

کنترل دما در شبکه‌های روی تراشه سه‌بعدی با استفاده از مهاجرت وظیفه مبتنی بر الگوریتم تبرید تدریجی

منیره محبی مقدم، سیدحمید میرمحمدی و محمدحسین منشی

چندین لایه فعال را بر روی یکدیگر ممکن می‌سازد. کاهش چگالی ارتباطات و افزایش انعطاف‌پذیری در سیم‌بندی این ساختار، امکان مجتمع‌سازی بهتری را فراهم می‌آورد. ترکیب دو رویکرد اشاره‌شده که با عنوان 3D NoC شناخته می‌شود، طراحان را قادر می‌سازد که از مزایای هر دو روش بهره‌گیرند [۲].

با وجود مزایای زیاد این ترکیب، سه‌بعدی‌سازی چالش‌هایی را نیز به همراه دارد. عیب اصلی این طرح‌ها، افزایش چگالی توان در واحد سطح به دلیل کوچک‌شدن مساحت تراشه و متعاقب آن افزایش دمای تراشه و ایجاد نقاط داغ است. این افزایش دما، تأثیرات نامطلوبی همچون کاهش قابلیت اطمینان، تنزل کارایی و افزایش توان مصرفی را به دنبال دارد [۳]. از آنجا که این مسأله، شبکه‌های روی تراشه سه‌بعدی را نیز به طور مشابه تحت تأثیر قرار می‌دهد، طراحی و استفاده از الگوریتم‌های مدیریت دما در این نوع تراشه‌ها از اهمیت بالایی برخوردار است.

تاکنون محققان با ارائه راه‌حل‌های گوناگونی سعی نموده‌اند تا این مشکل را حل کرده یا کاهش دهند. به دلیل تأثیر زیاد مسیریاب‌ها در ایجاد نقاط داغ، طراحی مناسب الگوریتم‌های مسیریابی و معماری مسیریاب‌ها می‌تواند نقش مؤثری را در این زمینه ایفا نماید [۳] و [۴]. از طرف دیگر، برخی دیگر از محققان با استفاده از شیوه‌هایی همچون نگاشت و زمان‌بندی وظایف به نقش هسته‌های پردازشی و چگونگی نگاشت و جایابی وظایف بر روی این عناصر پرداخته‌اند [۵] تا [۸]. مقاله پیش رو در دسته دوم از این گروه قرار می‌گیرد از آنجا که تمرکز آن بیشتر بر روی توان محاسبات است.

به طور کلی نگاشت وظایف می‌تواند به صورت ایستا یا پویا انجام شود. در روش‌های نگاشت ایستا که نوعی روش برون‌خط^۳ است، تخصیص پردازنده به وظایف قبل از اجرای برنامه انجام می‌پذیرد و به همین علت، الگوریتم باید به ازای هر کاربرد، مجدداً اجرا شود تا نحوه نگاشت وظایف تعیین شود. ضمن این که در زمان اجرا، امکان تغییر نگاشت وجود ندارد. در مقابل، روش‌های نگاشت پویا امکان تخصیص پردازنده را به وظایفی که در زمان اجرا به سیستم وارد می‌شوند نیز فراهم می‌آورند. هر چند که این روش‌ها نیز امکانی را برای تغییر مکان وظایف در حین اجرا فراهم نمی‌آورند ولی عملکرد بهتری نسبت به روش‌های نگاشت ایستا دارند. مهاجرت وظیفه^۴ گونه‌ای از نگاشت پویاست که امکان تغییر مکان وظایف تخصیص داده شده فعلی را فراهم می‌آورد.

یکی از کارکردهای مهاجرت وظایف، تعادل گرمای تولیدی در میان هسته‌های پردازشی است [۸]. از این نظر در مقابل روش‌های همچون مقیاس‌گذاری پویای فرکانس و ولتاژ (DVFS) قرار می‌گیرد که با کاهش چگالی توان سعی می‌کنند دمای تراشه را کاهش دهند بدون این که

چکیده: ترکیب پیشته‌سازی سه‌بعدی و شبکه روی تراشه که با عنوان شبکه روی تراشه سه‌بعدی شناخته می‌شود، مزایای متعددی همانند کاهش تأخیر انتشار، کاهش مساحت تراشه و همچنین افزایش پهنای باند را به همراه دارد. علی‌رغم مزایای متعدد، پیشته‌سازی سه‌بعدی سبب افزایش چگالی توان در واحد سطح تراشه و متعاقب آن افزایش دمای تراشه می‌شود که این مسأله چالش‌های جدیدی را ایجاد می‌نماید. افزایش دما، کاهش قابلیت اطمینان و تنزل کارایی را به دنبال خواهد داشت و در نتیجه طراحی الگوریتم‌های مدیریت حرارتی برای این نوع سیستم‌ها امری ضروری به نظر می‌رسد. در این مقاله برای این منظور یک الگوریتم مبتنی بر مهاجرت وظایف ارائه شده است. انتخاب مقصد مهاجرت برای وظایف روی هسته‌های داغ، یک مسأله NP-Complete است که می‌توان آن را با روش‌های ابتکاری حل کرد. برای این منظور در روش پیشنهادی از الگوریتم تبرید تدریجی بهره‌گرفته‌ایم. در تعیین مقصد مهاجرت، علاوه بر دمای هسته‌ها، سربار ناشی از مهاجرت را نیز لحاظ کرده‌ایم. نتایج ارزیابی حاکی از آن است که این روش می‌تواند تا ۲۸ درصد، دمای بیشینه تراشه را برای محکی با بیشترین تعداد وظایف کاهش دهد، ضمن این که تأثیر آن بر روی کارایی اندک است.

کلیدواژه: الگوریتم تبرید تدریجی، شبکه روی تراشه سه‌بعدی، مدیریت دما، مهاجرت وظیفه.

۱- مقدمه

سیستم‌های روی تراشه چندپردازنده (MPSoC) شامل چندین پردازنده، حافظه و اجزای آنالوگ هستند که بر روی یک تراشه قرار گرفته‌اند. در این سیستم‌ها که ده‌ها تا صدها عنصر پردازشی بر روی یک زیرلایه قرار گرفته‌اند، ارتباطات نقش مهمی را در میزان کارایی و توان مصرفی سیستم ایفا می‌کند [۱]. معماری باس به دلیل مشکلاتی همچون قابلیت مقیاس‌پذیری کم و تأخیر انتشار بالا، زیرساخت ارتباطی مناسبی را برای این منظور فراهم نمی‌آورد. از این رو ارائه یک زیرساخت ارتباطی کارا و قابل اطمینان ضروری به نظر می‌رسد. یکی از راه‌حل‌های ارائه‌شده، شبکه روی تراشه^۱ (NoC) است. NoC، قابلیت مقیاس‌بندی و کارایی مصرف توان ساختارهای SoC را بهبود می‌بخشد. راهکار پیشنهادی دیگر، پیشته‌سازی سه‌بعدی^۲ است. پیشته‌سازی سه‌بعدی، پیشته‌سازی عمودی

این مقاله در تاریخ ۹ خرداد ماه ۱۳۹۳ دریافت و در تاریخ ۳۰ دی ماه ۱۳۹۴ بازنگری شد.

منیره محبی مقدم، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، (email: monireh.mohebbi@cc.iut.ac.ir).

سیدحمید میرمحمدی، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه صنعتی اصفهان، اصفهان، (email: h_mirmohammadi@cc.iut.ac.ir).

محمدحسین منشی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، (email: manshaei@cc.iut.ac.ir).

