Journal of Systems Smartening and Data Processing
Vol. 2 - No. 1 - Spring 2024

Congestion- and Energy-aware Run-time Task Mapping for Network-on-Chip Architecture

Mohammad Ahmadpour[1], *MSc,* Farhad Rad[2], *Assistant Professor*, Davood Heidarnasab[2], *MSc*

1-Faculty of Engineering, Yasuj Branch, Islamic Azad University, Yasuj, Iran
Mohammad_t200@yahoo.com

2-Faculty of Engineering, Yasuj Branch, Islamic Azad University, Yasuj, Iran
Fa.rad@iau.ir

3-Faculty of Engineering, Kazerun Branch, Islamic Azad University, Kazerun, Iran
Heidarnasab.b@gmail.com

Received: 13 January 2024 **Revised:** 07 April 2024 **Accepted:** 02 May 2024

ABSTRACT

Today, with the advancement of semiconductor technology, the number of processing components on the chip has increased. Networks on-chip or NoC has been proposed as an efficient and scalable communication scheme within chips to overcome bus problems. One of the major challenges for NoCs is mapping tasks on processing cores. Given that the optimal solution to the mapping problem is an NP-hard problem, the proposed scheme provides an efficient mapping of tasks using the particle swarm optimization algorithm. In order to be able to prevent energy consume in the infrastructure of Networks on-chip, in addition reducing latency, which is in fact one of the objectives of this study. The particle structure in this algorithm is like an array of switches in a mesh topology. The value of each particle represents a core of the communication graph. Finally, the proposed solution in the simulation environment was compared and evaluated in comparison with other common solutions. The results of different scenarios indicate that the method presented through appropriate mapping, has been able to reduce the amount of latency as well as energy consumption, which is due to the use of a solution based on particle swarm optimizing in mapping operations.

Keywords: Network on chip, task mapping, particle swarm optimization.

Corresponding Author: Dr. Farhad Rad
Corresponding Author Address: Department of Electrical and Computer Engineering- Kilometer 14 Sisakht road- Yasuj Branch -Islamic Azad University- Iran**.**

ارائه روشی آگاه از انرژی و ازدحام برای نگاشت در شبکه بر روی تراشه

محمد احمدی پور[1]، کارشناسی ارشد، فرهاد راد [2*]، استادیار، داود حیدرنسب[3]، کارشناسی ارشد

۱-دانشکده فنی مهندسی، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران
Mohammad_t200@yahoo.com

۲- گروه مهندسی پزشکی، دانشکده فنی مهندسی، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران
Fa.rad@iau.ir

۳- دانشکده فنی مهندسی، واحد کازرون، دانشگاه آزاد اسلامی، کازرون، ایران
Heidarnasab.b@gmail.com

تاریخ ارسال مقاله: ۱۴۰۲/۱۰/۲۳ **تاریخ بازنگری مقاله:** ۱۴۰۳/۰۱/۱۹ **تاریخ پذیرش مقاله:** ۱۴۰۳/۰۲/۱۳

چکیده: امروزه با پیشرفت فناوری نیمه‌هادی‌ها، تعداد مولفه‌های پردازشی روی تراشه افزایش یافته است. معماری ارتباطی این سیستم‌ها مبتنی بر گذرگاه می‌باشد. شبکه روی تراشه یا NOC به عنوان یک طرح ارتباطی درون تراشه‌ای جهت غلبه بر مشکلات گذرگاه‌ها مطرح شده است. یکی از چالش‌های مهم مربوط به NOCها، نگاشت وظایف بر روی هسته‌های پردازشی می‌باشد. با توجه به این که حل بهینه مسئله نگاشت یک مسئله NP-hard است، در طرح پیشنهادی یک نگاشت کارای وظایف به کمک الگوریتم بهینه‌سازی ازدحام ذرات ارایه شده است بطوریکه بتوان از طریق بکارگیری آن، علاوه بر کاهش میزان تاخیر از اتلاف انرژی، در زیرساخت شبکه‌های روی تراشه جلوگیری کرد که در واقع از اهداف این پژوهش می‌باشد. ساختار ذرات در این الگوریتم مانند آرایه هایی از سوئیچ‌ها در همبندی توری می‌باشد. مقدار هر ذره بیانگر یک هسته از گراف ارتباطی می‌باشد. در انتها راهکار پیشنهادی در محیط شبیه‌ساز در مقایسه با سایر راهکارهای متداول مورد بررسی و ارزیابی قرار گرفت. نتایج حاصل از سناریوهای مختلف، بیانگر آن است که روش ارایه شده از طریق نگاشت‌های مناسب، بخوبی توانسته است میزان تاخیر و همچنین انرژی مصرفی را کاهش دهد که این امر به دلیل بکارگیری راهکار مبتنی بر بهینه‌سازی ازدحام ذرات در عملیات نگاشت وظایف می‌باشد.

کلمات کلیدی: شبکه روی تراشه، نگاشت وظایف، بهینه سازی ازدحام ذرات

نام نویسنده‌ی مسئول: دکتر فرهاد راد
نشانی نویسنده‌ی مسئول: یاسوج- کیلومتر۱۴ جاده سی سخت- دانشگاه آزاد اسلامی واحد یاسوج- دانشکده مهندسی برق و کامپیوتر

۱- مقدمه

شبکه بر روی تراشه یک زیرسیستم ارتباطی درون یک مدار مجتمع (که به‌طورمعمول یک «تراشه» نامیده می شود) می باشد که نوع ارتباط بین هسته هایIP در یک سیستم درون یک تراشه (SoC) را فراهم می‌کند. تکنولوژی شبکه روی تراشه، تئوری شبکه و روش‌های ارتباط درون تراشه ای را به کار می برد و پیشرفت‌های قابل توجهی را نسبت به اتصالات بر پایه گذرگاه و کراس‌بار اولیه به همراه می آورد[۱]. در واقع شبکه بر روی تراشه جایگزین سیستم‌های ارتباطی مبتنی بر گذرگاه معرفی شده‌اند و همچنین برای مقابله با پیچیدگی در سیستم‌های نهفته طراحی شده‌اند. تراشه‌های مبتنی بر گذرگاه در ابعاد بزرگتر قابلیت انجام با عملکرد بالا را ندارند، بنابراین الگوی شبکه روی تراشه در عصر حاضر یا درآینده برای سیستم‌های چندپردازنده ای مطرح شده است و تمام محققان تلاش می‌کنند تا کارآیی و قابلیت اطمینان را در اینگونه سیستم ها بالا ببرند و مصرف توان و تاخیر را کم کنند[۲]. کاهش مصرف انرژی و تاخیر یک مساله ی بسیار مهم در طول طراحیی ک شبکه بر روی تراشه است.با ادغام هرچه بیشتر مدارهای الکترونیکی مقیاس آنها افزایش یافته درنتیجه احتمال افزایش ترافیک و ازدحام در سیستم افزایش می‌یابد. نگاشت از مراحل بسیار مهم در طراحی شبکه روی تراشه می باشد که در وهله‌ی اول شامل تخصیص وظایف به هسته‌ها است. یک راه نگاشت که استفاده می شود پیدا کردن بهترین جایگذاری هسته‌ها در گره‌های توپولوژی شبکه روی تراشه می‌باشد. یک تکنیک نگاشت محدودیت های زیادی دارد از جمله زمان ارتباطی، مصرف توان ، تاخیر یا توان عملیاتی[۳]. راه حل نگاشت عمدتاً ابتکاری هستند که می‌توانند در طبقه‌بندی‌های مختلف دسته‌بندی شوند. نگاشت به دسته‌ی هسته‌های یکسان و انواع مختلف از هسته‌ها طبقه‌بندی می‌شود. پدیده های مختلف از قبیل تکنولوژی مقیاس پذیری، تغییر پذیری پردازش ها، باعث بالا رفتن نیازها در قابلیت اطمینان در شبکه بر تراشه شده است که در برابر خطاهایی که در سیستم اتفاق می‌افتد بتواند، سیستم را پایدار نگاه دارد. برهمین اساس در این پایان نامه، به مطالعه الگوریتم‌های نگاشت در شبکه روی تراشه و سیستم های چندپردازنده ای در یک تراشه می‌پردازیم وکارایی، مصرف انرژی و کاهش ازدحام هر یک را در شرایط مختلف کاری بررسی می‌کنیم، سپس با توجه به ضرورت مساله نگاشت، یک الگوریتم نگاشت در شبکه روی تراشه معرفی می شود[۴]. انتظار داریم الگوریتم پیشنهادی تاثیر بسزایی در کاهش مصرف انرژی شبکه داشته باشد و با درنظر گرفتن محدودیت های زمانی و هزینه ارتباطی داخلی شبکه، تاثیر مطلوبی در کاهش ترافیک شبکه و در نتیجه کاهش تاخیر بسته‌های ارسالی در شبکه را به همراه داشته باشد. باس ارتباطی شبکه در سیستم روی تراشه‌های سنتی مبتنی بر سیم‌های اختصاصی یا باس‌های به اشتراک گذاشته شده است. در این روش در هر لحظه، فقط یک تراکنش ارتباطی در باس امکان پذیر است درنتیجه پهنای باند ارتباط و مقیاس‌پذیری محدود می‌شود[۵]. بنابراین برای سیستم روی تراشه‌های بزرگ مناسب نیست و شبکه بر روی تراشه روش خوبی برای کاهش این مشکلات است. یک شبکه بر تراشه یک معماری کاشی محور منظم است که زیرساخت ارتباطی کارآمد در هسته IP فراهم می‌کند. نگاشت هسته IP روی یک بستر داده شده (موردنظر)، فاز مهمی در طراحی شبکه روی تراشه است که به‌طور قابل ملاحظه‌ای روی مصرف انرژی، کارایی شبکه و هزینه سخت‌افزاری بستر تاثیر می‌گذارد. بر همین اساس روش‌ها و الگوریتم‌های متفاوتی نیز برای حل این مساله ارائه شده است[۶]. به دلیل اینکه مساله نگاشت جزء مسائل NP-Hard است و بدست آوردن بهترین جواب هزینه بالایی دارد، اکثر الگوریتم‌های ارائه شده اکتشافی هستند.روش‌های اکتشافی یا فرا اکتشافی به طور گسترده در مسائل مختلف اعمال می‌شوند. این روش‌ها کمک می‌کنند تا به راه‌حلهای بهینه یا نزدیک به بهینه در مسائل بزرگ بازمان منطقی دست یافته شود. بر همین اساس در این پژوهش نیز سعی بر ارایه یک راهکار فرا اکتشافی به منظور بهبود نگاشت وظایف می شود تا از طریق بکارگیری آن بتوان میزان تاخیر و انرژی را در این شبکه ها کاهش داد[۷].

