

POLITECHNIKA POZNAŃSKA

WYDZIAŁ ELEKTRONIKI I TELEKOMUNIKACJI

KATEDRA TELEKOMUNIKACJI MULTIMEDIALNEJ

I MIKROELEKTRONIKI

Autoreferat pracy doktorskiej

**Architecture and protocols for networks-on-chip
implemented in FPGA devices
(Architektura i protokoły dla sieci w mikroukładzie
realizowanej w bezpośrednio programowalnych macierzach
bramek)**

Jakub Siast

Promotor: Prof. dr hab. inż. Marek Domański

Poznań, 2019

Spis treści

Spis treści.....	2
Rozdział 1. Wprowadzenie	3
1.1. Zakres rozprawy	3
1.2. Cele i tezy rozprawy	5
Rozdział 2. Zaproponowana architektura RingNet	7
2.1. Podstawowe założenia architektury RingNet	8
2.2. Protokół sieci RingNet.....	12
2.3. Implementacja sieci RingNet.....	13
Rozdział 3. Wyniki symulacji RingNet.....	14
Rozdział 4. Wyniki syntezy RingNet.....	19
Rozdział 5. Porównanie implementacji pierścieni RingNet i AXI4 Interconnect	23
Rozdział 6. Podsumowanie	26
6.1. Osiągnięcia rozprawy	27
6.2. Wnioski.....	30
WYBRANE POZYCJE SPISU LITERATURY	31
DORÓBEK PUBLIKACYJNY AUTORA.....	35
Cytowania prac autora	35
Publikacje autora.....	35
Czasopisma o zasięgu międzynarodowym	35
Polskie czasopisma	36
Materiały konferencyjne	36
Patenty	38

Rozdział 1. Wprowadzenie

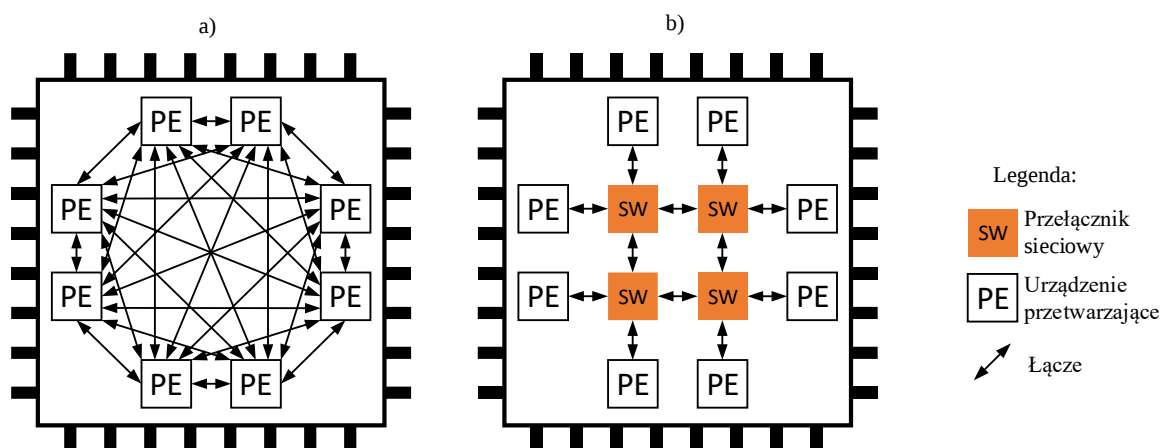
1.1. Zakres rozprawy

Pierwszy układ scalony został opracowany w 1958 roku i składał się z pojedynczego tranzystora i pasywnych elementów [Kil00]. Od tego czasu liczba tranzystorów w układach scalonych zwiększa się, co doprowadziło do opracowania układów o wielkiej skali integracji (ang. *very-large-scale integration*, VLSI). W największych układach scalonych realizuje się całe systemy przetwarzające, tzw. systemy w mikroukładzie (ang. *systems-on-chip*, SoCs) złożone z wielu urządzeń przetwarzających (ang. *processing elements*, PEs).

Skomunikowanie urządzeń przetwarzających składających się na system jest jednym z istotnych zagadnień badawczych. Na wczesnych etapach rozwoju układów scalonych wykorzystywano do tego celu bezpośrednie łącza wewnątrz mikroukładu (rys. 1a). Niemniej jednak, wraz ze wzrostem złożoności układów scalonych, realizacja łączy bezpośrednich staje się coraz trudniejsza a nawet niemożliwa. Jest to związane z niekorzystną zmianą niektórych parametrów elektrycznych ścieżek przewodzących tworzących łącza w mikroukładach. Mianowicie, przy coraz gęstszym upakowaniu elementów w mikroukładzie, w tym ścieżek przewodzących, rośnie pojemność elektryczna pomiędzy ścieżkami przewodzącymi, dodatkowo zwężenie ścieżek przewodzących powoduje wzrost ich rezystancji. W takiej sytuacji zwiększa się czas potrzebny na zmianę napięcia na ścieżkach przewodzących, czyli czas potrzebny na transmisję sygnału przez łącze. Ponadto, pogorszeniu parametrów elektrycznych ścieżek przewodzących towarzyszy zwiększenie częstotliwości pracy tranzystorów w nowoczesnych mikroukładach, a co za tym idzie zmniejsza się czas przeznaczony na zmianę napięcia na ścieżce przewodzącej.

W nowoczesnych układach scalonych ogranicza się długość ścieżek przewodzących, dzięki czemu poprawia się ich parametry elektryczne. Krótsze ścieżki przewodzące o lepszych parametrach elektrycznych pozwalają na tworzenie łączy pracujących z większymi częstotliwościami. Długość łącza często jest ograniczana do wartości wielokrotnie mniejszych niż gabaryty mikroukładu [Dal99], [Muk17]. Wymusza to rezygnację z bezpośredniego łączenia urządzeń przetwarzających w mikroukładzie. Zamiast pojedynczych łączy bezpośrednich zaczęto wykorzystywać większą liczbę krótszych łączy, poprzedzielanych elementami synchronicznymi. Propagacja sygnału przez kolejne łącza zajmuje łącznie kilka cykli sygnału zegarowego. Ostatecznie, pod koniec zeszłego wieku, ewolucja

zaprezentowanego podejścia doprowadziło to do opracowania sieci w mikroukładzie (ang. *network-on-chip*, NoC) [Dal99], [Dal01], [Ben02] (rys. 1b). Sieć w mikroukładzie, oprócz umożliwienia pracy z większą częstotliwością niż częstotliwości osiągalne dla łączy bezpośrednich, cechuje się również wyższym wykorzystaniem przepustowości łączy, ujednoliconym interfejsem, który ułatwia opracowanie, rozwój i testowanie funkcjonalne wykorzystujących ją systemów.



Rys. 1. System w mikroukładzie skomunikowany a) łączyami bezpośrednim b) siecią w mikroukładzie.

Większość dotychczasowych badań w zakresie NoC dotyczyła specjalizowanych układów scalonych (ang. *application specific integrated circuits*, ASICs). Niemniej jednak szybko rosnąca skala integracji układów programowalnych sprawiła, że i w nich zaczęto stosować NoC [Łucz11], [Pap12], [She14], [Ret14], [Mai15], [Pap15], [Kap15], [Was17], [Kap17a], [Kap17b], [Kap17c], [Vip17], [Mai17], [Sid18], [Ahm18], [Red19]. Pomimo że mikroukłady FPGA różnią się od ASIC, większość znanych NoC została zaczerpnięta z ASIC bez uwzględnienia specyfiki architektury FPGA. Z tego względu, potencjał NoC w mikroukładach FPGA nie został w pełni wykorzystany. W tej sytuacji inne architektury komunikacyjne takie jak przełączniki krzyżowe (np. AXI4 Interconnect) są wciąż używane, pomimo ich ograniczonej skalowalności [Mai15]. Opracowanie nowej architektury NoC, lepiej dopasowanej do architektury FPGA, stanowi wciąż istotne wyzwanie.

Rosnąca liczba zasobów mikroukładów FPGA sprawia, że realizuje się w nich coraz bardziej złożone zadania, w tym przetwarzanie danych multimedialnych, np.: filtrację obrazów medycznych [Lic18], kodowanie wizji techniką *Multiview Video Coding* (MVC) [Stę10b], kodowanie wizji techniką *High Efficiency Video Coding* (HEVC) [Buk17], estymację

rozbieżności na potrzeby stereowizji [Dom15], [Tto16], detekcję cech w sekwencjach wizyjnych [Cha15], kodowanie kanałowe multimedialnych na potrzeby transmisji [Bre17], kształtowanie wiązki akustycznej i syntezę pola dźwięku [The11]. Każde z powyższych przykładowych zadań, ze względu na swoją złożoność, jest implementowane jako system w mikroukładzie. W systemie w mikroukładzie złożone zadanie dzielone jest na podzadania realizowane w osobnych urządzeniach przetwarzających. Wydajność powyższych systemów przetwarzania danych multimedialnych zależy, między innymi, od infrastruktury komunikacyjnej zastosowanej do skomunikowania urządzeń przetwarzających. Wydajność systemów w mikroukładzie może zostać zwiększona w wyniku użycia NoC efektywnie wykorzystującej zasoby architektury mikroukładów FPGA.

W rozprawie poruszono zagadnienie NoC w kontekście właściwości typowych mikroukładów FPGA oferowanych przez czołowych producentów. Na podstawie bogatego własnego doświadczenia w przetwarzaniu danych multimedialnych ([Dom11a], [Dom11b], [Dom12a]–[Dom12c], [Sia12], [Stanki12a]–[Stanki12c], [Stanko12], [Weg12a]–[Weg12f], [Dom13a], [Dom13b], [Sia13], [Dom14a]–[Dom14c], [Łucz14], [Sia14], [Stanko14], [Stanko15], [Sia16]) oraz, w oparciu o wcześniejsze doświadczenie w opracowaniu implementacji dla mikroukładów FPGA [Łucz10], [Stę10a], [Stę10b], [Łucz11] autor rozpoczął prace nad uniwersalną, dedykowaną architekturze FPGA siecią w mikroukładzie, odpowiednią dla zastosowań multimedialnych. Wynikiem podjętych prac jest architektura sieci, którą nazwano RingNet oraz odpowiedni protokół komunikacyjny. **Osiągnięcia pracy zostały podsumowane w artykule „RingNet: A Memory-Oriented Network-On-Chip Designed for FPGA,” opublikowanym w czasopiśmie IEEE Transactions on Very Large Scale Integrated (VLSI) Systems w 2019 roku [Sia19].** Publikacja uzupełniona jest o 78 stron materiałów dodatkowych dostępnych w serwisie IEEE *Xplore*, w tym wyniki symulacji i opis implementacji elementów sieciowych.

