_s = 120$ oraz $N_h = 16$ albo $N_h = 18$.

2.5. OBNIŻANIE POZIOMU WIDMA – METODA TRZECIA

Trzecia z opisanych w literaturze [3] metod kształtowania widma sygnału UWB dotyczy obniżania poziomu tego widma dla wybranych częstotliwości. Uzyskiwane tłumienie wybranej częstotliwości jest skończone i w praktyce ograniczone do około 30dB. Jednakże, w odróżnieniu od metod opisanych powyżej, wytłumiane częstotliwości mogą być dowolnie wybierane z przedziału $0 < f_0 < 1/(2T_c)$.

W metodzie tej, gdy mamy ustalone wartości kolejnych b elementów ciągu $\{c_{n,k}\}$, dla $0 \le n \le b-1$, wartość następnego elementu $c_{b,k}$ wybiera się spośród wartości $0 \le c_{b,k} \le N_h - 1$ w taki sposób, aby zminimalizować poniższą sumę:

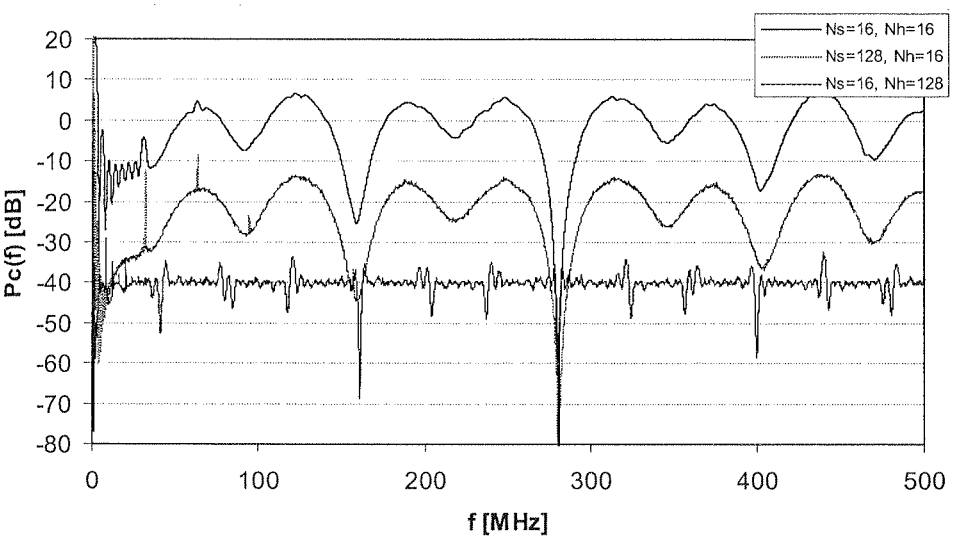
$$S = \left| \sum_{n=0}^{b} e^{-j2\pi \cdot f_0 \cdot (c_{n,k} \cdot T_c + n \cdot N_h \cdot T_c)} \right|. \tag{32}$$

Algorytm tej metody może być zapisany w poniższy sposób:

- dla $n = 0, 1, \ldots, N_s - 1$, przy $([n]_2 = 0)$, wybrać $c_{n,k}$ losowo spośród $0 \le c_{n,k} \le N_h - 1$,

- jeśli $([n]_2 = 1)$, wybrać takie $c_{n,k}$, które minimalizuje sumę (32).

Przykładowe widma sygnałów UWB otrzymane na drodze symulacji komputerowych, w których obniżono poziom widma przy częstotliwości $f_0 = 280MHz$, przedstawiono na rys. 10.



Rys. 10. Widma impulsowych sygnałów UWB otrzymane przy użyciu algorytmu trzeciego

Fig. 10. UWB impulse signal spectrum obtained by using the third algorithm

3. METODY ZMODYFIKOWANE – BADANIE SYMULACYJNE

Przedstawione poniżej modyfikacje metod kształtowania widma impulsowego sygnału UWB pozwalają na usuwanie z jego zawartości niekorzystnych składników i wygładzanie zaburzeń kształtu widma wokół usuwanych składników, czego bezpośrednią miarą jest zawężenie szerokości wytłumianego pasma.

Opisane rozwiązania mogą posłużyć do implementacji adaptacyjnego wytłumiania składowych widma w celu zminimalizowania możliwości wzajemnych interferencji z pracującymi wąskopasmowymi systemami łączności radiowej, co stanowi ważne zagadnienie z dziedziny międzysystemowej kompatybilności elektromagnetycznej.

3.1. ALGORYTM PIERWSZY

Na podstawie tego, co napisano i zobrazowano w p. 2.4.1. oraz spostrzeżeń wynikających z wykonanych badań symulacyjnych, można zauważyć, że regularność w

rozmieszczeniu impulsów, których położenie czasowe jest dobierane losowo lub z zachowaniem warunku (30), nie wpływa na kształt widma w najbliższym sąsiedztwie wytłumianej częstotliwości f_0, jednakże istotnie wpływa na kształt tego widma poza tym sąsiedztwem.

Mając to na uwadze, w pierwszym kroku proponuje się następującą modyfikację metody pierwszej opisanej w p. 2.4.1, polegającą na przypadkowym rozmieszczeniu w ciągu $\{c_{n,k}\}$ elementów losowanych oraz obliczanych przy spełnieniu warunku wytłumiania częstotliwości f_0 (por. (30)). Wówczas warunek ten przyjmuje postać:

$$[c_{n_2,k}]_K = \left[c_{n_1,k} - (n_2 - n_1) \cdot N_h - \frac{K}{2}\right]_K, \tag{33}$$

przy czym $n_1, n_2 \in \{0, \ldots, N_h - 1\} \cap n_1 \neq n_2$.

Przy takiej modyfikacji algorytm postępowania przedstawia się następująco:

- wygenerować wektor $[p_0, \ldots, p_{N,-1}]$, będący losową permutacją liczb od 0 do $N_s - 1$,
- dla $n = 0, 1, \ldots, \frac{N_s}{2} - 1$ wylosować $c_{p_n,k}$ spośród liczb $\{0, \ldots, n_h - 1\}$,
- dla $n = \frac{N_s}{2}, \ldots, N_s - 1$ wylosować liczbę s spośród wartości $\left\{0, \ldots, \frac{N_s}{K} - 1\right\}$ i obliczyć:

$$c_{p_n,k} = \left[c_{k,p_{n-N_s/2}} - (p_n - p_{n-N_s/2}) \cdot N_h - \frac{K}{2}\right]_K + s \cdot K. \tag{34}$$

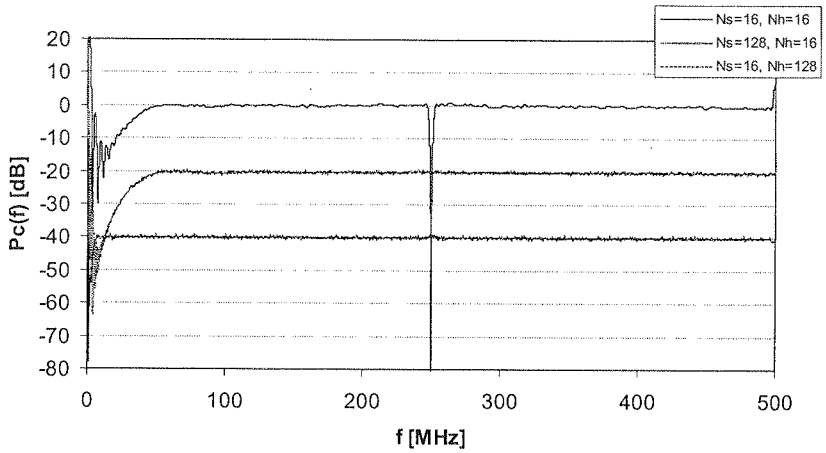
Na rys. 11 przedstawiono otrzymane na drodze symulacji komputerowej widma sygnałów UWB o wartościach parametrów dobranych tak samo jak w przykładach zilustrowanych na rys. 8. Jak można zauważyć, zastosowana modyfikacja w sposobie doboru wartości elementów ciągu $\{c_{n,k}\}$ skutkuje wyrównaniem kształtu widma poza wytłumianą częstotliwością oraz znacznym zawężeniem wytłumianego wraz z tą częstotliwością pasma. W pierwszym przypadku, a więc dla metody niezmodyfikowanej szerokość wytłumionego pasma (na poziomie –10 dB) wynosi około $262/N_h$ MHz, natomiast w przypadku metody zmodyfikowanej szerokość ta zmalała do wartości około $43/N_h$ MHz.

Pozostałe cechy charakterystyczne widma pozostają niezmienione, a w szczególności możliwe jest:

- wytłumienie składowych o częstotliwościach $f_h = (2h + 1) \cdot f_0$, dla dowolnego naturalnego h, czemu towarzyszy
- podwyższenie składowych o częstotliwościach $f_h = 2h \cdot f_0$, także dla dowolnego naturalnego h.

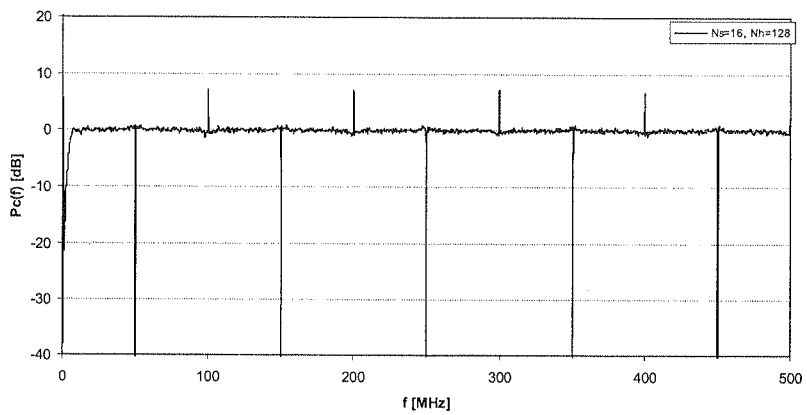
Powyższe właściwości wynikają z grupowania parami elementów ciągu $\{c_{n,k}\}$.

Przykładowe widmo impulsowego sygnału UWB z wytłumionymi i podwyższonymi jego składnikami pokazano na rys. 12.



Rys. 11. Widma impulsowych sygnałów UWB otrzymane przy użyciu zmodyfikowanego algorytmu pierwszego

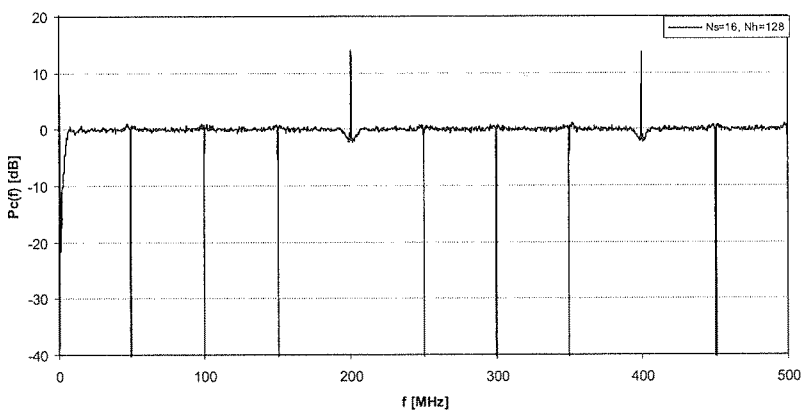
Fig. 11. UWB impulse signal spectrum obtained by using modified the first algorithm



Rys. 12. Wzmocnienie i wytłumienie składników w widmie sygnału UWB uzyskanego przy użyciu zmodyfikowanego algorytmu pierwszego

Fig. 12. Amplification and attenuation frequency components in UWB signal spectrum obtained by using modified the first algorithm

Obok tego, co napisano powyżej, zachodzi także możliwość dalszego wytłumiania niektórych składowych widma poprzez grupowanie elementów ciągu $\{c_{n,k}\}$ w zestawy po cztery elementy i stosowanie metody eliminacji opisanej zależnością (34), dla dwóch powiązanych ze sobą częstotliwości. Tzn., przykładowo dla pary $c_{n1,k}$ i $c_{n2,k}$ można uzyskać eliminację f_0, zaś dla par $c_{n1,k}$ i $c_{n3,k}$ oraz $c_{n2,k}$ i $c_{n4,k}$ eliminację $2f_0$. Powoduje to jednak dalsze podwyższanie poziomu widma dla składowych nie wytłumianych w obu poprzednich krokach. Przykład widma z dodatkowo w ten sposób wytłumionymi jego składowymi pokazano na rys. 13.



Rys. 13. Kształt widma sygnału przy jednoczesnej eliminacji dwóch składników w widmie sygnału UWB przy użyciu zmodyfikowanego algorytmu pierwszego

Fig. 13. Spectrum shape during simultaneous reduction of two components in UWB spectrum obtained by using modified the first algorithm

Wyniki wykonanych badań symulacyjnych, co do dalszego grupowania z kolei czwórek elementów ciągu $\{c_{n,k}\}$ pokazują, że jest to niecelowe. Zaobserwowano wówczas zmniejszenie proporcji pomiędzy liczbą elementów losowanych do obliczanych, co pogarsza właściwości statystyczne ciągu pseudoprzypadkowego.

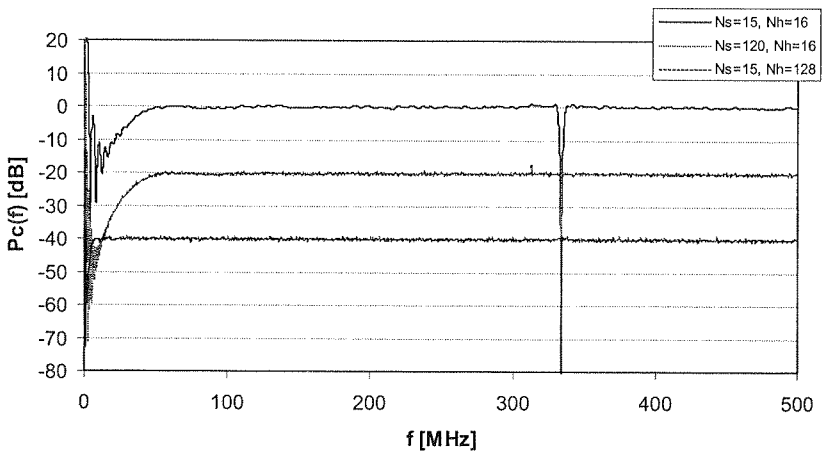
3.2. ALGORYTM DRUGI

Z kolei na podstawie tego, co napisano i zobrazowano w p. 2.4.2., oraz spostrzeżeń wynikających z wykonanych badań symulacyjnych, można zauważyć, że możliwe jest wyrównywanie kształtu widma sygnału UWB poza bezpośrednim sąsiedztwem wytłumianej częstotliwości. Wymaga to zagwarantowania, niezależnie od wartości N_h — jednakże spełniającej warunek $K \leq N_h$, aby jednakowe było prawdopodobieństwo występowania każdej z wartości ciągu $\{c_{n,k}\}$, spośród liczb $\{0, \ldots, N_h - 1\}$.

Zatem, mając dodatkowo także na uwadze to, że usunięcie regularności w podziale N_s wektorów na grupy o liczebności K powoduje zmniejszenie szerokości pasma wytłumianych częstotliwości, proponuje się następujący algorytm zmodyfikowanej *metody drugiej*:

- wygenerować wektor $[p_0, \ldots, p_{N_s-1}]$, będący losową permutacją liczb od 0 do $N_s - 1$,
- jeśli $[p_n - n \cdot N_h]_K < [N_h]_K$ to wylosować liczbę s z przedziału $0 \le s \le (N_h/K)$, w przeciwnym razie wylosować s z przedziału $0 \le s \le (N_h/K) - 1$,
- obliczyć $c_{n,k} = [p_n - n \cdot N_h]_K + s \cdot K$.

Przykład otrzymanego w taki sposób widma przedstawiono na rys. 14. W widmie tym wytłumieniu podlegają częstotliwości $f_h = h/(K \cdot T_c)$, dla naturalnych h. W tym przypadku szerokość wytłumionego pasma częstotliwości wynosi około $47/N_h$ MHz.



Rys. 14. Widma impulsowych sygnałów UWB otrzymane przy użyciu zmodyfikowanego algorytmu drugiego

Fig. 14. UWB impulse signal spectrum obtained by using modified the second algorithm

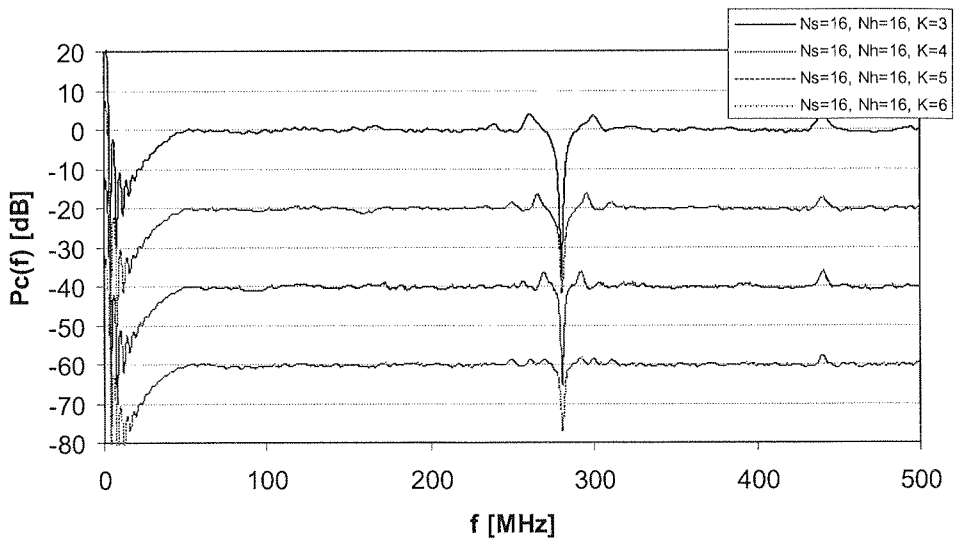
3.3. ALGORYTM TRZECI

Analizując natomiast co napisano i zobrazowano w p. 2.5 oraz spostrzeżenia wynikające z wykonanych badań symulacyjnych, można zauważyć, że duża nierównomierność przebiegu widma poza wytłumioną częstotliwością jest skutkiem stosowania w ciągu rozpraszającym $\{c_{n,k}\}$ naprzemiennie elementów o wartościach losowanych oraz obliczanych pod kątem zminimalizowania sumy (32). Efekt ten można usunąć poprzez modyfikację *metody trzeciej* w taki sposób, aby możliwe było ustalenie stosunku liczby elementów o wartościach losowanych do liczby elementów o wartościach obliczanych. Proponowany w tym przypadku algorytm postępowania przedstawia się następująco:

- dla $m = 0, 1, \ldots, N_s - 1$, przy $([m]_K < K - 1) \cap (m < N_s - 1)$, wybrać $\{c_{m,k}\}$ losowo spośród $0 \leq c_{m,k} \leq N_h - 1$,
- w przeciwnym razie wybrać takie $\{c_{m,k}\}$, które minimalizuje sumę (32).

W algorytmie tym wartość parametru K może być ustalona dowolnie z zakresu od 2 do kilkunastu. Przy powyższym sposobie generacji ciągu rozpraszającego występuje jeden obliczany element tego ciągu na nie więcej niż $K - 1$ elementów losowych.

Przykładowe widma impulsowych sygnałów UWB wygenerowanych przy zastosowaniu powyższego algorytmu pokazano na rys. 15, dla $N_s = N_h = 16$ oraz $f_0 = 280MHz$.



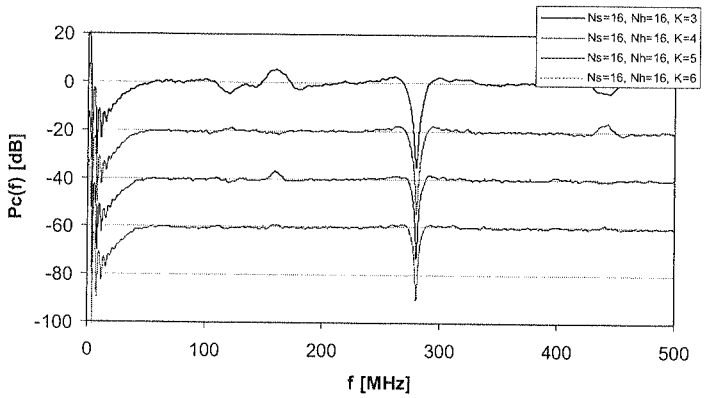
Rys. 15. Widma impulsowych sygnałów UWB otrzymane przy użyciu zmodyfikowanego algorytmu trzeciego

Fig. 15. UWB impulse signal spectrum obtained by using modified the third algorithm

Jak tego można było oczekiwać, zwiększanie wartości parametru K pociąga za sobą zmniejszenie wielkości stłumienia składowej widma o wybranej częstotliwości f_0, przykładowo z $38dB$ przy $K = 3$ do $16dB$ przy $K = 6$. Częściowym rozwiązaniem tego problemu może być rezygnacja z przyjętego ograniczenia, że tylko ostatni z grupy K elementów ciągu $\{c_{n,k}\}$ jest dobierany pod kątem zminimalizowania sumy (32).

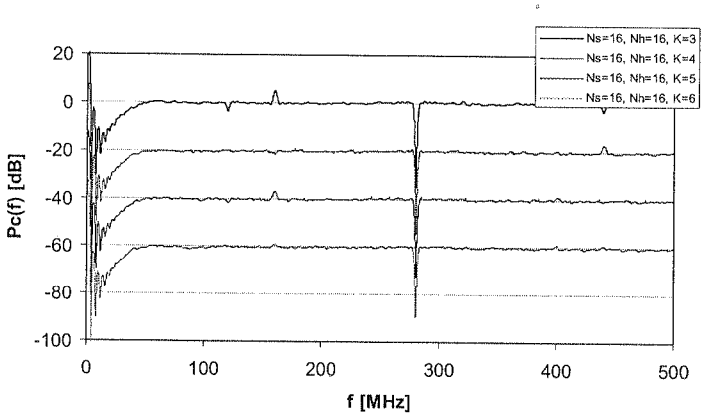
Korygując opisany algorytm postępowania tak, aby wszystkie elementy ciągu $\{c_{n,k}\}$ były dobierane najpierw w sposób pseudoprzypadkowy, zaś następnie w poszukiwaniu najlepszej sekwencji $K-1$ elementów losowych i jednego elementu obliczonego, dokonywane jest podstawianie za każdy z elementów losowych każdej możliwej wartości z przedziału od 0 do $N_h - 1$. Skutek takiego podejścia przedstawiono na rys. 16. Wyniki wykonanych symulacji pokazują, że stłumienie wybranej częstotliwości jest na pozio-

mie około 35dB, niezależnie od wartości parametru K, zaś nierównomierności widma wynoszą około 4,5dB. Wartości te nie ulegają zmianie przy modyfikacji algorytmu mającej na celu zmniejszenie regularności łączenia elementów ciągu $\{c_{n,k}\}$ w grupy, analogicznie jak przy modyfikacji metody pierwszej i drugiej, istotnemu zawężeniu ulega jednak szerokość pasma wytłumianych częstotliwości, jak pokazano na rys. 17.



Rys. 16. Widma impulsowych sygnałów UWB otrzymane przy rozszerzeniu zakresu poszukiwania optymalnego ciągu rozpraszającego w zmodyfikowanym algorytmie trzecim

Fig. 16. UWB impulse signal spectrum obtained by expanded search of optimal spreading code in modified the third algorithm



Rys. 17. Widma impulsowych sygnałów UWB otrzymane przy zmniejszeniu regularności ciągu pseudoprzypadkowego w zmodyfikowanym algorytmie trzecim

Fig. 17. UWB impulse signal spectrum obtained by reducing regularity in pseudorandom code in modified the third algorithm

4. PODSUMOWANIE

Opisany impulsowy system UWB stanowi obiecujący sposób transmisji na nieduże odległości strumieni danych o dużych przepływnościach. Może on służyć do tworzenia teleinformatycznych sieci bezprzewodowych bardzo małego zasięgu, np. w obrębie pojedynczych pomieszczeń. Stanowi to alternatywę do intensywnie rozwijanej techniki zwielokrotnionego nadawania i zbiorczego odbioru, znanej pod skrótową nazwą MIMO (*Multiple Input Multiple Output*) [14]. Każda z tych technik służy do uzyskania dużej przepustowości bezprzewodowego kanału transmisyjnego, w celu sprostania rosnącemu w tym zakresie zapotrzebowaniu.

Poruszony w artykule problem kształtowania widma ma duże znaczenie praktyczne, z punktu widzenia sprostania określonym wymogom kompatybilności elektromagnetycznej pomiędzy klasycznymi systemami wąskopasmowymi a impulsowymi systemami UWB, pracującymi w tych samych lub sąsiadujących ze sobą zakresach częstotliwości. Opracowane modyfikacje opisanych w literaturze metod pozwalają na bardziej efektywne pod tym względem kształtowanie widma impulsowych sygnałów UWB.

5. BIBLIOGRAFIA

1. M.Z. Win, R.A. Scholtz: *Ultra-Wide Bandwidth Time-Hopping Spread- Spectrum Impulse Radio for Wireless Multiple-Access Communications*, IEEE Trans. on Communications, vol. COM-48, no. 4, 2000, pp. 679-689.
2. M.Z. Win, R.A. Scholtz: *Comparisons of Analog and Digital Impulse Radio for Wireless Multiple-Access Communications*, Proc. of the IEEE International Conference on Communiocations, vol. 1, June 1997, pp. 91-95.
3. L. Piazzo, J. Romme: *Spectrum Control by Means of the TH Code in UWB Systems*, Proc. of the IEEE Vehicular Technology Conf., Spring 2003, pp. 1649-1653.
4. L. Piazzo, J. Romme: *On the Power Spectral Density of Time-Hopping Impulse Radio*, Proc. of the IEEE Ultra Wideband Systems and Technologies Conf., May 2002, pp. 241-244.
5. FCC CFR 47 Part 15: *Radio Frequency Devices*, Code of Federal Regulations Title 47, chapter 1, part 15.
6. C.A. Corral, S. Sibecas, S. Emami, G. Stratis: *Pulse Spectrum Optimization for Ultra-Wideband Communication*, Proc. of the IEEE Conferece on Ultra Wideband Systems and Technologies 2002, pp. 31-35.
7. A.Parr, B. Cho, Z. Ding: *A New UWB Pulse Generator for FCC Spectral Masks*, Proc. of the IEEE Vehicular Technology Conf., Spring 2003, pp. 1664-1666.
8. A. Parr, B. Cho, K. Wallace, Z. Ding: *A Novel Ultra-Wideband Pulse Design Algorithm*, IEEE Communications Letters, vol. 7, no. 5, may 2003, pp. 219-221.
9. G. Durisi, S. Benedetto: *Performance Evaluation of TH-PPM UWB Systems in the Presence of Multiuser Interference*, IEEE Communications Letters, vol. 7, no. 5, 2003, pp. 224-226.
10. S. Licul, J. Noronha, W. Davis, D. Sweeney, C. Anderson, T. Bielawa: *A Parametric Study of Time-Domain Characteristics of Possible UWB Antenna Architectures*, Proc. of the IEEE Vehicular Technology Conf., Fall 2003, pp. 3110-3114.
11. K. Taniguchi, R. Kohno: *Design and Analysis of Template Waveform for Receiving UWB Signals*, Proc. of the International Workshop on Ultra Wideband Systems 2004, pp. 125-129.

12. R.J. K a t u l s k i, J. S a d o w s k i: *Kształtowanie widma sygnału UWB*, mat. Krajowego Sympozjum Telekomunikacji - KST, Bydgoszcz, 2005, ss. 327-336.
13. S.S. M o, A.D. G e l m a n: *On the Power Spectral Density of UWB Signals in IEEE 802.15.3a*, Proc. of the IEEE Wireless Communications and Networking Conference 2004, vol. 2. pp. 999- 1002.
14. R.J. K a t u l s k i, M. M i k o ł a j s k i: *Technika zwielokrotnionego nadawania i zbiorczego odbioru w telekomunikacji bezprzewodowej*, Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne, vol. LXXVIII, nr 2-3, 2005, ss. 55-59.

R. J. KATULSKI, J. SADOWSKI

ULTRA WIDE BAND (UWB) IMPULSE RADIO

S u m m a r y

The UWB impulse radio concept for WPAN applications is presented. The impulse signal time structures and the UWB radio receivers have been described. A problem of the impulse signal spectrum from the electromagnetic compatibility point of view has been characterized. The signal spectrum forming methods are analyzed. The modified form of the methods have been proposed. The results of investigations of the spectrum shape obtained by simulation way are presented. In conclusion, the practical aspects of the UWB impulse radio are described.

Keywords: ultra wide band (UWB) systems, impulse radio (IR), wireless personal area network (WPAN), spectrum forming, simulation investigations

maga
użyte
stawi
zentu
czas

Słow

Zaaw
czas szcz
reprezentu
sowej rep
addytywn
sji przez
oraz addy
tłumienie
łu propag
sygnału u

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 617–628

STATYSTYKI DRUGIEGO RZĘDU KANAŁU RAYLEIGHA ORAZ NAKAGAMIEGO

KRYSTYNA MARIA NOGA

Akademia Morska w Gdyni, Katedra Automatyki Okrętowej
81-225 Gdynia, ul. Morska 83,
e-mail: jagat@am.gdynia.pl

Otrzymano 2006.01.05
Autoryzowano 2006.04.27

Zagadnienia związane z przebiegiem czasowym obwiedni i parametrów zaników wymagają wyznaczenia statystyk rzędu drugiego. W artykule przedstawiono statystyki sygnału użytecznego aktualnie stosowane do opisu kanału radiokomunikacyjnego z zanikami. Przedstawiono średnią częstość zaników, czyli średnią częstość przejść zmiennej losowej reprezentującej obwiednię sygnału użytecznego przez określoną wartość progową, oraz średni czas trwania zaników w kanale Rayleigha oraz Nakagamiego.

Słowa kluczowe: radiokomunikacja, kanał Nakagamiego, kanał Rayleigha, zaniki, średnia częstość zaników, średni czas trwania zaników.

1. WSTĘP

Zaawansowane modele sygnałów radiowych częściowo rozproszonych są dotychczas szczegółowo opisane w zakresie rozkładów rzędu pierwszego zmiennej losowej reprezentującej obwiednię sygnału użytecznego w kanale z zanikami lub zmiennej losowej reprezentującej chwilowy stosunek mocy sygnału użytecznego do mocy szumu addytywnego (ang. SNR — *signal-to-noise ratio*) [1]. Sygnał nadany podczas transmisji przez kanał radiokomunikacyjny podlega losowym zakłóceniom multiplikatywnym oraz addytywnym. Wpływ zakłóceń multiplikatywnych traktujemy jako przypadkowe tłumienie sygnału; zjawisko to określamy jako zaniki. Dla analogowego modelu kanału propagacyjnego sygnał $y(t)$ odbierany przez odbiornik należy traktować jako sumę sygnału użytecznego $s(t)$ oraz zakłócenia addytywnego $n(t)$, czyli

$$y(t) = s(t) + n(t) = r(t)cos[2\pi f_0 t + \phi(t)] + n(t) \tag{1}$$

gdzie $r(t) \geq 0$ jest obwiednią sygnału użytecznego, f_0 jest częstotliwością nośnej, $\phi(t) \in [-\pi, \pi)$ jest fazą chwilową sygnału użytecznego. Z analizy literatury [1, 2] wynika, że do opisu zakłóceń addytywnych stosuje się szum gaussowski o zerowej wartości średniej i widmowej gęstości mocy N_0, przy czym zakłada się, że szum jest niezależny od zaników. Natomiast do opisu zaników stosuje się różne modele probabilistyczne. Szeroką klasę mechanizmów propagacyjnych uwzględnia rozkład Nakagamiego, który ponadto w szczególnych przypadkach staje się rozkładem Rayleigha, jednostronnym normalnym lub rozkładem Rice'a [1, 3]. Zagadnienia związane z przebiegiem czasowym obwiedni i parametrów zaników wymagają wyznaczenia statystyk rzędu drugiego. W artykule zostanie przedstawiony średni czas trwania zaników oraz średnia częstość zaników, czyli średnia częstość przejść zmiennej losowej reprezentującej obwiednię sygnału użytecznego przez określoną wartość progową, w kanale Rayleigha oraz Nakagamiego.

2. ROZKŁAD RAYLEIGHA

Jeżeli sygnał użyteczny $s(t)$ nie zawiera składowej regularnej wówczas obwiednia tego sygnału posiada rozkład Rayleigha o gęstości prawdopodobieństwa

$$p_r(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right); r \geqslant 0 \tag{2}$$

gdzie $2\sigma^2 = E\{r^2\} = a_{sk}^2$ jest wartością średniokwadratową obwiedni sygnału użytecznego, czyli kwadratem amplitudy skutecznej, przy czym $E\{\}$ oznacza operator uśredniania zmiennej losowej. Wartość średnia i wariancja obwiedni o rozkładzie Rayleigha wyrażone są odpowiednio wzorami

$$\mathrm{E}\{r\} = \sqrt{\frac{\pi}{2}}\sigma \tag{3a}$$

$$\mathrm{var}\{r\} = \sigma_r^2 = \left(2 - \frac{\pi}{2}\right)\sigma^2 \tag{3b}$$

Faza sygnału użytecznego o obwiedni Rayleigha posiada rozkład równomierny

$$p_\phi(\phi) = \frac{1}{2\pi}; \phi(t) \in [-\pi, \pi) \tag{4}$$

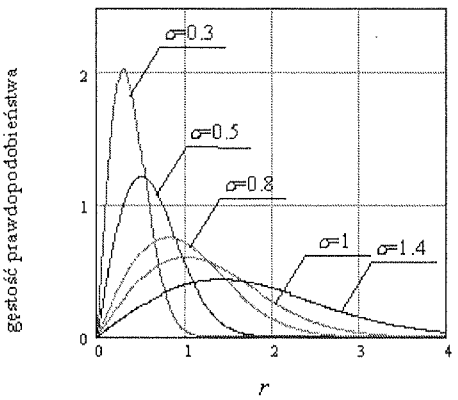
Ponadto dystrybuanta rozkładu Rayleigha jest określona poprzez prawdopodobieństwo

$$\mathrm{P}(r \leqslant a) = 1 - \exp(-\frac{a^2}{2\sigma^2}) \tag{5}$$

Z literatury [2] wynika, że w analizie często stosuje się unormowaną zmienną losową Rayleigha zdefiniowaną jako $k = r/\sigma$, której rozkład można przedstawić przy pomocy zależności

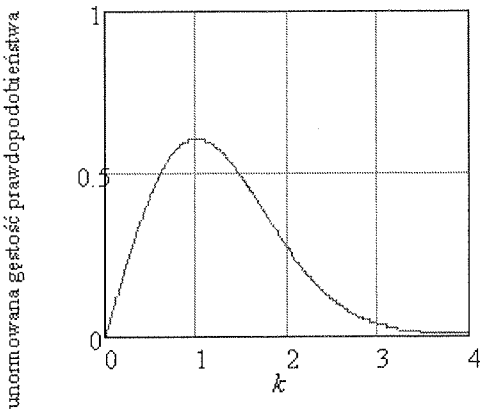
$$p_k(k) = k \exp(-\frac{k^2}{2}) \tag{6}$$

Wykres gęstości prawdopodobieństwa rozkładu Rayleigha przedstawiono na rysunku 1, natomiast jego wersję unormowaną na rysunku 2.



Rys. 1. Gęstość prawdopodobieństwa rozkładu Rayleigha

Fig. 1. Rayleigh probability density function



Rys. 2. Unormowana gęstość prawdopodobieństwa rozkładu Rayleigha

Fig. 2. Normalized Rayleigh probability density function

Do oceny jakości transmisji bardzo często wykorzystuje się parametr kanału SNR zdefiniowany jako $\gamma = r^2/2N_0$. W kanale Rayleigha zmienna losowa reprezentująca γ posiada rozkład wykładniczy o gęstości prawdopodobieństwa

$$p_\gamma(\gamma) = \frac{1}{\overline{\gamma}}\exp\left(-\frac{\gamma}{\overline{\gamma}}\right); \gamma \geqslant 0 \tag{7}$$

gdzie $\overline{\gamma} = \mathrm{E}\{\gamma\}$ jest średnim stosunkiem mocy sygnału użytecznego do mocy zakłócenia addytywnego. Ponadto moment k–tego rzędu zmiennej losowej reprezentującej γ w kanale Rayleigha określa zależność

$$\mathrm{E}\left\{\gamma^k\right\} = \Gamma(1+k)(\overline{\gamma})^k; k = 1, 2, ... \tag{8}$$

gdzie $\Gamma(m) = \int_0^\infty t^{m-1}\exp(-t)\mathrm{d}t$ jest funkcją gamma [4].

Do opisu zaników oraz do oceny jakości transmisji w kanale z zanikami stosuje się funkcję tworzącą momenty (ang. MGF — *moment generating function*). Funkcja ta jest wygodnym narzędziem do obliczania momentów i ich pochodnych w zerze przy

pomocy tablic transformat Laplace'a. Dla zmiennej losowej reprezentującej γ funkcja tworząca momenty określona jest jako [1]

$$M_\gamma(s) = \int_0^\infty p_\gamma(\gamma) \exp(s\gamma)\, d\gamma \tag{9}$$

W kanale z zanikami Rayleigha funkcję tworzącą momenty dla zmiennej losowej reprezentującej γ można obliczyć na podstawie zależności

$$M_\gamma(s) = \frac{1}{(1 - s\overline{\gamma})} \tag{10}$$

W analizie propagacji w kanale radiowym często zakłada się, że zmienne losowe reprezentujące obwiednię sygnału pasmowego o normalnym rozkładzie wartości chwilowej i zerowej wartości średniej oraz obwiednię sygnału radiowego rozproszonego posiadają rozkłady Rayleigha. Rozkłady te są, między innymi, stosowane do opisu zjawiska propagacji fal w systemach komórkowych pracujących w obszarach miejskich lub górzystych.

3. ROZKŁAD NAKAGAMIEGO

Do opisu obwiedni sygnału użytecznego $s(t)$ często stosuje się także rozkład Nakagamiego o gęstości prawdopodobieństwa

$$p_r(r) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m r^{2m-1} \exp\left(-\frac{m}{\Omega} r^2\right);\ r \geq 0 \tag{11}$$

gdzie: $\Omega = E\{r^2\}$, zaś $m = [\Omega^2/E[(r^2 - \Omega)^2] \geq 1/2$ jest odwrotnością unormowanej wariancji kwadratu obwiedni sygnału użytecznego. Jednocześnie parametr m charakteryzuje głębokość zaników (ang. AF — *amplitude fading*). W kanale tym AF = $1/m$.

Wykresy gęstości rozkładu Nakagamiego dla różnych wartości parametrów m i Ω zostały przedstawione na rysunku 3.

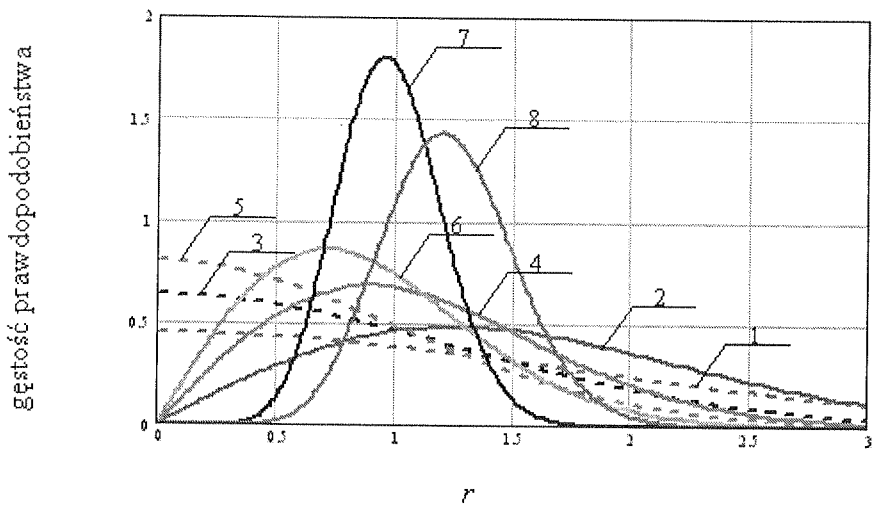
Niech q jest unormowaną zmienną losową określoną jako $q = r\sqrt{2/\Omega}$. Zmienna ta posiada unormowany rozkład Nakagamiego o gęstości prawdopodobieństwa

$$p_q(q) = \frac{m^m}{2^{m-1}} q^{2m-1} \exp\left(-\frac{mq^2}{2}\right) \tag{12}$$

Wykresy gęstości unormowanego rozkładu Nakagamiego zostały przedstawione na rysunku 4 dla różnych wartości parametru m.

Sygnał użyteczny $s(t)$ o rozkładzie obwiedni Nakagamiego posiada następujący moment k–tego rzędu

$$E\{r^k\} = \frac{\Gamma\left(m + \frac{k}{2}\right)}{\Gamma(m)} \sqrt{\left(\frac{\Omega}{m}\right)^k} \tag{13}$$



Rys. 3. Wykresy gęstości prawdopodobieństwa rozkładu Nakagamiego; parametry rozkładu wynoszą odpowiednio: 1) $m = 0,5$, $\Omega = 5$ dB, 2) $m = 1$, $\Omega = 5$ dB, 3) $m = 0,5$, $\Omega = 2$ dB, 4) $m = 1$, $\Omega = 2$ dB, 5) $m = 0,5$, $\Omega = 0$ dB, 6) $m = 1$, $\Omega = 0$ dB, 7) $m = 5$, $\Omega = 0$ dB, 8) $m = 5$, $\Omega = 2$ dB

Fig. 3. Nakagami probability density function, the parameters properly equal: 1) $m = 0,5$, $\Omega = 5$ dB, 2) $m = 1$, $\Omega = 5$ dB, 3) $m = 0,5$, $\Omega = 2$ dB, 4) $m = 1$, $\Omega = 2$ dB, 5) $m = 0,5$, $\Omega = 0$ dB, 6) $m = 1$, $\Omega = 0$ dB, 7) $m = 5$, $\Omega = 0$ dB, 8) $m = 5$, $\Omega = 2$ dB

W szczególności wartość średnia oraz wariancja wynoszą odpowiednio

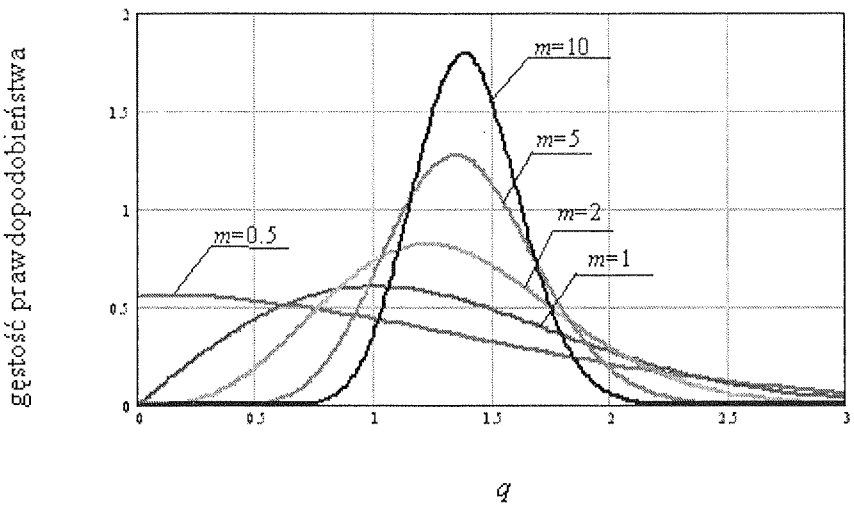
$$E\{r\} = \frac{\Gamma\left(m + \frac{1}{2}\right)}{\Gamma(m)} \sqrt{\frac{\Omega}{m}} \tag{14a}$$

$$\operatorname{var}\{r\} = \sigma_r^2 = \Omega\left[1 - \frac{\Gamma^2\left(m + \frac{1}{2}\right)}{m\Gamma^2(m)}\right] \tag{14b}$$

Ponadto dystrybuantę rozkładu Nakagamiego określa prawdopodobieństwo

$$\mathrm{P}(r \leqslant a) = \frac{\Gamma\left(m, \frac{ma^2}{\Omega}\right)}{\Gamma(m)} \tag{15}$$

gdzie $\Gamma(m, x) = \int_0^x t^{m-1}\exp(-t)\mathrm{d}t$ jest komplementarną niepełną funkcją gamma [4], przy czym dla m należących do zbioru liczb naturalnych spełniona jest zależność $\gamma(m, x) = (m-1)!\,\exp(-x)\sum_{k=m}^{\infty}\frac{1}{k!}x^k$.



Rys. 4. Wykresy gęstości prawdopodobieństwa unormowanego rozkładu Nakagamiego

Fig. 4. Normalized Nakagami probability density function

W kanale Nakagamiego zmienna losowa reprezentująca γ posiada rozkład gamma o gęstości prawdopodobieństwa

$$p_\gamma(\gamma) = \frac{m^m \gamma^{m-1}}{\Gamma(m)\,\overline{\gamma}^m} \exp\left(-\frac{m\gamma}{\overline{\gamma}}\right); \gamma \geqslant 0 \tag{16}$$

Funkcja tworząca momenty w kanale Nakagamiego określona jest jako [1]

$$M_\gamma(s) = \left(1 - \frac{s\overline{\gamma}}{m}\right)^{-m} \tag{17}$$

W kanale z zanikami miarą jakości transmisji jest między innymi średnie prawdopodobieństwo błędu elementowego, które dla odbioru niekoherentnego (ang. NC — *noncoherent*) można obliczyć na podstawie zależności [1]

$$\mathrm{P_D}(*) = \int_0^\infty P_s(\gamma) p(\gamma)\, d\gamma = \frac{1}{2} M_\gamma(-\alpha) \tag{18}$$

gdzie $P_s(\gamma)$ jest prawdopodobieństwem błędu elementowego w kanale bez zaników, $\alpha = 0,5$ dla odbioru NCFSK (ang. *noncoherent frequency shift keying*) oraz $\alpha = 1$ dla DPSK (ang. *differentially phase shift keying*). Ponadto moment k-tego rzędu zmiennej losowej reprezentującej SNR γ można określić na podstawie wzoru [1]

$$E\left\{\gamma^k\right\} = \frac{\Gamma(m+k)}{\Gamma(m)\,m^k} \overline{\gamma}^k \tag{19}$$

Rozkład Nakagamiego dobrze aproksymuje inne rozkłady, np. Rice'a, czteroparametrowy, Hoyta [1, 3, 5]. W szczególności, gdy $m = 0.5$ otrzymujemy rozkład jednostronny normalny oraz gdy $m = 1$ rozkład Rayleigha. Natomiast dla $m \to \infty$ mamy kanał bez zaników, czyli tylko z addytywnym białym szumem gaussowskim. Rozkład ten uwzględnia więc szeroką klasę zaników. Przykłady zastosowania rozkładu Nakagamiego do opisu sygnałów z zanikami oraz do oceny jakości transmisji przedstawiono w [5, 6].

4. ŚREDNIA CZĘSTOŚĆ ZANIKÓW

Do opisu przejść sygnałów losowych przez określony poziom niezbędne jest zastosowanie statystyk drugiego rzędu. Średnią częstość $\overline{N}\left(r = R_p\right)$ jednokierunkowych przejść zmiennej losowej reprezentującej obwiednię sygnału użytecznego w kanale z zanikami przez określoną wartość progową (ang. LCR — *level crossing rate*) można obliczyć na podstawie zależności [2, 6, 7, 8]

$$\overline{N}(r = R_p) = \int_0^\infty |\dot{r}|\, p_{r\dot{r}}(r = R_p, \dot{r})\mathrm{d}\,\dot{r} \tag{20}$$

przy czym $\dot{r} = dr/dt$ jest pochodną obwiedni, $p_{r\dot{r}}(r, \dot{r})$ jest łączną gęstością prawdopodobieństwa obwiedni sygnału i jej pochodnej oraz R_p jest wartością progową obwiedni. Zmienne losowe reprezentujące r oraz $\dot{r}$ są niezależne [7, 8], dlatego też łączna gęstość prawdopodobieństwa $p_{r,\dot{r}}\left(r = R_p, \dot{r}\right) = p_{\dot{r}}\left(\dot{r}\,|r = R_p\right) p_r(r) = p_{\dot{r}}\left(\dot{r}\right) p_r(r)$.

Dla obwiedni o rozkładzie Nakagamiego średnia częstość zaników poniżej R_p jest określona wzorem

$$\overline{N}(r = R_p) = \sqrt{2\pi} f_m \frac{m^{m-1/2}}{\Gamma(m)} \left(\frac{R_p^2}{\Omega}\right)^{m-1/2} \exp(-\frac{mR_p^2}{\Omega}) \tag{21}$$

gdzie $f_m = v/\lambda$ jest maksymalnym dopplerowskim przesunięciem częstotliwości przy odbiorze fali pojedynczej, v jest składową prędkości ruchu w kierunku zgodnym z trajektorią propagacji, λ jest długością fali radiowej.