۲- روش شناسی تحقیق

در این مقاله یک نگاشت بهینه وظایف به کمک الگوریتم بهینه‌سازی ازدحام ذرات ارایه می‌شود بطوریکه بتوان از طریق بکارگیری آن علاوه بر کاهش میزان تاخیر بتوان اتلاف انرژی را نیز در زیرساخت شبکه‌های روی تراشه کاهش داد که در واقع از اهداف اصلی این مقاله می‌باشند.

نگاشت وظایف که در این مقاله ارایه می‌شود بر پایه الگوریتم بهینه‌سازی ازدحام ذرات عمل می‌کند. ساختار ذرات در این الگوریتم مانند آرایه‌ای از سوئیچها در همبندی توری می‌باشد. مقدار هر ذره بیانگر یک هسته از گراف ارتباطی می‌باشد. یکی از پارامترهای مهم در نگاشت، هزینه ارتباطی می‌باشد. از آنجاکه نگاشت بر روی گراف بر روی کاربردهای خاص انجام می‌شود در ادامه این کاربردها مورد بررسی قرار می‌گیرد. در واقع نگاشت تصمیم می‌گیرد که هر یک از وظایف کاربرد بر روی کدام یک از هسته‌های تراشه اجرا شود. برای این منظور برنامه کاربردی که قرار است بر روی تراشه اجرا شود بصورت گراف وزن‌دار و جهت‌دار توصیف می شود که هر گره آن نشانگر یک وظیفه خاص و جهت یال مشخص کننده چگونگی ارتباطات منطقی بین وظایف هستند[۸]. وزن یال ها هم می‌تواند معرف پارامترهای مختلفی از قبیل پهنای باند، انرژی مصرفی و یا نرخ داده انتقالی باشد. سپس عمل نگاشت با توجه به پارامترهای مورد توجه انجام می‌شود. مفهوم نگاشت این است که به هر وظیفهt_i باید قسمتی از بستر شبکه روی تراشه تخصیص داده شود تا وظیفه مورد نظر اجرا گردد و در همبندی مش هر خانه همبندی که شامل یک منبع استمی‌تواند به یک یا چند وظیفه تخصیص یابد. ساختار شبکه بر تراشه شامل زیر ساخت از سوئیچها و ار تباطات مابین آنها می باشد که این سوئیچ ها توسط یک شبکه از پیوندها در توپولوژی مشبه هم متصل است و در ادامه کاربرد مورد نظر رابه قطعاتی تقسیم کرده و منابع شبکه روی تراشه به آنها اختصاص داده می‌شود و مسیریابی به صورتی انجام میشود تا انرژی مصرفی کل کاهش یابد. عملکردی که کاربر یا طراح از یک سیستم انتظار دارد کاربرد نامیده می‌شود. برای اجرای یک یا چندین کاربرد بر روی یک زیر ساخت شبکه روی تراشه در ابتدا باید مفهوم کاربرد به صورت یک مدل ریاضی مطرح شود و از سوی دیگر طریقه ای برای مشخص کردن زیر ساخت مشخص شود. همچنین معیاری جهت سنجش بهینگی لازم است. برای فرموله کردن مسائل نگاشت و ظایف در معماری شبکه روی تراشه تعریف‌های زیر ضروری می‌باشند.

۲-۱- شبکه‌های روی تراشه

در طول دهه ۱۹۹۰ هسته‌های پردازشی و کامپونت‌های بیشتری بر روی یک چیپ سیلیکونی ادغام شدند. مشکل اصلی که در این حوزه وجود داشت و امروزه نیز کماکان وجود دارد؛ استانداردسازی رابط‌های مربوط به مولفه‌ها و اعتبارسنجی کلی سیستم با توجه به مشخصه‌های عملکردی می‌باشد. در چنین سیستم‌هایی ارتباط بین پردازنده‌ها اهمیت زیادی پیدا می‌کند و سیستم بر روی تراشه (SoC) و شبکه روی تراشه (NoC) دو تکنیک پیاده‌سازی برای این سیستم‌ها می‌باشند[۵].

به تعدادی از عناصر پردازشی که بر روی یک تراشه واقع شده اند. سیستم روی یک تراشه یا به اختصار SoC گفته می شود. از آنجا که ارتباطات SoC مبتنی بر bus می باشند این امر موجب می شود تا با افزایش انواع عناصر پردازشی، ساختار SoC نیز بسیار پیچیده تر شود. به منظور کاهش این پیچیدگی بهتر است که از شبکه روی تراشه یا به اختصار (NoC) استفاده شود؛ زیرا NoC امکان استفاده از ارتباطات موازی سطح بالا را فراهم می آورد و موجب بهبود کارایی ارتباطات مبتنی بر تراشه نیز می شود. منظور از توپولوژی, اتصالات داخلی شبکه است.شبکه درون تراشه ای یا شبکه روی تراشه یک زیرسیستم ارتباطی درون یک مدار مجتمع (که به طور معمول یک "تراشه" نامیده می شود) است که نوعا ارتباط بین هسته های IP در یک سیستم درون یک تراشه را فراهم می سازد NoC ها می توانند دامنه های ساعت همزمان و ناهمزمان را پوشش دهند یا منطق ناهمزمان بدون ساعت را استفاده کنند. تکنولوژی NoC تئوری شبکه و روش های ارتباط درون تراشه ای را به کار می برد و پیشرفتهای قابل توجه ای را نسبت به اتصالات بر پایه گذرگاه و کراس بار (crossbar) اولیه به همراه می آورد Noc . مقیاسپذیری SoC را بهبود می دهد و استفاده از انرژی در SoC های پیچیده را نسبت به طرح های دیگر بهینه می سازد. در NoCبرای اتصال هسته به سایر بخش هاء تعدادی توپولوژی وجود دارد. شبکه روی تراشه یکی از طراحی‌های اساسی تراشه است و دارای مزایا و چالش‌هایی هست، به‌ویژه زمانی که تعداد ترانزیستورها افزایش می یابد و سرعت انتقال داده بین بسیاری از دستگاه های روی برد نقش مهمی را ایفا می کنند.

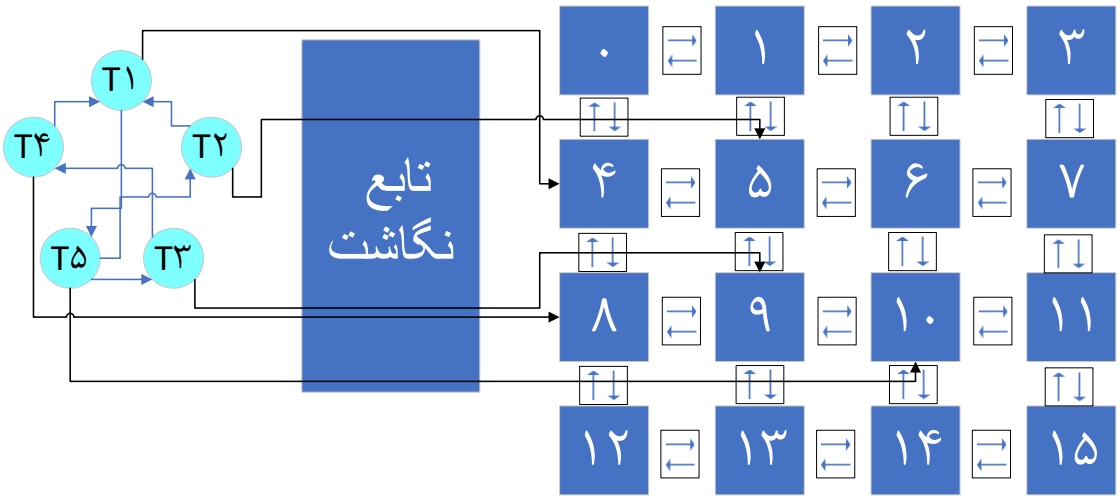
سیستم های روی تراشه برای برطرف کردن نیازهای عملکردی برنامه های مختلف طراحی شده اند که باید این نیازها برطرف شوند. شبکه روی تراشه به عنوان زیرساخت ارتباطی مقیاس پذیر با عملکرد و مصرف توان مناسب برای تراشه های چند هسته ای پیشنهادگردیده است؛ولی با بالا رفتن تعداد هسته های پردازش در یک تراشه، تأخیرشبکه و مصرف توان، به یک چالش

بزرگ در طراحی تبدیل گردیده است و مقیاس پذیری شبکه را متوقف مینماید. حجم ارتباط، به دلیل اینکه برای ارسال تمامی بسته ها از طریق منابع شبکه روی تراشه به زمان نیاز است، بر مصرف انرژی اثر میگذارد.حجم ارتباط بر زمان اجرای برنامه، به ویژه برای برنامه های محدود به ورودی/خروجی هم اثر دارد. به عنوان یک چالش مهم در طراحی سیستم مبتنی برشبکه روی تراشه میتوان از معین نمودن وابستگی مسیریاب های ساختمان با هسته های یک برنامه نام برده شود. نگاشت، جایگذاری گره های گراف وظایف بر روی واحدهای پردازشی شبکه روی تراشه است.