1.2. Cele i tezy rozprawy

Głównym celem rozprawy jest opracowanie uniwersalnej architektury NoC dopasowanej do różnych mikroukładów FPGA oraz odpowiedniego protokołu komunikacyjnego. Opracowana architektura NoC powinna cechować się takimi właściwościami jak regulowana przepustowość i sprawiedliwy dostęp do sieci. Sprawiedliwy dostęp do sieci oznacza, że urządzenia przetwarzające otrzymują część przepustowości sieci

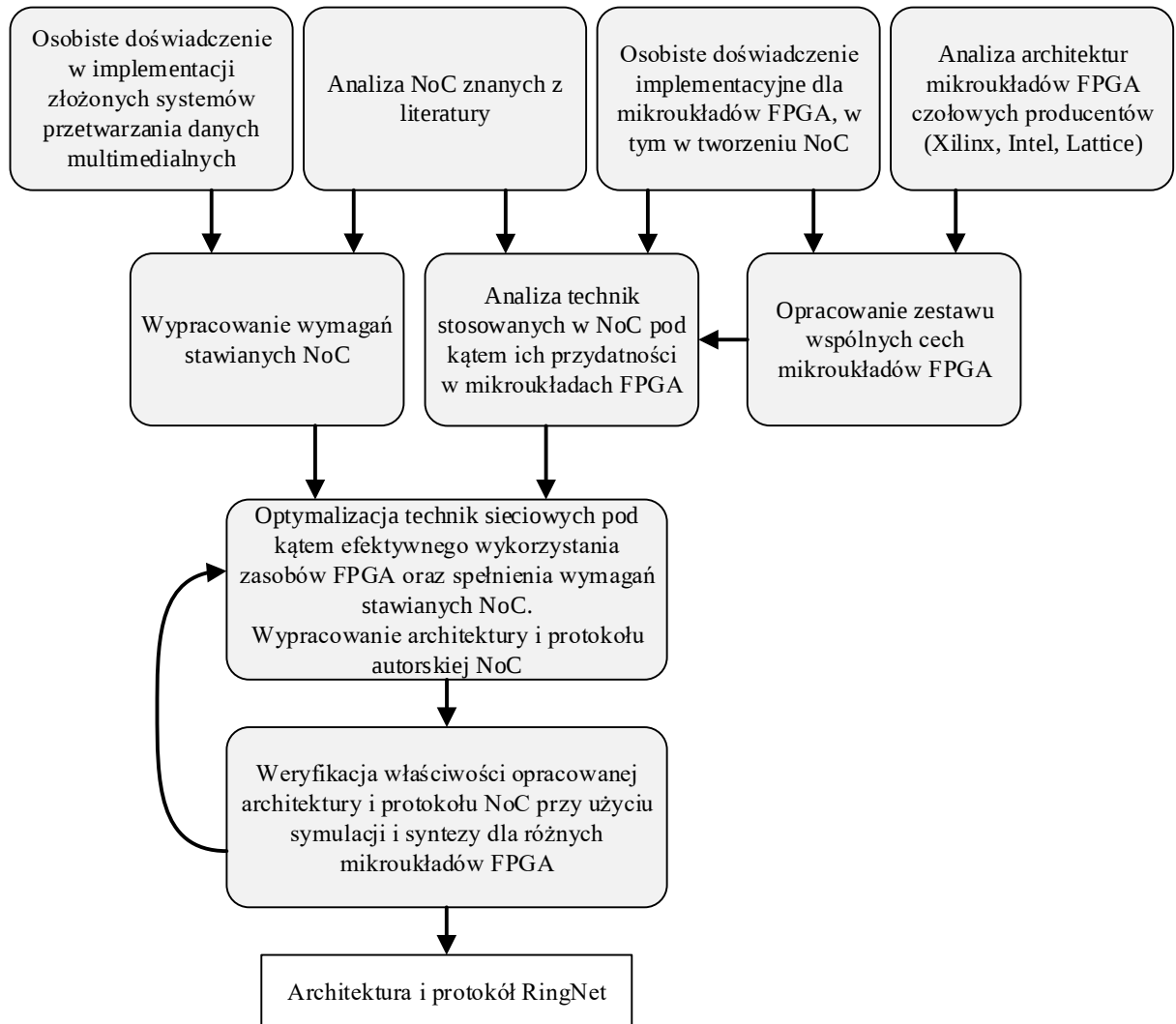
proporcjonalną do zgłoszonych żądań, oraz że wartości opóźnień, z jakimi pakiety od różnych urządzeń przetwarzających są przesyłane przez sieć, są podobne.

Tezy rozprawy zostały sformułowane następująco:

- T1) Możliwe jest opracowanie architektury i protokołu sieci w mikroukładzie charakteryzującej się regulowaną przepustowością, sprawiedliwym dostępem do sieci oraz maksymalną częstotliwością taktowania większą niż 90% maksymalnej częstotliwości taktowania zasobów sprzętowych FPGA, dla mikroukładów FPGA głównych producentów.*
- T2) Możliwe jest opracowanie architektury i protokołu sieci w mikroukładzie charakteryzującej się regulowaną przepustowością oraz sprawiedliwym dostępem do sieci, dedykowanej FPGA, która użyje mniej zasobów oraz będzie charakteryzowała się większą maksymalną częstotliwością taktowania niż przełącznik krzyżowy będący stanem techniki (AXI4 Interconnect).*

Rozdział 2. Zaproponowana architektura RingNet

Prace nad architekturą i protokołem sieci w mikroukładzie, spełniającej wymagania opisane w tezach rozprawy, przeprowadzono w oparciu o metodologię przedstawioną na rys. 2.



Rys. 2. Metodologia pracy.

W pracy, na bazie analizy NoC znanych z literatury, w szczególności rozwiązań dedykowanych mikroukładowi FPGA ([Pap12], [She14], [Ret14], [Abd14], [Xil15], [Mai15], [Pap15], [Kap15], [Abd16a], [Abd16b], [LiuT16], [Hud16], [Abd17], [Was17], [Kap17a], [Kap17b], [Kap17c], [Vip17], [Mai17], [Sid18], [Ahm18], [Red19]) oraz doświadczenia w implementacji złożonych systemów multimedialnych ([Łucz08], [Dom13], [Dom14]) wypracowano zestaw wymagań stawianych NoC. Przeanalizowano cechy mikroukładów

FPGA głównych producentów, użytecznych z punktu widzenia implementacji SoC ([Alt11]–[Alt16], [Int16], [Int17], [Xil16a]–[Xil17], [Lat13]–[Lat16]). Na bazie przeprowadzonej analizy oraz w oparciu o własne doświadczenie implementacyjne związane z NoC ([Stę10a], [Stę10b], [Łucz10], [Łucz11]), opracowano zestaw wspólnych cech mikroukładów FPGA istotnych przy implementacji NoC. Następnie, na bazie wyników wcześniejszych etapów, przeanalizowano znane techniki sieciowe pod kątem ich przydatności w analizowanych architekturach FPGA. Na bazie przeanalizowanych technik oraz z uwzględnieniem wypracowanych wymagań stawianych NoC, autor zaproponował architekturę NoC wraz z protokołem, zwaną RingNet. Spełnienie wymagań stawianych NoC, oraz efektywność wykorzystania zasobów FPGA zostały sprawdzone z użyciem symulacji oraz w procesie syntezy dla mikroukładów FPGA różnych architektur.

2.1. Podstawowe założenia architektury RingNet

Podstawowymi cechami RingNet są:

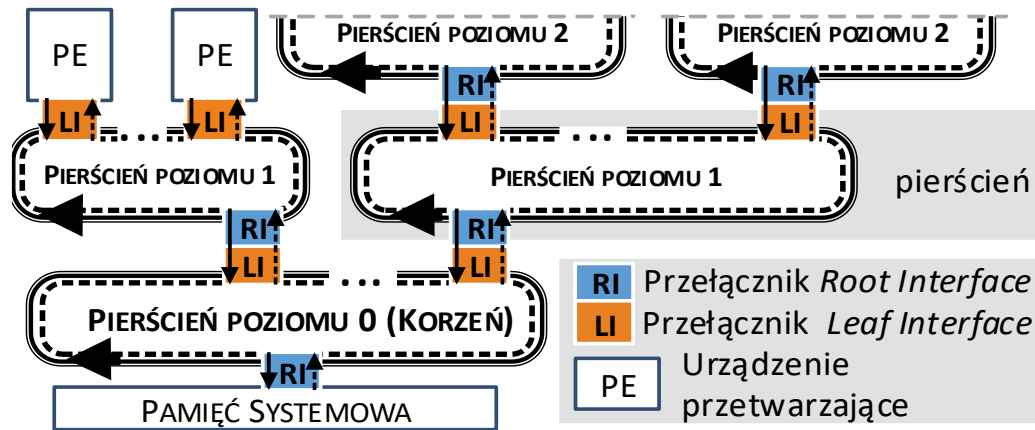
- komunikacja prowadzona wyłącznie poprzez tak zwaną *pamięć systemową* (podłączoną do mikroukładu FPGA pamięć SDRAM lub wbudowaną pamięć blokową);
- sterowanie ruchem w sieci przez podłączone do sieci urządzenia przetwarzające (ang. *processing elements*);
- wykorzystanie zoptymalizowanych pod mikroukłady FPGA trójbramowych przełączniki sieciowych zorganizowanych w topologię drzewa pierścieni;
- wykorzystanie pamięci rozproszonej (ang. *look-up-table-based random-access memory*, LUTRAM) do implementacji małych buforów w przełącznikach;
- zastosowanie przełączania *virtual cut-through*.

Komunikacja prowadzona poprzez pamięć systemową, w formie zaproponowanej dla RingNet, pozwala urządzeniom przetwarzającym regulować wielkości docierających do nich strumieni pakietów. Taka regulacja pozwoliła na wyeliminowania zatorów związanych z przepełnieniem buforów odbiorczych urządzeń przetwarzających. Brak zatorów pozwolił na

wykorzystanie buforów sieciowych o małej pojemności, które efektywnie wykorzystują szeroko dostępną pamięć rozproszoną (LUTRAM).

Zastosowanie przełączania *virtual cut-through* jest rekomendowane w pracach [Mai15], [Pap12] dla mikroukładów FPGA firmy Xilinx. Podstawą do takiej rekomendacji jest efektywne wykorzystanie pamięci rozproszonych przez przełączniki *virtual cut-through*. Autor przeprowadził analizę mikroukładów FPGA głównych producentów, z której wynika powszechna dostępność pamięci rozproszonych. W oparciu o przeprowadzoną analizę, autor uogólnił wniosek o zasadności użycia przełączania *virtual cut-through*, na sieci implementowane w mikroukładach innych głównych producentów mikroukładów FPGA. Autor zastosował przełączniki *virtual cut-through* w RingNet. Na podstawie przeprowadzonej analizy mikroukładów FPGA oraz w oparciu o prace [Mai15], [Pap12], autor postuluje wykorzystanie buforów sieciowych o głębokości nieprzekraczającej głębokość pamięci rozproszonych dostępnych w analizowanych mikroukładach FPGA. Takie ograniczenie ma na celu optymalne wykorzystanie pamięci rozproszonych. W celu zapewnienia poprawnego działania mechanizmu przełączania *virtual cut-through* autor postuluje ograniczenie liczby flitów (w sieciach implementowanych w układach FPGA, flit to część pakietu przesyłana w jednym takcie zegara pomiędzy przełącznikami, od ang. *flow control unit*) w pakietach do wartości mniejszej niż głębokość zastosowanych buforów sieciowych.

W [Abd16a] i [Pap12] wykazano wysoką efektywność wykorzystania zasobów mikroukładów FPGA w przypadku sieci NoC implementowanych w topologii pierścienia. Ponadto, jak wynika z [Pap15], zastosowanie przełączników o małej liczbie wejść pozwala na uzyskanie dużych maksymalnych częstotliwości taktowania sieci. Na podstawie wyników powyższych prac, autor zaproponował dla RingNet oryginalną topologię drzewa pierścieni (rys. 3) z *pamięcią systemową* podłączoną do pierścienia znajdującego się w korzeniu drzewa, oraz z *urządzeniami przetwarzającymi* (PE) podłączonymi do pierścieni wyższych warstw drzewa. Autor zastosował topologię wielu pierścieni połączonych w drzewo w celu ograniczenia opóźnień czystej topologii pierścienia. Zastosowana topologia drzewa pierścieni używa prostych trójbramowych przełączników zwanych *Leaf Interface* (LI) oraz *Root Interface* (RI). W prezentowanej topologii istnieje tylko jedna droga pomiędzy urządzeniem przetwarzającym a korzeniem drzewa, dlatego możliwe jest zastosowanie w RingNet trasowania statycznego, co upraszcza logikę sterującą i pozwala na uzyskanie dużych maksymalnych częstotliwości taktowania sieci.



Rys. 3. Topologia RingNet.