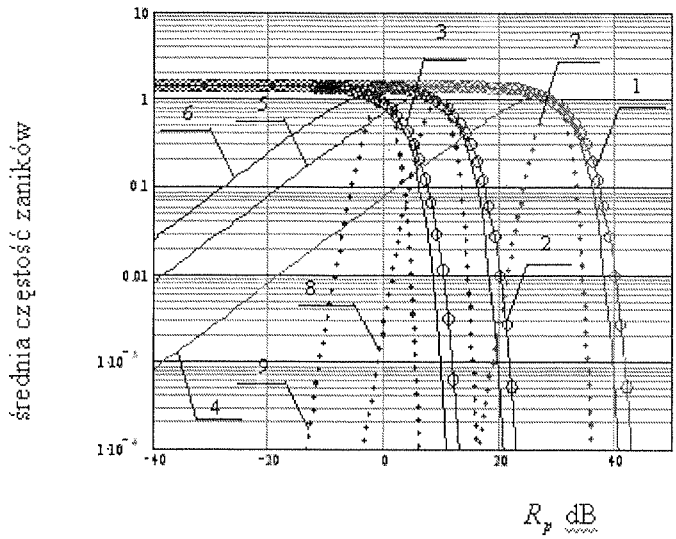
Unormowany wykres średniej częstości zaników poniżej R_p w kanale Nakagamiego, czyli $\overline{N}(r = R_p)/f_m$, został przedstawiony na rysunku 5.

Wartość R_p^2 najczęściej mierzymy względem wartości średniokwadratowej obwiedni. Niech β^2 oznacza kwadrat unormowanego progu zgodnie z zależnością

$$\beta^2 = \frac{R_p^2}{\mathrm{E}\{r^2\}} = \frac{R_p^2}{\Omega} \tag{22}$$

Wówczas średnią częstość zaników w kanale Nakagamiego można zapisać jako [7]

$$\overline{N}(\frac{R_p}{\sqrt{\Omega}} = \beta) = \sqrt{2\pi} f_m \frac{m^{m-1/2}}{\Gamma(m)} \beta^{2m-1} \exp(-m\beta^2) \tag{23}$$



Rys. 5. Unormowany wykres średniej częstości zaników w kanale Nakagamiego poniżej progu R_p; parametry wynoszą odpowiednio:1) $m = 0,5$, $\Omega = 30$ dB; 2) $m = 0,5$, $\Omega = 10$ dB; 3) $m = 0,5$, $\Omega = 0,1$ dB, 4) $m = 1$, $\Omega = 30$ dB; 5) $m = 1$, $\Omega = 10$ dB 6) $m = 1$, $\Omega = 0,1$ dB 7) $m = 5$, $\Omega = 30$ dB; 8) $m = 5$, $\Omega = 10$ dB; 9) $m = 5$, $\Omega = 0,1$ dB

Fig. 5. Normalized level crossing rate for Nakagami channel; the parameters properly equal: 1) $m = 0,5$, $\Omega = 30$ dB; 2) $m = 0,5$, $\Omega = 10$ dB; 3) $m = 0,5$, $\Omega = 0,1$ dB, 4) $m = 1$, $\Omega = 30$ dB; 5) $m = 1$, $\Omega = 10$ dB 6) $m = 1$, $\Omega = 0,1$ dB 7) $m = 5$, $\Omega = 30$ dB; 8) $m = 5$, $\Omega = 10$ dB; 9) $m = 5$, $\Omega = 0,1$ dB

Dla obwiedni o rozkładzie Rayleigha średnia częstość zaników poniżej R_p jest określona wzorem [2]

$$\overline{N}(r = R_p) = \sqrt{\pi} f_m \frac{R_p}{\sigma} \exp(-\frac{R_p^2}{2\sigma^2}) \tag{24}$$

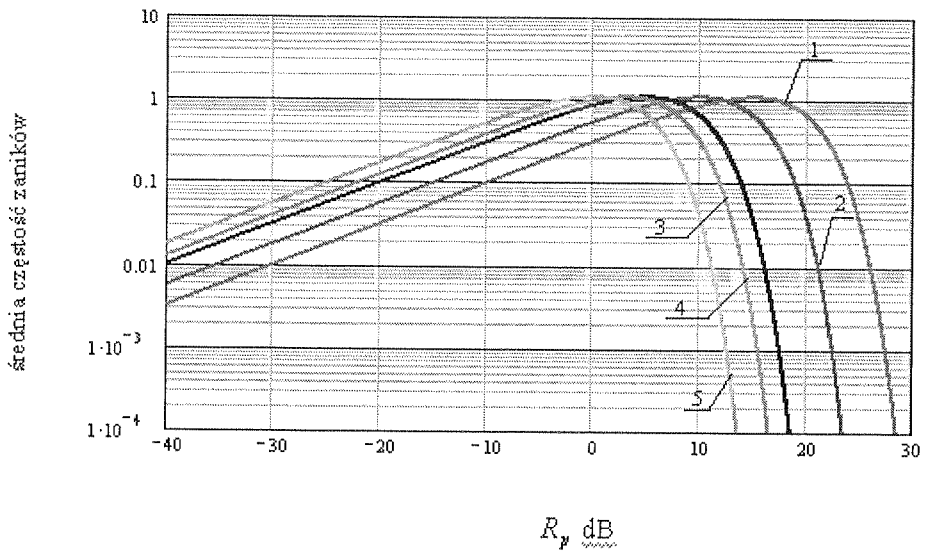
Z zależności (24) wynika, że średnia częstość zaników poniżej określonego progu jest proporcjonalna do gęstości prawdopodobieństwa obwiedni sygnału. W kanale Rayleigha unormowany wykres średniej częstości zaników poniżej R_p, czyli $\overline{N}(r = R_p)/f_m$, został przedstawiony na rysunku 6.

W kanale Rayleigha kwadrat unormowanego progu wynosi

$$\beta^2 = \frac{R_p^2}{a_{sk}^2} = \frac{R_p^2}{2\sigma^2} \tag{25}$$

Wówczas średnią częstość zaników poniżej określonego progu określa zależność [2]

$$\overline{N}\left(\frac{R_p}{a_{sk}} = \beta\right) = \sqrt{2\pi} f_m \beta \exp(-\beta^2) \tag{26}$$



Rys. 6. Unormowany wykres średniej częstości zaników w kanale Rayleigha poniżej określonego progu, parametr σ^2 wynosi odpowiednio: 1) σ^2 = 15 dB 2) σ^2 = 10 dB, 3) σ^2 = 5 dB, 4) σ^2 = 3 dB, 5) σ^2 = 0.1 dB

Fig. 6. Normalized level crossing rate for Rayleigh channel, the parametr σ^2 is equal properly: 1) σ^2 = 15 dB 2) σ^2 = 10 dB, 3) σ^2 = 5 dB, 4) σ^2 = 3 dB, 5) σ^2 = 0.1 dB

5. ŚREDNI CZAS TRWANIA ZANIKÓW

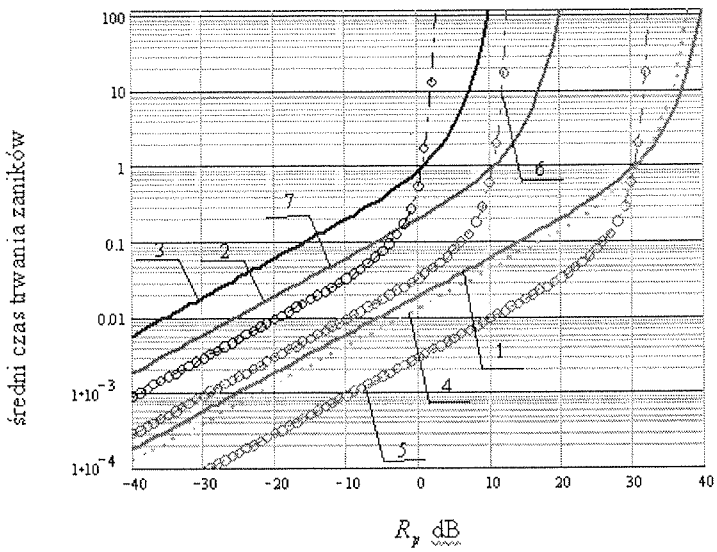
Inną ważną statystyką drugiego rzędu sygnałów rozproszonych jest średni czas trwania $\overline{\tau}$ zaniku obwiedni sygnału poniżej określonej wartości (ang. AFD — *average fade duration*). Pomiędzy $\overline{N}$ i $\overline{\tau}$ istnieje następująca zależność [6, 7]

$$\overline{\tau}\left(r = R_p\right) = \frac{P\left(r \leqslant R_p\right)}{\overline{N}\left(r = R_p\right)} = \frac{1}{\overline{N}\left(r = R_p\right)} \int_0^{R_p} p\,(r)\mathrm{d}r \tag{27}$$

Po odpowiednich przekształceniach średni czas trwania zaniku w kanale Nakagamiego możemy obliczyć na podstawie zależności

$$\overline{\tau}\left(r = R_p\right) = \frac{\Gamma\left(m, m\frac{R_p^2}{\Omega}\right)\exp(\frac{mR_p^2}{\Omega})}{\sqrt{2\pi} f_m \left(\frac{mR_p^2}{\Omega}\right)^{m-1/2}} \tag{28}$$

Na rysunku 7 przedstawiono unormowany średni czas trwania $\overline{\tau}\left(r = R_p\right)$ zaniku obwiedni sygnału poniżej określonej wartości dla kanału Nakagamiego, czyli $\overline{\tau}\left(r = R_p\right) f_m$.



Rys. 7. Unormowany średni czas trwania $\overline{\tau}$ zaniku w kanale Nakagamiego, parametry wynoszą odpowiednio: 1) $m = 0,5$, $\Omega = 30$; 2) $m = 0,5$, $\Omega = 10$; 3) $m = 0,5$, $\Omega = 0,1$; 4) $m = 1$, $\Omega = 30$; 5) $m = 20$, $\Omega = 30$; 6) $m = 20$, $\Omega = 10$, 7) $m = 20$, $\Omega = 0,1$

Fig. 7. Normalized average fade duration in Nakagami channel; the parameters properly equal: 1) $m = 0,5$, $\Omega = 30$; 2) $m = 0,5$, $\Omega = 10$; 3) $m = 0,5$, $\Omega = 0,1$; 4) $m = 1$, $\Omega = 30$; 5) $m = 20$, $\Omega = 30$; 6) $m = 20$, $\Omega = 10$, 7) $m = 20$, $\Omega = 0,1$

Ponadto po podstawieniu do wzoru (28) $m = 1$ otrzymujemy średni czas trwania zaniku w kanale Rayleigha. Dla zaników głębokich, czyli dla $(R_p^2/\Omega) \ll 1$, słuszne jest przybliżenie $1 - \exp\left(-X^2\right) \approx X^2$ [2]. Wówczas dla kanału Rayleigha, czyli gdy $E\left(r^2\right) = 2\sigma^2 = \Omega$, zależność (28) możemy uprościć do postaci [7]

$$\overline{\tau}\left(r = R_p\right) \approx \frac{\sigma\left[\exp\left(\frac{R_p^2}{2\sigma^2}\right) - 1\right]}{\sqrt{\pi} f_m R_p} \tag{29}$$

6. PODSUMOWANIE

Modele sygnałów radiowych są w literaturze polskiej szczegółowo opisane w zakresie rozkładów rzędu pierwszego zmiennej losowej reprezentującej obwiednię sygnału użytecznego oraz zmiennej losowej reprezentującej SNR. Niniejszy artykuł jest próbą syntetycznego i kompleksowego przedstawienia statystyk drugiego rzędu kanału Rayleigha i Nakagamiego.

Przy projektowaniu systemów radiokomunikacyjnych, a przede wszystkim odbiornika i nadajnika, ważną rolę odgrywa analiza obwiedni sygnału użytecznego. Poprzez

odpowiednie zaprojektowanie układów odpowiedzialnych za przeplot i rozplot oraz kodowanie i dekodowanie można zmniejszyć prawdopodobieństwo błędu elementowego, a także skutki błędów seryjnych. Strumień błędów seryjnych został opisany między innymi w pracach [3, 5]. Przedstawione w niniejszym artykule zależności można wykorzystać między innymi do budowy symulatora kanału radiokomunikacyjnego lub do oceny jakości transmisji w kanale z zanikami. Skutecznym sposobem przeciwdziałania skutkom zaników jest zastosowanie odbioru zbiorczego przestrzennego, który w miarę dokładnie został omówiony w pracach [1, 6, 8, 9]. Przedstawione w tym artykule statystyki drugiego rzędu można również wykorzystać do opisu systemu z odbiorem zbiorczym. W literaturze anglojęzycznej taka analiza jest dostępna, np. [8]. Natomiast w literaturze polskiej, zdaniem autorki artykułu, problem ten wymaga omówienia.

7. BIBLIOGRAFIA

1. M.K Simon, M.S. Alouini: *Digital communication over fading channels*, A unified approach to performance analysis. John Wiley Publication, 2005.
2. A. Wojnar: *Teoria sygnałów*, WNT, Warszawa 1988.
3. K. Noga: *Zaawansowane modele probabilistyczne sygnałów w kanałach z zanikami*, Kwartalnik Elektroniki i Telekomunikacji, 2000, Nr. 1, ss. 47–62.
4. S. Gradstein, M. Ryzhik: *Tables of Integrals, Sums, Series and Products*, London, UK., Academic, 1964.
5. K. Noga: *Właściwości strumienia błędów dla transmisji binarnej w kanale z wolnymi zanikami opisanymi rozkładem czteroparametrowym*, Kwartalnik Elektroniki i Telekomunikacji, 1994, Nr. 3, ss. 357-368.
6. C.D. Iskander, P.T. Mathiopoulos: *Analytical level crossing rates and average fade durations for diversity techniques in Nakagami fading channels*, IEEE Transactions on Communications, Vo. 50, No.8, August 2002, pp. 1301-1309.
7. M.D. Yacoub, J.E Vargas Bautista, LG. De Rezende Guedes: *On higher order statistics of the Nakagami — m distribution*, IEEE Transactions on Vehicular Technology, Vol. 48, No. 3, May 1999, pp. 790-794.
8. M.D Yacoub, C.R. Cunha da Silva, J. E. Vargas Bautista: *Second order statistics for diversity combining techniques in Nakagami fading channels*, IEEE Transactions on Vehicular Technology, Vol. 50, No. 6, November 2001, pp. 1464-1470.
9. K. Noga: *The performance of binary transmission in slow Nakagami fading channels with MRC diversity*, IEEE Transaction on Communications, Vol. 46, No. 7, July 1998, pp. 863-865

K. M. NOGA

SECOND ORDER STATISTICS FOR THE RAYLEIGH AND NAKAGAMI CHANNEL

Summary

The performance of wireless systems is affected by fading. Fading is basically caused by multipath propagation. A great number of distributions well describe the statistics of the mobile radio signal. The

Nakagami and Rayleigh distributions have been found to be a very good fitting for the mobile radio channel. The average level crossing rate (LCR) and the average fade duration (AFD) are two quantities that statistically characterize a fading communication channel. The LCR is defined as the average number of times per unit duration that the envelope of a fading signal crosses a given signal threshold R_p The AFD corresponds to the average length of time that the envelope remains under the threshold R_p. In this paper the formulas for LCR and AFD for Nakagami and Rayleigh fading channel are presented.

Keywords: radiocommunications, Nakagami fading channel, Rayleigh fading channel, average fade duration, level crossing rate

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 629–641

Modyfikacja algorytmów transformacji FDCT dla potrzeb kompresji obrazów implementowanej w układach FPGA

AGNIESZKA DĄBROWSKA[1,2], KAZIMIERZ WIATR[1,2]

[1] *Akademia Górniczo-Hutnicza, Katedra Elektroniki,*
al. Mickiewicza 30, 30-059 Kraków,
[2] *Akademickie Centrum Komputerowe CYFRONET AGH,*
ul. Nawojki 11, 30-950 Kraków,
email: adabrow@agh.edu.pl, wiatr@agh.edu.pl

Otrzymano 2005.06.10
Autoryzowano 2005.11.30

Dyskretna transformacja kosinusowa DCT (ang. *Discrete Cosinus Transform*) jest jedną z podstawowych odmian algorytmów kodowania transformatorowego. Jest ona stosowana w standardowych algorytmach kompresji obrazu nieruchomego (JPEG) jak również w algorytmach kompresji obrazów ruchomych (MPEG, H.26x). W przypadku algorytmów kompresji obrazu dyskretnej transformacji kosinusowej poddawane są bloki 8×8 pikseli. W artykule przedstawiono podstawy i główne algorytmy realizacji transformacji DCT, ze szczególnym uwzględnieniem zagadnień dotyczących implementacji w układach programowalnych FPGA (XCV200BG352).

Słowa kluczowe: kompresja obrazów, systemy wizyjne, rekonfigurowalne systemy obliczeniowe, układy programowalne

1. WSTĘP

Dwuwymiarową dyskretną transformatę kosinusową można rozłożyć na dwie transformaty jednowymiarowe (1D-DCT) [3] z tym, że jedna będzie przeprowadzana względem wierszy a druga względem kolumn. Jeżeli x_n oznacza wektor wartości wejściowych to wektor współczynników transformaty y_n można wyznaczyć na podstawie zależności (1)

$$y_k = \frac{1}{2}\alpha_k \sum_{n=0}^{N-1} x_n \cos\left[\frac{2\pi}{4N}(2n+1)k\right] \tag{1}$$

gdzie: $k = 0, 1, \ldots, N-1$, $\alpha_0 = 1/\sqrt{2/N}$ dla $k = 0$ oraz $\alpha_k = 1/\sqrt{2/N}$ w pozostałych przypadkach.

Dyskretną transformatę można również przedstawić w postaci macierzowej. Dla $N = 8$ postać macierzowa 1D-DCT jest zobrazowana zależnością (2)

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \alpha_k \begin{bmatrix} c_0 & c_0 & c_0 & c_0 & c_0 & c_0 & c_0 & c_0 \\ c_1 & c_3 & c_5 & c_7 & -c_7 & -c_5 & -c_3 & -c_1 \\ c_2 & c_6 & -c_6 & -c_2 & -c_2 & -c_6 & c_6 & c_2 \\ c_3 & -c_7 & -c_1 & -c_5 & c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 & c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 & -c_3 & -c_7 & c_1 & -c_5 \\ c_6 & -c_2 & c_2 & -c_6 & -c_6 & c_2 & -c_2 & c_6 \\ c_7 & -c_5 & c_3 & -c_1 & c_1 & -c_3 & c_5 & -c_7 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \tag{2}$$

gdzie współczynniki c_k wynoszą: $c_k = cos\dfrac{k\pi}{16}$.

Jak widać z zależności (2) do obliczenia pojedynczego wektora współczynników transformaty trzeba wykonać 64 mnożenia i 56 dodawań.

Liczbę wykonywanych operacji mnożenia ze wzoru (2) można zredukować z 64 do 32 poprzez wprowadzenie zerowych elementów c_k macierzy współczynników, lecz wtedy wzrośnie liczba dodawań i odejmowań. W tym przypadku postać macierzowa będzie miała postać (3). Warto zauważyć pewne symetrie w macierzy współczynników we wzorze (3): górna lewa ćwiartka macierzy ma pionową oś symetrii, a dolna prawa ma ukośną oś symetrii.

$$\begin{bmatrix} y_0 \\ y_2 \\ y_4 \\ y_6 \\ y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \alpha_k \left[\begin{array}{cccc|cccc} c_0 & c_0 & c_0 & c_0 & 0 & 0 & 0 & 0 \\ c_2 & c_6 & -c_6 & -c_2 & 0 & 0 & 0 & 0 \\ c_4 & -c_4 & -c_4 & c_4 & 0 & 0 & 0 & 0 \\ c_6 & -c_2 & c_2 & -c_6 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & c_1 & c_3 & c_5 & c_7 \\ 0 & 0 & 0 & 0 & c_3 & -c_7 & -c_1 & -c_5 \\ 0 & 0 & 0 & 0 & c_5 & -c_1 & c_7 & c_3 \\ 0 & 0 & 0 & 0 & c_7 & -c_5 & c_3 & -c_1 \end{array}\right] \cdot \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \\ x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix} \tag{3}$$

Postępując dalej podobnie jak poprzednio, można zredukować liczbę mnożeń do zaledwie 22, co zaowocuje następującą zależnością (4)

$$\begin{bmatrix} y_0 \\ y_2 \\ y_4 \\ y_6 \\ y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \alpha_k \left[\begin{array}{cccc|cccc} c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_2 & c_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_6 & -c_2 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & c_1 & c_3 & c_5 & c_7 \\ 0 & 0 & 0 & 0 & c_3 & -c_7 & -c_1 & -c_5 \\ 0 & 0 & 0 & 0 & c_5 & -c_1 & c_7 & c_3 \\ 0 & 0 & 0 & 0 & c_7 & -c_5 & c_3 & -c_1 \end{array}\right] \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} \tag{4}$$

gdzie wektor w_n ma postać (5)

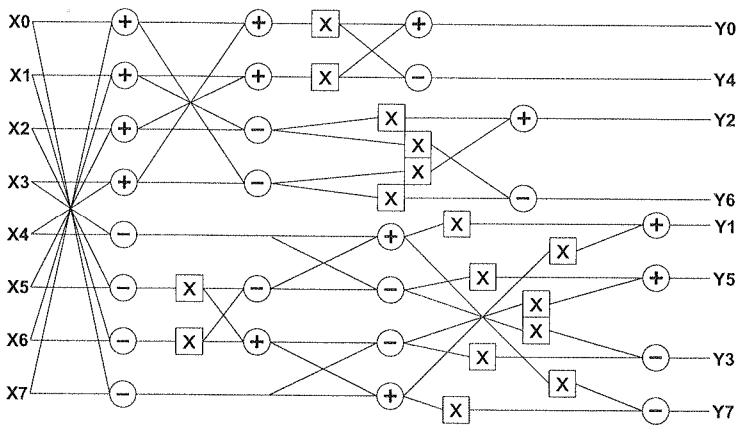
$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} = \begin{bmatrix} x_0 + x_7 + x_3 + x_4 + x_1 + x_6 + x_2 + x_5 \\ x_0 + x_7 + x_3 + x_4 - x_1 - x_6 - x_2 - x_5 \\ x_0 + x_7 - x_3 - x_4 \\ x_1 + x_6 - x_2 - x_5 \\ x_0 - x_7 \\ x_1 - x_6 \\ x_3 - x_4 \\ x_2 - x_5 \end{bmatrix} \tag{5}$$

Tak otrzymana macierz współczynników, pomimo że wymaga mniejszej liczby mnożeń, to jednak w przypadku realizacji sprzętowej powoduje wystąpienie problemu synchronizacji zadań obliczeniowych, wynikających z asymetrii lokalizacji zer w macierzy współczynników c_k we wzorze (4). W związku z tym liczba potrzebnych mnożeń do wyliczenia poszczególnych składowych wektora y_n będzie różna. Konsekwencją tego jest konieczność dobrania odpowiednich opóźnień w przypadku sprzętowej implementacji takiej niesymetrycznej dyskretnej transformaty kosinusowej.

2. SZYBKIE ALGORYTMY DCT

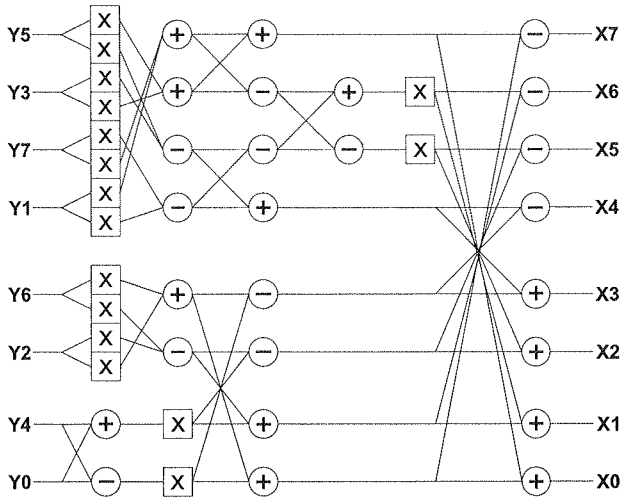
Dalsza redukcja wymaganej liczby mnożeń prowadzi do tak zwanych szybkich algorytmów dyskretnej transformacji kosinusowej FDCT (ang. *Fast DCT*).

Jednym z algorytmów FDCT jest algorytm Chena (rys. 1 i 2). Liczba mnożeń została w nim zredukowana do 16 oraz liczba dodawań-odejmowań do 26 operacji.



Rys. 1. Graficzny schemat przepływu danych dla transformaty DCT w algorytmie Chena

Fig. 1. The graphic scheme of data flow for the DCT transform in the Chen's algorithm

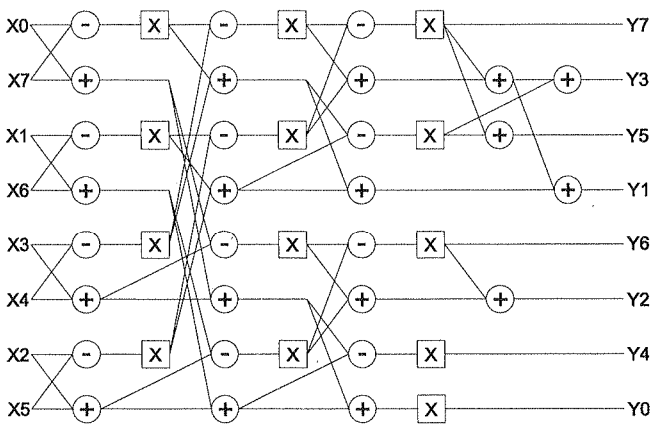


Rys. 2. Graficzny schemat przepływu danych dla transformaty IDCT w algorytmie Chena

Fig. 2. The graphic scheme of data flow for the IDCT transform in the Chen's algorithm

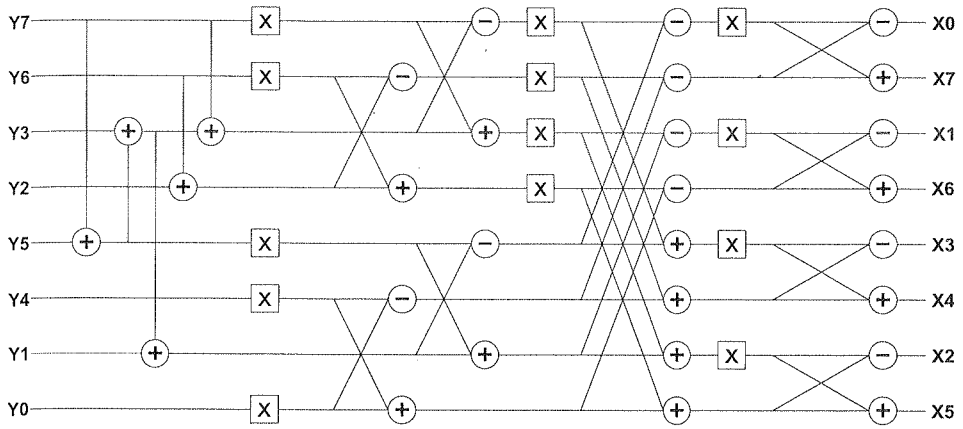
Jeszcze większą redukcję liczby potrzebnych operacji arytmetycznych otrzymano w algorytmie Lee [2] (rys. 3 i 4), w którym liczba mnożeń została zredukowana do 13, a liczba dodawań-odejmowań do 29.

Kolejnym algorytmem FDCT jest algorytm Loefflera [2, 3] (rys. 5). W tym przypadku liczba wymaganych mnożeń wynosi 11, a liczba operacji dodawań-odejmowań wynosi 29 (analogicznie jak dla algorytmu Lee).



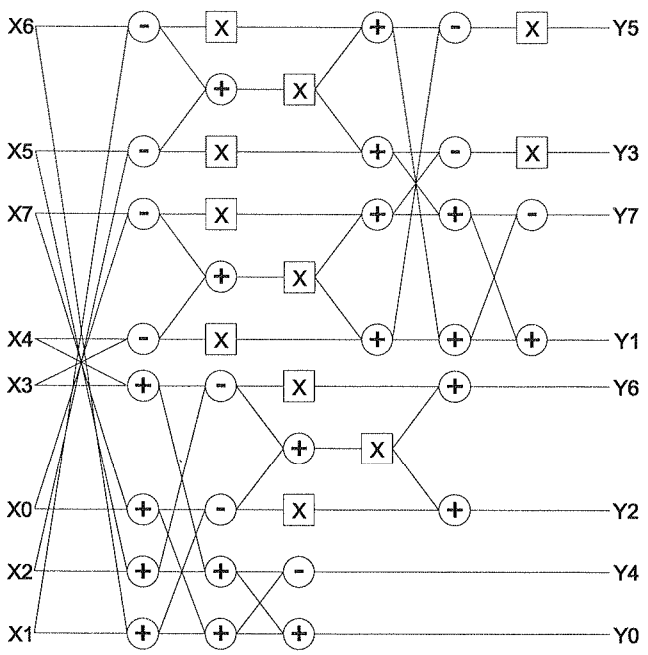
Rys. 3. Graficzny schemat przepływu danych dla transformaty DCT w algorytmie Lee

Fig. 3. The graphic scheme of data flow for the DCT transform in the Lee's algorithm



Rys. 4. Graficzny schemat przepływu danych dla transformaty IDCT w algorytmie Lee

Fig. 4. The graphic scheme of data flow for the IDCT transform in the Lee's algorithm



Rys. 5. Graficzny schemat przepływu danych dla transformaty DCT w algorytmie Loefflera

Fig. 5. The graphic scheme of data flow for the DCT transform in the Loeffler's algorithm

3. IMPLEMENTACJA W UKŁADACH FPGA

Podczas implementacji algorytmu 1D-DCT powstaje problem z jaką dokładnością należy stosować współczynniki mnożenia. Czy wystarczy dokładność rzędu 10^{-2}, czy też trzeba przyjąć współczynniki mnożenia z dokładnością rzędu wyższego. Szczególnie ważne jest, jak przyjęta dokładność przenosi się na wartości współczynników transformaty.

Dla dokładności odwzorowania współczynników mnożenia do drugiego miejsca po przecinku otrzymana maksymalna różnica pomiędzy współczynnikami transformacji obliczonymi analitycznie, a współczynnikami otrzymanymi z zaimplementowanego algorytmu Chena wynosi 12,751. Występuje ona przy maksymalnych wartościach wejściowych pikseli (x_0 do x_7 wynoszą maksymalnie 255). W tym przypadku wartość współczynnika transformacji y_0 wynosi 734, podczas kiedy wartość tego współczynnika obliczona analitycznie wynosi 721,249. Teoretycznie maksymalna różnica przy dokładności odwzorowania współczynników mnożenia do drugiego miejsca po przecinku wynosi 20,4 czyli jest większa od wartości otrzymanej z zaimplementowanego algorytmu.

Jeżeli współczynniki mnożenia zostały zrealizowane z dokładnością do trzeciego miejsca po przecinku to maksymalna różnica pomiędzy współczynnikami transformacji obliczonymi analitycznie, a otrzymanymi z zaimplementowanego algorytmu wynosi 1,182. Przy tej dokładności maksymalna różnica wyznaczona teoretycznie wynosi 2,04.

Pomimo wcześniejszych rozważań pozostaje nadal problem braku odpowiedzi na pytanie o wystarczającą dokładność odwzorowania współczynników mnożenia. W związku z tym należy zbadać jak otrzymane różnice pomiędzy poszczególnymi wartościami współczynników transformacji przenoszą się na wartości pikseli otrzymanych poprzez odwrotną dyskretną transformacje kosinusową (IDCT).

Tabela 1

Dokładność transformacji Chena dla przykładowego wektora wejściowego

The Chen's transform accuracy for an exemplary input vector

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Chena z dokładnością rzędu		Różnica pomiędzy y_n otrzymanym a y_n obliczonym		x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Chena	
		10^{-2}	10^{-3}	10^{-2}	10^{-3}	10^{-2}	10^{-3}
$x_0 = 252$	$y_0 = 645{,}5885$	$y_0 = 657$	$y_0 = 645$	-11,4115	0,58849	255	252
$x_1 = 250$	$y_1 = 55{,}33119$	$y_1 = 55$	$y_1 = 55$	0,33119	0,33119	255	250
$x_2 = 249$	$y_2 = 5{,}037243$	$y_2 = 5$	$y_2 = 5$	0,037243	0,03724	253	248
$x_3 = 247$	$y_3 = -27{,}3337$	$y_3 = -28$	$y_3 = -27$	0,66633	-0,33367	251	247
$x_4 = 207$	$y_4 = -22{,}6274$	$y_4 = -23$	$y_4 = -23$	0,372583	0,37258	211	206
$x_5 = 223$	$Y_5 = -21{,}849$	$y_5 = -23$	$y_5 = -21$	1,151043	-0,848958	226	223
$x_6 = 159$	$y_6 = 36{,}18185$	$y_6 = 37$	$y_6 = 36$	-0,81815	0,181849	162	159
$x_7 = 239$	$Y_7 = -32{,}817$	$y_7 = -33$	$y_7 = -33$	0,183041	0,183041	244	238

Tabela 2

Przypadek dla maksymalnej różnicy w algorytmie Chena pomiędzy y_n otrzymanym a y_n obliczonym dla dokładności rzędu 10^{-3}

A case for the maximum difference in the Chen's algorithm between y_n received and y_n calculated with the accuracy in the order of 10^{-3}

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Chena z dokładnością rzędu		Różnica pomiędzy y_n otrzymanym a y_n obliczonym		x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Chena	
		10^{-2}	10^{-3}	10^{-2}	10^{-3}	10^{-2}	10^{-3}
$x_0 = 253$	$y_0 = 393{,}15137$	$y_0 = 400$	$Y_0 = 393$	-6,84862	0,15137	256	253
$x_1 = 33$	$y_1 = 83{,}52634$	$y_1 = 82$	$Y_1 = 83$	1,5263376	0,526338	34	34
$x_2 = 247$	$y_2 = -25{,}132389$	$y_2 = -23$	$Y_2 = -25$	-2,132389	-0,132389	250	247
$x_3 = 121$	$y_3 = 78{,}72797$	$y_3 = 80$	$Y_3 = 78$	-1,2720269	0,72797	122	120
$x_4 = 87$	$y_4 = 46{,}669048$	$y_4 = 48$	$Y_4 = 47$	-1,3309524	-0,330952	87	87
$x_5 = 219$	$y_5 = 123{,}22545$	$y_5 = 126$	$Y_5 = 123$	-2,7745484	0,2254516	223	217
$x_6 = 123$	$y_6 = 157{,}36061$	$y_6 = 159$	$Y_6 = 157$	-1,639385	0,3606145	127	123
$x_7 = 29$	$y_7 = 41{,}818001$	$y_7 = 42$	$Y_7 = 43$	-0,1819986	-1,1819986	32	29

Tabela 3

Przykład dla małych różnic pomiędzy sąsiednimi pikselami dla zaimplementowanego algorytmu Chena

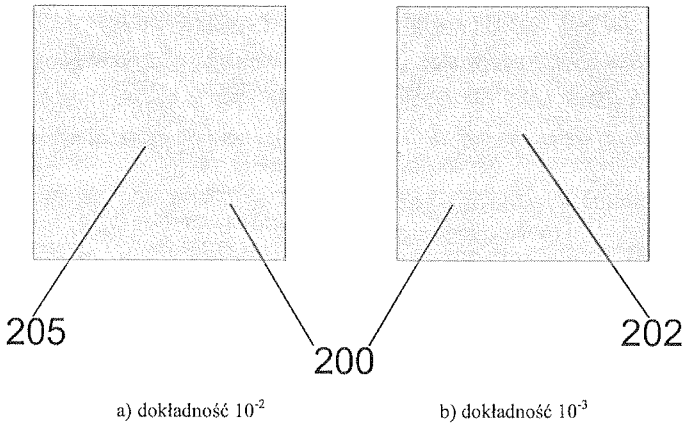
The example for small differences between neighbouring pixels for implemented Chen's algorithm

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Chena z dokładnością rzędu		Różnica pomiędzy y_n otrzymanym a y_n obliczonym		x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Chena	
		10^{-2}	10^{-3}	10^{-2}	10^{-3}	10^{-2}	10^{-3}
$x_0 = 191$	$y_0 = 530{,}3301$	$y_0 = 540$	$Y_0 = 530$	-9,67	0,3301	194	190
$x_1 = 190$	$y_1 = 6{,}442323$	$y_1 = 6$	$y_1 = 6$	0,4423	0,4423	194	190
$x_2 = 189$	$y_2 = 0$	$y_2 = 0$	$y_2 = 0$	0	0	193	189
$x_3 = 188$	$y_3 = 0{,}673455$	$y_3 = 0$	$y_3 = 0$	0,67345	0,67345	191	187
$x_4 = 187$	$y_4 = 0$	$y_4 = 0$	$y_4 = 0$	0	0	190	186
$x_5 = 186$	$y_5 = 0{,}200903$	$y_5 = 0$	$y_5 = 0$	0,2009	0,2009	189	185
$x_6 = 185$	$y_6 = 0$	$y_6 = 0$	$y_6 = 0$	0	0	188	184
$x_7 = 184$	$y_7 = 0{,}050702$	$y_7 = 0$	$y_7 = 0$	0,0507	0,0507	188	184

W tabelach od 1 do 3 można zauważyć, że przy dokładności rzędu 10^{-2} maksymalna różnica MD (ang. *Maximal Difference*) pomiędzy wartością piksela wejściowego, a wartością piksela zrekonstruowanego przy pomocy zaimplementowanego algorytmu 1D-DCT Chena wynosi 5. Jeżeli dokładność odwzorowania współczynników mnożenia jest rzędu 10^{-3} to maksymalna różnica MD pomiędzy pikselami wynosi 2.

Na rysunku 6 widać, że różnica dwóch pikseli jest prawie niezauważalna natomiast różnica 5 pikseli jest już różnicą widoczną.

Implementacja DCT Chena z dokładnością rzędu 10^{-3} w układzie XCV200BG352-6 firmy Xilinx zajmuje 574 z 2352 bloków SLICE tego układu (co stanowi 24% tych zasobów) oraz 958 z 4704 bloków LUT (20% tych zasobów układu). Maksymalna osiągalna częstotliwość dla tej implementacji wynosi 76,115MHz. Dla implementa-



Rys. 6. Maksymalne różnice pomiędzy pikselami dla dokładności 10^{-2} i 10^{-3}

Fig. 6. The maximum differences between pixels for the accuracy in the order of 10^{-2} and 10^{-3}

cji DCT Chena z dokładnością odwzorowania współczynników mnożenia rzędu 10^{-2} częstotliwość ta sięga 105,831MHz (również dla układu XCV200BG352-6). W tym przypadku bloki SLICE są zajęte w 16% (382 z 2352) a bloki LUT w 13% (614 z 4704).

Istnieje jeszcze jedna możliwość, która będzie kompromisem pomiędzy szybkością działania zaimplementowanego algorytmu, a dokładnością rekonstrukcji pikseli. Wystarczy zaimplementować algorytm DCT Chena z dokładnością odwzorowania współczynników mnożenia rzędu 10^{-2}, z wyjątkiem współczynnika mnożenia występującego przy wyznaczaniu współczynnika transformacji y_0. Rozwiązanie takie będzie nazywane dalej zmodyfikowanym algorytmem Chena.

Tabela 4

Dokładność zmodyfikowanego algorytmu Chena dla przykładowego wektora wejściowego

The accuracy of the modified Chen's algorithm for an exemplary input vector

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Chena z dokładnością rzędu 10^{-2} z wyjątkiem współczynnika dla y_0	Różnica pomiędzy y_n otrzymanym a y_n obliczonym	x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Chena
$x_0 = 253$	$y_0 = 393{,}15137$	$y_0 = 393$	0,15137	255
$x_1 = 33$	$y_1 = 83{,}52634$	$y_1 = 82$	1,526338	31
$x_2 = 247$	$y_2 = -25{,}132389$	$y_2 = -23$	-2,13239	247
$x_3 = 121$	$y_3 = 78{,}72797$	$y_3 = 80$	-1,27203	119
$x_4 = 87$	$y_4 = 46{,}669048$	$y_4 = 47$	-0,33095	84
$x_5 = 219$	$y_5 = 123{,}22545$	$y_5 = 126$	-2,77455	220
$x_6 = 123$	$y_6 = 157{,}36061$	$y_6 = 159$	-1,63939	124
$x_7 = 29$	$y_7 = 41{,}818001$	$y_7 = 42$	-0,182	30

Tabela 5

Przykład dla małych różnic pomiędzy sąsiednimi pikselami dla zmodyfikowanego algorytmu Chena

The example for small differences between neighbouring pixels for modified Chen's algorithm

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Chena z dokładnością rzędu 10^{-2} z wyjątkiem współczynnika dla y_0	Różnica pomiędzy y_n otrzymanym a y_n obliczonym	x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Chena
$x_0 = 191$	$y_0 = 530{,}3301$	$y_0 = 530$	0,330086	190
$x_1 = 190$	$y_1 = 6{,}442323$	$y_1 = 6$	0,442323	190
$x_2 = 189$	$y_2 = 0$	$y_2 = 0$	0	189
$x_3 = 188$	$y_3 = 0{,}673455$	$y_3 = 0$	0,673455	187
$x_4 = 187$	$y_4 = 0$	$y_4 = 0$	0	186
$x_5 = 186$	$y_5 = 0{,}200903$	$y_5 = 0$	0,200903	185
$x_6 = 185$	$y_6 = 0$	$y_6 = 0$	0	184
$x_7 = 184$	$y_7 = 0{,}050702$	$y_7 = 0$	0,050702	184

W tabelach 4 i 5 widać, że przy dokładności odwzorowania współczynników rzędu 10^2 (z wyjątkiem współczynnika mnożenia występującego przy wyznaczaniu y_0) maksymalna różnica MD pomiędzy wartością piksela wejściowego, a wartością piksela zrekonstruowanego wynosi 3. Otrzymana maksymalna różnica pomiędzy współczynnikami transformacji obliczonymi analitycznie, a współczynnikami otrzymanymi z zaimplementowanego algorytmu Chena w tym przypadku wynosi 2,89. Implementacja DCT Chena z tak zmodyfikowaną dokładnością rzędu 10^{-2} w układzie XCV200BG352-6 firmy Xilinx zajmuje 402 z 2352 bloków SLICE (co stanowi 17% zasobów układu) oraz 648 z 4704 bloków LUT (13% zasobów układu). Maksymalna osiągnięta częstotliwość dla tej implementacji wynosi 93,932MHz.

Algorytm 1D-DCT Lee został zaimplementowany z dokładnością odwzorowania współczynników mnożenia rzędu 10^{-3}. Zatem teoretycznie maksymalna różnica pomiędzy współczynnikami transformacji obliczonymi analitycznie a współczynnikami otrzymanymi z zaimplementowanego algorytmu nie powinna być większa niż 2,04. W tym przypadku maksymalna różnica wynosi 2,594. Maksymalna różnica MD pomiędzy wartością piksela wejściowego a odtworzoną wartością piksela wynosi 2.

Implementacja algorytmu 1D-DCT Lee z dokładnością rzędu 10^{-3} w układzie XCV200BG352 zajmuje 1277 z 2 352 bloków SLICE (czyli 54% bloków SLICE) oraz 2237 z 4704 bloków LUT (czyli 47% LUT). Maksymalna osiągalna częstotliwość dla implementacji algorytmu 1D-DCT Lee wynosi 35,668MHz.

Kolejnym algorytmem, który został zaimplementowany z dokładnością odwzorowania współczynników mnożenia rzędu 10^{-3}, jest algorytm 1D-DCT Loefflera. W przypadku tej implementacji maksymalna różnica wynosi 2,40562. Jest to wartość większa niż teoretyczna maksymalna różnica dla tej dokładności odwzorowania współczynników mnożenia. Maksymalna różnica MD pomiędzy wartością piksela wejściowego, a jego rekonstrukcją wynosi 2, co można zaobserwować w tabelach 8 i 9.

Tabela 6

Dokładność algorytmu Lee dla przykładowego wektora wejściowego

The accuracy of the Lee's algorithm for an exemplary input vector

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Lee	Różnica pomiędzy y_n otrzymanym a y_n obliczonym	x_n otrzymane z zaimplementowaneg o algorytmu 1D-IDCT Lee
$x_0 = 252$	$y_0 = 645{,}5885$	$y_0 = 646$	-0,411509	$x_0' = 251$
$x_1 = 250$	$y_1 = 55{,}33119$	$y_1 = 54$	1,3311912	$x_1' = 249$
$x_2 = 249$	$y_2 = 5{,}037243$	$y_2 = 5$	0,03724323	$x_2' = 247$
$x_3 = 247$	$y_3 = -27{,}3337$	$y_3 = -27$	-0,3336705	$x_3' = 247$
$x_4 = 207$	$y_4 = -22{,}6274$	$y_4 = -23$	0,372583	$x_4' = 207$
$x_5 = 223$	$y_5 = -21{,}849$	$y_5 = -22$	0,1510427	$x_5' = 223$
$x_6 = 159$	$y_6 = 36{,}18185$	$y_6 = 36$	0,1818488	$x_6' = 159$
$x_7 = 239$	$y_7 = -32{,}817$	$y_7 = -22$	0,1830408	$x_7' = 239$

Tabela 7

Przykład dla małych różnic pomiędzy sąsiednimi pikselami dla zaimplementowanego algorytmu Lee

The example for small differences between neighbouring pixels for implemented Lee's algorithm

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Lee	Różnica pomiędzy y_n otrzymanym a y_n obliczonym	x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Lee
$x_0 = 191$	$y_0 = 530{,}3301$	$y_0 = 530$	0,330086	$x_0' = 189$
$x_1 = 190$	$y_1 = 6{,}442323$	$y_1 = 4$	2,442323	$x_1' = 189$
$x_2 = 189$	$y_2 = 0$	$y_2 = 0$	0	$x_2' = 188$
$x_3 = 188$	$y_3 = 0{,}673455$	$y_3 = 0$	0,6734548	$x_3' = 187$
$x_4 = 187$	$y_4 = 0$	$y_4 = 0$	0	$x_4' = 187$
$x_5 = 186$	$y_5 = 0{,}200903$	$y_5 = 0$	0,200903	$x_5' = 186$
$x_6 = 185$	$y_6 = 0$	$y_6 = 0$	0	$x_6' = 185$
$x_7 = 184$	$y_7 = 0{,}050702$	$y_7 = 0$	0,0507023	$x_7' = 185$

Tabela 8

Przykład dla małych różnic pomiędzy sąsiednimi pikselami dla zaimplementowanego algorytmu Loefflera

The example for small differences between neighbouring pixels for implemented Loefflers's algorithm

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Loefflera z dokładnością rzędu 10^{-3}	Różnica pomiędzy y_n otrzymanym a y_n obliczonym	x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Loefflera
$x_0 = 191$	$y_0 = 530{,}3301$	$y_0 = 530$	0,330086	190
$x_1 = 190$	$y_1 = 6{,}442323$	$y_1 = 6$	0,442323	189
$x_2 = 189$	$y_2 = 0$	$y_2 = 0$	0	190
$x_3 = 188$	$y_3 = 0{,}673455$	$y_3 = -1$	1,673455	189
$x_4 = 187$	$y_4 = 0$	$y_4 = 0$	0	186
$x_5 = 186$	$y_5 = 0{,}200903$	$y_5 = 1$	-0,7991	185
$x_6 = 185$	$y_6 = 0$	$y_6 = 0$	0	185
$x_7 = 184$	$y_7 = 0{,}050702$	$y_7 = 0$	0,050702	185

Implementacja algorytmu 1D-DCT Loefflera z dokładnością rzędu 10^{-3} w układzie XCV200BG352-6 firmy Xilinx zajmuje 508 z 2352 bloków SLICE (co stanowi 21% zasobów układu) oraz 877 z 4704 bloków LUT (18% zasobów układu). Maksymalna osiągnięta częstotliwość dla implementacji tego algorytmu wynosi 79,195MHz.

Tabela 9

Przypadek dla maksymalnej różnicy w algorytmie Loefflera pomiędzy y_n otrzymanym a y_n obliczonym dla dokładności rzędu 10^{-3}

A case for the maximum difference in the Loeffler's algorithm between y_n received and y_n calculated with the accuracy in the order of 10^{-3}

x_n	y_n obliczone analitycznie	y_n otrzymane z zaimplementowanego algorytmu 1D-DCT Loefflera z dokładnością rzędu 10^{-3}	Różnica pomiędzy y_n otrzymanym a y_n obliczonym	x_n otrzymane z zaimplementowanego algorytmu 1D-IDCT Loefflera
$x_0 = 8$	$y_0 = 101{,}82338$	$y_0 = 102$	-0,17662	8
$x_1 = 16$	$y_1 = -51{,}53858$	$y_1 = -52$	0,461416	14
$x_2 = 24$	$y_2 = 0$	$y_2 = 0$	0	24
$x_3 = 32$	$y_3 = -5{,}387638$	$y_3 = -4$	-1,38764	30
$x_4 = 40$	$y_4 = 0$	$y_4 = 0$	0	41
$x_5 = 48$	$y_5 = -1{,}307223$	$y_5 = -1$	-0,60722	48
$x_6 = 56$	$y_6 = 0$	$y_6 = 0$	0	57
$x_7 = 64$	$y_7 = -0{,}405619$	$y_7 = 2$	-2,40562	63

4. PODSUMOWANIE

Porównując parametry osiągane przez zaimplementowane algorytmy Loefflera, Chena oraz Lee najlepszym algorytmem z dokładnością odwzorowania współczynników mnożenia rzędu 10^{-3} jest algorytm Loefflera. Przetwarzanie jednego wektora wejściowego wymaga 6 cykli zegara zarówno dla algorytmu Loefflera jak i dla algorytmu Chena, natomiast algorytm Lee wymaga 8 cykli. Pod kątem osiąganych maksymalnych częstotliwości lepszym algorytmem jest algorytm Loefflera, ponieważ maksymalna częstotliwość dla dokładności odwzorowania współczynników mnożenia rzędu 10^{-3} wynosi 79,195MHz a przy tej samej dokładności dla algorytmu Chena wynosi ona 76,115MHz a dla algorytmu Lee 35,668MHz. Również pod względem liczby wykorzystanych bramek (tab. 10) algorytm Loefflera jest korzystniejszy. Implementacja tego algorytmu zajmuje 13723 bramek w przypadku dokładności rzędu 10^{-3}, natomiast dla algorytmu Chena potrzebne jest 14514 bramek oraz 28257 bramek dla algorytmu Lee. Liczby bramek przekładają się na wielkość potrzebnych zasobów zastosowanego układu. Fakt ten również przemawia na korzyść implementacji algorytmu Loefflera. Dla zaimplementowanych algorytmów maksymalna różnica MD pomiędzy wartością pikseli wejściowych a wartością ich rekonstrukcji wynosi 2. Argumentem przemawiającym na korzyść algorytmu Chena jest maksymalna różnica pomiędzy współczynnikiem

obliczonym analitycznie a współczynnikiem otrzymanym z zaimplementowanego algorytmu. Różnica ta wynosi 1,182 dla algorytmu Chena, a dla algorytmu Loefflera 2,40562. W przypadku algorytmu Lee różnica ta jest większa i wynosi 2,594. Praca finansowana ze środków Komitetu Badań Naukowych w latach 2003-2005 jako projekt badawczy.