3. Offline

4. Task Migration

1. Network on Chip

2. 3D Stacking

بهتری را در مقایسه با الگوریتم دوم از خود نشان می‌دهد. این الگوریتم نیز از نوع نگاشت ایستا است. در مقایسه با این دو روش، مهاجرت وظایف به جهت امکان تغییر مکان وظایف پس از تخصیص اولیه و با توجه به شرایط دمایی تراشه، بر نگاشت ایستا برتری دارد.

مبحث مهاجرت وظایف به منظور کنترل دمایی تراشه‌های سه‌بعدی در [۱۲] و با تکیه بر نظریه بازی‌ها به کار گرفته شده است. در این روش که نوعی الگوریتم توزیع شده است هر هسته پس از این که خود را به عنوان یک هسته داغ تشخیص می‌دهد، سعی می‌کند بهترین مقصد را برای مهاجرت بیابد. برای این منظور هر هسته داغ، لیستی از هسته‌های خنک را می‌یابد که در صورت جابه‌جایی وظایف موجود بر روی آنها، دمای هر دو هسته در پایان کمتر از یک آستانه دمایی مشخص باشد. پس از آن هر هسته بدون در نظر گرفتن عوامل دمایی، بهترین موقعیت را برای جابه‌جایی از لحاظ کارایی (با در نظر گرفتن نرخ ارتباطی بین وظایف و فاصله بین هر دو وظیفه موجود) می‌یابد. سپس در لیست هسته‌های خنک، هسته‌ای که نسبت به سایرین به موقعیت گفته شده نزدیک‌تر باشد انتخاب می‌شود. همان طور که در این مقاله آمده است، ساختار توزیع شده الگوریتم سبب می‌شود که مقاصد مهاجرت انتخابی توسط هسته‌های داغ هم‌پوشانی داشته باشند. به سبب این مورد، در فرایندی پس از انتخاب مقصد مهاجرت، هسته‌ها باید اطلاعات خود را با یکدیگر مبادله نمایند تا در صورت وجود هم‌پوشانی، از بین هر دو وظیفه با مقاصد مهاجرت یکسان، وظیفه‌ای با بالاتر برای مهاجرت انتخاب می‌شود. در مقایسه با این روش، الگوریتم پیشنهادی ما از چند جنبه متفاوت بوده و برتری دارد. اول این که تعیین مقصد مهاجرت در [۱۲] بر پایه دو معیار دما و وزن ارتباطی بین وظایف انجام گرفته است حال آن که در الگوریتم پیشنهادی در این مقاله، علاوه بر موارد ذکر شده، سربار مهاجرت نیز لحاظ شده است. اهمیت این امر زمانی مشهودتر خواهد بود که تعداد هسته‌های داغ زیاد باشد. از طرفی در تابع هزینه که به جهت انتخاب مقصد مهاجرت در نظر گرفته شده می‌توان میزان تأثیر سه عامل دما، وزن ارتباطی وظایف و سربار مهاجرت را با کمک ضرایب پارامتری تعیین نمود. تفاوت موجود میان کاربردهای گوناگون سبب می‌شود که تعریف پارامتری تابع هزینه، آزادی عمل بیشتری را برای کاربر فراهم آورد. از طرفی از آنجا که روش ارائه شده در [۱۲] از جمله الگوریتم‌های توزیع شده است، هر وظیفه به طور مجزا از سایرین در مورد مقصد مهاجرت خود تصمیم‌گیری می‌کند. این دیدگاه سبب می‌شود که در پایان هر بار اجرای الگوریتم، نیاز به تبادل اطلاعات بین وظایف باشد و گاهی نیاز به چندین بار اجرای الگوریتم برای یافتن مقصد مناسب برای تمامی هسته‌های داغ است. از طرفی با این که برای انتخاب مقصد مهاجرت، پیش‌بینی دمایی انجام می‌گیرد تا دمای مقصد مهاجرت پس از جابه‌جایی از آستانه TH_1 بالاتر نرود، با این وجود ممکن است پس از پایان مهاجرت و به دلیل ساختار پشته‌ای تراشه‌های سه‌بعدی، هسته‌هایی که بر روی یکدیگر قرار می‌گیرند، توان مصرفی بالایی داشته و سبب ایجاد نقاط داغ جدیدی در سطح تراشه گردند. به عبارت دیگر از آنجایی که در انتخاب مقصد مهاجرت توسط یک پردازنده تا زمان عدم وجود هم‌پوشانی، انتخاب سایرین دخالت داده نمی‌شود، امکان تبدیل مقاصد مهاجرت به نقاط داغ در مراحل بعدی وجود دارد. در الگوریتم پیشنهادی ما، مقاصد مهاجرت برای تمامی هسته‌های داغ به صورت هم‌زمان و نه مجزا از سایرین تعیین می‌گردند.

۳- الگوریتم پیشنهادی

در این بخش در ابتدا فرضیات مسأله را بیان خواهیم نمود. سپس با

وارپانس دمایی بین هسته‌ها را در نظر بگیرند. ضمن این که DVFS به دلیل کاهش فرکانس پردازنده، کارایی سیستم را به میزان زیادی تحت تأثیر قرار می‌دهد [۹].

با توجه به مزایای روش مهاجرت در قیاس با سایر انواع روش‌های اشاره شده، در این مقاله از این شیوه برای نیل به هدف کاهش دمای بیشینه تراشه و توزیع مناسب دمایی بین هسته‌های پردازشی استفاده نموده‌ایم. جابه‌جایی وظایف بین هسته‌ها از لحاظ ترافیکی و دمایی نوعی سربار را به سیستم تحمیل می‌کند که آن را نیز در مدل‌سازی وارد نموده‌ایم. انتخاب مقصد مهاجرت با توجه به این اهداف، یک مسأله NP-Complete است اما می‌توان آن را با تقریب با الگوریتم‌های ابتکاری حل نمود. بدین منظور از الگوریتم تیرید تدریجی^۱ (SA) استفاده کرده‌ایم و تابع هزینه موجود در آن را با توجه به اهداف مسأله، تعیین نموده‌ایم. SA یکی از روش‌های ابتکاری حل مسایل بهینه‌سازی است که مزایایی همچون سادگی پیاده‌سازی، قابلیت اعمال به مسایل بهینه‌سازی ترکیبی و ارائه پاسخ‌های منطقی را به همراه دارد، ضمن این که نوعی مصالحه بین بهینگی جواب و پیچیدگی زمانی اجرا را در نظر می‌گیرد [۱۰] و [۱۱]. ادامه مقاله به صورت زیر خواهد بود: در بخش ۲ مروری بر کارهای پیشین خواهیم داشت. در بخش ۳ جزئیات الگوریتم پیشنهادی تشریح خواهد شد. در بخش ۴ به بیان روش ارزیابی و نتایج حاصل از آن خواهیم پرداخت و نهایتاً مقاله در بخش ۵ با نتیجه‌گیری به پایان خواهد رسید.

۲- پژوهش‌های مرتبط

نویسندگان در [۵] و [۶] از زمان‌بندی وظایف به منظور کنترل دما در NoC ۳D بهره گرفته‌اند. در [۶] یک روش زمان‌بندی وظایف به منظور محدود نمودن دمای تراشه و حداقل نمودن کاهش کارایی با ممانعت از تخصیص وظایف به هسته‌های با فاصله زیاد از داده‌های مورد نیازشان، معرفی شده است. با وجود این که این الگوریتم و همچنین روش پیشنهادی در [۵] دو پارامتر دما و کارایی را مد نظر قرار داده‌اند، اما تأکید آنها بر کارایی سیستم بوده و نه حداکثر نمودن کاهش دمای تراشه. روش ارائه شده در این مقاله، بیشتر بر کاهش دمای تراشه متمرکز می‌باشد ضمن این که تأثیر آن بر تنزل کارایی را نیز مد نظر قرار می‌دهد.