Wymaganą cechą sieci jest możliwość sterowania jej przepustowością. Przepustowość pojedynczego pierścienia RingNet jest funkcją szerokości flit oraz częstotliwości sygnału zegarowego. Ponieważ pierwszy z tych dwóch parametrów, czyli szerokość flit w RingNet jest stała, dlatego do sterowania przepustowością pierścienia wykorzystuje się sterowanie częstotliwością sygnału zegarowego. Niemniej jednak maksymalna częstotliwość sygnału zegarowego dla danego mikroukładu FPGA jest ograniczona co ogranicza również maksymalną przepustowość pierścienia RingNet. Ponadto, wybrana topologia ogranicza maksymalną przepustowość sieci. Ma to związek z agregacją ruchu w pierścieniu znajdującym się w korzeniu drzewa i w szczególności przepustowość tego pierścienia może ograniczać przepustowość całej sieci RingNet. W literaturze znane jest rozwiązanie problemu ograniczonej przepustowości wynikającej z agregacji ruchu w sieciach z topologią drzewa poprzez fizyczne zwielokrotnienie łączy i przełączników w korzeniu drzewa [Lei85], [Mat09], [Add17]. **Biorąc pod uwagę rozwiązania znane z literatury, w celu zapewnienia regulacji przepustowości sieci, autor umożliwił fizyczne zwielokrotnienie pierścieni w architekturze RingNet, tj. zamianę dowolnego pierścienia (patrz rys. 3) na dowolną liczbę równolegle połączonych pierścieni (szczegółowo opisane w rozprawie). Pomiędzy równolegle połączone pierścienie, ruch jest rozdzielany równomiernie, co skutkuje zwiększeniem przepustowości. W RingNet ruch agreguje się w pierścieniu znajdującym się w korzeniu drzewa i ten pierścień również może zostać zwielokrotniony, aby dostosować jego przepustowość do przepustowości podłączonej pamięci *Synchronous Dynamic RAM* (SDRAM).**

W publikacjach podkreślany jest duży udział ruchu sieciowego przypadający na komunikację z pamięcią SDRAM w systemach przetwarzających dane multimedialne [Abd16a], sięgający 50% całości ruchu w sieci [Ber04]. Dla wydajności systemu, istotne jest, aby umożliwić pełne wykorzystanie przepustowości pamięci SDRAM. Na podstawie analizy dostępnych mikroukładów FPGA ustalono szerokość flit w architekturze RingNet w celu dopasowania przepustowości RingNet do przepustowości interfejsów urządzeń SDRAM wspieranych w analizowanych FPGA. Flit RingNet, w implementacji opisanej w rozprawie to osiem bajtów danych przesyłanych wraz z ośmioma jednobitowymi flagami informującymi o ważności bajtów danych. Taka szerokość flit umożliwia uzyskanie większych przepustowości RingNet niż przepustowości interfejsów pamięci SDRAM rozważanych mikroukładów FPGA. Poniżej podano mikroukłady FPGA wyposażone w interfejsy do pamięci SDRAM charakteryzujące się skrajnymi wartościami maksymalnej przepustowości spośród wszystkich analizowanych mikroukładów FPGA:

- Wśród rozważanych mikroukładów FPGA, mikroukład LatticeECP2 charakteryzuje się najmniejszą maksymalną przepustowością interfejsu pamięci SDRAM wynoszącą 4,3 Gb/s. Jednocześnie, RingNet, taktowany zegarem o umiarkowanej (jak na mikroukład LatticeECP2) częstotliwości 75 MHz, pozwala na osiągnięcie przepustowości 7 Gb/s.
- W rodzinie produktów UltraScale+ firmy Xilinx i tzw. serii 10 firmy Intel możliwe jest uzyskanie przepustowości interfejsu pamięci SDRAM do 192 Gb/s. Jest to największa przepustowość interfejsu do pamięci SDRAM wśród analizowanych mikroukładów FPGA. Dla porównania, cztery pierścienie RingNet pozwalają na transfer 196 Gb/s przy częstotliwości taktowania równej 525 MHz, która jest częstotliwością taktowania łatwo osiągalną dla mikroukładów podanych rodzin.

W sieci RingNet, rola pamięci SDRAM jest szczególna za względu na jej funkcję punktu wymiany danych pomiędzy urządzeniami przetwarzającymi. Dla wydajności sieci RingNet oraz dla systemów budowanych w oparciu o tę sieć, istotne jest, aby umożliwić pełne wykorzystanie przepustowości pamięci SDRAM. W związku z powyższym, **uzyskanie wyższych przepustowości sieci niż przepustowości interfejsów pamięci SDRAM, w które są wyposażone analizowane mikroukłady FPGA jest istotną cechą RingNet.**

2.2. Protokół sieci RingNet

Autor opracował protokół dla sieci RingNet zapewniający komunikację pomiędzy podłączonymi urządzeniami przetwarzającymi. **Komunikacja zgodna z opracowanym protokołem cechuje się sprawiedliwym dostępem do sieci**, tj. urządzenia przetwarzające otrzymują część przepustowości sieci proporcjonalną do zgłoszonych żądań oraz wartości opóźnień, z jakimi pakiety od różnych urządzeń przetwarzających są przesyłane przez sieć, są podobne. Ponadto, opracowany protokół umożliwia przydzielenie pakietom różnych priorytetów. Protokół obejmuje cztery warstwy modelu OSI, od warstwy łącza danych do warstwy sesji.

Warstwa łącza danych definiuje strukturę pakietów, w szczególności definiuje długość pakietów. Z literatury [Mai15], [Pap12] oraz z własnej analizy mikroukładów FPGA wynika, że ograniczenie długości pakietów do maksymalnie N flitów, gdzie N jest mniejsze niż głębokość pamięci rozproszonych, umożliwia efektywne wykorzystanie zasobów rozważanych mikroukładów FPGA. W pracy przyjęto $N = 32$. Opracowany protokół uwzględnia te ograniczenia. Druga warstwa protokołu komunikacyjnego, warstwa sieciowa, definiuje adresację w sieci RingNet. Zastosowana adresacja bierze pod uwagę podstawowe założenie architektury RingNet, czyli komunikację urządzeń przetwarzających z wykorzystaniem pamięci systemowej. Z tego względu, w opracowanym protokole stosuje się dwa rodzaje adresów. Pierwszy rodzaj adresu, adres pamięciowy, umożliwia identyfikację obszaru pamięci używanej do wymiany danych pomiędzy urządzeniami przetwarzającymi. Drugi rodzaj adresu identyfikuje łącza, które musi przebyć pakiet na drodze od pamięci systemowej do urządzenia przetwarzającego. Opracowany protokół umożliwia zaadresowanie do 128 GB pamięci i do 759 375 urządzeń przetwarzających. Kolejna warstwa protokołu komunikacyjnego, warstwa transportowa, definiuje kanały logiczne sieci RingNet. Cechą przygotowanego protokołu jest rozdzielenie logicznego kanału służącego do zapisu danych do pamięci systemowej i logicznego kanału służącego do odczytu danych z pamięci systemowej. Ostatnia zdefiniowana warstwa protokołu komunikacyjnego, warstwa sieciowa, definiuje sposób informowania urządzeń przetwarzających o danych oczekujących na przetworzenie w pamięci systemowej. Opracowany protokół, szczegółowo opisany w rozprawie, ma na celu zapewnienie założonych funkcjonalności, takich jak komunikacja pośrednia z wykorzystaniem pamięci systemowej i sprawiedliwy dostęp do sieci.

2.3. Implementacja sieci RingNet

W celu oceny zaproponowanej architektury autor zaimplementował elementy sieciowe RingNet w języku opisu sprzętu Verilog. Aby wykazać, że przepustowość RingNet może być regulowana, co jest postulowane w tezach rozprawy, autor wyznaczył teoretyczną wartość przepustowości sieci dla przygotowanej implementacji. Cechą przygotowanej implementacji RingNet jest rozdzielenie dwóch typów ruchu. Pierwszy typ ruchu przenosi dane od urządzeń przetwarzających podłączonych w liściach drzewa sieciowego (por. rys. 3) do pamięci systemowej podłączonej w korzeniu i odbywa się w tak zwanym logicznym kanale zapisu. Drugi typ ruchu przenosi dane w kierunku przeciwnym, tj. od pamięci systemowej do urządzeń przetwarzających, i odbywa się w tak zwanym logicznym kanale odczytu. Teoretyczna przepustowość, wyrażona w bitach na takt sygnału zegarowego, spodziewana w obu logicznych kanałach opisana jest wzorem:

$$T_{RW_MAX} = R \cdot 64 \cdot \frac{8}{11} [b/takt], \quad (1)$$

gdzie 64 to liczba bitów danych we flit, czynnik $\frac{8}{11}$ to udział flitów danych w ogólnej liczbie przesyłanych flitów (wynika z zastosowanego protokołu komunikacji), a R jest to współczynnik zwielokrotnienia pierścienia znajdującego się w korzeniu drzewa sieciowego.

Rozdział 3. Wyniki symulacji RingNet

Zaproponowana architektura RingNet została zasymulowana w celu określenia jej przepustowości i wprowadzanych opóźnień. Ponadto, przebadane zostały parametry opisujące niezawodność sieci, w szczególności sprawiedliwość w dostępie do sieci.

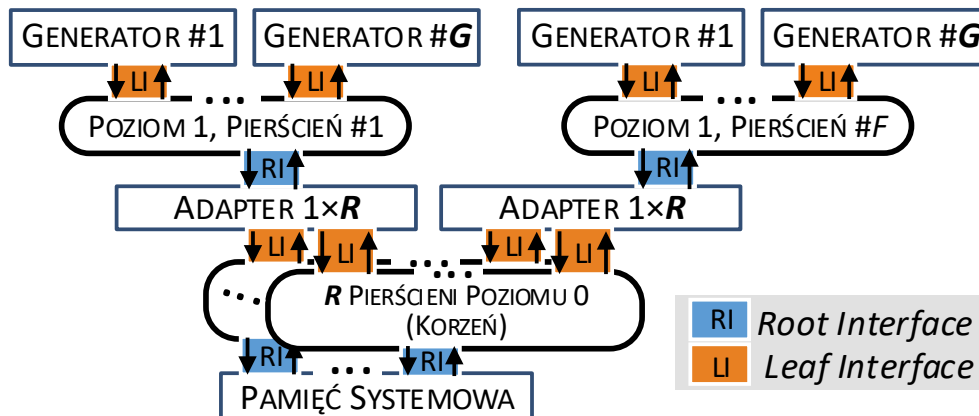
W symulacjach użyto modelu pamięci systemowej o nieograniczonej przepustowości i zerowym opóźnieniu, dzięki czemu zastosowany model pamięci systemowej nie wpłynął na wyniki wydajności sieci.

Urządzenia przetwarzające zasymulowano, używając generatorów pakietów. Każdy generator niezależnie nadaje pakiety w logicznym kanale odczytu i logicznym kanale zapisu z regulowanym średnim opóźnieniem D_{sr} pomiędzy kolejnymi pakietami:

$$D_{sr} = \frac{N \cdot 64 \cdot 8}{11 \cdot L \cdot T_{RW_MAX}}, \quad (2)$$

gdzie T_{RW_MAX} jest teoretyczną przepustowością kanałów logicznych, obliczoną według wzoru (1), L jest zagregowanym obciążeniem, generowanym przez wszystkie generatory, wyrażonym jako procent teoretycznej przepustowości T_{RW_MAX} , N jest liczbą podłączonych generatorów. Faktyczna wartość opóźnienia pomiędzy nadaniem kolejnych pakietów jest wynikiem działania generatora liczb pseudolosowych o rozkładzie równomiernym w zakresie $\mathcal{U}\{0.8D_{sr}, 1.2D_{sr}\}$.

W eksperymentach zasymulowano RingNet z dwoma poziomami pierścieni (rys. 4).



Rys. 4. Topologia RingNet wykorzystana w symulacjach.

Rozmiar sieci regulowano za pomocą parametrów:

- R : Współczynnik zrównoleglenia pierścienia znajdującego się w korzeniu drzewa,
- F : Liczba pierścieni pierwszego poziomu,
- G : Liczba generatorów podłączonych do pojedynczego pierścienia pierwszego poziomu.

Opóźnienie zmierzono wyznaczając liczbę cykli zegarowych, które upłynęły pomiędzy nadaniem nowego pakietu z generatora pakietów a otrzymaniem przez ten generator potwierdzenia dotarcia pakietu do pamięci systemowej. Liczba podłączonych generatorów N jest równa $F \cdot G$. Maksymalna zasymulowana liczba urządzeń podłączonych do sieci została ograniczona do 75 w związku z ograniczoną pojemnością rozważanych mikroukładów FPGA.