Tabela 10

Parametry implementacji w układzie FPGA

The parameters of the implementation in FPGA's device

XCV200BG352-6	Algorytm				
	Chena			Loefflera 10^{-3}	Lee 10^{-3}
	10^{-2}	Mod.10^{-2}	10^{-3}		
MD	5	3	2	2	2
max. różnica pomiędzy y_n obliczonymi analitycznie a y_n otrzymanymi	12,75	2,89	1,182	2,406	2,594
SLICE	382(16%)	402(17%)	574(24%)	508(21%)	1277(54%)
LUT	614(13%)	648(13%)	958(20%)	877(18%)	2237(47%)
Częstotliwość [MHz]	105,831	93,932	76,115	79,195	35,668

5. BIBLIOGRAFIA

1. R. Woods, A. Cassidy, J. Gray: *VLSI Architectures for Field Programmable Gate Arrays: A Case Study*, www.icspat.com.
2. R. Chotin, Y. Dumonteix, H. Mehrez: *Use of Redundant Arithmetic on Architecture and Design of a High Performance DCT Macro-block Generator*, www.asim.lip6.fr.
3. K. Bukhari, G. Kuzmanov, S. Vassiliadis: *DCT and IDCT Implementations on Different FPGA Technologies*, www.stw.nl.
4. J. Heron, D. Trainor, R. Woods: *Image Compression Algorithms Using Re-configurable Logic*, www.vcc.com.
5. D. Trainor, J. Heron, R. Woods: *Implementation of the 2D DCT Using a Xilinx XC6264 FPGA*, www.ee.qub.ac.uk.
6. K. Wiatr: *Dedicated Hardware Processors for a Real-Time Image Data Pre-Processing Implemented in FPGA Structure*, Lecture Notes in Computer Science - – no 1311, Springer-Verlag 1997, vol. II, pp. 69–75.
7. K. Wiatr: *Dedicated System Architecture for Parallel Image Computation used Specialised Hardware Processors Implemented in FPGA Structures*, International Journal of Parallel and Distributed Systems and Networks, vol. 1, No. 4, Pittsburgh 1998, pp. 161–168. 8
8. K. Wiatr, E. Jamro: *Implementation of Multipliers in FPGA Structure*, Proc. of the IEEE Int. Symp. on Quality Electronic Design, Los Alamitos CA, IEEE Computer Society 2001, pp. 415–420.
9. K. Wiatr: *Sprzętowe implementacje algorytmów przetwarzania obrazów w systemach wizyjnych czasu rzeczywistego*, Kraków, wyd. Naukowo-Dydaktyczne AGH 2002.
10. E. Jamro, K. Wiatr: *Dynamic Constant Coefficient Convolvers Implemented in FPGA*, Lecture Notes in Computer Science — no 2438, Springer Verlag 2002, pp. 1110–11.
11. K. Sayood: *Kompresja danych wprowadzenie*, Warszawa, RM 2002, wyd.1.

A. DĄBROWSKA, K. WIATR

TRANSFORM FDCT ALGORITHM MODIFIED FOR IMAGE COMPRESSION IMPLEMENTED IN FPGA STRUCTURES

Summary

The Discrete Cosine Transform (DCT) is one of the basic varieties of transform coding algorithms. DCT is used in standard algorithms of compression of still image (JPEG) and video compression algorithms (MPEG, H.26x). In case of compression's images algorithms there are used the blocs: 8x8 pixels. Paper presents basics and main algorithms of the DCT transform execution with special regard to issues of the implementation in programmable devices FPGA (XCV200BG352).

Keywords: image compression, vision systems, reconfigurable computation systems, programmable logic device

(Te
wyk
prow
met
moc
trów
waru
rowa
wspo
now
waru

Słow

Projek
nadprzewo
nów w po
o częstotli
relatywisty
[3].

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 643–653

Modelowanie systemu sterowania nadprzewodzącej wnęki rezonansowej akceleratora liniowego

TOMASZ CZARSKI

Instytut Systemów Elektronicznych, Politechnika Warszawska,
ul. Nowowiejska 15/19, 00-665 Warszawa,
tczarski@elka.pw.edu.pl

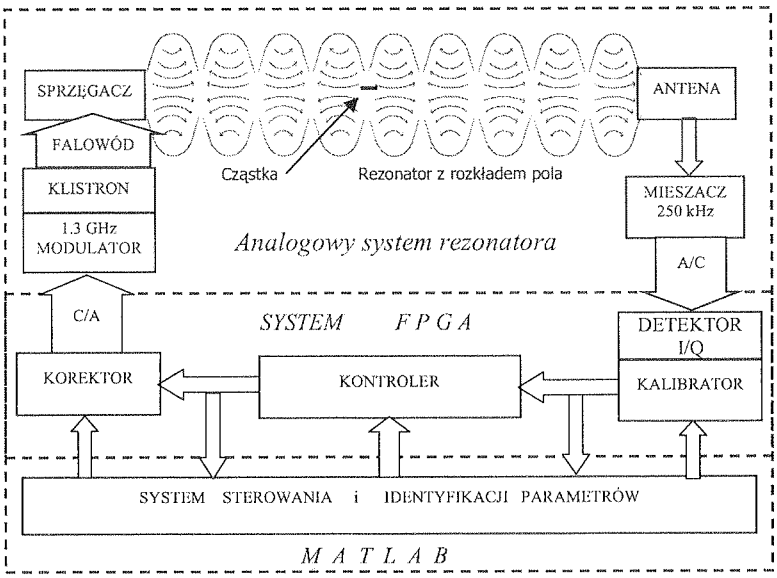
Otrzymano 2005.10.08
Autoryzowano 2006.03.24

Artykuł przedstawia system sterowania akceleratora linowego w technologii TESLA (TeV–Energy Superconducting Linear Akcelerator). Sterowanie nadprzewodzącej wnęki z wykorzystaniem układów FPGA zrealizowane zostało w ośrodku DESY w Hamburgu. Przeprowadzone testy koncentrowały się na rozpoznaniu obiektu i weryfikacji proponowanych metod sterowania. Elementarna analiza rezonatora wnękowego w oparciu o uproszczony model elektryczny obejmuje rozważania sygnałowe i energetyczne. Identyfikacja parametrów wnęki pozwala osiągnąć wymagane warunki pracy: napełnianie wnęki polem EM w warunkach rezonansu a następnie stabilizację pola z dopuszczalnym poborem mocy. Sterowanie bezpośrednie w układzie otwartym zgodnie z modelem procesu jest skutecznym wspomaganiem metody sprzężenia zwrotnego. Kalibracja i korekcja toru sygnałowego stanowi kluczowy fragment procedury. Przedstawiono wybrane rezultaty testu w typowych warunkach pracy rezonatora.

Słowa kluczowe: TESLA, FPGA, nadprzewodząca wnęka rezonansowa, sterowanie akceleratora

1. WSTĘP

Projekt akceleratora liniowego w technologii TESLA bazuje na 9-komorowych nadprzewodzących, niobowych rezonatorach do przyśpieszania elektronów i pozytonów w polu elektrycznym. Wnęka rezonansowa pracuje z falą stojącą w modzie π o częstotliwości 1.3 GHz. Pole elektromagnetyczne zsynchronizowane jest z wiązką relatywistycznych cząstek poruszających się wzdłuż komór rezonatora (Rys. 1) [1], [2], [3].



Rys. 1. Schemat funkcjonalny systemu sterowania wnęki rezonansowej

Fig. 1. Functional block diagram of cavity control system

Projektowany system sterowania akceleratora ma zapewnić kontrolowane przyśpieszanie cząstek poprzez skuteczną stabilizację pola wnęki. Sekcja sterująca zasilana przez jeden klistron separowany cyrkulatorem może zawierać wiele modułów wnęk rezonansowych sprzężonych z falowodem. Szybka kontrola amplitudy i fazy pola wnęki realizowana jest poprzez modulację sygnału sterującego klistronem. Rezonator pobudzany jest impulsowo przez okres ok. 1.3 ms z częstotliwością 10 Hz, a przeciętne natężenie pola sięga 25 MV/m. Sygnał wyjściowy wnęki przetwarzany jest do częstotliwości pośredniej 250 kHz zachowując informację o amplitudzie i fazie pola. Przetworniki A/C i C/A łączą część analogową i cyfrową systemu z częstotliwością próbkowania 1 MHz. W procesie cyfrowego przetwarzania następuje detekcja obwiedni zespolonej (demodulacja), kalibracja i filtrowanie sygnału. Układ sprzężenia zwrotnego reguluje sumę wektorową pola przyśpieszającego w przypadku wielu rezonatorów. Cyfrowy kontroler stabilizuje obwiednię zespoloną reprezentowaną przez składową rzeczywistą (I - synfazową) i urojoną (Q - kwadraturową) dekodowanego sygnału zgodnie z zadanym odniesieniem (*set-point*). Ponadto, dodatkowy sygnał bezpośredni (*feed-forward*) w oparciu o model procesu iteracyjnie kompensuje powtarzalne zakłócenia.

Układ sterowania na podstawie estymowanych parametrów procesu generuje odpowiednie dane (tablice sterujące) dla układu wykonawczego kontrolera.

Skuteczne sterowanie wnęki rezonansowej jest istotnym warunkiem optymalizacji systemu akceleratora TESLA.

2. MODELOWANIE SYSTEMU NADPRZEWODZĄCEJ WNĘKI REZONANSOWEJ

Modelowanie nadprzewodzącej wnęki rezonansowej stanowi ważny etap projektu skutecznego systemu sterowania.

Elektryczny model wnęki reprezentowany jest obwodem rezonansowym o parametrach skupionych RLC. W praktycznej analizie rezonator charakteryzowany jest następującymi parametrami wtórnymi: częstotliwość rezonansowa $\omega_0 = 2\pi f_0 = (LC)^{-1/2}$, rezystancja charakterystyczna $\rho = (L/C)^{1/2}$ ($\sim 520\Omega$), efektywna dobroć układu obciążonego $Q = R/\rho(\sim 3 \cdot 10^6)$, pasmo połówkowe $\omega_{1/2} = 1/2RC = \omega_0/2Q$ ($\sim 2\pi \cdot 230\text{Hz}$).

Wnęka dostrojona jest do częstotliwości wzorcowej generatora $\omega_g = 2\pi \cdot 1.3$ GHz, ale w polu elektromagnetycznym pod wpływem siły Lorentza ulega deformacji zmniejszając częstotliwość rezonansową ω_0. W typowych warunkach pracy (25 MV/m) odstrojenie $\Delta\omega \equiv \omega_0 - \omega_g$ porównywalne jest z pasmem rezonatora.

Wiązka przyśpieszanych cząstek — elektronów lub pozytonów pobiera energię pola EM powodując efekt obciążenia wnęki. Typowa wiązka ładunków jest impulsem prądowym o czasie 2 ps, częstotliwości 1 MHz i średnim natężeniu 8 mA.

Sprzężony z falowodem rezonator zasilany jest sygnałem z klistronu (wzmacniacz mocy) modelowanym jako prądowy generator separowany cyrkulatorem. Moc dostarczana z klistronu odbija się częściowo wskutek niedopasowania sprzęgacza i wytraca w obciążeniu cyrkulatora. Pozostała moc dostarczana do wnęki zwiększa energię pola w fazie ładowania oraz zwiększa energię przyśpieszanych cząstek w fazie stabilizacji pola. Znikoma część mocy rozprasza się w nadprzewodzących ściankach rezonatora.

Efektywne napięcie wnęki zależy, od wartości zastępczej prądu klistronu (po transformacji w sprzęgaczu) oraz obciążającego strumienia prądowego przyśpieszanych cząstek.

Wąskopasmowy sygnał rezonatora modelowany jest w dziedzinie czasu *sygnałem analitycznym*, gdzie amplituda i faza są wolnozmienne względem nośnej o częstotliwości 1.3 GHz. W analizie nisko-częstotliwościowej reprezentacją sygnału wnęki jest *obwiednia zespolona* uzyskana w procesie demodulacji *sygnału analitycznego* i reprezentowana przez składową rzeczywistą (synfazową) i składową urojoną (kwadraturową).

Efektywna transmitancja modelu wnęki dla obwiedni zespolonej jest wynikiem dolnopasmowej transformacji $Z(\mathbf{s} + \mathbf{j}\omega_g)$ pierwotnej funkcji przenoszenia $Z(\mathbf{s})$ modelu rezonatora.

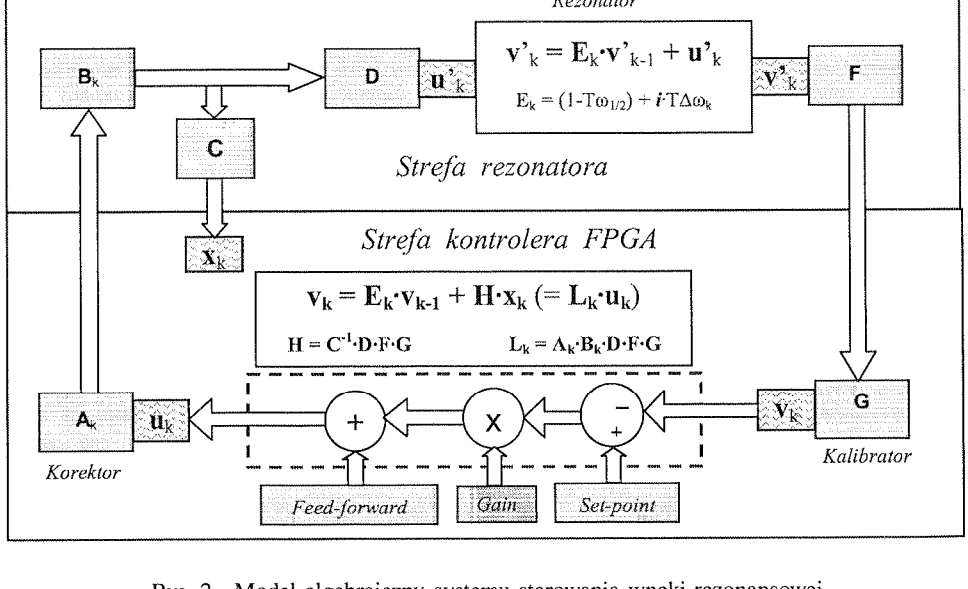
Elektryczny model wnęki reprezentowany jest równoważnie w dziedzinie czasu równaniem stanu dla sygnału *obwiedni zespolonej*. Dla potrzeb sterowania cyfrowego rozważany jest model dyskretny reprezentowany równaniem różnicowym z czasem próbkowania $T = 1\mu s$ dla obwiedni zespolonej $\mathbf{v}_k$ i wypadkowego pobudzenia $\mathbf{u}_k$ zgodnie z zależnością [4] :

$$\mathbf{v}_k = \mathbf{E}_k \cdot \mathbf{v}_{k-1} + \mathbf{u}_k, \quad \text{dla} \quad \mathbf{E}_k = (1 - \omega_{1/2}T) + \mathbf{i}\Delta\omega_k T.$$

Dla celów testowych i symulacji procesu stworzony został heurystyczny model mechaniczny nadprzewodzącej wnęki rezonansowej. Model ten opisuje dynamiczne odstrojenie $\Delta\omega$ wnęki powodowane siłą Lorentza w zależności od gradientu pola elektrycznego. Elektromechaniczny model nadprzewodzącej wnęki rezonansowej został zaimplementowany w układzie FPGA jako cyfrowy symulator procesu [4].

Dla celów kompleksowej analizy rozważany jest model algebraiczny systemu sterowania przedstawiony na rys 2. Elektryczny model wnęki rezonansowej stanowi jedyny człon dynamiczny systemu. Pozostałe statyczne ogniwa charakteryzowane są zespolonymi współczynnikami kształtującymi amplitudę i fazę sygnału obwiedni zespolonej. Otoczenie rezonatora: sprzęgacze, falowód, antena, mieszacz, kable opisane są liniowymi członami o charakterystykach **D** i **F**. Nieliniowy układ klistronu charakteryzowany jest współczynnikiem $\mathbf{B}_k$ zależnym od punktu pracy dla danego kroku „k". Dodatkowo dla celów analizy dekodowany jest pomocniczy sygnał **x** za pośrednictwem liniowego członu **C**.

Skuteczne sterowanie rezonatora wymaga odpowiedniej kalibracji i korekcji toru sygnałowego. Estymacja parametrów obiektu pozwala określić wymagany liniowy kalibrator **G** i niestacjonarny korektor $\mathbf{A}_k$ dla implementacji w strukturze FPGA układu kontrolera (rys 2).



Rys. 2. Model algebraiczny systemu sterowania wnęki rezonansowej

Fig. 2. Algebraic model of cavity control system

3. IDENTYFIKACJA PARAMETRÓW PROCESU

Wykorzystanie modelu procesu do sterowania w układzie otwartym (*feed-forward*) wymaga estymacji parametrów obiektu. Przedstawione rozważania dotyczą „jałowego" cyklu akceleratora bez obciążenia wiązką przyśpieszanych cząstek. Algorytm identyfikacji bazuje na modelu elektrycznym wnęki rezonansowej reprezentowanym dyskretnym równaniem stanu dla wektora obwiedni [5]. Uwzględnienie charakterystyki kanałów transmisji sygnału prowadzi do zewnętrznego modelu obiektu widzianego przez układ kontrolera zgodnie z oznaczeniami na rys 2:

$$\mathbf{v}_k = \mathbf{E}_k \cdot \mathbf{v}_{k-1} + \mathbf{H} \cdot \mathbf{x}_k \text{ oraz } \mathbf{H} \cdot \mathbf{x}_k = \mathbf{L}_k \cdot \mathbf{u}_k$$

W pierwszej fazie analizy estymuje się dwa zespolone parametry: stacjonarny **H** i niestacjonarny $\mathbf{E}_k = (1 - \omega_{1/2}T) + \mathbf{i}\Delta\omega_{k\cdot T}$. Rozważany algorytm estymacji sprzężony jest z procedurą kalibracji, a fazę kalibratora **G** ustala się iteracyjnie celem spełnienia warunku $\arg(\mathbf{H}) = 0$ równoważnego początkowej zgodności faz: $\arg(\mathbf{v}_1) = \arg(\mathbf{x}_1)$.

Zredukowane równanie zespolone modelu dla kroku „k" zawiera więc trzy niewiadome: moduł $|\mathbf{H}|$, pasmo połówkowe $\omega_{1/2}$ i odstrojenie $\Delta\omega_k$.

W praktycznej analizie niestacjonarnego procesu w obecności szumów rozważa się zestaw 2N równań modelu rozpisanych skalarnie w pełnym zakresie pomiarowym N kroków.

Dla niestacjonarnej częstotliwości rezonansowej odstrojenie wnęki w każdym kroku „k" można aproksymować szeregiem L wybranych funkcji bazowych f_i o nieznanych współczynnikach wagowych α_i zgodnie z zależnością:

$$\Delta\omega_k = \sum_{i=1}^{L} \alpha_i f_i(k), \qquad \text{L – rząd funkcji bazowych.}$$

W całym dyskretnym zakresie pomiarowym N kroków wypadkowy układ 2N równań modelu zawiera L + 2 nieznane wielkości zestawione w wektor kolumnowy $\mathbf{z} = [|\mathbf{H}|; (1 - \omega_{1/2}T); \alpha_i; \ldots \alpha_L]$. Rozważany model procesu jest więc reprezentowany równaniem macierzowym zgodnie z zależnością:

$$\mathbf{X} = \mathbf{Y} \star \mathbf{z}, \text{ gdzie}$$

X — całkowity wektor kolumnowy [2N x 1] wyrazów wolnych zależny od kolejnych rejestrowanych *obwiedni zespolonych* $\mathbf{v}_k$

Y — macierz [2N × L + 2] struktury modelu zależna od mierzonych wartości $\mathbf{v}_{k-1}$ i $\mathbf{x}_k$ oraz od szeregu danych funkcji bazowych f_i.

Dla nad-określonego równania macierzowego ($2N \gg L + 2$) wektor parametrów **z** jest skutecznie estymowany metodą najmniejszych kwadratów *(least squares)* zgodnie z zależnością:

$$\mathbf{z} = (\mathbf{Y}^T * \mathbf{Y})^{-1} * \mathbf{Y}^T * \mathbf{X}.$$

Algorytm identyfikacji parametrów wnęki rezonansowej zaimplementowany został w MATLAB-ie w różnych wersjach aplikacji z wykorzystaniem wielomianów sklejanych trzeciego stopnia (*cubic b-spline*) jako funkcji aproksymujących.

Dla powtarzalnego procesu, o deterministycznej składowej zakłóceń, procedura ta może być zastosowana do estymacji odstrojenia wnęki w czasie pomiędzy impulsami (*off- line*).

W drugiej fazie analizy estymuje się niestacjonarny parametr $\mathbf{L}_k$ z wykorzystaniem określonego już członu $\mathbf{H} \cdot \mathbf{x}_k$.

Wysterowany analogicznie model w MATLAB-ie wykazuje zadowalającą zgodność z rzeczywistością weryfikując jednocześnie algorytm identyfikacji parametrów rezonatora.

4. STEROWANIE WNĘKI REZONANSOWEJ

4.1. UKŁAD KONTROLERA WNĘKI REZONANSOWEJ

Moduł FPGA kontrolera stanowi układ wykonawczy dla algorytmów DSP (*Digital Signal Procesing*) sterowania nadprzewodzącej wnęki rezonansowej. Schemat funkcjonalny bloku sterującego przedstawia rys.1 a model algebraiczny rys. 2.

Sygnał pośredniej częstotliwości 250 kHz próbkowany jest z okresem T = 1μs z możliwością uśredniania próbek dla maksymalnej częstotliwości 64 MHz 14-bitowego przetwornika A/C.

Podstawową funkcją stopnia wejściowego jest detekcja *obwiedni zespolonej* z sygnału pośredniej częstotliwości 250 kHz (demodulacja). Stosowany algorytm detekcji skutecznie estymuje szybkozmienną obwiednię z jednoczesną redukcją wolnozmiennego *offsetu*.

Moduł kalibracji wektora obwiedni koryguje amplitudę i przesunięcie fazy kanału pomiarowego dla pojedynczej wnęki rezonansowej.

Estymowany sygnał obwiedni zespolonej v porównywany jest z wektorem odniesienia z tablicy *Set-point*. Uzyskany wektor błędu wzmacniany jest z wykorzystaniem mnożników w tablicy *Gain* (kontroler proporcjonalny) tworząc sygnał sprzężenia zwrotnego (*feed-back*).

Sygnał sterowania bezpośredniego (*feed-forward*) kompensuje deterministyczną składową zakłóceń i niezależnie wspomaga sygnał sprzężenia zwrotnego.

Końcowy moduł korekcji w oparciu o tablicę współczynników kalibracji dla każdego kroku procesu linearyzuje charakterystykę statyczną kanału wejściowego wnęki rezonansowej (układ klistronu) dla sygnału obwiedni sterującej **u**. Tablica korekcji dla kolejnych cykli n wyznaczana jest rekurencyjnie celem spełnienia warunku kalibracji: $\mathbf{L}_k = \mathbf{1}$ zgodnie z zależnością: $[\mathbf{A}_k]_{n+1} = [\mathbf{A}_k]_n / [\mathbf{L}_k]_n$.

4.2. STEROWANIE W UKŁADZIE OTWARTYM (*FEED-FORWARD*) I ZAMKNIĘTYM (*FEED-BACK*)

Skuteczna estymacja parametrów obiektu procesu połączona z kalibracją i korekcją toru sygnałowego pozwala wykorzystać wewnętrzny model rezonatora (Rys. 2) do określenia wymaganych tablic sterujących kontrolera: *Feed_forward* i *Set-Point*.

• Faza ładowania (*filling*)

Optymalne napełnianie wnęki energią pola elektromagnetycznego zakłada sterowanie w rezonansie ze stałą mocą generatora. Amplituda jest stała a częstotliwość sygnału nadąża za zmianami częstotliwości rezonatora $\Delta\omega_k$ względem pulsacji odniesienia generatora ω_g. Warunek ten determinuje wymagany sygnał sterujący $\mathbf{u}_k$ w układzie otwartym *(feed-forward)* zgodnie z zależnością:

$$\mathbf{u}_k = u_1 \cdot \exp(\mathbf{i}\varphi_k) \text{ z modulacją fazy } \varphi_k = \varphi_{k-1} + T\Delta\omega_k$$

Wynikowa obwiednia zespolona napięcia wnęki określa sygnał odniesienia (*set-point*) w układzie sprzężenia zwrotnego zgodnie z zależnością:

$$\mathbf{v}_k = \mathbf{u}_k[1-\exp(-k \cdot T\omega_{1/2})]/T\omega_{1/2}$$

Wartości początkowe $\mathbf{ru}_1$ i φ_1 wynikają z wymaganej wartości docelowej $\mathbf{V} = V \cdot \exp(\mathbf{i}\Phi)$ dla fazy stabilizacji.

• Faza stabilizacji (*flattop*)

Stabilizacja amplitudy V i fazy Φ sygnału jednoznacznie określa wartość odniesienia (*set-point*) w układzie sprzężenia zwrotnego zgodnie z zależnością:

$$\mathbf{v}_k = \mathbf{V} = V \cdot \exp(\mathbf{i}\Phi) = \text{const.}$$

Dokładność stabilizacji ograniczona jest wzmocnieniem kontrolera, które limitowane jest stabilnością układu z opóźnieniem w pętli.

Rozwiązanie równania modelu wnęki w stanie ustalonym określa wymagany sygnał *obwiedni zespolonej* sterowania w układzie otwartym (*feed-forward*) zgodnie z zależnością:

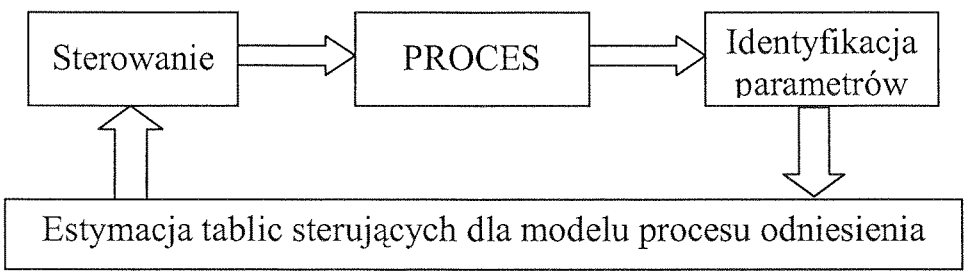
$$\mathbf{u}_k = \mathbf{V} \cdot (\omega_{1/2} - \mathbf{i}\Delta\omega_k) \cdot T$$

Obwiednia sterująca kompensuje zmiany impedancji rezonatora kosztem dodatkowej mocy wskutek niedopasowania wnęki. Celem minimalizacji tej mocy, symetryzuje się odchylenie częstotliwości rezonansowej względem pulsacji wzorcowej generatora poprzez wstępnie zadane dodatnie odstrojenie wnęki (*predetuning*).

4.3. ADAPTACYJNE STEROWANIE WNĘKI

Zasadniczym celem sterowania wnęki rezonansowej jest stabilizacja obwiedni zespolonej w fazie *flattop*, a stąd wynikająca powtarzalność procesu. Algorytm generowania tablic sterujących *Feed_forward* bazuje na estymacji parametrów minionego procesu i realizowany jest pomiędzy kolejnymi cyklami pracy akceleratora (*off-line*).

Iteracyjne sterowanie odbywa się cyklicznie zgodnie z powtarzającą się sekwencją wg rysunku 3. Dewiacja trajektorii procesu powoduje zmianę estymowanych parametrów, które decydują o tablicach sterujących oddziałujących ponownie na sterowany proces. Dla deterministycznych zakłóceń algorytm sterowania adaptacyjnego jest zbieżny do ustalonej trajektorii procesu odniesienia. Procedura uśredniająca identyfikowane parametry w kolejnych cyklach niweluje fluktuacje od stanu ustalonego powodowane zakłóceniami stochastycznymi. Natomiast sprzężenie zwrotne w czasie rzeczywistym procesu kompensuje bezpośrednio wpływ szybkich, przypadkowych zakłóceń pożądanego stanu równowagi.

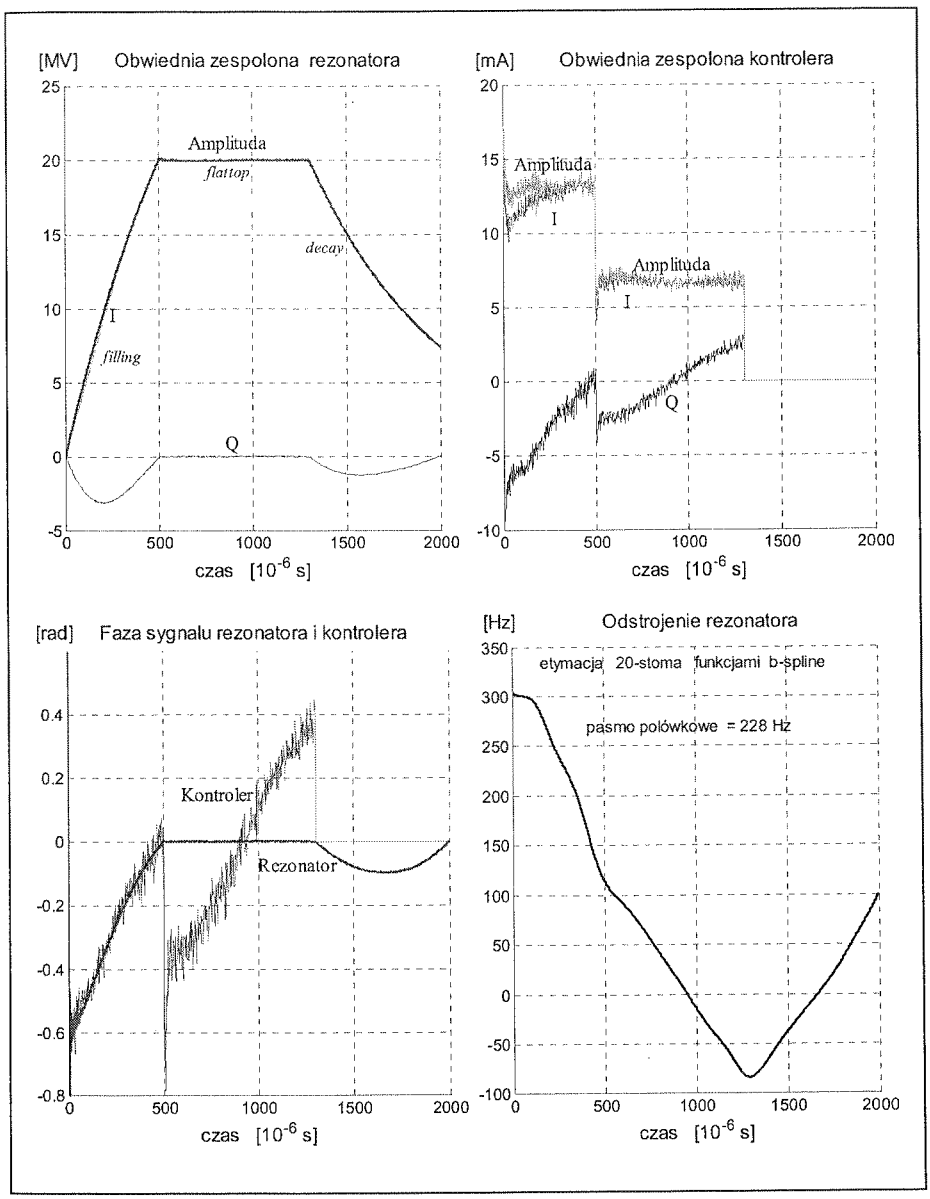


Rys. 3. Schemat ideowy algorytmu sterowania adaptacyjnego wnęki rezonansowej

Fig. 3. Schematic diagram for adaptive control process

Rezultat testów sterowania modułu rezonatora w ośrodku DESY w Hamburgu w układzie ze sprzężeniem zwrotnym (wzmocnienie pętli=100) wspomaganym sygnałem *feed-forward* przedstawia rys. 4.

Wnęka pobudzana jest impulsowo przez okres ok 1.3 ms z częstotliwością ok. 10 Hz. Sygnał sterujący moduluje amplitudę i fazę nośnej 1.3 GHz celem uzyskania pożądanej reakcji rezonatora. Sterowanie wnęki odbywa się w systemie sprzężenia zwrotnego (*feed-back*) wspomaganego sygnałem sterowania wprost (*feed-forward*). W pierwszej fazie ~ 0.5 ms (*filling*) wnęka jest ładowana ze stałą mocą generatora wymuszającą wykładnicze narastanie pola EM w warunkach rezonansu. Wstępne dodatnie odstrojenie rezonatora (zadane mechanicznie) symetryzuje (względem częstotliwości wzorcowej generatora) zmianę częstotliwości rezonansowej pod wpływem wzrastającej siły Lorentza. Po osiągnięciu pożądanej wartości amplitudy i fazy pola, wymuszany jest stan ustalony i następuje faza stabilizacji ~ 0.8 ms (*flattop*). Po wyłączeniu mocy klistronu następuje swobodny wykładniczy zanik pola wnęki (*decay*).



Rys. 4. Wyniki testów sterowania modułu rezonatora w typowych warunkach pracy w układzie ze sprzężeniem zwrotnym wspomaganym sygnałem *feed-forward*

Fig. 4. Feed-forward with feedback driving the cavity for typical operational condition

5. PODSUMOWANIE

W artykule przedstawiono zasadnicze aspekty systemowe sterowania nadprzewodzącej wnęki rezonansowej akceleratora liniowego w technologii TESLA.

Model elektryczny rezonatora reprezentowany jest liniowym, niestacjonarnym równaniem stanu dla wektora *obwiedni zespolonej* pola.

Optymalne sterowanie zakłada ładowanie (*filling*) wnęki w rezonansie stałą mocą i stabilizację *obwiedni zespolonej* w fazie *flattop*.

Sygnał sprzężenia zwrotnego (*on-line feed-back*) wspomagany jest sterowaniem adaptacyjnym (*off-line adaptive feed-forward*), które bazuje na estymacji parametrów powtarzalnego procesu.

Dla założonej struktury modelu możemy estymować nieznane parametry procesu na podstawie zarejestrowanych danych z wejścia i wyjścia obiektu. Proponowany algorytm identyfikacji metodą najmniejszych kwadratów (*least squares*) wykorzystuje nad-określone równanie macierzowe procesu obejmujące odpowiedni zakres pomiarowy. Zmienne w czasie odchylenie od częstotliwości rezonansowej wnęki (*detuning*) aproksymowane jest szeregiem funkcji bazowych (*cubic b-spline*).

Układ wykonawczy kontrolera cyfrowego został zrealizowany w technologii FPGA. Model wnęki, metoda estymacji parametrów procesu i algorytmy sterowania zostały zweryfikowane eksperymentalnie dla modułu akceleratora liniowego w ośrodku DESY w Hamburgu.

Dalsze badania przewidują optymalizację proponowanych metod sterowania i rozszerzenie technologii FPGA w implementacji procedur DSP wspomagających sterowanie wnęki.

6. BIBLIOGRAFIA

1. T. Schilcher: *Vector Sum Control of Pulsed Accelerating Fields in Lorentz Force Detuned Superconducting Cavities*, Ph. D. thesis, Hamburg, 1998.
2. A.H. Hofler, V. Ayvazyan, A. Brandt, S. Simrock, T. Czarski, J.R. Delayen, T. Matsumoto: *RF Control Modelling Issues for Future Superconducting Accelerators*, LINAC 2004 — Proceedings Lübeck, Germany.
3. K.T. Poźniak, T. Czarski, R.S. Romaniuk: *System pomiarowokontrolny dla nadprzewodzącej, mikrofalowej wnęki rezonansowej akceleratora TESLA i europejskiego lasera XEL*, Kwartalnik Elektroniki i Telekomunikacji PAN, Vol. 51, No 1, 2005, pp. 175–210.
4. T. Czarski, R.S. Romaniuk, K.T. Pozniak, S. Simrock: *TESLA cavity modeling and digital implementation in FPGA technology for control system development*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment -Volume 556, Issue 2, 15 January 2006, Pp. 565–576.
5. T. Czarski, R.S. Romaniuk, K.T. Pozniak, S. Simrock: *Cavity parameters identification for TESLA control system development*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment-Volume 548, Issue 3, 21 August 2005, pp. 283–618.
6. http://www.desy.de [Deutsche Elektronen-Synchrotron DESY]

T. CZARSKI

CONTROL SYSTEM MODELING FOR SUPERCONDUCTING ACCELERATOR

Summary

The cavity control system for the TESLA — TeV–Energy Superconducting Linear Accelerator project is preliminary introduced in this paper. The elementary analysis of the cavity resonator with signal and power considerations is presented. The digital control of the superconductive cavity has been performed by applying the FPGA (Field Programmable Gate Array) technology system in DESY Hamburg. These experiments focused attention on the general recognition of the cavity features and projected control methods. An electrical model of the resonator was taken as a starting point. Calibration and correction of the signal path is considered key in preparation for the efficient driving of a cavity. Identification of the resonator parameters has been proven to be a successful approach in achieving required performance; i.e. driving on resonance during *filling* and field stabilization during *flattop* time while requiring reasonable levels of power consumption. Feed-forward and feedback modes were successfully applied in operating the cavity. Representative results of the experiments are presented for the typical operational condition.

Keywords: TESLA, FPGA, superconductive cavity, cavity control

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 655–668

Modularny system fotoniczny z programowalną warstwą sterowania i komunikacji w układzie FPGA

ARTUR DYBKO[1], RAFAŁ GRACZYK[2], KRZYSZTOF T. POŹNIAK[2], RYSZARD S. ROMANIUK[2]

1) Wydział Chemiczny, Politechnika Warszawska,
ul. Noakowskiego 3, 00-664 Warszawa,
2)Instytut Systemów Elektronicznych, Politechnika Warszawska,
ul. Nowowiejska 15/19, 00-665 Warszawa

Otrzymano 2005.11.15
Autoryzowano 2006.05.08

W pracy przedstawiono modularny system fotoniczny wyposażony w programowalny układ FPGA dużej skali, optyczne łącza synchronicznej transmisji gigabitowej oraz zaimplementowane przemysłowe interfejsy komunikacyjne. Omówiono zadania funkcjonalne dla poszczególnych bloków urządzenia oraz krótko scharakteryzowano rozwiązanie warstwy sprzętowej uzyskane w realizacji niniejszego projektu. Przedstawiono praktyczne przykłady zastosowań systemu w fotonicznych układach pomiarowych.

Słowa kluczowe: urządzenia fotoniczne, układ programowalny Xilinx, programowanie behawioralne w VHDL z parametryzacją, optyczna transmisja gigabitowa, przemysłowe interfejsy komunikacyjne

1. WSTĘP

Praktyczne zastosowanie zróżnicowanych czujników fotonicznych w zintegrowanych, wielokanałowych systemach pomiarowych, wymaga sprzężenia ich z elektroniczną warstwą sterowania, akwizycji danych, monitoringu, diagnostyki, synchronicznej dystrybucji danych oraz komunikacji. Spełnienie powyższych wymagań doprowadziło autorów do opracowania uniwersalnego Systemu Fotonicznego (SF) o rozproszonej, wielowęzłowej i standaryzowanej strukturze. Poszczególne rodzaje torów optycznych zostały zintegrowane z SF poprzez odpowiedniego typu interfejsy. Część elektroniczna interfejsów stanowi w konsekwencji pomost łączący optoelektroniczny tor pomiarowy z cyfrową częścią SF.

Wykorzystanie układów *Field-Programmable Gate Array* (FPGA) [1, 2] w realizacji SF pozwoliło na uzyskanie rozwiązania stabilnego strukturalnie oraz programowalnego funkcjonalnie. W wyniku gwałtownego postępu technologicznego, współczesne duże układy FPGA udostępniają w jednym układzie scalonym do:

- kilkuset tysięcy konfigurowalnych bloków logicznych,
- dziesięciu megabajtów parametryzowanej pamięci statycznej,
- dwustu szybkich bloków przetwarzania numerycznego, takich jak stałoprzecinkowe układy mnożące lub mnożąco-sumujące [3, 4],
- kilkunastu niezależnych modułów gigabitowej transmisji synchronicznej łączami elektrycznymi lub optycznymi [5, 6],
- kilku procesorów wbudowanych (klasy Power-PC 407 [7] w seriach VirtexIIPro i Virtex4 firmy Xilinx [1]) lub syntezowanych (np. typu „NIOS" [8] w układach firmy Altera lub „MicroBlaze" [9] dla produktów firmy Xilinx [2]).

Wymienione zasoby logiczne umożliwiają implementacje szybkich, wielowątkowych i bardzo złożonych funkcjonalnie procesów nadążnego sterowania, rejestracji danych oraz przetwarzania numerycznego wraz z analizą danych w czasie rzeczywistym. Zaimplementowano standardy szybkiej komunikacji danych, jak m. in. Ethernet, FastEthernet, RocketIO [10], PCI, LVDS, które umożliwiają realizację modularnego systemu rozproszonego. Rozwiązanie tego typu posiada wbudowany system sterowania, monitoringu i diagnostyki oraz standaryzowane kanały łączności z nadrzędnymi systemami komputerowymi.

2. MODULARNY SYSTEM FOTONICZNY

Przedstawiony w niniejsze pracy Modularny System Fotoniczny (MSF) opracowano w formie rozproszonych modułów funkcjonalnych (węzłów) połączonych wzajemnie szybką, synchroniczną siecią fotoniczną. Konstrukcje węzła zrealizowano w formie pojedynczej płyty bazowej PCB w standardzie mechanicznym EURO-6HE-160 [11]. Płyta stanowi uniwersalną, modularną i rekonfigurowaną platformę dla specjalizowanych optoelektronicznych modułów wykonawczych osadzonych w standardzie mechanicznym PMC [12]. Powyższe rozwiązanie pozwala na wyodrębnienie wspólnych elementów MSF, a nie na ich wielokrotne powielanie w każdym osobnym module wykonawczym. Uzyskano zarówno ujednolicenie standardu mechanicznego jak i ujednolicenie sterowania z poziomu oprogramowania nadzorującego. W wyniku zastosowanych rozwiązań, proces projektowania obejmuje mniejszy zakres wymagań funkcjonalnych i prowadzi do zmniejszenia czasu oraz kosztów realizacji.

Płyta bazowa MSF składa się z następujących elementów funkcjonalnych:

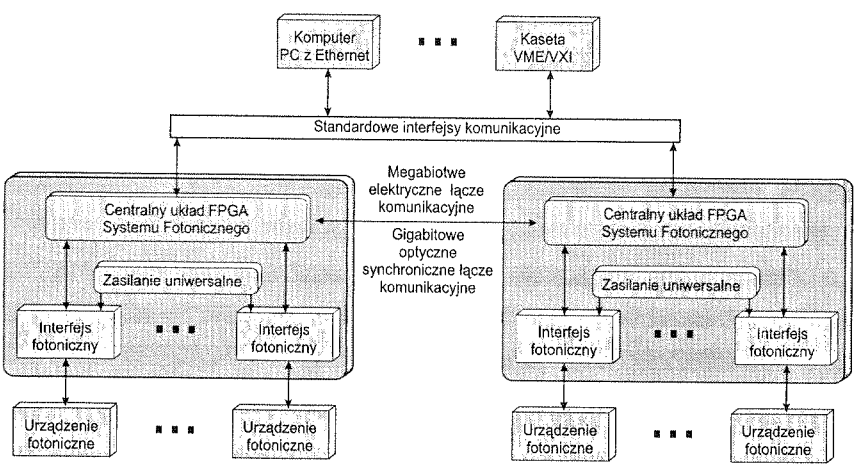
1. programowalnego sterownika MSF zaimplementowanego w całości w układzie FPGA firmy XILINX z serii VIRTEX II Pro [13]. Jego zadaniem jest m.in. zapewnienie:
   - komunikacji z innymi węzłami MSF poprzez dwa gigabitowe synchroniczne łącza optyczne wyposażone w nadajnik i odbiornik sygnału optycznego w każdym z nich,

- komunikacji ze standardowymi interfejsami zewnętrznymi, takimi jak: VME/VXI [14], Ethernet-100Tb, CAN-BUS [15], USB2.0, RS-232, LPT/EPP.
- wewnętrznej komunikacji z poszczególnymi modułami osadzanymi PMC z wykorzystaniem magistrali PCI-33/66 MHz[16] lub transmisji szeregowej w standardzie LVDS dla szybkości transferu danych od 120 do 460 Mb/s,
- dystrybucji w standardzie LVDS szybkich sygnałów sterujących i synchronizujących pracę poszczególnych modułów osadzanych PMC.

2. uniwersalnych gniazd dla optoelektronicznych modułów osadzanych. Każde z trzech gniazd posiada bezpośrednie, szybkie połączenia w standardzie LVDS z układem centralnym FPGA. Przyjęte rozwiązanie pozwala na realizację synchronicznej pracy poszczególnych modułów. Poprzez zaprogramowanie wzajemnych połączeń wewnątrz układu FPGA można odpowiednio dostosować dystrybucję danych, sygnałów kontrolnych i sterujących do wymagań technicznych urządzeń fotonicznych,
3. wspólnego systemu zasilania, które dostarcza do poszczególnych gniazd typowe napięcia dla współczesnych układów FPGA, konwerterów, jak również elementów fotonicznych i analogowych.

Na rys. 1 przedstawiono przykład realizacji MSF w postaci rozproszonego urządzenia dwuwęzłowego. Połączenia pomiędzy poszczególnymi węzłami mogą być realizowane poprzez:

- optoelektroniczne gigabitowe łącza szeregowej transmisji synchronicznej,
- elektryczne megabitowe łącza asymetryczne (np. w kasecie VME lub VXI) lub symetryczne krótkie połączenia kablowe LVDS.



Rys. 1. Ogólny model Systemu Fotonicznego na przykładzie urządzenia dwu- węzłowego

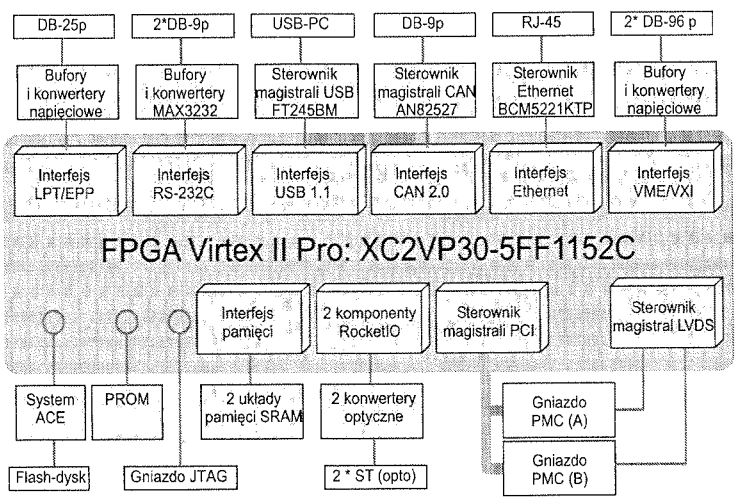
Fig. 1. Block diagram of Photonic System — example of two nodes of measurement network

Wspólna warstwa komunikacji komputerowej obejmuje standardową transmisję Ethernet, USB lub CAN, a w przemysłowej kasecie EURO także magistrale VME i VXI.

Urządzenia fotoniczne, opracowane zarówno w formie złożonych podsystemów jak i pojedynczych czujników, są integrowane z danymi węzłami MSF poprzez *interfejsy fotoniczne*. W celu umożliwienia przyłączenia do MSF różnych zestawów urządzeń fotonicznych, zrealizowano warstwę interfejsów w formie osadzanych, wymiennych modułów, w standardzie mechanicznym PMC. W następnym rozdziale krótko scharakteryzowano realizację pojedynczego węzła MSF.

3. WĘZEŁ SYSTEMU FOTONICZNEGO

Pojedynczy węzeł MSF opracowano w formie płyty bazowej PCB, w standardzie mechanicznym EURO-6HE-160 dostosowanym do kaset przemysłowych z interfejsem VME/VXI [11, 14]. Na rys. 2 przedstawiono jego ogólny schemat funkcjonalny.