شبکه روی تراشه‌یک معماری ارتباطی کارآمد از جهت انرژی و توان برای سیستم‌های روی تراشه‌ی دارای چند کاشی می‌باشد. شبکه روی تراشه معایب سیستم بر روی تراشه را برطرف کرده است و قابلیت استفاده مجدد و کارایی بالایی دارد.

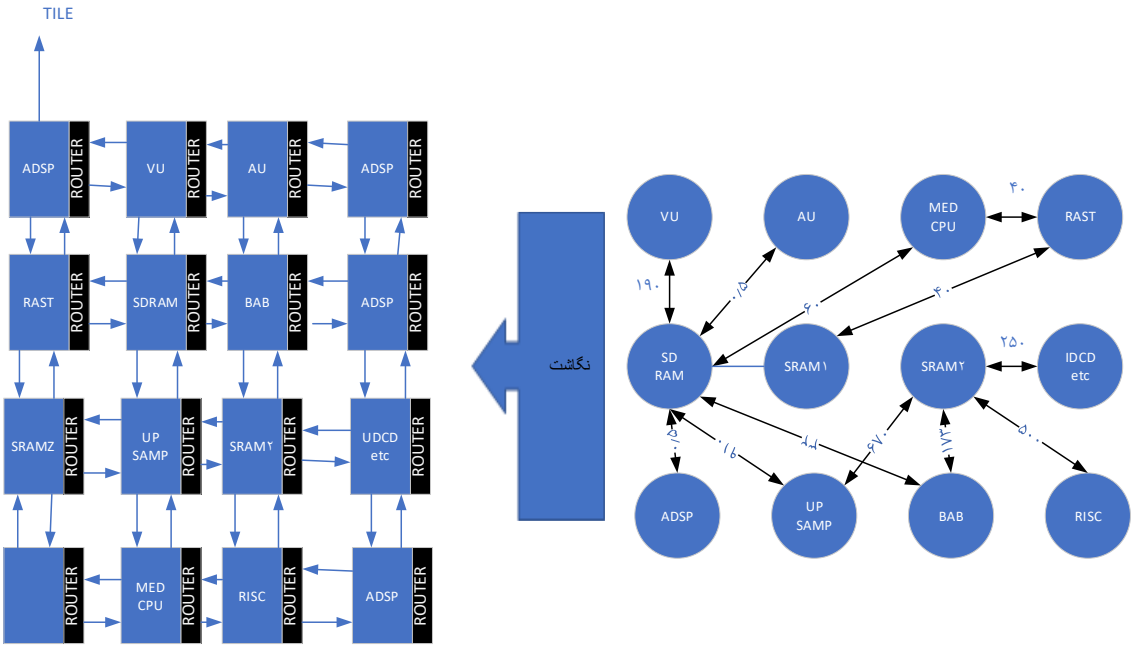
۲-۲- نگاشت مبتنی بر بهینه‌سازی ازدحام ذرات (PSO)

اخیراً الگوریتم‌های بسیاری در زمینه نگاشت هسته‌های شبکه روی تراشه ارائه شده است که سعی در کمینه کردن هزینه ارتباطی داشته‌اند. استفاده از تعداد گام‌های کمتر بین هسته‌های یکه با یکدیگر ارتباط دارند به طور قابل ملاحظه‌ای هزینه ارتباطی، انرژی مصرفی و تأخیر را کاهش می‌دهد. در واقع کاهش تعداد گام‌ها مشخصه تحمل پذیری خطا را کاهش می‌دهد چرا که هر چه تعداد گام‌ها کمتر باشد تعداد مسیرهای بین دو هسته که با یکدیگر ارتباط دارند کمتر شده و خرابی یک لینک باعث از کار افتادن این ارتباط می‌شود. در صورتی که اگر تعداد گام‌ها بیشتر باشد. مسیرها بین هسته‌های مرتبط بیشتر شده و از کار افتادن یک لینک از یک مسیر منجر به از کار افتادن این ارتباط نمی‌شود چرا که می‌توان از مسیر ارتباطی دیگراستفاده کرد در اینجا یک تناقض وجود دارد زیرا سعی در افزایش تحمل‌پذیری خطا، هزینه ارتباطی زیاد را که مطلوب نیست به همراه دارد و بر عکس. به همین دلیل راهکار ارایه شده، سعی در یافتن مناسبترین طراحی با در نظر گرفتن این دو تابع هدف متناقض دارد. همچنین به منظور این حذف بن بست وهمبندی توری، در این پایان نامه از مسیریابی XY استفاده شده است. منظور از همبندی چگونگی اتصال عناصر پردازشی به یکدیگر می باشد اگر یک همبندی را به گراف نگاشت کنیم، گره‌های آن مجموعه ای از هسته‌هایIP و مسیریاب ها و یال های آن لینکه ای ارتباطی این مجموعه ها به هم است. درشکل۳-۱زیرساخت ارتباطی نمایش داده شده است.



شکل (۱): زیرساخت ارتباطی در مساله نگاشت

بعنوان نمونه در شکل ۲ نحوه نگاشت گراف هسته یک برنامه کاربردی بر روی گراف معماری شبکه بروی تراشه به کمک تابع نگاشت نشان داده شده است.



شکل(۲): نگاشت گراف هسته بر روی شبکه روی تراشه

برای حل مساله نگاشت وظایف، در این مقاله از یک الگوریتم PSO اصلاح شده استفاده است. PSO یک الگوریتم هوشمند بر اساس ازدحام است[۹] که از رفتار اجتماعی حیوانات؛ نشات گرفته شده است. یک ذره در PSO مشابه یک پرنده یا ماهی در فضای مسئله است. حرکت هر ذره بوسیله یک سرعت که دارای اندازه (بُعد) و سرعت است،انجام می‌گیرد. موقعیت ذره در هر نمونه از زمان بوسیله بهترین موقعیت و موقعیت بهترین ذره در فضای مسئله تاثیرپذیر است. کارایی هر ذره توسط مقدار شایستگی که مختص مسئله است اندازه گیری می شود. الگوریتم PSO مشابه دیگر الگوریتم های تکاملی است. در PSO جمعیت برابر با تعداد ذره‌ها در فضای مسئله است. ذرات بصورت تصادفی مقداردهی اولیه می شوند. هر ذره یک مقدار سازگاری خواهد داشت و بوسیله تابع سازگاری که بایستی در هر نسل بهینه شود، ارزیابی خواهد شد. به عبارت دیگر این راهکار معادل یک پرنده در الگوی حرکت جمعی پرندگان می‌باشد. هر ذره یک مقدار شایستگی دارد که توسط یک تابع شایستگی محاسبه می‌شود. هر چه ذره در فضای جستجو به هدف (غذا در مدل حرکت پرندگان) نزدیک تر باشد،شایستگی بیشتری دارد. همچنین هر ذره دارای یک سرعت است که هدایت حرکت ذره را برعهده دارد. هر ذره با دنبال کردن ذرات بهینه در حالت فعلی،به حرکت خود در فضای مساله ادامه می‌دهد. به این شکل است که گروهی از ذرات PSO آغاز کار به صورت تصادفی به وجود می‌آیند و با به روز کردن نسل‌ها سعی دریافتن راه حل بهینه می‌نمایند.

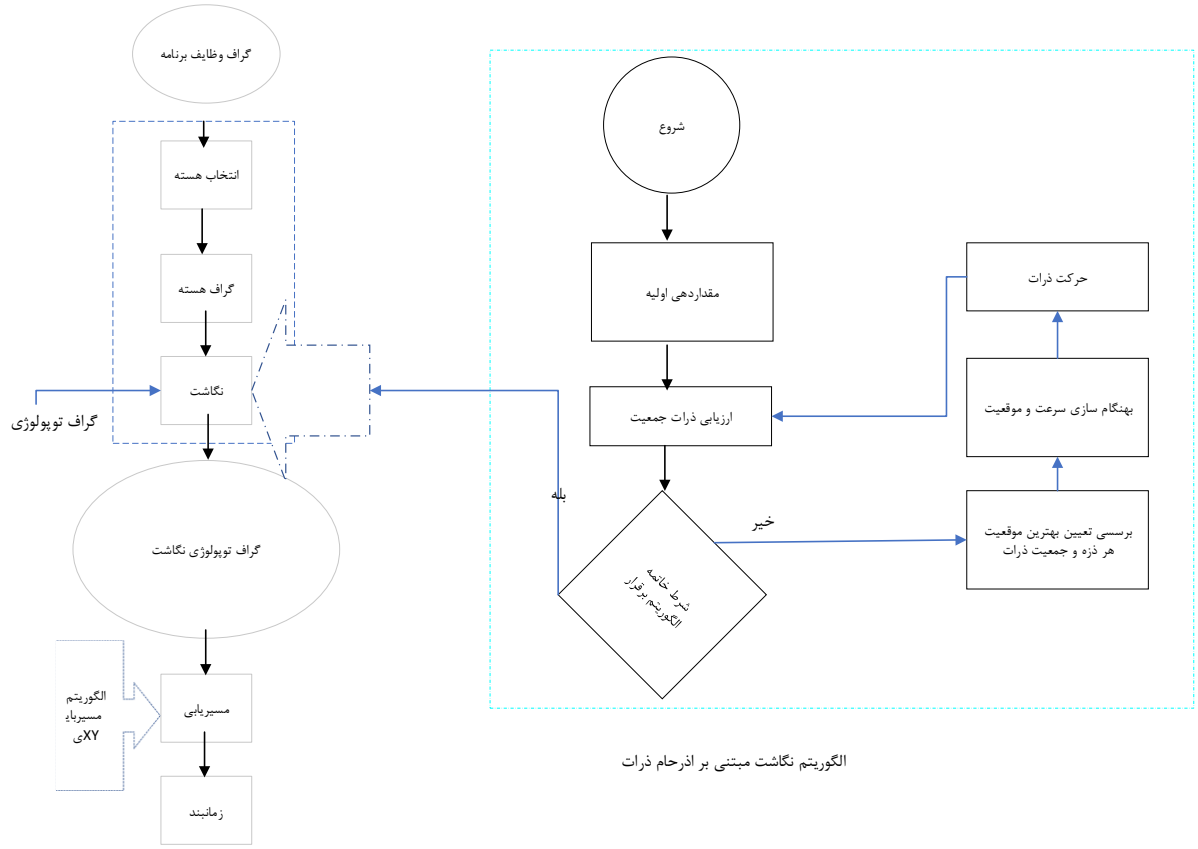
در واقع هر ذره بهترین مکان (pbest) و بهترین موقعیت تا بحال خود را از بین گروه کاملی از ذرات gbestپیدا می کند. pbest هر ذره بهترین نتیجه تا بحال بدست آمده بوسیله ذره است، در حالیکه gbest بهترین ذره به نسبت معیار سازگاری در کل جمعیت می باشد. در هر نسل سرعت و موقعیت ذرات به ترتیب توسط روابط ۱ و ۲ بروزرسانی خواهد شد.

