

Wydział Elektroniki i Telekomunikacji

Politechnika Poznańska

ul. Piotrowo 3A, 60-965 Poznań

Autoreferat rozprawy doktorskiej

**Sterowanie wielopłaszczyznowymi
polami typu banyan**

Autor:

mgr inż. Marek Michalski

Promotor:

prof. dr hab. inż. Wojciech Kabaciński

Poznań 2011

Spis treści

1	Wstęp	2
1.1	Wprowadzenie	2
1.2	Teza, cel i zakres pracy	3
2	Pola komutacyjne rozważane w rozprawie	4
2.1	Architektura pola	4
2.2	Reprezentacja grafowa	5
2.3	Własności kombinatoryczne	5
3	Reprezentacja stanu pola	7
3.1	Macierz stanu płaszczyzny p – macierz A_p	7
3.2	Macierz stanu pola – macierz E	8
3.3	Macierz blokowanych relacji w płaszczyźnie p – macierz B_p	10
3.4	Macierz kosztu realizacji połączenia – macierz C	12
3.5	Macierz wyboru płaszczyzny – macierz D	13
3.6	Graf stanu pola	14
4	Algorytmy sterowania zaproponowane w pracy	16
4.1	Macierzowy algorytm sterowania polem	16
4.2	Kolejnościowy algorytm zestawiania połączeń	18
4.3	Algorytmy przestrojeń w polu o parzystej liczbie sekcji	18
4.4	Algorytmy przestrojeń w polu o nieparzystej liczbie sekcji	19
5	Ocena zaproponowanych algorytmów	22
6	Implementacja algorytmów	24
6.1	Implementacja algorytmu macierzowego w układzie FPGA	24
6.2	Implementacja pozostałych algorytmów	26
7	Najważniejsze osiągnięcia pracy	27
8	Publikacje	29

1

Wstęp

1.1. Wprowadzenie

Funkcjonowanie współczesnego świata jest uwarunkowane istnieniem i sprawnym działaniem sieci telekomunikacyjnych. Każda taka sieć składa się z ogromnej liczby węzłów połączonych między sobą oraz dołączonych do niej użytkowników. Węzłami takiej sieci są rozbudowane centra operatorów telekomunikacyjnych składające się z wielu urządzeń sieciowych, a także stosunkowo proste pojedyncze urządzenia poszczególnych klientów. Urządzenia sieciowe są ze sobą połączone odpowiednimi łączami i mogą realizować między sobą transmisję danych. Sieci telekomunikacyjne działają w oparciu o różne technologie i różne protokoły komunikacyjne. Jednak w każdym przypadku istnieje warstwa fizyczna, która jest odpowiedzialna za realne przenoszenie informacji do sąsiednich węzłów. Konieczne są również odpowiednie mechanizmy odpowiedzialne za odbieranie informacji z interfejsów wejściowych urządzeń sieciowych, decydowanie o przeznaczeniu poszczególnych informacji oraz za ich przekazanie do odpowiednich punktów, którymi mogą być między innymi interfejsy wyjściowe urządzeń sieciowych. Bardzo ważnym elementem każdego urządzenia sieciowego, które przesyła dane pomiędzy swymi interfejsami sieciowymi, jest matryca przełączająca, zwana też polem komutacyjnym. Ten element jest odpowiedzialny za fizyczne przenoszenie transportowanej informacji z interfejsów wejściowych na odpowiednie interfejsy wyjściowe. Istnieje wiele technik przesyłania i komutacji informacji. Typowo wykorzystuje się elektryczną reprezentację przesyłanych sygnałów w medium transmisyjnym, a ogromną popularność zdobyła transmisja optyczna. Jednak typowe podejście do komutacji wymusza konwersję sygnałów wykorzystywanych w łączach optycznych do postaci elektrycznej, gdyż zdecydowana większość aktualnie używanych urządzeń sieciowych przetwarza informację w postaci elektrycznej. Nawet jeśli łącza pomiędzy węzłami są zrealizowane w technologii światłowodowej, na wejściu węzła sieciowego następuje konwersja opto-elektorniczna, następnie informacja jest przetwarzana w postaci elektronicznej, a na wyjściu następuje konwersja elektro-optyczna. Bardzo szerokie pasmo, jakie oferują współczesne optyczne media transmisyjne, jest w pełni wykorzystywane przede wszystkim w przełącznicach optycznych oraz urządzeniach dostępowych. W przypadku takich urządzeń rekonfiguracja elementów komutacyjnych występuje stosunkowo rzadko, a dopuszczalny czas realizacji zmiany stanu elementów fizycznych jest rzędu milisekund. W przypadku w pełni optycznej komutacji pakietów dopuszczalny czas przełączania jest dużo krótszy, wymagane są szybkie elementy przełączające, które mogą zmieniać swój stan w ciągu pojedynczych nanosekund. Ważnym ogra-

niczeniem w zestawianiu połączeń w polu, zwłaszcza przy optycznej komutacji pakietów, jest nie tylko czas przełączania elementów komutacyjnych, ale także możliwość szybkiego wyznaczenia dróg połączeniowych w polu. Właśnie ten aspekt jest przedmiotem rozprawy - algorytmy umożliwiające szybkie wyznaczenie drogi połączeniowej.

1.2. Teza, cel i zakres pracy

Rozprawa porusza tematykę związaną z algorytmami sterowania polami komutacyjnymi typu banyan. Pola takie po raz pierwszy zostały przedstawione w pracy „Banyan Networks for Partitioning Multiprocessor Systems” (Goke L. R., Lipowski G. J., 1st Annual Symposium on Computer Architecture, 1973). Tam też została zdefiniowana ich struktura oraz podstawowe algorytmy sterowania. Pola tego typu są szeroko wykorzystywane w wieloprocessorowych systemach komputerowych, w telekomunikacji, a w szczególności w sieciach optycznych. Celem pracy jest zaproponowanie nowej metody reprezentacji i analizy stanu pola typu banyan oraz nowych algorytmów sterowania na niej bazujących. W metodzie tej do reprezentacji stanu pola wykorzystywane są macierze, wartości elementów tych macierzy symbolizują połączenia w polu i ich właściwości. Proponowane algorytmy powinny być szybkie, umożliwiać sprzętową realizację oraz wyznaczać drogi połączeniowe w czasie rzędu nanosekund lub dziesiątek nanosekund. W pracy zaprezentowano algorytmy, które dotyczą trzech charakterystycznych sytuacji: obsługi pojedynczych połączeń, komutacji jednoczesnej oraz przestrojeń. Algorytmy zostały przedstawione w postaci teoretycznej oraz zoptymalizowanej pod kątem szybkiej realizacji w strukturach sprzętowych, przedstawiono również ich implementację w układach FPGA.

Teza pracy: *Możliwe jest opracowanie takiej metody analizy stanu pola komutacyjnego typu banyan, która umożliwi zaproponowanie efektywnych algorytmów sterowania połączeniami i ich implementację w szybkich strukturach sprzętowych.*

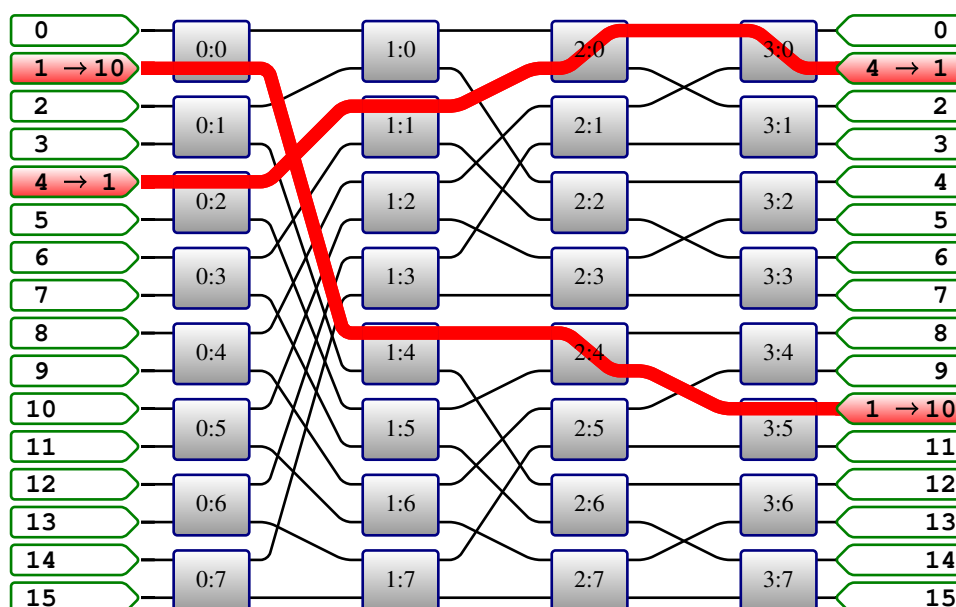
Powyższa teza zostanie udowodniona poprzez przedstawienie metody reprezentacji i analizy stanu pól oraz zaproponowanie trzech grup algorytmów ją wykorzystujących wraz z ich sprzętową implementacją.

Praca składa się z ośmiu rozdziałów. W rozdziale 2 opisano pola typu banyan, ich struktura oraz najważniejsze właściwości, w tym własności kombinatoryczne. W rozdziale 3 przedstawiono nowy sposób analizy tychże pól, na bazie której zaproponowano nowy algorytm sterowania. Algorytm ten wykorzystuje macierze do reprezentacji stanu pola, dokonuje obliczeń i podejmuje decyzję o przydziale poszczególnych zasobów pola do realizacji nowych połączeń. Rozdział 4 zawiera analizę wykorzystania zaproponowanego algorytmu do komutacji jednoczesnej. Algorytm ten zakłada kolejnościowy przydział zasobów dla kolejno analizowanych połączeń, może być potraktowany jako algorytm przestrojeń. W rozdziale 5 przeanalizowano stan pola pod kątem realizacji przestrojeń i zaproponowano nowy algorytm przestrojeń, który szczególnie dobrze sprawdza się przy sterowaniu pól o parzystej liczbie sekcji. Ponadto algorytm znajduje i realizuje minimalną liczbę przestrojeń konieczną do odblokowania trasy nowego połączenia. Rozdział 6 opisuje sprzętową implementację poszczególnych algorytmów w układach FPGA. Odpowiednia adaptacja mechanizmów algorytmów pozwoliła na skonstruowanie sterownika, który wyznacza drogę połączeniową w bardzo krótkim czasie. W rozdziale 7 przedstawiono symulacyjne porównanie algorytmów oraz wnioski jakie z niego płyną. Rozdział ostatni stanowi podsumowanie pracy.

Pola komutacyjne rozważane w rozprawie

2.1. Architektura pola

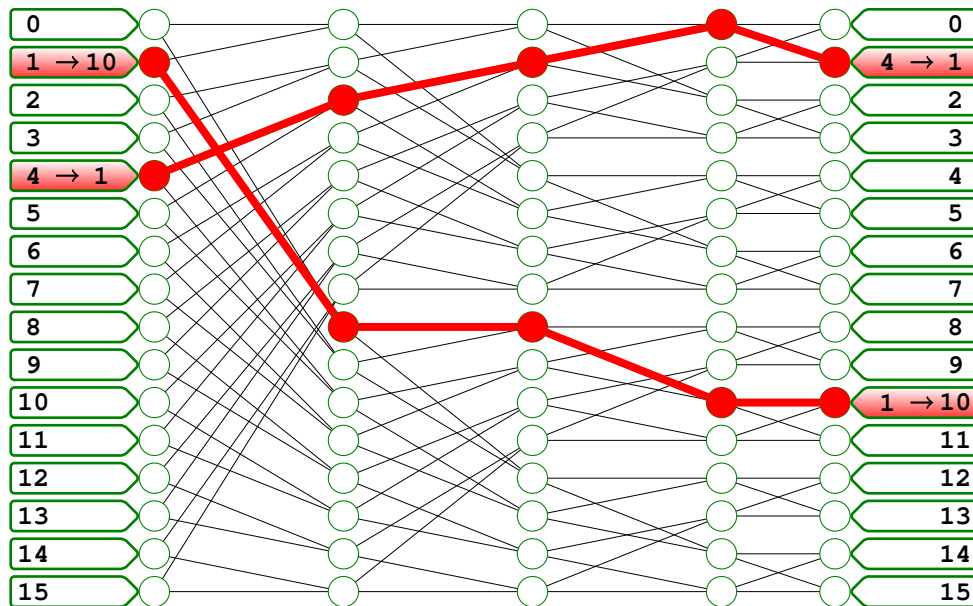
Pola typu banyan to pola wielosekcyjne. Podstawowym elementem, z którego są zbudowane takie pola to komutator kwadratowy o dwóch wejściach i dwóch wyjściach. W ogólnym przypadku pole posiada n sekcji takich komutatorów, połączone one są ze sobą w taki sposób, że całe pole posiada $N = 2^n$ wejść i wyjść. Na rysunku 2.1 przedstawiono pole o $N = 16$. Poszczególne komutatory zostały oznaczone dwoma wartościami; pierwsza z nich oznacza numer sekcji, druga - numer danego komutatora w obrębie danej sekcji. Połączenie z wejścia x do wyjścia y będzie oznaczane jako $\langle x, y \rangle$, na rysunku zaznaczono połączenia $\langle 1, 10 \rangle$ oraz $\langle 4, 1 \rangle$. W zależności od sposobu łączenia sąsiednich sekcji komutatorów można wyróżnić kilka topologicznie równoważnych konfiguracji, wszelkie właściwości, twierdzenia oraz wzory przedstawiane w pracy dotyczą konfiguracji baseline, jednak mogą zostać zaadaptowane do pól w innych konfiguracjach.