Rys. 2. Ogólny schemat funkcjonalny pojedynczego węzła Systemu Fotonicznego

Fig. 2. Functional block diagram of a single node of Photonic System

Do realizacji zintegrowanego modułu sterowania wykorzystano układ programowalny FPGA Virtex II-Pro XC2VP30 [13]. Układ ten jest przedstawicielem rodziny układów FPGA najnowszej generacji, który zawiera m.in.:

- 30816 bloków logicznych (LCELL), gdzie każdy z bloków może realizować niezależnie dowolnie zaprogramowaną 4-wejściową funkcję logiczną,
- 136 wbudowanych układów mnożących 18×18 bitów realizujących pojedynczą operację w czasie 6-8ns. Szybkie układy mnożące umożliwiają implementację w FPGA złożonych algorytmów DSP pracujących w czasie rzeczywistym,
- 2448 kB pamięci o podwójnym dostępie (DUAL-PORT SRAM). Pamięć jest podzielona na 138 niezależnie konfigurowalnych bloków o pojemności 18kB każdy,

- 8 modułów dystrybucji szybkich sygnałów zegarowych (Digital Lock Manager — DCM), które umożliwiają reskalowanie częstotliwości bazowej oraz jej fazowanie,
- 8 modułów RocketIO realizujących gigabitową transmisji szeregową. Każdy z modułów może osiągnąć transfer danych do 3.125Gb/s. Rozwiązanie układowe pozwala na bezpośrednią integrację modułów RocketIO z konwerterami optycznymi [10]. Zastosowano powszechnie używane łącza światłowodowe typu ST.
- 2 wbudowane procesory Power PC 405 klasy RISC. Jednostka obliczeniowa procesora zawiera sprzętowe układy arytmetyczne oraz umożliwia definiowanie specjalnych instrukcji przez użytkownika [7]. Procesory mogą pracować z dystrybucją systemu operacyjnego LINUX, systemami czasu rzeczywistego lub innymi.
- 644 wyprowadzenia definiowane przez użytkownika do podłączenia zewnętrznych sygnałów elektrycznych. Dla poszczególnych nóżek, w sposób programowy, określa się właściwości elektryczne (np. wybór standardu typu TTL, LVDS, PCI itp.) oraz kierunek dystrybucji sygnału.

Zastosowanie dużego, współczesnego układu FPGA zawierającego powyższe zasoby, w praktyce umożliwiło skoncentrowanie wszystkich niezbędnych funkcjonalności pojedynczego węzła MSF w jednym programowalnym układzie scalonym. W powyższym kontekście, pozostałe elementy węzła stanowią wyłącznie dedykowane peryferia. Są one podzielone na cztery niezależne grupy:

1. peryferia systemowe, które zapewniają węzłowi wymaganą inicjalizację i konfigurację układu FPGA. W zależności o sposobu wykorzystania płyty bazowej, użytkownik może określić metodę, wybierając jedno z trzech poniższych rozwiązań:
   - *układ programowalny typu EPROM*, który zawiera stabilny program konfiguracyjny dla układu Xilinx [17]. Zawartość EPROMu może być modyfikowana przez wprowadzenie nowych danych interfejsem JTAG. To rozwiązanie zostało przygotowane przede wszystkim dla użytkownika końcowego, który w praktyce nie modyfikuje samodzielnie zawartości programu konfiguracyjnego FPGA. Zakłada się rzadkie modyfikacje konfiguracji EPROMu, które głównie będą wynikać z prowadzenia bieżącego serwisu węzła lub rozwoju jego funkcjonalności,
   - *system ACE* umożliwia przeprowadzenie procesów rozruchowych węzła MSF z zewnętrznego dysku typu FLASH osadzonego w slocie dysku [18]. Łatwa wymienność dysku pozwala na częste i proste w obsłudze zmiany konfiguracji układu XILINX. Ten rodzaj inicjalizacji przewidziano dla użytkownika, który samodzielnie i często przeprowadza zmiany w zawartości kodu układu FPGA,
   - *gniazdo JTAG* pozwala bezpośrednio dołączyć układ programujący i monitorujący do FPGA [19, 20]. Jest to rozwiązanie dedykowane pracom serwisowym, uruchomieniowym lub testowym. Zawartość konfiguracji FPGA jest tracona po wyłączeniu zasilania płyty i musi być ponownie wprowadzona przez urządzenie programujące.
2. peryferia komunikacyjne, które zapewniają węzłowi kontakt z urządzeniami zewnętrznymi, takimi jak komputery, inne węzły MSF, zewnętrzne urządzenie pomia-

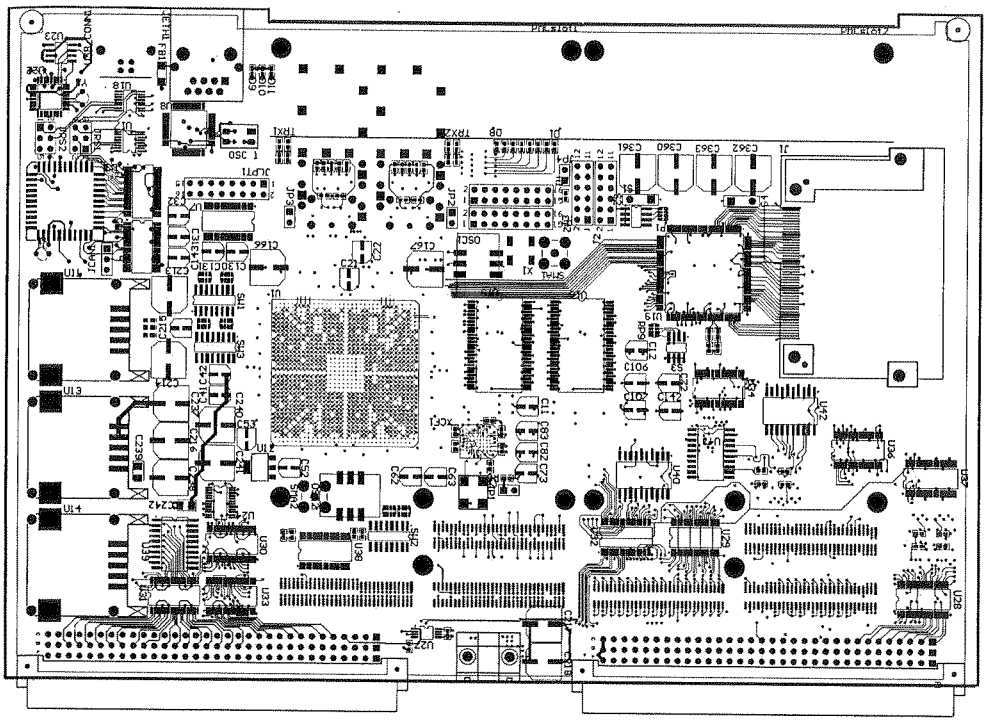
rowe itp. Przy realizacji interfejsów wykorzystano zarówno zasoby i właściwości układu FPGA, jak i dodatkowe układy peryferyjne, bufory i konwertery napięciowe. W skład grupy peryferii komunikacyjnych wchodzą:

- *interfejs VME/VXI*, który pozwala bezpośrednio dołączyć płytę do przemysłowej magistrali VME-bus lub VXI [11, 14]. Z tego względu konstrukcja mechaniczna płyty PCB węzła została dostosowana do rozmiarów standardowej płyty umieszczanej w przemysłowej kasecie EURO-6HE-160 wyposażonej w magistralę VME-bus lub VXI. Sprzętowa implementacja interfejsu VME-bus umożliwia węzłowi odpowiednio na pracę w modach: SLAVE, MASTER oraz jako sterownik kasety VME,
- *interfejs Ethernet 100TB* zapewnia połączenie węzła z innymi urządzeniami komputerowymi poprzez standardowe rozwiązania sieciowe. Obsługa warstwy sygnałowej jest realizowana przez sterownik BCM5221KTP firmy BROADCOM, a warstwa logiczna przez specjalizowany blok funkcjonalny oraz procesor Power PC w układzie Xilinx Virtex II Pro,
- *interfejs CAN 2.0* umożliwia bezpośrednie podłączenie węzła do pojedynczej magistrali CAN [15]. Przewiduje się wykorzystanie interfejsu do integracji MSF z pomocniczymi urządzeniami pomiarowo-kontrolnymi, głównie komercyjnymi,
- *interfejs USB 1.1* został zaimplementowany w celu bezpośredniego podłączenia węzła MSF do pojedynczego komputera użytkownika lub urządzenia peryferyjnego,
- *dwa interfejsy RS-232C oraz interfejs LPT/EPP.* charakteryzują się prostotą realizacji sprzętowej i duża popularnością. Pozwalają na integrację węzła z aparaturą pomiarową i urządzeniami peryferyjnymi starszego typu oraz tanimi w realizacji rozwiązaniami własnymi.

3. peryferia gigabitowej transmisji optoelektronicznej umożliwiają realizacje uniwersalnych połączeń węzłów MSF i budowę rozproszonej sieci urządzeń. Zapewniają szybką, synchroniczną transmisje danych, odporną na zakłócenia i przekłamania, nawet przy transferze informacji na znaczne odległości. Każdy węzeł jest wyposażony w dwa niezależne układy nadawczo-odbiorcze z wymiennymi konwerterami optycznymi. Obsługa warstwy sygnałowej oraz logicznej jest realizowana przez moduły RocketIO wbudowane w układ FPGA Virtex II Pro [10],
4. peryferia modułów osadzonych pozwalają na dołączanie do węzła dwóch płyt nakładkowych zrealizowanych w standardzie mechanicznym PMC [12]. Warstwa komunikacyjna obejmuje standardową magistralę PCI [16] w wersji 33MHz lub 66MHz oraz dedykowane szybkie linie LVDS konfigurowane przez użytkownika. Sterowniki obu magistral oraz parametry elektryczne nóżek sterujących są odpowiednio zaimplementowane i skonfigurowane w układzie FPGA.

Płytę bazową pojedynczego węzła MSF opracowano w standardzie mechanicznym 6HE-160 (233mm x 160mm), w formie jednolitej, 12-to warstwowej PCB na laminacie FR4 [21]. Osiem warstw przeznaczono na dystrybucję sygnałów, a pozostałe warstwy wykorzystano na rozprowadzenie płaszczyzn zasilań. Różnicowe linie dla sygnałów

gigabitowych łączących tranceivery optyczne z blokiem RocketIO w układzie FPGA poprowadzono na wewnętrznych warstwach PCB jako linie paskowe o impedancji $Z_0 = 100\Omega$ [22–24]. Konstrukcja płyty została zaprezentowana na rys. 3.



Rys. 3. Widok strony przedniej płyty bazowej węzła Systemu Fotonicznego

Fig. 3. Layout design of the upper side of PCB for a node of Photonic System

W części frontalnej płyty rozmieszczono krytyczne gniazda interfejsów, jak tranceivery optyczne, gniazdo Ethernet 100TB oraz USB. Pozostałe interfejsy są oddzielnie umieszczanie na front-panelu i odpowiednio łączone kablami z płytą. Przyjęte rozwiązanie pozwala na zachowanie minimalnej grubości płyty w celu jej wygodnego umieszczenia i podłączenia w innym urządzeniu lub obudowie.

Na stronie przedniej płyty rozmieszczone wszystkie większe elementy aktywne oraz pasywne. Dwa pozostałe gniazda PMC umieszczono na stronie tylniej. Rozmieszczenie elementów wynika bezpośrednio z przewidywanych sposobów wykorzystania płyty w kasecie VME jako:

1. Kontrolera kasety, gdzie jest wymagane jej umieszczenie w pierwszym slocie kasety EURO. Dla tej formy użytkowania płyty, jej dwa gniazda PMC na stronie tylnej nie będą montowane
2. Płyty użytkowej przeznaczonej do umieszczenia w każdym z pozostałych gniazd kasety. Płyta może pracować zarówno w modzie MASTER jak i SLAVE.

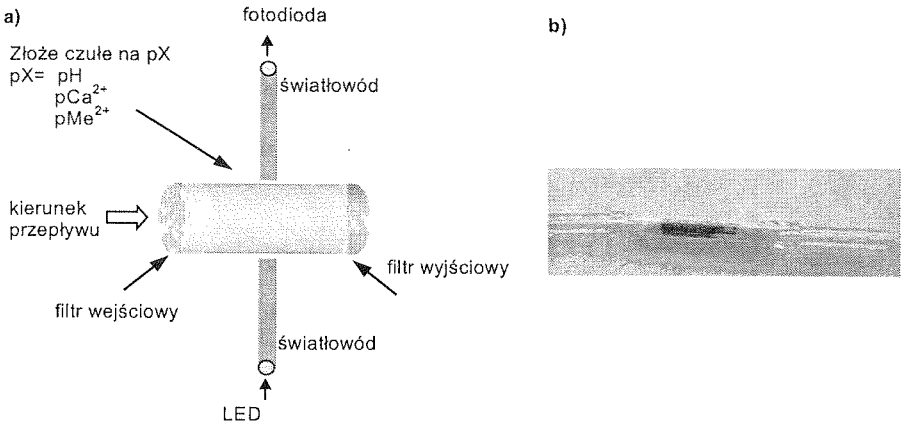
4. APLIKACJA SYSTEMU FOTONICZNEGO DO OCENY JAKOŚCI WODY PITNEJ

Funkcjonalność opracowanego rozwiązania konstrukcyjnego sprawdzono w trakcie badań wieloparametrowego systemu do oceny jakości wody pitnej [25, 26]. Czujnik ma służyć do monitorowania parametrów wody w warunkach środowiskowych. Oznaczanymi wielkościami przez system są: pH, stężenie jonów wapnia i metali ciężkich (jony cynku, kadmu, ołowiu i rtęci). Opracowano i wykonano warstwy receptorowe, których parametry optyczne (widmo absorpcji) zależą od zmian mierzonych wielkości. W oparciu o uzyskane materiały receptorowe opracowano założenia konstrukcyjne wieloparametrowego systemu pomiarowego, umożliwiającego akwizycję wyników pomiarów [27]. Konstruowane czujniki chemiczne zbudowane są w wersji przepływowej przystosowanej do monitorowania próbek wody w trybie on-line. Modułowa konstrukcja głowicy umożliwia jej łatwą konserwację oraz ewentualną wymianę czujników. Jej budowa zapewnia małe zużycie odczynników, a próbka pobierana do analizy może mieć małą objętość. Badana próbka dostarczana jest do głowicy przepływowej z wykorzystaniem pompy perystaltycznej. Budowa głowicy przepływowej pokazana jest na rys. 4.

W projektowanych czujnikach obszar aktywny, wykazujący zmiany absorbancji w zależności od mierzonego parametru chemicznego, wykonany jest w postaci mikrokulek polimerowych, na powierzchni których zaimmobilizowano odpowiedni wskaźnik chemiczny. Tak przygotowane kulki noszą nazwę złoża. Konieczne stało się opracowanie właściwej konstrukcji obudowy czujnika optycznego. Obudowa ta musi zapewnić przepływ analitu/roztworów kalibrujących, zapewnić trwałe utrzymanie złoża wewnątrz obudowy oraz umożliwić dostęp wiązki światła do złoża a następnie do układu detekcyjnego. Analizując możliwe rozwiązania techniczne oraz wykorzystując posiadane doświadczenie w pomiarach chemicznych, najbardziej właściwe wydaje się wykonanie obudowy czujnika w postaci rurki szklanej wypełnionej złożem.

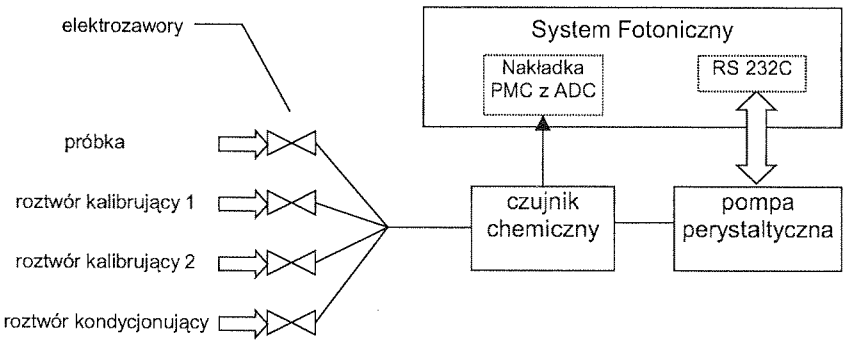
Schemat układu pomiarowego współpracującego z Systemem Fotonicznym pokazany jest na rys. 5. Wykorzystano interfejs komunikacyjny RS-232 do sterowania pompą perystaltyczną oraz dedykowaną nakładkę PMC z przetwornikiem A/C.

Procedura pomiarowa zakłada kalibrację każdego z czujników przed każdym pomiarem. W ten sposób eliminowany jest ewentualny wpływ temperatury na działanie czujników. Wyznaczona w warunkach laboratoryjnych charakterystyka kalibracji czujnika umożliwia przeprowadzenie skalowania wskazań na podstawie pomiarów wartości sygnału w dwóch roztworach kalibrujących o dokładnie znanym stężeniu badanego parametru. W pierwszym etapie do czujnika doprowadzany jest roztwór kalibrujący 1 i wartość sygnału jest zapamiętywana przez oprogramowanie. Następnie pompowany jest roztwór kondycjonujący i kolejny roztwór kalibrujący. Na podstawie tych dwóch pomiarów wyznacza się równanie prostej opisującej działanie czujnika. Po skalibrowaniu systemu można dokonać pomiaru próbki. Ustalenie się wartości sygnału powoduje wyświetlenie bieżącej wartości oznaczeń na ekranie monitora, po czym system prze-



Rys. 4. a) Schematyczna budowa głowicy przepływowej czujnika chemicznego, b) fotografia czujnika pH

Fig. 4. a) A schematic construction of the flow head of a chemical sensor, b) a photo of pH sensor



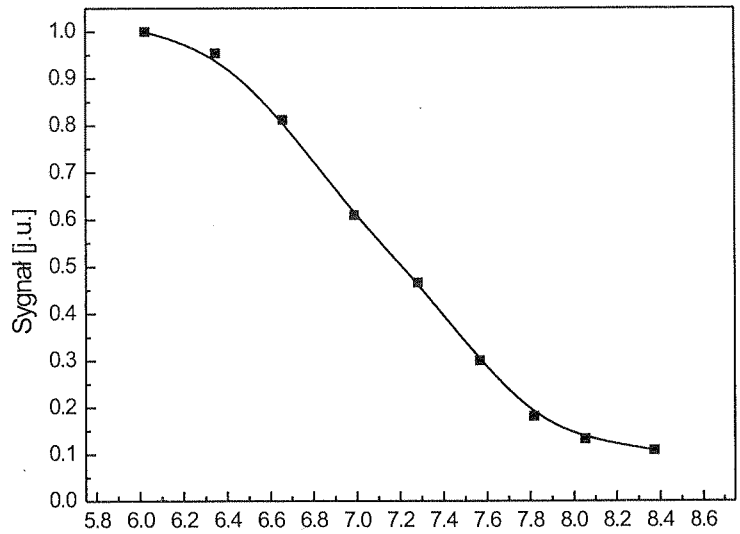
Rys. 5. Schematyczna budowa jednego kanału pomiarowego systemu do monitorowania jakości wody pitnej

Fig. 5. A schematic construction of a single measurement channel for potable water monitoring

chodzi do stanu stand-by, w którym przez czujnik przepływa roztwór kondycjonujący. Zadaniem roztworu kondycjonującego jest wstępna regeneracja czujnika chemicznego, wymycie ze złoża pozostałości próbki oraz utrzymanie złoża zawierającego zaimmobilizowany indykator w stanie gotowym do następnego pomiaru. Na rys. 6–8 pokazano przykładowe wyniki kalibracji czujników chemicznych.

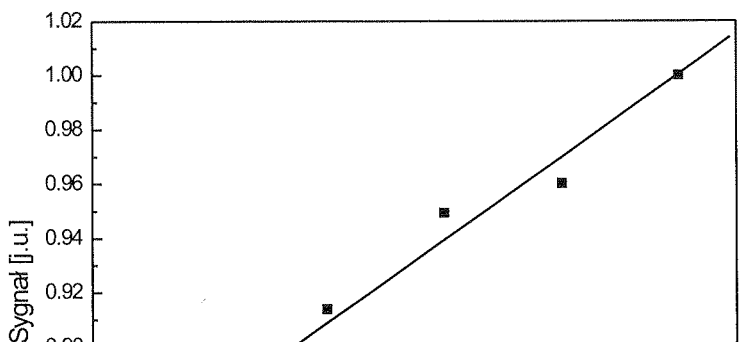
5. PODSUMOWANIE

W artykule syntetycznie przedstawiono koncepcje funkcjonalną oraz realizację sprzętową Modułowego Systemu Fotonicznego. Rozwiązanie strukturalne MSF obej-



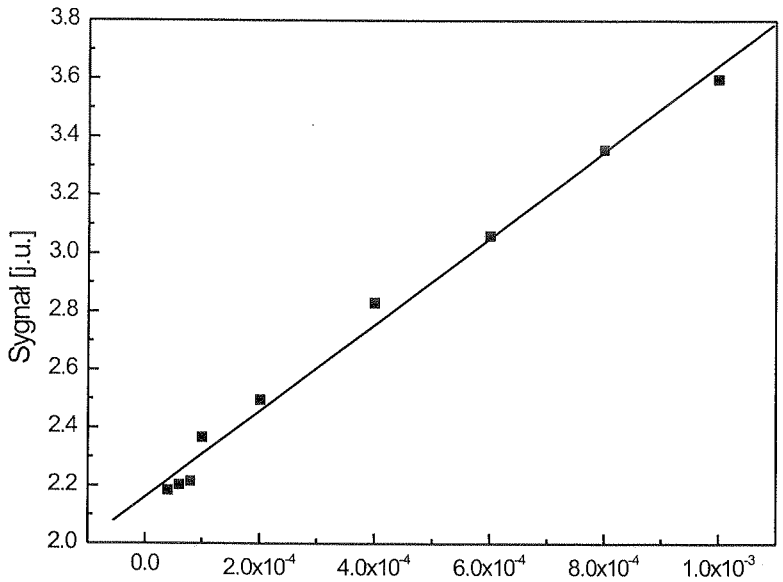
Rys. 6. Krzywa kalibracji czujnika pH (czerwień fenolowa immobilizowana na złożu XAD4)

Fig. 6. Calibration curve for pH sensor (fenol red immobilized on the basis of XAD 4)



Rys. 7. Krzywa kalibracji czujnika jonów wapnia
(chlorofosfonazo III immobilizowany na złożu IRA 400)

Fig. 7. Calibration curve for Ca ion sensor (chlorophosphanaso III immobilised on the basis of IRA 400)



Rys. 8. Krzywa kalibracji czujnika jonów metali ciężkich (4-(2-pirydylazo)-rezorcynol immobilizowany na złożu XAD 4).

Fig. 8. Calibration curve for heavy metal ion sensor (4-(2-pirydylazo)-rezorcynol immobilised on the basis of XAD 4

muje niezależne węzły połączone siecią gigabitowych łączy optycznych. Dokonano standaryzacji budowy węzła poprzez zastosowanie centralnego układu FPGA, implementację licznej grupy uniwersalnych interfejsów komunikacyjnych (Ethernet, USB, CAN, RS-232 i innych.), interfejsów przemysłowych (VME, VXI) oraz umieszczenie uniwersalnych gniazd PMC dla interfejsów fotonicznych.

Zróżnicowane wymagania funkcjonalne, wynikające z określonego zastosowania MSF, są realizowane w układzie programowalnym. Została zapewniona możliwość łatwej i wielokrotnej rekonfiguracji układu FPGA w celu zastosowania tego samego węzła w różnych konfiguracjach pracy i z różnymi urządzeniami fotonicznymi.

Wymienność urządzeń fotonicznych współpracujących z węzłem wynika z wykorzystania płyt nasadzanych PMC lub podłączania danych urządzeń do określonych portów komunikacyjnych węzła, co zaprezentowano na przykładzie aplikacyjnym zamieszczonym w rozdz. 4. Dostosowanie wymaganych funkcjonalności dla obsługi urządzeń fotonicznych lub ich interfejsów jest w głównej części realizowane w układzie programowalnym FPGA.

W celu umieszczenia węzła w przemysłowej kasecie VME/VXI, jego płytę PCB zrealizowano w standardzie mechanicznym EURO-6HE-160. To rozwiązanie pozwala zintegrować węzły MSF z innymi komercyjnymi modułami oraz nadrzędnym systemem sterowania kasety. Interfejs VME-bus zaimplementowany w układzie FPGA pozwala

na zastosowanie płyty węzła w roli sterownika kasety i integrację MSF z innych modułami osadzonymi w kasecie. Zastosowanie oddzielnego gniazda zasilania na płycie umożliwia jej umieszczenie w innym urządzeniu lub oddzielnej obudowie.

Omówiony w rozdz. 4 zestaw czujników, zintegrowany z pojedynczym węzłem systemu fotonicznego, służy do pomiarów laboratoryjnych. Prowadzone są prace rozwojowe nad budową dużej, rozproszonej struktury systemu fotonicznego dostosowanego do eksploatacji w warunkach półprzemysłowych. Poszczególne węzły systemu mogą realizować lokalnie procesy pomiarowe, przetwarzanie i akwizycję danych. Ich wyposażenie aparaturowe umieszczone na modułach rozszerzeń w standardzie PMC oraz oprogramowanie w warstwie sprzętu FPGA wraz z procesorem Power-PC będzie dostosowywane do aktualnych potrzeb. Konstrukcja węzła pozwala na łatwą wymianę sprzętu oraz oprogramowania również po instalacji całego systemu. Gigabitowe łącza optoelektroniczne zaimplementowane w każdym węźle (por. rozdz. 1) umożliwiają realizację redundantnej struktury sieciowej, w dużym stopniu odpornej na zakłócenia transmisji danych, a nawet lokalne uszkodzenia.

Docelowo omawiany system fotoniczny będzie mógł współpracować m.in. z siecią wodno-kanalizacyjną. Monitoring bieżących parametrów dla całej sieci oraz ich przesyłanie do dyspozytorni pozwala na efektywne zarządzanie siecią, a w przypadku zaistnienia nieprawidłowości na szybkie uruchomienie procedur awaryjnych. Zastosowanie sieci światłowodowej stanowi alternatywę dla połączeń realizowanych drogą radiową.

6. PODZIĘKOWANIA

Praca była częściowo finansowana z grantu PBZ-MIN-009/T11/2003 p.t.: „Elementy i Moduły Optoelektroniczne do zastosowań w medycynie, przemyśle, ochronie środowiska i technice wojskowej". Grant był koordynowany przez ITME-Warszawa oraz CTT-PW.

7. BIBLIOGRAFIA

1. http://www.xilinx.com/ [Xilinx Homepage].
2. http://www.altera.com/ [Altera Homepage].
3. U. Meyer-Baese: *Digital Signal Processing with Field Programmable Gate Arrays*, Wydanie drugie, Springer, 2004, ISBN: 3540211195.
4. http://www.altera.com/literature/technology/dsp/dsp-literature.pdf, *DSP Literature*, Altera Corporation, 2005.
5. A. Athavale, C. Christensen: *High-Speed Serial I/O Made Simple, A Designer's Guide with FPGA Applications*, Preliminary Edition, 2005, Xilinx Connectivity Solutions, PN0402399.
6. R.S. Romaniuk, K.T. Poźniak, G. Wrochna, S.Simrock: *Optoelectronics in TESLA, LHC, and pi-of-the-sky experiments*, Proc. SPIE Vol. 5576, 2005, pp. 299–309.
7. *Xilinx Viretx 2 Pro PowerPC 405 Processor Block Reference Guide*, Xilinx, Inc., http://direct.xilinx.com/bvdocs/userguides/ug018.pdf.

8. *Nios 3.0 CPU*, DS-NIOSCPU-2.1 Technical Note, Ver. 2.2, 2004, Altera Corporation, http://www.altera.com/literature/ds/ds_nios_cpu.pdf.
9. MicroBlaze Processor Reference Guide, UG081 (v5.3) October 5, 2005, Xilinx, Inc., http://www.xilinx.com/ise/embedded/mb_ref_guide.pdf.
10. Xilinx Virtex 2 Pro RocketIO Transceiver user Guide, UG035 (v1.5), 2004, Xilinx, Inc., http://direct.xilinx.com/bvdocs/userguides/ug035.pdf.
11. *VMEbus Card Form Factors*, http://www.interfacebus.com/Design_VME_Card_size.html.
12. *IEEE Standard Physical and Environmental Layers for PCI Mezzanine Cards (PMC)*, IEEE Std 1386.1-2001.
13. *Virtex-II Pro and Virtex-II Pro X Platform FPGAs:Complete Data Sheet*, 2005, Xilinx, Inc., http://direct.xilinx.com/bvdocs/publications/ds083.pdf.
14. W.D. Peterson: *The VMEbus Handbook*, 2nd edition, VITA.
15. CAN Specification 2.0 B, http://www.cancia.org/downloads/specifications/.
16. PCI SIG, PCI Local Bus Specification v2.3, PCI SIG, Portland, 2002.
17. *Platform Flash PROM User Guide*, UG161 (v1.0), 2005, Xilinx, Inc., http://direct.xilinx.com/bvdocs/userguides/ug161.pdf.
18. *System ACE Compact Flash Solution*, 2002, Xilinx, Inc., http://direct.xilinx.com/bvdocs/publications/ds080.pdf.
19. *JTAG Programmer Guide*, 1999, Xilinx, Inc., http://www.xilinx.com/support/sw_manuals/2_1i/download/jtag.pdf.
20. *ChipScope Pro Software and Cores User Guide (ChipScope Pro Software v8.1i)*, UG029 (v8.1), 2005, Xilinx, Inc., http://www.xilinx.com/ise/verification/chipscope_pro_sw_cores_8_1i_ug029.p df.
21. T. Williams: *The Circuit Designer's Companion*, 2nd edition, Elsevier, Oxford, 2005.
22. S.C. Thierauf: *High-Speed Circuit Board Signal Integrity*, Artech House, Norwood, 2004.
23. S.H. Hall, G.W. Hall: *High-Speed Digital System Design — A Handbook of Interconnect Theory and Design Practices*, John Wiley & Sons, New York, 2000.
24. H.W. Johnson, M. Graham: *High-Speed Digital Design: A Handbook of Black Magic*, Prentice Hall, New Jersey.
25. U.E. Spichiger-Keller: *Chemical Sensors and Biosensors for Medical and Biological Applications*, Willey-VCH, Weinheim, 1998 .
26. O.S. Wolfbeis (ed.): *Fiber optic chemical sensors and biosensors*, Vol. 1 and 2, CRC, Boca Raton, FL, 1991.
27. D. Stadnik, M. Chudy, W. Wroblewski, Z. Brzozka, A. Dybko: *Flow-through sensor system for environmental monitoring*, Eighth European Conference on Optical Chemical and Biosensors, Tuebingen, April 2-5, 2006, Germany.

A. DYBKO, R. GRACZYK, K. T. POŹNIAK, R. S. ROMANIUK

FPGA BASED MODULAR PHOTONIC SYSTEM WITH PROGRAMMABLE CONTROL AND COMMUNICATION LAYER

Summary

The paper concisely presents a functional idea and hardware realization of a Modular Photonic System. Solution of the system structure is composed of nondependent nodes connected by a network of gigabit optical links. The node construction was standardized via the application of a central, large FPGA chip, implementation of a considerable number of basic and universal communication interfaces (like Ethernet, USB, CAN, RS-232 and others), implementation of industrial interfaces (VME, VXI, ATCA in the future) and inclusion of universal PMC slots for photonic interfaces. Different functional requirements are realized in the FPGA chip. There is a possibility to reconfigure easily, reversibly and comparatively fast the FPGA chip. Tree sockets for universal embedded boards in the PMC standard were implemented.

This provides exchangeability of dedicated interfaces for various devices. The FPGA chip realizes also fitting of requested functionalities for servicing of photonic devices or their interfaces.

The work concerns application of the described system for quality determination of drinking water. A multi-parameter sensor was used for monitoring of drinking water parameters in industrial conditions. The following parameters are determined: pH, concentration of ions — calcium and heavy metals (zinc, cadmium, lead, mercury). The practical functionality of the constructed system has been checked during long term investigations. The calibration results for a number of used chemical sensors were presented. There are carried broad developmental works on a large, distributed, photonic network based, telemetric system fit to the requirements of industrial conditions. The aim is that the photonic telemetric system cooperates seamlessly with the large municipal water — sewage network. It enables monitoring of current system parameters in the municipal environment. Creation of maps for parameter distribution will be one of the aims.

Keywords: photonic devices, programmable Xilinx matrices, behavioral programming, VHDL parameterization, optical gigabit transmission, industrial communication interfaces

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 669–695

Projektowanie topografii systemów VLSI Cz. 3. Metody analityczne

ZBIGNIEW NAGÓRNY, ANDRZEJ KOS

Zbigniew Nagórny,
Studium Generale Sandomiriense,
Wyższa Szkoła Humanistyczno-Przyrodnicza w Sandomierzu,
ul. Krakowska 26, 27-600 Sandomierz,
e-mail: nagorny@interia.pl,
Andrzej Kos,
Akademia Górniczo-Hutnicza w Krakowie,
Katedra Elektroniki,
al. Mickiewicza 30, 30-059 Kraków,
e-mail: kos@agh.edu.pl

Otrzymano 2005.12.05
Autoryzowano 2006.02.06

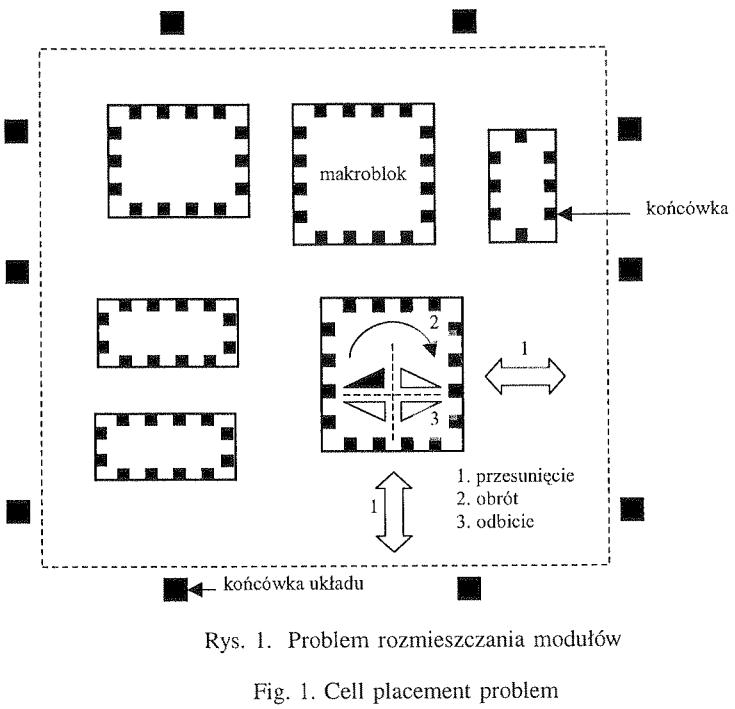
Niniejsza praca jest trzecią częścią przeglądu metod rozmieszczania modułów, stosowanych podczas projektowania topografii układów VLSI. W pracy szczegółowo został opisany algorytm zamiany parami oraz metody analityczne. Przedstawiono liczne modyfikacje algorytmu zamiany parami, łącznie z algorytmami wykorzystującymi metody relaksacyjne. Modyfikacje algorytmu zamiany parami oraz metody relaksacyjne są stosowane w programach rozmieszczania opartych na metodach analitycznych. Następnie, opisano podstawy zastosowania programowania kwadratowego i liniowego w rozmieszczaniu modułów. Ze względu na dużą liczbę rozwiązań stosowanych w metodach analitycznych, poszczególne rozwiązania szczegółowo przedstawiono na przykładzie wybranych programów rozmieszczania. W tym celu scharakteryzowano następujące programy rozmieszczania: GORDIAN / DOMINO, KraftWerk, FastPlace, mPL, PROUD, ATLAS, FAR, mFAR, BloBB, APlace. Przedstawiono również sposób zastosowania metody relaksacyjnej w układach o topografii swobodnej oraz możliwość optymalizacji topografii układu ze względu na aspekt termiczny.

Słowa kluczowe: VLSI, projektowanie topografii układu, układ scalony, układ Standard Cell, układ Sea-of-Gates, rozmieszczanie modułów, algorytm zamiany parami, metody relaksacyjne, relaksacja sterowana siłami, zamiana sterowana siłami, relaksacja parami sterowana siłami, algorytm Goto, metody analityczne, programowanie nieliniowe, programowanie kwadratowe, pro-

gramowanie liniowe, sieć przepływowa, wielopoziomowa optymalizacja, GORDIAN / DOMINO, KraftWerk, FastPlace, mPL, PROUD, ATLAS, FAR, mFAR, BloBB, APlace

1. WPROWADZENIE

Niniejsza praca stanowi trzecią część przeglądu metod rozmieszczania modułów, stosowanych podczas projektowania topografii układów VLSI. Rozmieszczanie modułów w układzie VLSI jest bardzo ważnym etapem projektowania jego topografii. Danymi wejściowymi dla metod rozmieszczania są kształty, rozmiary i położenie końcówek poszczególnych modułów układu, położenie końcówek całego układu oraz lista połączeń między modułami i końcówkami całego układu. Rozwiązanie problemu rozmieszczania polega na znalezieniu współrzędnych prostokątnych, określających położenie modułów w układzie tak, aby zapewnić spełnienie celów stawianych procesowi rozmieszczania. W układach o topografii wierszowej często zakłada się, że wszystkie końcówki modułów są położone w środku geometrycznym modułów, ponieważ rozmiary modułów są niewielkie w porównaniu do rozmiarów podłoża układu. W układach o topografii swobodnej położenie końcówek modułów (makrobloków) musi być uwzględnione, ponieważ poszczególne moduły mogą znacznie różnić się od siebie



Rys. 1. Problem rozmieszczania modułów

Fig. 1. Cell placement problem

rozmiarami i zajmować znaczną część podłoża układu. Uwzględniana jest również orientacja modułu (makrobloku). Każdy moduł może posiadać osiem różnych orienta-

cji, które są rezultatem obrotu oraz odbicia względem jego osi symetrii [29]. Rysunek 1 przedstawia problem rozmieszczania modułów.

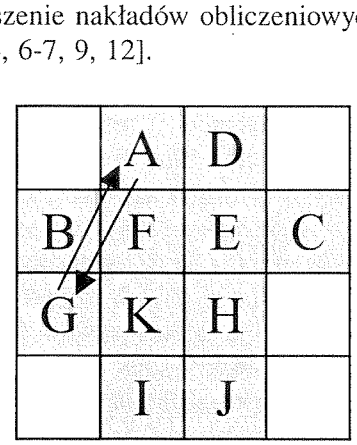
W pracy szczegółowo został opisany algorytm zamiany parami oraz metody analityczne (numeryczne). Opisano programy rozmieszczania, wykorzystujące przedstawione algorytmy. Algorytm zamiany parami jest zaliczany do algorytmów iteracyjnych. Istnieją jednak liczne modyfikacje algorytmu, które wykorzystują również metody relaksacyjne. Modyfikacje algorytmu zamiany parami oraz metody relaksacyjne są stosowane w programach rozmieszczania opartych na metodach analitycznych [13, 15, 23, 26-31, 35]. Metody analityczne są wykorzystywane do rozmieszczania modułów od wielu lat. Nadal zaliczane są do najczęściej stosowanych rozwiązań. W ostatnich latach pojawiło się wiele programów rozmieszczania, których podstawą są metody analityczne [10, 13-15, 21-23, 26-31, 35-37]. W rozdziale 2 przedstawiono algorytm zamiany parami oraz jego modyfikacje. Metody analityczne są opisane w rozdziale 3. Rozdział 4 zawiera charakterystykę obecnie stosowanych programów, których podstawą są metody analityczne. Podsumowanie trzeciej części przeglądu metod rozmieszczania zamieszczono w rozdziale 5. Bibliografia znajduje się w rozdziale 6.

2. METODA ITERACYJNA ZAMIANY PARAMI

Algorytm zamiany parami (ang. pairwise interchange algorithm) polega na zamianie miejscami par modułów. Zamiana modułów jest akceptowana, jeżeli powoduje zmniejszenie kosztu połączeń (estymowanego kosztu połączeń). W przeciwnym przypadku, zamiana modułów nie jest wykonywana. Zamiany wykonuje się dla różnych par modułów. Wybór modułów do zamiany może odbywać się losowo lub w wyniku systematycznego rozpatrywania wszystkich możliwych par modułów. W przypadku układu składającego się z n modułów, liczba wszystkich możliwych par modułów jest równa $n(n-1)/2$. Sprawdzenie wszystkich możliwych modułów składa się wtedy na jeden cykl algorytmu. Zamiany miejscami par modułów są kontynuowane do momentu, kiedy nie następuje dalsze zmniejszanie kosztu połączeń. Rysunek 2 ilustruje sposób postępowania w algorytmie zamiany parami. Dodatkowym założeniem jest jednakowy rozmiar wszystkich modułów oraz podział podłoża układu na siatkę prostokątnych części, posiadających takie same rozmiary jak moduły. Zakłada się również, że wszystkie końcówki modułów są położone w środku geometrycznym tej części podłoża układu, w której został umieszczony dany moduł [1, 3, 5-8, 11-12, 17].

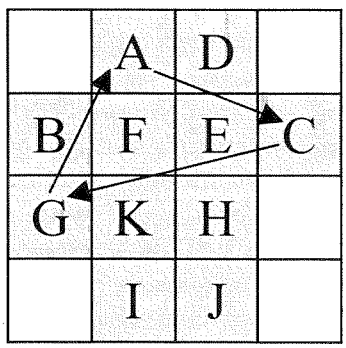
Możliwa jest również równoczesna zamiana miejscami λ modułów, gdzie $\lambda > 2$. Rysunek 3 ilustruje sposób wykonywania zamiany miejscami 3 modułów, dla $\lambda = 3$. Jeżeli w wyniku zamiany miejscami nie więcej niż λ modułów, nie jest możliwe uzyskanie rozwiązania o mniejszym koszcie połączeń, to otrzymane rozwiązanie nazywamy rozwiązaniem λ-optymalnym. Wzrost λ umożliwia osiągnięcie nieznacznie lepszych rezultatów, lecz powoduje znaczny wzrost nakładów obliczeniowych. Istnieje również modyfikacja algorytmu zamiany parami, polegająca na zamianie modułów znajdujących się w ε-sąsiedztwie (ang. neighborhood interchange). Podczas zamiany

miejscami rozpatrywane są jedynie moduły, które odległe są od siebie o nie więcej niż ε (w metryce Manhattan). Rozpatrzenie zamiany wszystkich możliwych modułów z modułami sąsiednimi składa się na jeden cykl algorytmu. Powyższe rozwiązanie umożliwia znaczne zmniejszenie nakładów obliczeniowych, jednak jakość rozwiązań jest wówczas gorsza [1, 3-4, 6-7, 9, 12].



Rys. 2. Algorytm zamiany parami

Fig. 2. Pairwise interchange algorithm



Rys. 3. Zamiana miejscami $\lambda = 3$ modułów

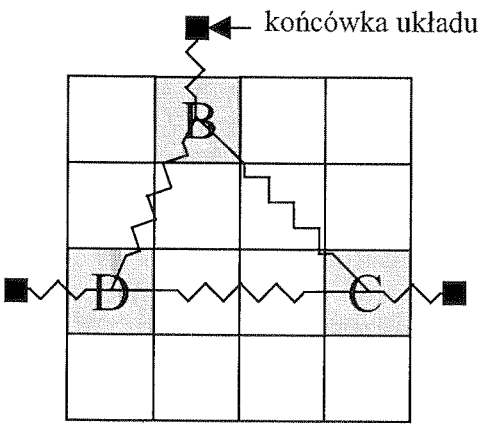
Fig. 3. Interchange of $\lambda = 3$ modules

Istnieją również modyfikacje metody zamiany parami, które wykorzystują metody relaksacyjne (ang. force-directed placement algorithms). W metodach relaksacyjnych rozpatrywany jest układ modułów, połączonych z użyciem sprężyn. Powyższą sytuację przedstawia rysunek 4. Zgodnie z prawem Hooke'a siła oddziaływania wzajemnego **f** dla każdej pary modułów jest określona wzorem

$$\mathbf{f} = k\,\mathbf{s} \tag{1}$$

gdzie: **f** — siła oddziałująca na dany moduł, k — stała sprężyny dla danej pary modułów, **s** — wektor wyznaczony przez uporządkowaną parę współrzędnych położenia danego modułu oraz modułu oddziałującego na ten moduł.

Stała sprężyny k dla danej pary modułów jest równa sumie wag połączeń między tymi modułami we wszystkich węzłach układu. Uwzględniane są również połączenia między modułami a końcówkami całego układu. W przypadku, gdy topologia połączeń w danym węźle nie jest z góry narzucona, mogą być rozpatrywane połączenia między wszystkimi końcówkami węzła (końcówki modułów i końcówki całego układu), tak jak w grafie pełnym. Waga każdego z tych połączeń może być wówczas równa $2w/n$, gdzie w jest wagą odpowiadającą danemu węzłowi, która umożliwia preferencję wybranych węzłów, a n jest liczbą końcówek węzła [1, 3-4, 11, 21, 24-28, 32, 35]. Całkowita siła oddziałująca na dany moduł jest równa sumie sił wszystkich modułów lub końcówek całego układu, które oddziałują na ten moduł. W metodach relaksacyjnych stosowane jest pojęcie punktu docelowego modułu, w którym wypadkowa sił oddziałujących na moduł jest równa **0**, przy ustalonym położeniu pozostałych modułów. Celem metod relaksacyjnych jest znalezienie położenia modułów, które odpowiada stanowi równowagi układu sprężyn. W stanie równowagi wypadkowe siły oddziałujące na poszczególne moduły są równe **0**, a energia układu osiąga minimum. Powyższy stan odpowiada minimalnemu kosztowi połączeń w układzie [1, 5-6, 9, 12].



Rys. 4. Rozmieszczanie modułów metodą relaksacyjną

Fig. 4. Force-directed placement

W algorytmie nazwanym relaksacją sterowaną siłami (ang. force-directed relaxation) moduły sortuje się według malejącego stopnia połączenia z innymi modułami. Następnie, pierwszy moduł z listy próbnie umieszcza się w jego punkcie docelowym, a dokładnie w tej części podłoża układu, która znajduje się najbliżej punktu docelowego modułu. Ta część podłoża układu nazywana jest miejscem docelowym modułu. Jeżeli w miejscu docelowym nie ma żadnego innego modułu, rozpatrywany jest na-

stępny moduł z listy. Jeżeli miejsce docelowe modułu jest zajęte, moduł znajdujący się wcześniej w tym miejscu jest zawsze dalej przenoszony. Przenoszenie kolejnych modułów jest wykonywane do momentu znalezienia wolnego miejsca dla ostatniego z modułów. Ostatecznie, nowe częściowe rozmieszczenie modułów jest akceptowane tylko w przypadku, gdy powoduje zmniejszenie kosztu połączeń. W przeciwnym razie, przywracane jest poprzednie rozmieszczenie modułów. Rozpatrzenie wszystkich modułów tworzy jeden cykl algorytmu. Moduły, które zostały już przeniesione w danym cyklu, nie są już rozpatrywane w dalszej części cyklu [1, 3, 6, 9, 12].

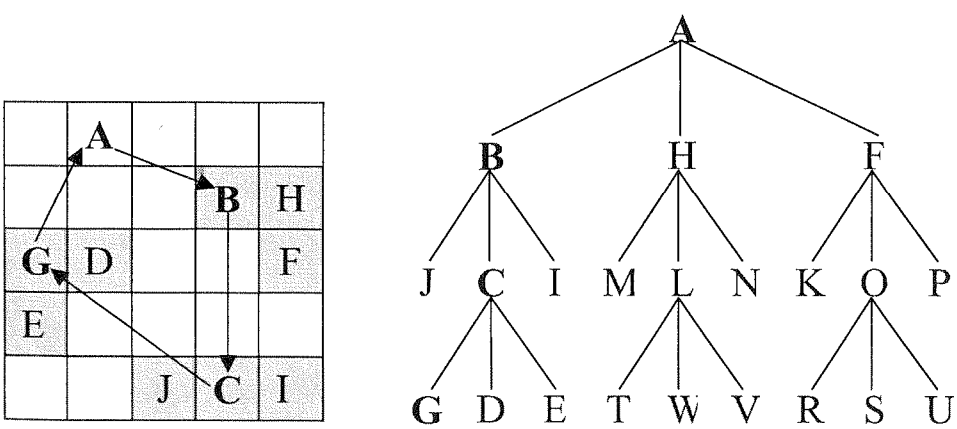
W algorytmie nazwanym zamianą sterowaną siłami (ang. force-directed interchange) dla kolejnych rozpatrywanych modułów, wykonywane są zamiany miejscami tych modułów z sąsiednimi modułami. Podczas zamiany brane są pod uwagę jedynie trzy sąsiednie moduły, zgodnie z kierunkiem i zwrotem wypadkowej siły oddziałującej na dany moduł. Zamiany dokonuje się w przypadku, gdy prowadzi do zmniejszenia kosztu połączeń. Po każdej zamianie ponownie wyznaczana jest siła oddziałująca na rozpatrywany moduł. Zamiany danego modułu z sąsiednimi modułami mogą być wykonywane wielokrotnie, do momentu, gdy nie następuje dalsze zmniejszanie kosztu połączeń. Rozpatrzenie wszystkich modułów układu kończy jeden cykl algorytmu [1, 3, 6, 12].