$$v[\,] = v[\,] + c1 * rand() * (pbest[\,] - position[\,]) + c2 * rand() * (gbest[\,] - position[\,]) \qquad (۱)$$

$$position[\,] = position[\,] + v[\,] \qquad (۲)$$

سمت راست معادله(۱)از سه قسمت تشکیل شده که قسمت اول، سرعت فعلی ذره است و قسمت های دوم و سوم تغییر سرعت ذره و چرخش آن به سمت بهترین تجربه شخصی و بهترین تجربه گروه را به عهده دارند. اگر قسمت اول این معادله در نظر گرفته نشود ، آنگاه سرعت ذرات تنها با توجه به موقعیت فعلی و بهترین تجربه ذره و بهترین تجربه جمع تعیین می‌شود.

به این ترتیب، بهترین ذره جمع، در جای خود ثابت می‌ماند و سایرین به سمت آن ذره حرکت می‌کنند. در واقع حرکت دسته جمعی ذرات بدون قسمت اول معادله (۱)، پروسهای خواهد بود که طی آن فضای جستجو به تدریج کوچک می‌شود و جستجویی محلی حول بهترین ذره شکل می‌گیرد. در مقابل اگر فقط قسمت اول معادله (۱) درنظرگرفته شود، ذرات راه عادی خود را می‌روند تا به دیواره محدوده برسند و به نوعی جستجویی سراسری را انجام می‌دهند.



الگوریتم نگاشت مبتنی بر اذرحام ذرات

شکل (۳): فلوچارت راهکار پیشنهادی

۲-۳- الگوریتم نگاشت

راهکار به اینصورت عمل می‌کند که در ابتدا و به کمک الگوریتم ازدحام ذرات، وظایف آماده به پردازنده ارسالمی‌شود. سپس با توجه به تعداد وظایف تکمیل شده، لیست آماده بروزرسانی می‌گردد. سپس نگاشت‌های PSO دوباره محاسبه می‌شوند. در واقع این محاسبات، تعادل پویای نگاشت‌های وظایف را برقرار می‌کند. در ادامه و بر اساس محاسبه مجدد نگاشت PSO، وظایف به پردازنده واگذار می‌شود. این قدمها تازمانیکه همه وظایف در جریان کاری نگاشت داده شوند، ادامه می‌یابد. اولین مرحله در اجرای الگوریتم PSO، مقداردهی اولیه ازدحام و پارامترهای کنترلی می‌باشد. در این راهکار موقعیت ذرات به گونه‌ای تعیین می‌شوند که بصورت یکنواخت فضای جستجو را پوشش دهند. لازم به ذکر است که کارائی PSO تحت تاثیر تنوع اولیه در ازدحام می‌باشد. یعنی چه میزان از فضای جستجو پوشش داده شده و چگونه ذرات در فضای جستجو توزیع شده‌اند[۱۰]. چنانچه نقاط بهینه در ناحیه‌هائی از فضای جستجو که توسط ازدحام اولیه پوشش داده نمی‌شوند، قرار داشته باشند، PSO در یافتن آنها با مشکل مواجه می‌شود. بر همین اساس مقداردهی اولیه مناسب برای موقعیت هر ذره بصورت زیر می‌باشد:

$$x_{i,j}(\cdot) = x_{min,j} + r_j\left(x_{max,j} - x_{min,j}\right) \forall\, j = 1, \dots n_x, \quad \forall\, i = 1. \dots n_s \qquad (۳)$$

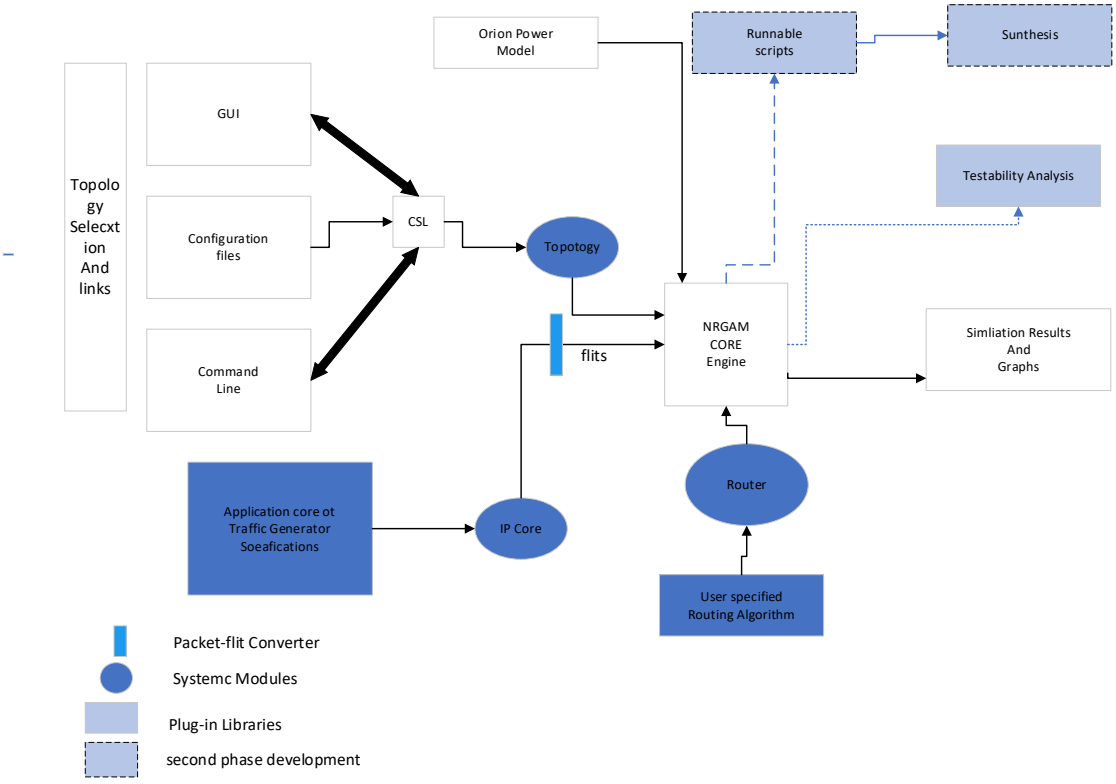
که در رابطه فوق x_{max} و x_{min} به ترتیب کمترین و بیشترین مقدار برای هر بعد مساله می‌باشند و همچنین r_j مقادیر تصادفی در دامنه 0 و 1 می‌باشد. بردارهای سرعت اولیه نیز در ابتدا بصورت تصادفی مقداردهی می‌شوند، مقداردهی تصادفی منجر به تولید جهت‌های حرکت و موقعیت‌های تصادفی می‌شود. البته می‌توان سرعت‌ها را مقداردهی اولیه نیز نمود، اما مقادیر آنها نباید زیاد

بزرگ فرض شود زیرا سرعت‌های اولیه بزرگ، اندازه حرکت بزرگتری برای ذرات به همراه خواهند داشت و در نتیجه، بهنگام-سازی بزرگی را در موقعیت اولیه ذره ایجاد می‌کنند، چنین بهنگام‌سازی‌های بزرگی ممکن است که منجر به خارج شدن ذره از محدوده فضای جستجو شود و در نتیجه زمان بیشتری برای همگرا شدن ذرات به یک راه حل مورد نیاز باشد. در ابتدا در شکل ۳ فلوچارت راهکار پیشنهادی نشان داده شده است [۱۱].

قدم‌های الگوریتم PSOتغییر یافته، در شکل ۳ لیست شده است. این الگوریتم با مقداردهی اولیه تصادفی مکان و سرعت ذره آغاز به کار می‌کند. در این مسئله ذرات وظایفی هستند که واگذار می شوند و اندازه (بُعد) ذرات، تعداد وظایف در نگاشت جریان کاری هستند. مقدار تخصیص داده شده به هر بُعد از ذرات شاخص‌های منابع پردازشی هستند. به این ترتیب هر ذره، نگاشت یک منبع به یک وظیفه را نشان می‌دهد. سرعت ذرات را با استفاده از رابطه (۱) محاسبه می‌کنند و بر اساس رابطه (۲) موقعیت مکانی‌شان را به‌روزرسانی می‌کنند. این ارزیابی تا زمانی‌که به تعداد مشخصی از تکرار رسد، ادامه پیدا می‌یابد[۱۲].