Wyniki symulacji zebrano dla sieci będącej w stanie ustalonym zgodnie z metodologią przedstawioną w publikacji [Dal03] będącej jedną z licznie cytowanych prac w dziedzinie.

Sieć opisaną w języku Verilog przetworzono z wykorzystaniem kompilatora Verilator [Sny17], a wynikowy kod C++ został skompilowany z wykorzystaniem kompilatora GCC, następnie, skompilowany kod został wykonany w celu przeprowadzenia symulacji funkcjonalnej. **W pracy autor przedstawił wyniki symulacji dla ponad 3000 różnych konfiguracji rozmiaru sieci i obciążeń.**

W tabeli I przedstawiono średnie opóźnienie dla logicznego kanału odczytu, w sytuacji dużego obciążenia sieci (92% to 97%) oraz różnych rozmiarów sieci ($R=1$, $F=1-5 \times G=1-15$). Można zauważyć, że opóźnienie rośnie wraz ze zwiększeniem liczby pierścieni pierwszego poziomu oraz z liczbą generatorów podłączonych do każdego pierścienia pierwszego poziomu.

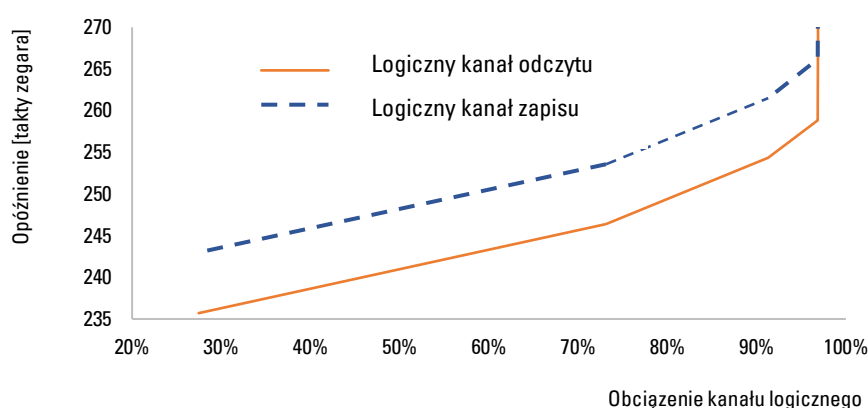
TABELA I

ŚREDNIE OPÓŹNIENIE (WYRAŻONE W CYKLACH ZEGARA) DLA KANAŁU ODCZYTU, DLA JEDNEGO PIERŚCIENIA W KORZENIU SIECI ORAZ OBCIĄŻENIA KANAŁÓW ODCZYTU I ZAPISU W ZAKRESIE 92% – 97%

		Liczba generatorów podłączonych do pierścienia pierwszego poziomu (G)					
		1	2	3	4	7	15
Liczba pierścieni pierwszego poziomu (F)	1	95	118	120	122	147	194
	2	113	134	142	145	182	225
	3	129	148	155	161	185	241
	4	130	151	157	163	189	245
	5	140	163	169	175	201	258

Dla kanału zapisu, zmierzone opóźnienia są średnio większe o 7 taktów, niż te przedstawione w tabeli I. Dla większej liczby równoległych pierścieni użytych w korzeniu sieci ($R=2-4$), zauważono zwiększenie opóźnienia o 6 cykli zegarowych przypadający na każdy dodatkowy pierścień.

Sieci w mikroukładzie, często są opisywane za pomocą krzywych przedstawiających opóźnienie w funkcji obciążenia [Son03], [Ber04], [Yoo13], [Pos13], [LiuS14], [Hel15], [Pap15], [Kap15], [Kum16]. Na rys. 5, przedstawiono taką krzywą dla RingNet z pięcioma pierścieniami pierwszego poziomu ($F=5$), piętnastoma generatorami podłączonymi do każdego z tych pierścieni ($G=15$) oraz czterema pierścieniami użytymi w korzeniu drzewa ($R=4$).



Rys. 5. Opóźnienie w funkcji obciążenia dla RingNet z podłączonymi 75 generatorami pakietów ($F=5 \times G=15$) oraz czterema równoległymi pierścieniami użytymi w korzeniu sieci ($R=4$). Krzywe prezentują średnie opóźnienie dla każdego kanału logicznego w funkcji obciążenia tego kanału. Linia ciągła odpowiada logicznemu kanałowi odczytu, linia przerywana odpowiada logicznemu kanałowi zapisu.

Średnie opóźnienie, dla obu kanałów logicznych, zwiększa się wraz z obciążeniem. Zwiększenie obciążenia z 27% do 97% skutkowało średnim zwiększeniem opóźnienia o jedynie 10%. Dla zaprezentowanej konfiguracji sieci oraz dla wszystkich przetestowanych konfiguracji (szczegółowe wyniki przedstawiono w rozprawie), osiągnięto przepustowość T_{RW_MAX} , obliczona według wzoru (1), dla obu kanałów logicznych, niezależnie od współczynnika zwielokrotnienia R , liczby pierścieni pierwszego poziomu F i liczby podłączonych generatorów pakietów G . Z punktu widzenia tez rozprawy, **osiągnięcie przepustowości T_{RW_MAX} , obliczonej według wzoru (1) niezależnie od konfiguracji sieci RingNet jest najważniejszym wnioskiem płynącym z przedstawionych wyników.**

Za pomocą symulacji sprawdzono również sprawiedliwość dostępu urządzeń przetwarzających do zasobów sieci RingNet. Dwa aspekty sprawiedliwego dostępu do sieci zostały przetestowane w symulacjach: proporcjonalność pomiędzy żadaną i otrzymaną częścią przepustowości sieci oraz różnice w wartościach opóźnień z jakimi pakiety od różnych urządzeń przetwarzających są przesyłane przez sieć.

W tabeli II przedstawiono wyniki symulacji sieci w konfiguracji z pięcioma pierścieniami pierwszego poziomu ($F=5$), piętnastoma generatorami pakietów podłączonymi do każdego z tych pierścieni ($G=15$) oraz czterema równoległymi pierścieniami użytymi w korzeniu drzewa ($R=4$). Sieć przetestowano przy obciążeniach równych 27%, 97% i 100% T_{RW_MAX} . W celu sprawdzenia sprawiedliwości w dostępie do sieci, autor zebrał statystyki osobno dla każdego podłączonego generatora. Zebrane statystyki obejmowały średnie opóźnienie L_{PG} , wyrażone w liczbie cykli zegarowych, oraz przydzieloną wartość przepustowości T_{PG} , wyrażoną w bitach na takt zegara. Statystyki zostały zebrane niezależnie dla każdego z kanałów logicznych. W tabeli II przedstawiono średnie wartości L_{PG} i T_{PG} policzone po wszystkich generatorach pakietów (odpowiednio $\overline{L_{PG}}$ i $\overline{T_{PG}}$) oraz odchylenia standardowe tych parametrów (odpowiednio $\sigma_{\overline{L_{PG}}}$ i $\sigma_{\overline{T_{PG}}}$).

TABELA II
STATYSTYKI RUCHU DLA 75 PODŁĄCZONYCH GENERATORÓW PAKIETÓW

Obciążenie (procent przepustowości T_{RW_MAX} (1))	Średnie opóźnienie (cykle zegara)				Średnia przepustowość (bity na cykl zegara)			
	Zapis		Odczyt		Zapis		Odczyt	
	$\overline{L_{PG}}$	$\sigma_{\overline{L_{PG}}}$	$\overline{L_{PG}}$	$\sigma_{\overline{L_{PG}}}$	$\overline{T_{PG}}$	$\sigma_{\overline{T_{PG}}}$	$\overline{T_{PG}}$	$\sigma_{\overline{T_{PG}}}$
27%	236	6	243	6	0.7	0.01	0.68	0.01
97%	259	5	267	5	2.4	0.01	2.4	0.01
100%	1157	7	1175	9	2.5	0.00	2.48	0.00

Wartości odchyłeń standardowych są znacznie mniejsze niż odpowiadające im wartości średnie. Oznacza to, że wszystkie generatory pakietów doznają podobnej obsługi wyrażonej jako średnie opóźnienie i przyznana przepustowość. Przedstawione wyniki demonstrowują sprawiedliwy dostęp do zasobów sieci RingNet.

Zaprezentowane wyniki symulacyjne wykazały zgodność osiągniętych przepustowości RingNet z teoretyczną wartością T_{RW_MAX} , obliczoną wg wzoru (1). **Co należy podkreślić, właściwość sprawiedliwego dostępu do sieci wyróżnia RingNet spośród innych NoC dedykowanych architekturze FPGA.**

Rozdział 4. Wyniki syntezy RingNet

RingNet została zsyntetyzowana dla różnych mikroukładów FPGA głównych producentów w celu określenia jej skalowalności oraz użyteczności w różnych architekturach FPGA.

Autor zsyntetyzował pierścienie RingNet o różnym rozmiarze, użyteczne w konstrukcji sieci w topologii drzewa pierścieni, tj. pierścienie z jednym przełącznikiem *Root Interface* i od dwóch do piętnastu przełączników *Leaf Interfaces* (Ring 2×1 – Ring 15×1). Prezentowane wyniki syntezy obejmują liczbę wykorzystanych zasobów FPGA i maksymalną częstotliwość taktowania.

Wyniki syntezy przedstawiono dla mikroukładów od firmy Xilinx (Artix7, Kintex UltraScale, Virtex7), Intel (Stratix V, Arria V) oraz Lattice (ECP5). Przetestowane mikroukłady stanowią reprezentację architektur stosowanych przez każdego z producentów.

Do syntezy użyto programu Synplify Premier 2017.03-SP1.

Wyniki syntezy zaprezentowano w tabeli III. Otrzymane wartości maksymalnej częstotliwości taktowania zostały porównane z maksymalnymi częstotliwościami podanymi przez producentów dla sprzętowych bloków każdego z mikroukładów FPGA. W tym celu zaprezentowano częstotliwości dla powszechnie występujących sprzętowych bloków cyfrowego przetwarzania sygnałów (ang. *digital signal processing*, DSP) i pamięci blokowej (*block random-access memory*, BRAM). Prezentowane częstotliwości dla bloków sprzętowych, są oszacowaniem maksymalnej częstotliwości pracy dla optymalnej (pod względem częstotliwościowym) implementacji realizowanej w danym mikroukładzie FPGA. **Z tabeli III widać, że maksymalna częstotliwość pracy pierścieni RingNet jest porównywalna lub większa niż częstotliwości podane dla sprzętowych bloków DSP i BRAM.**

TABELA III

WYKORZYSTANIE ZASOBÓW I MAKSYMALNA CZĘSTOTLIWOŚĆ TAKTOWANIA DLA PIERŚCIENI RINGNET ORAZ DLA PRZELĄCZNIKÓW BĘDĄCYCH STANEM TECHNIKI