W algorytmie nazwanym relaksacją parami sterowaną siłami (ang. force-directed pairwise relaxation) dla kolejnych rozpatrywanych modułów A wykonywane są zamiany par modułów A i B, które posiadają tę własność, że miejsce docelowe modułu A znajduje się w ε-sąsiedztwie modułu B i odwrotnie. Zamiana jest akceptowana pod warunkiem, że prowadzi do zmniejszenia kosztu połączeń. Moduły zamienione miejscami nie są brane pod uwagę podczas dalszej części cyklu [1, 3, 6, 9, 12].

Podczas cyklu w algorytmach zamiany miejscami wykorzystujących metody relaksacyjne, moduły mogą być również rozpatrywane w kolejności losowej lub według malejącej wartości siły oddziałującej na moduł [9].

Jedną z modyfikacji algorytmu zamiany parami jest algorytm opracowany przez Goto (ang. generalized force-directed relaxation). Początkowe rozmieszczenie modułów jest wykonywane z użyciem algorytmu konstrukcyjnego. W algorytmie Goto koszt połączeń jest estymowany z wykorzystaniem metody połowy obwodu. Miejscem docelowym dla rozpatrywanego modułu A jest ta część podłoża układu, dla której estymowany koszt połączeń jest minimalny. Inaczej zdefiniowane jest również ε-sąsiedztwo miejsca docelowego modułu. Jest ono określone jako ε różnych położeń modułu, dla których koszt połączeń przyjmuje ε najmniejszych wartości. Drugi krok w algorytmie dla $\lambda = 2$, gdzie λ oznacza numer kroku, rozpoczyna się znalezieniem modułów, które znajdują się w ε-sąsiedztwie miejsca docelowego modułu A. Każdy z tych modułów jest próbnie zamieniany miejscami z modułem A. Jeżeli przynajmniej jedna z tych prób zapewnia zmniejszenie kosztu połączeń, to jedna z zamian, zapewniająca maksymalne zmniejszenie kosztu połączeń jest akceptowana i wykonywana. Zmiana kosztu w algorytmie jest zawsze wyznaczana w stosunku do kosztu połączeń po wykonaniu ostatniej zaakceptowanej zamiany miejscami modułów. Jeżeli żadna z próbnych zamian nie powoduje zmniejszenia kosztu połączeń, wybierana jest zamiana, która

powoduje najmniejszy wzrost kosztu połączeń. Załóżmy, że moduł A został umieszczony w miejscu, gdzie znajdował się wcześniej moduł B Trzeci krok w algorytmie dla $\lambda = 3$, rozpoczyna się znalezieniem modułów, które znajdują się w ε-sąsiedztwie miejsca docelowego modułu B. Wykonywane są próbne zamiany modułu B z modułami znajdującymi się w ε-sąsiedztwie miejsca docelowego modułu B. Zamiana, która zapewnia maksymalne zmniejszenie kosztu połączeń jest akceptowana i wykonywana. Jeżeli żadna z próbnych zamian nie powoduje zmniejszenia kosztu połączeń, wybierana jest zamiana, która powoduje najmniejszy wzrost kosztu połączeń. W takim przypadku, w następnym kroku dla $\lambda = 4$, rozpatrywane jest sąsiedztwo miejsca docelowego modułu C, który w poprzednim kroku został zamieniony miejscami z modułem B. Wartość λ w algorytmie jest ograniczona, ponieważ wraz ze wzrostem λ następuje lawinowy wzrost nakładów obliczeniowych. W przypadku, gdy zamiana miejscami określonej liczby modułów ostatecznie umożliwia zmniejszenie kosztu połączeń, zamiana tych modułów jest akceptowana. Kolejny rozpatrywany początkowy moduł jest określany na podstawie specjalnie tworzonego drzewa. Wierzchołkami drzewa są moduły, natomiast krawędzie określają możliwe zamiany miejscami między danym modułem a modułami znajdującymi się w ε-sąsiedztwie miejsca docelowego modułu. Zamiany miejscami grup modułów są wykonywane do momentu, gdy nie następuje dalsze zmniejszanie kosztu połączeń. Jeżeli nie jest możliwe znalezienie rozwiązania o mniejszym koszcie dla ustalonej wartości λ, wówczas mówimy, że rozwiązanie jest λ-optymalne. Algorytm może być wykonywany kilka razy dla różnych początkowych rozmieszczeń modułów. Goto podaje, że wartości $\lambda = 3 \div 4$ i $\varepsilon = 4 \div 5$ gwarantują zapewnienie kompromisu między jakością rozwiązań a czasem obliczeń [4, 6, 9, 11, 17]. Rysunek 5 ilustruje powyższy algorytm, kolorem szarym oznaczono ε-sąsiedztwo miejsc docelowych przenoszonych modułów.



Rys. 5. Algorytm Goto: a) zamiana miejscami $\lambda = 4$ modułów, b) drzewo przeszukiwania

Fig. 5. Generalized force–directed relaxation: a) interchange of $\lambda = 4$ modules, b) search tree

Algorytm zamiany parami oraz jego modyfikacje zwykle nie są stosowane samodzielnie. Algorytmy zamiany parami oraz metody relaksacyjne, wykorzystujące analogię połączeń między modułami do układu sprężyn, są bardzo często stosowane w programach rozmieszczania opartych na metodach analitycznych [13, 15, 23, 26–31, 35]. Należy dodać, że w wyniku modyfikacji algorytmu zamiany parami powstał również algorytm symulowanego wyżarzania, który jest powszechnie stosowany [1, 3, 5–6, 8–11, 13–15].

3. METODY ANALITYCZNE

Metody analityczne matematycznie wyznaczają położenie modułów. W tym celu stosowane są różne metody programowania matematycznego: programowanie nieliniowe (ang. nonlinear programming NLP), programowanie kwadratowe (ang. quadratic programming QP), które jest szczególnym przypadkiem programowania nieliniowego oraz programowanie liniowe (ang. linear programming LP) [3, 18–20]. Celem tych metod jest znalezienie minimum globalnego funkcji kosztu problemu rozmieszczania. W ten sposób wykonywane jest rozmieszczanie globalne modułów. Podczas rozmieszczania modułów z użyciem metod analitycznych występuje jednak problem zachodzenia modułów na siebie. Jest to spowodowane tym, że funkcja kosztu nie uwzględnia rozmiarów modułów oraz ograniczeń wynikających z wybrania konkretnego stylu topografii układu. Moduł traktowany jest jak punkt, w którym znajdują się wszystkie końcówki modułu. W celu rozwiązania powyższego problemu powszechnie stosowane są dwa rozwiązania: wykonywany jest odpowiedni podział układu lub wprowadzane są dodatkowe warunki (siły), które wymuszają konieczność rozwiązania nowego problemu optymalizacyjnego. W wyniku zastosowania tych rozwiązań, stopniowo następuje równomierne rozmieszczenie modułów na całym podłożu układu. W przypadku równomiernego rozmieszczenia modułów jest możliwe rozmieszczenie modułów bez zachodzenia na siebie, podczas rozmieszczania szczegółowego [9–11, 14, 21–23, 26–29, 32–33, 35–36].

Metody analityczne bardzo często wykorzystują kwadrat długości euklidesowej połączeń. W przypadku zastosowania kwadratu długości euklidesowej funkcja kosztu f, która jest ważoną sumą miar długości wszystkich połączeń w układzie, dana jest wzorem [1, 3, 6, 9–11, 21–22, 26–28, 30–33, 35]

$$f = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left((x_i - x_j)^2 + (y_i - y_j)^2 \right) \tag{2}$$

gdzie: w_{ij} — waga połączenia między modułami i oraz j, (x_i, y_i), (x_j, y_j) — współrzędne modułów i oraz j na podłożu układu, n — liczba modułów w układzie.
Udowodniono, że funkcję kosztu f można zapisać macierzowo w następującej postaci [1, 9, 11, 32–33]

$$f = \mathbf{x}^T \mathbf{B}\, \mathbf{x} + \mathbf{y}^T \mathbf{B}\, \mathbf{y} \tag{3}$$

gdzie: $\mathbf{x}$ — wektor n–wymiarowy współrzędnych x wszystkich modułów, $\mathbf{y}$ — wektor n-wymiarowy współrzędnych y wszystkich modułów, $\mathbf{B}$ — symetryczna macierz o wymiarze $n \times n$.

Jeżeli uwzględnimy połączenia modułów układu z końcówkami całego układu, które mają ustalone położenie w układzie, to funkcję kosztu f można zapisać macierzowo w następującej postaci [21, 26–28, 35]

$$f = \frac{1}{2}\mathbf{x}^T\mathbf{C}\mathbf{x} + \mathbf{d}_x^T\mathbf{x} + \frac{1}{2}\mathbf{y}^T\mathbf{C}\mathbf{y} + \mathbf{d}_y^T\mathbf{y} + \text{constans} \tag{4}$$

gdzie: $\mathbf{C}$ — symetryczna macierz o wymiarze $n \times n$, $\mathbf{d}_x$, $\mathbf{d}_y$ — wektory n-wymiarowe dla osi x oraz osi y.

Wektory $\mathbf{d}_x$, $\mathbf{d}_y$ odzwierciedlają połączenia modułów układu z końcówkami całego układu. Obydwa składniki funkcji kosztu f we wzorze 4 są niezależne od siebie. W związku z tym minimalizacja funkcji kosztu może być wykonana oddzielnie dla osi x oraz osi y [1, 9, 11, 21, 26–28, 32–33, 35]. Dla węzłów zawierających więcej niż 2 końcówki, dla których topologia połączeń nie jest z góry narzucona, mogą być stosowane połączenia między wszystkimi końcówkami węzła, tak jak w grafie pełnym [1, 3–4, 11, 21, 24–28, 32, 35].

3.1. PROGRAMOWANIE KWADRATOWE

Zadanie programowania nieliniowego polega na znalezieniu ekstremum globalnego funkcji celu, która jest określona na zbiorze wyznaczonym przez funkcje ograniczeń, zwanym zbiorem rozwiązań dopuszczalnych. W przypadku zadania programowania nieliniowego przynajmniej jedna z funkcji, tzn. funkcja celu lub jedna z funkcji ograniczeń jest funkcją nieliniową. Zadanie programowania kwadratowego jest szczególnym przypadkiem zadania programowania nieliniowego, w którym funkcja celu jest sumą formy liniowej i kwadratowej, natomiast ograniczenia są liniowe. Zadanie programowania kwadratowego polega na znalezieniu wektora $\mathbf{x}_0$, który minimalizuje (maksymalizuje) funkcję celu f [18-19]

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{D}\mathbf{x} + \mathbf{c}^T\mathbf{x} \tag{5}$$

pod warunkiem spełnienia następujących ograniczeń

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{6}$$

$$\mathbf{x} \geq \mathbf{0} \tag{7}$$

gdzie: $\mathbf{x}$, $\mathbf{c}$ — wektory n-wymiarowe, $\mathbf{b}$ — wektor m-wymiarowy, $\mathbf{A}$ — macierz o wymiarze $m \times n$, $\mathbf{D}$ — macierz o wymiarze $n \times n$, dodatkowo zakłada się, że macierz $\mathbf{D}$ jest symetryczna.

W przypadku braku ograniczeń otrzymujemy zadanie programowania kwadratowego bez ograniczeń (ang. unconstrained quadratic programming problem). Jeżeli macierz

D jest dodatnio określona (ang. positive definite) i funkcja celu f jest różniczkowalna, to rozwiązanie następującego układu równań jest jedynym minimum funkcji celu f, które jest jednocześnie minimum globalnym

$$2\mathbf{D}\,\mathbf{x} + \mathbf{c} = \mathbf{0} \tag{8}$$

Rozwiązanie powyższego układu równań jest rozwiązaniem zadania programowania kwadratowego bez ograniczeń [18-19].

Ze wzorów 4 i 5 wynika, że postać funkcji kosztu problemu rozmieszczania modułów dla jednej z osi układu współrzędnych jest taka sama, jak postać funkcji celu zadania programowania kwadratowego. Rozwiązanie zadania programowania kwadratowego bez ograniczeń jest równocześnie rozwiązaniem problemu rozmieszczania modułów dla jednej z osi układu współrzędnych. Rozwiązanie problemu rozmieszczania modułów sprowadza się zatem do rozwiązania układu równań. Rozwiązanie tego układu jest jedynym minimum funkcji kosztu problemu rozmieszczania, które jest jednocześnie minimum globalnym, ponieważ funkcja kosztu jest różniczkowalna, natomiast macierz **C** dla problemu rozmieszczania we wzorze 4 jest dodatnio określona. Powyższy układ musi być rozwiązany dla osi x oraz dla osi y. Istnieje wiele efektywnych metod rozwiązania powyższego układu równań. Dzięki temu, możliwe jest efektywne rozwiązanie problemu rozmieszczania globalnego modułów, nawet w przypadku bardzo dużych układów. Otrzymane rozwiązanie nie uwzględnia jednak rozmiarów modułów oraz ograniczeń wynikających z wybrania konkretnego stylu topografii układu, co powoduje m.in. występowanie zachodzenia modułów na siebie [3, 9–11, 14, 18–19, 21–22, 26–29, 32–33, 35–36]. W przypadku uwzględnienia powyższych ograniczeń funkcja kosztu może jednak posiadać wiele minimów lokalnych [3, 30].

3.2. PROGRAMOWANIE LINIOWE

Zadanie programowania liniowego polega na minimalizacji lub maksymalizacji liniowej funkcji celu w zbiorze określonym przez układ liniowych ograniczeń, tzn. równań lub nierówności liniowych. Zgodnie ze standardową (kanoniczną) postacią, zadanie programowania liniowego polega na znalezieniu wektora $\mathbf{x}_0$, który minimalizuje funkcję celu f [18–20]

$$f(\mathbf{x}) = \mathbf{c}\ \mathbf{x} \tag{9}$$

pod warunkiem spełnienia następujących ograniczeń

$$\mathbf{A}\ \mathbf{x} = \mathbf{b} \tag{10}$$

$$\mathbf{x} \geq \mathbf{0} \tag{11}$$

gdzie: **x**, **c** — wektory n-wymiarowe, **c x** — iloczyn skalarny wektorów **c** i **x**, **A** — macierz o wymiarze $m \times n$, **b** — nieujemny wektor m-wymiarowy, dodatkowo zakłada się, że $n \geq m$.

Możliwy jest również inny zapis zadania programowania liniowego, w którym zamiast ograniczenia wyrażonego wzorem 10, występuje ograniczenie $\mathbf{A}\ \mathbf{x} \geq \mathbf{0}$ lub $\mathbf{A}\ \mathbf{x} \leq \mathbf{0}$. Takie postacie zadania programowania liniowego można jednak sprowadzić do standardowej postaci zadania, w wyniku wprowadzenia nowych, nieujemnych zmiennych. Zadanie programowania liniowego można rozwiązać metodą simpleks (sympleksów) (ang. simplex method) [18–20].

Problem rozmieszczania modułów może być rozwiązany z wykorzystaniem programowania liniowego. Możliwe jest rozmieszczanie modułów w układach o topografii swobodnej, w której moduły posiadają dowolne rozmiary i położenie. Funkcja kosztu f problemu rozmieszczania modułów jest bardzo często ważoną sumą całkowitej estymowanej długości połączeń (metoda połowy obwodu) oraz całkowitej powierzchni układu, według wzoru [16, 40]

$$f = \alpha\, wh + \sum_{i=1}^{n} \left(w_i^w + h_i^w \right) \tag{12}$$

gdzie: w, h — szerokość oraz wysokość podłoża układu, w_i^w, h_i^w — szerokość oraz wysokość prostokąta obejmującego wszystkie końcówki węzła i, α — współczynnik określający udział całkowitej powierzchni układu w funkcji kosztu, n — liczba węzłów w układzie.

Powyższa funkcja kosztu nie jest funkcją liniową, dlatego funkcja f nie może być funkcją celu w zadaniu programowania liniowego. W celu rozwiązania tego problemu, zakłada się stałą wartość jednego z wymiarów podłoża układu, np. stałą szerokość podłoża równą W. Wówczas składnik $\alpha\ w\ h$ jest zastępowany przez liniową funkcję $\alpha\ W\ h$. Innym rozwiązaniem może być aproksymacja składnika $\alpha\ w\ h$ przez $\beta(w+h)$, gdzie β jest współczynnikiem zapewniającym dobrą aproksymację składnika [16, 40]. Zastosowana modyfikacja umożliwia użycie funkcji kosztu f jako funkcji celu w zadaniu programowania liniowego. Rozwiązanie zadania programowania liniowego powinno określać optymalne rozmieszczenie modułów w układzie o topografii swobodnej. W związku z tym rozwiązanie musi spełniać również szereg dodatkowych ograniczeń: moduły nie mogą na siebie zachodzić, moduły w całości muszą być położone na podłożu układu, moduły mogą posiadać orientację poziomą lub pionową. Powyższe ograniczenia są liniowe i można ich użyć w zadaniu programowania liniowego, jednak wymagają wprowadzenia całkowitych zmiennych. Możliwości zastosowania programowania liniowego dla większych układów są jednak ograniczone. Wraz ze wzrostem rozmiaru problemu następuje bardzo szybki wzrost czasu potrzebnego na przeprowadzenie obliczeń. Programowanie liniowe może być również stosowane podczas etapu planowania topografii układu [9, 16, 22, 36, 40].

4. PROGRAMY ROZMIESZCZANIA WYKORZYSTUJĄCE METODY ANALITYCZNE

Ze względu na bardzo dużą różnorodność rozwiązań stosowanych w metodach analitycznych, najczęściej stosowane rozwiązania zostaną przedstawione na przykładzie

programów rozmieszczania opartych na metodach analitycznych. Metody analityczne nadal są powszechnie stosowane, a w ostatnich latach pojawiło się wiele nowych programów rozmieszczania. W kolejnych podrozdziałach zostaną scharakteryzowane następujące programy rozmieszczania: GORDIAN / DOMINO, KraftWerk, FastPlace, mPL, PROUD, ATLAS, FAR, mFAR, BloBB, APlace. Program DOMINO wykorzystuje teorię sieci przepływowej, jest zalecany w programach rozmieszczania opartych na metodach analitycznych, szczególnie w przypadku programów GORDIAN i KraftWerk. Zostanie opisany również sposób zastosowania metody relaksacyjnej w układach o topografii swobodnej oraz problem optymalizacji topografii układu ze względu na aspekt termiczny. Przedstawione programy rozmieszczania mogą być stosowane w układach o różnych stylach topografii [2, 9-11, 13-15, 21-23, 26-37].

4.1. GORDIAN / DOMINO

W programie GORDIAN rozmieszczanie modułów polega na rozwiązaniu ciągu zadań programowania kwadratowego, z coraz większą liczbą warunków ograniczających swobodę zmiany położenia modułów. W ten sposób wyznaczane jest równomierne rozmieszczenie modułów, posiadające minimalną długość połączeń. Ograniczenia są rezultatem podziału podłoża układu na części, który odbywa się po rozwiązaniu każdego z kolejnych zadań programowania kwadratowego. Podczas rozwiązywania następnego zadania programowania kwadratowego, środek ciężkości modułów położonych w powstających częściach układu musi pokrywać się ze środkiem geometrycznym tych części. Każdy podział układu określa więc ograniczenia dla następnego zadania programowania kwadratowego. Metoda programowania kwadratowego umożliwia rozmieszczanie globalne modułów, które są równocześnie rozmieszczane we wszystkich częściach układu. Etap rozmieszczania globalnego modułów kończy się w momencie, gdy w każdej z części układu znajduje się co najwyżej k modułów, gdzie k jest stałą o ustalonej wartości. Następnie, odbywa się rozmieszczanie szczegółowe modułów dla wybranego stylu topografii układu [21].

Funkcja celu zadania programowania kwadratowego wykorzystuje kwadrat długości euklidesowej połączeń. Dla każdego węzła układu są rozpatrywane połączenia między wszystkimi końcówkami węzła, tak jak w grafie pełnym. Waga każdego z tych połączeń jest równa $2w/n$, gdzie w jest wagą odpowiadającą danemu węzłowi, która umożliwia preferencję wybranych węzłów, a n jest liczbą końcówek w węźle. Postać funkcji celu określona jest wzorem 4, dlatego problem rozmieszczania może być rozwiązany oddzielnie dla osi x oraz osi y. Ograniczenia powstają w wyniku podziału układu. Kolejne podziały układu posiadają przypisany poziom l, będący kolejną liczbą naturalną, poczynając od poziomu $l = 0$ dla całego układu. Podczas podziału układu na danym poziomie, wszystkie części układu są dzielone na dwie części. W wyniku podziału układu na poziomie l, układ składa się więc z $q = 2^l$ części, pod warunkiem, że wszystkie części układu z poprzednich poziomów podziału zostały dalej podzielo-

ne. Ograniczenia na kolejnych poziomach podziału układu dla osi x można wyrazić w następujący sposób

$$\mathbf{A}^{(l)}\mathbf{x} = \mathbf{b}^{(l)} \tag{13}$$

gdzie: $\mathbf{A}^{(l)}$ — macierz dla poziomu l podziału układu o wymiarze $q \times n$, $\mathbf{x}$ — wektor n-wymiarowy współrzędnych x modułów, $\mathbf{b}^{(l)}$ — wektor q-wymiarowy współrzędnych x środków geometrycznych części układu, dla poziomu l podziału układu.
Wartości elementów macierzy $\mathbf{A}^{(l)}$ zależą od powierzchni modułów w poszczególnych częściach układu. Rozwiązanie problemu rozmieszczenia modułów wymaga więc rozwiązania zadania programowania kwadratowego z ograniczeniami, które w programie GORDIAN jest przekształcane na zadanie programowania kwadratowego bez ograniczeń. Zadanie programowania kwadratowego bez ograniczeń jest rozwiązywane z użyciem metody gradientu sprzężonego (ang. conjugate gradient method) [18, 21].

Na każdym poziomie podziału układu, wszystkie części układu są dzielone na dwie części, pod warunkiem, że dana część zawiera więcej niż k modułów, gdzie k jest stałą o ustalonej wartości. Poszczególne części układu na kolejnych poziomach podziału układu najczęściej są dzielone na przemian liniami pionowymi i poziomymi. Podczas podziału danej części układu, stosunek sumarycznych powierzchni modułów w obydwu powstających częściach jest z góry ustalony. Zwykle jest stosowany podział na dwie części o równych lub zbliżonych sumarycznych powierzchniach modułów. Podczas podziału danej części układu, przynależność modułów do obydwu powstających części jest wyznaczana na podstawie sortowania odpowiedniej współrzędnej modułu. Dla podziału liniami pionowymi moduły są sortowane według współrzędnej x, dla linii poziomych według współrzędnej y. Do jednej z powstających części wybierana jest odpowiednia ilość modułów o najmniejszej współrzędnej x lub y, pozostałe moduły są przypisywane do drugiej części. W programie GORDIAN stosowane są dodatkowo trzy modyfikacje podziału układu, których celem jest minimalizacja szkodliwego wpływu podziału układu na jakość rozmieszczania szczegółowego modułów [21].

Rozmieszczanie szczegółowe modułów odbywa się po wykonaniu rozmieszczania globalnego i wymaga uwzględnienia stylu topografii układu. Program GORDIAN można stosować dla układów Standard Cell, Sea-of-Gates, a także dla układów o topografii swobodnej. W przypadku układów Standard Cell, moduły są najpierw dzielone na wiersze, według współrzędnej y modułów. Kolejność modułów w wierszu jest ustalana na podstawie współrzędnej x modułów. Program GORDIAN umożliwił otrzymanie rozwiązań o znacznie lepszej jakości i w znacznie krótszym czasie, w porównaniu do ówczesnych programów, wykorzystujących algorytm min-cut oraz symulowane wyżarzanie. Dotyczy to zarówno całkowitej długości połączeń, jak również całkowitej powierzchni układu po wyznaczeniu wszystkich połączeń [21].

Istnieje również odmiana programu o nazwie GORDIANL. W programie GORDIANL długość połączeń jest wyznaczana z użyciem metryki Manhattan. Nie jest stosowany kwadrat długości euklidesowej, który kładzie większy nacisk na minimalizację długich połączeń [3, 14, 22, 31]. W przypadku zastosowania liniowej funkcji kosztu nie jest jednak możliwe bezpośrednie zastosowanie efektywnych metod progra-

mowania kwadratowego. Możliwość zastosowania metod programowania liniowego jest jednak dla większych układów ograniczona, ze względu na bardzo długi czas obliczeń. W związku z tym, w programie GORDIANL zastosowano odpowiednie przekształcenie liniowej funkcji kosztu, które umożliwia minimalizację funkcji kosztu z użyciem efektywnych metod programowania kwadratowego. W każdym węźle układu moduły są łączone w gwiazdę. Wszystkie moduły należące do danego węzła są połączone z dodatkowym punktem. Punkt ten jest położony w środku geometrycznym prostokąta, obejmującego wszystkie końcówki należące do danego węzła. Ponadto, w programie GORDIANL zmodyfikowano również sposób podziału podłoża układu. Liniowa funkcja kosztu oraz nowa metoda podziału układu zapewnia otrzymanie lepszych wyników. W przypadku zastosowania programu GORDIANL dla układów Standard Cell, całkowita powierzchnia układu po wyznaczeniu wszystkich połączeń jest do 20% mniejsza, w porównaniu do programu GORDIAN. Lepsze rezultaty otrzymywane z użyciem programu GORDIANL są możliwe dzięki mniejszej długości połączeń węzłów zawierających dwie i trzy końcówki, w porównaniu do programu GORDIAN. Powyższe węzły posiadają znaczący udział w całkowitej długości połączeń [22].

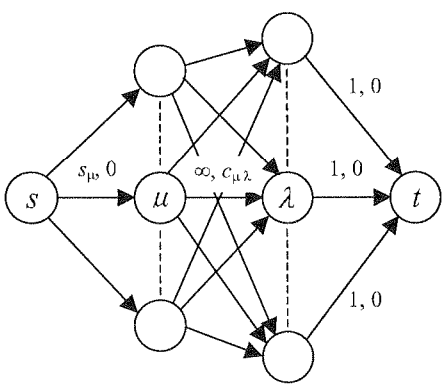
4.1.1. Domino

Rozmieszczenie globalne modułów, otrzymane w wyniku zastosowania programów GORDIAN lub GORDIANL, może być poddane dalszej iteracyjnej korekcji. Powyższe rozwiązanie zostało zastosowane w programie DOMINO i umożliwia otrzymanie lepszych rozwiązań. Program DOMINO jest stosowany dla układów o topografii wierszowej. Początkowym rozmieszczeniem modułów w programie DOMINO jest rozmieszczenie globalne modułów, wykonane tak jak w programie GORDIANL. W programie DOMINO problem rozmieszczania modułów jest sformułowany jako problem transportowy, który jest rozwiązywany z użyciem sieci przepływowej (ang. flow network) [23-25].

Początkowe rozmieszczenie modułów jest korygowane podczas kolejnych iteracji, do momentu, gdy w kolejnych kilku iteracjach nie następuje już dalsza wyraźna poprawa jakości rozmieszczenia. Podłoże układu jest dzielone na prostokątne regiony, które częściowo zachodzą na siebie (pokrywają się ze sobą). Każda iteracja polega na kolejnym rozwiązywaniu podproblemów, dla wszystkich regionów układu. W każdym podproblemie rozmieszczane są moduły, znajdujące się w danym regionie. Regiony są rozpatrywane kolejno od lewej do prawej strony układu. Każdy moduł w regionie jest dzielony na mniejsze części składowe, co umożliwia rozmieszczanie modułów o różnej szerokości (układ Standard Cell). Położenie wszystkich części składowych modułów danego regionu jest wyznaczane równocześnie, w wyniku rozwiązania problemu transportowego. Części składowe modułów w danym regionie mogą zajmować tylko ściśle określone położenia, które są wyznaczane na podstawie dotychczasowego rozmieszczenia. W ten sposób następuje pakowanie modułów od lewej do prawej strony układu tak, aby nie występowały zachodzenia modułów na siebie oraz wolne

przestrzenie między nimi. Zachodzenie regionów układu na siebie umożliwia zmianę przydziału modułów do poszczególnych regionów, który na początku został określony w wyniku rozmieszczania globalnego modułów [23].

Rozmieszczenie części składowych modułów w danym regionie polega na rozwiązaniu problemu transportowego. Koszt umieszczenia dowolnej części składowej modułu μ w jednym z dopuszczalnych położeń λ, określa specjalna funkcja. Problem transportowy jest równoważny problemowi maksymalnego przepływu o minimalnym koszcie (ang. minimum-cost maximum flow problem), w sieci przepływowej przedstawionej na rysunku 6. Węzeł s jest źródłem (ang. source), natomiast węzeł t jest ujściem (ang. sink). Węzeł μ odpowiada modułowi μ, który jest jednym z modułów położonych w rozpatrywanym regionie. Węzeł λ jest jednym z dopuszczalnych położeń części składowych modułów. Każda krawędź sieci przepływowej jest opisana pojemnością (ang. capacity) i kosztem. Pojemność krawędzi s_μ jest równa liczbie części składowych modułu μ. Koszt krawędzi $c_{\mu\lambda}$ jest wyznaczany na podstawie estymacji całkowitej długości połączeń wszystkich węzłów, do których należy moduł μ, w przypadku umieszczenia jednej z części składowych tego modułu w położeniu λ. Koszt $c_{\mu\lambda}$ musi być niezależny od położenia wszystkich pozostałych modułów w rozpatrywanym regionie, ponieważ położenie tych modułów nie jest znane podczas wyznaczania tego kosztu. Z tego względu stosowane są specjalne metody estymacji długości połączeń. Pojemność wszystkich krawędzi prowadzących do ujścia t jest równa 1, ponieważ każde dopuszczalne położenie może pomieścić tylko jedną część składową modułów. Rozwiązanie problemu maksymalnego przepływu o minimalnym koszcie, wyznacza położenie części składowych wszystkich modułów w rozpatrywanym regionie. Części składowe danego modułu zwykle są położone obok siebie. W programie DOMINO problem maksymalnego przepływu o minimalnym koszcie jest rozwiązywany z użyciem metody Forda-Fulkersona. Powyższy problem jest uważany za szczególny przypadek zadania programowania liniowego [23–25].



Rys. 6. Sieć przepływowa dla problemu rozmieszczania modułów

Fig. 6. Flow network for the cell placement problem

Po wykonaniu iteracyjnej korekcji rozmieszczenia modułów, wykonywane są zamiany parami sąsiednich modułów w wierszach. Każda zamiana jest akceptowana w przypadku, gdy powoduje zmniejszenie estymowanego kosztu połączeń. DOMINO jest programem typu variable-die. Program DOMINO umożliwia otrzymanie rozwiązań, dla których całkowita powierzchnia układu po wyznaczeniu wszystkich połączeń jest około 6% mniejsza, w porównaniu do programu GORDIANL. Dodatkowe obliczenia w programie DOMINO powodują umiarkowany wzrost czasu obliczeń [23].

4.2. KRAFTWERK

Funkcja kosztu problemu rozmieszczania w programie KraftWerk wykorzystuje kwadrat długości euklidesowej połączeń. Dla każdego węzła układu są rozpatrywane połączenia między wszystkimi końcówkami węzła, tak jak w grafie pełnym. Postać funkcji kosztu określona jest wzorem 4, dlatego można wykorzystać metodę programowania kwadratowego. Problem rozmieszczania modułów można rozwiązać oddzielnie dla osi x oraz osi y, zgodnie ze wzorem 8. Jeżeli zastosujemy analogię połączeń między modułami do układu sprężyn, to rozwiązanie problemu rozmieszczania modułów o najmniejszym koszcie odpowiada stanowi równowagi układu sprężyn. Powyższa analogia stała się motywacją do wprowadzenia dodatkowych sił oddziałujących na moduły. Celem dodatkowych sił jest zapewnienie równomiernego rozmieszczenia modułów na podłożu układu. Po wprowadzeniu dodatkowych sił oraz uwzględnieniu postaci funkcji kosztu (1/2 w formie kwadratowej), rozwiązanie problemu rozmieszczenia modułów dla osi x wymaga rozwiązania następującego układu równań

$$\mathbf{C}\,\mathbf{x} + \mathbf{d}_x + \mathbf{e}_x = \mathbf{0} \tag{14}$$

gdzie: $\mathbf{x}$ — wektor n-wymiarowy współrzędnych x wszystkich modułów, $\mathbf{C}$ — symetryczna macierz o wymiarze $n \times n$, dodatnio określona, $\mathbf{d}_x$ — wektor n-wymiarowy dla osi x, $\mathbf{e}_x$ — wektor n-wymiarowy współrzędnych x dodatkowych sił oddziałujących na moduły.
Ze wzoru 14 wynika, że rozwiązanie problemu rozmieszczenia modułów wymaga najpierw wyznaczenia wektora $\mathbf{e}_x$ [26–27].

W programie KraftWerk wprowadzono następujące ograniczenia na dodatkowe siły, oddziałujące na moduły:

a. siła oddziałująca na moduł zależy jedynie od jego współrzędnych na podłożu układu
b. regiony układu o dużym skupieniu modułów są źródłem sił, oddziałujących na moduły położone w regionach o małym skupieniu
c. pole dodatkowych sił jest polem potencjalnym (zachowawczym), tzn. istnieje skalarna funkcja, której gradient wyznacza wektor dodatkowych sił
d. w nieskończoności wartość dodatkowych sił jest równa 0.

Powyższe ograniczenia prowadzą do równania Poissona, które umożliwia wyznaczenie wektora dodatkowych sił [20, 26-27].

Rozmieszczanie globalne modułów jest wyznaczane iteracyjnie. Na początku pierwszej iteracji wszystkie moduły są położone w środku podłoża układu, natomiast wektor dodatkowych sił $\mathbf{e}_x$ jest równy wektorowi zerowemu. Następnie, jest wyznaczane rozmieszczenie modułów, zgodnie ze wzorem 14. Na początku każdej następnej iteracji wyznaczany jest wektor dodatkowych sił $\mathbf{e}_x$, na podstawie bieżącego rozmieszczenia modułów. Wektor $\mathbf{e}_x$ nie ulega zmianie do końca danej iteracji, umożliwiając wyznaczenie nowego rozmieszczenia modułów według wzoru 14. Kolejne iteracje są wykonywane dopóki w obszarze podłoża układu istnieje kwadrat wolnej przestrzeni, którego powierzchnia jest ponad cztery razy większa od średniej powierzchni wszystkich modułów. W programie KraftWerk układ równań określony wzorem 14 jest rozwiązywany z użyciem metody gradientu sprzężonego [26–27].

KraftWerk jest programem typu fixed-die. KraftWerk nie zapewnia jednak całkowitej eliminacji zachodzenia modułów na siebie. Z tego względu, do rozmieszczania szczegółowego modułów w układach Standard Cell, zalecany jest program DOMINO. Program KraftWerk / DOMINO umożliwia otrzymanie rozwiązań, których całkowita estymowana długość połączeń jest około 6% mniejsza, w porównaniu do programu GORDIANL / DOMINO, przy umiarkowanym wzroście czasu obliczeń. Program KraftWerk umożliwia rozmieszczanie modułów z minimalizacją opóźnień występujących w układzie oraz może być stosowany w układach o topografii mieszanej [13, 26-27].

4.3. FASTPLACE

FastPlace jest programem przeznaczonym dla układów Standard Cell. Rozmieszczanie modułów składa się z trzech etapów. Podczas kolejnych iteracji pierwszego etapu jest wyznaczane wstępne rozmieszczenie globalne modułów. W każdej iteracji kolejno odbywa się rozmieszczanie modułów metodą programowania kwadratowego, przesuwanie modułów (ang. cell shifting) oraz wyznaczenie dodatkowych sił (ang. spreading forces). Kolejne iteracje pierwszego etapu są wykonywane do momentu, kiedy moduły są prawie równomiernie rozmieszczone na podłożu układu. Podczas każdej iteracji drugiego etapu kolejno odbywa się rozmieszczanie modułów metodą programowania kwadratowego, iteracyjna korekcja rozmieszczenia modułów (ang. iterative local refinement), przesuwanie modułów oraz wyznaczenie dodatkowych sił. Kolejne iteracje drugiego etapu są wykonywane do momentu, kiedy moduły są równomiernie rozmieszczone na podłożu układu. Trzeci etap stanowi rozmieszczanie szczegółowe modułów. Podczas każdej iteracji trzeciego etapu odbywa się minimalizacja długości połączeń z użyciem algorytmu zachłannego oraz ostateczne rozmieszczanie modułów, zgodne z granicami wierszy i bez zachodzenia modułów na siebie. Kolejne iteracje trzeciego etapu wykonywane są do momentu, kiedy nie występuje już dalsza redukcja całkowitej długości połączeń [28].

Rozmieszczanie modułów metodą programowania kwadratowego ma na celu znalezienie rozwiązania posiadającego minimalną całkowitą długość połączeń. Funkcja

kosztu problemu rozmieszczania wykorzystuje kwadrat długości euklidesowej połączeń. Postać funkcji kosztu określona jest wzorem 4, dlatego można zastosować metodę programowania kwadratowego. Problem rozmieszczania modułów jest rozwiązywany zgodnie ze wzorem 14, oddzielnie dla osi x oraz osi y — podobnie jak w programie KraftWerk. Zadanie programowania kwadratowego bez ograniczeń jest rozwiązywane metodą gradientu sprzężonego. W programie FastPlace zastosowano specjalny model dla połączeń końcówek danego węzła. Dla węzłów zawierających dwie i trzy końcówki stosowane są połączenia typu każdy z każdym, tak jak w grafie pełnym. Dla węzłów zawierających więcej końcówek stosowane jest połączenie w gwiazdę. Dla dodatkowego punktu gwiazdy wprowadzane są dwie dodatkowe zmienne (dla osi x oraz osi y). Powyższe rozwiązanie ma wpływ na postać macierzy $\mathbf{C}$ i wektora $\mathbf{d}_x$ we wzorze 4, umożliwiając znaczne zmniejszenie czasu potrzebnego na znalezienie rozwiązania [28].

Program FastPlace stosuje podział podłoża układu na siatkę prostokątnych regionów (ang. bin). Przesuwanie modułów jest stosowane podczas pierwszych dwóch etapów. Przesuwanie modułów ma na celu osiągniecie równomiernego rozmieszczenia modułów w regionach, przy zachowaniu wzajemnego położenia modułów, które wcześniej zostało określone metodą programowania kwadratowego. Dla każdego regionu podłoża układu jest wyznaczane wykorzystanie (ang. utilization) danego regionu, które jest równe stosunkowi sumy powierzchni modułów położonych w danym regionie do całkowitej powierzchni regionu. Przesuwanie modułów odbywa się w kolejnych wierszach regionów układu, wzdłuż osi x. Następnie, przesuwanie modułów odbywa się w kolejnych kolumnach regionów, wzdłuż osi y. Nowe współrzędne modułów, położonych w poszczególnych wierszach lub kolumnach regionów układu są wyznaczane na podstawie wykorzystania sąsiednich regionów. Podczas pierwszego etapu rozmiar regionów układu jest ustalany tak, aby region mógł pomieścić cztery moduły o średniej powierzchni [28].

Program FastPlace stosuje analogię połączeń między modułami do układu sprężyn, podobnie jak program KraftWerk. Po przesunięciu modułów wzdłuż osi x lub osi y podczas pierwszych dwóch etapów, przed ponownym rozmieszczaniem modułów metodą programowania kwadratowego, do przesuniętych modułów powinny zostać przyłożone dodatkowe siły. W przeciwnym razie, podczas kolejnego rozmieszczania metodą programowania kwadratowego, moduły wrócą do swoich wcześniejszych położeń, które zajmowały przed przesuwaniem. W tym celu przesuwane moduły są łączone z dodatkowymi końcówkami, które są odpowiednio umieszczane na obrzeżach podłoża układu. Podczas kolejnego rozmieszczania metodą programowania kwadratowego, dodatkowe końcówki są traktowane tak samo jak końcówki całego układu. Po każdym przesuwaniu modułów, dotychczasowe dodatkowe końcówki są usuwane z obrzeża podłoża układu, a w ich miejsce wyznaczane są nowe [28].

Celem iteracyjnej korekcji rozmieszczenia modułów podczas drugiego etapu jest minimalizacja długości połączeń oraz zmniejszenie maksymalnego wykorzystania wśród wszystkich regionów układu. Podczas każdej iteracji korekcji rozmieszczenia modułów rozpatrywane są kolejno wszystkie moduły. Każdy moduł może być przeniesiony

tylko do sąsiedniego regionu, zachowując swoje położenie względem obrzeży regionu. Dla każdego modułu wyznaczana jest ważona suma redukcji długości połączeń oraz składnika zależnego od wykorzystania regionów po przeniesieniu modułu. Powyższa suma jest wyznaczana oddzielnie dla wszystkich czterech sąsiednich regionów. Moduł jest przenoszony do regionu, który zapewnia największą pozytywną zmianę. Moduł nie jest przenoszony w przypadku braku pozytywnej zmiany. Podczas drugiego etapu większy nacisk jest kładziony na minimalizację długości połączeń, która jest estymowana metodą połowy obwodu. Iteracje korekcji rozmieszczenia modułów są powtarzane do momentu, kiedy nie następuje dalsze zmniejszanie długości połączeń. Na początku drugiego etapu rozmieszczania rozmiary regionów są pięć razy większe od rozmiarów regionów, stosowanych w pierwszym etapie. Podczas drugiego etapu rozmiary regionów są stopniowo zmniejszane, aż do rozmiarów stosowanych w pierwszym etapie. Konieczność takiego rozwiązania wynika z możliwości przenoszenia modułów jedynie do sąsiednich regionów. Dzięki takiemu rozwiązaniu możliwa jest minimalizacja długości połączeń węzłów, które obejmują znaczną część układu — na początku drugiego etapu rozmieszczania [28].

Minimalizacja długości połączeń podczas trzeciego etapu rozmieszczania jest wykonywana podobnie jak korekcja rozmieszczenia modułów podczas drugiego etapu. Różnica polega jedynie na rozpatrywaniu modułów, które zostały przypisane do poszczególnych wierszy oraz kładzeniu większego nacisku na równomierne rozmieszczenie modułów [28].

Program FastPlace umożliwia otrzymanie rezultatów zbliżonych do programu GORDIAN / DOMINO pod względem całkowitej estymowanej długości połączeń, ale w znacznie krótszym czasie. FastPlace jest znacznie szybszym programem, również w porównaniu do innych programów rozmieszczania. Znacznie krótszy czas obliczeń jest możliwy dzięki zastosowaniu przesuwania modułów oraz iteracyjnej korekcji rozmieszczenia modułów, co znacznie poprawia zbieżność rozwiązań [28]. Program FastPlace jest powszechnie dostępny w postaci plików wykonywalnych [39].

4.4. MPL

Program mPl jest przeznaczony dla układów Standard Cell. Program wykorzystuje wielopoziomową optymalizację (ang. multilevel optimization). W programie mPl zastosowano podobny mechanizm, jak w przypadku wielopoziomowego podziału układu, przedstawionego w drugiej części przeglądu. Łączenie modułów odbywa się z użyciem algorytmu hMETIS i umożliwia zmniejszanie liczby modułów w liście połączeń dla kolejnych poziomów. Współrzędne powstających modułów są wyznaczane na podstawie średniej ważonej współrzędnych łączonych modułów. Łączenie modułów na kolejnych poziomach jest powtarzane do momentu, kiedy lista połączeń zawiera odpowiednio małą liczbę modułów, około 1000 modułów. Na najwyższym poziomie moduły są rozmieszczane z wykorzystaniem metody programowania nieliniowego z ograniczeniami. Na kolejnych niższych poziomach, wcześniej połączone moduły są

rozdzielane. Następnie, dla małych grup modułów, które nie sąsiadują ze sobą, wykonywana jest korekcja ich rozmieszczenia. Położenie modułów powstających w wyniku rozdzielenia połączonych modułów jest wyznaczane na podstawie położenia modułu, który jest rozdzielany. Uwzględniany jest również stopień połączenia rozdzielanych modułów z innymi modułami, łącznie z położeniem tych modułów. Po wykonaniu wielopoziomowej optymalizacji, moduły poddawane są rozmieszczaniu szczegółowemu. Wykonywane są losowe zamiany miejscami modułów położonych w ograniczonym sąsiedztwie. Program mPl pakuje moduły od lewej do prawej strony układu w wyniku sortowania współrzędnych modułów, co jest własnością programów typu variable-die. Rozmieszczanie szczegółowe może być również wykonane z użyciem programu DOMINO [13-15, 30-31].

Rozmieszczanie modułów z wykorzystaniem metody programowania nieliniowego z ograniczeniami jest wykonywane tylko na najwyższym poziomie, ponieważ lista połączeń zawiera wówczas najmniejszą liczbę modułów. Funkcja kosztu problemu rozmieszczania wykorzystuje kwadrat długości euklidesowej połączeń, tak jak w programie GORDIAN. Celem rozmieszczania jest minimalizacja całkowitej długości połączeń między modułami rozmieszczanymi w wierszach, pod warunkiem, że moduły nie mogą zachodzić na siebie. Mogą być również postawione dodatkowe ograniczenia, np. minimalizacja opóźnień w układzie. Powyższy problem rozmieszczania został sformułowany jako zadanie programowania nieliniowego z ograniczeniami, które jest rozwiązywane z użyciem metody interior-point (ang.). Zastosowanie metody programowania nieliniowego z ograniczeniami, które uwzględniają zachodzenie modułów na siebie jest bardzo istotne dla minimalizacji opóźnień w układzie, ponieważ zachodzenia modułów na siebie w początkowej fazie rozmieszczania, utrudnia tę minimalizację [30-31].

Podczas korekcji rozmieszczenia małych grup modułów, rozpatrywane są połączenia modułów w gwiazdę, podobnie jak w programie GORDIANL. Zastosowano również takie samo przekształcenie, które umożliwia minimalizację liniowej funkcji kosztu z użyciem metody programowania kwadratowego. Po wyznaczeniu nowego położenia modułów, moduły nie są od razu przenoszone w nowe położenie. Program mPl stosuje podział podłoża układu na siatkę prostokątnych regionów. Moduły są przenoszone w nowe położenie pojedynczo. Jeżeli przeniesienie modułu spowoduje powstanie zbyt dużego skupienia modułów w danym regionie, wybrany moduł z tego regionu jest przenoszony do innego regionu. Przenoszenie kolejnych modułów jest wykonywane do momentu, gdy przeniesienie ostatniego z modułów nie spowoduje zbyt dużego skupienia modułów w danym regionie układu (ang. ripple-move). Po przeniesieniu wszystkich modułów rozpatrywanej grupy w nowe położenie, nowe rozmieszczenie modułów jest akceptowane jedynie w przypadku, gdy powoduje zmniejszenie długości połączeń. Następnie, wykonywana jest korekcja rozmieszczenia dla innych grup modułów [31].

Program mPl wykonuje kilka prób rozmieszczania modułów z użyciem wielopoziomowej optymalizacji, w których początkowe położenie modułów określa rozwiązanie osiągnięte w poprzedniej próbie (ang. V-cycle). Rozmieszczenie modułów otrzy-

mane w poprzedniej próbie jest również wykorzystywane podczas łączenia modułów w następnej próbie. Program mPl dzieli większe moduły na mniejsze jednostkowe moduły, które mogą być rozmieszczane niezależnie. Jednostkowe moduły są połączone przez „sztuczne" węzły, o odpowiednio większej wadze. Powyższe rozwiązanie nie zapewnia jednak dobrych rezultatów w przypadku układów o znacznie różniących się rozmiarach modułów. Program mPl umożliwia otrzymanie rezultatów zbliżonych do programu GORDIANL / DOMINO pod względem całkowitej długości połączeń, ale w znacznie krótszym czasie. Zastosowanie liniowej i kwadratowej funkcji długości połączeń, które wzajemnie się uzupełniają, ma duży wpływ na jakość rozwiązań [3, 14-15, 22, 31]. Program mPl jest powszechnie dostępny w postaci plików wykonywalnych [39].

4.5. PROGRAM DLA TOPOGRAFII SWOBODNEJ

W niniejszym podrozdziale zostanie przedstawiony program rozmieszczania dla układów o topografii swobodnej. Moduły (makrobloki) w układach o topografii swobodnej mogą posiadać dowolne położenie i rozmiar. Program uwzględnia rozmiary modułów oraz położenie końcówek modułów, ponieważ poszczególne moduły mogą znacznie różnić się od siebie rozmiarami i zajmować znaczną część podłoża układu. Program stosuje analogię połączeń między modułami do układu sprężyn i stara się znaleźć stan równowagi układu, który odpowiada optymalnemu rozmieszczeniu modułów. Dla wszystkich węzłów są rozpatrywane połączenia jego końcówek w gwiazdę, również dla węzłów zawierających tylko dwie końcówki. Dodatkowy punkt gwiazdy każdego węzła podlega również rozmieszczaniu, tak samo jak moduły. Podłoże układu jest podzielone na siatkę kwadratowych regionów, których rozmiar nie powinien być większy niż połowa najmniejszego boku dowolnego modułu [29].