Rysunek 2.1. Pole komutacyjne w konfiguracji baseline z $N = 16$ i $n = 4$

2.2. Reprezentacja grafowa

Najbardziej intuicyjny sposób przedstawiania przestrzennych pól komutacyjnych wykorzystuje symbole komutatorów i łączy międzysekcyjnych. Innym sposobem reprezentacji pola jest graf dwudzielny pola. Każde wejście, wyjście oraz łączy międzysekcyjne jest reprezentowane przez węzeł grafu, natomiast krawędzie łączące poszczególne węzły reprezentują możliwość połączenia przez komutator łączy przyłączonego do jego wejścia (lub wejścia pola w przypadku komutatorów pierwszej sekcji) do jednego z dwóch łączy międzysekcyjnych przyłączonych do jego wyjść (lub wyjść pola w przypadku komutatora ostatniej sekcji). Na rysunku 2.2 przedstawiono graf dwudzielny pola z rysunku 2.1 wraz z tymi samymi połączeniami.



Rysunek 2.2. Graf dwudzielny pola o $N = 16$ z zestawionymi połączeniami $\langle 1, 10 \rangle$ oraz $\langle 4, 1 \rangle$

2.3. Własności kombinatoryczne

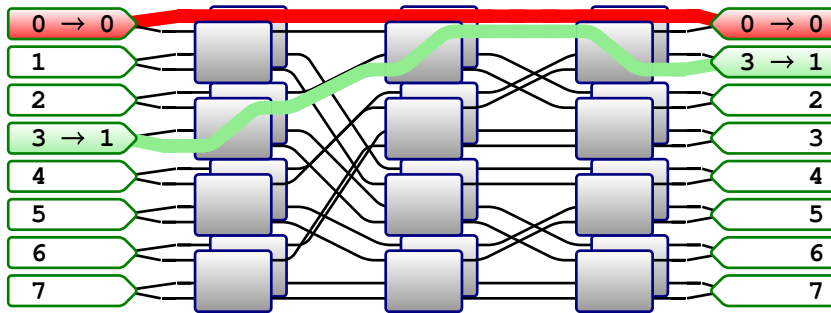
Z uwagi na powiązanie liczby sekcji z liczbą wejść pola ($n = \log_2 N$) pola typu banyan bywają nazywane polami typu $\log_2 N$. W polach tego typu istnieje dokładnie jedna trasa dla każdego połączenia $\langle x, y \rangle$. Na podstawie sposobu łączenia komutatorów sąsiednich sekcji można określić numery zasobów zajmowanych przez poszczególne połączenia $\langle x, y \rangle$. Numer zajmowanego węzła W w sekcji s przez połączenie $\langle x, y \rangle$ w polu n -sekcyjnym może być określony na podstawie wzoru:

$$W(x, y, s, n) = x \operatorname{div} 2^s + y - y \operatorname{mod} 2^{n-s}, \quad (2.1)$$

gdzie operacja div oznacza dzielenie bez reszty, operacja mod oznacza resztę z dzielenia. Podczas zestawiania połączenia $\langle x, y \rangle$ nie ma możliwości wyboru trasy dla tego połączenia, jest ona jednoznacznie określona przez numer wejścia i wyjścia. Może się okazać, że zasoby konieczne do realizacji nowego połączenia są już wykorzystywane do realizacji połączenia zestawionego wcześniej, czyli te dwa połączenia wzajemnie się blokują, gdyż ich trasy muszą być realizowane przy pomocy tych samych zasobów. Na podstawie wzoru:

$$l_w(\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle) = \sum_{s=1}^{n-1} ((x_1 \operatorname{div} 2^s + y_1 - y_1 \operatorname{mod} 2^{n-s}) \operatorname{xnor} (x_2 \operatorname{div} 2^s + y_2 - y_2 \operatorname{mod} 2^{n-s})) \quad (2.2)$$

można wyznaczyć liczbę wspólnych zasobów dla tras połączeń $\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle$. Określa on liczbę węzłów w grafie dwudzielnym pola, które są wspólne dla tras dwóch połączeń. Jeśli jest ona niezerowa, takie połączenia wzajemnie się blokują i nie mogą być jednocześnie realizowane w polu z rysunku 2.2. Do realizacji takich połączeń będą wykorzystywane wielopłaszczyznowe pola typu banyan, zwane również polami typu $multi - \log_2 N$. Zbudowane one są z p kopii struktury przedstawionej na rysunku 2.2, jedna taka struktura nazywana jest płaszczyzną pola. Połączenia, które wzajemnie się blokują, będą realizowane w różnych płaszczyznach. Sytuację taką przedstawia rysunek 2.3. Podczas zestawiania nowego połączenia jest wybierana dla niego taka płaszczyzna, w której dostępne są wszystkie zasoby konieczne do realizacji jego trasy. Jeśli nowe połączenie jest blokowane w każdej płaszczyźnie pola, czyli ani jedna płaszczyzna pola nie jest w stanie zrealizować nowego połączenia, jest ono odrzucane. Stan taki jest zwany stanem blokady danego połączenia.



Rysunek 2.3. Pole o $N = 8$ wejściach i $p = 2$ płaszczyznach

Liczba płaszczyzn jest bardzo ważnym parametrem wielopłaszczyznowych pól typu banyan, gdyż od niej zależy występowanie stanów blokady. Z uwagi na występowanie stanów blokady i sposoby ich omijania, pola możemy podzielić na kilka rodzajów (Beneš V. E., *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, Nowy Jork, 1965). Aby wielopłaszczyznowe pole typu banyan było polem nieblokowanym w wąskim sensie (*ang. strict sense nonblocking – SSNB*) musi składać się z p płaszczyzn, gdzie

$$p = \begin{cases} \frac{3}{2}2^{\frac{n}{2}} - 1, & \text{dla } n \text{ parzystych;} \\ 2^{\frac{n+1}{2}} - 1, & \text{dla } n \text{ nieparzystych.} \end{cases} \quad (2.3)$$

Aby pole typu $multi - \log_2 N$ było przestrajalne (*ang. Rearrangeably Nonblocking – RRNB*) musi składać się z przynajmniej p płaszczyzn, gdzie

$$p = 2^{\lceil n/2 \rceil}. \quad (2.4)$$

W polu, w którym liczba płaszczyzn jest mniejsza niż wynika to z warunków nieblokowności i przestrajalności, mogą występować sytuacje, gdy wykorzystywany algorytm sterowania nie jest w stanie wskazać płaszczyzny dla nowego połączenia i połączenie takie nie zostanie zrealizowane. Pola takie nazywane są polami blokowanymi.

3

Reprezentacja stanu pola

3.1. Macierz stanu płaszczyzny p – macierz A_p

Do reprezentacji stanu pola wykorzystano macierze. W nich jest przechowywana informacja o istniejących w polu połączeniach oraz wyniki pośrednie realizacji algorytmu sterowania. Połączenie $\langle x, y \rangle$ jest traktowane jako połączenie z komutatora wejściowego o numerze i , gdzie

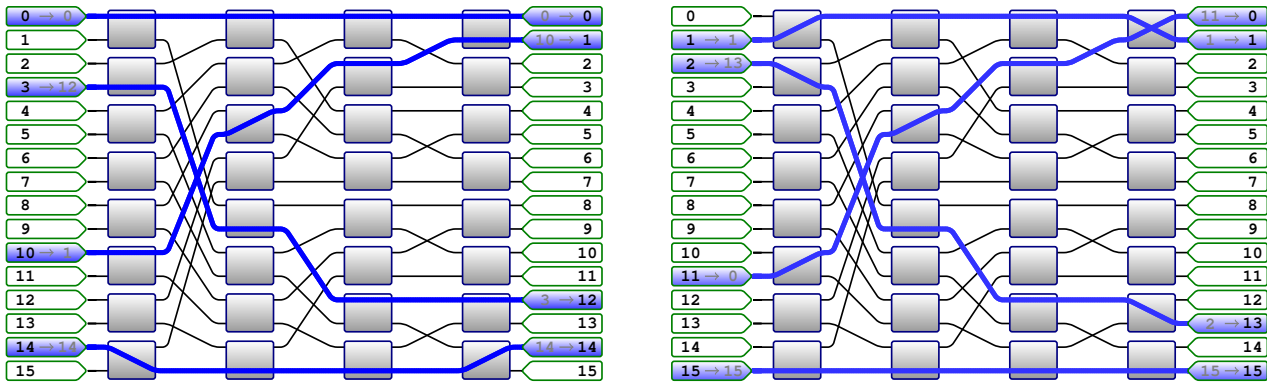
$$i = \lfloor \frac{x}{2} \rfloor, \quad (3.1)$$

do komutatora wyjściowego o numerze j , gdzie

$$j = \lfloor \frac{y}{2} \rfloor. \quad (3.2)$$

Takie połączenie jest nazywane połączeniem relacji (i, j) . W dalszych częściach pracy każde połączenie relacji (i, j) jest w skrócie określane jako połączenie (i, j) . Połączenia są analizowane z dokładnością do numeru komutatora wejściowego i wyjściowego. Taki sposób traktowania połączeń zmniejsza ilość danych, które są analizowane w kolejnych etapach. Na podstawie wzorów (3.1) i (3.2) można stwierdzić, że w obydwu polach przedstawionych na rysunku 3.1 są ukazane połączenia relacji $(0, 0)$, $(1, 6)$, $(5, 0)$, $(7, 7)$ mimo, że na jednym z rysunków są połączenia $\langle 0, 0 \rangle$, $\langle 3, 12 \rangle$, $\langle 10, 1 \rangle$ i $\langle 14, 14 \rangle$, a na drugim $\langle 1, 1 \rangle$, $\langle 2, 13 \rangle$, $\langle 11, 0 \rangle$ i $\langle 15, 15 \rangle$. Z punktu widzenia blokady istotna jest zajętość łączy międzysekcyjnych, stąd dwa stany na rysunku 3.1 można uważać za równoważne.

Każde połączenie jest realizowane przez konkretną płaszczyznę w polu komutacyjnym. Do reprezentacji wszystkich połączeń zestawionych w jednej płaszczyźnie pola jest użyta macierz kwadratowa nazywana macierzą A . Macierz A składa się z $N/2$ wierszy i $N/2$ kolumn, gdzie numer wiersza oznacza numer komutatora wejściowego, natomiast numer kolumny oznacza numer komutatora wyjściowego. Dla uproszczenia wzorów, wiersze i kolumny wszystkich wykorzystywanych macierzy są numerowane od wartości 0. Do reprezentacji połączeń w polu składającym się z p płaszczyzn wykorzystywanych jest p macierzy A , po jednej dla każdej z płaszczyzn. Wartości poszczególnych elementów $a_k[i, j]$ w macierzy A_k ($0 \leq k \leq p - 1$) jednoznacznie określa definicja 3.1.



Rysunek 3.1. Dwa różne zbiory połączeń, które są realizowane przez ten sam zbiór łączny międzysekcyjnych

Definicja 3.1:

$$\mathbf{A}_k = \left[a_k[i, j] = \begin{cases} 1, & \text{jeśli w płaszczyźnie } k \text{ jest zestawione połączenie} \\ & \text{z komutatora sekcji wejściowej o numerze } i \\ & \text{do komutatora sekcji wyjściowej o numerze } j \\ 0, & \text{w pozostałych przypadkach.} \end{cases} \right] \quad (3.3)$$

Macierze \mathbf{A}_k przechowują informacje o połączeniach poszczególnych relacji (i, j) . Każde połączenie zestawione przez płaszczyznę k ma swoją reprezentację przez odpowiedni element $a_k[i, j] = 1$.