۳- روش کار و شبیه سازی

در این بخش یک بستر برای شبیه‌سازی ساختار و چهارچوب شبکه‌های روی تراشه ارائه می‌گردد. با ارائه این محیط می‌توان آزمایشات و ارزیابی این شبکه‌ها را انجام داد و جزئیات نتایج بدست آمده از شبکه روی تراشه را تحلیل نمود. چهارچوب این سیستم، بر اساس نرم‌افزار شبیه‌سازی متن‌باز Nirgam، طراحی شده است. Nirgam بر بستر زبان System C توسعه داده شده است. کامپونت‌های اصلی در این شبیه‌ساز مسیریاب‌ها، المان‌های پردازشی، لینک‌ها و بافرها می‌باشند. المان‌های پردازشی در واقع ایجادکننده‌های بار کاری برای شبکه‌های روی تراشه هستند و مسیریاب‌ها نیز جزء اصلی کامپونت‌ها می‌باشند که ارتباطات این شبکه‌ها را برقرار می‌کنند. هنگامیکه یک بسته به آنها می‌رسد، فیلدهای مربوطه را از سرایند فلیت آن‌ها خوانده و بر اساس آن گام بعدی را انتخاب نموده و سپس در دسترس‌پذیری لینک‌ها را بررسی، لینک‌ها را رزرو و نهایتاً بسته را انتقال می‌دهد. با توجه به آنکه توپولوژی مش یکی از اصلی‌ترین توپولوزی‌های شبکه روی تراشه می‌باشد در این شبیه‌ساز نیز بصورت پیش‌فرض از این نوع توپولوژی پشتیبانی شده است. ساختار این برنامه در شکل ۴ نشان داده شده است [۱۳].



شکل(۴): یک ساختار Nirgam [۱۳]

Nirgam به راحتی مکانیزم‌هایی را جهت آزمایش الگوریتم‌های نگاشت و برنامه‌های کاربردی مربوط به شبکه‌های روی تراشه، فراهم می‌آورد. کاربر براحتی می‌تواند برنامه‌ها و نگاشت‌های مورد نظر را به برنامه وارد کند. شبیه‌ساز قادر است بصورت پویا مسیریاب‌ها و تمامی کتابخانه‌ها و کدهای مربوط به کاربر، را بارگزاری نماید. تعدادی از قابلیت‌های کلیدی این شبیه‌ساز به شرح زیر است:

فرستادن درخواستCBR از یک هسته به هسته دیگری، بصورت تصادفی یا انتخابی

فرستادن پیک درخواست از یک هسته به هسته دیگری، بصورت تصادفی یا انتخابی

فرستادن پیک درخواست و CBR بعد از رزرو کردن باند مورد نیاز برای ترافیک

فرستادن پیک درخواست و CBR در توپولوژی مش در صورت بوجود آمدن خطای لینک

محاسبه میزان انرژی مسیریاب برای هر تعداد سیکل بخصوص

هر یک از کاربردهای فوق می‌تواند بر اساس نوع الگوریتم انتخاب شده شبیه‌سازی شود. در جدول ۱ خلاصه‌ای از قابلیت‌های اصلی این شبیه‌ساز نشان داده شده است.

جدول (۱): یک قابلیت‌های اصلی شبیه‌ساز Nirgam

نوع توپولوژی	مکانیزم سوئیچینگ	نوع الگوریتم‌های مسیریابی	انواع بارکاری تولیدی
Mesh و Torus	**Wormhole**	**Odd-Even -XY**	**Constant Bit Rate Bursty,Trace-based**

همچنین در [۱۴] مقایسه‌ای بین انواع شبیه‌سازهای ارایه شده برای NoC شده است که در جدول ۴-۲ نشان داده شده است.

جدول (۲): مقایسه‌ای بین انواع شبیه‌سازهای NoC [۱۴]

	Fram-work	Availability	Parallel-ism	Topologies	Open-source
SICOSYS	C++	+	-	Limited	+
Noxim	System C	+	-	Mesh	+
NNSE	System C	+	-	Mesh/Torus	+
Nirgam	System C	+	-	All	+
gpNoCsim	Java	+	-	All	+
DARSIM	++	-	+	All	+

۳-۱- پارامترهای پیکربندی شبیه‌ساز

به منظور پیکربندی شبیه ساز باید سه فایل زیر را تغییر داد:

Nigram.config: که داخل آن پارامترهای مربوط به طراحی قرار می‌گیرند.

Application.config: جهت نگاشت برنامه‌های کاربردی به کاشی‌ها کاربرد دارد

Traffic : پارامترهای مربوط به پیکربندی بارکاری جهت تولید درخواست‌های ترافیکی متفاوت در این قسمت انجام می‌گیرد.

در جدول (۳) نمونه‌ای از تنظیمات مربوط به فایل Nirgrm.config برای توپولوژی مش نشان داده شده است.

ساختار فوق برای شبیه سازی یک توپولوژی مش با اندازه ۴ سطر و ۵ ستون بکار می رود و الگوریتم X-Y برای مسیریابی استفاده شده می‌شود. شروع ایجاد بار کاری بعد از ۵ سیکل ساعت انجام می‌گیرد و تا ۱۰۰ سیکل ساعت ادامه می‌یابد و نهایتا شبیه سازی پس از ۲۰۰۰ سیکل ساعت به اتمام می‌رسد. جدول (۴) نیز نمونه تنظیمات مربوط به ایجاد بارکاری فرضی را نشان می‌دهد. این تنظیمات برای کاشی شماره ۱ و۷ می‌باشد که در آن عدد مورد نظر شناسه آن کاشی است. کاشی n برای

یک مقصد تصادفی یک بسته ۱۰ بایتی ایجاد می‌کند. فاصله بین هر فلیت ۲ سیکل ساعت فرض شده و بارکاری نیز ۹۰٪ در نظر گرفته شده است.

جدول (۳): نمونه نظیمات فایل Nigram.config

TOPOLOGY	MESH
NUMLROWS	4
NUMLCOLS	5
RT_ALGO	XY
WARMUP	5
SIM_NUM	2000
TG_NUM	100

جدول (۴) : نمونه تنظیمات ایجاد بارکاری

TILE	1,7
PCKT_SIZE	10
FLIT_INTERVAL	2
LOAD	90
DESTINATION	RANDOM

۳-۲- معیارهای و پارامترهای ارزیابی

میانگین تأخیر، حداکثر زمان تأخیر و مصرف انرژی، به عنوان معیارهای کارایی درنظر گرفته شده‌اند. تأخیر، مدت زمان بین وارد شدن سرایند فلیت به شبکه، تا زمان رسیدن ادامه فلیت به گره مقصد فرض شده است. فرمول میانگین تأخیر (D) که توسط شبیه‌ساز محاسبه می‌شود، بصورت زیر است:

$$D= 1/n \sum_{i=1}^{n} D_i \quad (۴)$$

که در آن، n مجموع بسته هایی است که به مقصد رسیده اند و D_i تأخیر بسته i می‌باشد. حداکثر تأخیر (m) نیز بصورت زیر تعریف می‌شود:

$$M= \max(D_i): i= 1, 2,\ldots,n \quad (۵)$$

توان به مقدار انرژی گفته می شود که توسط مسیریاب و لینک‌ها، هدر رفته است.

آزمایشات برای یک شبکه Mesh با اندازه ۵*۵ انجام گرفته و گره ها ۶ بسته فلیت با توزیع نمایی و نرخ دوره‌ای بین بازه pir <10=> ایجاد می‌کنند که در آن pir* سرعت تزریق بسته است. بعنوان مثال مقدار pir=0.1 به معنای آن است که هر گره بسته‌ها را در ۱۰ سیکل ساعت می‌فرستد. بافرهای FIFO با ظرفیت ۴ ردیفی در نظر گرفته شده‌اند. هر مرحله، شبیه سازی‌ها ابتدا ۳۰۰۰ سیکل ساعت بعنوان زمان آماده‌سازی و نهایتا ۵۵۰۰ سیکل ساعت اجرا شده‌اند. بارگذاری جهت سنجش راهکارها، نیاز به یک نوع بارکاری فرضی می‌باشد، تا راهکارها در بارکاری متفاوت مورد سنجش و ارزیابی قرار گیرند. به این بار کاری فرضی بارگذاری گفته می‌شود.

* packet injection rate

۳-۲-۱- بارگذاری

جهت سنجش راهکارها باید حالت اجرای واقعی مورد شبیه‌سازی قرار گیرد در نتیجه نیاز به ایجاد ترافیک و بارهای فرضی است. به این بار کاری فرضی، بارگذاری گفته می‌شود. بر همین اساس جهت ایجاد بار کاری فرضی از استراتژی Bursty استفاده می‌شود. در این حالت توسط این الگوریتم در بازه‌های زمانی بخصوص ترافیک کم و زیاد می‌شود. این روش قابلیت پشتیبانی از بارگذاری‌های متفاوتی را دارا می‌باشد، در نتیجه با بهره‌گیری از آن می‌توان راهکارهای را در حالات بارگذاری مختلف مقایسه نمود. بر همین اساس، در ارزیابی‌های انجام شده بارهای کاری ۴۰، ۷۰ و ۱۰۰ درصد ایجاد شده است که بارکاری ۴۰ درصد برای شبیه‌سازی وضعیتی که ترافیک شبکه کم، بار کاری ۷۰ درصد برای وضعیتی که میزان بار متوسط و در نهایت بار کاری ۱۰۰ درصد، جهت ارزیابی راهکارها در حالت حداکثر بارگذاری مورد استفاده قرار گرفته است.

۳-۳- الگوریتم‌های مورد ارزیابی

راهکار نگاشت تصادفی: راهکار نگاشت پیش‌فرض در شبیه‌ساز Nirgam می باشد. در این حالت، بطور تصادفی گره‌های موجود در گراف وظایف انتخاب و در هسته‌های خالی شبکه روی تراشه که حداقل شرایط مورد نیاز جهت پردازش را داشته باشد، نگاشت می‌شود.