در [۷] یک روش نگاشت و جایابی آگاه از دما و ارتباطات با استفاده از الگوریتم ژنتیک برای ۳D NoC معرفی شده که بر طبق نتایج ارائه شده در این مقاله، کاهش قابل ملاحظه‌ای در دمای سیستم در این روش نسبت به نگاشت تصادفی مشاهده می‌شود. در این مقاله ۳ نوع بهینه‌سازی مطرح شده است. دو بهینه‌سازی اول به ترتیب برای دما و ارتباطات انجام شده و روش سوم ترکیبی از دو روش پیشین می‌باشد. تمامی این روش‌ها از نوع نگاشت ایستا می‌باشد و قابلیت تغییر تخصیص وظایف به پردازنده‌ها را در زمان اجرا ندارند.

در [۸] سه روش نگاشت آگاه از دما و مبتنی بر برنامه‌نویسی خطی صحیح (ILP) برای ۳D NoC با هدف حداقل نمودن دمای بیشینه تراشه معرفی شده است. اولین الگوریتم که یک روش نگاشت بهینه است به دلیل پیچیدگی بالا، قابلیت پیاده‌سازی بر روی این نوع تراشه‌ها را ندارد. دو الگوریتم دیگر، سعی در کاهش این پیچیدگی داشته‌اند. الگوریتم دوم با استفاده از بخش‌بندی به این هدف دست یافته است. الگوریتم سوم که ترکیبی از دو الگوریتم قبل بوده و یک الگوریتم تقریباً بهینه است در مقایسه با روش اول پیچیدگی قابل قبولی داشته، ضمن این که کارایی

1. Simulated Annealing

آستانه دمایی مشخصی قرار داشته باشند. مدل دمایی مورد استفاده برای اندازه‌گیری دمای هسته‌های پردازشی، مشابه با مدل استفاده‌شده در [۸] و به صورت (۳) است. در این مدل، وابستگی حرارتی بین عناصر پردازشی موجود در لایه‌های عمودی در قالب مقاومت حرارتی بین لایه‌ها در نظر گرفته شده و واضح است که دمای صفحات بالاتر به دلیل تبادل حرارتی کمتر، بیشتر است

$$Th_{i,j,k} = T_{amb} + \sum_{m=1}^k \frac{R_{i,j,m}}{A} \times \left(\sum_{s=m}^n (P_{i,j,s} + PR_{i,j,s}) \right) \quad (3)$$

پارامترهای موجود در این رابطه به صورت زیر تعریف می‌شوند:

- $Th_{i,j,k}$: دمای عنصر پردازشی موجود در موقعیت (i, j, k) در شبکه روی تراشه سه‌بعدی.
- T_{amb} : دمای محیط.
- $R_{i,j,m}$: مقاومت حرارتی عنصر موجود در موقعیت (i, j, m) . این مقاومت بین عناصر موجود در لایه i و عناصر لایه $i-1$ وجود دارد. در لایه اول نیز این مقاومت با سینک حرارتی تعریف می‌شود.
- A : مساحت هر عنصر پردازشی.
- $P_{i,j,s}$: میانگین توان مصرفی پردازنده موجود در موقعیت (i, j, s) که بر حسب میانگین مصرف توان وظیفه در حال اجرا بر روی آن محاسبه می‌شود.
- $RP_{i,j,s}$: میانگین توان مصرفی مسیریاب موجود در مکان (i, j, k) که مقدار آن بر اساس میزان توان مصرفی مسیریاب به ازای هر بیت و میزان داده مسیریابی شده از طریق آن محاسبه می‌شود.
- در رابطه دمایی میانگین توان مصرفی هسته‌ها و مسیریاب‌ها محاسبه می‌گردد تا دمای حالت پایدار به دست آید.

۳-۲ الگوریتم تبرید

SA یک الگوریتم غیرحریصانه احتمالی است که فضای جستجوی یک مسأله را با حرکت از یک وضعیت دمایی بالا به یک وضعیت دمایی پایین‌تر پوشش می‌کند [۱۰]. در مسأله تعریف‌شده در هر سطح انرژی، بسته به تعداد هسته‌های داغ که مبدأ مهاجرت می‌باشند، یک یا چند وظیفه به عناصر پردازشی متفاوتی منتقل شده و هزینه این نگاشت جدید و مهاجرت ارزیابی می‌شود. الگوریتم همواره یک حرکت به مکان بهتر را می‌پذیرد و یک وضعیت بدتر با یک احتمال متغیر پذیرفته می‌شود. این احتمال با پیشروی الگوریتم کاهش می‌یابد، بنابراین الگوریتم حریصانه‌تر می‌شود. الگوریتم در نهایت با رسیدن به دمای نهایی و همچنین تعداد مشخصی حرکت ردشده متوالی یا اتمام هسته‌های داغ به پایان می‌رسد.

شکل ۱، شبه‌کد الگوریتم SA را نشان می‌دهد و علائم و توابع به کار رفته در این الگوریتم در جدول ۱ تشریح شده است. در انتخاب برخی پارامترهای این الگوریتم، نتایج حاصل از بررسی‌های انجام‌شده توسط نویسندگان در [۱۳] در نظر گرفته شده است. Orsila و همکاران در [۱۳] بر کاربرد SA در نگاشت وظایف تمرکز نموده‌اند و به بیان پارامترهای مختلف دخیل در این الگوریتم متناسب با بحث نگاشت وظایف پرداخته‌اند. در ادامه، پنج Case Study که در آنها از این روش جهت نگاشت وظایف بهره برده شده را بررسی کرده و در نهایت تعدادی قانون جهت انتخاب توابع و پارامترهای مختلف SA برای استفاده در مسأله نگاشت وظایف استخراج کرده‌اند به نحوی که یک مصالحه مناسب بین کیفیت پاسخ ارائه‌شده و زمان اجرا برقرار باشد. اگرچه این رهنمون‌ها برای نگاشت وظایف ذکر شده اما می‌تواند برای انتخاب برخی پارامترهای موجود در SA در مسأله مهاجرت نیز به کار برده شود چرا که مهاجرت، نوعی

SIMULATED_ANNEALING (S0)

```

S ← S0
C ← COST(S0)
R ← 0
for i ← 0 to ∞
do T ← TEMP(i)
   Snew ← MOVE(S,T)
   Cnew ← COST(Snew)
   ΔC ← Cnew - C
   if ΔC < 0 or ACCEPT(ΔC,T)
then if Cnew < Cbest
then Sbest ← Snew
   Cbest ← Cnew
   S ← Snew
   C ← Cnew
   R ← 0
else R ← R+1
if TERMINATE(i,R) = true
then break
return Sbest
    
```

شکل ۱: شبه‌کد الگوریتم SA (پارامترها در جدول ۱ تشریح شده‌اند) [۱۱].

تعریف مختصری از الگوریتم تبرید و پارامترهای دخیل در آن به بیان نحوه انتخاب مقصد مهاجرت خواهیم پرداخت.