Z uwagi na liczbę wejść, z każdego komutatora sekcji wejściowej mogą być zestawione nie więcej niż dwa połączenia. Analogicznie, z powodu ograniczonej liczby wyjść, do każdego komutatora sekcji wyjściowej mogą być prowadzone co najwyżej dwa połączenia. Oznacza to, że jeśli p macierzy \mathbf{A}_k , $0 \leq k \leq p - 1$ ma reprezentować połączenia realizowane w polu $\log_2(N, 0, p)$, to spełnione muszą być nierówności:

$$0 \leq \forall_{j \in \langle 0, N/2-1 \rangle} \sum_{k=0}^{p-1} \sum_{x=0}^{N/2-1} a_k[x, j] \leq 2, \quad (3.4)$$

oraz

$$0 \leq \forall_{i \in \langle 0, N/2-1 \rangle} \sum_{k=0}^{p-1} \sum_{y=0}^{2^{n-1}-1} a_k[i, y] \leq 2, \quad (3.5)$$

czyli w każdej kolumnie j , $0 \leq j \leq N/2 - 1$, wszystkich macierzy występują co najwyżej dwa niezerowe elementy i w każdym wierszu i , $0 \leq i \leq N/2 - 1$, wszystkich macierzy występują co najwyżej dwa niezerowe elementy.

Na rysunku 3.2 przedstawiono macierz \mathbf{A}_0 , w której zrealizowano połączenia z rysunku 3.1, natomiast na rysunku 3.3 przedstawiono macierze \mathbf{A} dla wszystkich płaszczyzn pola, w którym są realizowane połączenia z rysunków 3.4 oraz 3.5. Wartości zerowe są brane pod uwagę w trakcie obliczeń wykonywanych w dalszych etapach, lecz nie wpływają na ich wynik. Dla podniesienia czytelności na rysunkach zostały one pominięte.

3.2. Macierz stanu pola – macierz E

W niektórych sytuacjach analiza stanu pola z dokładnością do numeru komutatora wejściowego i wyjściowego będzie niewystarczająca. Na potrzeby takiej szczegółowej analizy jest wykorzystywana dodatkowa macierz kwadratowa \mathbf{E} o wymiarach $N \times N$. Wartości jej elementów określa definicja 3.2. W tej macierzy wiersz odpowiada pojedynczemu wejściu pola, a kolumna – pojedynczemu wyjściu pola, zatem warunki 3.4

A0	0	1	2	3	4	5	6	7
0	1							
1						1		
2								
3								
4								
5	1							
6								
7								1

Rysunek 3.2. Macierz A_0 dla pola o $N = 16$ z połączeniami z rysunku 3.1

A0	0	1	2	3	4	5	6	7
0								
1								
2					1			
3								
4								
5	1							
6						1		
7								

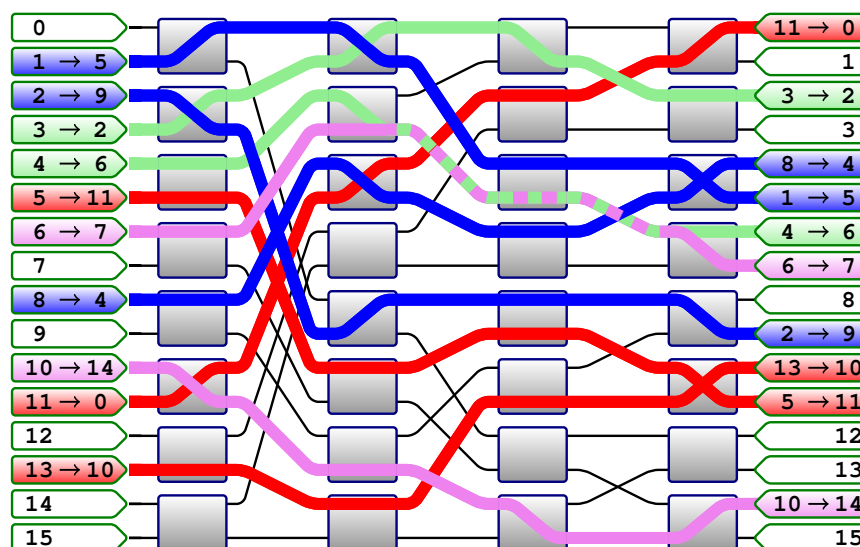
A1	0	1	2	3	4	5	6	7
0								
1		1						
2				1				
3								
4								
5								
6								
7								

A2	0	1	2	3	4	5	6	7
0			1					
1					1			
2								
3								
4				1				
5								
6								
7								

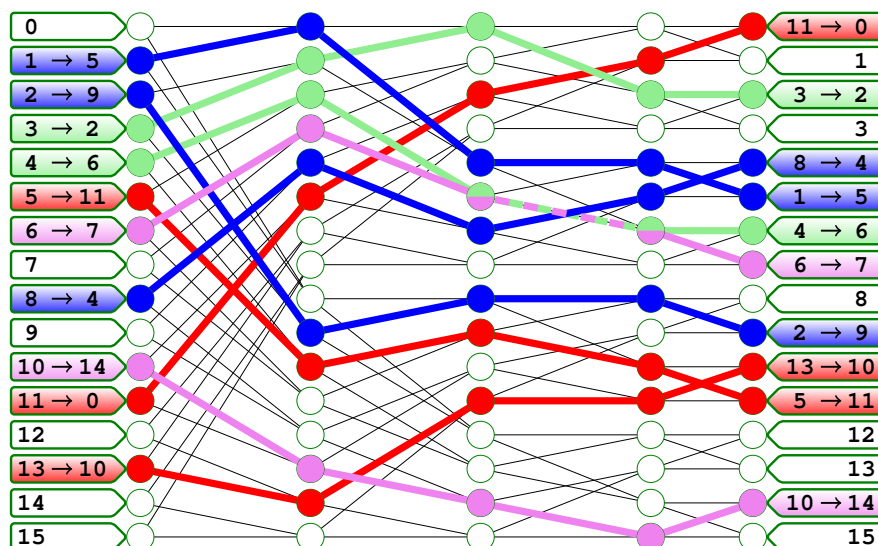
A3	0	1	2	3	4	5	6	7
0								
1								
2								
3				1				
4								
5							1	
6								
7								

A4	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

Rysunek 3.3. Macierze A dla pola o $N = 16$ dla wszystkich płaszczyzn z połączeniami z rysunku 3.5



Rysunek 3.4. Pole komutacyjne z połączeniami w różnych płaszczyznach, połączenia w różnych płaszczyznach są oznaczone różnymi kolorami ($N = 16, p = 4$)



Rysunek 3.5. Graf dwudzielny pola komutacyjnego z rysunku 3.4

oraz 3.5 muszą być odpowiednio zmodyfikowane – w każdej kolumnie i każdym wierszy może być co najwyżej jedna wartość.

Definicja 3.2:

$$E = \left[e[x, y] = \begin{cases} p, & \text{jeśli w płaszczyźnie } p \text{ jest zestawione połączenie } \langle x, y \rangle \\ \text{brak wartości,} & \text{w pozostałych przypadkach.} \end{cases} \right] \quad (3.6)$$

Macierz E przedstawiającą przykładowy stan pola pokazano na rysunku 3.6.

E	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1					2											
2										2						
3		1														
4						1										
5											0					
6							3									
7																
8				2												
9																
10														3		
11	0															
12																
13										0						
14																
15																

Rysunek 3.6. Macierz E – macierz stanu pola z ujętymi wszystkimi połączeniami w różnych płaszczyznach

3.3. Macierz blokowanych relacji w płaszczyźnie p – macierz B_p

Realizacja połączenia $\langle x, y \rangle$ wiąże się z zajęciem przez to połączenie łączy międzysekcyjnych. W reprezentacji pola za pomocą grafu dwudzielnego, trasa takiego połączenia w polu n -sekcyjnym zawiera $n + 1$ węzłów reprezentujących łącza. W skład trasy wchodzi po jednym węźle z każdej sekcji węzłów (jeden z

węzłów reprezentujących wejścia, po jednym z każdego z $n - 1$ podzbiorów węzłów reprezentujących łącza międzysekcyjne, oraz jeden z węzłów z podzbioru reprezentujących wyjścia). Numery węzłów w poszczególnych sekcjach wykorzystywanych przez trasy połączeń są określone wzorem (2.1). Analiza tego wzoru oraz przebiegu tras różnych połączeń pokazuje, że możliwe jest wskazanie różnych połączeń $\langle x, y \rangle$, których trasa będzie wykorzystywała ten sam węzeł w co najmniej jednej sekcji. W danej płaszczyźnie realizacja połączenia $\langle x, y \rangle$ jest możliwa tylko wtedy, jeśli żaden z węzłów koniecznych do budowy trasy tego połączenia nie jest wykorzystywany przez trasy innych połączeń realizowanych przez tę płaszczyznę. Jeśli choć jeden z nich jest wykorzystywany przez trasę dla innego połączenia, realizacja połączenia $\langle x, y \rangle$ w tej płaszczyźnie jest niemożliwa – połączenie to jest w tej płaszczyźnie zablokowane, a płaszczyzna ta jest dla tego połączenia niedostępna. Z każdego węzła sekcji s jest osiągalnych 2^s wejść oraz 2^{n-s} wyjść. Przez każdy węzeł może przechodzić $2^{n-s} \cdot 2^s = 2^n = N$ tras różnych połączeń. Zablokowanie jednego węzła w danej płaszczyźnie blokuje tę płaszczyznę dla N różnych połączeń (pomiędzy wejściami i wyjściami osiągalnymi z danego węzła).

Do reprezentowania dostępności poszczególnych relacji w danym stanie pola wykorzystywanych jest p macierzy B_k o rozmiarach $N/2 \times N/2$, gdzie $0 \leq k \leq p - 1$, przy czym dostępność płaszczyzny k dla połączenia relacji (i, j) w płaszczyźnie o numerze k reprezentuje element $b_k[i, j]$ z macierzy B_k . Znaczenie poszczególnych elementów macierzy \mathbf{B}_k określa definicja 3.3.

Definicja 3.3:

$$\mathbf{B}_k = \left[b_k[i, j] = \begin{cases} 0, & \text{jeśli w płaszczyźnie } k \text{ połączenie } (i, j) \text{ nie jest zablokowane} \\ w, & \text{jeśli w płaszczyźnie } k \text{ połączenie } (i, j) \text{ jest zablokowane przez } w \text{ połączeń} \end{cases} \right] \quad (3.7)$$

Wartości poszczególnych elementów $b_k[i, j]$ macierzy B_k mogą być obliczone na podstawie wzorów:

$$b_k[i, j] = \sum_{y=W_1}^{W_2} a_k[i, y] + \sum_{s=0}^{(n-3)} \sum_{x=W_3}^{W_4} \sum_{y=W_5}^{W_6} a_k[x, y], \quad (3.8)$$

gdzie:

$$W_1 = j - j \bmod 2^{n-2}, \quad (3.9)$$

$$W_2 = W_1 + 2^{n-2} - 1, \quad (3.10)$$

$$W_3 = (i - i \bmod 2^s) + 2^s(1 - 2 \cdot \lfloor i/2^s \rfloor \bmod 2), \quad (3.11)$$

$$W_4 = W_3 + 2^s - 1, \quad (3.12)$$

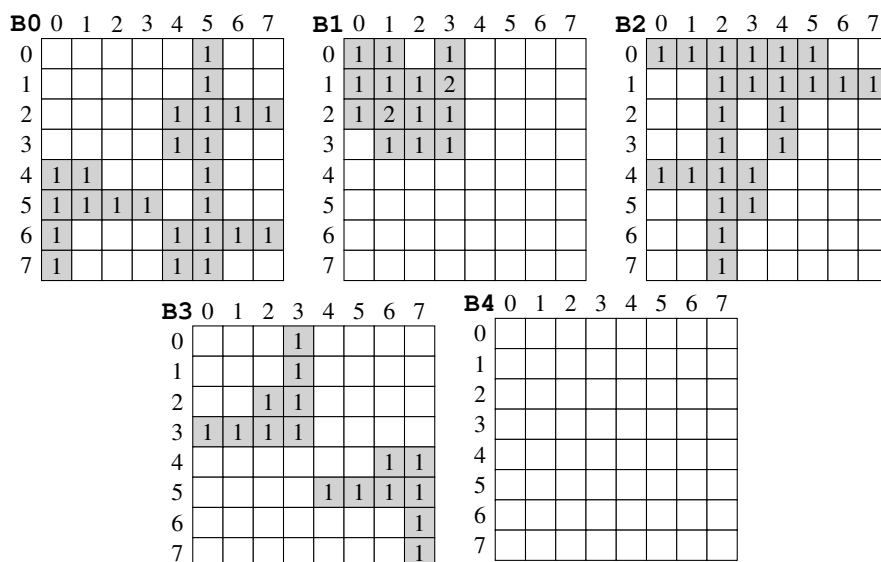
$$W_5 = j - j \bmod 2^{n-3-s}, \quad (3.13)$$

$$W_6 = W_5 + 2^{n-3-s} - 1, \quad (3.14)$$

a $a \bmod b$ oznacza resztę z dzielenia a przez b .