Pierścienie RingNet, przełączniki znane z literatury oraz sprzętowe moduły mikroukładów FPGA	Liczba wykorzystanych przerzutników / Liczba LUT wykorzystanych jako logika + liczba LUT wykorzystanych jako RAM / Maksymalna częstotliwość zegara w MHz					
	Xilinx			Intel		Lattice
	Artix7 xc7a100tcs324-1	Kintex UltraScale xc7ku060-ffva1156-3-e	Virtex7 xc7vx550tffg1158-1	Stratix V 5SGXMABK2H40C3	Arria V 5AGXBA7D6F31C6	ECP5 lfe5u_85f-8
Ring 2×1	2k8 / 805 + 424 / 354	2k8 / 801 + 424 / 753	2k8 / 805 + 424 / 445	2k8 / 631 + 916 / 449	2k8 / 648 + 914 / 273	2k8 / 1k3 + 1k4 / 235
Ring 4×1	4k3 / 1k2 + 632 / 363	4k3 / 1k2 + 632 / 700	4k3 / 1k2 + 632 / 460	4k3 / 934 + 1k2 / 430	4k3 / 966 + 1k2 / 250	4k3 / 2k0 + 2k3 / 235
Ring 6×1	5k8 / 1k7 + 840 / 333	5k8 / 1k7 + 840 / 763	5k8 / 1k7 + 840 / 440	5k8 / 1k3 + 1k6 / 409	5k8 / 1k3 + 1k6 / 262	5k9 / 2k8 + 3k3 / 235
Ring 8×1	7k3 / 2k1 + 1k0 / 328	7k3 / 2k1 + 1k0 / 726	7k3 / 2k1 + 1k0 / 440	7k3 / 1k6 + 1k9 / 397	7k3 / 1k7 + 1k9 / 272	7k4 / 3k6 + 4k2 / 235
Ring 11×1	9k5 / 2k8 + 1k4 / 333	9k5 / 2k8 + 1k4 / 753	9k5 / 2k8 + 1k4 / 446	9k5 / 2k1 + 2k4 / 434	9k6 / 2k2 + 2k4 / 241	9k7 / 4k9 + 5k8 / 235
Ring 15×1	12k5 / 3k7 + 1k8 / 325	12k5 / 3k7 + 1k8 / 763	12k5 / 3k7 + 1k8 / 443	12k5 / 2k7 + 3k1 / 403	12k5 / 2k8 + 3k0 / 222	12k8 / 6k5 + 7k6 / 235
Przełącznik z [Mai15]		--- / 1678 / 470				
Przełącznik z [Pap15]	--- / 430 / 241 (wyniki dla Virtex6 LX760 w wersji <i>speed grade -2</i>)					
DSP	/ 392	/ 687	/ 463	/ 400	/ 200	/ 185
BRAM	/ 339	/ 575	/ 400	/ 455	/ 240	/ 214

Najmniejsze wartości częstotliwości maksymalnej, dla pierścieni RingNet oraz dla bloków sprzętowych, są zaznaczone kolorem **czzerwonym**. Najmniejsze wartości są wyznaczone osobno dla każdego mikroukładu FPGA.

W tabeli III podano liczbę wykorzystanych przerzutników i tablic funkcji (ang. *look up tables*, LUTs) dla pierścieni RingNet. Liczba wykorzystanych przerzutników jest niemal identyczne we wszystkich mikroukładach FPGA, z maksymalnym odchyleniem od wartości średniej równej 2%.

W tabeli III, ze względu na przeznaczenie wykorzystanych LUT, rozróżniono dwa ich typy: LUT wykorzystane do implementacji logiki i LUT wykorzystane jako pamięć. Zgodnie z przewidywaniami wyrażonymi w rozprawie, liczba LUT wykorzystanych do implementacji logiki jest średnio 1,8 razy większa w mikroukładzie Lattice, niż w mikroukładach Xilinx i Intel. Różnice pomiędzy liczbą LUT wykorzystanych w mikroukładach Xilinx i Intel jest znacznie mniejsza i sięga 19%, co również zgadza się z przewidywaniem opisanym w rozprawie.

Liczba LUT wykorzystanych jako pamięć różni się znacznie pomiędzy producentami. W tabeli IV przedstawiono uśrednione relacje pomiędzy liczbą LUT wykorzystanych jako pamięć w mikroukładach różnych producentów.

TABELA IV
WZGLĘDNA LICZBA LUT WYKORZYSTANYCH JAKO PAMIĘĆ W MIKROUKŁADACH FPGA RÓŻNYCH
PRODUCENTÓW

Pierścień RingNet	Intel : Xilinx	Lattice : Xilinx	Lattice : Intel
Ring 2×1	2.16	3.41	1.58
Ring 4×1	1.97	3.71	1.88
Ring 6×1	1.88	3.98	2.12
Ring 8×1	1.82	4.04	2.22
Ring 11×1	1.73	4.15	2.40
Ring 15×1	1.70	4.18	2.46
Średnia po pierścieniach	1.88	3.91	2.11
Szacunek	> 1.5	< 4.5	< 3

Opierając się na własnej analizie architektur FPGA różnych producentów, autor oszacował spodziewany stosunek liczby LUT wykorzystanych w mikroukładach różnych producentów. Przewidywane wartości, zamieszczone w ostatnim rzędzie tabeli IV, zgadzają się z wartościami uzyskanymi w procesie syntezy. Najbardziej istotną różnicą w architekturach FPGA, która wpływa na liczbę LUT wykorzystanych jako pamięć, jest różna liczby bitów pamięci

przypadających na jedną LUT: średnio 32 bity dla LUT od Intel, 48 dla LUT od Xilinx i 10.7 dla LUT od Lattice.

W rozprawie porównano pierścienie RingNet z przełącznikami uznanymi za stan techniki, zaproponowanymi w [Mai15] oraz w [Pap15]. Autor wykazał, że dla danej liczby podłączonych urządzeń przetwarzających, pierścienie RingNet używają mniej zasobów niż sieci z [Mai15] i [Pap15]. Ten efekt uzyskano dzięki współdzieleniu w RingNet części logiki sterującej przepływem pomiędzy wszystkie przełączniki z danego pierścienia, oraz dzięki użyciu małych i zoptymalizowanych pod mikroukłady FPGA trójbramowych przełączników. W odróżnieniu od RingNet, w [Mai15] używane są przełączniki pięciobramowe. W [Pap15], podobnie jak w RingNet, użyto małych przełączników trójbramowych, jednak w [Pap15] nie wykorzystano współdzielenia logiki sterującej pomiędzy przełącznikami. W efekcie podłączenie jednego urządzenia przetwarzającego do sieci [Mai15] generuje koszt 1678 LUT, w przypadku sieci [Pap15] jest to 430 LUT, a dla RingNet ten koszt wynosi 367 LUT. Należy jednak zauważyć, że mniejsza liczba wykorzystanych zasobów dla sieci RingNet pociąga za sobą większą liczbę taktów zegarowych potrzebnych na transfer pakietu przez sieć niż w NoC z [Mai15] i [Pap15]. Z drugiej strony, większa liczba potrzebnych taktów, może zostać złagodzona przez znacznie większe częstotliwości taktowania dostępne dla RingNet. W rozprawie autor wykazał, że pierścienie RingNet mogą być taktowane z większą częstotliwością niż przełączniki z [Mai15] i [Pap15], gdy rozpatrywane są wyniki dla tych samych mikroukładów FPGA. Przykładowo, dla mikroukładu Kintex UltraScale od Xilinx, maksymalna częstotliwość taktowania RingNet to ponad 700 MHz, podczas gdy dla przełącznika [Mai15] jest to tylko 470 MHz. Dla mikroukładu Virtex6 maksymalna częstotliwość taktowania sieci [Pap15] stanowi pomiędzy 31% – 59% maksymalnej częstotliwości taktowania bloku DSP, podczas gdy odpowiadające wartości dla RingNet to od 84% do 107%.

Poprzez wyniki syntezy, autor zaprezentował, że sieć RingNet może być efektywnie zaimplementowana w mikroukładach FPGA różnych producentów, dając porównywalne wykorzystanie zasobów i dużą maksymalną częstotliwość taktowania. Jednocześnie wykazano, że RingNet charakteryzuje się mniejszą liczbą wykorzystanych zasobów i większą maksymalną częstotliwością taktowania niż inne przełączniki będące stanem techniki.

Rozdział 5. Porównanie implementacji pierścieni RingNet i AXI4 Interconnect

Autor porównał RingNet z AXI4, czyli z infrastrukturą komunikacyjną powszechnie używaną w mikroukładach FPGA. Implementacja AXI4 dla mikroukładów Xilinx jest dostępna w [Xil15]. AXI4 łączy urządzenia przetwarzające używając modułu AXI4 Interconnect. Jest on zbudowany z przełącznika krzyżowego, kolejek danych typu *first-in-first-out* (fifo), używanych do buforowania danych, przerzutników potokowych (ang. *pipeline flip-flops*), używanych do przecięcia ścieżek krytycznych, oraz z dekodera adresów.

Mając na celu sprawiedliwe porównanie RingNet i AXI4 Interconnect, skonfigurowano oba rozwiązania komunikacyjne tak, aby miały podobne właściwości. Zastosowane konfiguracje są opisane w tabeli V.

TABELA V
KONFIGURACJA PIERŚCIENIA RINGNET I AXI4 INTERCONNECT

Parametr	Pierścień RingNet	AXI4 Interconnect
Szerokość adresu	37 bitów	37 bitów
Szerokość słów danych	64 bitów (pojedynczy pierścień)	64 bitów
Liczba i typ interfejsów	$N \times 1$: - 1 interfejs (<i>Root Interface</i>) do podłączenia kolejnego pierścienia lub pamięci systemowej - $N (2 - 15)$ interfejsów (<i>Leaf Interface</i>) dla urządzeń przetwarzających	$N \times 1$: - 1 interfejs urządzenia podrzędnego (<i>slave interface</i>) do podłączenia pamięci, - $N (2 - 15)$ interfejsów nadrzędnych (<i>master interfaces</i>) do podłączenia urządzeń przetwarzających
Moduł sterowania przepływem	<i>Leaf-to-Root Manager</i>	<i>Arbiter Round-Robin</i>
Buforowanie	Kolejki fifo oparte o LUTRAM	
Optymalizacja wydajności	---	Przełącznik krzyżowy w wersji zoptymalizowanej na wydajność, zwanej <i>Shared-Address, Multiple-Data</i> (SAMD), Wykorzystanie przerzutników potokowych zwanych <i>AXI Register Slice</i>
Znaczniki danych	Jeden bit wskazujący ważność danych na każdy przesłany bajt danych.	

Autor rozprawy, w drugiej jej tezie, zakłada możliwość niższego wykorzystania zasobów i większej częstotliwości taktowania sieci RingNet niż przełącznika AXI4 Interconnect. W celu sprawdzenia prawdziwości postawionej tezy, autor dokonał implementacji obu rozwiązań komunikacyjnych. Wyniki przedstawiono dla przykładowego mikroukładu, tj. Artix7 firmy Xilinx (xc7a100tcs9324-1). Porównywalne wyniki RingNet dla różnych architektur FPGA zostały wcześniej omówione w rozdziale 4, dlatego, pomimo ograniczenia wyników implementacji do jednego mikroukładu, wnioski z implementacji mogą być rozciągnięte również na inne mikroukłady FPGA.

Do konfiguracji modułu AXI4 Interconnect oraz do uzyskania wyników implementacji użyto pakietu oprogramowania Vivado Design Suite 2016.4 od firmy Xilinx, producenta mikroukładu Artix7.

Uzyskane wyniki implementacji zaprezentowane są w tabeli VI.

TABELA VI

WYKORZYSTANIE ZASOBÓW I MAKSYMALNA CZĘSTOTLIWOŚĆ TAKTOWANIA WYZNACZONA DLA PIERŚCIENI RINGNET I AXI4 INTERCONNECT

		Liczba wykorzystanych LUT	Liczba wykorzystanych przerzutników	Częstotliwość maksymalna [MHz]
AXI4 Interconnect	2×1	1370	2801	268
	4×1	2205	4480	226
	6×1	3175	6151	192
	15×1	7181	13650	151
Pierścień RingNet	2×1	1185	2047	396
	4×1	1846	3188	392
	6×1	2557	4343	382
	15×1	5650	9595	365

Dla wszystkich przetestowanych konfiguracji, pierścienie RingNet używają mniej zasobów niż AXI4 Interconnect. Liczba wykorzystanych przerzutników jest około 29% mniejsza, dla wszystkich konfiguracji. Redukcja liczby wykorzystanych LUT wynosi od 14% (dla konfiguracji 2×1) do 21% (dla konfiguracji 15×1).

Maksymalne częstotliwości taktowania AXI4 Interconnect, przedstawione w tabeli VI, wskazują, że rozwiązanie to nie skaluje się dobrze i maksymalna częstotliwość taktowania drastycznie maleje wraz z rosnącym rozmiarem przełącznika krzyżowego. Głównym powodem takiej sytuacji jest wydłużanie połączeń w miarę zwiększania liczby interfejsów przełącznika AXI4 Interconnect [Mai15]. Z drugiej strony, **RingNet, dzięki użyciu zoptymalizowanego pod mikroukłady FPGA przełącznika trójbramowego i topologii pierścienia, charakteryzuje się dużą maksymalną częstotliwością taktowania dla szerokiego zakresu rozmiarów pierścienia.** Dla odpowiadających sobie konfiguracji, pierścień RingNet pozwala na taktowanie zegarem o większej częstotliwości niż AXI4 Interconnect. Dla konfiguracji 2×1 maksymalna częstotliwość zegara dla RingNet jest o 48% większa niż dla AXI4 Interconnect, a dla konfiguracji 15×1 różnica ta sięga już 142%.