راهکار نگاشت CA[†]: در این روش از طریق پارتیشن‌بندی نواحی مختلف شبکه‌های روی تراشه و محاسبه میزان ترافیک و ازدحام در هر بخش، عملیات نگاشت بهینه را با توجه به وضعیت موود انجام می‌دهد. الگوریتم CAبه‌منظور حداکثر نمودن دقت در محاسبه ترافیم (ازدحام هر پارتیشین) تعیین شده است که این کار از طریق قرار دادن مقادیر جداگانه برای ترافیک در هر چهار ربع شبکه انجام می‌شود [۳۷].

راهکار نگاشت پیشنهادی(Proposed): در این روش عملیات نگاشت مبتنی بر راهکار تشریح شده در فصل سوم و با بهره-گیری از الگوریتم بهینه‌سازی ازدحام ذرات انجام می‌گیرد.

۳-۴- نحوه اجرای شبیه‌ساز

جهت اجرای شبیه‌ساز ابتدا لازم است که فایل کاشی‌های مورد نظر (tiles) ایجاد و تعریف شوند، در این راهکار ۴ کاشی تعریف می‌شود که به هریک از کاشی‌ها در سناریوهای مختلف و بصورت تصادفی بار کاری ارسال می‌شود.پارامترهای کلی که در کاشی‌ها تعریف می‌شود، بصورت جدول ۵ می‌باشد.

جدول (۵): نحوه تنظیم کاشی‌ها

توضیح	مقدار	پارامتر
اندازه بسته	16	PKT_SIZE
میزان بارگذاری	40, 70, 100	LOAD
میانگین سایز بسته‌های تولید شده توسط بارکاری Bursty	3	AVG_BURST_LEN
میانگین زمان توقف بین تولید دو بارکاری	4	AVG_OFFTIME
مقصد	RANDOM	DESTINATION
فاصله بین دو ارسال در سیکل ساعت	2	FLIT_INTERVAL

[†] Congestion Awareness

پس از تعریف کشی‌ها، حال در فایل Nirgam.config تنظیمات مربوط به پارامترهای شبیه‌سازی انجام می‌گیرد، مقادیر این پارامترها در جدول ۶ آورده شده است.

جدول (۶): ارامترهای شبیه‌سازی

نوع توپولوژی	مش
تعداد سطر و ستون توپولوژی	۸*۸
الگوریتم نگاشت	پیشنهادی
تعداد سیکل ساعت قبل از ایجاد درخواست	۵
تعداد سیکل‌های شبیه‌سازی	۵۰۰۰
تعداد سیکل ساعت پس از ایجاد درخواست بعدی	۳۰۰۰
تعداد بافر	۶۴

۳-۵- نتایج آزمایشات

برای سناریوی اعمالی؛ میانگین تأخیر در کانال، تأخیر کل و توان مصرفی مورد ارزیابی قرار گرفته‌اند. سناریوی بارکاری مورد استفاده Bursty می‌باشد که بصورت تصادفی و با نرخ‌های مختلف، بارکاری مورد نظر را اعمال می‌کنند.

میانگین تأخیر بسته: مدت زمان بین وارد شدن سرآیند فلیت[‡] به شبکه تا زمان رسیدن ادامه فلیت[§] به گره مقصد فرض شده است. فرمول میانگین تأخیر (D) که توسط شبیه‌ساز محاسبه می‌شود، به‌صورت زیر است:

$$D= 1/n \sum_{i=1}^{n} D_i \quad (۶)$$

که در آن، n مجموع بسته‌هایی است که به مقصد رسیده‌اند و Di تأخیر بسته i می‌باشد. حداکثر تأخیر (M)نیز به‌صورت زیر تعریف می‌شود:

$$M= \max(D_i): i= 1, 2,\ldots,n \quad (۷)$$

توان مصرفی (انرژی مصرفی): توان به مقدار انرژی گفته می‌شود که توسط مسیریاب و لینک‌ها، هدر رفته است. انرژی مصرفی بر حسب ژول محاسبه شده است. شکل‌های (۷) تا (۱۵) نتایجی را نشان می‌دهند. همانگونه که در نمودارها نشان داده شده برای هر استراتژی نگاشت، میانگین تأخیر و میزان مصرف انرژی در بارهای متفاوت، اندازه گیری شده‌اند.

با توجه با نتایج ارزیابی، روش پیشنهادی در مقایسه با دیگر روش‌ها به خاطر دو ویژگی مهم، عملکرد بهتری دارد؛

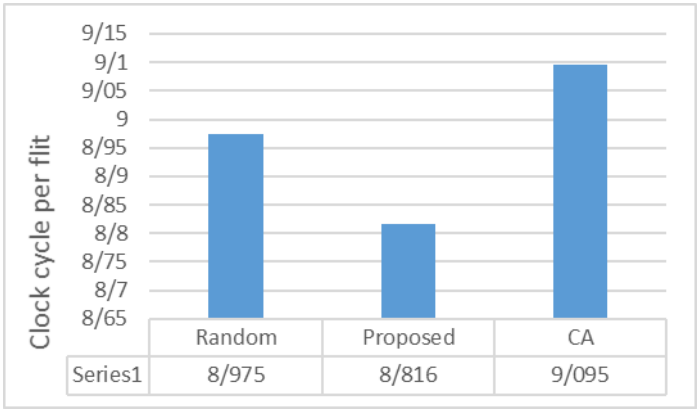
- به کمک نگاشت مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات، علاوه بر کاهش میزان تاخیر، مصرف انرژی نیز بهینه شده است.
- با محاسبات تعیین شده، وضعیت بار مسیر ارتباطی در نظر گرفته می‌شود.

‡ - Header flit
§ - Tail flit

نتایج بررسی مصرف انرژی، که در شکل های۷ و ۱۲ قابل مشاهده است، نشان می‌دهد که متوسط ژول/سیکل مصرف شده برای سناریوی ارایه شده در سرعت های مختلف تزریق، بهبودی نسبی در مقایسه با دیگر استراتژی های نگاشت داشته است. نتیجه مهم دیگر اینکه بهبود مصرف انرژی استراتژی پیشنهادی هنگامی که سرعت تزریق افزایش می یابد، حتی به نرخ های بالاتری نیز می‌رسد. این نکته نشان می دهد که تعادل طبیعی بارگذاری بهتر این روش در بارگذاری‌های بالاتر نیز به خوبی تحمل‌پذیری خطا و همچنین مصرف انرژی را کنترل می‌کند.

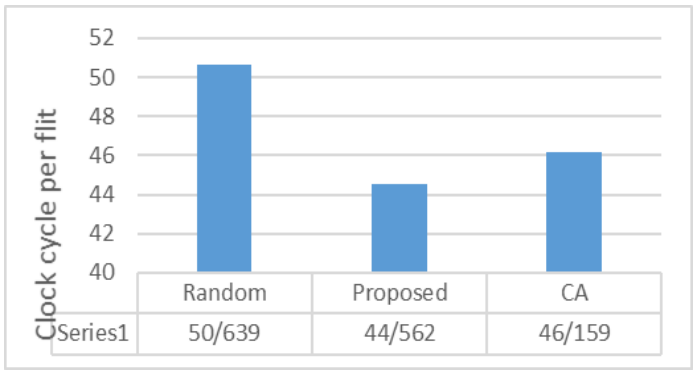
۳-۵-۱- آزمایش اول-حداقل بار (load=40%)

در این آزمایش راهکارها در حالتیکه بارکاری ۴۰ درصد می‌باشد مورد ارزیابی قرار گرفته‌اند. از طریق این آزمایش می‌توان میزان عملکرد را در حالت کم بار مورد بررسی قرار داد. نتایج آزمایش در شکل‌های ۴-۵ تا ۴-۷ نمایش داده شده است. در شکل ۴-۵ میانگین تاخیر کلی در هر کانال نشان داده شده است. همانطورکه مشاهده می‌شود. راهکار پیشنهادی در این حالت از هر نظر بهینه‌تر از دو راهکار انتخابی دیگر می باشد.



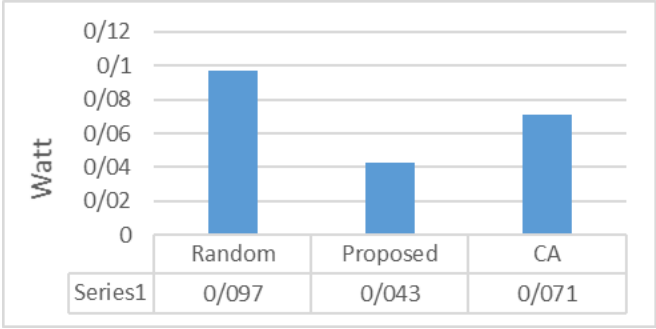
شکل(۷): میانگین تاخیر کلی در هر کانال – (load=40%)

همانطورکه ملاحظه می‌شود راهکار پیشنهادی با توجه به آنکه نگاشت وظایف از طریق الگوریتم بهینه‌سازی ازدحام ذرات انجام می‌گیرد، در نتیجه بالاترین کارایی را از نظر میزان تاخیر کلی در هر کانال از خود نشان داده است. از طرفی راهکار نگاشت Random (تصادفی) با توجه به آنکه بصورت تصادفی عملیات نگاشت را انجام داده و در نتیجه هیچ‌گونه بررسی در تراشه‌ها وجود ندارد و همچنین مکانیزم خاصی برای بررسی وضعیت هسته‌ها ارایه نمی‌دهد در نتیجه کارایی چندانی را از نظر میزان تاخیر نداشته است. در ادامه در شکل ۸: میزان تاخیر کلی، نشان داده شده است.