Elementy $b_k[i, j]$ mogą przyjmować wartości z przedziału $\langle 0, n - 1 \rangle$ i zgodnie z definicją 3.3 oznaczają liczbę połączeń blokujących połączenie relacji (i, j) w płaszczyźnie k . Aby w płaszczyźnie k można zestawić połączenie $\langle x, y \rangle$ musi być spełniona równość:

$$b_k[\lfloor x/2 \rfloor, \lfloor y/2 \rfloor] = 0, \quad (3.15)$$



Rysunek 3.7. Macierze B dla pięciu płaszczyzn pola o $N=16$

czyli płaszczyzna k jest dostępna dla połączenia relacji (i, j) tylko jeśli $b_k[i, j] = 0$. Z punktu widzenia dostępności dla połączeń danej relacji (i, j) płaszczyzna k ma status dostępnej ($b_k[i, j] = 0$) lub niedostępnej ($b_k[i, j] > 0$). Macierze B dla płaszczyzn realizujących przykładowe połączenia rysunku 3.6 przedstawiono na rysunku 3.7.

3.4. Macierz kosztu realizacji połączenia – macierz C

Zestawienie połączenia w jednej z płaszczyzn czyni tę płaszczyznę niedostępną dla połączeń pewnych relacji. Zestawienie połączenia relacji (i, j) blokuje połączenia tych relacji, których połączenia blokowałby możliwość zestawienia połączenia relacji (i, j) , czyli jeśli połączenie $\langle a, b \rangle$ blokuje połączenie $\langle c, d \rangle$ to połączenie $\langle c, d \rangle$ blokuje połączenie $\langle a, b \rangle$. Konsekwencją tej swoistej wzajemności jest fakt, iż zestawienie połączenia relacji (i, j) w płaszczyźnie k (czyli podstawienie $a_k[i, j] = 1$) oznacza zablokowanie tych relacji (wystąpienie wartości $b_k[i, j] > 0$), których połączenia zablokowałyby połączenie relacji (i, j) .

Na podstawie wzoru (3.8), oprócz liczby zablokowanych relacji, można także określić, które relacje zostałyby zablokowane w przypadku zestawienia połączenia relacji (i, j) . Spośród $m = n2^{n-3}$ relacji, które zostałyby zablokowane w płaszczyźnie k przez zestawienie połączenia relacji (i, j) , część może być zablokowana również przez istniejące wcześniej połączenia (nie zmieni się ich dostępność dla nowych połączeń po zestawieniu połączenia relacji (i, j) w tej płaszczyźnie), a część relacji zmieni swój status z dostępnych na niedostępne (będą blokowane tylko przez połączenie relacji (i, j) a nie przez wcześniejsze połączenia). W danym stanie pola można wskazać liczbę relacji, które zmieniłyby swój status z dostępnego na niedostępny jeśli połączenie relacji (i, j) zostałyby zestawione w płaszczyźnie k . Te informacje będą przechowywane w p macierzach C_k o rozmiarach $N/2 \times N/2$, gdzie $1 \leq k \leq p - 1$. Wartość elementów macierzy C_k określa definicja 3.4.

Definicja 3.4:

$$\mathbf{C}_k = \left[c_k[i, j] = \begin{cases} m + 1, & \text{jeśli w płaszczyźnie } k \text{ połączenie relacji } (i, j) \text{ jest zablokowane} \\ m - d, & \text{jeśli w płaszczyźnie } k \text{ połączenie relacji } (i, j) \text{ może być zestawione} \end{cases} \right], \quad (3.16)$$

gdzie:

$$d[i, j] = \sum_{y=W_1}^{W_2} b_k[i, y] + \sum_{s=0}^{(n-3)} \sum_{x=W_3}^{W_4} \sum_{y=W_5}^{W_6} f(b_k[x, y]), \quad (3.17)$$

$$m = n2^{n-3}, \quad (3.18)$$

$$W_1 = j - j \bmod 2^{n-2}, \quad (3.19)$$

$$W_2 = W_1 + 2^{n-2} - 1, \quad (3.20)$$

$$W_3 = (i - i \bmod 2^s) + 2^s(1 - 2 \cdot \lfloor i/2^s \rfloor \bmod 2), \quad (3.21)$$

$$W_4 = W_3 + 2^s - 1, \quad (3.22)$$

$$W_5 = j - j \bmod 2^{n-3-s}, \quad (3.23)$$

$$W_6 = W_5 + 2^{n-3-s} - 1, \quad (3.24)$$

oraz

$$f(z) = \begin{cases} 0 \Leftrightarrow z = 0 \\ 1 \Leftrightarrow z \neq 0 \end{cases}. \quad (3.25)$$

Wartość m oznacza liczbę relacji blokowanych przez każde zestawione połączenie, natomiast d oznacza liczbę tych relacji, które w danym stanie są już zablokowane przez połączenia istniejące w polu i zostaną także zablokowane przez połączenie relacji (i, j) . Wartość m jest zależna wyłącznie od pojemności pola, natomiast wartość d zależy od pojemności pola oraz od jego aktualnego stanu.

Element macierzy C_k na pozycji i, j nie pokazuje liczby relacji, które zostałyby zablokowane, gdyby w płaszczyźnie k zestawione zostało połączenie relacji (i, j) (gdyż ta jest stała dla danego n i wynosi m). Wartość elementu macierzy C_k na pozycji i, j określa liczbę relacji połączeń, dla których płaszczyzna k w danym stanie pola jest dostępna, a zostanie zablokowana po zestawieniu połączenia (i, j) w tej płaszczyźnie. Liczba takich relacji będzie traktowana jako koszt realizacji połączenia (i, j) w płaszczyźnie k . Do zestawienia połączenia (i, j) będzie wybierana płaszczyzna o takim numerze k , dla którego wartość elementu $c_k[i, j]$ jest najmniejsza.

Macierze C dla wszystkich płaszczyzn pola realizującego połączenia reprezentowane w macierzach A z rysunku 3.3, są przedstawione na rysunku 3.8.

3.5. Macierz wyboru płaszczyzny – macierz D

Aby zmniejszyć liczbę wykonanych operacji do momentu uzyskania wyniku algorytmu i przyspieszyć moment podjęcia decyzji można dokonać hipotetycznego wyboru płaszczyzn dla wszystkich relacji, których połączeń możemy się spodziewać, zanim nowe połączenie nadejdzie. Dla całego pola składającego się z p

C0	0	1	2	3	4	5	6	7	C1	0	1	2	3	4	5	6	7	C2	0	1	2	3	4	5	6	7
0	8	8	8	8	4	9	6	6	0	9	9	1	9	8	8	8	8	0	9	9	9	9	9	9	4	4
1	8	8	8	8	4	9	6	6	1	9	9	9	9	8	8	8	8	1	4	4	9	9	9	9	9	9
2	8	8	8	8	9	9	9	9	2	9	9	9	9	8	8	8	8	2	6	6	9	4	9	4	6	6
3	8	8	8	8	9	9	4	4	3	1	9	9	9	8	8	8	8	3	6	6	9	4	9	4	6	6
4	9	9	4	4	4	9	6	6	4	8	8	8	8	8	8	8	8	4	9	9	9	9	8	8	8	8
5	9	9	9	9	4	9	6	6	5	8	8	8	8	8	8	8	8	5	4	4	9	9	8	8	8	8
6	9	4	6	6	9	9	9	9	6	8	8	8	8	8	8	8	8	6	6	6	9	4	8	8	8	8
7	9	4	6	6	9	9	4	4	7	8	8	8	8	8	8	8	8	7	6	6	9	4	8	8	8	8

C3	0	1	2	3	4	5	6	7	C4	0	1	2	3	4	5	6	7
0	6	6	4	9	8	8	8	8	0	8	8	8	8	8	8	8	8
1	6	6	4	9	8	8	8	8	1	8	8	8	8	8	8	8	8
2	4	4	9	9	8	8	8	8	2	8	8	8	8	8	8	8	8
3	9	9	9	9	8	8	8	8	3	8	8	8	8	8	8	8	8
4	8	8	8	8	4	4	9	9	4	8	8	8	8	8	8	8	8
5	8	8	8	8	9	9	9	9	5	8	8	8	8	8	8	8	8
6	8	8	8	8	6	6	4	9	6	8	8	8	8	8	8	8	8
7	8	8	8	8	6	6	4	9	7	8	8	8	8	8	8	8	8

Rysunek 3.8. Macierze C dla pięciu płaszczyzn pola o N=16

płaszczyzn będzie wykorzystywana jedna macierz D o wymiarach $N/2 \times N/2$. Każdy jej element $d[i, j]$ będzie wyznaczony według wzoru:

$$d[i, j] = k : c_k[i, j] = \min\{c_l[i, j] : l \in \langle 0, p - 1 \rangle\}. \tag{3.26}$$

Wartości poszczególnych elementów macierzy D są z przedziału $\langle 0, p - 1 \rangle$, gdyż określają numery płaszczyzn, w których koszt zestawienia połączenia jest najniższy.

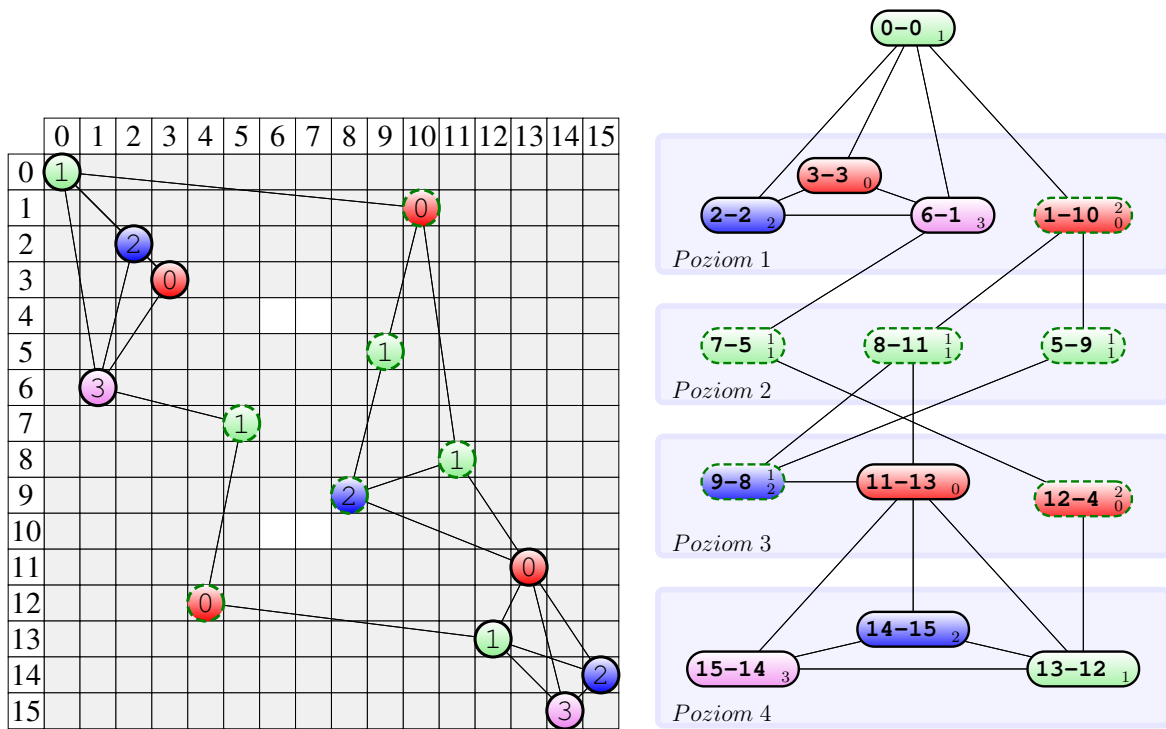
3.6. Graf stanu pola

Innym sposobem reprezentacji stanu pola jest graf stanu pola, który jest definiowany w następujący sposób:

Definicja 3.5: *Graf stanu pola* $G(V, K)$ składa ze zbioru węzłów V oraz zbioru krawędzi K . Do zbioru V należą węzły W , każdy z nich reprezentuje połączenie istniejące w polu. Każdy węzeł grafu jest opisany przez parametry x, y i p , gdzie x i y to odpowiednio numer wejścia i wyjścia reprezentowanego połączenia, a p to numer płaszczyzny, w której jest realizowane połączenie $\langle x, y \rangle$. W zależności od dostępności i istotności danych, węzeł taki będzie określany jako W_x , $W_{x,y}$ lub $W_{x,y,p}$. Zbiór krawędzi K to zbiór dwuelementowych par złożonych z węzłów będących elementami zbioru V . Krawędź tworzą dwa węzły W_{x_1,y_1} i W_{x_2,y_2} jeżeli połączenia przez nie reprezentowane ($\langle x_1, y_1 \rangle$ i $\langle x_2, y_2 \rangle$) wzajemnie się blokują.