Rozdział 6. Podsumowanie

W obu tezach autor rozprawy stwierdza, że możliwe jest opracowanie architektury i protokołu sieci w mikroukładzie, charakteryzujących się regulowaną przepustowością i sprawiedliwym dostępem do sieci. W celu weryfikacji postawionych tez, zaprezentowano oryginalną sieć w mikroukładzie, zwaną RingNet. W rozprawie przedstawiono architekturę i protokół RingNet oraz omówił przykładową budowę elementów sieciowych. **Właściwość gwarantowanej przepustowości dla sieci RingNet została wykazana z użyciem symulacji.** Zaprezentowano zgodność pomiędzy maksymalną przepustowością kanałów logicznych sieci a wartością obliczoną według wzoru (1). Ponadto, **poprzez wyniki symulacyjne, autor zademonstrował sprawiedliwy dostęp do sieci RingNet**, dla podłączonych urządzeń przetwarzających, wyrażony w podziale przepustowości proporcjonalnym do zgłaszanych żądań oraz zbliżonych wartościach doznawanego opóźnienia.

Ad T1) Możliwe jest opracowanie architektury i protokołu sieci w mikroukładzie charakteryzującej się regulowaną przepustowością, sprawiedliwym dostępem do sieci oraz maksymalną częstotliwością taktowania większą niż 90% maksymalnej częstotliwości taktowania zasobów sprzętowych FPGA, dla mikroukładów FPGA głównych producentów.

Maksymalne częstotliwości taktowania zostały sprawdzone w procesie syntezy i implementacji opracowanych elementów sieciowych i całych pierścieni sieci RingNet. Testy przeprowadzono dla mikroukładów FPGA głównych producentów. Uzyskane maksymalne częstotliwości taktowania zostały porównane z maksymalnymi częstotliwościami podawanymi przez producentów dla bloków cyfrowego przetwarzania sygnałów (ang. *digital signal processing*, DSP) oraz pamięci blokowych (ang. *block random-access memory*, BRAM) dostępnych w testowanych mikroukładach. **Maksymalna częstotliwość taktowania elementów sieciowych i pierścieni RingNet jest równa lub większa niż 96% częstotliwości, z jaką mogą być taktowane sprzętowe bloki dostępne w testowanych mikroukładach (DSP i BRAM).**

Przedstawione duże wartości częstotliwości taktowania są rezultatem efektywnego wykorzystania zasobów dostępnych we wszystkich rozważanych architekturach FPGA. W szczególności prosta logika trójbramowych przełączników jest zaimplementowana z wykorzystaniem małej liczby tablic funkcji (ang. *look up tables*, LUTs), co pozwala na dużą

maksymalną częstotliwość taktowania. Ponadto, uniemożliwienie bezpośredniej komunikacji pomiędzy urządzeniami przetwarzającymi zmniejszyło wymagania na rozmiar buforów sieciowych, dzięki czemu, możliwa jest implementacja tych buforów z wykorzystaniem pamięci rozproszonej o małej pojemności, ale charakteryzującej się dużą dostępnością i dużymi częstotliwościami taktowania. Wykorzystanie zasobów FPGA zostało sprawdzone dla wybranych mikroukładów głównych producentów. Zaobserwowano różnice w liczbie wykorzystanych zasobów, pomiędzy mikroukładami różnych producentów. Różnice te są zgodne z przewidywaniami, które są szeroko omówione w rozprawie. Przewidywania autora zostały oparte na analizie różnic w architekturach stosowanych przez głównych producentów mikroukładów FPGA. Zgodność wyników eksperymentalnych z szacunkami autora wskazuje, że w procesie opracowania sieci RingNet uwzględniono, krytyczne dla implementacji NoC, różnice w rozważanych architekturach FPGA.

Ad. T2) Możliwe jest opracowanie architektury i protokołu sieci w mikroukładzie charakteryzującej się regulowaną przepustowością oraz sprawiedliwym dostępem do sieci, dedykowanej FPGA, która użyje mniej zasobów oraz będzie charakteryzowała się większą maksymalną częstotliwością taktowania niż przełącznik krzyżowy będący stanem techniki (AXI4 Interconnect).

Wyniki implementacji pierścieni RingNet zostały porównane z wynikami implementacji AXI4 Interconnect. W rozprawie autor wykazał, że pierścień RingNet wymaga mniej zasobów niż AXI4 Interconnect, dla odpowiadających sobie konfiguracji obu architektur. Liczba przerzutników wykorzystanych do implementacji pierścienia RingNet jest średnio o 30% mniejsza, natomiast redukcja liczby wykorzystanych LUT wynosi od 14% do 21%, w zależności od konfiguracji. Przedstawione w rozprawie maksymalne częstotliwości taktowania są większe dla RingNet o co najmniej 48%. **Przedstawione wyniki dowodzą, że architektura RingNet wyraźnie przewyższa rozwiązanie będące stanem techniki.**

6.1. Osiągnięcia rozprawy

Podstawowym osiągnięciem rozprawy jest propozycja architektury NoC efektywnie wykorzystującej zasoby nowoczesnych mikroukładów FPGA. W przypadku wszystkich przetestowanych mikroukładów, pochodzących od głównych producentów, NoC o zaproponowanej architekturze RingNet może być taktowana częstotliwością równą lub większą

niż 96% maksymalnej częstotliwości taktowania zasobów sprzętowych tych mikroukładów FPGA. Co więcej, dla przykładowych mikroukładów, maksymalne częstotliwości taktowania dla RingNet są znacznie większe niż te raportowane dla przełączników sieciowych będących stanem techniki, i dla szeroko stosowanego przełącznika AXI4 Interconnect. Duża częstotliwość taktowania RingNet oraz wydajne wykorzystanie zasobów FPGA, odnotowane dla szerokiej gamy mikroukładów FPGA głównych producentów, czynią RingNet siecią kompatybilną międzyukładowo. **RingNet jest jedyną NoC zaprojektowaną dla FPGA, o której wie autor, która osiągnęła ambitny cel kompatybilności z różnymi architekturami FPGA.**

Autor zaprezentował oryginalne rozwiązanie problemu określenia rozmiaru buforów w architekturze NoC. Główną przyczyną tego problemu jest *a priori* nieznane obciążenie sieci [Son03], [Abb14]. **Autor zaproponował, aby w sieci RingNet, obciążenie było regulowane przez docelowe urządzenie przetwarzające.** Mając to na uwadze, zabroniono bezpośredniej komunikacji pomiędzy urządzeniami przetwarzającymi w sieci RingNet. Zamiast tego, urządzenia przetwarzające, podłączone do RingNet, wymieniają dane przez pamięć systemową (np. zewnętrzną SDRAM). **To oryginalne podejście ogranicza zatory sieciowe, dzięki czemu możliwe jest wykorzystanie buforów o z góry ustalonym rozmiarze.** Opisany schemat komunikacji pośredniej pociąga za sobą zwiększenie liczby taktów zegarowych potrzebnych na transfer pakietu przez sieć, w porównaniu do NoC znanych z literatury. Niemniej jednak większa liczba potrzebnych taktów zegarowych, może zostać złagodzona przez znacznie większe częstotliwości taktowania dostępne dla RingNet.

W sieci RingNet użyto techniki przełączania *virtual cut-through* i buforów wykorzystujących pamięć rozproszoną, jako że takie połączenie techniki przełączania i rodzaju buforów jest znane ze swojej efektywności w NoC zaimplementowanych dla mikroukładów FPGA od Xilinx. **Oryginalnym osiągnięciem autora jest zademonstrowanie wydajności techniki przełączania *virtual cut-through* i buforów opartych o pamięć rozproszoną także w innych architekturach FPGA.**

Autor zaproponował oryginalną topologię drzewa pierścieni, odpowiednią dla mikroukładów FPGA, co zostało sprawdzone analizę wyników przeprowadzonych syntez i symulacji. Celem zaproponowanej topologii hybrydowej jest połączenie zalet topologii pierścienia i topologii drzewa w jedną efektywną topologię, dedykowaną mikroukładowi

FPGA. Topologia pierścienia charakteryzuje się małą liczbą wykorzystanych zasobów mikroukładów FPGA i pozwala na implementację o dużej częstotliwości taktowania, co zawdzięcza użyciu trójbramowych przełączników. Połączenie topologii pierścienia z topologią drzewa ma na celu zredukowanie znacznych opóźnień charakterystycznych dla czystej topologii pierścienia.

Autor zaproponował oryginalny mechanizm sterowania przepływem dla RingNet, który zapewnia sprawiedliwy dostęp do sieci pod kątem doznawanych opóźnień oraz sprawiedliwy podział przepustowości sieci.

W celu oceny osiągnięć rozprawy, przeprowadzono serię eksperymentów. Na ich potrzeby, zaimplementowano komponenty sieci RingNet w języku Verilog. Przygotowany opis komponentów został przystosowany do wymagań składniowych różnych producentów mikroukładów FPGA. Wymagało to przygotowania alternatywnych fragmentów kodu, szczególnie opisujących pamięci buforów sieciowych. **Autor szacuje liczbę własnego kodu Verilog na 8000 linii.** Przeprowadzone eksperymenty, w szczególności symulacje, wymagały dużego nakładu obliczeniowego. Wyniki zaprezentowane w rozprawie podsumowują ponad 3000 symulacji. Dodatkowe symulacje przeprowadzono w celu dobrania parametrów sieci. Łącznie autor **przeprowadził ponad 15000 symulacji**, które wymagały równowartości około 500 dni obliczeń pojedynczego rdzenia procesora.

Ponadto, autor przeanalizował podobieństwa i różnice w architekturze bloków logicznych występujących w mikroukładach FPGA głównych producentów oraz **zaproponował praktyczny zestaw przeliczników, pozwalających oszacować liczbę wykorzystanych LUT i przerzutników**, gdy projekt zostanie przeniesiony pomiędzy mikroukładami FPGA od Lattice, Intel i Xilinx.

Oryginalne osiągnięcia rozprawy zostały opisane w artykule “RingNet: A Memory-Oriented Network-On-Chip Designed for FPGA,” opublikowanym w czasopiśmie Institute of Electrical and Electronics Engineers (IEEE) Transactions on Very Large Scale Integrated (VLSI) Systems w 2019 roku [Sia19]. Publikacja uzupełniona jest o 78 stron materiałów dodatkowych, takich jak szczegółowe wyniki symulacyjne i opis implementacji elementów sieciowych, dostępnych w serwisie IEEE Xplore.

6.2. Wnioski

Cechami charakterystycznymi RingNet są: komunikacja prowadzona poprzez pamięć systemową (podłączoną do mikroukładu FPGA pamięć SDRAM lub wbudowaną pamięć blokową), regulacja ruchu w sieci prowadzona przez urządzenia przetwarzające podłączone do sieci (ang. *processing elements*), zoptymalizowane pod mikroukłady FPGA trójbramowe przełączniki sieciowe zorganizowane w topologię drzewa pierścieni, pamięć rozproszona (LUTRAM) użyta do implementacji małych buforów w przełącznikach, oraz zastosowanie przełączania *virtual cut-through*.

W rozprawie zademonstrowano takie właściwości RingNet jak: gwarantowaną przepustowość, dające się przewidzieć opóźnienie i sprawiedliwy dostęp do sieci. Analiza implementacji sieci RingNet zademonstrowała jej dużą wydajność rozumianą jako dużą maksymalną częstotliwość sygnału taktującego i niskie wykorzystanie zasobów dla flagowych mikroukładów FPGA głównych producentów. Implementacja została porównana z powszechnie stosowaną architekturą komunikacyjną AXI4 Interconnect, co wykazało większą maksymalną częstotliwość taktowania i mniejszą liczbę wykorzystanych zasobów na korzyść RingNet. Z tego powodu, autor wierzy, że architektura i protokół sieci RingNet mogą być powszechnie wykorzystane w systemach opartych o układy programowalne, w szczególności w aplikacjach przetwarzania dużej ilości danych, takich jak przetwarzanie sekwencji wizyjnych.