۳-۱ تعریف مسأله

فرض شده که سیستم از تعدادی عنصر پردازشی یکسان تشکیل شده است بدین معنا که هر یک از وظایف می‌توانند توسط هر یک از عناصر پردازشی اجرا شوند. در ابتدا تمامی وظایف به پردازنده‌ها به صورت تصادفی نگاشت می‌شوند و به هر عنصر پردازشی حداکثر یک وظیفه تخصیص می‌یابد. چگونگی ارتباط وظایف با یکدیگر و نرخ ارتباطی آنها، با گراف ارتباطی وظایف نشان داده می‌شود. گراف ارتباطی وظایف یک گراف جهت‌دار بدون دور است که با $CTG = (T, E)$ نمایش داده می‌شود و $T_i \in T$ و $e_{i,j} \in E$ به ترتیب نشان‌دهنده وظیفه i ام و وابستگی ارتباطی بین دو وظیفه T_i و T_j است. حجم ارتباطی هر یال نیز با $w(i, j)$ نشان داده می‌شود [۸].

نحوه نگاشت وظایف به عناصر پردازشی توسط ماتریس نگاشت M با $|PE| \times |T|$ عنصر نشان داده می‌شود که $|PE|$ و $|T|$ به ترتیب تعداد پردازنده‌ها و تعداد وظایف موجود در CTG است. به هر عنصر ماتریس M ، $M_{T_m}^{Pe_i}$ ، بسته به این که وظیفه T_m بر روی هسته Pe_i قرار گرفته باشد یا خیر، مقدار یک یا صفر را اختصاص می‌دهیم. روابط (۱) و (۲) بیان ریاضی این مطلب را نشان می‌دهند

$$\forall T_m \in T: \sum_{Pe_i \in PE} M_{T_m}^{Pe_i} = 1 \quad (1)$$

$$M_{T_m}^{Pe_i} = \begin{cases} 1 & \text{if } map(T_m) = Pe_i \\ 0 & \text{else} \end{cases} \quad (2)$$

تعیین نقاط داغ در تراشه به توجه به آستانه دمایی مشخصی که آن را با TH نمایش می‌دهیم، انجام می‌پذیرد. این آستانه دمایی باید کمتر از آستانه دمایی آسیب‌پذیری تراشه تعریف شود. هر هسته با دمای بالاتر از این آستانه به عنوان یک نقطه داغ برچسب می‌خورد.

اصلی‌ترین هدف در الگوریتم پیشنهادی، مدیریت دمایی تراشه است به نحوی که تا حد امکان دمای بیشینه تراشه کاهش یابد. به عبارت دیگر، دمای تمامی هسته‌های پردازشی پس از مهاجرت در سطحی پایین‌تر از

نتایج عکس مورد قبل را به دنبال خواهد داشت. در این مطالعه، مورد اخیر یعنی محدود کردن فضای جستجو با استفاده از آستانه دمایی لحاظ شده است.

در صورتی که هیچ وظیفه‌ای به هسته خنکی که به عنوان مقصد این جابه‌جایی در نظر گرفته می‌شود تخصیص نیافته باشد، تعداد انتقال‌ها به ازای نقطه داغ مورد نظر برابر با یک خواهد بود و در غیر این صورت دو انتقال انجام خواهد گرفت. در هر بار اجرای این تابع در صورتی که تعداد هسته‌ها با دمای کمتر از آستانه دمایی TH_2 بیشتر از تعداد هسته‌های داغ موجود در سیستم باشد، به ازای هر هسته داغ یک هسته خنک انتخاب شده و جابه‌جایی‌ها انجام می‌پذیرد. در صورتی که تعداد هسته‌های خنک، کمتر از هسته‌های داغ موجود در سیستم باشد هسته‌های داغ با توجه به پارامتر دمای هسته به صورت نزولی مرتب شده و از ابتدای این لیست به تعداد هسته‌های خنک، عنصر پردازشی داغ انتخاب شده و انتقال انجام می‌پذیرد.

۳-۲-۲ تابع ACCEPT

از بین توابع مختلفی که می‌توان برای تابع پذیرش برگزید، تابع نمایشی معکوس شده که اختلاف هزینه در آن نرمالیزه شده، انتخاب گردیده که به صورت (۴) می‌باشد. این انتخاب، برگرفته از نتایج تحلیل انجام شده در [۱۳] است. نویسندگان در این مقاله با مقایسه توابع مختلف استفاده شده برای نگاشت وظایف با استفاده از روش تبرید، نتیجه می‌گیرند که تابعی به فرم نمایشی معکوس می‌تواند انتخاب مناسب‌تری برای این گونه مسایل باشد. نرمال‌سازی انجام شده در این رابطه سبب می‌شود که احتمال پذیرش در یک محدوده مناسب قرار داشته باشد حتی اگر تابع هزینه تغییر کند. ضمن این که تابع این امکان را فراهم می‌آورد که مقدار دمای T مستقل از تابع هزینه باشد یعنی در محدوده $(0, 1)$ قرار گیرد [۱۱]. مقدار T_f و T_f به ترتیب ۱ و ۰٫۰۰۱ انتخاب شده‌اند

$$\text{Accept}(\Delta C, T) = \text{True} \Leftrightarrow \text{random}() < \frac{1}{1 + e^{\frac{\Delta C}{CT}}} \quad (4)$$

۳-۲-۳ تابع TEMP

خروجی این تابع، یک سطح دمایی در فرایند تبرید را بیان می‌کند و به صورت تابعی از تعداد تکرارهای الگوریتم است. تعریف این تابع در این مقاله به صورت (۵) می‌باشد که در آن q یک فاکتور هندسی است که در بازه $(0.98, 0.99)$ قرار دارد. در این رابطه T_f مقدار دمای اولیه، i شمارنده تکرار و L تعداد تکرارها در هر سطح دمایی است [۱۳]

$$\text{Temp}(i) = T_f \left[\frac{i}{L} \right] \quad (5)$$

۳-۲-۴ تابع TERMINATE

این تابع که با رابطه زیر نشان داده شده است، تعیین‌کننده شرط پایان الگوریتم است و ترکیبی از دمای نهایی، تعداد پاسخ‌های رده‌ده- که نشان‌دهنده تعداد حالاتی است که مقدار تابع هدف پس از جابه‌جایی به وضعیت بهتری نرسیده است- و تعداد هسته‌های داغ موجود در سیستم خواهد بود

$$\text{Terminate}(i, R) = \text{True} \Leftrightarrow (\text{Temp}(i) < T_f \wedge R \geq R_{\max}) \vee (T_{Pe_i} < TH_1, \forall Pe_i \in PE) \quad (6)$$

مقدار R_{\max} در این رابطه برابر L در نظر گرفته شده که معادل تعداد تکرارها در هر سطح دمایی است. $L \geq N \times M$ در نظر گرفته شده که N تعداد وظایف موجود بر روی هسته‌های داغ و M نقاط خنک موجود

جدول ۱: علائم و پارامترهای به کار برده شده در شبه‌کد الگوریتم SA [۱۱].