Reprezentacja grafu stanu na płaszczyźnie może zostać zrealizowana jako nałożenie grafu na rysunek macierzy stanu pola (macierz E) lub też samodzielny graf. Macierz E z nałożonym na nią grafem stanu pola będzie nazywana **macierzą G**. O pozycji poszczególnych węzłów w macierzy G decydują parametry reprezentowanego połączenia. Wówczas węzeł będzie nazwany numerem płaszczyzny zajmowanej przez jego połączenie. Macierz G reprezentującą przykładowy stan pola przedstawiono na rysunku 3.9.

W przypadku samodzielnej reprezentacji grafu nie ma bezpośredniego przełożenia parametrów reprezentowanego połączenia na lokalizację (pozycję) węzła na rysunku. Samodzielna reprezentacja będzie przygotowywana pod kątem analizy stanu pola z punktu widzenia konkretnego połączenia. Wówczas pozycja poszczególnych węzłów będzie zależała od wzajemnych relacji między połączeniami przez nie reprezentowanymi. Węzeł



Rysunek 3.9. $1/4$ macierzy G dla pola $n = 5$ ($N = 32$) reprezentująca przykładowy stan pola oraz ten sam graf w reprezentacji samodzielnej

reprezentujący rozważane połączenie będzie na górze rysunku. Najczęściej będzie to węzeł tymczasowy, reprezentujący połączenie blokowane we wszystkich płaszczyznach i chwilowo nie będzie możliwe wskazanie dla niego płaszczyzny. Przykładowy graf stanu z rysunku 3.9 został narysowany pod kątem analizy sytuacji połączenia $(0, 0)$, które zostało zestawione w płaszczyźnie $p = 1$. Węzły grafu, które reprezentują połączenia blokujące rozważane połączenie umieszczone są w jednej linii poniżej i stanowią poziom 1. Na poziomie 2 są umieszczone te węzły spośród dotychczas nierozmieszczonych, które blokują połączenia reprezentowane przez węzły na poziomie 1. Na poziomie 3 znajdują się węzły reprezentujące połączenia blokujące połączenia z poziomu drugiego. Kolejne poziomy grafu będą budowane analogicznie, tzn. na poziomie n znajdują się dotychczas nierozmieszczone węzły, które blokują węzły na poziomie $n - 1$. W reprezentacji samodzielnej nazwa każdego węzła będzie postaci $x - y_p^w$, gdzie x i y oznaczają odpowiednio numer wejścia i wyjścia reprezentowanego połączenia, p – numer płaszczyzny zajmowanej przez reprezentowane połączenie, natomiast w oznacza liczbę płaszczyzn, na które reprezentowane połączenie może być przeniesione w danym stanie pola. Ta reprezentacja jest wykorzystywana podczas definiowania algorytmu przestrożeń.

Algorytmy sterowania zaproponowane w pracy

4.1. Macierzowy algorytm sterowania polem

Na bazie macierzowej reprezentacji stanu pola został skonstruowany algorytm sterowania polem, którego zadaniem jest wskazanie płaszczyzny dla nowego połączenia. Wskazana płaszczyzna jest optymalna ze względu na liczbę połączeń, które zostaną zablokowane w płaszczyźnie realizującej nowe połączenie. Czynności, które należy wykonać, aby określić numer płaszczyzny dla nowego połączenia, bazując bezpośrednio na macierzowej reprezentacji stanu pola, zestawiono na rysunku 4.1.

Algorytm 1: Algorytm macierzowy sterowania polem

- Krok 1:** Oczekuj na żądanie zestawienia nowego lub rozłączenia istniejącego połączenia.
 - Krok 2:** Jeśli żądane jest rozłączenie połączenia relacji (i, j) w płaszczyźnie P , idź do kroku **9**.
 - Krok 3:** Jeśli żądane jest zestawienie nowego połączenia, idź do kroku **4**.
 - Krok 4:** Dokonaj aktualizacji wartości elementów wszystkich macierzy \mathbf{B} na pozycjach, które reprezentują relacje połączeń blokowanych przez połączenie relacji (i, j) .
 - Krok 5:** Dokonaj aktualizacji wartości elementów na pozycji $[i, j]$ we wszystkich macierzach \mathbf{C} .
 - Krok 6:** Znajdź numer płaszczyzny P , dla którego wartość $c_P[i, j]$ osiąga wartość najmniejszą.
 - Krok 7:** Zestaw połączenie relacji (i, j) w płaszczyźnie P .
 - Krok 8:** Uaktualnij macierz $\mathbf{A}_P - a_P[i, j] = 1$, idź do kroku **1**.
 - Krok 9:** Rozłącz połączenie relacji (i, j) w płaszczyźnie P .
 - Krok 10:** Uaktualnij macierz $\mathbf{A}_P - a_P[i, j] = 0$, idź do kroku **1**.
-

Rysunek 4.1. Kroki algorytmu macierzowego

Dokładniejsza analiza przepływu informacji pozwala wyróżnić cztery etapy, które muszą być wykonane. Zestawione są one w tabeli 4.1. Etapy te mogą być wykonane w dowolnym momencie, pod warunkiem, że nie zostanie zaburzona zasada korzystania z aktualnej informacji, tzn. poszczególne dane mogą być wykorzystane jedynie, jeśli od ich ostatniego uaktualnienia nie zmienił się stan pola reprezentowany przez wartości, na podstawie których one są obliczane. Z zachowaniem tej zasady można skonstruować cztery różne algorytmy, które

Tabela 4.1.

Czynności, które muszą być wykonane w każdym pełnym przebiegu algorytmu

Etap	Zadania etapu
A	Aktualizacja elementów odpowiedniej macierzy \mathbf{A}_P w zależności od zestawianego lub rozłączanego połączenia (i, j)
B	Aktualizacja elementów odpowiednich macierzy \mathbf{B} na podstawie aktualnych elementów macierzy \mathbf{A} .
C	Aktualizacja elementów odpowiednich macierzy \mathbf{C} na podstawie aktualnych elementów odpowiednich macierzy \mathbf{B} .
D	Wybór płaszczyzny dla nowego połączenia poprzez porównanie wartości elementu $c_k[i, j]$ z macierzy dla każdej z płaszczyzn.

prowadzą do takiego samego wyniku, lecz ich pojedyncze przebiegi zaczynają się od różnych etapów. Analiza algorytmu 1 pozwala stwierdzić, że każdy przebieg algorytmu związany z zestawieniem połączenia rozpoczyna się od etapu B (krok 4), następnie są realizowane czynności etapu C (krok 5) oraz etapu D (krok 6), a cały przebieg kończy się etapem A (krok 8). Ten sposób realizacji algorytmu można określić jako konfigurację BCDA, w pracy stanowi ona implementację I_1 . Możliwa jest realizacja algorytmu w taki sposób, że poszczególne jego przebiegi rozpoczynają się od etapu D, wówczas cały przebieg jest realizowany w konfiguracji DABC, taka realizacja jest opisana w pracy jako implementacja I_2 . Analogicznie, realizacja algorytmu w konfiguracji CDAB jest opisana jako implementacja I_3 , natomiast konfiguracja ABCD jako implementacja I_4 . Najefektywniejsza jest implementacja I_4 , gdyż w pierwszym etapie poszczególnych przebiegów algorytmu dokonywane jest zestawienie połączenia. Numer płaszczyzny dedykowanej dla nowego połączenia jest znany już na samym początku przebiegu algorytmu – jest on odczytywany z macierzy \mathbf{D} . Zestawienie połączenia (i, j) w płaszczyźnie P oznacza wprowadzenie wartości 1 do macierzy \mathbf{A}_P . Zmiana nastąpiła w jednej macierzy \mathbf{A}_P , więc wystarczy aktualizować elementy tylko jednej macierzy \mathbf{B}_P . Co więcej – z uwagi na pewne zależności można ograniczyć się do aktualizacji elementów tylko w jednej ćwiartce macierzy \mathbf{B}_P . Realizacja etapu C polega na aktualizacji elementów macierzy \mathbf{C}_P – wiadomo, że zmiany są tylko w macierzy powiązanej z płaszczyzną o numerze P . Ostatnim krokiem jest aktualizacja elementów macierzy \mathbf{D} – w niej są przechowywane numery płaszczyzn, które w danym stanie pola mogą realizować dane połączenie z najniższym kosztem. Aktualizacja macierzy \mathbf{D} na końcu każdego przebiegu sprawia, że na początku następnego przebiegu znany jest numer płaszczyzny dla każdego możliwego do zestawienia połączenia. Macierz \mathbf{D} będzie użyta jako źródło informacji o numerze płaszczyzny, w której zostanie zestawione nowe połączenie. Taka realizacja algorytmu pozwala uzyskać wynik już na samym początku przebiegu algorytmu, natomiast wszelkie obliczenia mają charakter uaktualnień i są wykonywane po wskazaniu płaszczyzny dla nowego połączenia.

Implementacja I_4 algorytmu macierzowego

Krok 1: Zestaw połączenie (i, j) w płaszczyźnie $P = d[i, j]$, zaktualizuj A_P .

Krok 2: Zaktualizuj odpowiednią ćwiartkę jednej macierzy B_P .

Krok 3: Zaktualizuj odpowiednią ćwiartkę jednej macierzy C_P .

Krok 4: Zaktualizuj odpowiednią ćwiartkę macierzy D .

Rysunek 4.2. Implementacja I_4

4.2. Kolejnościowy algorytm zestawiania połączeń

W pracy został przeanalizowany kolejnościowy sposób przydziału płaszczyzn dla kolejno zestawianych połączeń. Podczas tej analizy założyłem, że wszystkie połączenia są zestawiane jednocześnie, a połączeniom z kolejnych wejść przydzielane są dostępne płaszczyzny o najniższych numerach. Zaletą tego sposobu jest prostota sterowania. Okazuje się, że przy takim podejściu warunki nieblokowania dla pól o parzystej liczbie sekcji określone są wzorem:

$$p \geq 2^{\frac{n}{2}}. \quad (4.1)$$

i są dokładnie takie same jak warunki przestrajalności, natomiast w przypadku pól o nieparzystej liczbie sekcji są nieco wyższe i określone są wzorem:

$$p \geq \begin{cases} 2 & \text{dla } n = 3; \\ 1,25 * 2^{\frac{n-1}{2}} & \text{dla } n > 3. \end{cases} \quad (4.2)$$

Dodatkowe w stosunku do warunków przestrajalności płaszczyzny są konieczne, gdyż w pewnych sytuacjach występuje niekorzystny układ połączeń i przez przydział płaszczyzn o kolejnych numerach zajmowane są zasoby, które są ostatnimi w obrębie warunków przestrajalności dostępnymi dla połączeń analizowanych później. To właśnie ich realizacja wymaga dodatkowych płaszczyzn powodując wzrost warunków w stosunku do warunków przestrajalności. Modyfikacja polegająca na zaburzeniu kolejnościowego przydziału zasobów w niektórych sytuacjach pozwala zredukować warunki konieczne do warunków przestrajalności. Ceną za obniżenie wymagań na liczbę płaszczyzn jest komplikacja sterowania, niemniej algorytm ten może być z powodzeniem wykorzystany do komutacji jednoczesnej lub też jako prosty algorytm przestrojeń, przy czym w momencie nadejścia zablokowanego połączenia wszystkie istniejące połączenia byłyby rozłączane i zestawiane na nowo.

4.3. Algorytmy przestrojeń w polu o parzystej liczbie sekcji

Przeprowadzone przeze mnie badania wykazały, że w przestrajalnych polach o parzystej liczbie sekcji maksymalna liczba przestrojeń koniecznych do odblokowania płaszczyzny dla nowego połączenia wynosi 1. Oznacza to, że stosując odpowiedni algorytm można wskazać płaszczyznę dla nowego połączenia przestrajając maksymalnie jedno już istniejące w polu połączenie. Taki algorytm został przeze mnie zaproponowany w pracy. Algorytm ten został przedstawiony w dwóch wersjach: w wersji podstawowej, która dopuszcza przerywanie połączeń istniejących w polu na czas realizacji algorytmu, oraz wersję rozszerzoną, która realizuje wskazany

Algorytm 4: Algorytm przestrojeń w polach o parzystej liczbie sekcji bez przerywania istniejących połączeń

Krok 1: Oczekuj żądania zestawienia nowego połączenia.

Krok 2: Spróbuj znaleźć płaszczyznę dla nowego połączenia zgodnie z używanym algorytmem sterowania polem.