شکل (۸): میانگین تاخیر کلی– (load= 40%)

همانطورکه مشاهده می‌شود در تاخیر کلی، میزان کارائی CA نیز بالا رفته است اما با توجه به آنکه در راهکار پیشنهادی، نگاشت وظایف به کمک ناحیه‌بندی هسته‌ها و در نتیجه وضعیت هر هسته انجام می‌گیرد پس طبیعی است که بکارگیری الگوریتم PSO باعث افزایش کارایی کلی شده است و این امر توانسته زمینه‌سازی توازن‌بار در شبکه و همچنین کاهش مصرف انرژی را در پی داشته باشد که در شکل ۹ نیز قابل مشاهده است.

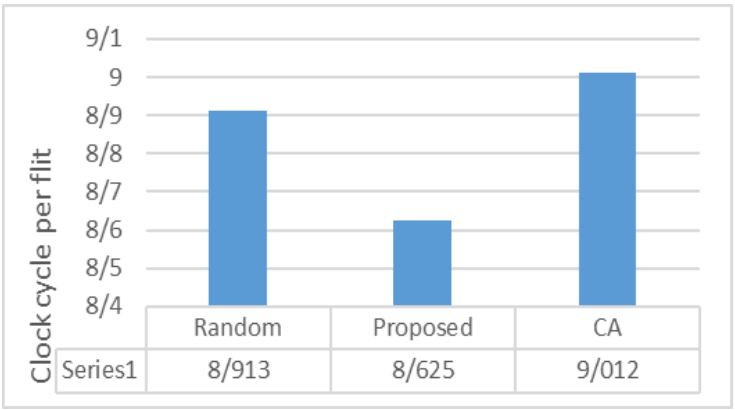


شکل (۹): میزان انرژی مصرفی– (load= 40%)

همانطورکه در شکل ۹ مشاهده می‌شود، راهکار ارایه شده، بخوبی توانسته است، میزان مصرف انرژی را کاهش دهد، بطوریکه نسبت به راهکار Random، این میزان به بیش از ۵۰ درصد می‌رسد، در واقع یکی از دلایل اصلی این میزان کاهش مصرف در بکارگیری نگاشت مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات (PSO) می‌باشد و همچنین در کنار آن هسته‌هایی انتخاب شده‌اند که دارای بیشترین نیاز ارتباطی، هستند. بر همین اساس یک نوع توازن بار بهینه بوجود می ایـد و همین توازن‌بار سبب کاهش میزان مصرف انرژی شده است، زیرا در روش‌های دیگر مانند Random و CA هیچگونه ناحیه‌بندی انجام نمی‌شود و همچنین توجه چندانی به نیازهای ارتباطی نمی‌شود و در نتیجه در ادامه سبب ازدیاد ترافیک و ایجاد گلوگاه می‌شود که در نتیجه آن میزان مصرف انرژی و حتی حرارت افزایش پیدا می‌کند.

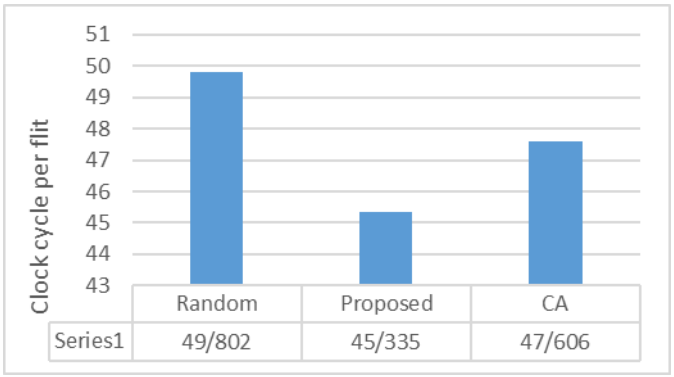
۳-۵-۲- آزمایش دوم-میزان بار نرمال (load=70%)

در این آزمایش راهکارها در حالتیکه بارکاری ۷۰ درصد می‌باشد مورد ارزیابی قرار گرفته‌اند. از طریق این آزمایش می‌توان میزان عملکرد را در حالتیکه میزان بار متوسط به بالا می‌باشد، مورد بررسی قرار داد. نتایج آزمایش در شکل‌های ۱۰ تا ۱۲ نمایش داده شده است.



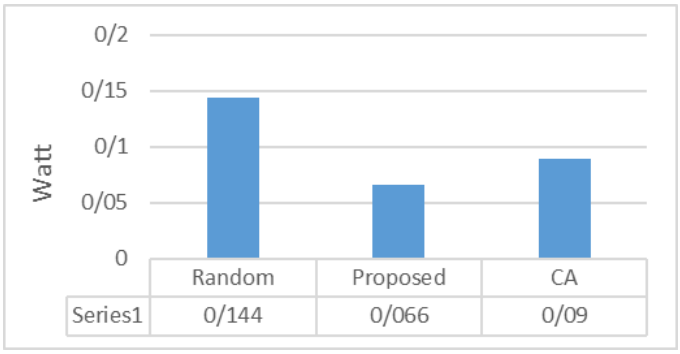
شکل (۱۰): میانگین تاخیر کلی در هر کانال – (load=70%)

همانطورکه در شکل ۱۰نشان داده شده است، میزان تاخیر هر چند افزایش محسوسی داشته است، اما راهکار پیشنهادی با بهره‌گیری از ناحیه‌بندی و همچنین بکارگیری نگاشت مبتنی بر PSO از ایجاد ازدحام و همچنین خطا در زیرساخت شبکه‌های روی تراشه جلوگیری به عمل آورده و در نتیجه کاهش میزان تاخیر کمتری به نسبت سایر راهکارها داشته است.



شکل (۱۱): میانگین تاخیر کلی– (load= 70%)

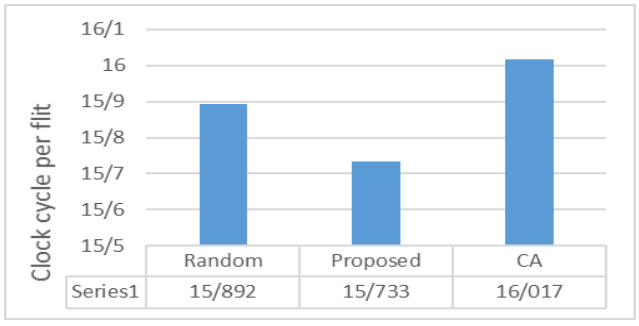
میانگین تاخیر کلی نیز در شکل ۱۱ نشان داده شده است و دوباره بهینه‌تر بودن راهکار قابل اثبات است. نهایتاً در شکل ۴-۱۰ میزان مصرف انرژی در حالت load=70 درصد نشان داده شده است.



شکل (۱۲): میزان انرژی مصرفی– (load= 70%)

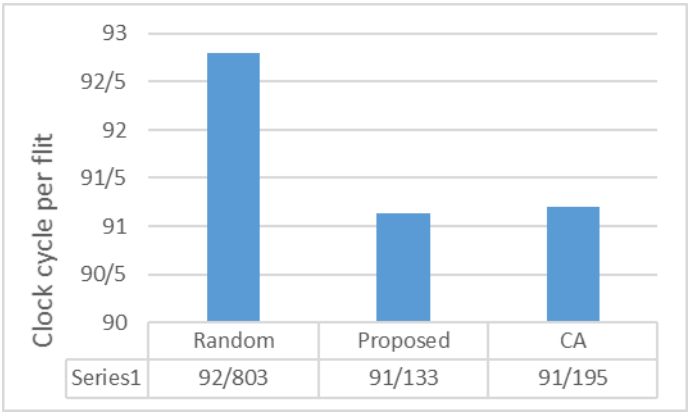
۳-۵-۳- آزمایش سوم-حداکثر بار (load=100%)

در این آزمایش راهکارها در حالتیکه بار کاری ۱۰۰ درصد می‌باشد، مورد ارزیابی قرار گرفته‌اند. از طریق این آزمایش می‌توان میزان عملکرد را در حالتیکه زیرساخت روی تراشه ۱۰۰ درصد درگیر می‌باشند، مورد بررسی قرار داد. نتایج آزمایش در شکل-های ۱۳ تا ۱۵ نمایش داده شده است.



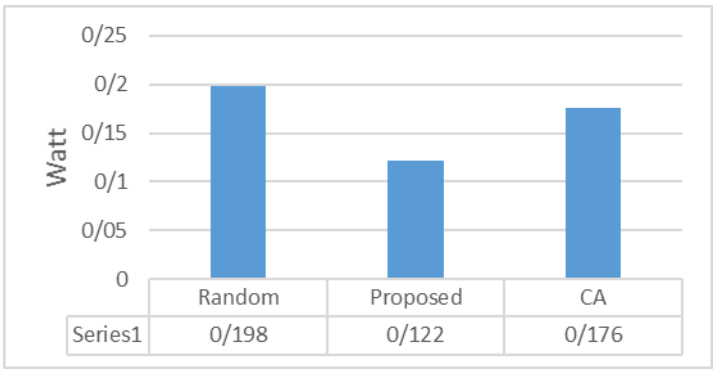
شکل (۱۳): میانگین تاخیر کلی در هر کانال – (load=100%)

با توجه به آنکه در این ارزیابی، میزان بارگذاری در حالت حداکثری می باشد پس طبیعی است که میزان تاخیر نیز به نسبت سایر حالات حداکثری است. اما همانطورکه در شکل۱۳ مشاهده می‌شود راهکار پیشنهادی در این حالت بهترین عملکرد را از خود نشان داده و در واقع در این میزان بار زیاد، قابلیت‌های آن نمایان شده، بطوریکه میزان تاخیر تقریباً نصف دو راهکار دیگر می‌باشد که این به دلیل بهره‌گیری از الگوریتم بهینه‌سازی ازدحام ذرات در فرایند نگاشت می‌باشد و در نتیجه افزایش قابلیت اطمینان و تحمل‌پذیری خطا می‌باشد.