علامت	محدوده مقادیر	تعریف
Accept($\Delta C, T$)	{درست، غلط}	در صورت پذیرش پاسخ تولیدشده، مقدار درست و در غیر این صورت مقدار غلط را برمی‌گرداند
$C = \text{Cost}()$	$C > 0$	هزینه پذیرش (که باید حداقل شود)
C_i	$C_i > 0$	هزینه اولیه
C_{new}	$C_{new} > 0$	هزینه وضعیت بعدی
i	$i > 0$	شمارنده تکرار
L	$L > 0$	تعداد تکرار در هر سطح دما
N	$N > 0$	تعداد هسته‌های داغ
M	$M > 0$	تعداد هسته‌های خنک
q	$0 < q < 1$	فاکتور مقیاس‌بندی هندسی دما
R	$R \geq 0$	تعداد حرکات رده‌ده متوالی
S	فضای مهاجرت	وضعیت پذیرفته‌شده
S_i	فضای مهاجرت	وضعیت اولیه
S_{new}	فضای مهاجرت	وضعیت بعدی
Move(S, T)	فضای مهاجرت	تابعی است که مقدار وضعیت بعدی را برمی‌گرداند
$T = \text{Temp}(i)$	$T > 0$	مقدار دما را با توجه به عدد تکرار برمی‌گرداند
T_i	$T_i > 0$	دمای اولیه
T_f	$0 < T_f < T_i$	دمای نهایی
Terminate(i, R)	{درست، غلط}	در صورت پایان، مقدار درست و در صورت ادامه، مقدار غلط را برمی‌گرداند

نگاشت پویاست که در زمان اجرا انجام می‌پذیرد ضمن این که تمامی وظایف موجود در سیستم نیازی به جابه‌جایی و تخصیص مجدد نخواهند داشت و تنها وظایف موجود بر روی نقاط داغ موجود در سطح سیستم باید تغییر مکان دهند. البته قابل ذکر است که تابع هزینه موجود در این الگوریتم که اصلی‌ترین بخش آن است با توجه به اهداف تعیین شده، انتخاب شده است. در ادامه به بیان جزئی‌تر نحوه انتخاب و مقدار هر یک از این توابع و پارامترها با توجه به کاربرد مورد نظر خواهیم پرداخت.

۳-۲-۱ تابع MOVE

تابع MOVE، ایجاد یک همسایگی را به منظور انتقال وظیفه موجود بر روی هسته داغ به گره مربوط بر عهده دارد. این جابه‌جایی می‌تواند بین هسته‌های موجود بر روی لایه‌های مختلف انجام پذیرد. پذیرش یا عدم پذیرش همسایگی تولیدشده به میزان اختلاف هزینه وضعیت بعدی با وضعیت فعلی و نیز تابع ACCEPT وابسته است. تولید همسایگی به صورت تصادفی در فضای همسایگی انجام می‌پذیرد. فضای همسایگی می‌تواند شامل کلیه هسته‌هایی باشد که دمایی کمتر از آستانه TH_1 دارند و اولویت‌دهی به هسته‌های با دمای کمتر صرفاً با توجه به تابع هزینه انجام پذیرد یا این که به جهت اهمیت بیشتر پارامتر دما، فضای جستجو شامل هسته‌هایی با دمای کمتر از یک آستانه دمایی دیگری با عنوان TH_2 باشد ($TH_2 < TH_1$). انتخاب مقدار مناسب برای TH_2 از اهمیت زیادی برخوردار است چرا که تخصیص مقدار کم، سبب کوچک شدن فضای جستجو و متعاقب آن، سریع‌تر شدن فرایند همگرایی خواهد شد. از طرفی ریسک عدم وجود مقصد مهاجرت به ازای تمامی هسته‌های داغ را افزایش می‌دهد. انتخاب مقدار زیاد و نزدیک به TH_1 ،

$$dist(Pe_i, Pe_j) = |x_i - x_j| + |y_i - y_j| + |z_i - z_j| \quad (11)$$

تابع هزینه، یک تابع پارامتری است که از سه بخش C_1 تا C_7 تشکیل شده است. به منظور یکسان‌سازی واحدهای اندازه‌گیری دمای تراشه، فاصله منتهن و نرخ ارتباطی بین وظایف، هر یک از این بخش‌ها با استفاده از مقادیر حداکثرشان نرمال شده‌اند. به علاوه، این نرمال‌سازی کمک می‌کند تا مقادیر این سه بخش در محدوده [0, 1] قرار گیرند. بدین ترتیب می‌توان به کمک پارامترهایی که در تابع Cost در هر بخش ضرب شده است، میزان اهمیت هر کدام را بدون وجود نگرانی در مورد محدوده مقادیر آن، تعیین نمود.

در تابع C_1 دمای بیشینه هسته‌های پردازشی پس از جابه‌جایی با استفاده از دمای حداکثر تراشه در وضعیت اولیه نرمال شده است و تابع C_7 ، سربار مهاجرت را محاسبه می‌نماید. هر فرایند مهاجرت، دو هسته خنک (CCore) و داغ (HCore) را درگیر می‌نماید. در هر اجرای تابع Move بسته به تعداد هسته‌های خنک، تمام یا بخشی از وظایف موجود بر روی هسته‌های داغ مهاجرت می‌نمایند و این سربار به تعداد انتقال‌ها و همچنین اندازه وظایف (TS) بستگی دارد. فاصله بین دو هسته داغ و خنک (dist) نیز پارامتر مؤثر دیگر در این بخش است. C_7 تأثیر نرخ ارتباطی بین وظایف را در مهاجرت مدل می‌کند. در رابطه متناسب به C_7 ، در صورت کسر، مجموع نرخ ارتباطی بین هر جفت وظایف در فاصله بین هسته‌های پردازشی که این جفت وظایف بر روی آنها قرار دارند، ضرب شده است. این مقدار با عبارتی به همین صورت ولی با در نظر گرفتن حداکثر فاصله بین هر دو جفت وظیفه نرمال شده است. در نهایت تابع هزینه، ترکیب خطی C_i ها است. ضرایب این سه بخش با توجه به میزان اهمیت هر یک از C_i ها تعیین می‌گردد.

۴- پیاده‌سازی و ارزیابی نتایج

در این بخش نتایج حاصل از شبیه‌سازی آورده شده است. شبیه‌ساز BookSim [۱۴] به عنوان شبیه‌ساز شبکه روی تراشه مورد استفاده قرار گرفته است. به منظور محاسبه توان مصرفی اتصالات و مسیریاب‌های موجود در شبکه روی تراشه، کتابخانه Orion۲ [۱۵] به آن افزوده شده است. این کتابخانه میزان توان مصرفی شبکه روی تراشه را در سطح معماری تخمین می‌زند. از آنجا که مدل توانی در Orion۲ برای شبکه‌های دوبعدی ارائه شده است، اطلاعات اضافی مورد نیاز به منظور در نظر گرفتن اتصالات عمودی به آن اضافه شده است [۱۶]. شبیه‌سازی‌ها در یک سیستم ۱۲۸ بیت با دو کانال مجازی به ازای هر لینک انجام شده است. فرکانس کاری برابر با ۲ گیگاهرتز تنظیم شده و مسیریابی به ترتیب بعد به عنوان الگوریتم مسیریابی به کار گرفته شده است. هر شبیه‌سازی برای ۳ میلیون سیکل انجام می‌پذیرد. الگوریتم پیشنهادی در MATLAB پیاده شده است و خروجی این الگوریتم به صورت یک فایل TraceFile به منظور شبیه‌سازی و محاسبه تأخیر و توان مصرفی به شبیه‌ساز داده می‌شود.

برای ارزیابی الگوریتم پیشنهادی از شش محک^۱ با گراف وظایف و مشخصه‌های مختلف متعلق به کاربردهای واقعی بهره‌برده‌ایم (جدول ۲). اندازه شبکه روی تراشه از نظر تعداد لایه‌ها و تعداد عناصر پردازشی در هر لایه، تعداد وظایف و نرخ ارتباطی به ازای محک‌های مختلف متفاوت است.

جدول ۲: اطلاعات محک‌های مورد ارزیابی [۸].

