

Robert TOMASZEWSKI, Stanisław DENIZIAK

POLITECHNIKA ŚWIĘTOKRZYSKA,
Al. 1000-lecia PP 7, 25-314 Kielce

Synteza bezkolizyjnych sieci jednoukładowych dla systemów wbudowanych

Mgr inż. Robert TOMASZEWSKI

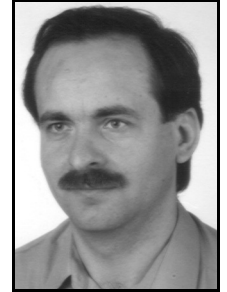
Ukończył studia na Wydziale Elektrotechniki, Automatyki i Informatyki Politechniki Świętokrzyskiej. Jest asystentem w Katedrze Informatyki WEAiI na Politechnice Świętokrzyskiej w Kielcach. Jego zainteresowania naukowe to sieci komputerowe, protokoły komunikacyjne, architektury systemów wbudowanych typu Network-on-Chip.



e-mail: r.tomaszewski@tu.kielce.pl

Dr hab. inż. Stanisław DENIZIAK

Ukończył studia na Wydziale Elektroniki Politechniki Warszawskiej, obronił pracę doktorską w 1994 r., a habilitacyjną w 2006r. Jest profesorem nadzwyczajnym w Katedrze Informatyki Politechniki Świętokrzyskiej oraz w Katedrze Informatyki Technicznej Politechniki Krakowskiej. Jego zainteresowania naukowe to metody szybkiego prototypowania systemów informatycznych, projektowanie systemów wbudowanych, testowanie i diagnostyka systemów cyfrowych.



e-mail: pedenizi@cyf-kr.edu.pl

Streszczenie

Praca prezentuje metodologię generowania dedykowanych topologii bezkolizyjnych dla systemów wbudowanych, realizowanych w architekturze sieci jednoukładowych (ang. *Network on Chip*, NoC). Kombinacja uszeregowania zadań i transmisji systemu z nowym sposobem doboru topologii i routingu w sieci NoC eliminuje kolizje transmisyjne, zapewniając jednocześnie spełnienie ograniczeń czasowych nałożonych na projektowaną aplikację. Przeprowadzone eksperymenty dowodzą przewagi zaprezentowanego rozwiązania nad typowymi metodami budowania sieci NoC.

Słowa kluczowe: systemy wbudowane, sieci jednoukładowe NoC, generowanie topologii, szeregowanie zadań i transmisji.

Contention-free and application-specific Network-on-Chip generation for embedded systems

Abstract

Although Network-on-Chip (NoC) [1] architectures present an interesting alternative to bus-based multiprocessor systems-on-chip (MPSoCs), they also need some effort to tackle communication contention problem. In this work the authors propose an efficient approach to application-specific irregular topology generation for contention-free NoC. There is taken advantage of a priori knowledge of the communication characteristic of the application (embedded system) to perform computation, communication scheduling and route generation with regard to performance constraints (Fig. 1). In the result there is obtained customized and minimal topology. The clear benefits of the authors' approach are: complete removal of network contention, simple router architecture without virtual channels, minimal network topology (in term of links necessary to guarantee contentionless communication), which meets system latency constraint, and input model of the system derived from co-synthesis of the embedded systems domain [6]. As demonstrated through experiments, this approach achieves far better performance than typical, random mesh networks and is comparable with well-known bandwidth-based solutions [3] (Tables 1, 3). In the resource-requirement field (interconnection fabric, Table 2) the authors' NoC is more saving than meshes (on average of 35%) and bandwidth-based custom topologies (on average of 25%).

Keywords: embedded systems, Network-on-Chip, topology generation, computation and communication scheduling.