WYBRANE POZYCJE SPISU LITERATURY

- [Abb14] S. Abba and Jeong-A Lee, "A parametric-based performance evaluation and design trade-offs for interconnect architectures using FPGAs for networks-on-chip," in *Microprocessors and Microsystems*, vol. 38, no. 5, pp. 375-398, July 2014.
- [Abd14] M. S. Abdelfattah and V. Betz, "The case for embedded networks-on-chip on field-programmable gate arrays," in *IEEE Micro*, vol. 34, no. 1, pp. 80-89, Jan.-Feb. 2014.
- [Abd16a] M. S. Abdelfattah and V. Betz, "Power analysis of embedded NoCs on FPGAs and comparison with custom buses," in *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 1, pp. 165-177, Jan. 2016.
- [Abd16b] M. Abdelfattah, V. Betz, "Embedded networks-on-chip for FPGA," in *Reconfigurable Logic: Architecture, Tools, and Applications*, Boca Raton, FL, USA: CRC Press, 2016.
- [Abd17] M. S. Abdelfattah, A. Bitar and V. Betz, "Design and applications for embedded networks-on-chip on FPGAs," in *IEEE Trans. Comput.*, vol. 66, no. 6, pp. 1008-1021, June 2017.
- [Add17] M. Adda and A. Peratikou, "Routing and fault tolerance in Z-Fat Tree," in *IEEE Trans. Parallel and Distrib. Syst.*, vol. 28, no. 8, pp. 2373-2386, 1 Aug. 2017.
- [Ahm18] R. Ahmed, H. Mostafa and A. H. Khalil, "Impact of dynamic partial reconfiguration on CONNECT network-on-chip for FPGAs," in *Proc. 2018 13th Int. Conf. on Design & Technol. of Integrated Systems In Nanoscale Era (DTIS)*, Taormina, 2018, pp. 1-5.
- [Alt09] *Quartus II Handbook, Recommended HDL Coding Styles*, QII51007-9.1.0, Altera Co., Nov. 2009.
- [Alt11] *Stratix V device handbook*, SV51002, v1.3, vol. 2, Altera Co., Nov. 2011.
- [Alt15a] *Stratix V device overview*, SV51001, Altera Co., Oct. 2015.
- [Alt15b] *Devices: 28nm device portfolio*, Altera Product Catalog, Altera Co., 2015.
- [Alt16] *Arria 10 device overview*, A10-overwiev, Altera Co., Oct. 2016.
- [Ben02] L. Benini and G. D. Micheli, "Networks-on-chips: A new SoC paradigm," in *IEEE Computers*, vol. 35, no. 1, pp. 70-78, Jan. 2002.
- [Ber04] D. Bertozzi and L. Benini, "Xpipes: A network-on-chip architecture for gigascale systems-on-chip," in *IEEE Circuits and Syst. Magazine*, vol. 4, no. 2, pp. 18-31, 2004.
- [Buk17] A. V. Bukit and Wirawan, "3D video coding development based on FPGA platform Xilinx Zynq-7000," in *Proc. 2017 Int. Seminar on Intelligent Technol. and Its Applications (ISITIA)*, Surabaya, 2017, pp. 29-34.
- [Bre17] M. F. Brejza, R. G. Maunder, B. M. Al-Hashimi and L. Hanzo, "A high-throughput FPGA architecture for joint source and channel decoding," in *IEEE Access*, vol. 5, pp. 2921-2944, 2017.

- [Cha15] H. Y. Chang, I. H. R. Jiang, H. P. Hofstee, D. Jamsek and G. J. Nam, "Feature detection for image analytics via FPGA acceleration," in *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 8:1-8:10, March-May 2015.
- [Dal99] W. J. Dally, "Interconnect-limited VLSI architecture," in *Proc. of the IEEE 1999 Inter. Interconnect Technol. Conf.*, San Francisco, CA, USA, 1999, pp. 15-17.
- [Dal01] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Proc. of the 38th Design Automation Conf. (IEEE Cat. No.01CH37232)*, 2001, pp. 684-689.
- [Dal03] W. Dally and B. Towles, *Principles and Practices of Interconnection Network*, San Francisco, USA: Morgan Kaufmann, 2003.
- [Hel15] K. A. Helal, S. Attia, T. Ismail and H. Mostafa, "Comparative review of NoCs in the context of ASICs and FPGAs," in *Proc. 2015 IEEE Int. Symp. Circuits and Syst. (ISCAS)*, Lisbon, pp. 1866-1869.
- [Hud16] S. Huda, J. Anderson, "Circuits and embedded networks-on-chip for FPGA," in *Reconfigurable Logic: Architecture, Tools, and Applications*, Boca Raton, FL, USA: CRC Press, 2016.
- [Int16] *Stratix 10 GX/SX device overview*, S10-overview, Intel Co., Oct. 2016.
- [Int17] *Intel Cyclone 10 GX device overview*, C10GX51001, Intel Co., Nov. 2017.
- [Int18] *Intel Stratix 10 Logic Array Blocks and Adaptive Logic Modules*, UG-S10LAB, Intel Co., Sep. 2018.
- [Kap15] N. Kapre and J. Gray, "Hoplite: Building austere overlay NoCs for FPGAs," in *Proc. Int. Conf. Field Programmable Logic and Applications (FPL)*, London, 2015, pp. 1-8.
- [Kap17a] N. Kapre, "On bit-serial NoCs for FPGAs," in *Proc. IEEE Int. Symp. Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, 2017, pp. 32-39.
- [Kap17b] N. Kapre, "Implementing FPGA overlay NoCs using the Xilinx UltraScale memory cascades," in *Proc. IEEE Annu. Int. Symp. Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, 2017, pp. 40-47.
- [Kap17c] N. Kapre, "Deflection-routed butterfly fat trees on FPGAs," in *Proc. Int. Conf. Field Programmable Logic and Applications (FPL)*, Ghent, 2017, pp. 1-8.
- [Kil00] J. S. Kilby, "The integrated circuit's early history," in *Proc. of the IEEE*, vol. 88, no. 1, pp. 109-111, Jan. 2000.
- [Kum16] R. Kumar and A. Gordon-Ross, "MACS: A highly customizable low-latency communication architecture," in *IEEE Trans. Parallel and Distrib. Syst.*, vol. 27, no. 1, pp. 237-249, Jan. 2016.
- [Lat13] *LatticeECP2/M family data sheet*, DS1006, v04.1, Lattice Semiconductor Co., Sept. 2013.
- [Lat15a] *LatticeECP3 family data sheet*, DS1021, v02.8EA, Lattice Semiconductor Co., Mar. 2015.

- [Lat15b] *ECP5 and ECP5-5G Memory Usage Guide*, TN1264, v1.2, Lattice Semiconductor Co., Nov. 2015.
- [Lat16] *ECP5 and ECP5-5G family*, DS1044, v1.6, Lattice Semiconductor Co., Feb. 2016.
- [Lei85] C. E. Leiserson, "Fat-trees: Universal networks for hardware-efficient supercomputing," in *IEEE Trans. on Computers*, vol. C-34, no. 10, pp. 892-901, Oct. 1985.
- [Lic18] G. D. Licciardo, C. Cappetta and L. D. Benedetto, "Design of a gabor filter HW accelerator for applications in medical imaging," in *IEEE Trans. on Components, Packaging and Manufacturing Technol.*, pp. 1-8, April, 2018.
- [LiuS14] S. Liu, A. Jantsch and Z. Lu, "A fair and maximal allocator for single-cycle on-chip homogeneous resource allocation," in *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 10, pp. 2230-2234, Oct. 2014.
- [LiuT16] T. Liu, N. K. Dumpala and R. Tessier, "Hybrid hard NoCs for efficient FPGA communication," in *Proc. 2016 Int. Conf. on Field-Programmable Technol. (FPT)*, Xi'an, 2016, pp. 157-164.
- [Łucz08] A. Łuczak, P. Garstecki, O. Stankiewicz and M. Stępniewska, "Network-on-chip based architecture of H. 264 video decoder," in *Proc. Int. Conf. on Signals and Electronic Syst.*, Krakow, Poland, 14-17 September 2008, pp. 419-422.
- [Mai15] P. Maidee and A. Kaviani, "Improving FPGA NoC performance using virtual cut-through switching technique," in *Proc. Int. Conf. ReConFigurable Computing and FPGAs (ReConFig)*, Mexico City, 2015, pp. 1-6.
- [Mai17] P. Maidee, A. Kaviani and K. Zeng, "LinkBlaze: Efficient global data movement for FPGAs," in *Proc. Int. Conf. on ReConFigurable Computing and FPGAs (ReConFig)*, Cancun, 2017, pp. 1-8.
- [Mat09] H. Matsutani, M. Koibuchi, Y. Yamada, D. F. Hsu and H. Amano, "Fat H-Tree: A cost-efficient tree-based on-chip network," in *IEEE Trans. Parallel and Distrib. Syst.*, vol. 20, no. 8, pp. 1126-1141, Aug. 2009.
- [Muk17] A. K. Mukhopadhyay and A. Sarkar, "Comparative analysis of some crosstalk avoidance coding techniques in deep submicron VLSI interconnects," in *Proc. 2017 Devices for Integrated Circuit (DevIC)*, Kalyani, 2017, pp. 1-5.
- [Pap12] M. K. Papamichael and J. C. Hoe, "CONNECT: Re-examining conventional wisdom for designing NoCs in the context of FPGAs," in *Proc. ACM/SIGDA Int. Symp. Field Programmable Gate Arrays*, New York, 2012, pp. 37-46.
- [Pap15] M. K. Papamichael and J. C. Hoe, "The CONNECT network-on-chip generator," in *Computer*, vol. 48, no. 12, pp. 72-79, Dec. 2015.
- [Pos13] J. Postman, T. Krishna, C. Edmonds, L. S. Peh and P. Chiang, "SWIFT: A low-power network-on-chip implementing the token flow control router architecture with swing-reduced interconnects," in *IEEE Trans. on Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 8, pp. 1432-1446, Aug. 2013.
- [Red19] K. S. Reddy and K. Vipin, "OpenNoC: An open-source NoC infrastructure for FPGA-based hardware acceleration," in *IEEE Embedded Systems Letters*, Early Access, Mar. 2019.