نام محک	تعداد وظایف	تعداد یال‌ها	اندازه ابعاد NoC
GSM Coder	۵۳	۸۱	۴،۴،۴
GSM Decoder	۳۴	۴۸	۳،۴،۳
MMS	۲۹	۲۳	۳،۴،۳
VOPD	۱۲	۱۵	۲،۳،۲
MWD	۱۲	۱۳	۲،۳،۲
MPEG۴	۱۲	۲۳	۲،۳،۲

در سطح تراشه است. الگوریتم SA در دو صورت به پایان می‌رسد. حالت اول زمانی است که پس از جابه‌جایی وظایف و ایجاد نگاشت جدید، بیشینه دمای هسته‌ها (T_{pe_i}) کمتر از آستانه دمایی TH_{λ} باشد و حالت دوم زمانی است که $Temp(i)$ کمتر از مقدار نهایی بوده، ضمن این که تعداد پاسخ‌های نامطلوب حداقل به اندازه R_{max} باشد.

۳-۲-۵ تابع COST

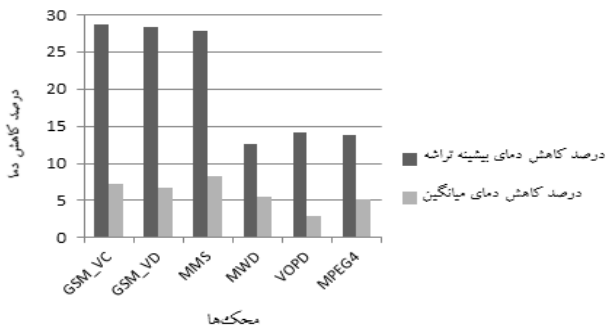
تابع هزینه یکی از پارامترهای اساسی الگوریتم SA است که باید با توجه به نوع مسأله تعیین گردد. از آنجا که هدف از مهاجرت، کاهش حداکثر دمای تراشه است یکی از اجزای اصلی تابع هزینه، معیار دمایی خواهد بود. از طرفی، واضح است که باید سربار مهاجرت نیز در تابع هزینه لحاظ شود. این سربار می‌تواند از حیث دمایی یا ترافیک افزوده شده به سیستم در نظر گرفته شود. آنچه مسلم است هرچه فاصله بین مبدأ و مقصد مهاجرت بیشتر باشد، انتقال داده به منظور جابه‌جایی وظایف، زمان بیشتری را صرف خواهد کرد ضمن این که ترافیک سیستم را بیشتر تحت تأثیر قرار خواهد داد. علاوه بر آن، تعداد مسیریاب‌های میانی که برای این انتقال به کار می‌روند افزایش خواهند یافت و به دلیل تأثیر مقدار داده عبوری از مسیریاب در میزان توان مصرفی آن، دمای حالت پایدار تراشه افزایش می‌یابد. با توجه به این نکته، سربار مهاجرت در این مطالعه، بر حسب این فاصله لحاظ شده و از آنجا که پیاده‌سازی‌ها برای یک شبکه با توپولوژی مش انجام گرفته که رایج‌ترین توپولوژی مورد استفاده در NoC است، فاصله در نظر گرفته شده، فاصله منتهن خواهد بود. فاکتور دیگر، میزان نرخ ارتباطی بین وظایف است. هرچه وظایف با وزن ارتباطی بیشتر بر روی پردازنده‌های نزدیک‌تر به یکدیگر قرار گیرند، ترافیک کمتری به شبکه تحمیل خواهد شد ضمن این که به دلیل استفاده از مسیریاب‌های میانی کمتر جهت انتقال داده، دمای هسته‌ها کمتر خواهد بود. در واقع این عامل در میزان دمای وضعیت پایدار و ترافیک سیستم تأثیرگذار خواهد بود و بنابراین تابع هزینه در نظر گرفته شده به صورت (۷) تا (۱۱) می‌باشد

$$Cost = \alpha \times C_1 + \beta \times C_7 + \gamma \times C_7 \quad (7)$$

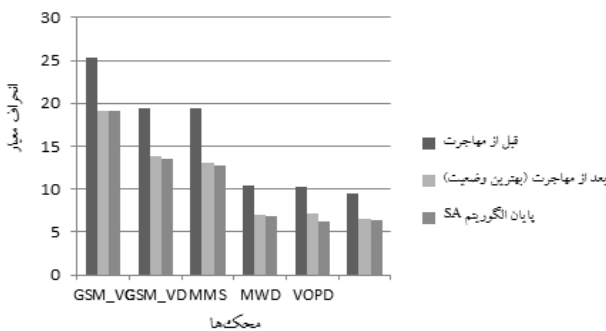
$$C_1 = \frac{\max T_{pe_i} \text{ (after move)}}{\max T_{pe_i} \text{ (for initial state)}} \quad (8)$$

$$C_7 = \frac{\sum_{\substack{i \in HCores \& \\ j \in CCores}} [(M_{T_m}^{Pe_i} \times TS_{Pe_i} + M_{T_m}^{Pe_j} \times TS_{Pe_j}) \times dist(Pe_i, Pe_j)]}{(\#ofHCores) \times \max TS_{Pe_i} \times 2 \times \max dist(Pe_i, Pe_j)} \quad (9)$$

$$C_7 = \frac{\sum_{i \neq j} (W_{T_m, T_n} \times dist(Pe_i \times Pe_j) \times M_{T_m}^{Pe_i} \times M_{T_n}^{Pe_j})}{\sum_{i \neq j} (W_{T_m, T_n} \times \max dist(Pe_i \times Pe_j) \times M_{T_m}^{Pe_i} \times M_{T_n}^{Pe_j})} \quad (10)$$



شکل ۴: درصد کاهش دمای بیشینه و دمای میانگین تراشه به ازای محک‌های مختلف.



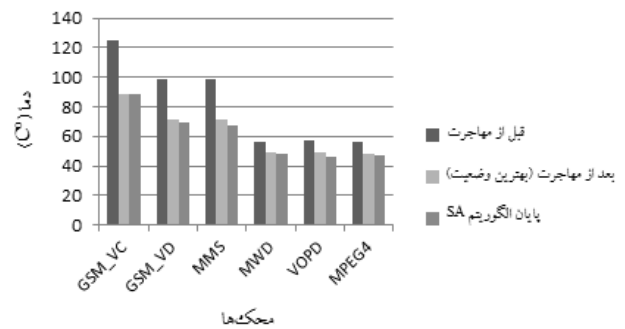
شکل ۵: انحراف معیار دمای هسته‌های پردازشی، قبل از مهاجرت و بعد از آن (به ازای وضعیت بهینه از لحاظ هزینه و پس از پایان الگوریتم SA).

یکی از مزایای مهاجرت، تعادل دمایی تراشه خواهد بود و این مسأله به معنی پراکندگی تقریباً متوازن دمای هسته‌ها در اطراف دمای میانگین است، به نظر می‌رسد انتخاب این مقدار برای این آستانه دمایی می‌تواند مفید باشد. از جمله پارامترهایی که بر میزان مصالحه بین کاهش دما و سربار ناشی از مهاجرت تأثیرگذارند، پارامترهای α ، β و γ هستند. انتخاب بین مقادیر مختلف برای این پارامترها باید با توجه به میزان اهمیت هر یک از عوامل دما و کارایی انجام پذیرد.