Rozmieszczanie modułów jest podzielone na trzy etapy. Podczas każdego etapu wykonywane jest iteracyjne przenoszenie modułów oraz dodatkowych punktów gwiazd. W każdej iteracji wszystkie moduły oraz dodatkowe punkty gwiazd są przenoszone zgodnie z wyznaczoną siłą, która w danej chwili oddziałuje na dany moduł lub dodatkowy punkt gwiazdy. Siła oddziałująca na dany moduł jest ważoną wypadkową sił pochodzących od dodatkowych punktów gwiazd poszczególnych węzłów układu oraz od regionów układu. Regiony układu o dużym wykorzystaniu są źródłem sił, oddziałujących na moduły położone w regionach o małym wykorzystaniu, podobnie jak w programie KraftWerk. Powyższe rozwiązanie umożliwia równomierne rozmieszczenie modułów na podłożu układu. Siła oddziałująca na dodatkowy punkt gwiazdy danego węzła jest wypadkową sił pochodzących od wszystkich końcówek modułów oraz końcówek całego układu, należących do tego węzła. W każdej iteracji moduły i dodatkowe punkty gwiazd są przenoszone oddzielnie, najpierw przenoszone są dodatkowe punkty gwiazd, a następnie moduły (po aktualizacji sił). Podczas przenoszenia dodatkowych punktów gwiazd są unieruchamiane moduły, i na odwrót. Podczas pierwszego etapu rozmieszczania nie są uwzględniane rozmiary modułów oraz siły

pochodzące od regionów układu. Celem pierwszego etapu jest otrzymanie dobrego wzajemnego położenia modułów. Podczas drugiego etapu są uwzględniane rozmiary modułów oraz siły pochodzące od regionów układu. Rozmieszczanie dodatkowych punktów gwiazd poszczególnych węzłów układu umożliwia estymację powierzchni, która jest niezbędna do poprowadzenia wszystkich połączeń (ang. routing estimation). Estymowane powierzchnie połączeń są uwzględniane podczas wyznaczania całkowitego wykorzystania regionów układu i mają wpływ na siłę oddziaływania regionu. W przeciwnym razie, zbyt duże upakowanie modułów w regionie mogłoby uniemożliwić wyznaczenie wszystkich połączeń układu. W kolejnych iteracjach drugiego etapu stopniowo wzrasta wpływ sił pochodzących od regionów układu na wypadkową siłę oddziałującą na moduły. Dzięki temu możliwe jest osiągniecie prawie równomiernego rozmieszczenia modułów. Podczas trzeciego etapu siły pochodzące od regionów układu odgrywają dominującą rolę, co umożliwia równomierne rozmieszczenie modułów. Ostatecznie, usuwane są wszelkie zachodzenia modułów na siebie [29].

Moduł (makroblok) może posiadać osiem różnych orientacji, które są rezultatem obrotu oraz odbicia względem jego osi symetrii. Podczas drugiego etapu rozmieszczania zmieniana jest orientacja ograniczonej liczby modułów. Zmiana orientacji modułu może być wykonana, jeżeli zapewni zmniejszenie siły oddziałującej na moduł lub zmniejszenie skupienia modułów w regionach układu [29].

Podczas wszystkich etapów wykonywane jest również iteracyjne rozmieszczanie końcówek całego układu na obrzeżach układu. W tym celu stosowana jest taka sama metoda, jak w przypadku rozmieszczania modułów. Rozmieszczanie końcówek całego układu odbywa się po każdym przeniesieniu modułów oraz dodatkowych punktów gwiazd, które są na ten czas unieruchamiane [29].

Przedstawiony program jest programem typu variable-die. Program umożliwił otrzymanie rozwiązań o znacznie lepszej jakości i w znacznie krótszym czasie, w porównaniu do komercyjnego programu rozmieszczania. Dotyczy to zarówno całkowitej długości połączeń, jak również całkowitej powierzchni układu po wyznaczeniu wszystkich połączeń [29].

4.6. INNE PROGRAMY

Program PROUD wykorzystuje analogię problemu rozmieszczania modułów do problemu wyznaczania napięć w węzłach liniowego obwodu rezystancyjnego prądu stałego. Postać funkcji kosztu połączeń problemu rozmieszczania jest taka sama jak postać całkowitej mocy wydzielanej w obwodzie rezystancyjnym. Końcówki całego układu są źródłami napięciowymi w obwodzie rezystancyjnym, wartości źródeł są ustalane na podstawie położenia końcówek na podłożu układu. Połączenia między modułami układu określają wartości rezystancji w rozpatrywanym obwodzie rezystancyjnym. Napięcia w poszczególnych węzłach obwodu rezystancyjnego wyznaczają położenie modułów na podłożu układu. Rozpływ prądów w obwodzie rezystancyjnym jest określony prawami Kirchhoffa i zapewnia minimalną wartość całkowitej mocy wydzielanej

w tym obwodzie. Wyznaczenie napięć w węzłach obwodu rezystancyjnego umożliwia więc optymalne rozmieszczenie modułów na podłożu układu. Eliminacja zachodzenia modułów na siebie odbywa się w wyniku hierarchicznego podziału podłoża układu na części. W każdej z części jest wykonywana dalsza optymalizacja rozmieszczenia modułów, która odbywa się jednak oddzielnie dla każdej z części. Program PROUD może być stosowany w układach o topografii wierszowej [9, 11, 32–33].

Program ATLAS jest przeznaczony dla układów o topografii swobodnej. Program uwzględnia rozmiary modułów oraz położenie końcówek modułów. Program uwzględnia również możliwość obrotu oraz odbicia względem osi symetrii modułu. ATLAS wykorzystuje kwadrat długości euklidesowej połączeń, końcówki we wszystkich węzłach układu są łączone w gwiazdę. Problem rozmieszczania modułów wymaga rozwiązania ciągu zadań programowania nieliniowego bez ograniczeń. Funkcja celu wszystkich zadań uwzględnia ograniczenia związane z rozmieszczaniem modułów m.in. zachodzenie modułów na siebie. Początkowo ograniczenia są uwzględniane w niewielkim stopniu. Udział ograniczeń w funkcji celu stopniowo wzrasta, aż do osiągnięcia poprawnego rozwiązania. Powyższe rozwiązanie umożliwiło poprawę jakości rozwiązań [9, 34].

Metody analityczne optymalizacji topografii układu ze względu na aspekt termiczny przedstawiono w pracy [2]. Uszkodzenia termiczne bardzo często są przyczyną uszkodzeń układów elektronicznych. Nieoptymalne rozmieszczenie modułów istotnych termicznie, powoduje powstanie zróżnicowania temperatur podłoża układu. Powstały gradient temperatury przyczynia się do występowania naprężeń mechanicznych w układzie, które to naprężenia zwiększają jego zawodność. Optymalne rozmieszczenie modułów istotnych termicznie zapewnia rozkład temperatury podłoża zbliżony do równomiernego [2].

Program FAR jest programem przeznaczonym dla układów Standard Cell. Funkcja kosztu problemu rozmieszczania modułów wykorzystuje kwadrat długości euklidesowej połączeń. Dla wszystkich węzłów układu stosowane są połączenia między wszystkimi końcówkami danego węzła, tak jak w grafie pełnym. Rozmieszczanie globalne modułów jest wykonywane metodą programowania kwadratowego bez ograniczeń. Równomierne rozmieszczenie modułów zapewniają dodatkowe „sztuczne" moduły, które powodują przenoszenie modułów z regionów układu o dużym skupieniu. FAR jest podstawą programu wielopoziomowej optymalizacji mFAR [14, 35].

Program BloBB jest przeznaczony dla układów o topografii swobodnej. Program uwzględnia rozmiary oraz możliwość obrotu modułów. BloBB wykorzystuje algorytm podziału i ograniczenia (ang. branch and bound), który jest zaliczany do metod analitycznych. Program umożliwia minimalizację powierzchni układu oraz spełnienie dodatkowych kryteriów rozmieszczania np. minimalizację całkowitej długości połączeń. BloBB zapewnia otrzymanie rozwiązań o lepszej jakości i w znacznie krótszym czasie, w porównaniu do programów opartych na algorytmie symulowanego wyżarzania [37]. Program BloBB jest powszechnie dostępny w postaci plików źródłowych i wykonywalnych [38].

Program APlace jest nowym programem, który wykorzystuje metodę programo-

wania nieliniowego. Nieliniowa funkcja celu uwzględnia całkowitą długość połączeń, a także równomierność rozmieszczenia modułów oraz skupienie połączeń na podłożu układu. Zadanie programowania nieliniowego bez ograniczeń jest rozwiązywane metodą gradientu sprzężonego, umożliwiając rozmieszczanie globalne modułów. W programie zastosowano algorytm wielopoziomowej optymalizacji. Program APlace umożliwia również rozmieszczanie modułów w układach o topografii mieszanej oraz minimalizację opóźnień w układzie. Program APlace zapewnia otrzymanie rozwiązań o wyraźnie lepszej jakości w porównaniu do innych programów, wymaga jednak dłuższego czasu na wykonanie obliczeń [36].

5. PODSUMOWANIE

Niniejsza praca jest trzecią częścią przeglądu, poświęconego metodom rozmieszczania w projektowaniu topografii układów VLSI. Zawiera szczegółowy opis algorytmu zamiany parami oraz obecnie stosowanych metod analitycznych. Przedstawiono charakterystykę wybranych programów rozmieszczania opartych na metodach analitycznych.

Metody analityczne należą do najstarszych metod rozmieszczania, jednak nadal są powszechnie stosowane. W ostatnich latach powstało wiele programów rozmieszczania dla różnych stylów topografii układu, których podstawą są metody analityczne. Powodem ciągłego zainteresowania badaczy metodami analitycznymi jest ich duża efektywność, która umożliwia rozmieszczanie modułów w układach o bardzo dużych rozmiarach. Większość obecnych programów rozmieszczania opartych na metodach analitycznych, wykorzystuje metodę programowania kwadratowego, która umożliwia efektywne rozmieszczanie globalne modułów. Rozmieszczanie szczegółowe zwykle jest wykonywane z użyciem innych metod. Dalsze badania dotyczą skutecznego usuwania zachodzenia modułów na siebie oraz zastosowania metody wielopoziomowej optymalizacji, podczas rozmieszczania globalnego.

6. BIBLIOGRAFIA

1. M.J.S. Smith: *Application-Specific Integrated Circuits*, Addison Wesley Longman, 1997.
2. A. Kos: *Modelowanie hybrydowych układów mocy i optymalizacja ich konstrukcji ze względu na rozkład temperatury*, Kraków, Wydawnictwa AGH, 1994.
3. B.T. Preas, M.J. Lorenzetti (red.): *Physical Design Automation of VLSI Systems*, Menlo Park, Benjamin-Cummings, 1988.
4. T. Ohtsuki (red.): *Layout Design and Verification*, Elsevier Science Publishers B.V. (North-Holland), 1986.
5. M.M. Vai: *VLSI Design*, CRC Press, 2001.
6. C. Sechen: *VLSI Placement and Global Routing Using Simulated Annealing*, Boston, Kluwer Academic Publishers, 1988.
7. M. Gajęcki, A. Kos: *A Problem of Optimization of Topography of VLSI Circuit*, Proc. of the XIXth National Conference on Circuit Theory and Electronic Networks, Kraków–Krynica (Poland), October 1996, pp. II/307–312.

8. W. Wolf: *Modern VLSI Design: a systems approach*, Englewood Cliffs, New Jersey, PTR Prentice Hall, 1994.
9. K. Shahookar, P. Mazumder: *VLSI Cell Placement Techniques*, ACM Computing Surveys, 1991, vol. 23, pp. 143–220.
10. C.J. Alpert, G.J. Nam, P.G. Villarrubia: *Effective Free Space Management for Cut-Based Placement via Analytical Constraint Generation*, IEEE Transactions on Computer-Aided Design, 2003, vol. 22, pp. 1343–1353.
11. T.C. Hu, E.S. Kuh (red.): *VLSI Circuit Layout: Theory and Design*, New York, IEEE Press, 1985.
12. M.A. Breuer (red.): *Automatyczne projektowanie maszyn cyfrowych*, Warszawa, PWN, 1976.
13. S.N. Adya, M.C. Yildiz, I.L. Markov, P.G. Villarrubia, P.N. Parakh, P.H. Madden: *Benchmarking for Large-Scale Placement and Beyond*, IEEE Transactions on Computer-Aided Design, 2004, vol. 23, pp. 472–486.
14. O. Liu, M. Marek-Sadowska: *A Study of Netlist Structure and Placement Efficiency*, IEEE Transactions on Computer-Aided Design, 2005, vol. 24, pp. 762–772.
15. C.C. Chang, J. Cong, M. Romesis, M. Xie: *Optimality and Scalability Study of Existing Placement Algorithms*, IEEE Transactions on Computer-Aided Design, 2004, vol. 23, pp. 537–549.
16. S. Sutanthavibul, E. Shragowitz, J.B. Rosen: *An Analytical Approach to Floorplan Design and Optimization*, IEEE Transactions on Computer-Aided Design, 1991, vol. 10, pp. 761–769.
17. S. Goto: *An Efficient Algorithm for the Two-Dimensional Placement Problem in Electrical Circuit Layout*, IEEE Transactions on Circuits and Systems, 1981, vol. 28, pp. 12–18.
18. W. Findeisen, J. Szymanowski, A. Wierzbicki: *Teoria i metody obliczeniowe optymalizacji*, Warszawa, PWN, 1977.
19. W. Grabowski: *Programowanie matematyczne*, Warszawa, Państwowe Wydawnictwo Ekonomiczne, 1982.
20. I. Dziubiński, T. Świątkowski (red.): *Poradnik matematyczny*, Warszawa, PWN, 1982.
21. J.M. Kleinhans, G. Sigl, F.M. Johannes, K.J. Antreich: *GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization*, IEEE Transactions on Computer-Aided Design, 1991, vol. 10, pp. 356–365.
22. G. Sigl, K. Doll, F.M. Johannes: *Analytical Placement: A Linear or a Quadratic Objective Function?*, Proc. of the Design Automation Conference, 1991, pp. 427–432, http://www.sigda.org/Archives/ProceedingArchives/Dac/.
23. K. Doll, F.M. Johannes, K.J. Antreich: *Iterative Placement Improvement by Network Flow Methods*, IEEE Transactions on Computer-Aided Design, 1994, vol. 13, pp. 1189–1200.
24. R. Sedgewick: *Algorithms in C, Part 5: Graph Algorithms*, Addison Wesley Professional, 2002.
25. R. Sedgewick: *Algorytmy w C++. Część 5. Grafy*, Warszawa, Wydawnictwo RM, 2003.
26. H. Eisenmann, F.M. Johannes: *Generic Global Placement and Floorplanning*, Proc. of the Design Automation Conference, 1998, pp. 269–274, http://www.sigda.org/Archives/ProceedingArchives/Dac/.
27. B. Obermeier, H. Ranke, F.M. Johannes: *Kraftwerk — A Versatile Placement Approach*, Proc. of the International Symposium on Physical Design, 2005, pp. 242–244, http://www.sigda.org/Archives/ProceedingArchives/Ispd/.
28. N. Viswanathan, C.C.-N. Chu: *FastPlace: Efficient Analytical Placement Using Cell Shifting, Iterative Local Refinement and a Hybrid Net Model*, IEEE Transactions on Computer-Aided Design, 2005, vol. 24, pp. 722–733.
29. F. Mo, A. Tabbara, R.K. Brayton: *A Force-Directed Macro-Cell Placer*, Proc. of the International Conference on Computer-Aided Design, 2000, pp. 177– 180, http://www.sigda.org/Archives/ProceedingArchives/Iccad/.
30. T.F. Chan, J. Cong, T. Kong, J.R. Shinnerl: *Multilevel Optimization for Large-Scale Circuit Placement*, Proc. of the International Conference on Computer-Aided Design, 2000, pp. 171–176, http://cadlab.cs.ucla.edu/~cong/papers/iccad00_placement.pdf.

31. T.F. Chan, J. Cong, T. Kong, J.R. Shinnerl, K. Sze: *An Enhanced Multilevel Algorithm for Circuit Placement*, Proc. of the International Conference on Computer-Aided Design, 2003, pp. 299–306, http://ballade.cs.ucla.edu/~cong/papers/mpl2.pdf.
32. C.-K. Cheng, E.S. Kuh: *Module Placement Based on Resistive Network Optimization*, IEEE Transactions on Computer-Aided Design, 1984, vol. 3, pp. 218-225.
33. R.-S. Tsay, E.S. Kuh, C.-P. Hsu: *PROUD: A Fast Sea-Of-Gates Placement Algorithm*, Proc. of the Design Automation Conference, 1988, pp. 318-323.
34. L. Sha, T. Blank: *ATLAS — A Technique for Layout using Analytic Shapes*, Proc. of the International Conference on Computer-Aided Design, 1987, pp. 84-87, http://www.sigda.org/Archives/ProceedingArchives/Iccad/Last20/Papers/1987/.
35. B. Hu, M. Marek-Sadowska: *Multilevel Fixed-Point-Addition-Based VLSI Placement*, IEEE Transactions on Computer-Aided Design, 2005, vol. 24, pp. 1188-1203.
36. A.B. Kahng, Q. Wang: *Implementation and Extensibility of an Analytic Placer*, IEEE Transactions on Computer-Aided Design, 2005, vol. 24, pp. 734-747.
37. H.H. Chan, I.L. Markov: *Practical Slicing and Non-slicing Block-Packing without Simulated Annealing*, Proc. of the Great Lakes Symposium on VLSI, 2004, pp. 282-287, http://vlsicad.eecs.umich.edu/BK/BloBB/PAPERS/p037-chan.pdf.
38. VLSI CAD Bookshelf Slots and Entries, A.B. Kahng, I.L. Markov, http://vlsicad.eecs.umich.edu/BK/Slots/slots/BlockPacking.html.
39. VLSI CAD Bookshelf Slots and Entries, A.B. Kahng, I.L. Markov, http://vlsicad.eecs.umich.edu/BK/Slots/slots/WirelengthdrivenStandardCellPlacement.html.
40. J.-G. Kim, Y.-D. Kim: *A Linear Programming-Based Algorithm for Floorplanning in VLSI Design*, IEEE Transactions on Computer-Aided Design, 2003, vol. 22, pp. 584-592.

Z. NAGÓRNY, A. KOS

THE DESIGN OF THE VLSI CIRCUIT LAYOUT PART 3. ANALYTICAL METHODS

Summary

The design process of the VLSI circuits requires the use of computer aided design tools. This paper is the third part of the survey of the cell placement techniques for digital VLSI circuits. In this part of the survey, the pairwise interchange algorithm and some analytical methods are presented. The force-directed placement algorithm and some modifications of the pairwise interchange algorithm, which are used in analytical algorithms are described. Then, the nonlinear programming, quadratic programming and linear programming techniques are presented. An application of these techniques to the cell placement problem is described. Nowadays the tools used for the cell placement, which utilize the presented algorithms are characterized: GORDIAN, DOMINO, KraftWerk, FastPlace, mPL, PROUD, ATLAS, FAR, mFAR, BloBB, APlace. A force-directed placer for a building block design style is described. The principles of the multilevel optimization for the cell placement problem are presented. Applications of the flow network and branch and bound algorithm to the cell placement are characterized. Some conclusions concerning described techniques and tools are presented.

Keywords: VLSI, physical design process, integrated circuit, Standard Cell, Sea-of-Gates, cell placement, pairwise interchange algorithm, force-directed placement algorithm, force-directed relaxation, force-directed interchange, force-directed pairwise relaxation, generalized force-directed relaxation, analytical methods, nonlinear programming, quadratic programming, linear programming, flow network, multilevel optimization, GORDIAN, DOMINO, KraftWerk, FastPlace, mPL, PROUD, ATLAS, FAR, mFAR, BloBB, APlace

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 4, ss. 697–727

Projektowanie topografii systemów VLSI Cz. 4. Symulowane wyżarzanie, sieci neuronowe

ZBIGNIEW NAGÓRNY, ANDRZEJ KOS

Zbigniew Nagórny,
Studium Generale Sandomiriense,
Wyższa Szkoła Humanistyczno-Przyrodnicza w Sandomierzu,
ul. Krakowska 26, 27-600 Sandomierz,
e-mail: nagorny@interia.pl,
Andrzej Kos,
Akademia Górniczo-Hutnicza w Krakowie,
Katedra Elektroniki,
al. Mickiewicza 30, 30-059 Kraków,
e-mail: kos@agh.edu.pl

Otrzymano 2005.12.05
Autoryzowano 2006.02.06

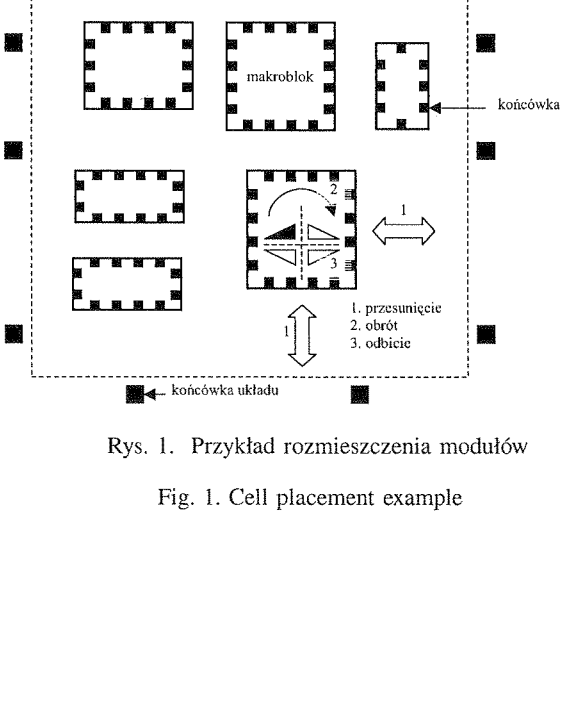
Niniejsza praca jest czwartą, ostatnią częścią przeglądu metod rozmieszczania modułów, stosowanych podczas projektowania topografii układów VLSI. Modułem jest fragment systemu wyodrębniony ze względu na pełnioną funkcję. Praca jest poświęcona algorytmowi symulowanego wyżarzania oraz sieciom neuronowych. Przedstawiono dokładny opis algorytmu symulowanego wyżarzania oraz sposób zastosowania algorytmu do rozmieszczania modułów. Programy wykorzystujące algorytm symulowanego wyżarzania zostały szczegółowo opisane. W tym celu scharakteryzowano następujące programy rozmieszczania: TimberWolf, MGP, MPG-MS, VPR. Następnie, opisano sposób zastosowania sieci samoorganizującej się oraz sieci Hopfielda w optymalizacji topografii układów VLSI. Przedstawiono rezultaty rozmieszczania modułów otrzymane z użyciem sieci Hopfielda. Następnie, scharakteryzowano inne metody stosowane podczas rozmieszczania modułów: algorytmy genetyczne, strategie ewolucyjne, schemat rozmieszczanie-planowanie topografii-rozmieszczanie, programy dla układów 3D VLSI oraz sprzętowe metody rozwiązania problemu rozmieszczania modułów. Porównano metody rozmieszczania modułów przedstawione w przeglądzie.

Słowa kluczowe: VLSI, projektowanie topografii systemu, system scalony, matryca bramkowa GA, układ Standard Cell, układ Sea-of-Gates, układ FPGA, planowanie układu, rozmieszczanie modułów, trasowanie połączeń, algorytm zamiany parami, algorytm symulowanego wyżarzania, harmonogram chłodzenia,

wielopoziomowa optymalizacja, sieci neuronowe, sieć samoorganizująca się, algorytm Kohonena, sieć Hopfielda, algorytm genetyczny, strategia ewolucyjna, system rekonfigurowalny, TimberWolf, MGP, MPG-MS, VPR, Parquet, Gravity

1. WPROWADZENIE

Niniejsza praca stanowi czwartą część przeglądu metod rozmieszczania modułów, stosowanych podczas projektowania topografii układów VLSI. Rozmieszczanie modułów w układzie VLSI jest bardzo ważnym etapem projektowania jego topografii. Danymi wejściowymi dla metod rozmieszczania są kształty, rozmiary i położenie końcówek poszczególnych modułów układu, położenie końcówek całego układu oraz lista połączeń między modułami i końcówkami całego układu. Rozwiązanie problemu rozmieszczania polega na znalezieniu współrzędnych prostokątnych, określających położenie modułów w układzie tak, aby zapewnić spełnienie celów stawianych procesowi rozmieszczania. W układach o topografii wierszowej często zakłada się, że wszystkie końcówki modułów są położone w środku geometrycznym modułów, ponieważ rozmiary modułów są niewielkie w porównaniu do rozmiarów podłoża układu. Położenie końcówek modułów (makrobloków) musi być uwzględnione w układach o topografii swobodnej, ponieważ poszczególne moduły mogą znacznie różnić się od siebie rozmiarami i zajmować znaczną część podłoża układu. Uwzględniana jest również orientacja modułu (makrobloku). Każdy moduł może posiadać osiem różnych orientacji, które są rezultatem obrotu oraz odbicia względem jego osi symetrii. Rysunek 1 przedstawia przykład rozmieszczenia modułów.



Rys. 1. Przykład rozmieszczenia modułów

Fig. 1. Cell placement example

W pracy przedstawiono zastosowanie algorytmu symulowanego wyżarzania oraz sieci neuronowych do rozmieszczania modułów. Opisano programy rozmieszczania, wykorzystujące przedstawione rozwiązania. Algorytm symulowanego wyżarzania jest zaliczany do algorytmów iteracyjnych, jest uważany za modyfikację algorytmu zamiany parami. Algorytm symulowanego wyżarzania jest stosowany w projektowaniu topografii układów VLSI od wielu lat [6–7, 9, 11–12, 17–21]. Jest zaliczany do grupy rozwiązań, które są najczęściej stosowane. Umożliwia otrzymanie rozwiązań o bardzo dobrej jakości. Istnieje wiele programów rozmieszczania, które wykorzystują ten algorytm [6–7, 9-10, 17–27, 29-36, 56–57]. W rozdziale 2 przedstawiono algorytm symulowanego wyżarzania oraz sposób zastosowania algorytmu w problemie rozmieszczania modułów. Programy rozmieszczania, które wykorzystują algorytm symulowanego wyżarzania są opisane w rozdziale 3. W rozdziale 4 przedstawiono zastosowanie sieci neuronowych w problemie rozmieszczania modułów. Inne rozwiązania stosowane podczas rozmieszczania modułów scharakteryzowano w rozdziale 5. Komercyjne programy rozmieszczania opisano w rozdziale 6. Podsumowanie całego przeglądu metod rozmieszczania zamieszczono w rozdziale 7.

2. SYMULOWANE WYŻARZANIE

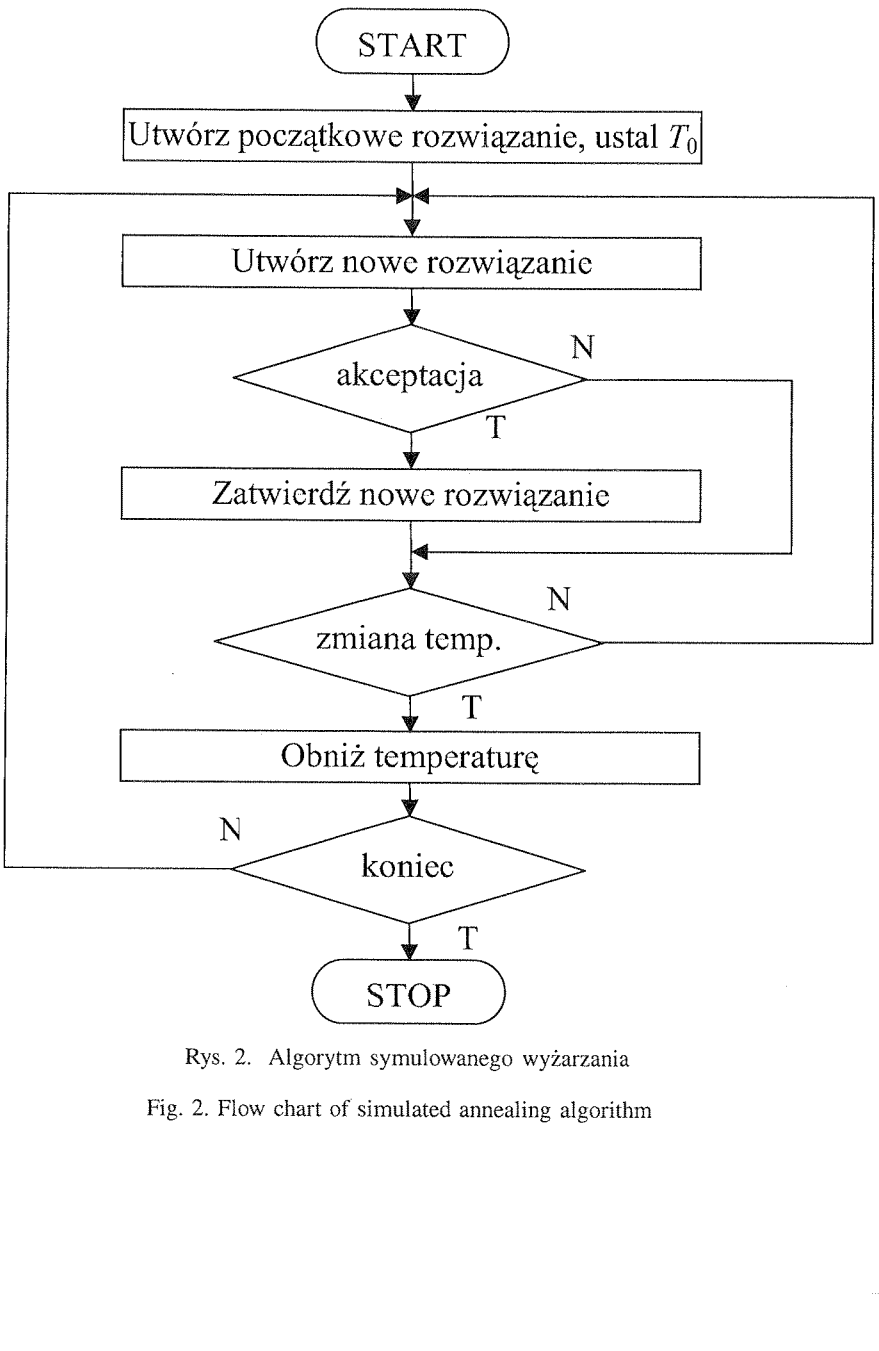
Algorytm symulowanego wyżarzania (ang. simulated annealing) wykorzystuje analogię do procesu krzepnięcia ciekłego ciała. Powolne schładzanie ciekłego, rozżarzonego ciała umożliwia powstanie regularnej struktury atomów stałego ciała. Zbyt szybkie schładzanie powoduje natomiast powstanie wielu nieregularności w strukturze atomów. Nieregularność struktury atomów jest powodem wyższego stanu energetycznego ciała, w porównaniu do stanu energetycznego ciała schładzanego powoli. Zgodnie z mechaniką statystyczną stan energetyczny ciała może wzrastać w temperaturze powyżej zera absolutnego. W trakcie powolnego schładzania ciała, możliwość chwilowego wzrostu stanu energetycznego umożliwia opuszczenie minimum lokalnego funkcji energii ciała. W efekcie otrzymujemy regularną strukturę, której stan energetyczny jest równy lub bliski wartości minimum globalnego. Powyższe zjawisko stało się inspiracją do zastosowania algorytmu symulowanego wyżarzania w problemach optymalizacyjnych. Funkcja kosztu problemu optymalizacyjnego odpowiada funkcji energii ciała. Zastosowanie analogii do procesu krzepnięcia ciała umożliwia zatem znalezienie minimum globalnego funkcji kosztu problemu optymalizacyjnego [3–9, 11–13, 15–16].

Rysunek 2 przedstawia sposób postępowania w algorytmie symulowanego wyżarzania. Na początku tworzone jest początkowe rozwiązanie, najczęściej w sposób losowy. W tym celu musi być ustalony sposób reprezentacji rozwiązania problemu optymalizacyjnego. Algorytm symulowanego wyżarzania jest algorytmem iteracyjnym. Podstawową czynnością w algorytmie jest tworzenie nowego rozwiązania w wyniku zmiany obecnego rozwiązania. Nowe rozwiązanie jest akceptowane na podstawie wartości zmiany funkcji kosztu problemu optymalizacyjnego. Zmiana rozwiązania, która umożliwia zmniejszenie wartości funkcji kosztu jest zawsze akceptowana. Nowe roz-

wiązania, które zwiększają wartość funkcji kosztu są akceptowane, jeżeli spełniony jest następujący warunek [12]

$$e^{-\Delta E/T} > \zeta \tag{1}$$

gdzie: ΔE — zmiana wartości funkcji kosztu, równa różnicy nowej i poprzedniej wartości funkcji kosztu, ζ — liczba losowa z przedziału $< 0, 1 >$, wylosowana zgodnie z rozkładem jednostajnym, T — „temperatura" układu (parametr algorytmu), zgodnie z analogią do procesu krzepnięcia z wyżarzaniem.



Rys. 2. Algorytm symulowanego wyżarzania

Fig. 2. Flow chart of simulated annealing algorithm

Temperatura początkowa T_0 (ang. initial temperature) w algorytmie symulowanego wyżarzania jest ustalana na podstawie symulacji wstępnych. Temperatura początkowa jest ustalana tak, aby procentowa wartość akceptacji nowych rozwiązań w temperaturze początkowej przekraczała określony próg, który może mieć różną wartość, w zależności od rodzaju problemu optymalizacyjnego. Najczęściej po wykonaniu określonej liczby L generacji nowego rozwiązania, temperatura jest obniżana zgodnie z następującym harmonogramem chłodzenia (ang. cooling schedule, annealing schedule) [12]

$$T_{i+1} = \alpha T_i \tag{2}$$

gdzie: T_{i+1} temperatura dla kroku $i + 1$ zmiany temperatury, T_i — temperatura dla kroku i, α — współczynnik redukcji temperatury.

Wartość współczynnika α wynosi zwykle $0,8 \div 0,95$. Temperatura może być obniżana również w inny sposób. Jeżeli podczas kilku kolejnych kroków obniżania temperatury nie następuje dalsza redukcja funkcji kosztu problemu optymalizacyjnego lub temperatura zostanie obniżona poniżej pewnej wartości (ang. stopping criterion), to wykonywanie algorytmu jest przerywane, a najlepsze otrzymane rozwiązanie jest rozwiązaniem problemu optymalizacyjnego [1, 3-7, 9, 11-14, 17, 19, 23, 26].

Algorytm symulowanego wyżarzania jest uważany za udoskonaloną postać algorytmu zamiany parami. Algorytm zamiany parami akceptuje jedynie zmiany rozwiązania prowadzące do zmniejszenia funkcji kosztu, co jest powodem „utykania" algorytmu w minimach lokalnych funkcji kosztu. Akceptacja zmian rozwiązań prowadzących do wzrostu wartości funkcji kosztu w algorytmie symulowanego wyżarzania, umożliwia opuszczanie minimów lokalnych funkcji kosztu. Jeżeli temperatura jest wysoka to prawdopodobieństwo akceptacji gorszego rozwiązania jest bardzo duże. Obniżanie temperatury powoduje stopniowe zmniejszanie prawdopodobieństwa akceptacji gorszych rozwiązań. Algorytm symulowanego wyżarzania teoretycznie zapewnia osiągnięcie minimum globalnego funkcji kosztu, w przypadku utworzenia nieskończenie wiele nowych rozwiązań dla każdej temperatury. W praktyce stosuje się odpowiednio wolne sposoby obniżania temperatury. Algorytm symulowanego wyżarzania umożliwia otrzymanie rozwiązań o bardzo dobrej jakości. Wadą algorytmu jest jednak długi czas obliczeń i konieczność bardzo dokładnego dostrojenia parametrów algorytmu [1, 3–7, 9, 11–12, 17, 27].

Podczas rozmieszczania modułów w układach VLSI z użyciem algorytmu symulowanego wyżarzania wykorzystywana jest estymacja wartości funkcji kosztu, ponieważ wyznaczenie połączeń dla wszystkich utworzonych rozwiązań zajęłoby zbyt wiele czasu. Nowe rozwiązania są tworzone na podstawie bieżącego rozmieszczenia modułów. Nowe rozwiązania najczęściej są tworzone w wyniku zamiany miejscami losowo wybranej pary modułów, przenoszenia losowo wybranego modułu w losowo wybrane miejsce, obrotu losowo wybranego modułu lub odbicia losowo wybranego modułu względem jednej z jego osi symetrii. Liczba nowych rozwiązań L, tworzonych podczas każdego kroku obniżania temperatury jest zwykle wielokrotnością liczby rozmieszczanych modułów. W następnym rozdziale zostaną opisane programy rozmieszczania,

które wykorzystują algorytm symulowanego wyżarzania [1, 3, 5–9, 11–12, 17–21, 23, 25–27].

3. PROGRAMY ROZMIESZCZANIA WYKORZYSTUJĄCE ALGORYTM SYMULOWANEGO WYŻARZANIA

Algorytm symulowanego wyżarzania należy do najczęściej stosowanych rozwiązań podczas rozmieszczania modułów w układach VLSI. Programy wykorzystujące symulowane wyżarzanie umożliwiają otrzymanie rozwiązań o bardzo dobrej jakości. W kolejnych podrozdziałach zostaną scharakteryzowane następujące programy rozmieszczania: TimberWolf, MGP, MPG-MS i VPR. Zostaną również przedstawione specjalne sposoby kodowania rozwiązań, które bardzo często są stosowane w algorytmie symulowanego wyżarzania podczas rozmieszczania modułów w układach o topografii swobodnej lub podczas etapu planowania układu. Przedstawione programy rozmieszczania mogą być stosowane w układach o różnych stylach topografii [6–7, 9–10, 17–27, 29–36, 56–57].

3.1. TIMBERWOLF

TimberWolf jest programem, który umożliwia zarówno rozmieszczanie modułów, jak również trasowanie globalne połączeń. Ze względu na temat pracy, szczególna uwaga zostanie zwrócona na problem rozmieszczania modułów. Program TimberWolf był wielokrotnie modyfikowany, może być stosowany w układach o różnych stylach topografii. Wersje programu dla różnych stylów topografii układu zostaną przedstawione w kolejnych podrozdziałach. Obecnie program jest produktem komercyjnym, posiadającym nazwę iTools [6–7, 9, 17–22].

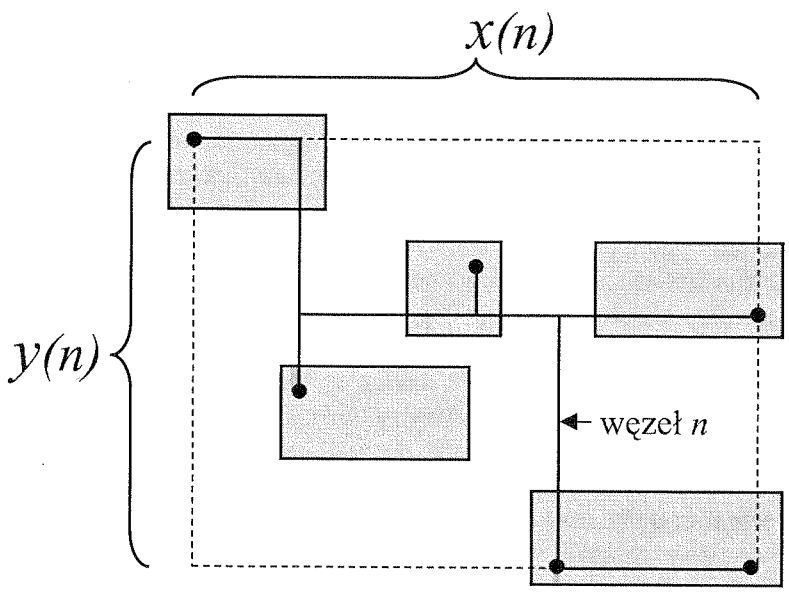
3.1.1. TimberWolf — topografia wierszowa

TimberWolfSC jest programem przeznaczonym dla układów Standard Cell. TimberWolfSC wersja 3.2 wykonuje rozmieszczanie modułów oraz trasowanie globalne podczas trzech etapów. Pierwsze dwa etapy wykorzystują algorytm symulowanego wyżarzania. Podczas pierwszego etapu minimalizowana jest całkowita estymowana długość połączeń. Podczas drugiego etapu dodawane są specjalne moduły dla połączeń, które muszą przeciąć wiersz modułów, odbywa się dalsza minimalizacja długości połączeń oraz jest wykonywane trasowanie globalne połączeń, które określa skupienie połączeń w poszczególnych kanałach układu. Drugi etap rozpoczyna się po obniżeniu temperatury poniżej pewnej wartości. Podczas drugiego etapu moduły nie mogą być przenoszone do innych wierszy, może odbywać się jedynie zamiana miejscami losowo wybranych sąsiednich modułów w danym wierszu układu lub odbicie względem poziomej osi symetrii wybranego modułu. Podczas trzeciego etapu są wykonywane zamiany

miejscami sąsiednich modułów w wierszach układu, które mają na celu zmniejszenie wysokości kanałów niezbędnych do wykonania wszystkich połączeń. Akceptowane są jedynie zmiany prowadzące do poprawy jakości rozwiązania. Zachodzenie modułów na siebie jest niedopuszczalne podczas drugiego i trzeciego etapu. Pierwszy etap zostanie dokładniej opisany. Program TimberWolf może być również stosowany dla matryc bramkowych GA [6-7, 9, 17, 19].

Nowe rozwiązania w algorytmie symulowanego wyżarzania (pierwszy etap) są tworzone w wyniku przenoszenia losowo wybranego modułu w losowo wybrane miejsce, zamiany miejscami losowo wybranej pary modułów oraz odbicia względem poziomej osi symetrii wybranego modułu. Zmiana położenia modułów może odbywać się w ograniczonym zakresie (ang. range limiter). W temperaturze początkowej zmiany położenia modułów mogą obejmować całe podłoże układu. Zakres dopuszczalnych zmian położenia modułów zmniejsza się logarytmicznie wraz z obniżaniem temperatury [6–7, 9, 17, 19].

Funkcja kosztu algorytmu symulowanego wyżarzania (pierwszy etap) składa się z trzech składników. Pierwszy składnik jest równy ważonej sumie estymowanych długości połączeń dla wszystkich węzłów układu. Estymacja długości połączeń odbywa się metodą połowy obwodu. Estymacja długości połączeń uwzględnia rzeczywiste położenie końcówek modułów, co przedstawia rysunek 3.



Rys. 3. Estymacja długości połączeń metodą połowy obwodu

Fig. 3. Half-perimeter wire length estimation

Podczas estymacji długości połączeń poszczególnych węzłów układu, szerokość oraz wysokość prostokąta obejmującego wszystkie końcówki należące do danego węzła,

posiadają własne wagi. Wagi te umożliwiają preferencję wybranego kierunku połączeń. Wszystkie węzły układu również posiadają własne wagi, co umożliwia preferencję wybranych węzłów. Zachodzenie modułów na siebie jest dozwolone, ale jest uwzględniane w wartości funkcji kosztu. Drugi składnik funkcji kosztu jest równy sumie kwadratów długości odcinków, na których długości zachodzą na siebie poszczególne moduły w kierunku poziomym. Sprawdzane są wszystkie możliwe pary modułów. Trzeci składnik funkcji kosztu jest sumą wartości bezwzględnych różnic między bieżącą i pożądaną długością wszystkich wierszy układu. Składnik ten umożliwia takie rozmieszczenie modułów, dla którego długość wszystkich wierszy jest zbliżona, co zapewnia oszczędność powierzchni podłoża układu. Udział drugiego oraz trzeciego składnika w funkcji kosztu określają wagi tych składników [6–7, 9, 17–19, 21].

Początkowe rozmieszczenie modułów na podłożu układu jest określane w sposób losowy. Temperatura początkowa w algorytmie symulowanego wyżarzania (pierwszy etap) wynosi 4 000 000. Temperatura jest obniżana zgodnie ze wzorem 2. Współczynnik redukcji temperatury α zmienia się w zakresie $0,8 \div 0,95$, w zależności od bieżącej wartości temperatury. Dla każdej temperatury jest tworzona określona liczba nowych rozwiązań, która jest równa wielokrotności liczby modułów. W przypadku układu składającego się z 200 modułów, zalecana liczba tworzonych rozwiązań jest 100 razy większa od liczby modułów. Drugi składnik funkcji kosztu zapewnia zupełny brak lub nieznaczne zachodzenie modułów na siebie na końcu pierwszego etapu. Ewentualne zachodzenie modułów na siebie jest eliminowane przed rozpoczęciem drugiego etapu, w wyniku odpowiedniego przesuwania modułów. TimberWolfSC wersja 3.2 zapewniał otrzymanie znacznie lepszych rozwiązań pod względem całkowitej długości połączeń oraz całkowitej powierzchni układu po wyznaczeniu wszystkich połączeń, w porównaniu do współczesnych programów opartych na algorytmie min-cut [6–7, 9, 17, 19].

W programie TimberWolfSC wersja 4.2 wprowadzono szereg zmian w algorytmie symulowanego wyżarzania, które znacznie skróciły czas obliczeń. Każdy wiersz modułów został podzielony na siatkę prostokątnych regionów (ang. bin), które nie zachodzą na siebie. Regiony umożliwiają szybsze obliczenie drugiego składnika funkcji kosztu, wyrażającego zachodzenie modułów na siebie. Dla każdego regionu jest rozpatrywana bezwzględna różnica między szerokością regionu a sumą szerokości modułów lub ich części, znajdujących się w danym regionie. Suma wartości bezwzględnych obliczona dla wszystkich regionów, wyraża stopień zachodzenia modułów na siebie i jest uwzględniana w funkcji kosztu z odpowiednią wagą. Powyższa zmiana umożliwia znacznie szybsze obliczenie drugiego składnika funkcji kosztu, ponieważ nie są rozpatrywane wszystkie możliwe pary modułów. W algorytmie symulowanego wyżarzania wagi drugiego i trzeciego składnika funkcji kosztu podlegają zmianom w trakcie obliczeń. Odrzucane są również nowe rozwiązania, dla których suma zmiany drugiego i trzeciego składnika funkcji kosztu jest odpowiednio duża, ze względu na bardzo małe prawdopodobieństwo akceptacji takiego rozwiązania. Powyższe rozwiązanie skraca czas obliczeń, ponieważ dla tych rozwiązań nie jest wyznaczany kosztowny

obliczeniowo pierwszy składnik funkcji kosztu. Wyznaczanie nowych rozwiązań polega na przenoszeniu losowo wybranego modułu w losowo wybrane miejsce. Jeżeli wybrane miejsce jest wolne procedura tworzenia nowego rozwiązania kończy się. Jeżeli wybrane miejsce jest zajęte przez inny moduł, to wówczas moduły są zamieniane miejscami między sobą. Temperatura początkowa algorytmu symulowanego wyżarzania wynosi 500, współczynnik redukcji temperatury $\alpha = 0,98$. Wprowadzono również parametr, który sprawia, że udział zaakceptowanych rozwiązań, w całej liczbie nowych rozwiązań tworzonych w danej temperaturze, zmienia się od 50% do 0% wraz ze zmniejszaniem temperatury. TimberWolfSC wersja 4.2 umożliwił znaczną redukcję czasu obliczeń, przy nieznacznie lepszej jakości rozwiązań, w porównaniu do programu TimberWolfSC wersja 3.2 [9, 18].

W programie TimberWolfSC wersja 7.0 wprowadzono dalsze, bardzo istotne zmiany. Zachodzenie modułów na siebie jest niedopuszczalne. Dzięki temu możliwa jest bardzo dokładna estymacja długości połączeń oraz opóźnień w układzie, od samego początku rozmieszczania modułów. Ponadto długość wierszy modułów rzeczywiście zawsze mieści się w ustalonym przedziale. Nowe rozwiązania w programie TimberWolfSC wersja 7.0 są tworzone tak, aby długość wierszy modułów mieściła się w ustalonym przedziale. W związku z tym, funkcja kosztu składa się tylko ze składnika wyrażającego całkowitą estymowaną długość połączeń. Nowe rozwiązania są tworzone w wyniku przenoszenia losowo wybranego modułu w losowo wybrane miejsce lub zamiany miejscami losowo wybranej pary modułów pod warunkiem, że długość wierszy modułów pozostanie w ustalonym przedziale. Zmiana położenia modułów może odbywać się w ograniczonym zakresie, podobnie jak w poprzednich wersjach programu. Zmiana funkcji kosztu odpowiadająca nowemu rozwiązaniu jest sumą dwóch składników. Pierwszy składnik jest równy sumie zmian długości połączeń wszystkich węzłów, do których należał przenoszony moduł lub para modułów podczas tworzenia nowego rozwiązania. Pierwszy składnik jest obliczany z wykorzystaniem metody połowy obwodu na podstawie dokładnie wyznaczonego nowego położenia przenoszonego modułu lub pary modułów. Drugi składnik jest specjalną aproksymacją sumy zmian długości połączeń wszystkich węzłów, do których należą wszystkie moduły, które powinny być przesunięte w przypadku akceptacji nowego rozwiązania, w celu usunięcia powstałego zachodzenia modułów na siebie. Podczas wyznaczania drugiego składnika zmiany funkcji kosztu, nie jest wyznaczane dokładne położenie powyższych modułów, a moduły nie są przesuwane. Takie rozwiązanie umożliwia wyraźne zmniejszenie czasu potrzebnego na wykonanie obliczeń. Nowe rozwiązanie jest akceptowane na podstawie zmiany funkcji kosztu oraz temperatury. Jedynie w przypadku akceptacji nowego rozwiązania, w wierszach, w których nowe rozwiązanie spowodowało zmianę położenia modułów, odbywa się odpowiednie przesuwanie modułów. Przesuwanie modułów ma na celu usunięcie ewentualnego zachodzenia modułów na siebie. Program TimberWolfSC wersja 7.0 wykorzystuje wielopoziomową optymalizację. W programie zastosowano podobny mechanizm, jak w przypadku wielopoziomowego podziału układu, przedstawionego w drugiej części przeglądu. Łączenie modułów umożliwia zmniej-

szanie liczby modułów w liście połączeń dla dwóch wyższych poziomów. Na wszystkich poziomach moduły rozmieszczane są z wykorzystaniem algorytmu symulowanego wyżarzania, uruchamianego w różnych temperaturach początkowych. Na najwyższym poziomie temperatura początkowa jest wysoka. Na kolejnym, niższym poziomie temperatura początkowa jest średnia. Na najniższym poziomie (poziom oryginalnej listy połączeń) temperatura początkowa jest niska. Na kolejnych niższych poziomach, wcześniej połączone moduły są rozdzielane. Położenie modułów powstających w wyniku rozdzielenia połączonych modułów jest wyznaczane losowo w obszarze modułu wyższego poziomu. Łączenie modułów również odbywa się z wykorzystaniem algorytmu symulowanego wyżarzania. Łączenie modułów stosowane w programie TimberWolfSC wersja 7.0 umożliwia tworzenie modułów o zbliżonej wielkości na kolejnych wyższych poziomach listy połączeń, co wyraźnie poprawia jakość rozmieszczania modułów. W algorytmie symulowanego wyżarzania dla problemu rozmieszczania modułów, wprowadzono parametr, który sprawia, że udział akceptowanych rozwiązań w całej liczbie tworzonych nowych rozwiązań zmienia się w ściśle określony sposób wraz ze zmniejszaniem temperatury. Udział akceptowanych rozwiązań jest wyznaczany wyłącznie na podstawie rozwiązań, które powodują zmianę wiersza przez moduł lub parę modułów. W przeciwnym razie, zmiany rozwiązań powodujące zmianę wiersza przez moduł lub parę modułów, posiadałyby bardzo niski stopień akceptacji, a jakość rozwiązań byłaby gorsza. TimberWolfSC wersja 7.0 zapewnia otrzymanie znacznie lepszych rozwiązań pod względem całkowitej estymowanej długości połączeń oraz całkowitej powierzchni układu po wyznaczeniu wszystkich połączeń, w porównaniu do poprzednich wersji tego programu. Jakość rozwiązań jest również wyraźnie lepsza w porównaniu do programu GORDIANL / DOMINO. Program umożliwia również dokładną minimalizację opóźnień w układzie [21].