Krok 3: Jeśli płaszczyzna p_{new} dla nowego połączenia została wskazana, $c = p_{new}$, idź do kroku 8.

Krok 4: Skonstruuj następujące zbiory płaszczyzn:

S_a - zbiór płaszczyzn używanych przez połączenia typu A

S_b - zbiór płaszczyzn używanych przez połączenia typu B

S_c - zbiór płaszczyzn używanych przez połączenia typu C

S'_c - zbiór płaszczyzn używanych przez połączenia typu C, a nieużywane przez połączenia typu A i B,

$S'_c = S_c \setminus (S_a \cup S_b)$.

Krok 5: Wybierz połączenie C_c realizowane w płaszczyźnie c ze zbioru S'_c , takie, które może być przeniesione do płaszczyzny a , $a \in (S_a \cup S_b)$

Krok 6: Zestaw połączenie C w płaszczyźnie a

Krok 7: Rozłącz połączenie C w płaszczyźnie c

Krok 8: Zestaw nowe połączenie w płaszczyźnie c

Rysunek 4.3. Algorytm przestrojeń w polach o parzystej liczbie sekcji bez przerywania istniejących połączeń

ciąg przestrojeń tak, że w każdym momencie jest zachowana ciągłość przestrajanego połączenia. Kroki algorytmu w wersji rozszerzonej przedstawiono na rysunku 4.3. Brak konieczności przerywania przestrajanego połączenia został osiągnięty poprzez odpowiedni dobór kolejności realizowanych czynności. Przestrajane połączenie jest najpierw zestawiane w docelowej płaszczyźnie, a dopiero następnie rozłączane w płaszczyźnie zwalnianej. Takie podejście umożliwia ciągłą transmisję danych i zwalnia z konieczności ewentualnego buforowania danych lub stosowania innych mechanizmów niedopuszczających do utraty danych.

4.4. Algorytmy przestrojeń w polu o nieparzystej liczbie sekcji

Analiza pól o nieparzystej liczbie sekcji jest o wiele bardziej złożona. Powodem różnicy są inne warunki przestrajalności. Mimo, że są one definiowane takim samym wzorem dla pól o parzystej i nieparzystej liczbie sekcji, jego realizacja sprowadza się do otrzymania innej zależności liczby koniecznych płaszczyzn od liczby sekcji. Ogólny wzór $p = 2^{\lfloor n/2 \rfloor}$ może być przedstawiony w postaci wariantowej:

$$p = 2^{\frac{n}{2}} \quad \text{dla } n \text{ parzystego}, \quad (4.3)$$

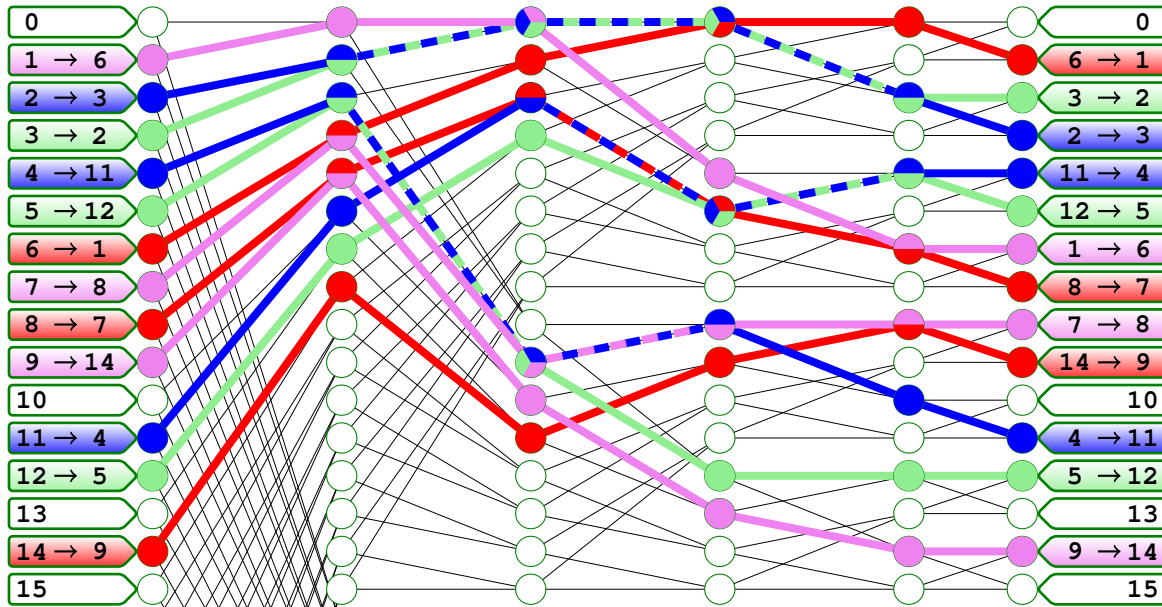
$$p = 2^{\frac{n-1}{2}} = 2^{(\frac{n}{2}-\frac{1}{2})} = \frac{\sqrt{2}}{2} 2^{\frac{n}{2}} = \frac{2^{\frac{n}{2}}}{\sqrt{2}} \quad \text{dla } n \text{ nieparzystego}. \quad (4.4)$$

Liczba płaszczyzn dostępna w przestrajalnych polach o różnych rozmiarach została zestawiona w tabeli 4.2. Okazuje się, że bez większych trudności można wskazać stany, w których konieczne jest więcej niż jedno

Tabela 4.2.

Liczba płaszczyzn p dla zapewnienia przestrajalności w polach o różnej liczbie sekcji n

n	2	3	4	5	6	7	8	9	10	11	12
N	4	8	16	32	64	128	256	512	1024	2048	4096
p	2	2	4	4	8	8	16	16	32	32	64

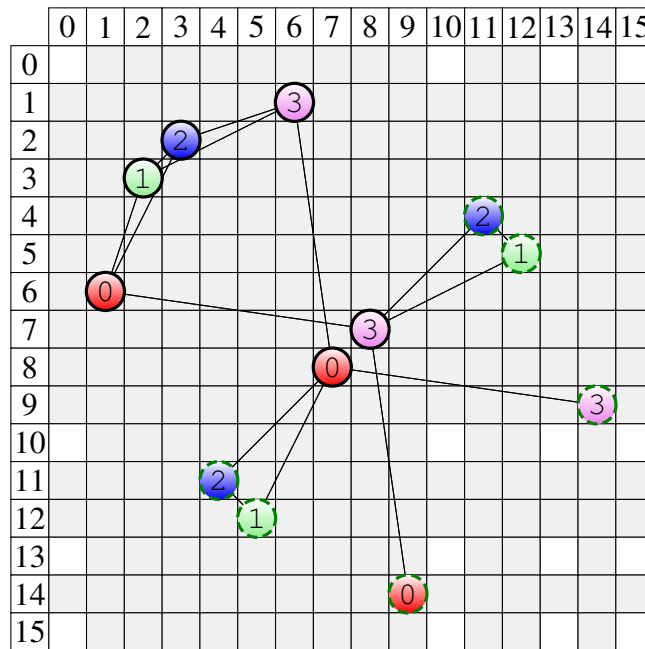


Rysunek 4.4. Graf pola ze stanem, w którym odblokowanie trasy dla połączenia $\langle 0, 0 \rangle$ wymaga trzech przestrojeń

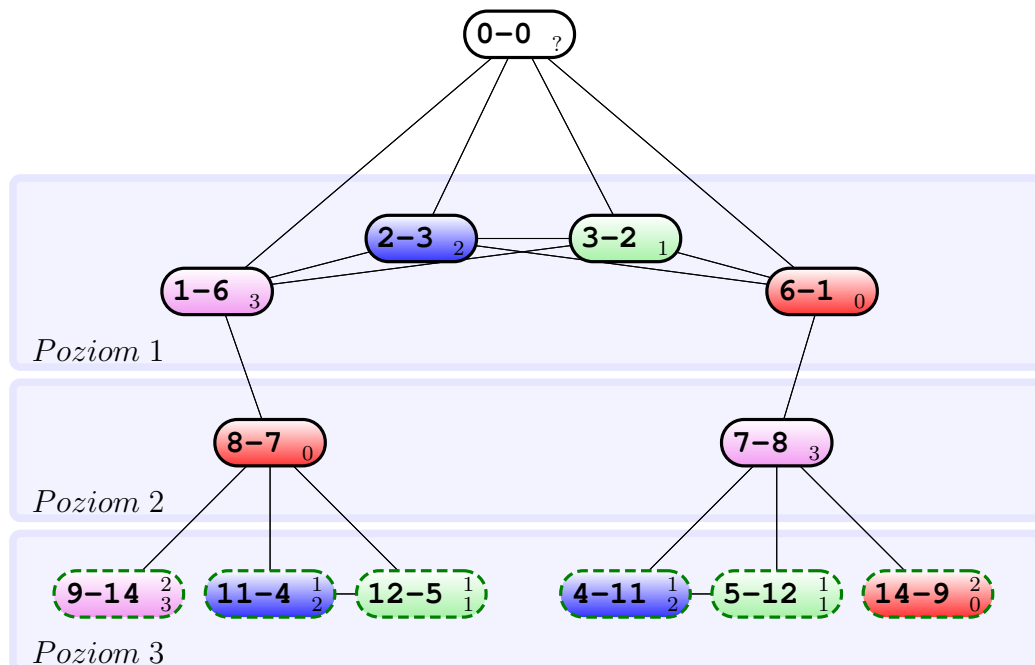
przestrojenie w celu odblokowania płaszczyzny dla nowego połączenia. Przeprowadzone badania wykazały, że maksymalna liczba koniecznych przestrojeń w przestrajalnych polach o nieparzystej liczbie sekcji wynosi 3.

Na rysunku 4.4 przedstawiono stan pola przestrajalnego o $N = 32$, w którym odblokowanie płaszczyzny dla połączenia $\langle 0, 0 \rangle$ wymaga trzech przestrojeń, a na rysunku 4.5 przedstawiono graf stanu tego pola wrysowany w macierz E . Rysunek 4.6 przedstawia graf tego samego stanu pola, lecz w postaci niezwiązanej z macierzą. Węzeł dla nowo zestawianego połączenia $\langle 0, 0 \rangle$ nie ma przypisanej płaszczyzny, nie może ona zostać w tym stanie wskazana, gdyż węzeł ten sąsiaduje z węzłami reprezentującymi połączenia w czterech różnych płaszczyznach tego pola, czyli połączenie $\langle 0, 0 \rangle$ jest zablokowane we wszystkich możliwych płaszczyznach.

Aby odblokować płaszczyznę dla połączenia $\langle 0, 0 \rangle$ należy znaleźć ścieżkę w grafie stanu pola, która prowadzi od węzła reprezentującego połączenie przestrajalne (posiadającego niezerową liczbę płaszczyzn alternatywnych, taki węzeł jest oznaczony linią przerywaną) do węzła reprezentującego zestawiane połączenie. Ścieżka ta musi spełniać szereg warunków, które zostały opisane w pracy. Przykładowy ciąg przestrojeń wygenerowany na bazie prezentowanego grafu stanu zaczyna się od przeniesienia połączenia $\langle 12, 5 \rangle$ do płaszczyzny różowej, po tej zmianie połączenie $\langle 8, 7 \rangle$ może zająć płaszczyznę zieloną, następnie połączenie $\langle 1, 6 \rangle$ może zostać przeniesione do płaszczyzny czerwonej. W tym momencie połączenie $\langle 0, 0 \rangle$ nie ma sąsiada w płaszczyźnie różowej i może zostać w niej zestawione. Tym samym odblokowana została płaszczyzna w