1. Wstęp

Sieci jednoukładowe NoC [1] stały się niezwykle interesującą alternatywą dla projektowania systemów wieloprocesorowych, w tym dla systemów wbudowanych. Ich skalowalność odniesiona do zapotrzebowania energetycznego dalece przewyższa rozwiązania oparte na wspólnej magistrali czy typu *full interconnect* [2]. Mikrościeć NoC zbudowana jest z szeregu elementów przetwarzających PE (ang. *Processing Elements*), zwanych również ogólnie procesorami, komunikujących się ze sobą za pomocą połączonych w rozmaite topologie ruterów. Jednym z najpoważniejszych wyzwań podczas projektowania takich architektur jest zapobieganie

kolizjom transmisji. Wyróżniamy trzy rodzaje kolizji: na porcie źródłowym (jednoczesna próba nadania więcej niż jednego komunikatu przez ten sam procesor), na porcie docelowym (do jednego odbiorcy wędruje naraz więcej niż jedna wiadomość) oraz na trasie (fragment mikrosieci – łącza – jest przydzielony w tym samym czasie kilku transmisjom). Typowe podejścia do zapobiegania konfliktom to projektowanie NoC z tzw. rezerwacją pasma [3]. Polega ono na różnicowaniu przepustowości tras w sieci (inne parametry łącza) oraz umieszczeniu blisko siebie procesorów intensywnie komunikujących się ze sobą. Powstają w ten sposób mikrosieci wysoce nieregularne. Routing dynamiczny również nie rozwiązuje sprawy – ma tendencję do koncentrowania ruchu w centrum sieci [1,4], a rutery wykazują wysoki stopień skomplikowania. Z kolei rozwiązania ruterów z trasowaniem statycznym oparte na koncepcji kanałów wirtualnych charakteryzują się wysokim zapotrzebowaniem na zasoby układu scalonego [5].

W niniejszej pracy zaprezentowano inne podejście, bazujące na przewidywalności wzorca komunikacyjnego w systemie wbudowanym. Poprzez analizę specyfikacji systemu można przewidzieć wszystkie konflikty w mikrosieci NoC. Kolizje na portach usuwane są za pomocą odpowiedniego uszeregowania zadań i transmisji aplikacji – zajmuje się tym etap kosyntezy. Konflikty na trasie rozwiązywane są dedykowaną topologią połączeń z odpowiednio dobranymi ścieżkami transmisji.

Zaletami przedstawionej metodologii są: całkowita eliminacja kolizji, bardzo prosta konstrukcja użytego do budowy sieci NoC rutera, minimalna (w sensie użytych łączy międzyruterowych) topologia spełniająca zadane ograniczenia czasowe projektowanej aplikacji, model wejściowy systemu wbudowanego o ugruntowanej pozycji w dziedzinie kosyntezy [6].

Metoda przeznaczona jest dla systemów wbudowanych przetwarzających dane w sposób rozproszony (wieloprocesorowe) i przekazujące informacje między jednostkami wykonawczymi PE za pomocą komunikatów.

2. Model systemu i schemat metody

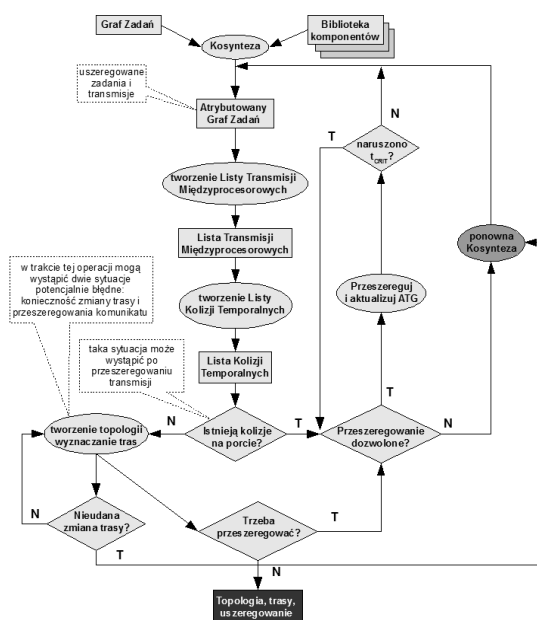
Dla potrzeb odwzorowania danego systemu wbudowanego w architekturę NoC stworzono kompleksowy model, jak również przyjęto szereg założeń. Bazowano przy tym na istniejących pracach z dziedziny kosyntezy oraz projektowania sieci jednoukładowych. W modelu najważniejszymi elementami są:

- **Atrybutowany Graf Zadań ATG** (ang. *Attributed Task Graph*) – wynik kosyntezy [7] przeprowadzonej dla aplikacji opisananej Grafem Zadań TG (ang. *Task Graph*) [6]; węzły ATG to zadania systemu z przypisanymi czasami przetwarzania na danym procesorze, mierzonymi w cyklach zegarowych, oraz z czasami rozpoczęcia wykonania zadania; krawędzie ATG to transmisje opisane wolumenem przesyłanych danych oraz czasem rozpoczęcia; ze względu na stosowany model rutera (znikome opóźnienia) założono, że czas przesłania dużych wiadomości (wzięto pod uwagę systemy wbudowane z intensywną

komunikacją miedzyzadaniową) jest równy jej wolumenowi, niezależnie od długości trasy,

- **ruter** – wyposażony w co najwyżej 4 dwukierunkowe porty międzyruterowe i jeden port lokalny do obsługi modułu PE, przełączanie typu *wormhole*, możliwość symultanicznego przesyłu wiadomości między różnymi portami rutera [1], bufor jednolity,
- **szeregowanie, przeszerogowanie** – szeregowania zadań i transmisji dokonuje algorytm kosyntezy [7], uwzględniając przy tym eliminację kolizji na portach (kolizje na trasie nie mogą być usunięte w tym kroku, gdyż nie ma jeszcze informacji o topologii); przeszerogowania dokonuje opcjonalnie metodologia budowy mikrosieci – w ogólności polega ono na opóźnieniu czasu rozpoczęcia jednej z kolidujących transmisji i wymaga ponownego przeliczenia czasów rozpoczęcia dla elementów podgrafu zapoczątkowanej przeszerogowaną transmisją (krawędzią),
- **Graf Topologii Sieci NTG** (ang. *Network Topology Graph*) – rezultat działania metodologii; węzły odpowiadają procesorom (z przypisanymi im ruterami) systemu wbudowanego, a krawędzie – jednokierunkowym łączom między ruterami.

Na rys. 1 przedstawiono sieć działań podejmowanych w ramach syntezy sieci NoC.



Rys. 1. Schemat sieci działań dla proponowanej metodologii
Fig. 1. Design flow for the proposed methodology

W pierwszym kroku, wykonywana jest kosynteza systemu specyfikowanego w formie grafu TG. Kosynteza tworzy graf ATG zoptymalizowany pod kątem minimalnego kosztu użytych zasobów (elementy przetwarzające). Następnie z grafu ATG wyodrębniane są wszystkie transmisje międzyprocesorowe. Rezultatem jest **Lista Transmisji Międzyprocesorowych** ITL (ang. *Interprocessor Transmission List*). Każda pozycja na liście opisana jest informacjami o procesorach: nadawczym i odbiorczym, czasie rozpoczęcia, czasie trwania oraz trasie (początkowo pusta - NULL). Na podstawie ITL powstaje, stanowiąca sedno modelu kolizji, **Lista Kolizji Temporalnych** TCL (ang. *Transmission Collision List*). Elementami TCL są pary nachodzących na siebie w czasie wiadomości. W każdej parze najpierw umieszczany jest dłuższy ze skonfliktowanych komunikatów, a potem krótszy. Dzięki temu algorytm w dalszych krokach podejmuje korzystniejsze decyzje - wyznacza krótszą trasę lub dedykowane połączenie między modułami PE dla dłuższych transmisji. W kolejnym kroku sprawdza się obecność kolizji na portach wśród pozycji TCL. Mimo, że kosynteza szereguje komunikaty wykluczając tego typu kolizje, to mogą się one pojawić na skutek przeszerogowania podczas kolejnych iteracji algorytmu. Jeśli algorytm stwierdzi powstanie kolizji na portce – obojętnie: źródłowym lub docelowym – to jedynym sposobem jej usunięcia jest przeszerogowanie. Może ono dopro-

wadzić do naruszenia któregoś z zadanych przez projektanta systemu ograniczeń czasowych aplikacji (t_{CRIT}), co skutkuje próbą przeszerogowania drugiej z kolidujących wiadomości. Porażka i w tym przypadku doprowadzi do ponownej kosyntezy z dodatkowym ograniczeniem czasowym wymuszającym wcześniejsze uszerogowanie transmisji powodującej naruszenie t_{CRIT} . Algorytm przeszerogowania opóźnia tą z wiadomości, która w mniejszym stopniu pogarsza czasy zakończenia zadań z ustanowionymi ograniczeniami czasowymi (ale bez ich przekroczenia). W ostatnim kroku budowana jest topologia połączeń dla architektury NoC. Dla każdej pozycji z listy TCL wykonywane są poniższe testy i podejmowane stosowne działania:

- 1) Jeżeli dla danych transmisji nakładających się w czasie już przypisano trasy (usunięto inne kolizje, w których brały udział rozpatrywane komunikaty) dokonywane jest sprawdzenie czy nie są to kolidujące trasy, jeśli tak - podejmowana jest próba modyfikacji (często – pogorszenia, wydłużenia) trasy dla krótszej z transmisji, a w przypadku niepowodzenia – dla dłuższej; zmiana wyznaczonych tras wymaga sprawdzenia bezkolizyjności uprzednio usuniętych konfliktów, niepowodzenie tej operacji prowadzi do ponownej kosyntezy; jednocześnie przekazywana jest informacja o zadaniu-nadawcy krótszego z komunikatów (wraz z czasem o jaki przekroczone ograniczenie czasowe), dla którego nie udało się zmienić trasy – zadanie takie podczas ponownej kosyntezy będzie miało ustanowione bardziej rygorystyczne ograniczenie czasowe (skrócone o przekazany czas).
- 2) Jeżeli dla danej transmisji nie wyznaczono dotąd trasy to sprawdza się czy w już utworzonej topologii istnieje najkrótsza droga dla komunikatu; to samo dotyczy drugiej kolidującej transmisji; w przypadku kolizji tras drugi komunikat będzie miał wyznaczoną inną drogę niż najkrótsza.
- 3) Niepowodzenie poprzedniego kroku skutkuje próbą wygenerowania kolejnego, dedykowanego połączenia międzyruterowego; rozbudowa topologii podlega ograniczeniom – jeden ruter może mieć nie więcej niż 4 łącza wejściowe i 4 wyjściowe.
- 4) Jeżeli poprzednie kroki okazały się niewystarczające dla danej pozycji na liście TCL (nie da się rozwiązać danego konfliktu za pomocą dedykowanej topologii) podejmowana jest próba przeszerogowania komunikatów opisana wcześniej – jej niepowodzenie spowoduje ponowną kosyntezę z ustanowieniem nowego, krótszego ograniczenia czasowego dla zadania-nadawcy mniejszej z transmisji.

Po przetworzeniu wszystkich pozycji z listy TCL sprawdza się, czy lista ITL zawiera węzły nieuwzględnione w TCL, czyli transmisje niekolidujące z żadnymi innymi. Jeśli tak, podejmowana jest próba wyznaczenia najkrótszej drogi dla każdej brakującej transmisji (może to również oznaczać konieczność wstawienia dodatkowego połączenia lub węzła do sieci).

3. Wyniki przeprowadzonych eksperymentów

Dla celów eksperymentalnych wygenerowano 10 syntetycznych aplikacji (grafów zadań) za pomocą narzędzia wzorowanego na TGFF [8]. Każdy z grafów ma 20-30 zadań komunikujących się za pomocą 15-30 transmisji. Następnie grafy zostały poddane kosyntezie metodą opisaną w [7] – zadania rozdzielono pomiędzy 5-9 elementów przetwarzających. W ostatnim kroku, metodami analitycznymi, porównano właściwości sieci NoC zbudowanej przy użyciu zaprezentowanej w artykule metodologii („DSB” - Dedykowana Sieć Bezkolizyjna) z kilkoma często spotykanymi w literaturze podejściami. W szczególności wzięto pod uwagę następujące warianty sieci jednokładowych: losowo zbudowana topologia siatki kwadratowej („mesh NxN”), losowo zbudowana topologia minimalnej siatki prostokątnej („mesh NxM”) oraz topologia nieregularna utworzona według zasady rezerwacji pasma („bndwdth”), podobnie do rozwiązań prezentowanych w [3]. Topologie „bndwdth” łączą ze sobą bezpośrednio (magistralami dwukierunkowymi) procesory wymieniające największe ilości danych. W mikrosieciach typu „mesh” (ruting typu XY) kolizje na trasie były usuwane wyłącznie poprzez przeszerogowanie,