3.1.2. TimberWolf — topografia swobodna

TimberWolfMC jest programem przeznaczonym dla układów o topografii swobodnej. TimberWolfMC wersja 1.0 wykonuje rozmieszczanie modułów oraz trasowanie globalne podczas dwóch etapów. Obydwa etapy wykorzystują algorytm symulowanego wyżarzania [6–7, 17].

Podczas pierwszego etapu jest minimalizowana całkowita estymowana długość połączeń. Nowe rozwiązania podczas pierwszego etapu są tworzone w wyniku przenoszenia losowo wybranego modułu w losowo wybrane miejsce, zamiany miejscami losowo wybranej pary modułów lub zmiany orientacji losowo wybranego modułu. Każdy moduł może posiadać osiem różnych orientacji, które są rezultatem obrotu oraz odbicia względem jego osi symetrii. Zmiana orientacji modułów może odbywać się również podczas zamiany miejscami pary modułów lub przenoszenia modułów. Zmiana położenia modułów może odbywać się w ograniczonym zakresie, podobnie jak w programie dla topografii wierszowej. W temperaturze początkowej zmiana poło-

żenia modułów może obejmować całe podłoże układu. Zakres dopuszczalnych zmian położenia modułów zmniejsza się wraz z obniżaniem temperatury [6–7, 17].

Funkcja kosztu algorytmu symulowanego wyżarzania (pierwszy etap) składa się z trzech składników. Pierwszy składnik jest równy ważonej sumie całkowitej estymowanej długości połączeń dla wszystkich węzłów układu, tak samo jak w programie dla topografii wierszowej. Uwzględniane jest rzeczywiste położenie końcówek modułów. Zachodzenie modułów na siebie jest dozwolone, ale jest uwzględnione w funkcji kosztu. Drugi składnik funkcji kosztu jest równy sumie powierzchni zachodzenia na siebie modułów. Dla każdego modułu wykonywana jest estymacja powierzchni niezbędnej do poprowadzenia wszystkich połączeń danego modułu. Podczas wyznaczania drugiego składnika funkcji kosztu powierzchnia wszystkich modułów jest zwiększana o wartość estymowanej powierzchni dla jego połączeń. Po każdej zmianie położenia modułu, estymowana powierzchnia dla połączeń modułu jest aktualizowana. Wyznaczenie drugiego składnika funkcji kosztu ułatwia podział podłoża układu na siatkę prostokątnych regionów. Udział drugiego składnika w funkcji kosztu określa waga tego składnika. Trzeci składnik funkcji kosztu występuje jedynie w przypadku, gdy położenie końcówek modułów nie jest ustalone [6–7, 17].

Temperatura początkowa w algorytmie symulowanego wyżarzania (pierwszy etap) jest specjalnie wyznaczana dla każdego układu, aby zapewnić akceptację prawie wszystkich nowych rozwiązań w tej temperaturze. Temperatura jest obniżana zgodnie ze wzorem 2. Współczynnik redukcji temperatury α zmienia się w zależności od bieżącej wartości temperatury. W każdej temperaturze jest tworzona określona liczba nowych rozwiązań, która jest równa wielokrotności liczby modułów. W przypadku układu składającego się z $30 \div 60$ modułów, zalecana liczba nowych rozwiązań jest około 400 razy większa od liczby modułów [6].

Podczas drugiego etapu kolejno odbywa się: wyznaczenie kanałów między modułami, przeznaczonych do prowadzenia połączeń, trasowanie globalne połączeń, które określa skupienie połączeń w poszczególnych kanałach oraz dalsza minimalizacja długości połączeń. Trzykrotne wykonanie powyższych czynności podczas tego etapu jest wystarczające do ustalenia rozwiązania. Minimalizacja długości połączeń wykonywana jest z wykorzystaniem algorytmu symulowanego wyżarzania, z niską temperaturą początkową. Postać funkcji kosztu jest taka sama, jak podczas pierwszego etapu. Nowe rozwiązania są tworzone w wyniku przenoszenia losowo wybranego modułu w losowo wybrane miejsce. Przenoszenie modułów może odbywać się w ograniczonym zakresie. Zakres dopuszczalnych zmian położenia modułów początkowo obejmuje niewielką część podłoża układu i zmniejsza się wraz z obniżaniem temperatury. Rozmiary wszystkich modułów są powiększane o powierzchnie niezbędne do poprowadzenia wszystkich połączeń danego modułu. Powierzchnie te są dokładnie wyznaczane na podstawie skupienia połączeń w poszczególnych kanałach, otrzymanego w wyniku trasowania globalnego. Trasowanie globalne zapewnia dokładniejsze wyznaczenie tych powierzchni, w porównaniu do metody polegającej na estymacji, stosowanej podczas pierwszego etapu. Wyznaczone powierzchnie mogą powodować konieczność zmian

rozmieszczenia modułów. W przeciwnym razie, zbyt mała pojemność kanału (moduły położone zbyt blisko siebie) mogłaby uniemożliwić poprowadzenie wszystkich połączeń. TimberWolfMC wersja 1.0 zapewniał otrzymanie znacznie lepszych rozwiązań pod względem całkowitej estymowanej długości połączeń oraz całkowitej powierzchni układu po wyznaczeniu wszystkich połączeń, w porównaniu do współczesnych programów oraz manualnego projektowania topografii układów [6–7].

W programie TimberWolfMC wersja 2.0 wprowadzono szereg zmian. Składnik funkcji kosztu wyrażający zachodzenie modułów na siebie jest efektywnie wyznaczany z wykorzystaniem podziału podłoża układu na siatkę prostokątnych regionów. Waga tego składnika funkcji kosztu podlega zmianom w trakcie obliczeń. Dodany został składnik funkcji kosztu, który umożliwia minimalizację opóźnień w układzie, w wyniku rozpatrywania krytycznych ścieżek w układzie. Tworzenie nowego rozwiązania polega na próbnym przenoszeniu losowo wybranego modułu w losowo wybrane miejsce. Jeżeli region, w którym znajduje się wybrane miejsce jest wolny, to moduł jest przenoszony i procedura tworzenia nowego rozwiązania kończy się. Jeżeli dany region jest zajęty, to wówczas moduł jest zamieniany miejscami z losowo wybranym modułem z danego regionu. W algorytmie symulowanego wyżarzania wprowadzono mechanizm utrzymywania stopnia akceptacji nowych rozwiązań na pożądanej wartości. TimberWolfMC wersja 2.0 zapewniał otrzymanie lepszych rozwiązań pod względem całkowitej długości połączeń oraz całkowitej powierzchni układu po wyznaczeniu wszystkich połączeń, w porównaniu do współczesnych programów, opartych na manualnym projektowania topografii układu. TimberWolfMC wersja 2.0 umożliwia rozmieszczenie modułów w znacznie krótszym czasie, w porównaniu do wersji 1.0, przy zachowaniu takiej samej jakości rozwiązań [20].

Program TimberWolf umożliwia również rozmieszczanie modułów typu „soft". Moduły typu „soft" posiadają jedynie określoną powierzchnię, natomiast stosunek wysokości i szerokości modułu może zmieniać się w zadanym przedziale, a położenie końcówek modułu nie jest określone. Z tego względu TimberWolf może być stosowany również podczas etapu planowania układu. Program może być również stosowany w układach o topografii mieszanej [6-7, 17, 22].

3.2. MGP, MPG-MS

Program MGP (wcześniej MPG) umożliwia globalne rozmieszczanie modułów. Program wykorzystuje wielopoziomową optymalizację oraz algorytm symulowanego wyżarzania. W programie MGP zastosowano podobny mechanizm, jak w przypadku wielopoziomowego podziału układu, przedstawionego w drugiej części przeglądu. Łączenie modułów odbywa się z użyciem algorytmu hMETIS i umożliwia zmniejszanie liczby modułów w liście połączeń dla kolejnych wyższych poziomów. Łączenie modułów na kolejnych poziomach jest powtarzane do momentu, kiedy lista połączeń zawiera odpowiednio małą liczbę modułów. Na najwyższym poziomie oraz na kolejnych niższych poziomach moduły są rozmieszczane z użyciem algorytmu symulowanego wy-

żarzania. Podłoże układu jest podzielone na siatkę prostokątnych regionów. Położenie modułu jest określone przynależnością do jednego z regionów. Na kolejnych niższych poziomach wcześniej połączone moduły są rozdzielane. Położenie modułów powstających w wyniku rozdzielenia połączonych modułów jest wyznaczane na podstawie położenia modułu, który jest rozdzielany. Po wykonaniu wielopoziomowej optymalizacji, moduły poddawane są rozmieszczaniu szczegółowemu, ponieważ przedstawiony algorytm nie wyklucza obecności zachodzenia modułów na siebie. Dla układów Standard Cell rozmieszczanie szczegółowe może być wykonane z użyciem programu DOMINO [23-25].

Podłoże układu jest podzielone na siatkę prostokątnych regionów. Położenie modułu jest określone przynależnością do jednego z regionów (współrzędnymi środka geometrycznego regionu). Program MGP wykorzystuje założenie, że jeżeli powierzchnia regionu nie będzie znacznie mniejsza od sumy powierzchni modułów umieszczonych w tym regionie, to w wyniku niewielkich przesunięć i niewielkiego wzrostu długości połączeń (kosztu), można otrzymać rozwiązanie bez zachodzenia modułów na siebie. Funkcja kosztu w algorytmie symulowanego wyżarzania jest równa całkowitej estymowanej długości połączeń lub jest miarą skupienia połączeń na powierzchni układu. W przypadku drugiej postaci funkcji kosztu, zmiana funkcji kosztu jest wyznaczana z użyciem szybkiego algorytmu trasowania globalnego. Program jedynie uaktualnia trasowanie globalne dla tych końcówek węzłów układu, które zmieniają położenie w nowym rozwiązaniu. Taka postać funkcji kosztu sprzyja rozwiązaniom o niskim skupieniu połączeń. Nowe rozwiązania są tworzone w wyniku przenoszenia losowo wybranego modułu do losowo wybranego regionu podłoża układu lub do regionu zapewniającego największy spadek wartości estymowanej długości połączeń. Jeżeli wybrany region nie jest w stanie pomieścić przenoszonego modułu, stosowana jest specjalna procedura, zmniejszająca zachodzenie modułów na siebie. Jest ona niezbędna, ponieważ zachodzenie modułów na siebie nie jest uwzględnione w funkcji kosztu. Dopuszcza się możliwość umieszczenia modułu w regionie, jeżeli jego powierzchnia przekracza wolną powierzchnię regionu, ale tylko w ograniczonym zakresie. Moduły umieszczone w regionie zmniejszają jego wolną powierzchnię. W celu określenia położenia modułów, które nie „mieszczą" się w regionach, procedura wykorzystuje hierarchiczny podział podłoża układu na regiony. Regiony wykorzystywane podczas rozmieszczania są grupowane i tworzą regiony na wyższym poziomie, np. na poziomie 0 podłoże jest podzielone na siatkę 8×8 regionów, na poziomie 1 na siatkę 4×4 regiony, a na poziomie 2 na siatkę 2×2 regiony. Jeżeli moduł nie mieści się w danym regionie na poziomie 0, rozpatrywane są kolejno regiony wyższego poziomu, które zawierają rozpatrywany region na poziomie 0. Położenie modułu określa najmniejszy region wyższego poziomu, który może „pomieścić" rozpatrywany moduł oraz posiada dodatkową własność, że wszystkie jego regiony wyższego poziomu, również są w stanie „pomieścić" rozpatrywany moduł. Podczas rozmieszczania modułów na kolejnych niższych poziomach listy połączeń, globalne położenie modułów jest coraz dokładniej określane, ponieważ moduły są rozdzielane na niższych poziomach listy połączeń i zmniejsza się

ich powierzchnia. Temperatura początkowa algorytmu symulowanego wyżarzania dla kolejnych niższych poziomów rozmieszczania jest odpowiednio obniżana. W przeciwnym przypadku, rozmieszczanie na niższych poziomach listy połączeń zaczynałoby się praktycznie od losowego rozmieszczenia modułów, bez uwzględnienia rozmieszczenia uzyskanego na wyższym poziomie. Temperatura w algorytmie symulowanego wyżarzania jest obniżana w zależności od stopnia akceptacji nowych rozwiązań. Program MGP / DOMINO dla układów Standard Cell umożliwia otrzymanie rozwiązań, o nieznacznie mniejszej całkowitej estymowanej długości połączeń i w znacznie krótszym czasie, w porównaniu do programu GORDIANL / DOMINO. Rozmieszczanie modułów może być prowadzone w sposób zapewniający znaczne zmniejszenie skupienia połączeń w układzie [23].

Program MPG-MS umożliwia rozmieszczanie modułów w układach o topografii mieszanej, w których moduły mogą znacznie różnić się rozmiarami. Sposób rozmieszczania modułów jest prawie taki sam jak w programie MGP. Różnice polegają głównie na odrębnym traktowaniu tzw. dużych modułów (ang. big objects), ustalanych przez program. Podczas łączenia modułów duże moduły uczestniczą w łączeniu jedynie na ostatnich, najwyższych poziomach listy połączeń. W ten sposób, podczas rozmieszczania modułów na najwyższych poziomach, duże moduły nie wchodzą w skład klastrów (ang. cluster). Po wykonaniu rozmieszczenia modułów na każdym poziomie listy połączeń, usuwane jest ewentualne zachodzenie dużych modułów na siebie, ponieważ rozwiązanie jest najpierw tworzone od określenia położenia dużych modułów. Zachodzenie dużych modułów na siebie jest usuwane w wyniku możliwie najmniejszych przesunięć. Po usunięciu zachodzenia dużych modułów na siebie, ich położenie jest trwale ustalane i nie zmienia się podczas rozmieszczania modułów na niższych poziomach listy połączeń. Likwidacja zachodzenia dużych modułów na siebie jest wykonywana tylko w wypadku stwierdzeniu jej wykonalności (ograniczone rozmiary podłoża układu). W tym celu stosowane jest kodowanie z użyciem pary kolejności modułów (ang. sequence-pair). Zachodzenie dużych modułów na siebie jest usuwane metodą heurystyczną (ang. longest path compaction). Jeżeli całkowite usunięcie zachodzenia dużych modułów na siebie nie jest możliwe, wykonywane jest częściowe jego usunięcie metodą heurystyczną, a następnie położenie dużych modułów, które nie zachodzą już na inne duże moduły jest trwale ustalane. W celu ustalenia położenia pozostałych dużych modułów kilkakrotnie kolejno mogą być wykonane następujące czynności: rozmieszczanie modułów na danym poziomie (z trwale ustalonym położeniem części dużych modułów), częściowe usuwanie zachodzenia dużych modułów na siebie i trwałe ustalanie położenia dużych modułów, których zachodzenie na siebie zostało usunięte. W programie MPG-MS nowe rozwiązania w algorytmie symulowanego wyżarzania są tworzone również w wyniku zmiany orientacji modułów. Program MPG-MS umożliwia otrzymanie rozwiązań o znacznie mniejszej całkowitej estymowanej długości połączeń, w porównaniu do rozwiązania wykorzystującego program Capo i Parquet [25, 56-57].

3.3. VPR

VPR jest programem przeznaczonym dla układów FPGA. VPR wykonuje rozmieszczanie modułów oraz trasowanie globalne i szczegółowe. Rozmieszczanie modułów odbywa się z wykorzystaniem algorytmu symulowanego wyżarzania. Nowe rozwiązania w algorytmie symulowanego wyżarzania są tworzone w wyniku zamiany miejscami losowo wybranej pary modułów lub końcówek całego układu oraz w wyniku przenoszenia losowo wybranego modułu lub końcówki całego układu w losowo wybrane miejsce. Końcówki całego układu mogą nie być rozmieszczane, jeżeli ich położenie zostanie określone przed rozmieszczaniem. Moduły oraz końcówki całego układu mogą przyjmować jedynie takie położenia, które są dozwolone w danym układzie FPGA. Zmiana położenia modułów może odbywać się w ograniczonym zakresie. W temperaturze początkowej zmiany położenia modułów mogą obejmować całe podłoże układu. Zakres dopuszczalnych zmian położenia modułów zmniejsza się stopniowo wraz z obniżaniem temperatury [26–27].

Funkcja kosztu algorytmu symulowanego wyżarzania jest równa sumie estymowanej długości połączeń dla wszystkich węzłów układu. Estymacja długości połączeń odbywa się metodą połowy obwodu, przy założeniu, że wszystkie końcówki modułów są położone w środku geometrycznym modułu. Wiadomo, że estymacja długości połączeń metodą połowy obwodu jest niedokładna dla węzłów zawierających więcej niż trzy końcówki. Rozbieżność między rzeczywistą długością połączeń po wyznaczeniu połączeń oraz wartością estymowaną wzrasta wraz z liczbą końcówek w węźle. Dlatego zastosowano specjalne współczynniki korygujące wartość estymowanej długości połączeń, w zależności od liczby końcówek w węźle. Podczas estymacji połączeń poszczególnych węzłów układu, wartość estymowana dla danego węzła jest mnożona przez współczynnik korygujący, odpowiedni dla liczby końcówek w tym węźle. Wartość współczynników korygujących estymację długości połączeń zmienia się w przedziale od 1 — dla 3 końcówek w węźle, do 2,79 — dla 50 końcówek w węźle [26–27].

Początkowe rozmieszczenie modułów na podłożu układu oraz końcówek całego układu na obrzeżach podłoża jest określane w sposób losowy. Temperatura początkowa w algorytmie symulowanego wyżarzania jest ustalana na podstawie symulacji wstępnych. Temperatura początkowa jest ustalana tak, aby prawie wszystkie nowe rozwiązania były akceptowane. Temperatura jest obniżana zgodnie ze wzorem 2. Współczynnik redukcji temperatury α zmienia się w zakresie $0,5 \div 0,95$, w zależności od bieżącej wartości temperatury. W harmonogramie chłodzenia programu VPR zastosowano wolniejsze obniżanie temperatury w zakresie średnich temperatur. Dla każdej temperatury jest tworzonych $10n^{4/3}$ nowych rozwiązań, gdzie n jest sumą rozmieszczanych modułów i końcówek całego układu. Stosowany jest mechanizm utrzymywania stopnia akceptacji nowych rozwiązań na pożądanej wartości, w wyniku zmiany zakresu dopuszczalnych zmian położenia modułów. Wykonywanie algorytmu symulowanego wyżarzania jest przerywane, jeżeli temperatura zostanie obniżona poniżej określonej wartości. Dla dostatecznie niskiej temperatury prawdopodobieństwo akceptacji rozwiązania powo-

dującego wzrost funkcji kosztu, które umożliwiłoby opuszczenie minimum lokalnego funkcji kosztu jest bardzo małe. VPR umożliwia również minimalizację opóźnień w układzie. Program VPR zapewnia otrzymanie rozwiązań o znacznie lepszej jakości, w porównaniu do wielu programów przeznaczonych dla układów FPGA [26-28]. Kod źródłowy programu jest powszechnie dostępny [29].

3.4. SPECJALNE SPOSOBY KODOWANIA ROZWIĄZAŃ

Podczas rozmieszczania modułów w układach o topografii swobodnej lub podczas etapu planowania układu bardzo często stosowany jest algorytm symulowanego wyżarzania, który stosuje specjalne sposoby kodowania rozwiązań. Rozwiązanie jest określane wzajemnym położeniem modułów względem siebie, z użyciem specjalnego kodu. Taki sposób reprezentacji rozwiązania znacznie zmniejsza liczbę możliwych rozwiązań, w porównaniu do reprezentacji rozwiązań opartych na bezwzględnych współrzędnych modułów na podłożu układu. Powyższe kodowanie rozwiązań umożliwia zatem skrócenie czasu obliczeń. Kodowanie rozwiązań powinno spełnić następujące warunki:

a. liczba wszystkich możliwych rozwiązań jest skończona
b. każde rozwiązanie jest fizycznie możliwe do wykonania
c. wyznaczenie topografii układu na podstawie rozwiązania jest możliwe w czasie wielomianowym, względem liczby modułów
d. istnieje rozwiązanie, które jest rozwiązaniem optymalnym.

Drugi warunek jest bardzo ważny, ponieważ w algorytmie symulowanego wyżarzania nowe rozwiązania są tworzone na podstawie obecnego rozwiązania. Jeżeli kodowanie nie gwarantuje, że wszystkie rozwiązania są fizycznie możliwe do wykonania, to skuteczność algorytmu nie może być zagwarantowana. Funkcja kosztu problemu rozmieszczania modułów zwykle jest ważoną sumą całkowitej estymowanej długości połączeń (metoda połowy obwodu) oraz całkowitej powierzchni układu. W celu określenia zmiany funkcji kosztu, a także ostatecznego rozwiązania problemu rozmieszczenia niezbędne jest wyznaczenie topografii układu, odpowiadającej danemu rozwiązaniu. Z tego względu trzeci warunek jest również bardzo ważny, a w szczególności czasowa złożoność algorytmu wyznaczania topografii dla danego rozwiązania. Temperatura początkowa w algorytmie symulowanego wyżarzania, harmonogram chłodzenia, liczba nowych rozwiązań tworzonych w każdej temperaturze oraz warunek zakończenia algorytmu są z reguły określane podobnie jak w typowym algorytmie symulowanego wyżarzania [7, 30–36, 56–57].

Kodowanie z użyciem następujących relacji dla wszystkich par modułów: moduł *a* „po lewej" modułu *b*, „po prawej", „poniżej" oraz „powyżej" nie może być stosowane, ponieważ w takim kodowaniu bardzo dużo rozwiązań byłoby fizycznie niemożliwych do wykonania. Przykładem takiego rozwiązania może być następujące rozwiązanie: moduł *a* po lewej modułu *b*, moduł *b* po lewej modułu *c*, moduł *c* po lewej modułu *a*. Z tego

względu są stosowane specjalne metody kodowania rozwiązania: z użyciem Odwrotnej Notacji Polskiej (ang. reverse Polish notation), z użyciem pary kolejności modułów SP (ang. sequence-pair), BSG (ang. bounded-sliceline grid), TCG (ang. transitive closure graph), O–drzewo (ang. O–tree, ordered tree), B*–drzewo (ang. B*–tree), CBL (ang. corner block list), TBS (ang. twin binary sequences). Powyższe metody kodowania wykluczają możliwość zachodzenia modułów na siebie [7, 30–37, 56–57]. Programy wykorzystujące kodowanie TCG i B*–drzewo są powszechnie dostępne w postaci kodu źródłowego [58].

4. SIECI NEURONOWE

Sieci neuronowe (ang. neural network) mogą być również stosowane podczas projektowania topografii układów VLSI. Istnieje wiele rodzajów sieci neuronowych [2, 4, 13–16, 38–42]. Sieci neuronowe cieszą się dużym zainteresowaniem, ze względu na ich zdolność równoległego przetwarzania danych przez wiele pracujących równocześnie elementów sieci. Zalety sieci neuronowych są najlepiej wykorzystane w przypadku sprzętowej realizacji sieci, co umożliwia uzyskanie rozwiązania w krótkim czasie. W dodatku, wzrost wymiaru problemu nie pociąga wtedy za sobą istotnego wzrostu czasu oczekiwania na rozwiązanie [39, 42]. W związku z tym, zalety sprzętowej realizacji sieci zasługują na uwagę, szczególnie ze względu na możliwość zastosowania w systemach rekonfigurowalnych, gdzie czas odgrywa decydującą rolę [61].

Przykładem sieci neuronowych, które mogą być stosowane do rozmieszczania modułów w układach VLSI jest sieć samoorganizująca się [13, 43–46] oraz sieć Hopfielda [2, 14, 47–53]. Przykłady te zostaną przedstawione w kolejnych podrozdziałach.

4.1. SIEĆ SAMOORGANIZUJĄCA SIĘ

W sieci samoorganizującej się (ang. self-organizing network) działającej na zasadzie współzawodnictwa każdy neuron jest połączony ze wszystkimi składowymi wektora wejściowego $\mathbf{x}$. Wagi połączeń synaptycznych tworzą wektor wag $\mathbf{W}$ danego neuronu w sieci. Uczenie sieci według algorytmu Kohonena (ang. Kohonen algorithm) polega na prezentacji ciągu wektorów $\mathbf{x}$. Neuron, którego wektor wag najmniej różni się od prezentowanego wektora $\mathbf{x}$ jest tzw. neuronem zwycięzcą. Dla neuronu zwycięzcy odległość między jego wektorem wag oraz prezentowanym wektorem $\mathbf{x}$ jest w wybranej metryce najmniejsza. Dla neuronu zwycięzcy jest wyznaczane sąsiedztwo o określonym promieniu. Neuron zwycięzca oraz wszystkie neurony położone w obszarze sąsiedztwa podlegają uczeniu, w wyniku zmiany ich wektorów wag w kierunku prezentowanego wektora $\mathbf{x}$, zgodnie ze wzorem

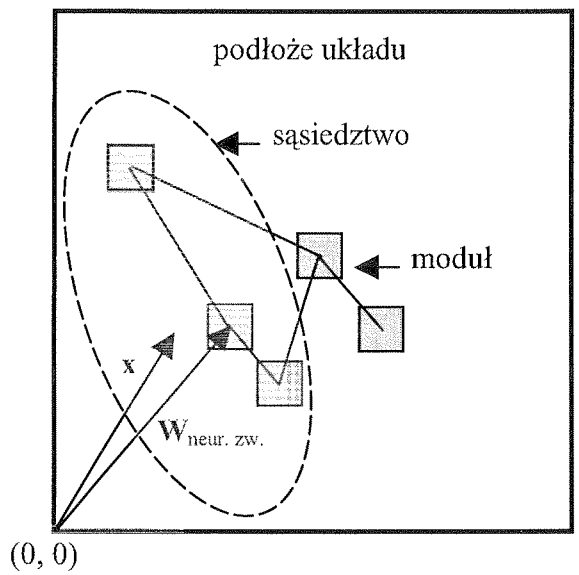
$$\mathbf{W}_i^{k+1} = \mathbf{W}_i^k + \eta_i^k \left(x - \mathbf{W}_i^k\right) \tag{3}$$

gdzie: $\mathbf{x}$ — prezentowany wektor, $\mathbf{W}_i^{k+1}$, $\mathbf{W}_i^k$ — wektor wag i-tego neuronu z sąsiedztwa neuronu zwycięzcy w chwilach $k+1$ i k, η_i^k — współczynnik uczenia i-tego neuronu w chwili k.

Promień sąsiedztwa neuronu zwycięzcy maleje wraz z upływem czasu. Wartość współczynnika uczenia neuronów z sąsiedztwa maleje wraz ze wzrostem odległości neuronu od neuronu zwycięzcy oraz wraz z upływem czasu. Neurony spoza sąsiedztwa neuronu zwycięzcy nie podlegają uczeniu [4, 13, 15, 40–42].

Rozmieszczanie modułów w układach VLSI może być wykonane z użyciem przedstawionej sieci samoorganizującej się. W przypadku rozmieszczania modułów z użyciem sieci samoorganizującej się nie jest stosowana funkcja kosztu. Każdy moduł rozmieszczany na podłożu układu jest reprezentowany przez jeden neuron w sieci. Wektor wag $\mathbf{W}$ odpowiadający każdemu neuronowi jest wektorem określonym przez współrzędne modułu na podłożu układu. Początkowe rozmieszczenie modułów, a tym samym początkowe wartości wektorów wag neuronów jest określone w sposób losowy. Uczenie sieci polega na prezentacji ciągu wektorów $\mathbf{x}$, które określają pewne położenie na podłożu układu. Neuron zwycięzca jest określany z wykorzystaniem miary odległości w postaci kwadratu odległości euklidesowej. W przypadku rozmieszczania modułów sąsiedztwo neuronu zwycięzcy nie jest określane poprzez zwykłe przestrzenne sąsiedztwo, lecz poprzez połączenia modułu reprezentowanego przez neuron zwycięzcę z pozostałymi modułami układu (reprezentowanymi przez pozostałe neurony). Tworzony jest hierarchiczny podział modułów (neuronów) układu. Moduł reprezentowany przez neuron zwycięzcę tworzy poziom 1. Na poziomie 2 znajdują się moduły, które są połączone z modułem znajdującym się na poziomie 1. Na kolejnych wyższych poziomach są umieszczane wszystkie te moduły, które są połączone z modułami położonymi na sąsiednim niższym poziomie. Sąsiedztwo neuronu zwycięzcy początkowo obejmuje wszystkie poziomy modułów (neuronów), wraz z upływem czasu ilość poziomów określających sąsiedztwo maleje. Współczynnik uczenia neuronów z sąsiedztwa neuronu zwycięzcy jest uzależniony od stopnia połączenia modułu reprezentowanego przez ten neuron z modułem neuronu zwycięzcy. Neurony modułów silnie połączonych z modułem neuronu zwycięzcy podlegają silniejszemu uczeniu. Współczynnik uczenia neuronów maleje również wraz z upływem czasu. Położenie końcówek modułów nie jest uwzględniane. Zastosowanie odpowiednich funkcji gęstości prawdopodobieństwa, określających położenie prezentowanych wektorów $\mathbf{x}$ na podłożu układu umożliwia rozmieszczanie modułów w matrycach bramkowych GA oraz układach Standard Cell. Rysunek 4 ilustruje sposób uczenia neuronów. Dla matryc bramkowych GA sieć samoorganizująca się umożliwiła otrzymanie rozwiązania o mniejszej długości połączeń, w porównaniu do współczesnego programu wykorzystującego metodę analityczną. Dla układu Standard Cell sieć samoorganizująca się umożliwiła otrzymanie rozwiązań o zbliżonej estymowanej długości połączeń oraz powierzchni układu, w porównaniu do współczesnego programu opartego na algorytmie symulowanego wyżarzania. Sieć potrzebowała jednak znacznie krótszego czasu na wykonanie obliczeń [13, 43].

Sieć samoorganizująca się umożliwia również rozmieszczanie modułów w układach o topografii swobodnej. W tym celu każdy moduł układu jest reprezentowany przez wiele neuronów w sieci, określających rozmiary modułów oraz położenie jego końcówek. Możliwe jest również rozmieszczanie końcówek całego układu na obrzeżach podłoża układu. Neurony odpowiadające końcówkom całego układu podlegają wtedy również uczeniu, na tych samych zasadach. Rozmieszczanie modułów w układach o topografii swobodnej umożliwiło zarówno dobrą minimalizację długości połączeń, jak również minimalizację powierzchni układu [13, 43].



Rys. 4. Uczenie sieci samoorganizującej się dla problemu rozmieszczania modułów

Fig. 4. Training of the self-organizing network for the cell placement problem

Sieć samoorganizująca się może być stosowana w celu optymalizacji topografii układu ze względu na aspekt termiczny [45–46].

4.2. SIEĆ HOPFIELDA

Sieć Hopfielda można zastosować do rozwiązywania problemów optymalizacyjnych. Wykorzystuje się własność sieci Hopfielda, która polega na minimalizacji energii sieci. Jeżeli rozwiązywany problem optymalizacyjny można sprowadzić do problemu minimalizacji funkcji kwadratowej, to sieć może posłużyć do rozwiązania tego problemu. Sieć neuronowa minimalizując swoją energię może jednocześnie znaleźć minimum funkcji, która jest sformalizowanym zapisem problemu optymalizacyjnego. Funkcja ta nazywa się funkcją kosztu lub celu i może być zapisana w postaci

$$E(V) = Ah_1(V) + Bh_2(V) + Ch_3(V) + Dh_4(V) \tag{4}$$

gdzie: E — funkcja kosztu, h_1, h_2, h_3, h_4 — funkcje stanowiące składniki funkcji kosztu, A, B, C, D — stałe, V — sygnał wyjściowy neuronu.

We wzorze 4 należy zastosować takie funkcje $h(V)$, które prawidłowo określą dany problem optymalizacyjny, razem ze wszystkimi ograniczeniami wymaganymi od poprawnego rozwiązania. W tym celu należy określić liczbę niezbędnych neuronów tworzących sieć oraz sposób interpretowania sygnałów wyjściowych poszczególnych neuronów. Do rozwiązania danego problemu optymalizacyjnego angażowana jest cała sieć neuronowa. Następnie należy określić wartości wag połączeń między neuronami oraz wartości zewnętrznych sygnałów wejściowych neuronów — poprzez przyrównanie wzoru określającego energię sieci Hopfielda do wzoru 4 wyrażającego funkcję kosztu problemu optymalizacyjnego. Podczas rozwiązywania problemów optymalizacyjnych z użyciem sieci Hopfielda zwykle pojawia się problem, polegający na osiąganiu przez sieć minimów lokalnych funkcji energii. Rozwiązaniem w ramach klasycznego modelu sieci jest wykonywanie wielu prób z inicjalizacją sieci losowymi wartościami. Wielokrotne próby uwzględniające różne punkty startowe zwiększają prawdopodobieństwo znalezienia optymalnego rozwiązania [2, 4, 14-16, 38-41, 47-51].

Sieć Hopfielda można zastosować do rozwiązania problemu optymalnej orientacji modułów w układach VLSI. W układach o topografii swobodnej każdy moduł może posiadać osiem różnych orientacji, które są rezultatem obrotu oraz odbicia względem jego osi symetrii, zgodnie z rysunkiem 1. Zakładając ustaloną pozycję modułu, sieć Hopfielda może posłużyć do określenia optymalnej orientacji modułu, tak aby całkowita długość połączeń między modułami była minimalna. Ze względu na ustaloną pozycję modułów, sieć rozpatruje jedynie odbicia względem poziomych oraz pionowych osi symetrii modułów. Obroty modułów nie są rozpatrywane, ponieważ mogłyby spowodować zachodzenie modułów na siebie lub zbytnie skupienie połączeń wokół modułów, uniemożliwiające poprowadzenie wszystkich połączeń. Sieć składa się z $4n$ neuronów, gdzie n oznacza liczbę modułów w układzie. Odpowiednia modyfikacja sieci Hopfielda zapewnia otrzymanie bardzo dobrych rezultatów dla rozpatrywanego problemu. Dla problemów o rozmiarze do 14 modułów, sieć umożliwia otrzymanie rozwiązania, które jest minimum globalnym funkcji kosztu. Optymalność rozwiązań została sprawdzona z zastosowaniem wyszukiwania wyczerpującego, które sprawdza wszystkie możliwe rozwiązania. Dla większych problemów optymalność rozwiązań nie została sprawdzona, ze względu na lawinowo narastający czas obliczeń. Sieć Hopfielda może być stosowana również dla dużych układów, składających się nawet z 300 modułów [14, 47].

Sieć Hopfielda może posłużyć również do rozmieszczania modułów w układach VLSI, umożliwiając minimalizację całkowitej długości połączeń w układzie. W tym celu zakłada się, że długość podłoża układu jest podzielona na K odcinków o rozmiarze jednostkowym, a szerokość płytki na L odcinków o rozmiarze jednostkowym. Podłoże układu jest zatem podzielone na siatkę kwadratowych części. Każdy moduł powinien być umieszczony w dokładnie jednej części. W każdej części nie może być umieszczonych więcej niż jeden moduł. W związku z tym, w celu rozmieszczenia X modułów

musi być spełniony warunek $X \leq KL$. Ponadto zakłada się, że końcówki modułów są położone w środku geometrycznym części układu, w której dany moduł jest umieszczony. W celu rozmieszczenia X modułów buduje się sieć, w której liczba neuronów jest równa XKL. Funkcja kosztu, która jest minimalizowana przez sieć jest sumą całkowitej estymowanej długości połączeń oraz ograniczeń. Ze względu na sposób połączeń między neuronami w sieci, do estymacji długości połączeń jest stosowana metoda oparta na grafie pełnym. Miarą długości połączeń jest metryka Manhattan. Ograniczenia w funkcji kosztu zapewniają rozmieszczenie wszystkich modułów układu w oddzielnych częściach podłoża. Przyrównanie wzoru określającego energię sieci Hopfielda do wzoru wyrażającego funkcję kosztu problemu rozmieszczania pozwala określić wartości wag połączeń między neuronami oraz wartości zewnętrznych sygnałów wejściowych neuronów. Symulacja sieci może być przeprowadzona z wykorzystaniem metody Eulera. Sieć Hopfielda może być również stosowana w celu optymalizacji topografii układu ze względu na aspekt termiczny [2, 48-51].

Obecnie zostaną przedstawione rezultaty rozmieszczania modułów z minimalizacją długości połączeń dla trzech badanych układów. Dla każdego z układów wykonano 100 prób rozmieszczenia modułów. Pierwszy układ składał się z 40 modułów oraz 73 węzłów, składających się z 2 końcówek. Moduły były rozmieszczane na podłożu o rozmiarach 7×7 części. Optymalne rozmieszczenie modułów dla tego układu posiada długość połączeń równą 205 w metryce Manhattan. Tabela 1 przedstawia rezultaty rozmieszczania modułów dla pierwszego układu. Długość połączeń dla najlepszego rozwiązania otrzymanego z użyciem sieci Hopfielda jest ok. 2,4% większa, w porównaniu do rozwiązania optymalnego. Rysunek 5 przedstawia połączenia między modułami dla optymalnego rozmieszczenia modułów oraz najlepszego rozwiązania otrzymanego z użyciem sieci Hopfielda. Połączenia poprowadzono w przestrzeni Euklidesa dla większej przejrzystości rysunku [51].

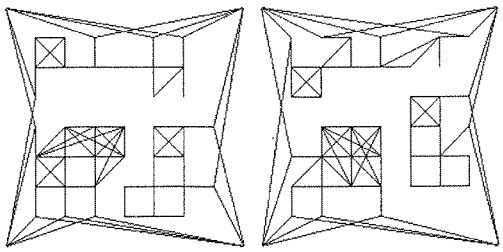
Tabela 1

Rezultaty otrzymane dla pierwszego układu

Achieved results for the first benchmark

	Wynik
koszt optymalnego rozwiązania	205
średni koszt	219,46
koszt najlepszego otrzymanego rozwiązania	210
średni koszt / koszt opt. rozw.	1,0705
koszt najlepszego rozw. / koszt opt. rozw.	1,0244

Drugi i trzeci układ są układami FPGA. Drugi układ składał się z 20 modułów (komórek CLB), 5 wejść całego układu, 16 wyjść całego układu oraz 25 węzłów,



(a) (b)

Rys. 5. Połączenia między modułami w pierwszym układzie: a) optymalne rozmieszczenie, b) najlepsze rozmieszczenie otrzymane siecią Hopfielda

Fig. 5. A diagram shows all the nets connections of the first circuit for: a) optimal placement, b) the best Hopfield network placement

zawierających $2 \div 17$ końcówek. Moduły były rozmieszczane na podłożu o rozmiarach 5×5 części. W obydwu układach wejścia i wyjścia całego układu posiadały ustalone położenie i nie podlegały rozmieszczaniu. W obydwu układach komórki CLB posiadały 4 wejścia i 1 wyjście. Tabela 2 przedstawia rezultaty rozmieszczania modułów dla drugiego układu. W tabeli zamieszczono rezultaty dla pięciu najlepszych rozwiązań otrzymanych siecią Hopfielda oraz dla rozmieszczenia modułów otrzymanego z użyciem programu VPR. Dla każdego rozmieszczenia modułów wykonano trasowanie połączeń z użyciem narzędzia trasowania połączeń programu VPR. W tabeli zamieszczono długość połączeń po wykonaniu trasowania połączeń oraz minimalną pojemność kanałów dla połączeń. Minimalna pojemność kanałów w tabeli jest równa minimalnej pojemności kanałów, która umożliwia poprowadzenie wszystkich połączeń w układzie. Minimalna pojemność kanałów jest równa maksymalnej liczbie poprowadzonych połączeń, w dowolnym kanale układu. Rysunek 6 przedstawia połączenia między modułami w drugim układzie dla rozmieszczenia modułów wykonanego programem VPR oraz dla najlepszego rozwiązania otrzymanego z użyciem sieci Hopfielda. Połączenia między modułami poprowadzono tak jak w grafie pełnym. Połączenia poprowadzono w przestrzeni Euklidesa dla większej przejrzystości rysunku [51].

Trzeci układ składał się z 54 modułów (komórek CLB), 34 wejść całego układu, 10 wyjść całego układu oraz 88 węzłów, zawierających $2 \div 11$ końcówek. Moduły były rozmieszczane na podłożu o rozmiarach 8×8 części. Tabela 3 przedstawia rezultaty rozmieszczania modułów dla trzeciego układu. Rysunek 7 przedstawia połączenia między modułami w trzecim układzie [51].

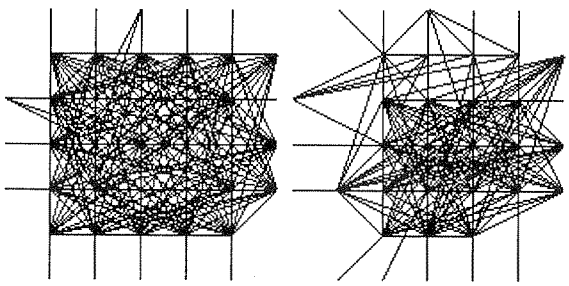
Rezultaty rozmieszczania modułów otrzymane z użyciem sieci Hopfielda są porównywalne do rezultatów rozmieszczania programu VPR. Niektóre rozwiązania otrzymane siecią Hopfielda umożliwiają wyznaczenie połączeń z użyciem kanałów, których pojemność jest mniejsza o jeden. Podczas projektowania topografii układów VLSI wskazane jest rozpatrzenie kilku najlepszych rozwiązań, ze względu na częste stosowanie różnych estymacji. Jedynie wyznaczenie połączeń umożliwia wiarygodną ocenę

Tabela 2

Rezultaty otrzymane dla drugiego układu

Achieved results for the second benchmark

Rozmieszczenie	Całkowita długość połączeń	Minimalna pojemność kanału
sieć Hopfielda 1	150	3
sieć Hopfielda 2	148	4
sieć Hopfielda 3	153	4
sieć Hopfielda 4	148	4
sieć Hopfielda 5	153	4
VPR	154	4



(a) (b)

Rys. 6. Połączenia między modułami w drugim układzie: a) rozmieszczenie programu VPR, b) najlepsze rozmieszczenie otrzymane siecią Hopfielda

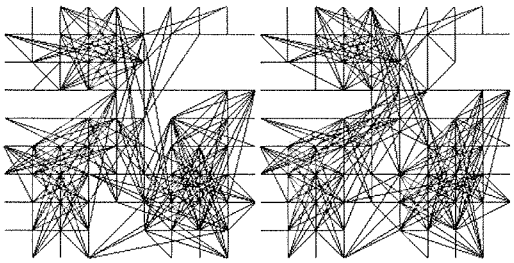
Fig. 6. A diagram shows all the nets complete-graph connections of the second circuit for: a) VPR placement, b) the best Hopfield network placement

Tabela 3

Rezultaty otrzymane dla trzeciego układu

Achieved results for the third benchmark

Rozmieszczenie	Całkowita długość połączeń	Minimalna pojemność kanału
sieć Hopfielda 1	418	5
sieć Hopfielda 2	423	5
sieć Hopfielda 3	425	4
sieć Hopfielda 4	430	4
sieć Hopfielda 5	425	5
VPR	408	5



Rys. 7. Połączenia między modułami w trzecim układzie: a) rozmieszczenie programu VPR, b) najlepsze rozmieszczenie otrzymane siecią Hopfielda

Fig. 7. A diagram shows all the nets complete-graph connections of the third circuit for: a) VPR placement, b) the best Hopfield network placement

jakości projektu układu. Potwierdzają to przedstawione przykłady, w których minimalizacja estymowanej długości połączeń jest tylko w przybliżeniu równoważna minimalizacji rzeczywistej długości połączeń po wykonaniu trasowania. Dla estymacji długości połączeń wykorzystującej graf pełny możliwa jest korekcja estymowanej długości tak, aby dokładniej określała rzeczywistą długość połączeń po ich wyznaczeniu [52–53]. Obliczenia zostały przeprowadzone na komputerze SGI2800, ACK CYFRONET AGH.

5. INNE METODY

Rozmieszczanie modułów w układach Standard Cell może być wykonane z użyciem algorytmu genetycznego lub strategii ewolucyjnej. Algorytm genetyczny operuje na wielu rozwiązaniach, natomiast strategia ewolucyjna na jednym rozwiązaniu. Obydwie metody umożliwiły otrzymanie rozwiązań, których jakość była porównywalna ze współczesną wersją programu TimberWolf. Czas potrzebny na wykonanie obliczeń metodą wykorzystującą algorytm genetyczny był porównywalny z czasem obliczeń programu TimberWolf. Metoda wykorzystująca strategię ewolucyjną zapewniała krótszy czas obliczeń [9, 54–55].

Rozmieszczanie modułów w układach o topografii mieszanej, które zawierają bardzo dużą liczbą komórek standardowych można wykonać posługując się następującym schematem: rozmieszczanie-planowanie topografii-rozmieszczanie. Powyższą metodę można zastosować w przypadku, gdy rozmiary podłoża układu są ustalone. Podczas pierwszego etapu makrobloki są dzielone na mniejsze części, porównywalne z rozmiarami komórek standardowych. Części makrobloków są łączone „sztucznymi" węzłami. Komórki standardowe oraz części makrobloków są rozmieszczane z użyciem dowolnego programu dla układów Standard Cell, który minimalizuje długość połączeń. „Stuczne" połączenia między częściami makrobloków mają zapewnić położenie wszystkich części danego makrobloku obok siebie po wykonaniu rozmieszczenia. Pod-

czas drugiego etapu określane jest położenie makrobloków na podstawie położenia ich części. Makrobloki mogą jednak zachodzić na siebie oraz na komórki standardowe i mogą nie mieścić się również na podłożu układu. Dlatego musi być wykonane planowanie układu z dalszą minimalizacją długości połączeń oraz z ustalonymi rozmiarami podłoża układu. Sąsiednie komórki standardowe są łączone w bloki typu „soft", a makrobloki są traktowane jak bloki typu „hard". Położenie bloków typu „soft" oraz typu „hard" jest wyznaczane podczas planowania układu, zakładając możliwość zmiany rozmiarów bloków typu „soft" w ustalonym przedziale. Drugi etap określa ostateczne położenie makrobloków w układzie. Podczas trzeciego etapu komórki standardowe są rozmieszczane z użyciem programu rozmieszczania dla układów Standard Cell, który również minimalizuje długość połączeń. Komórki standardowe są rozmieszczane w miejscach (wierszach), które nie są zajęte przez makrobloki. Zastosowanie programu Capo podczas pierwszego i trzeciego etapu oraz programu Parquet podczas drugiego etapu umożliwia otrzymanie rozwiązań o mniejszej całkowitej estymowanej długości połączeń, w porównaniu do programu komercyjnego [56–57]. Program Parquet umożliwia planowanie układu dla ustalonych rozmiarów podłoża układu, program jest powszechnie dostępny w postaci kodu źródłowego oraz plików wykonywalnych [58].

Istnieją również programy przeznaczone dla trójwymiarowych układów VLSI (3D VLSI). Programy te zwykle wykorzystują algorytm min-cut oraz metody analityczne. Ze względu na konieczność rozmieszczania znacznie większej liczby modułów, programy rozmieszczania dla układów 3D VLSI powinna cechować prawie liniowa czasowa złożoność obliczeniowa, względem liczby rozmieszczanych modułów. Przykładem programu rozmieszczania dla układów 3D VLSI jest program Gravity, powszechnie dostępny jest program dla układów 3D FPGA [29, 59].