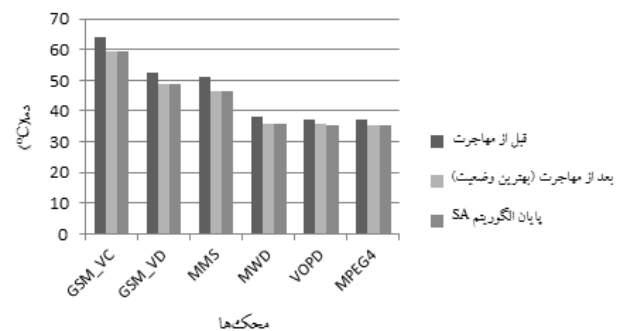
۴-۱ تأثیر مهاجرت بر کاهش دمای تراشه

انتخاب مقادیر α ، β و γ در مقادیر تابع هزینه و متناسب با آن در پذیرش و یا عدم پذیرش جواب‌های تولیدشده و در پاسخ نهایی تأثیرگذار خواهد بود. از آنجا که تمرکز اصلی در این پژوهش بر کاهش دمای تراشه است، وزن عبارت دمایی در این تابع، مقدار α ، برابر با ۰/۷۵ در نظر گرفته شده و مقدار دو پارامتر دیگر مساوی با یکدیگر و برابر با ۰/۱۲۵ لحاظ شده است.

به منظور بررسی تأثیر روش پیشنهادی در کاهش دمای تراشه، تحلیلی از نظر دمایی بر روی محک‌های گفته‌شده، انجام گرفته است. از آنجا که نگاشت اولیه یک نگاشت تصادفی است الگوریتم به ازای هر محک چندین بار اجرا شده و میانگین مقادیر به دست آمده به ازای هر حالت گزارش شده است. نتایج مربوط در شکل‌های ۲ تا ۵ آورده شده است. نمودار ۲ و ۳، بیشینه و میانگین دمای تراشه را قبل و بعد از مهاجرت نشان می‌دهند. از آنجایی که در تابع TERMINATE، وضعیت بهینه زمانی تعبیر خواهد کرد که علاوه بر پذیرش جواب، مقدار به دست آمده برای تابع هزینه از مقدار آن به ازای بهترین وضعیت قبلی کمتر باشد و در تابع هزینه علاوه بر پارامتر دمایی، سربار مهاجرت و نرخ ارتباطی لحاظ شده است، پس انتظار می‌رود که مقدار دمای تراشه به ازای حالت بهینه با مقدار متناظر آن در زمان پایان اجرای الگوریتم SA لزوماً یکسان نباشد. همان طور که انتظار می‌رفت هر چند تعداد وظایف موجود در سیستم و



شکل ۲: مقادیر دمای بیشینه تراشه قبل از مهاجرت و بعد از آن (به ازای وضعیت بهینه از لحاظ هزینه و پس از پایان الگوریتم SA).



شکل ۳: مقادیر دمای میانگین تراشه قبل از مهاجرت و بعد از آن (به ازای وضعیت بهینه از لحاظ هزینه و پس از پایان الگوریتم SA).

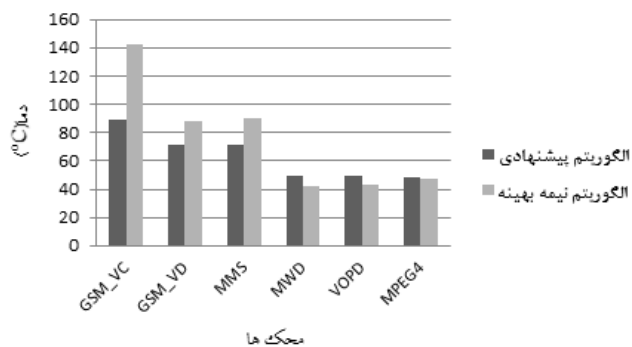
جدول ۳: اطلاعات فناوری ۴۵ نانومتر [۸].

تعداد لایه‌های فلز	ولتاژ منبع تغذیه (V_{dd} (ولت)
۱۲	۱٫۱
ضخامت (متر)	پهنای (متر)
$11e^{-6}$	$45e^{-6}$
Cu local	
$11e^{-6}$	$45e^{-6}$
Cu semi-global	
$162e^{-6}$	$67.5e^{-6}$
Cu global	
$72e^{-6}$	
ILD local	
$72e^{-6}$	
ILD semi-global	
$148e^{-6}$	
ILD global	
$2e^{-6}$	
Glue layer	
$10e^{-6}$	
Si layer	
$500e^{-6}$	
Si bulk	

جهت محاسبه دمای عناصر پردازشی از طریق مدل دمایی، نیاز به توان مصرفی وظایف در حال اجرا بر روی آنها می‌باشد که این میزان توان بر اساس فناوری مورد استفاده تخصیص داده شده است [۱۷]. اطلاعات مورد نیاز جهت محاسبه سایر پارامترهای موجود در مدل دمایی بیان شده در بخش ۳-۱ در جدول ۳ آورده شده است [۸].

مقدار TH_1 برای تراشه‌های مختلف متفاوت است. هرچه تعداد لایه‌های عمودی تراشه بیشتر باشد، این دما افزایش خواهد یافت ضمن این که نرخ ارتباطی وظایف نیز می‌تواند در تعیین مقدار این پارامتر اثرگذار باشد. برای این منظور مقدار پارامتر TH_1 برای محک‌هایی با ۲، ۳ و ۴ لایه به ترتیب ۵۰، ۷۰ و ۹۰ درجه لحاظ شده است.

مقدار TH_2 ، آستانه دمایی تعیین هسته‌های خنک، تأثیر مستقیمی در میزان کاهش دمای حداکثر خواهد داشت. این دما در نتایجی که در ادامه آورده شده برابر با دمای میانگین در نظر گرفته شده است. از آنجایی که



شکل ۸: مقایسه بین حداکثر دمای تراشه برای الگوریتم پیشنهادی و الگوریتم نیمه‌بهینه.

در تعیین مقاصد مهاجرت علاوه بر دمای هسته‌ها، وزن ارتباطی بین وظایف نیز لحاظ شده است، مشاهده می‌شود که برای برخی محک‌ها مقدار PDP بعد از مهاجرت نسبت به قبل از آن بهبود یافته و این بهبود برای محک MMS به دلیل نرخ ارتباطی بالاتر، بیشتر می‌باشد.

۳-۴ مقایسه با پژوهش‌های پیشین

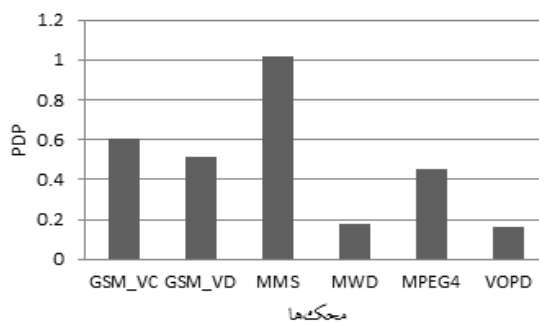
به جهت مقایسه با کارهای پیشین، مطالعه [۸] در نظر گرفته شده است که سعی در مدیریت دمایی شبکه‌های روی تراشه سه‌بعدی دارند.