شکل (۱۴): میانگین تاخیر کلی– (load= 100%)

همانطورکه در شکل ۱۴ مشاهده می‌شود، هر چند راهکار ارایه شده در مقایسه با راهکار CA در حدود ۸ درصد بهینگی و در مقایسه با Random، بیش از ۵۰ درصد بهینگی در میزان تاخیر کلی کاهش نشان داده است. دلیل کمتر بودن این میزان نسبت به راهکار CA افزایش بیش از حد بارکاری و در نتیجه داغ شدن بخش‌های مختلف زیرساخت NoC می‌تواند باشد.
نهایتاً در شکل ۱۵ میزان مصرف انرژی در حالت بارگذاری حداکثری نشان داده شده است، اگرچه بارکاری زیاد سبب افزایش مصرف انرژی شده است، اما در راهکار ارایه شده، با بهره‌گیری از بکارگیری نگاشت بهینه در کنار تکنیک ناحیه‌بندی هسته‌ها، یک نوع توازن بار ایجاد شده و در نتیجه میزان مصرف انرژی کمتر بوده است.



شکل (۱۵): میزان انرژی مصرفی– (load= 100%)

۴- نتیجه‌گیری

شبکه روی تراشه جایگزین سیستم‌های ارتباطی مبتنی بر گذرگاه معرفی شده‌اند و همچنین برای مقابله با پیچیدگی در سیستم‌های نهفته طراحی شده اند. تراشه‌های مبتنی بر گذرگاه در ابعاد بزرگتر قابلیت انجام با عملکرد بالا را ندارند بنابراین الگوی شبکه بر تراشه در عصر حاضر یا در آینده برای سیستم‌های چند پردازنده‌ای مطرح شده است و تمام محققان تلاش می‌کنند تا کارآیی، قابلیت اطمینان را در اینگونه سیستم‌ها بالا ببرند و مصرف توان و تاخیر را کم کنند. نگاشت وظایف یک

مساله‌ی بسیار مهم در طول طراحی یک شبکه بر تراشه است. با ادغام هرچه بیشتر مدارهای الکترونیکی مقیاس آنها افزایش یافته در نتیجه نیاز به راهکارهای بهینه به منظور نگاشت کارا، ضروری است.

نگاشت از مراحل بسیار مهم در طراحی شبکه بر تراشه می‌باشد که در وهله‌ی اول شامل تخصیص وظایف به هسته‌ها می باشد. یک راه نگاشت که استفاده می شود پیدا کردن بهترین جای گذاری هسته ها در نودهای توپولوژی شبکه بر تراشه می‌باشد. یک تکنیک نگاشت محدودیت های زیادی دارد مثل زمان ارتباطی، مصرف توان، تاخیر یا توان عملیاتی. راه حل نگاشت عمدتا ابتکاری هستند که می‌توانند در طبقه‌بندی‌های مختلف دسته‌بندی شوند. نگاشت به دسته‌ی هسته‌های یکسان و انواع مختلف از هسته‌ها طبقه بندی می‌شود . پدیده‌های مختلف از قبیل تکنولوژی مقیاس‌پذیری، تغییر پذیری پردازش‌ها، باعث بالا رفتن نیازها در قابلیت اطمینان در شبکه بر تراشه شده است که در برابر خطاهایی که در سیستم اتفاق می افتد بتواند سیستم را پایدار نگاه دارد.

بر همین اساس در مقاله یک راهکار نگاشت مبتنی بر بهینه‌سازی ازدحام ذرات ارایه گردید تا از طریق بکارگیری آن بتوان عملیات نگاشت را با کمترین تاخیر انجام داد. برای حل مساله نگاشت وظایف، در این پایان‌نامه از یک الگوریتم PSO اصلاح شده استفاده است. PSO یک الگوریتم هوشمند بر اساس ازدحام است که از رفتار اجتماعی حیوانات؛ نشات گرفته شده است. یک ذره در PSO مشابه یک پرنده یا ماهی در فضای مسئله است. حرکت هر ذره بوسیله یک سرعت که دارای اندازه (بُعد) و سرعت است،انجام می‌گیرد. موقعیت ذره در هر نمونه از زمان بوسیله بهترین موقعیت و موقعیت بهترین ذره در فضای مسئله تاثیرپذیر است. کارایی هر ذره توسط مقدار شایستگی که مختص مسئله است اندازه گیری می شود. الگوریتم PSO مشابه دیگر الگوریتم های تکاملی است. در PSO جمعیت برابر با تعداد ذره‌ها در فضای مسئله است. ذرات بصورت تصادفی مقداردهی اولیه می شوند. هر ذره یک مقدار سازگاری خواهد داشت و بوسیله تابع سازگاری که بایستی در هر نسل بهینه شود، ارزیابی خواهد شد. به عبارت دیگر این راهکار معادل یک پرنده در الگوی حرکت جمعی پرندگان می‌باشد. هر ذره یک مقدار شایستگی دارد که توسط یک تابع شایستگی محاسبه می‌شود. هر چه ذره در فضای جستجو به هدف (غذا در مدل حرکت پرندگان) نزدیکتر باشد، شایستگی بیشتری دارد. همچنین هر ذره دارای یک سرعت است که هدایت حرکت ذره را برعهده دارد .هر ذره با دنبال کردن ذرات بهینه در حالت فعلی، به حرکت خود در فضای مساله ادامه می‌دهد. به این شکل است که گروهی از ذرات PSO آغاز کار به صورت تصادفی به وجود می‌آیند و با به روز کردن نسل‌ها سعی دریافتن راه حل بهینه می‌نمایند.

در انتها، راهکار پیشنهادی به کمک نرم‌افزار منبع باز Nirgam جهت ارزیابی، شبیه‌سازی و مورد تست قرار گرفت. Nirgam به‌راحتی مکانیزم‌هایی را جهت آزمایش الگوریتم‌های نگاشت و برنامه‌های کاربردی مربوط به شبکه‌های روی تراشه، فراهم می‌آورد. کاربر به‌راحتی می‌تواند الگوریتم مورد نظر و نگاشت خود را به برنامه وارد کند. شبیه‌ساز قادر است به‌صورت پویا مسیریاب‌ها و تمامی کتابخانه‌ها و کدهای مربوط به کاربر را بارگزاری نماید در انتها در این محیط شبیه‌سازی نشان داده شد که این روش کارایی را به‌صورت چشم‌گیری افزایش می‌دهد. برای این منظور سه تست بر روی این الگوریتم‌ها انجام گرفت، و راهکار پیشنهادی با الگوریتم‌های Random و CA در حالات بارگذاری متفاوت، مقایسه و نشان داده شده است میزان تأخیر و مصرف انرژی به نسبت سایر روش‌ها به طور چشم‌گیری کاهش یافته است؛ بطوری که در هر سه آزمایش به نسبت راهکار مبتنی بر Random بیش از ۵۰ درصد کاهش در میزان تاخیر بوجود آمده است، در مقایسه با راهکار CA نیز این میزان کاهش هر چند به میزان Random کمتر بوده اما باز میزان تاخیر و کاهش مصرف انرژی چشمگیر بوده است.

References
مراجع

[1] E. Ofori-Attah, M. O. Agyeman, “A survey of power-aware Network-on-Chip design techniques,” in *The 13th Int. Multi-Conf. on Comput. in Global Inf. Technol.*, Apr. 2018.

[2] H. Zheng, A. Louri, “An energy-efficient network-on-chip design using reinforcement learning,” in *Proc. 56th Annu. Des. Autom. Conf.*, Jun. 2019, pp. 1-6.

[3] T. S. Arulananth et al., “Evaluation of low power consumption network on chip routing architecture,” *Microprocessors and Microsystems*, 2021.

[4] Y. Wu, C. Lu, Y. Chen, “A survey of routing algorithm for mesh Network-on-Chip,” *Front. Comput. Sci.*, vol. 10, no. 4, pp. 591-601, 2016.

[5] J. Liu, J. Harkin, Y. Li, L. Maguire, “Low cost fault-tolerant routing algorithm for networks-on-chip,” *Microprocessors and Microsystems*, vol. 39, no. 6, pp. 358-372, 2015.

[6] A. Jantsch and H. Tenhunen, Eds., *Networks on Chip*, vol. 396. Dordrecht: Kluwer Academic Publishers, 2013.

[7] N. L. Venkataraman, R. Kumar, P. M. Shakeel, “Ant lion optimized bufferless routing in the design of low power application specific network on chip,” *Circuits, Syst., Signal Process.*, vol. 39, no. 2, pp. 961-976, 2020.

[8] R. Kumar et al., “Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling,” *ACM SIGARCH Computer Architecture News*, vol. 33, no. 2, pp. 408-419, 2005.

[9] L. Shang et al., “Thermal Modeling, Characterization and Management of On-Chip Networks,” in *Proc. 37th Annu. IEEE/ACM Int. Symp.*, 2004.

[10] J. V. Escamilla, J. Flich, M. R. Casu, “ICARO-PAPM: Congestion Management with Selective Queue Power-Gating,” in *2017 Int. Conf. on High Perform. Comput. & Simul. (HPCS)*, Jul. 2017, pp. 259-266.

[11] M. Boyer, B. D. de Dinechin, A. Graillat, L. Havet, “Computing routes and delay bounds for the network-on-chip of the Kalray processor,” in *ERTS 2018-9th Eur. Congr. on Embedded Real Time Softw. and Syst.*, Jan. 2018.

[12] J. Hu, “Energy-aware mapping for tile-based noc architectures under performance constraints,” in *Proc. 2003 Asia and South Pacific Des. Autom. Conf.*, 2003, pp. 233-239.

[13] L. Benini, G. De Micheli, “Networks on chip: a new paradigm for systems on chip design,” in *Proc. Des., Autom. and Test in Eur. Conf. and Exhib.*, 2002, pp. 418-419.

[14] S. Kumar, A. Jantsch, K. Tiensyrja, A. Hemani, “A network on chip architecture and design methodology,” in *Proc. IEEE Comput. Soc. Annu. Conf. on VLSI*, 2002.