natomiast w modelu „bndwidth” (ruting prekonfigurowany, niekoniecznie minimalny) priorytet miały dedykowane dodatkowe łącza jednokierunkowe. Aby porównanie różnych architektur NoC wygenerowanych dla tych samych aplikacji było wiarygodne założono, że jeden ruter obsługuje tylko jeden moduł przetwarzający (w mikrosieci nie występują rutery bez przyłączonego procesora). Ponadto wszystkie łącza międzyruterowe mają jednakową przepustowość. Tabela 1 prezentuje (procentowo) stopień pogorszenia szybkości pracy systemu w stosunku do rozwiązania po etapie kosyntezy, gdzie uszeregowanie pomijało możliwość wystąpienia konfliktu na trasie. W każdym przypadku przewaga rozwiązania „DSB” nad losowo wygenerowanymi „meshami” jest bezdyskusyjna. Ponadto w trzech przypadkach osiągnięto zerowe pogorszenie czasu wykonania systemu. Szybkość pracy sieci typu „bndwidth” jest taka sama jak „DSB”, gdyż obie są bezkolizyjne (brak przeszerogowań). Kolumna 2 („obl./trans.”) to stosunek sumarycznego czasu obliczeń do czasu transmisji dla danej aplikacji. Im mniejsza wartość, tym wyższe prawdopodobieństwo wystąpienia kolizji w systemie. Z kolei współczynnik z kolumny 3 - „kol./trans.” to stosunek czasu wszystkich kolizji do czasu wszystkich transmisji. Im wyższy, tym większy natłok w mikrosieci (więcej jednocześnie transmitowanych danych).

Tab. 1. Pogorszenie szybkości działania systemów względem wyniku kosyntezy
Tab. 1. Performance loss as compared to co-synthesis stage

Aplika- cja	obl./ trans.	kol./ trans.	DSB [%]	NxM mesh [%]	NxN mesh [%]
G01	0,54	0,35	7,08	12,39	12,39
G02	0,69	0,64	10,77	38,25	19,84
G03	0,36	0,48	7,79	16,09	13,25
G04	0,66	0,3	8,66	13,52	20,14
G05	1,32	0,41	9,86	20,8	15,44
G06	1	0,4	5,02	10,55	7,14
G07	0,58	0,5	0	9,34	10,17
G08	1,22	0,4	0	8	8
G09	1,83	0,25	0	0	3,64
G10	1,06	0,52	10,9	15,2	17,18

W tabeli 2 zestawiono zapotrzebowanie na infrastrukturę komunikacyjną, rozumianą jako liczba użytych do budowy danej topologii łączy jednokierunkowych. Stanowi to zgrubny wskaźnik co do zasobo- oraz energochłonności danego rozwiązania. Topologią referencyjną jest „DSB” z podaną liczbą łączy. Dla pozostałych rozwiązań podano (procentowo) stopień pogorszenia (wartość dodatnia) lub poprawy (wartość ujemna). Również na tym polu proponowane rozwiązanie („DSB”) prezentuje się dużo lepiej niż topologie „mesh” (średnia oszczędność sięga 35%), a także popularne podejścia bazujące na rezerwacji pasma („bndwidth”). Tylko w jednym przypadku „bndwidth” dało wynik lepszy od „DSB” i w jednym równie dobry (pogorszenie „0”). W pozostałych przypadkach średnia strata liczby łączy to ponad 25%.

Tab. 2. Zapotrzebowanie na łącza międzyruterowe dla badanych topologii
Tab. 2. Router links requirements for the examined topologies

Aplika- cja	DSB [łącza]	NxM mesh strata [%]	NxN mesh strata [%]	Bndwidth strata [%]
G01	9	35,71	62,5	18,18
G02	9	55	62,5	35,71
G03	9	35,71	62,5	-12,5
G04	6	57,14	75	45,45
G05	10	50	58,33	28,57
G06	9	35,71	62,5	10
G07	12	40	50	0
G08	13	45,83	45,83	27,78
G09	8	42,86	66,67	27,27
G10	13	35	45,83	13,33

Ostatnia tabela zawiera porównanie parametru hop_{AVG} , czyli średniej ważonej liczby skoków,

$$hop_{AVG} = \frac{\sum_1^n |M_{i,j}| * hop(M_{i,j})}{\sum_1^n |M_{i,j}|} \quad (1)$$

gdzie n to liczba wszystkich transmisji międzyprocesorowych w systemie, $|M_{i,j}|$ to czas trwania komunikatu $M_{i,j}$, zaś $hop(M_{i,j})$ to długość trasy (w ruterach).