Rysunek 4.5. Macierz połączeń pola dla stanu z rys 4.4



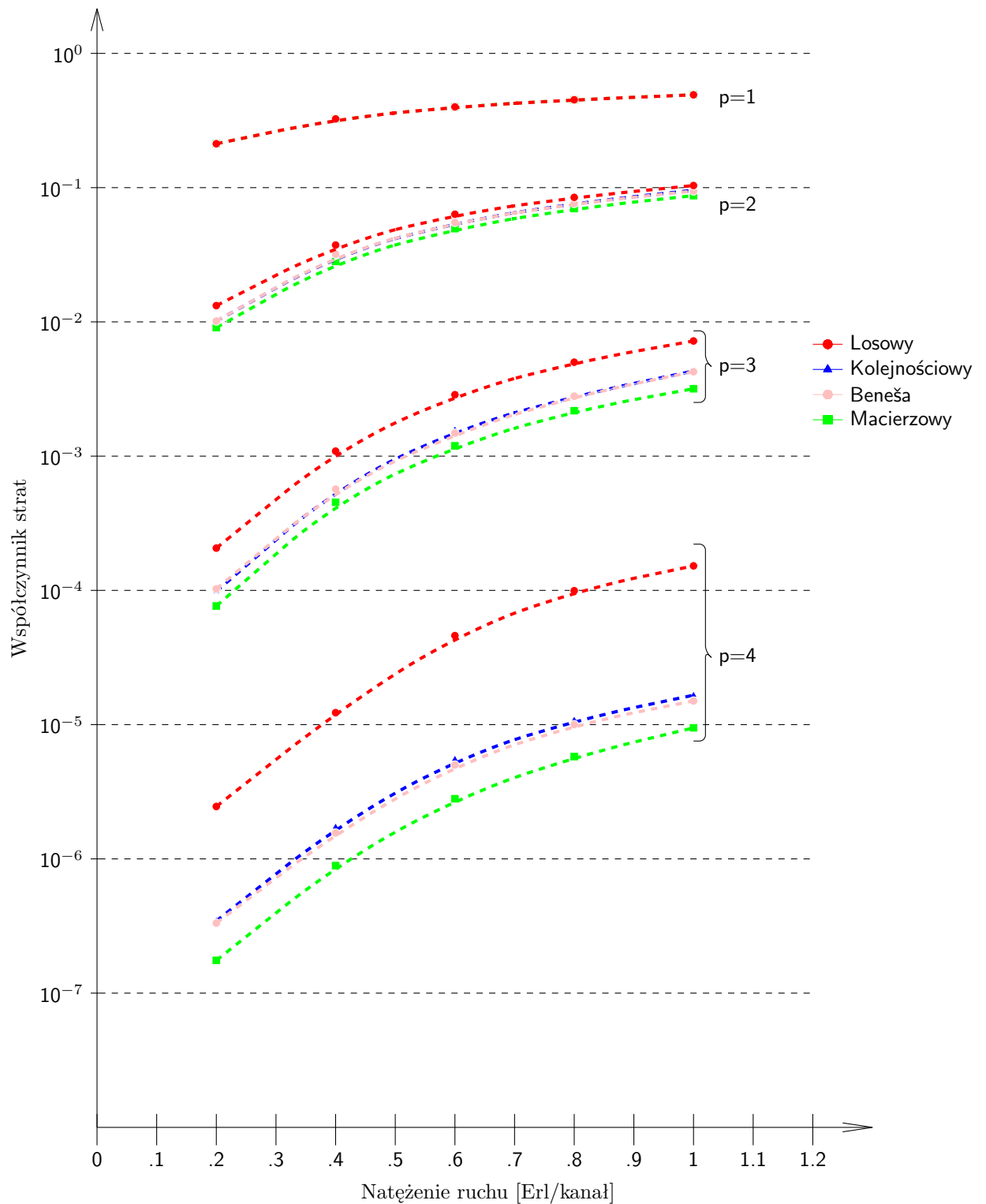
Rysunek 4.6. Graf stanu pola $\log_2(32, 0, 4)$ z punktu widzenia połączenia $\langle 0, 0 \rangle$ – wymagane są trzy przestrojenia do odblokowania płaszczyzny dla połączenia $\langle 0, 0 \rangle$, reprezentuje on połączenia z rysunku 4.4 i 4.5

5

Ocena zaproponowanych algorytmów

Rezultaty pracy macierzowego algorytmu sterowania polem zostały porównane z wynikami pracy algorytmów znanych z literatury. Zaproponowany nowy algorytm macierzowy został zestawiony z algorytmem kolejnościowym, losowym oraz Beneša. Wyzaczyłem poprzez symulację współczynniki strat poszczególnych algorytmów sterujących polami o różnej liczbie płaszczyzn i obciążonych ruchem o różnym natężeniu. Liczba płaszczyzn została tak dobrana, aby występowały pewne straty połączeń (tzn. niektóre połączenia zostaną stracone z powodu braku zasobów w polu do ich realizacji), dzięki temu można porównać jakość algorytmów. Na rysunku 5.1 przedstawiono graficznie wyniki symulacji współczynnika strat połączeń dla czterech algorytmów w polach o $N = 32$ zbudowanych z jednej, dwóch, trzech i czterech płaszczyzn. Poszczególne wartości były symulowane wielokrotnie, pozwoliło to uzyskać przedział ufności o rozmiarach poniżej rozmiarów symboli je reprezentujących. Algorytm losowo dobierający płaszczyzny dla nowych połączeń cechuje się najwyższym współczynnikiem strat, algorytm analizujący płaszczyzny kolejno oraz zgodnie z aktualnym obciążeniem poszczególnych płaszczyzn osiąga dużo lepszy (niższy) współczynnik strat, natomiast współczynnik strat osiągany przy sterowaniu algorytmem macierzowym jest najniższy ze wszystkich otrzymanych. Oznacza to, że w tych samych warunkach decyzje algorytmu macierzowego okazują się najlepsze z punktu widzenia optymalnego przydziału zasobów do realizacji poszczególnych połączeń. Mimo, iż matematyczny zapis funkcji kosztu, która jest wykorzystywana do podejmowania decyzji, wydaje się złożony, dzięki zrównolegleniu operacji wykonywanych w sprzętowej implementacji algorytmu, wszelkie obliczenia zajmują bardzo mało czasu (poniżej 10 ns.). Co więcej, odpowiednie uszeregowanie czynności algorytmu pozwala praktycznie natychmiast (na zasadzie odczytu z pamięci) uzyskać rezultat pracy algorytmu – numer płaszczyzny dedykowanej dla nowego połączenia.

W pracy określono minimalną liczbę przestrojeń koniecznych do odblokowania płaszczyzny dla nowego połączenia. W polach o parzystej liczbie sekcji nie ma stanów, których zestawienie nowego połączenia wymaga więcej niż jednego przestrojenia, natomiast w polach o nieparzystej liczbie sekcji maksymalna liczba koniecznych przestrojeń wynosi 3. Zaproponowane algorytmy korzystają z mechanizmów, które umożliwiają im znalezienie ciągów przestrojeń o minimalnej długości, czyli nowe połączenie jest zestawiane przez wykonanie możliwie najmniejszej liczby operacji.



Rysunek 5.1. Wyniki symulacji współczynnika strat dla algorytmów macierzowego, kolejnościowego, Beneša i losowego w polu o $N = 32$ oraz $p = 1, p = 2, p = 3$ i $p = 4$ płaszczyznach

6

Implementacja algorytmów

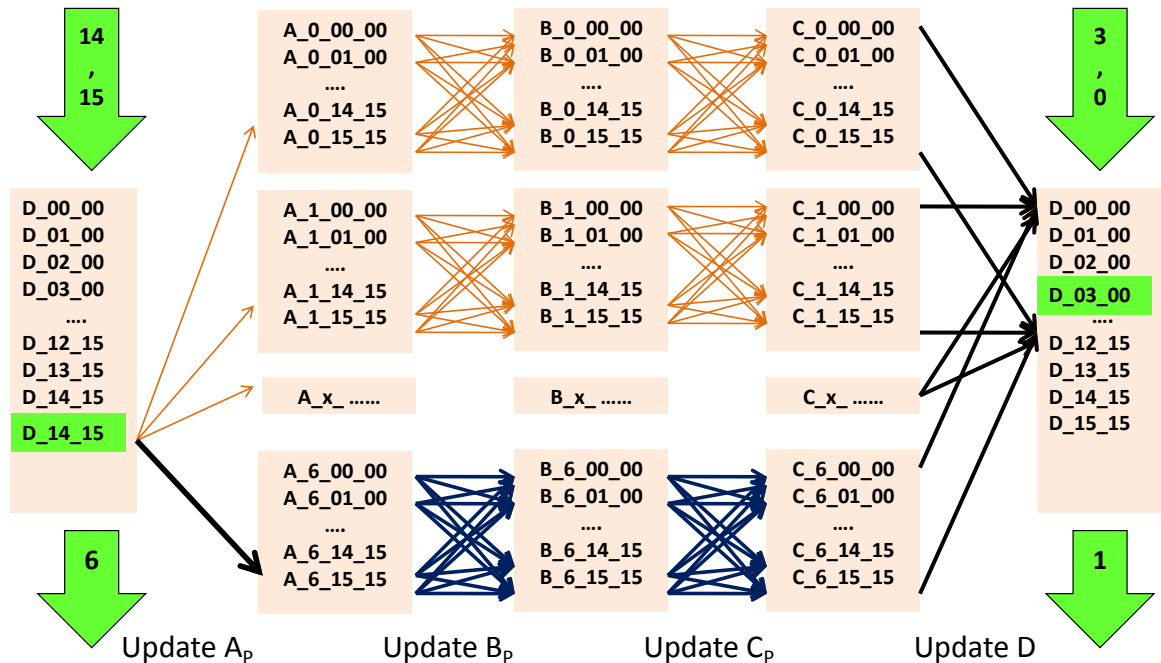
6.1. Implementacja algorytmu macierzowego w układzie FPGA

Jednym z założeń pracy było opracowanie takich algorytmów, których sprzętowa implementacja jest możliwa do zrealizowania w taki sposób, aby sterowniki pracujące w oparciu o nie miały znaczenie praktyczne. Mechanizmy algorytmów przedstawionych w pracy zostały zaadoptowane do wymagań i możliwości układów FPGA (*ang. Field Programmable Gate Array*). Jako fizyczną platformę sprzętową do realizacji algorytmów wybrałem dostępne w Katedrze płyty ewaluacyjne przygotowane przez firmę Xilinx dla układów Virtex5.

Spośród opisanych w pracy czterech wersji realizacji algorytmu macierzowego, do fizycznej implementacji w układzie FPGA została wybrana ta, która zapewnia najkrótszy możliwy czas oczekiwania na odpowiedź. W jej przypadku odpowiedź jest wyczytywana z macierzy **D** już w pierwszym kroku realizacji pojedynczego przebiegu, natomiast wszelkie obliczenia mają charakter aktualizacji informacji reprezentującej stan pola. Kod języka VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) realizujący implementację algorytmu został wygenerowany automatycznie przez przygotowane przeze mnie oprogramowanie. Odpowiednio przygotowany model pola posłużył za źródło informacji o tym, jakie zmienne są konieczne do przechowywania odpowiednich informacji. Wynikiem pracy jednej z funkcjonalności programu jest plik tekstowy zawierający gotowy fragment kodu VHDL definiujący odpowiednie zmienne. Również wszelkie zależności obliczeniowe konieczne do realizacji poszczególnych wzorów definiujących mechanizmy algorytmu zostały odwzorowane w odpowiednie instrukcje kodu VHDL. Poszczególne wzory określające wartości komórek macierzy **B**, **C** oraz **D** zostały zrealizowane dla konkretnych zbiorów danych wejściowych, tzn. na podstawie powiązań poszczególnych zmiennych w modelu sterownika pola został wygenerowany kod VHDL realizujący te same obliczenia w sposób sprzętowy. Przykład instrukcji dla obliczania wartości jednej komórki macierzy **B** przedstawiono na rysunku 6.1. Dla każdego innego elementu wykorzystywanych macierzy wygenerowano analogicznie fragmenty kodu VHDL. Dzięki wykorzystaniu precyzyjnie określonych wartości we wzorach, ich realizacja jest bardzo szybka (struktury sprzętowe nie realizują bezpośrednio wzorów ze zmiennymi parametrami). Każda obliczana wartość jest definiowana przez niezależny fragment kodu, co za tym idzie – jest implementowana przez niezależne zasoby sprzętowe i wszystkie obliczenia są wykonywane jednocześnie. Schemat realizacji przykładowego przebiegu algorytmu w układzie sprzętowym jest przedstawiony na rysunku 6.2. Na samym początku jest odczytywana informacja z komórki macierzy **D**, która została wskazana przez

```
B1_07_04 := A1_00_04 OR A1_01_04 OR A1_02_04 OR A1_03_04 OR A1_04_04 OR A1_04_05 OR
A1_05_04 OR A1_05_05 OR A1_06_04 OR A1_06_05 OR A1_06_06 OR A1_06_07 OR A1_07_00 OR
A1_07_01 OR A1_07_02 OR A1_07_03 OR A1_07_04 OR A1_07_05 OR A1_07_06 OR A1_07_07;
```