- [Ret14] J. Rettkowski and D. Göhringer, “RAR-NoC: A reconfigurable and adaptive routable network-on-chip for FPGA-based multiprocessor systems,” in *Proc. Int. Conf. ReConFigurable Computing and FPGAs (ReConFig)*, Cancun, 2014, pp. 1-6.
- [She14] S. N. Shelke and P. B. Patil, “Low-latency, low-area overhead and high throughput NoC architecture for FPGA based computing system,” in *Proc. Int. Conf. Electronic Syst., Signal Process. and Computing Technol.*, Nagpur, 2014, pp. 53-57.
- [Sid18] Siddhartha and N. Kapre, “Hoplite-Q: Priority-aware routing in FPGA overlay NoCs,” in *Proc. 2018 IEEE 26th Annual Int. Symp. on Field-Programmable Custom Computing Machines (FCCM)*, Boulder, 2018, pp. 17-24.
- [Sny17] W. Snyder, “Introduction to Verilator,” [Online], Available: <https://www.veripool.org/wiki/verilator>, Accessed on: Dec. 19, 2017.
- [Son03] Y. H. Song and T. M. Pinkston, “A progressive approach to handling message-dependent deadlock in parallel computer systems,” in *IEEE Trans. Parallel and Distrib. Syst.*, vol. 14, no. 3, pp. 259-275, Mar. 2003.
- [The11] D. Theodoropoulos, G. Kuzmanov and G. Gaydadjiev, “Multi-core platforms for beamforming and wave field synthesis,” in *IEEE Trans. on Multimedia*, vol. 13, no. 2, pp. 235-245, April 2011.
- [Tto16] C. Ttofis, C. Kyrkou and T. Theocharides, “A low-cost real-time embedded stereo vision system for accurate disparity estimation based on guided image filtering,” in *IEEE Trans. on Computers*, vol. 65, no. 9, pp. 2678-2693, Sept. 1 2016.
- [Was17] S. Wasly, R. Pellizzoni and N. Kapre, “HopliteRT: An efficient FPGA NoC for real-time applications,” in *Proc. Int. Conf. Field Programmable Technol. (ICFPT)*, Melbourne, 2017, pp. 64-71.
- [Vip17] K. Vipin, J. Gray and N. Kapre, “Enabling partial reconfiguration and low latency routing using segmented FPGA NoCs,” in *Proc. Int. Conf. Field Programmable Logic and Applications (FPL)*, Ghent, 2017, pp. 1-8.
- [Xil15] *Vivado AXI reference guide*, UG1037, v3.0, Xilinx Inc., June 2015.
- [Xil16a] *Zynq UltraScale+ MPSoC*, XMP104, v2.1, Xilinx Inc., 2016.
- [Xil16b] *UltraScale FPGA*, XMP102, v1.7, Xilinx Inc., 2016.
- [Xil16c] *All programmable 7 Series*, XMP101, v1.2, Xilinx Inc., 2016.
- [Xil16d] *Cost-optimized portfolio*, XMP100, v1.6, Xilinx Inc., 2016.
- [Xil16e] *7 Series FPGAs: Configurable Logic Block*, UG474, v1.8, Xilinx Inc., 2016.
- [Xil17] *UltraScale+ FPGAs*, XMP103, v1.10, Xilinx Inc., 2017.
- [Yoo13] Y. J. Yoon, N. Concer, M. Petracca and L. P. Carloni, “Virtual channels and multiple physical networks: Two alternatives to improve NoC performance,” in *IEEE Trans. Computer-Aided Design of Integr. Circuits and Syst.*, vol. 32, no. 12, pp. 1906-1919, Dec. 2013.

DOROBEK PUBLIKACYJNY AUTORA

Cytowania prac autora (stan we wrześniu 2019)

Ogółem 183 cytowania, indeks h równy 8 (według systemu Google Scholar), w tym między innymi:

- 22 cytowania w czasopismach IEEE Transactions / Access,
- 44 cytowania w innych czasopismach o zasięgu międzynarodowym,
- 72 cytowania w materiałach konferencji międzynarodowych,
- 7 cytowań w patentach US,
- 6 cytowań w książkach,
- 11 cytowań w rozprawach doktorskich,
- 12 cytowań w dokumentach ISO.

Publikacje autora

Czasopisma o zasięgu międzynarodowym

- [Dom13b] M. Domański, O. Stankiewicz, K. Wegner, M. Kurc, J. Konieczny, J. Siast, J. Stankowski, R. Ratajczak and T. Grajek, "High efficiency 3D video coding using new tools based on view synthesis," *IEEE Trans. on Image Processing*, vol.22, no. 9, pp. 3517-3527, 2013.
- [Łucz11] A. Łuczak, M. Stępniewska, J. Siast, M. Domański, O. Stankiewicz, M. Kurc and J. Konieczny, "Network-on-multi-chip (NoMC) with monitoring and debugging support," *J. Telecommun. and Inform. Techno.*, no. 3, pp. 81, 2011.
- [Sia16] J. Siast, J. Stankowski and M. Domański, "Hierarchical fast selection of intraframe prediction mode in HEVC," *Int. Journal of Electronics and Telecommunications*, vol. 62, no. 2, pp. 147-151, 2016.
- [Sia19] J. Siast, A. Łuczak and M. Domański, "RingNet: A memory-oriented network-on-chip designed for FPGA," in *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 6, pp. 1284-1297, 2019.
- [Stanko15] J. Stankowski, D. Karwowski, T. Grajek, K. Wegner, J. Siast, K. Klimaszewski, O. Stankiewicz and M. Domański, "Analysis of compressed data stream content in HEVC video encoder," *Int. Journal of Electronics and Telecommunications*, vol. 61, no. 2, pp. 121-127, 2015.

Polskie czasopisma

- [Dom13a] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski and K. Wegner, "Poznański kodek obrazów trójwymiarowych," *Przegląd Telekomunikacyjny*, no. 2-3, pp. 81-83, February/March 2013.
- [Łucz10] A. Łuczak, M. Kurc, M. Stępniewska and J. Siast, "Interfejs komunikacyjny dla układów FPGA serii Virtex," *Pomiary Automatyka Kontrola*, PAK, pp. 749-751, 2010.

Materiały konferencyjne

- a. Międzynarodowe konferencje (materiały opublikowane w bazie IEEE Xplore i indeksowane w bazie Web of Science):

- [Dom12a] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski and K. Wegner, "New coding technology for 3D video with depth maps as proposed for standardization within MPEG," in *Proc. 19th Int. Conf. on Syst., Signals and Image Processing (IWSSIP)*, 2012, pp. 401-404.
- [Dom12b] M. Domański, T. Grajek, D. Karwowski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski and K. Wegner, "Coding of multiple video+depth using HEVC technology and reduced representations of side views and depth maps," in *Proc. Picture Coding Symposium (PCS)*, 2012, pp. 5-8.
- [Dom12c] M. Domański, J. Konieczny, M. Kurc, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski and K. Wegner, "3D video compression by coding of disoccluded regions," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2012, pp. 1317-1320.
- [Dom14a] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz and K. Wegner, "Experiments on acquisition and processing of video for free-viewpoint television," in *Proc. 3DTV Conf. 2014*, Budapest, Hungary, 2-4 July 2014, pp. 1-4.
- [Dom15] M. Domański, J. Konieczny, M. Kurc, A. Łuczak, J. Siast, O. Stankiewicz and K. Wegner, "Fast depth estimation on mobile platforms and FPGA devices," in *Proc. 2015 3DTV-Conf.: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Lisbon, 2015.
- [Łucz14] A. Łuczak, S. Maćkowiak and J. Siast, "Depth map's 2D histogram assisted occlusion handling in video object tracking," in *Proc. Int. Conf. on Computer Vision and Graphics ICCVG 2014*, Warsaw, Poland, 15-17 September 2014, pp. 400-408.
- [Sia13] J. Siast, J. Stankowski, T. Grajek and M. Domański, "Digital watermarking with local strength adjustment for AVC-compressed HDTV bitstreams," in *Proc. 30th Picture Coding Symposium, PCS 2013*, San Jose, CA, USA, 8-11 December 2013, pp. 53-56.

- [Stanko12] J. Stankowski, M. Domański, O. Stankiewicz, J. Konieczny, J. Siast and K. Wegner, "Extensions of the HEVC technology for efficient multiview video coding," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2012, pp. 225-228.
- [Stanko14] J. Stankowski, D. Karwowski, T. Grajek, K. Wegner, J. Siast, K. Klimaszewski, O. Stankiewicz and M. Domański, "Bitrate distribution of syntax elements in the HEVC encoded video," in *Proc. Int. Conf. on Signal and Electronics Syst., ICSES 2014*, Poznań, Poland, 11-13 September 2014, pp. 1-4.
- [Stę10a] M. Stępniewska, O. Stankiewicz, A. Łuczak and J. Siast, "Embedded debugging for NoCs," in *Proc. 17th Int. Conf. on Mixed Design of Integrated Circuits and Syst.*, Wrocław, Poland, 24-26 June 2010, pp. 601-606.
- [Stę10b] M. Stępniewska, A. Łuczak and J. Siast, "Network-on-Multi-Chip (NoMC) for multi-FPGA multimedia systems," in *Proc. 13th Euromicro Conf. on Digital Syst. Design*, Lille, France, 1-3 September 2010, pp. 475-481.

b. Materiały konferencyjne grupy ekspertów *Motion Picture Experts Group* (MPEG):

- [Dom11a] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski and K. Wegner, "Multiview HEVC – experimental results," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, MPEG 2011 / M22147, Geneva, Switzerland, 28 Nov. - 02 Dec. 2011.
- [Dom11b] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski and K. Wegner, "Technical description of Poznan University of Technology proposal for call on 3D video coding technology," ISO/IEC JTC1/SC29/WG11, MPEG 2011 / M22697, Geneva, Switzerland, 28 Nov.-02 Dec. 2011.
- [Dom14b] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz and K. Wegner, "Poznan blocks – a multiview video test sequence and camera parameters for free viewpoint television," ISO/IEC JTC1/SC 29/WG11, MPEG2014, Doc. m32243, San Jose, USA, 2014.
- [Sia12] J. Siast, O. Stankiewicz, K. Wegner and M. Domański, "Independent intra-period coding in 3D-HTM," Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, a0091, Stockholm, Sweden, 16-20 July 2012.
- [Stanki12a] O. Stankiewicz, K. Wegner and J. Siast, "3D-CE2h results on Adaptive Depth Quantization combined with Nonlinear Depth Representation," ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m25022, Geneva, Switzerland, 30 April-4 May 2012.
- [Stanki12b] O. Stankiewicz, K. Wegner and J. Siast, "3D-CE2h results on Adaptive Depth Quantization combined with Nonlinear Depth Representation," ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m25022, Geneva, Switzerland, 30 April-4 May 2012.

- [Stanki12c] O. Stankiewicz, K. Wegner and J. Siast, “3D-CE2a results on Nonlinear Depth Representation,” ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m25017, Geneva, Switzerland, 30 April-4 May 2012.
- [Weg12a] K. Wegner, J. Siast, J. Konieczny, O. Stankiewicz and M. Domański, “Poznan University of Technology tools for 3DV coding integrated into 3D-HTM,” ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m23783, San Jose, USA, 6-10 February 2012.
- [Weg12b] K. Wegner, O. Stankiewicz and J. Siast, “3D-CE2h results on Nonlinear Depth Representation,” ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m25020, Geneva, Switzerland, 30 April-4 May 2012.
- [Weg12c] K. Wegner, O. Stankiewicz and J. Siast, “3D-CE1h results on Depth Map Disocclusion Coding by Poznan University of Technology,” ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m25014, Geneva, Switzerland, 30 April-4 May 2012.
- [Weg12d] K. Wegner, O. Stankiewicz and J. Siast, “3D-CE1h cross check of RWTH University proposal on Warping Based Prediction by Poznan University of Technology,” ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m25187, Geneva, Switzerland, 30 April-4 May 2012.
- [Weg12e] K. Wegner, O. Stankiewicz and J. Siast, “3D-CE2a cross check of Samsung proposal on Adaptive Depth Quantization by Poznan University of Technology,” ISO/IEC JTC1/SC29/WG11 MPEG 2012, Doc. m25018, Geneva, Switzerland, 30 April-4 May 2012.
- [Weg12f] K. Wegner, O. Stankiewicz J. Siast and M. Domański, “Independent intra-period coding in HEVC,” ISO/IEC JTC1/SC 29/WG11, Doc. JTCVC-K0332 11th Meeting: Shanghai, CN, 10–19, 2012.

Patenty

- [Dom14c] M. Domański, K. Klimaszewski, J. Konieczny, M. Kurc, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski and K. Wegner, “Image coding method,” Patent office: USPTO, Status: granted, Application number: US 13/680652, Filling date: 19.11.2012, Publication number: US 2013/0129235A1, Publication date: 23.05.2013, Patent number: US 8761527, Date of Patent: 24.06.2014.
- [Łucz17] A. Łuczak, S. Maćkowiak, M. Domański, J. Siast and T. Grajek, “A system and a method for tracking objects,” Patent office: USPTO, Status: granted, Application number: US 14/664862, Filling date: 22.03.2015, Patent number: US 9672634, Date of Patent: 06.06.2017.
- [Sia14] J. Siast and M. Domański, “A method and a system for video signal encoding and decoding with motion estimation,” Patent office: EPO, Status: granted, Application number: EP.13180012, Filling date: 12.08.2013, Publication number: EP 2699001A1, Publication date: 19.02.2014, Patent number: EP.2699001, Date of Patent: 06.05.2015.