Stale prowadzone są również prace umożliwiające sprzętowe rozmieszczanie modułów w układach VLSI. Sprzętowe rozwiązania umożliwiają wprowadzenie równoległego przetwarzania. Takie rozwiązania umożliwiłyby zmniejszenie czasu obliczeń o kilka rzędów wielkości. Z tego względu sprzętowe rozwiązania problemu rozmieszczania modułów mogą znaleźć zastosowanie w systemach rekonfigurowalnych, gdzie czas odgrywa decydującą rolę. Prowadzone są m.in. prace nad sprzętową realizacją algorytmu symulowanego wyżarzania, głównie pod kontem zastosowania w systemach rekonfigurowalnych [3, 60–61].

Wiele istotnych informacji dotyczących całej problematyki projektowania topografii układów VLSI można znaleźć na stronie internetowej VLSI CAD Bookshelf. Strona jest redagowana przez wybitnych naukowców, zajmujących się automatyzacją projektowania topografii układów VLSI. Wiele rozwiązań jest dostępnych w postaci kodu źródłowego [64].

6. KOMERCYJNE PROGRAMY ROZMIESZCZANIA

Program iTools jest produktem firmy InternetCAD.com. Program powstał na bazie programu akademickiego TimberWolf i wykorzystuje algorytm symulowanego wy-

żarzania. Program iTools umożliwia zarówno rozmieszczanie modułów, jak również trasowanie globalne połączeń oraz trasowanie szczegółowe. Umożliwia rozmieszczanie modułów z minimalizacją opóźnień w układzie. Program może być stosowany w układach Standard Cell, matrycach bramkowych GA oraz układach Sea-of-Gates. Program może być również stosowany w układach o topografii swobodnej i mieszanej oraz podczas etapu planowania topografii układu [22].

Firma Synopsys jest producentem pakietu projektowania układów scalonych Galaxy Design Platform. Składnikiem pakietu jest program Astro, który umożliwia rozmieszczanie modułów oraz trasowanie połączeń w układach Standard Cell. Program może być stosowany do projektowania układów w technologii 90nm i mniej, program zapobiega powstawaniu sprzężeń pojemnościowo-indukcyjnych w układzie. Program Astro wykorzystuje metodę gradientu sprzężonego podczas rozmieszczania modułów. Metoda rozmieszczania stosowana w programie Astro jest również wykorzystywana w innych programach pakietu: JupiterXT, Physical Compiler i IC Compiler. Program JupiterXT umożliwia planowanie topografii układu i może być również stosowany w układach o topografii mieszanej. Programy Physical Compiler i IC Compiler umożliwiają wykonanie syntezy logicznej, rozmieszczenia modułów oraz trasowania połączeń dla układów w technologii 130nm, 90nm i mniej. [65].

Firma Cadence jest producentem pakietów projektowania układów scalonych: Silicon Ensemble i SoC Encounter. Składnikiem pakietu Silicon Ensemble jest program QPlace, który umożliwia rozmieszczanie modułów. Program QPlace wykorzystuje metodę programowania kwadratowego oraz podział układu na części. Rozmieszczanie szczegółowe modułów odbywa się w wyniku wykonywania zamiany modułów parami, przenoszenia modułów oraz zmiany orientacji modułów. W końcowym etapie rozmieszczania program QPlace wykorzystuje również algorytm symulowanego wyżarzania. Pakiet SoC Encounter może być stosowany do projektowania układów w technologii 90nm, pakiet zapobiega powstawaniu sprzężeń pojemnościowo-indukcyjnych w układzie, może byś stosowany w układach o topografii mieszanej. Program rozmieszczania modułów w pakiecie SoC Encounter nazywa się nanoPlace. Program nanoPlace wykorzystuje metodę programowania kwadratowego oraz algorytmu.

Firma Mentor Graphics jest producentem pakietu projektowania układów scalonych. W skład pakietu wchodzi program AutoCells, który umożliwia rozmieszczanie modułów, trasowanie połączeń oraz kompresję topografii w układach Standard Cell [68].

Pakiet Xilinx Integrated Software Enviroment ISE firmy Xilinx umożliwia projektowanie układów w technologii FPGA. W skład pakietu wchodzi program PAR, który umożliwia rozmieszczanie komórek logicznych oraz trasowanie połączeń. Rozmieszczanie komórek logicznych odbywa się z wykorzystaniem metody konstrukcyjnej [69].

Pakiet firmy Altera dla układów FPGA nazywa się Altera Quartus II. W skład pakietu wchodzi program Quartus II PowerFit, który umożliwia rozmieszczanie komórek logicznych oraz trasowanie połączeń [70].

7. PODSUMOWANIE

Niniejsza praca jest czwartą, ostatnią częścią przeglądu, poświęconego metodom rozmieszczania bloków funkcjonalnych w projektowaniu topografii układów i systemów VLSI. Do metod rozmieszczania modułów, które są najczęściej stosowane podczas optymalizacji topografii układów VLSI należą: metody wykorzystujące algorytm min-cut, metody analityczne oraz metody oparte na algorytmie symulowanego wyżarzania [10]. Obecnie większość programów wykorzystuje minimalizację długości połączeń podczas rozmieszczania modułów. Minimalizacja długości połączeń może być prowadzona równocześnie z minimalizacją opóźnień w krytycznych częściach układu, co zapewnia spełnienie wymagań funkcjonalnych stawianych układowi. Podczas minimalizacji długości połączeń bardzo często jest wykorzystywana estymacja długości połączeń metodą połowy obwodu. Zaletą tej metody jest niski nakład obliczeniowy. Ponadto wykazano, że rozwój technologii wytwarzania układów i systemów scalonych, który charakteryzuje się coraz większą liczbą warstw metalizacji w układzie, przyczynia się do poprawy dokładności tej metody estymacji. W układach o dużej liczbie warstw metalizacji metoda połowy obwodu staje się bardziej dokładna, ponieważ większa część węzłów jest połączona najkrótszymi połączeniami [56-57]. Minimalizacja długości połączeń pośrednio przyczynia się do minimalizacji skupienia połączeń. Ze względu na brak skutecznych i dokładnych metod estymacji skupienia połączeń, dokładne określenie skupienia połączeń ciągle wymaga wykonania trasowania globalnego. Podejmowane są próby integracji algorytmów trasowania globalnego z algorytmami wykonującymi rozmieszczanie modułów [23].

Skuteczność programów rozmieszczania zależy od cech rozmieszczanego układu, co należy uwzględnić podczas wyboru programu. Programy minimalizujące długość połączeń oparte na metodach analitycznych gorzej radzą sobie z układami, które posiadają większą liczbę połączeń obejmujących znaczną część układu. W takich układach programy oparte na metodach analitycznych muszą kłaść większy nacisk na usuwanie zachodzenia między modułami, co powoduje pogorszenie długości połączeń. Problem ten nasila się w przypadku programów, których funkcja kosztu wykorzystuje kwadrat długości połączeń, ponieważ występuje wtedy jeszcze uprzywilejowanie długich połączeń. W przypadku programów wykorzystujących algorytm min-cut połączenia obejmujące znaczną część układu przyczyniają się do poprawy jakości otrzymywanych rozwiązań. W przypadku programów opartych na algorytmie symulowanego wyżarzania bardzo istotna jest dokładność, z jaką określane jest położenie modułów. Jeżeli położenie modułów określane jest przez przynależność modułów do określonych części podłoża układu, to współrzędne środków geometrycznych tych części określają położenie modułów. Zbyt duże rozmiary tych części mogą być powodem pogorszenia jakości rozwiązań. Powyższa sytuacja występuje w programie Dragon i jest szczególnie widoczna dla układów, które posiadają małą liczbę połączeń obejmujących znaczną część układu. Programy oparte na algorytmie symulowanego wyżarzania wymagają znacznie dłuższego czasu na wykonanie obliczeń, w porównaniu do programów wyko-

rzystujących algorytm min-cut oraz metody analityczne [24, 62-63]. Programy oparte na metodach analitycznych gorzej radzą sobie z układami, w których jest mało wolnej przestrzeni na podłożu układu, ze względu na utrudnioną eliminację zachodzenia modułów na siebie. W przypadku programów wykorzystujących algorytm min-cut sytuacja przedstawia się zupełnie inaczej. Programy oparte na algorytmie min-cut gorzej radzą sobie z układami, w których jest zbyt dużo wolnej przestrzeni na podłożu układu [10].

8. BIBLIOGRAFIA

1. M.J.S. Smith: *Application-Specific Integrated Circuits*, Addison Wesley Longman, 1997.
2. A. Kos: *Modelowanie hybrydowych układów mocy i optymalizacja ich konstrukcji ze względu na rozkład temperatury*, Kraków, Wydawnictwa AGH, 1994.
3. B.T. Preas, M.J. Lorenzetti (red.): *Physical Design Automation of VLSI Systems*, Menlo Park, Benjamin-Cummings, 1988.
4. S. Osowski: *Sieci neuronowe w ujęciu algorytmicznym*, Warszawa, WNT, 1996.
5. M.M. Vai: *VLSI Design*, CRC Press, 2001.
6. C. Sechen: *VLSI Placement and Global Routing Using Simulated Annealing*, Boston, Kluwer Academic Publishers, 1988, (dostępna w bibliotece WA 290).
7. D.F. Wong, H.W. Leong, C.L. Liu: *Simulated Annealing for VLSI Design*, Kluwer Academic Publishers, 1988.
8. W. Wolf: *Modern VLSI Design: a systems approach*, Englewood Cliffs, New Jersey, PTR Prentice Hall, 1994.
9. K. Shahookar, P. Mazumder: *VLSI Cell Placement Techniques*, ACM Computing Surveys, 1991, vol. 23, pp. 143–220.
10. C.J. Alpert, G.J. Nam, P.G. Villarrubia: *Effective Free Space Management for Cut-Based Placement via Analytical Constraint Generation*, IEEE Transactions on Computer-Aided Design, 2003, vol. 22, pp. 1343–1353.
11. T.C. Hu, E.S. Kuh (red.): *VLSI Circuit Layout: Theory and Design*, New York, IEEE Press, 1985.
12. S. Kirkpatrick, C.D. Gelatt, Jr., M.P. Vecchi: *Optimization by Simulated Annealing*, Science, 1983, vol. 220, no. 4598, pp. 671–680.
13. D.T. Pham, D. Karaboga: *Intelligent Optimisation Techniques*, Springer-Verlag London Limited, 2000.
14. J. Mańdziuk: *Sieci neuronowe typu Hopfielda. Teoria i przykłady zastosowań*, Warszawa, Akademicka Oficyna Wydawnicza EXIT, 2000.
15. J.A. Freeman, D.M. Skapura: *Neural networks: algorithms, applications, and programming techniques*, Addison-Wesley Publishing Company, 1991.
16. T. Khanna: *Foundations of neural networks*, Addison-Wesley Publishing Company, 1990.
17. C. Sechen, A. Sangiovanni-Vincentelli: *The TimberWolf Placement and Routing Package*, IEEE Journal of Solid-State Circuits, 1985, vol. 20, pp. 510–522.
18. C. Sechen, K.-W. Lee: *An Improved Simulated Annealing Algorithm for Row-Based Placement*, Proc. of the International Conference on Computer-Aided Design, 1987, pp. 478-481, http://www.sigda.org/Archives/ProceedingArchives/Iccad/.
19. C. Sechen, D. Braun, A. Sangiovanni-Vincentelli: *ThunderBird: A Complete Standard Cell Layout Package*, IEEE Journal of Solid-State Circuits, 1988, vol. 23, pp. 410–420.
20. W. Swartz, C. Sechen: *New Algorithms for the Placement and Routing of Macro Cells*, Proc. of the International Conference on Computer-Aided Design, 1990, pp. 336–339, http://www.sigda.org/Archives/ProceedingArchives/Iccad/.

21. W.-J. Sun, C. Sechen: *Efficient and Effective Placement for Very Large Circuits*, IEEE Transactions on Computer-Aided Design, 1995, vol. 14, pp. 349–359.
22. Strona internetowa firmy InternetCAD.com, http://www.internetcad.com.
23. C.-C. Chang, J. Cong, Z. Pan, X. Yuan: *Multilevel Global Placement With Congestion Control*, IEEE Transactions on Computer-Aided Design, 2003, vol. 22, pp. 395-409.
24. C.C. Chang, J. Cong, M. Romesis, M. Xie: *Optimality and Scalability Study of Existing Placement Algorithms*, IEEE Transactions on Computer-Aided Design, 2004, vol. 23, pp. 537–549.
25. C.-C. Chang, J. Cong, X. Yuan: *Multi-level Placement for Large-Scale Mixed-Size IC Design*, Proc. of the Asia and South Pacific Design Automation Conference, 2003, pp. 325-330, http://cadlab.cs.ucla.edu/~cong/papers/aspdac03.pdf.
26. V. Betz, J. Rose: *VPR: A New Packing, Placement and Routing Tool for FPGA Research*, Proc. 7th International Workshop on Field Programmable Logic and Applications, London, 1997, pp. 213–222, http://www.eecg.toronto.edu/~vaughn/papers/fpl97.pdf.
27. A. Marquardt, V. Betz, J. Rose: *Timing-Driven Placement for FPGAs*, Proc. International Symposium on FPGA, 2000, pp. 203–213, http://www.sigda.org/Archives/ProceedingArchives/Compendiums/papers/fpga/confsym.htm.
28. P. Maidee, C. Ababei, K. Bazargan: *Timing-Driven Partitioning-Based Placement for Island Style FPGAs*, IEEE Transactions on Computer-Aided Design, 2005, vol. 24, pp. 395–406.
29. VLSI CAD Bookshelf Slots and Entries, A.B. Kahng, I.L. Markov, http://vlsicad.eecs.umich.edu/BK/Slots/slots/FPGALayout.html.
30. H. Murata, K. Fujiyoshi, S. Nakatake, Y. Kajitani: *VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair*, IEEE Transactions on Computer-Aided Design, 1996, vol. 15, pp. 1518–1524.
31. H. Murata, K. Fujiyoshi, M. Kaneko: *VLSI/PCB Placement with Obstacles Based on Sequence Pair*, IEEE Transactions on Computer-Aided Design, 1998, vol. 17, pp. 60–68.
32. K. Fujiyoshi, H. Murata: *Arbitrary Convex and Concave Rectilinear Block Packing Using Sequence-Pair*, IEEE Transactions on Computer-Aided Design, 2000, vol. 19, pp. 224–233.
33. S. Nakatake, K. Fujiyoshi, H. Murata, Y. Kajitani: *Module Packing Based on the BSG-Structure and IC Layout Applications*, IEEE Transactions on Computer-Aided Design, 1998, vol. 17, pp. 519–530.
34. E.F.Y. Young, C.C.N. Chu, Z.C. Shen: *Twin Binary Sequences: A Nonredundant Representation for General Nonslicing Floorplan*, IEEE Transactions on Computer-Aided Design, 2003, vol. 22, pp. 457–469.
35. J.-G. Kim, Y.-D. Kim: *A Linear Programming-Based Algorithm for Floorplanning in VLSI Design*, IEEE Transactions on Computer-Aided Design, 2003, vol. 22, pp. 584–592.
36. F. Balasa, S.C. Maruvada, K. Krishnamoorthy: *On the Exploration of the Solution Space in Analog Placement With Symmetry Constraints*, IEEE Transactions on Computer-Aided Design, 2004, vol. 23, pp. 177–191.
37. H.H. Chan, I.L. Markov: *Practical Slicing and Non-slicing Block-Packing without Simulated Annealing*, Proc. of the Great Lakes Symposium on VLSI, 2004, pp. 282–287, http://vlsicad.eecs.umich.edu/BK/BloBB/PAPERS/p037-chan.pdf.
38. J.J. Hopfield, D.W. Tank: *„Neural" computation of decisions in optimization problems*, Biological Cybernetics, 1985, vol. 52, pp. 141–152.
39. R. Tadeusiewicz: *Sieci neuronowe*, Warszawa, Akademicka Oficyna Wydawnicza, 1993.
40. J. Hertz, A. Krogh, R.G. Palmer: *Wstęp do teorii obliczeń neuronowych*, Warszawa, WNT, 1995.
41. J. Żurada, M. Barski, W. Jędruch: *Sztuczne sieci neuronowe*, Warszawa, Wydawnictwo Naukowe PWN, 1996.
42. M. Glesner, W. Pöchmüller: *Neurocomputers*, London, Chapman & Hall, 1994.

43. C.-X. Zhang, D.A. Mlynski: *Mapping and Hierarchical Self-Organizing Neural Networks for VLSI Placement*, IEEE Transactions on Neural Networks, 1997, vol. 8, pp. 299–314.
44. A. Hemani, A. Postula: *Cell Placement by Self-Organisation*, Neural Networks, 1990, vol. 3, pp. 377–383.
45. P. Bratek, A. Kos: *Complex Optimisation of Topology of VLSI Circuits with Self-Organising Neural Nets*, Proc. of the MIXDES 1996 Mixed Design of Integrated Circuits and Systems, Łódź (Poland), June 1996, pp. 78–83.
46. P. Bratek, A. Kos: *Self-Organising Neural Computations for Optimisation of IC Topography in Thermal Aspect*, Proc. of the XIXth National Conference on Circuit Theory and Electronic Networks, Kraków-Krynica (Poland), October 1996, pp. II/627–632.
47. Y. Takefuji, K.-C. Lee, H. Aiso: *An artificial maximum neural network: a winner-take-all neuron model forcing the state of the system in a solution domain*, Biological Cybernetics, 1992, vol. 67, pp. 243–251.
48. H. Mączka, P. Dziurdzia, A. Kos: *Neural Algorithm for Minimisation of Total Length of Connections in VLSI Circuits*, Proc. of the XIXth National Conference on Circuit Theory and Electronic Networks, Kraków-Krynica (Poland), October 1996, pp. II/319–324.
49. A. Kos, Z. Nagórny: *Minimalizacja długości połączeń w układach elektronicznych z wykorzystaniem sieci Hopfielda*, Kwartalnik Elektroniki i Telekomunikacji, 2005, tom 51, z. 1, pp. 55–72.
50. Z. Nagórny, A. Kos: *Optymalizacja z wykorzystaniem zmodyfikowanej sieci Hopfielda*, Kwartalnik Elektroniki i Telekomunikacji, 2005, tom 51, z. 2, pp. 255–275.
51. A. Kos, Z. Nagórny: *A Modified Hopfield Neural Network for VLSI Placement*, Proc. of the MIXDES 2005 Mixed Design of Integrated Circuits and Systems, Kraków (Poland), June 2005, vol. 1, pp. 33–38.
52. A. Kos, Z. Nagórny: *Estymacja długości połączeń w układach VLSI*, IV Krajowa Konferencja Elektroniki, Darłówko Wschodnie, czerwiec 2005, pp. 159-164.
53. A. Kos, Z. Nagórny: *Estymacja długości połączeń w układach VLSI*, Elektronika, 2005, rok 46, nr 11, pp. 47–49.
54. K. Shahookar, P. Mazumder: *A Genetic Approach to Standard Cell Placement Using Meta-Genetic Parameter Optimization*, IEEE Transactions on Computer-Aided Design, 1990, vol. 9, pp. 500–511.
55. R.M. Kling, P. Banerjee: *ESP: Placement by Simulated Evolution*, IEEE Transactions on Computer-Aided Design, 1989, vol. 8, pp. 245–256.
56. S.N. Adya, I.L. Markov: *Fixed-Outline Floorplanning: Enabling Hierarchical Design*, IEEE Transactions on VLSI Systems, 2003, vol. 11, pp. 1120–1135.
57. S.N. Adya, I.L. Markov: *Consistent Placement of Macro-Blocks Using Floorplanning and Standard-Cell Placement*, Proc. of the International Symposium on Physical Design, 2002, pp. 12–17, http://www.sigda.org/Archives/ProceedingArchives/Ispd/.
58. VLSI CAD Bookshelf Slots and Entries, A.B. Kahng, I.L. Markov, http://vlsicad.eecs.umich.edu/BK/Slots/slots/BlockPacking.html.
59. S.T. Obenaus, T.H. Szymanski: *Gravity: Fast Placement for 3-D VLSI*, ACM Transactions on Design Automation of Electronic Systems, 2003, vol. 8, pp. 298–315.
60. M.G. Wrighton, A.M. DeHon: *Hardware-Assisted Simulated Annealing with Application for Fast FPGA Placement*, Proc. International Symposium on FPGA, 2003, pp. 33–42, http://www.sigda.org/Archives/ProceedingArchives/Compendiums/papers/fpga/confsym.htm.
61. K. Wiatr: *Akceleracja obliczeń w systemach wizyjnych*, Warszawa, WNT, 2003
62. O. Liu, M. Marek-Sadowska: *A Study of Netlist Structure and Placement Efficiency*, IEEE Transactions on Computer-Aided Design, 2005, vol. 24, pp. 762–772.
63. S.N. Adya, M.C. Yildiz, I.L. Markov, P.G. Villarrubia, P.N. Parakh, P.H. Madden: *Benchmarking for Large-Scale Placement and Beyond*, IEEE Transactions on Computer-Aided Design, 2004, vol. 23, pp. 472–486.

64. VLSI CAD Bookshelf 2, A.B. K a h n g, I.L. M a r k o v, http://vlsicad.eecs.umich.edu/BK/.
65. Strona internetowa firmy Synopsys, http://www.synopsys.com.
66. Strona internetowa firmy Cadence, http://www.cadence.com.
67. Cadence Design Systems: Envisia Silicon Ensemble Place-and-Route Reference, 2000.
68. Strona internetowa firmy Mentor Graphics, http://www.mentor.com.
69. Strona internetowa firmy Xilinx, http://www.xilinx.com.
70. Strona internetowa firmy Altera, http://www.altera.com.

Z. NAGÓRNY, A. KOS

THE DESIGN OF THE VLSI CIRCUIT LAYOUT PART 4. SIMULATED ANNEALING, NEURAL NETWORKS

S u m m a r y

The design process of the VLSI circuits requires the use of computer aided design tools. This paper is the fourth part of the survey of the cell placement techniques for digital VLSI circuits. In this part of the survey, the simulated annealing algorithm and neural networks are presented. An application of the simulated annealing algorithm to the cell placement problem is described. Nowadays the tools used for the cell placement, which utilize the presented algorithms are characterized: TimberWolfSC, TimberWolfMC, MGP, MPG-MS, VPR. Then, applications of neural networks to the cell placement problem are described. A self-organizing network and Hopfield network for the cell placement problem are presented. Some circuit layouts generated by using the Hopfield network are presented. Applications of a genetic algorithm, evolutionary strategy, three-stage placement-floorplanning-placement flow and special purpose hardware for the cell placement are described. Tools used for the 3D VLSI cell placement are characterized. Some conclusions concerning described techniques and tools are presented.

Keywords: VLSI, physical design process, integrated circuit, Standard Cell, Gate Array, Sea-of-Gates, Field Programmable Gate Array, floorplanning, cell placement, routing, pairwise interchange algorithm, simulated annealing algorithm, cooling schedule, multilevel optimization, neural networks, self-organizing network, Kohonen algorithm, Hopfield network, genetic algorithm, evolutionary strategy, reconfigurable system, TimberWolf, MGP, MPG-MS, VPR, Parquet, Gravity

Re
sze
me
się
mil
Art
lub
roz
Ele
Obj

Wy
Art
w f
w w
pow
Ukł
– T
– A
– M
– Z
– T

– U

Obsz

Stres
wzgl
w po
– W

Spos
Tekst
podk
(np.
pomo
strzel

INFORMACJE DLA AUTORÓW

Redakcja przyjmuje do publikowania prace oryginalne, przeglądowe i monograficzne wchodzące w zakres szeroko pojętej elektroniki. Ponieważ KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI jest czasopismem Komitetu Elektroniki i Telekomunikacji Polskiej Akademii Nauk, w związku z tym na jego łamach znajdują się prace naukowe dotyczące podstaw teoretycznych i zastosowań z zakresu elektroniki, telekomunikacji, mikroelektroniki, optoelektroniki, radiotechniki i elektroniki medycznej.
Artykuły powinno charakteryzować oryginalne ujęcie zagadnienia, własna klasyfikacja, krytyczna ocena (teorii lub metod), omówienie aktualnego stanu, lub postępu danej gałęzi techniki oraz omówienie perspektyw rozwojowych. Artykuły publikowane w innych czasopismach nie mogą być kierowane do druku w Kwartalniku Elektroniki i Telekomunikacji w drugiej kolejności zgłoszenia.
Objętość artykułu nie powinna przekraczać 30 stron po około 1800 znaków na stronie, w tym rysunki i tabele.

Wymagania podstawowe.
Artykuły należy nadsyłać na wyraźnym, jednostronnym, czarno-białym wydruku komputerowym. Wydruk w formacie A4 powinien mieć znormalizowaną liczbę wierszy i znaków w wierszu (30 wierszy po 60 znaków w wierszu), w dwóch egzemplarzach, w języku polskim lub angielskim wybranym przez autora. Do wydruku powinna być dołączona dyskietka z elektronicznym tekstem artykułu. Preferowane edytory to WORD 6 lub 8.
Układ artykułu (w wersji podstawowej) musi być następujący:
- Tytuł.
- Autor (imię i nazwisko autora/ów).
- Miejsce pracy (nazwa instytucji, miejscowość, adres. + ew. adres elektroniczny (e-mail).
- Zwięzłe streszczenie powinno być w języku takim, w jakim jest pisany artykuł (wraz ze słowami kluczowymi).
- Tekst podstawowy powinien mieć następujący układ:
  1. WPROWADZENIE
  2. np. TEORIA
  3. np. WYNIKI NUMERYCZNE
  3.1. .....
  3.2. .....
  4. .....
  5. .....
  6. PODSUMOWANIE
  7. ew. PODZIĘKOWANIA
  8. BIBLIIOGRAFIA
- Układ streszczenia w dodatkowej wersji językowej powinien być następujący:
  AUTOR (inicjał imienia i nazwisko).
  TYTUŁ (w języku angielskim – o ile artykuł pisany jest w języku polskim i na odrwót).
Obszerne do 3600 znaków streszczenia (wraz z słowami kluczowymi) w języku:
  a. angielskim, gdy artykuł pisany jest w języku polskim.
  b. polskim, gdy artykuł pisany jest w języku angielskim.
Streszczenie to powinno pozwolić czytelnikowi na uzyskanie istotnych informacji zawartych w pracy. Z tego względu w streszczeniu tym mogą być cytowane numery istotnych wzorów, rysunków i tabel zawartych w podstawowej wersji językowej.
- Wszystkie strony muszą mieć numerację ciągłą.

Sposób pisania tekstu.
Tekst powinien być pisany bez używania wyróżnień, a w szczególności nie dopuszcza się spacjowania, podkreślania i pisania tekstu dużymi literami z wyjątkiem wyrazów, które umownie pisze się dużymi literami (np. FORTRAN). Proponowane wyróżnienia Autor może zaznaczyć w maszynopisie zwykłym ołówkiem za pomocą przyjętych znaków adjustacyjnych, np. podkreślenie linią przerywaną oznacza spacjowanie (rozstrzelenie), podkreślenie linią ciągłą – pogrubienie, podkreślenie wężykiem — kursywa.

Tekst powinien być napisany z podwójnym odstępem między wierszami, tytuły i podtytuły małymi literami. Marginesy z każdej strony powinny mieć około 35 mm. Wielkość czcionki wydruku powinna być zbliżona co najmniej co wielkości czcionki maszyny do pisania (minimum 12 punktów). Przy podziale pracy na rozdziały i podrozdziały cyfrowe ich oznaczenia nie powinny być większe niż II stopnia (np. 4.1.1.).

Sposób pisania tabel.
Tabele powinny być pisane na oddzielnych stronach. Tytuły rubryk pionowych i poziomych powinny być napisane małymi literami z podwójnym odstępem między wierszami. Przypisy (notki) dotyczące tabel należy pisać bezpośrednio pod tabelami. Tabele należy numerować kolejno liczbami arabskimi, u góry każdej tabeli podać tytuł dwujęzyczny. W pierwszej kolejności w podstawowej wersji językowej, a później w dodatkowej wersji językowej. Tabele umieścić na końcu maszynopisu. Przyjmowane są tabele algorytmów i programy na wydrukach komputerowych. W tym przypadku zachowany jest ich oryginalny układ. Tabele powinny być cytowane w tekście.

Sposób pisania wzorów matematycznych.
Rozmieszczenie znaków, cyfr, liter i odstępów powinno być zbliżone do rozmieszczenia elementów druku. Wskaźniki i wykładniki potęg powinny być napisane wyraźnie i być prawidłowo obniżone lub podwyższone w stosunku do linii wiersza podstawowego. Znaki nad literami i cyframi, całkami i in. symbolami (strzałki, linie, kropki, daszki) powinny być umieszczone dokładnie nad tymi elementami, do których się odnoszą. Numery wzorów cyframi arabskimi powinny być kolejne i umieszczone w nawiasach okrągłych z prawej strony. Nazwy jednostek, symbole literowe i graficzne powinny być zgodne z wytycznymi IEE (International Electronical Commision) oraz ISO (International Organization of Standarization).

Powołania.
Powołania na publikacje powinny być umieszczone na ostatnich stronach tekstu pod tytułem „Bibliografia", opatrzone numeracją kolejną bez nawiasów. Numeracja ta powinna być zgodna z odnośnikami w tekście artykułu. Przykłady opisu publikacji:

- periodycznej — 1. F. Valdoni: A new milimetre wave satelite. E.T.T. 1990, vol. 2, no 5, pp. 141–148
- nieperiodyczne — 2. K. Andersen: A resource allocation framework. XVI International Symposium, Stockholm (Sweden), may 1991, paper A 2,4
- książki — 3. Y.P. Tvidis: Operation and modeling of the MOS transistors. New York, McGraw-Hill, 1987, p. 553

Materiały ilustracyjne.
Rysunki powinny być wykonane wyraźnie, na papierze gładkim lub milimetrowym w formacie nie mniejszym niż 9×12 cm. Mogą być także w postaci wydruku komputerowego (preferowany edytor Corel Draw). Fotografie lub diapozytywy przyjmowane są raczej czarno-białe w formacie nie przekraczającym 10×15 cm. Na marginesie każdego rysunku i na odwrocie fotografii powinno być napisane ołówkiem imię i nazwisko Autora oraz skrót tytułu artykułu, do którego są przeznaczone oraz numer rysunku. Spis podpisów pod rysunki i fotografie powinny być umieszczone na oddzielnej stronie. Podpisy pod rysunkami (fotografiami) powinny być dwujęzyczne: w pierwszej kolejności w podstawowej wersji językowej, a później w dodatkowej wersji językowej. Rysunki powinny być cytowane w tekście.

Uwagi końcowe.
Na odrębnej stronie powinny być podane następujące informacje:
- adres do korespondencji z kodem pocztowym (domowy lub do miejsca pracy),
- telefon domowy i/lub do miejsca pracy,
- adres e-mailowy (jeśli autor posiada).

Autorowi przysługuje bezpłatnie 20 odbitek artykułu. Dodatkowe egzemplarze odbitek, lub cały zeszyt Autor może zamówić u wydawcy na własny koszt.

Autora obowiązują korekta autorska, którą powinien wykonać w ciągu 3 dni od daty otrzymania tekstu z Redakcji oraz zwrócić osobiście lub listownie pod adres Redakcji. Korekta powinna być naniesiona na przekazanych Autorowi szpaltach na marginesach ew. na osobnym arkuszu w przypadku uzupełnień tekstu większych niż dwa wiersze. W przypadku nie zwrócenia korekty w terminie, korektę przeprowadza Redakcja Techniczna Wydawcy. Redakcja prosi Autorów o powiadomienie ją o zmianie miejsca pracy i adresu prywatnego.

INFORMATION FOR AUTHORS OF K.E.T.

The editorial stuff will accept for publishing only original monographic and survey papers concerning widely understood electronics. Because of the fact that KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI is a journal of the Committee for Electronics and Telecommunications of Polish Academy of Scince, it presents scientific works concerning theoretical bases and applications from the field of electronics, telecommunications, microelectronics, optoelectronics, radioelectronics and medical electronics.
Articles should be characterised by original depiction of a problem, its own classification, critical opinion (concerning theories or methods), discussion of an actual state or a progress of a given branch of a technique and discussion of development perspectives.
An article published in other magazines can not be submitted for publishing in K.E.T. TThe size of an article can not exceed 30 pages, 1800 character each, including figures and tables.

Basic requirements

The article should be submitted to the editorial stuff as a one side, clear, black and white computer printout in two copies. The article should be prepared in English or Polish. Floppy disc with an electronic version of the article should be enclosed. Preferred wordprocessors: WORD 6 or 8.
Layout of the article.

- Title.
- Author (first name and surname of author/authors).
- Workplace (institution, adress and e-mail).
- Concise summary in a language article is prepared in (with keywords).
- Main text with following layout:
  - Introduction
  - Theory (if applicable)
  - Numerical results (if applicable)
  - Paragraph 1
  - Paragraph 2
  - .....
  - .....
  - Conclusions
  - Acknowledgements (if applicable)
  - References
- Summary in additional language:
  - Author (firs name initials and surname)
  - Title (in Polish, if article was prepared in English and vice versa)
  - Extenstive summary, hawever not exceeching 3600 characters (along with keywords) in Polish, if artide wos prepared in English and vice versa). The summary should be prepared in a way allowing a reader to obtaoin essential information contained in the artide. For that reason in the summary author can place numbers of essential formulas, figures and tables from the article.

Pages should have continues numbering.

Main text

Main text can not contain formatting such as spacing, underlining, words written in capital letters (except words that are commonly written in capital letters). Author can mark suggested formatting with pencil on the margin of the article using commonly accepted adjusting marks.
Text should be written with double line spacing with 35 mro left and right margin. Titles and subtitles should be written with small letters. Titles and subtitles should be numberd using no mor than 3 levels (i.e. 4.1.1.).

Tables

Tables with their titles should be places on separate page at the end of the article. Titles of rows and columns should be written in small letters with double line spacing. Annotations concerning tables

should be placed directly below the table. Tables should be numbered with Arabic numbers on the top of each table. Table can consist algorithm and program listings. In such cases original layout of the table will be preserved. Table should be cited in the text.

Mathematical formulas

Characters, numbers, letters and spacing of the formula should be adequate to layout of main text. Indexes should be properly lowered or raised above the basic line and clearly written. Special characters such as lines, arrows, dots should be placed exactly over symbols which they ar attributed to. Formulas should be numbered with Arabic numbers place in brackets on the right side of the page. Units of measure, letter and graphic symbols should be printed according to requirements of IEE (International Electronical Commission) and ISO (International Organisation of Standardisation).

References

References should be placed at the and of the main text with th subtitle „References". Refrences should be numbered (without brackets) adequately to references placed in the text. Examples of periodical [@], non-periodical [2] and book [3] references:

1. F. Valdoni: A new milimetre wave satelite. E.T.T. 1990, vol. 2, no 5, pp. 141–148
2. K. Anderson: A resource allocation framework. XVI International Symposium (Sweden). May 1991. paper A 2.4
3. Y.P. Tvidis: Operation and modeling of the MOS transistors. Nw York. McGraw-Hill. 1987. p. 553

Figures

Figures should be clearly drawn on plain or milimetre paper in the format not smaller than 9×12 cm. Figures can be also printed (preferred editor – CorelDRAW). Photos or diapositives will be accepted in black and whit format not gratd than 10×15 cm. On th margin of each drawing and on th back side of each ohoto authorp name and abbreviation of the title of article should be placed. Figure's captures should be given in two longuaqes (first in the language the article is writtes in and then in additional language). Figure's captures should be also listed on separate page. Figures should be cited in the text.

Additional information

On the separate page following information should be placed:

- mailing address (home or office),
- phone (home or/and office),
- e-mail.

Author is entiled to free of charg 20 copies of articl. Additional copies or the whole magazine can be ordered at publisher at th one's expense.

Author is obliges to perform the author's correction, which should be accomplished within 3 days starting from the date of receiving of the text from the ditorial stuff. Corrected text should be return to the editorial stuff personally or by mail. Correction marks should be placed on th margin of copies received from the editorial stuff or if needed on separate pages. In the case when the correction is not returned in said time limit, correction will be performed by technical editorial stuff of the publisher.

In case of changing of workplace or home address Authors are asked to inform the editorial stuff.

LISTA RECENZENTÓW KWARTALNIKA ELEKTRONIKI I TELEKOMUNIKACJI

Dr inż. ***Adam Abramowicz***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Jarosław Arabas***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Jerzy Baranowski***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Andrzej Bąk***
Politechnika Warszawska
Instytut Telekomunikacji

Dr hab. inż. ***Zbigniew Bielecki***
Wojskowa Akademia Techniczna
Warszawa

Dr inż. ***Piotr Bora***
Wojskowa Akademia Techniczna
Warszawa

Dr inż. ***Michał Borecki***
Politechnika Warszawska
Instytut Mikro i Optoelektroniki

Dr inż. ***Tomasz Buczkowski***
Politechnika Warszawska
Instytut Radioelektroniki

Dr hab. inż. ***Bohdan Butkiewicz***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr hab. inż. ***Grzegorz Ciesielski***
Politechnika Łódzka
Instytut Elektrotechniki Teoretycznej,
Metrologii i Materiałoznawstwa

Dr hab. inż. ***Witold Czarnecki***
Politechnika Warszawska
Instytut Telekomunikacji

Dr inż. ***Krzysztof Czerwiński***
Politechnika Warszawska
Instytut Radioelektroniki

Prof. nzw. dr hab. ***Andrzej Dąbrowski***
Politechnika Warszawska
Instytut Telekomunikacji

Dr inż. ***Henryk Dobrowolski***
Politechnika Warszawska
Instytut Informatyki

Prof. dr hab. inż. ***Janusz A. Dobrowolski***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Jarosław Domaszewicz***
Politechnika Warszawska
Instytut Telekomunikacji

Dr inż. ***Przemysław Dymarski***
Politechnika Warszawska
Instytut Telekomunikacji

Prof. dr hab. inż. ***Jerzy Frączek***
Politechnika Śląska
Instytut Automatyki

Dr hab. inż. ***Piotr Gajewski***
Wojskowa Akademia Techniczna
Instytut Telekomunikacji

Dr inż. ***Zbigniew Gajo***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Bogdan Galwas***
Politechnika Warszawska
Instytut Mikroelektroniki i Optoelektroniki

Mgr inż. ***Marcin Golański***
Politechnika Warszawska
Instytut Telekomunikacji

Dr hab. inż. ***Konrad Hejn***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Ewa Hermanowicz***
Politechnika Gdańska
Wydział Elektroniki i Telekomunikacji
i Informatyki

Prof. dr hab. inż. ***Ryszard Jachowicz***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof nzw. dr hab. ***Andrzej Jakubiak***
Politechnika Warszawska
Instytut Telekomunikacji

Prof. dr hab. inż. ***Andrzej Jakubowski***
Politechnika Warszawska
Instytut Mikroelektroniki i Optoelektroniki

Dr hab. inż. ***Stanisław Jankowski***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Swietlana Januszkiewicz***
Politechnika Szczecińska
Wydział Informatyki

Dr inż. ***Jacek Jarkowski***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Krzysztof Jasiński***
Politechnika Warszawska
Instytut Telekomunikacji

Prof. dr hab. inż. ***Janusz Jaworski***
Politechnika Lubelska
Katedra Automatyki

Prof. dr hab. inż. ***Wojciech Kabaciński***
Politechnika Poznańska
Instytut Elektroniki i Telekomunikacji

Prof. dr hab. inż. ***Zbigniew Kaczkowski***
Polska Akademia Nauk, Warszawa
Instytut Fizyki

Mgr inż. ***Andrzej Kalinowski***
Politechnika Warszawska
Instytut Telekomunikacji

Dr inż. ***Wojciech Kazubski***
Politechnika Warszawska
Instytut Radioelektroniki

Dr inż. ***Ryszard Kisiel***
Politechnika Warszawska
Instytut Mikroelektroniki i Optoelektroniki

Prof. dr hab. inż. ***Władysław Kołosowski***
Wojskowa Akademia Techniczna
Instytut Systemów Telekomunikacji, Warszawa

Dr inż. ***Tomasz Kosiło***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr ***Ryszard Kossowski***
Politechnika Warszawska
Instytut Telekomunikacji

Dr inż. ***Zbigniew Kowalczyk***
Przemysłowy Instytut Telekomunikacji
Gdańsk-Wrzeszcz

Mgr inż. ***Henryk Kowalski***
Politechnika Warszawska
Instytut Informatyki

Prof. dr hab. inż. ***Andrzej Kraśniewski***
Politechnika Warszawska
Instytut Telekomunikacji

Prof. dr hab. inż. ***Jacek Kudrewicz***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Małgorzata Kujawińska***
Politechnika Warszawska
Instytut Mikromechaniki i Fotoniki

Dr inż. ***Sławomir Kukliński***
Politechnika Warszawska
Instytut Telekomunikacji

Dr inż. ***Sławomir Kula***
Politechnika Warszawska
Instytut Telekomunikacji

Prof. dr hab. inż. ***Juliusz Lech Kulikowski***
Polska Akademia Nauk, Warszawa
Instytut Biocybernetyki i Inżynierii Biomedycznej

Prof. nadzw. dr hab. ***Zbigniew Kulka***
Politechnika Warszawska
Instytut Radioelektroniki

Dr inż. ***Krzysztof Kulpa***
Politechnika Warszawska
Instytut Radioelektroniki

Prof. dr hab. inż. ***Zygmunt Kuśmierek***
Politechnika Łódzka
Instytut Elektroniki Teoretycznej
Meteorologii i Materiałoznawstwa

Dr hab. inż. ***Wiesław Kuźmicz***
Politechnika Warszawska
Instytut Mikroelektroniki
i Optolelektroniki

Dr inż. ***Lech Lewandowski***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Tadeusz Łuba***
Politechnika Warszawska
Instytut Telekomunikacji

Mgr inż. ***Grzegorz Mazur***
Politechnika Warszawska
Instytut Informatyki

Prof. dr hab. inż. ***Marian Miłek***
Politechnika Zielonogórska
Instytut Metrologii Elektrycznej

Prof. dr hab. inż. ***Józef Modelski***
Politechnika Warszawska
Instytut Radioelektroniki

Prof. dr hab. inż. ***Tadeusz Morawski***
Politechnika Warszawska
Instytut Radioelektroniki

Prof. dr hab. inż. ***Jan Mulawka***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Andrzej Napieralski***
Politechnika Łódzka
Katedra Mikroelektroniki
i Technik Informatycznych

Dr inż. ***Leszek Opalski***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Andrzej Pacut***
Politechnika Warszawska
Instytut Automatyki

Prof. dr ***Wincenty Pajewski***
Polska Akademia Nauk, Warszawa
Instytut Podstawowych Problemów Techniki

Dr hab. inż. ***Zdzisław Papir***
Akademia Górniczo-Hutnicza,
Kraków Katedra Telekomunikacji

Dr inż. ***Andrzej Paszkiewicz***
Politechnika Warszawska
Instytut Telekomunikacji

Dr hab. inż. ***Jan Petykiewicz***
Politechnika Warszawska
Instytut Fizyki

Prof. dr hab. inż. ***Andrzej Pfitzner***
Politechnika Warszawska
Instytut Mikroelektroniki i Optoelektroniki

Prof. dr hab. inż. ***Stanisław J. Piestrach***
Politechnika Wrocławska
Instytut Cybernetyki Technicznej

Prof. dr hab. inż. ***Marian Piekarski***
Politechnika Wrocławska
Instytut Telekomunikacji i Akustyki

Prof. dr hab. inż. ***Anna Piotrowska***
Instytut Technologii Elektronicznej
Warszawa

Prof. dr hab. inż. ***Kazimierz J. Pluciński***
Wojskowa Akademia Techniczna
Instytut Podstaw Elektroniki, Warszawa

Dr inż. ***Sławomir Pluta***
Politechnika Opolska
Wydział Elektrotechniki, Automatyki
i Informatyki

Prof. dr hab. ***Włodzimierz Pogribny***
Akademia Techniczno-Rolnicza
Instytut Telekomunikacji, Bydgoszcz

Dr inż. ***Artur Przelaskowski***
Politechnika Warszawska
Instytut Radioelektroniki

Dr inż. ***Mariusz Rawski***
Politechnika Warszawska
Instytut Telekomunikacji

Dr inż. ***Mirosław Rojewski***
Politechnika Gdańska
Katedra Systemów Informacyjnych

Prof. nadzw. dr hab. ***Stanisław Rosłaniec***
Politechnika Warszawska
Instytut Telekomunikacji

Dr hab. ***Krzysztof Sacha***
Politechnika Warszawska
Instytut Automatyki i Informatyki Stosowanej

Prof. dr hab. inż. ***Jerzy Siuzdak***
Politechnika Warszawska
Instytut Telekomunikacji

Prof. nadzw. dr hab. ***Władysław Skarbek***
Politechnika Warszawska
Instytut Radioelektroniki

Prof. nadzw. dr hab. ***Andrzej Skorupski***
Politechnika Warszawska
Instytut Informatyki

Mgr inż. ***Krzysztof Skowroński***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. zw. dr hab. ***Wojciech Sobczak***
Politechnika Gdańska
Katedra Systemów Informacyjnych

Prof. dr hab. inż. ***Waldemar Soluch***
Instytut Technologii Materiałów
Elektronicznych, Warszawa

Prof. dr hab. inż. ***Marek Stabrowski***
Politechnika Warszawska
Instytut Elektrotechniki Teoretycznej
i Miernictwa Elektrycznego

Dr hab. inż. ***Witold J. Stepowicz***
Akademia Morska
Katedra Elektroniki Morskiej
Gdynia

Prof. dr hab. inż. ***Jerzy Szabatin***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Paweł Szczepański***
Politechnika Warszawska
Instytut Mikroelektroniki i Optoelektroniki

Dr inż. ***Feliks Szczot***
Politechnika Opolska
Wydział Elektrotechniki, Automatyki
i Informatyki

Dr hab. ***Jan Szmidt***
Politechnika Warszawska
Instytut Mikroelektroniki i Optoelektroniki

Prof. dr hab. inż. ***Mieczysław Szustakowski***
Wojskowa Akademia Nauk
Instytut Optoelektroniki, Warszawa

Dr inż. ***Jacek Szymanowski***
Politechnika Warszawska
Instytut Automatyki i Informatyki
Stosowanej

Dr inż. ***Lech Śliwa***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Marek Średniawa***
Politechnika Warszawska
Instytut Telekomunikacji

Prof. dr hab. inż. ***Leszek Trybus***
Politechnika Rzeszowska
Katedra Automatyki i Informatyki

Prof. dr hab. inż. ***Aleksander Urbaś***
Politechnika Wrocławska
Instytut Telekomunikacji i Akustyki
Elektronicznych

Prof. dr hab. inż. ***Krzysztof W. Wawryn***
Politechnika Koszalińska
Katedra Teorii Obwodów
i Układów

Mgr inż. ***Krzysztof Włostowski***
Politechnika Warszawska
Instytut Telekomunikacji

Dr ***Piotr Witoński***
Politechnika Warszawska
Instytut Mikroelektroniki i Optoelektroniki

Prof. dr hab. inż. ***Jacek Wojciechowski***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. nzw. dr hab. inż. ***Andrzej Wojtkiewicz***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Dr inż. ***Adam W. Woźniak***
Politechnika Warszawska
Instytut Automatyki

Prof. dr hab. inż. ***Waldemar Wójcik***
Politechnika Lubelska
Instytut Technologii Elektronicznej
Katedra Elektroniki

Prof. dr hab. inż. ***Józef Woźniak***
Politechnika Gdańska
Katedra Systemów Informacyjnych

Dr hab. inż. ***Zygmunt Wróbel***
Uniwersytet Śląski
Instytut Problemów Techniki

Dr inż. ***Wojciech Zabłotny***
Politechnika Warszawska
Instytut Systemów Elektronicznych

Prof. dr hab. inż. ***Janusz Zarębski***
Akademia Morska
Katedra Radioelektroniki Morskiej, Gdynia

KUPON ZAMÓWIENIA PRENUMERATY

numer pisma l.dz.: ..... data: .....

Zamawiam prenumeratę (tytuł, nr rozpoczynający) .....

.....

Ilość egz. ..... Cena jedn. ..... Wartość zamówienia .....

Okres zamówienia: ☐ stałe* ..... ☐ okresowe** od ..... do .....

Przedpłata: ☐ tak, chcę dokonać przedpłaty na prenumeratę

☐ nie, nie chcę dokonać przedpłaty na prenumeratę

ZAMAWIAJĄCY

(proszę wypełnić drukowanymi literami)

Odbiorca (dane do wysyłki):

Imię .....
Nazwisko .....
Nazwa firmy .....
.....
Adres .....
.....
Tel. ..... fax:.....
e-mail .....

Płatnik (dane do faktury):

.....
.....
.....
.....
.....
.....
NIP: .....

☐ Firma jest płatnikiem VAT. Upoważniam DHN do wystawienia faktury VAT bez podpisu odbiorcy

☐ Nie jestem płatnikiem VAT. Upoważniam DHN do wystawienia faktury VAT bez podpisu odbiorcy

☐ Wyrażam zgodę na wykorzystanie moich danych przez DHN w celach marketingowych

---

* obejmuje prenumeratę publikacji począwszy od podanego numeru aż do odwołania
** obejmuje prenumeratę publikacji, które ukażą się w podanym okresie

.....
Podpis zamawiającego

Dom Handlowy Nauki Sp. z o.o.
Naukowa Księgarnia Wysyłkowa
ul. Szczęśliwicka 2/17, 02-352 Warszawa

tel. (+48 22) 658-15-58, fax (+48 22) 822-98-69
e-mail: ksiegarnia@dhn.pl lub czasopismakraj@dhn.pl
Nr Konta: Bank PEKAO S.A. IV O/W-wa Nr 50124010531111000004407829