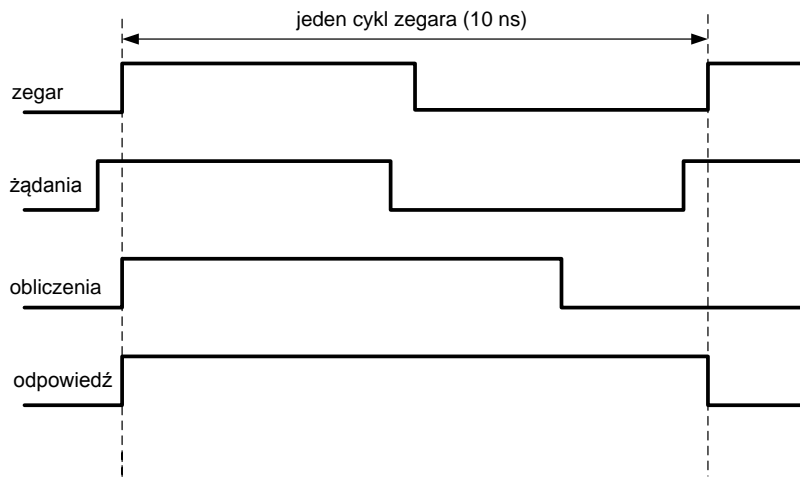
Rysunek 6.1. Jedna linia kodu VHDL realizująca wzór (3.8) dla elementu $b_1[7, 4]$ ($k=1, i=7, j=4$) – obliczanie wartości zmiennej $B1_07_04$



Rysunek 6.2. Realizacja poszczególnych kroków w czasie jednego przebiegu algorytmu

parametry zestawianego połączenia – dla połączenia (14, 15) dedykowana jest płaszczyzna $P = 6$, więc zostanie podstawiona wartość $A_{6_14_15} = 1$. Na podstawie wartości zmiennych reprezentujących komórki macierzy A ustalone są wartości zmiennych reprezentujących wartości komórek macierzy B , analogicznie, na podstawie wartości zmiennych reprezentujących wartości komórek macierzy B zostają ustalone wartości zmiennych reprezentujących elementy macierzy C . Ostatnim etapem każdego przebiegu jest uaktualnienie wartości zmiennych reprezentujących komórki macierzy D , po tym etapie w macierzy D zebrane są numery płaszczyzn dedykowanych dla wszystkich możliwych połączeń, układ jest gotowy do realizacji kolejnego przebiegu algorytmu, czyli do wskazania płaszczyzny dla kolejnego połączenia. Dla ścisłości, należy zauważyć, że rozłączenie połączenia oznacza wpisanie wartości 0 w odpowiedniej zmiennej w macierzy A oraz aktualizację wszystkich zmiennych od niej zależnych.

Na rysunku 6.3 przedstawiono charakterystykę czasową realizacji jednego przebiegu algorytmu. Układ na płycie ewaluacyjnej taktowany jest zegarem 100 MHz, czyli długość jednego cyklu wynosi 10 ns. Wynik pracy algorytmu jest dostępny już na samym początku cyklu, natomiast procedura aktualizacji danych trwa ok 8 ns. Na początku kolejnego cyklu zegara sterownik jest gotowy do realizacji kolejnego przebiegu algorytmu. Da-



Rysunek 6.3. Schemat parametrów czasowych pracy układu przy założeniu częstotliwości $f=100$ MHz

nymi wejściowymi dla przygotowanego sterownika są numery wejść i wyjść do połączenia lub rozłączenia, natomiast danymi wyjściowymi są numery płaszczyzn dla poszczególnych połączeń oraz reprezentacja aktualnego stanu pola, na podstawie której generowane są bitowe sygnały sterujące stanem poszczególnych układów przełączających.

6.2. Implementacja pozostałych algorytmów

W pracy przedstawiono również sprzętową implementację kolejnościowego algorytmu zestawiania połączeń w przypadku komutacji jednoczesnej. Z uwagi na sekwencyjne rozważanie poszczególnych połączeń, w przypadku tego algorytmu nie ma możliwości równoległego wykonywanych tylu obliczeń jak w przypadku algorytmu macierzowego. Implementacja algorytmu przestrojeń w polach o parzystej liczbie sekcji została przedstawiona w jednym z artykułów, natomiast w rozprawie została opisana implementacja algorytmu przestrojeń bazującego na grafie stanu pola. Ten algorytm działa poprawnie w obu rodzajach pól. W przypadku tej implementacji szeroko skorzystałem z możliwości równoległego wykonywania operacji przez niezależne struktury sprzętowe. W kodzie VHDL zostały przygotowane struktury definiujące wszystkie możliwe węzły grafu stanu oraz powiązania między nimi. Wpisanie wartości 1 w odpowiedniej komórce macierzy **A** oznacza istnienie danego połączenia i powoduje aktywację odpowiednich struktur obliczeniowych. Zestawienie nowego połączenia w przypadku braku blokady odbywa się na zasadach realizowanych przez algorytm macierzowy, natomiast w sytuacji, gdy dla nowego połączenia nie ma dostępnej płaszczyzny, zostaje ona odblokowana przez realizację przestrojeń wskazanych przez algorytm przestrojeń. Wybór ciągu przestrojeń odbywa się przez analizę informacji przekazywanej między sąsiednimi węzłami, natomiast realizacja wybranego ciągu przestrojeń sprowadza się do zmiany płaszczyzny realizującej połączenie reprezentowane przez węzły tworzące ciąg przestrojeń. W przypadku każdego z algorytmów, wynikiem pracy sterownika jest zestaw sygnałów sterujących poszczególnymi elementami przełączającymi tworzącymi sterowane pole.

Najważniejsze osiągnięcia pracy

W rozprawie przedstawiono algorytmy sterowania polami komutacyjnymi typu banyan. Zaproponowano nowy sposób analizy stanu pola oparty na macierzach. Bazując na tym sposobie zdefiniowano nowy algorytm sterowania polem, który osiąga niższy współczynnik strat od algorytmów dotychczas znanych. Opisano kilka sposobów implementacji tego algorytmu, w tym postać optymalną pod kątem bardzo szybkiej realizacji w układach sprzętowych. Zaproponowana implementacja została zrealizowana w układzie FPGA z serii Virtex 5 firmy Xilinx. Przeanalizowano również warunki kolejnościowego przydziału zasobów dla kolejno rozważanych połączeń. Wskazano modyfikację tego sposobu przydziału zasobów mającą na celu zredukowanie warunków nieblokowania do poziomu warunków koniecznych przestrajalności, sposób taki może być z powodzeniem wykorzystany do komutacji jednoczesnej w polach typu banyan. W pracy przedstawiono implementację takiego sterowania polem w układzie sprzętowym. Dokonano analizy stanów pola pod kątem maksymalnej liczby przestrojeń i udowodniono, że w polach o parzystej liczbie sekcji w żadnym stanie pola nie będzie konieczne więcej niż jedno przestrojenie do odblokowania dowolnego zablokowanego połączenia, natomiast w polach o nieparzystej liczbie sekcji liczba koniecznych przestrojeń nie jest większa niż trzy. Zdefiniowano graf stanu pola, na podstawie którego zaproponowano algorytm przestrojeń znajdujący ciąg przestrojeń o minimalnej długości. Przedstawiono sposób implementacji tego algorytmu w układzie sprzętowym, w tym również w wersji bez konieczności przerywania istniejących połączeń. Implementacja taka została dokonana we wspomnianym wcześniej układzie Virtex 5. Dokonano również porównania wyników symulacji współczynnika strat zaproponowanego algorytmu sterowania polem ze współczynnikiem strat osiąganym przez inne algorytmy. Wyniki tego porównania wykazują, że zaproponowany algorytm jest w każdym przypadku lepszy od znanych dotychczas.

Podczas realizacji badań na potrzeby niniejszej pracy przygotowano i wykorzystano kilka narzędzi programistycznych. Z ich złożenia powstał system o pewnym znaczeniu dydaktycznym oraz dużym znaczeniu analitycznym. Przygotowane oprogramowanie umożliwia w wygodny sposób analizowanie stanów pola pracującego pod kontrolą różnych algorytmów, demonstrację mechanizmów i zjawisk dotyczących wewnętrznych stanów pola oraz przeprowadzenie i nadzorowanie symulacji pracy pola z różnymi parametrami. Narzędzie to posiada szeroki zakres użytecznej funkcjonalności, począwszy od generowania kodu VHDL, do możliwości tworzenia rysunków (większość prezentowanych rysunków powstała przy użyciu tego narzędzia).

Podsumowując, za główne osiągnięcia niniejszej rozprawy należy uznać:

- zaproponowanie macierzowej reprezentacji stanu pola, która umożliwia analityczne określenie zależności między relacją zestawianego połączenia a blokowanymi przez nie połączeniami oraz inne właściwości stanu pola,
- zdefiniowanie macierzowego algorytmu sterowania polem i jego implementacji I_4 , która umożliwia natychmiastowe uzyskanie wyniku algorytmu,
- zdefiniowanie grafu stanu pola, który jest podstawą efektywnego algorytmu przestrojeń,
- przygotowanie i uruchomienie sterowników realizujących zaproponowane algorytmy,

przy czym bez wątpienia najważniejszym osiągnięciem jest sposób reprezentacji stanu pola, gdyż to on jest podstawą pozostałych przedstawianych mechanizmów.

8

Publikacje

Badania prowadzone pod kątem przygotowanej rozprawy dotyczyły kilku niezależnych aspektów. Efekty poszczególnych etapów prac stały się podstawą wielu publikacji, których jestem autorem lub współautorem. Dwie z nich to artykuły w bardzo prestiżowym czasopiśmie naukowym z zakresu telekomunikacji (lista filadelfijska):

1. Kabaciński W., Michalski M.: „The routing algorithm and wide-sense nonblocking conditions for multiplane baseline switching networks”, *IEEE Journal on Selected Areas in Communications*, vol. 24, nr 12, str. 35–44, 2006.
2. Danilewicz G., Kabaciński W., Michalski M., Żal M.: „A New Control Algorithm For Wide-Sense Nonblocking Multiplane Photonic Banyan-Type Switching Fabrics With Zero Crosstalk”, *IEEE Journal on Selected Areas in Communications*, vol 26, nr 3, part 2, str. 54 - 64, 2008.

Wyniki moich badań zostały zaprezentowane na licznych renomowanych międzynarodowych konferencjach z zakresu telekomunikacji:

3. Kabaciński W., Michalski M.: „Simultaneous Connections Routing in *Multi – Log₂N* Switching Fabrics”, Workshop on High Performance Switching and Routing (HPSR 2004), Phoenix, Arizona, USA, str. 214-218, 19-21 kwietnia 2004.
4. Kabaciński W., Michalski M.: „Wide-sense Nonblocking $Log_2(N, 0, p)$ Switching Networks with Even Number of Stages”, IEEE International Conference on Communications (ICC 2005), Seul, Korea, 16-20 maja 2005.
5. Danilewicz G., Kabaciński W., Michalski M., Żal M.: „Wide-Sense Nonblocking Multiplane Banyan-Type Switching Fabrics With Zero Crosstalk”, IEEE International Conference on Communications (ICC 2006), Istambuł, Turcja, 11-15 czerwca 2006.
6. Danilewicz G., Kabaciński W., Michalski M., Żal M.: „Wide-Sense Nonblocking Multiplane Baseline Switching Networks Composed of $d \times d$ Switches”, IEEE International Conference on Communications (ICC 2007), Glasgow, Wielka Brytania, 24-28 czerwca 2007.

7. Kabaciński W., Michalski M., Pattavina A.: „The Control Algorithm and WSNB Operation of $\text{Log}_2(N, 1, p)$ Switching Fabrics”, IEEE GLOBECOM 2007, Washington, DC, USA, str. 2374-2378, 23-27 listopada 2007.
8. Kabaciński W., Kleban J., Michalski M., Żal M., Pattavina A., Maier G.: „Rearranging Algorithms for $\text{Log}_2(N, 0, p)$ Switching Networks with Even Number of Stages”, International Workshop on High Performance Switching and Routing (HPSR 2009), Paryż, Francja 22 - 24 czerwca 2009.
9. Kabaciński, W. Michalski, M.: „The FPGA implementation of the $\text{Log}_2(N, 0, p)$ switching fabric control algorithm”, 2010 International Conference on High Performance Switching and Routing (HPSR 2010), Dallas, USA, 13-16 czerwca 2010.
10. Kabaciński W., Michalski M.: „The Algorithm for Rearrangements in the $\text{Log}_2(N, 0, p)$ Fabrics with an Odd Number of Stages”, IEEE International Conference on Communications (ICC 2011), Kioto, Japonia, 5-9 czerwca 2011.
11. Kabaciński, W. Michalski, M.: „The FPGA Controller for the Rearrangeable $\text{Log}_2(N, 0, p)$ Fabrics with an Even Number of Stages”, IEEE International Conference on High Performance Switching and Routing (HPSR 2011), 4-6 lipca 2011.

W moim dorobku naukowym znajduje się jeszcze 14 innych publikacji związanych z tematyką rozprawy doktorskiej oraz jedno zgłoszenie patentowe. Ponadto jestem autorem lub współautorem 11 publikacji niezwiązanych z tematyką doktoratu.