Referencyjnym systemem jest DSB. Wyniki potwierdzają przewagę zaproponowanego rozwiązania nad topologiami siatek z rutingiem XY. Co ciekawe – mimo iż rozwiązania klasy „bndwidth” mają na celu minimalizację ruchu w mikrosieci poprzez skrócenie tras dla największych komunikatów to rozwiązanie DSB tylko w czterech przypadkach dało wyniki gorsze. W pozostałych wyniki są lepsze lub równie dobre co w przypadku podejścia „bndwidth”.

Tab. 3. Pogorszenie parametru hop_{AVG}
Tab. 3. Deterioration of the hop_{AVG} parameter

Aplika- cja	DSB hop_{AVG}	NxM mesh strata [%]	NxN mesh strata [%]	Bndwidth strata [%]
G01	2,28	17,09	19,72	4,2
G02	1,69	44,41	47,35	1,17
G03	1,25	51,55	52,11	3,85
G04	2,54	12,41	10,88	0
G05	2,46	19,34	29,91	-3,36
G06	2,51	11,93	13,75	-18,96
G07	1,86	41,32	40,76	1,59
G08	1,84	32,85	32,85	-11,52
G09	2,47	6,44	22,57	0
G10	1,36	53,9	50	-3,03

4. Wnioski

W pracy przedstawiono nowe podejście do problemu generowania sieci jednokierunkowych. Nacisk położono na bezkolizyjność osiąganą poprzez dedykowane łącza oraz odpowiednie do wymagań czasowych aplikacji uszeregowanie zadań i transmisji systemu. Zaproponowana metodologia projektowania architektur NoC znajduje zastosowanie dla systemów wbudowanych przetwarzających w sposób rozproszony o przewidywalnym wzorcu komunikacji. Przeprowadzone badania dowodzą przewagi opisanej metody nad typowymi, opartymi na topologii siatki architekturami NoC. Również porównanie z popularnym podejściem bazującym na rezerwacji pasma wypada na korzyść zaproponowanej metodologii. Uzyskano rozwiązania lepsze, jeśli chodzi o zapotrzebowanie na zasoby układu i porównywalne na polu wydajności.

W chwili obecnej trwają prace nad algorytmami efektywnego rozmieszczania elementów przetwarzających na powierzchni układu. Wyniki zostaną zintegrowane z opracowaną metodologią.

5. Literatura

- [1] Bjerregaard T., Mahadevan S.: A survey of research and practices of network-on-chip, ACM Computing Surveys, 38(1), 71-121, 2006.
- [2] Lee H.G., Chang N., Ogras U.Y., Marculescu R.: On-Chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches, ACM Transactions on Design Automation of Electronic Systems, 12(3), Article 23, 2007.
- [3] Murali S., Meloni P., Angiolini F., Atienza D., Carta S., Benini L., De Micheli G., Raffo L.: Designing application-specific networks on chips with floorplan information, Int. Conf. on CAD, 355 – 362, 2006.
- [4] Deniziak S., Tomaszewski R.: Adaptive routing protocols validation in NoC systems via rapid prototyping, Proceedings of the IEEE Human System Interaction, 115-120, 2008.
- [5] Schelle G., Grunwald D.: Exploring FPGA Network on Chip implementations across various application and network loads, Intl Conference on Field Programmable Logic and Application, 41-46, 2008.
- [6] Staunstrup J., Wolf W.: Hardware/software co-Design: Principles and practice, Kluwer Academic Publishers, Norwell, MA, USA, 1997.
- [7] Deniziak S., Górski A.: Hardware/software co-synthesis of distributed embedded systems using genetic programming, Lecture Notes in Computer Science, Springer, 5216, 83-93, 2008.
- [8] Dick R.P., Rhodes D.L., Wolf W.: TGFF: Task graphs for free, Int. Workshop on Hardware/Software Codesign, 97-101, 1997.