در پژوهش [۸] سه روش نگاشت ایستای مبتنی بر ILP ارائه شده که از بین آنها، سومین الگوریتم انتخاب شده است و همچنان که بیان گردید یک الگوریتم نیمه‌بهینه است که علاوه بر این که قابلیت پیاده‌سازی بر روی شبکه‌های روی تراشه سه‌بعدی را داراست، کارایی بهتری نسبت به الگوریتم دوم دارد. این مقاله بدین جهت برای مقایسه برگزیده شده که اولاً بر بررسی کاهش دما تمرکز دارد که از این نظر با کار انجام‌شده همخوانی دارد. ثانیاً نویسندگان در [۸] بر سیستم‌هایی از نوع شبکه‌های روی تراشه سه‌بعدی و روش نگاشت وظایف تأکید دارند و طبق ارزیابی‌های انجام‌گرفته در مقاله، نسبت به روش‌های پیشین که در همین راستا ارائه شده‌اند بهتر عمل می‌کند. ضمن این که مدل دمایی، محک‌های مورد آزمایش و روش شبیه‌سازی مشابهی با آنچه ما در اینجا استفاده کرده‌ایم به کار گرفته شده است.

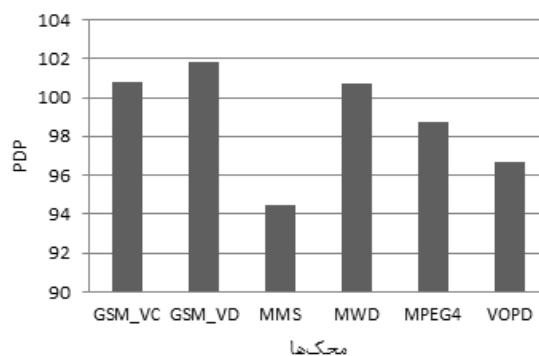
مقایسه از حیث حداکثر دمای بیشینه پس از اعمال الگوریتم اشاره‌شده و الگوریتم پیشنهادی در شکل ۸ آورده شده است. الگوریتم نیمه‌بهینه وظایف را از ابتدا با هدف کاهش دمای بیشینه به هسته‌ها نگاشت می‌کند، در حالی که در الگوریتم پیشنهادی، نگاشت اولیه یک نگاشت تصادفی می‌باشد و تنها وظایف موجود بر روی مبدأ و مقصد مهاجرت تغییر موقعیت خواهند داد. با این وجود مشاهده می‌شود که برای محک‌هایی با تعداد زیاد وظایف یا نرخ ارتباطی بالا، نگاشت ایستا عملکرد خوبی نخواهد داشت زیرا امکان تغییر نگاشت را در طول اجرا فراهم نمی‌آورد هر چند که برای محک‌هایی با تعداد وظایف و هسته‌های کم، بهتر از الگوریتم ارائه‌شده عمل نموده است.

۵- نتیجه‌گیری

در این پژوهش، چالش دمایی شبکه‌های روی تراشه سه‌بعدی را مخاطب قرار داده و از تکنیک مهاجرت وظایف مبتنی بر الگوریتم تبرید تدریجی جهت حل این مسأله بهره بردیم. هدف از این روش، کاهش دمای بیشینه تراشه بود. با این وجود سعی کردیم مسایل مرتبط به ترافیک شبکه و سربار مهاجرت را نیز در تابع هدف لحاظ کنیم. وجود ضرایب پارامتری به کاربر این امکان را می‌دهد که با توجه به نوع کاربرد، درصد



شکل ۶: PDP شبکه در زمان مهاجرت.



شکل ۷: PDP شبکه بعد از مهاجرت، نرمال شده نسبت به قبل از آن.

تعداد لایه‌های عمودی بیشتر می‌شود، دمای بیشینه و میانگین تراشه بیشتر شده و تأثیر مهاجرت مشهودتر خواهد بود. به منظور مشاهده بهتر میزان تأثیر مهاجرت، درصد کاهش دما بیشینه و میانگین آن در شکل ۴ آورده شده است. درصد کاهش برای محک‌های GSM_VC و MMS با سه لایه و محک GSM_VD با چهار لایه قابل توجه است و این مقدار به ازای محک GSM_VC بیش از ۲۸ درصد می‌باشد. علاوه بر کاهش دمای بیشینه و دمای میانگین تراشه، مهاجرت به تعادل دمایی هسته‌ها نیز کمک خواهد کرد به نحوی که اختلاف دمای هسته‌های پردازشی با دمای میانگین پس از مهاجرت، کاهش می‌یابد. این مطلب با محاسبه میزان انحراف معیار در شکل ۵ نشان داده شده است.

۴-۲ تأثیر مهاجرت بر کارایی سیستم

هدف اصلی در الگوریتم پیشنهادی کاهش دمای بیشینه تراشه است و با این وجود باید این روش از سربار قابل قبولی نیز برخوردار باشد. سربار مهاجرت، شامل تأخیر و توان مصرفی تحمیل‌شده به شبکه به منظور جابه‌جایی کد و داده به مکان جدید خواهد بود. به منظور مشاهده میزان تأثیر مهاجرت بر این پارامتر، حاصل ضرب توان در تأخیر (PDP) شبکه در زمان مهاجرت در شکل ۶ آورده شده است. نتایج به دست آمده نشان می‌دهد که میزان سربار مهاجرت در این نوع سیستم‌ها تا حد قابل قبولی خواهد بود. البته باید در نظر داشت که در روش پیشنهادی میزان تأثیر به مقدار پارامترهای دخیل در تابع هزینه نیز بستگی خواهد داشت. مقدار این پارامترها برای این آزمایش برای α ، β و γ به ترتیب برابر با ۰،۰۵، ۰،۱۲۵ و ۰،۳۷۵ لحاظ شده است. همان‌طور که انتظار می‌رفت، میزان این سربار برای کاربردی همانند MMS به دلیل حجم ارتباطی بالا در مقایسه با سایر کاربردها بیشتر است. این میزان سربار در مقایسه با این مزیت که این روش قابلیت اعمال در زمان اجرا را دارد می‌تواند قابل اغماض باشد. به منظور بررسی بیشتر میزان تأثیر مهاجرت بر کارایی سیستم، PDP شبکه قبل و بعد از مهاجرت محاسبه شده است (شکل ۷). از آنجایی که

Computers & Digital Techniques, vol. 7, no. 6, pp. 274-281, Nov. 2013.

- [13] H. Orsila, E. Salminen, and T. D. Hamalainen, "Best practices for simulated annealing in multiprocessor task distribution problems," in *Simulated Annealing*, Ch. 16, pp. 321-342, 2008.
- [14] <https://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/BookSim>, Accessed Mar. 2014.
- [15] A. Kahng, B. Li, L. Peh, and K. Samadi, "ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration," in *Proc. Design, Automation & Test in Europe Conf. & Exhibition, DATE'09*, pp. 423-428, Apr. 2009.
- [16] A. Sharifi and H. Sarbazi-Azad, "Power consumption and performance analysis of 3D NoCs," in *Proc. Asia-Pacific Computer Systems Architecture Conf.*, pp. 209-219, Aug. 2007.
- [17] S. Priyadarshi, et al., "Hetero 3D integration: a scheme for optimizing efficiency/cost of chip multiprocessors," in *Proc. 14th IEEE Int. Sym. on Quality Electronic Design, ISQED'13*, 7 pp., 4-6 Mar. 2013.

منیره محبی مقدم تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۸۷ و ۱۳۹۱ از دانشگاه شهید باهنر کرمان و دانشگاه صنعتی شریف به پایان رسانده است و هم‌اکنون دانشجوی مقطع دکتری دانشکده مهندسی کامپیوتر دانشگاه صنعتی اصفهان می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: محاسبات سبک، شبکه هوشمند برق، محاسبات موازی، محاسبات ابری، نظریه بازی و طراحی مکانیزم، امنیت شبکه، شبکه روی تراشه و مباحث مدیریت انرژی.

سیدحمید میرمحمدی تحصیلات خود را در مقاطع کارشناسی، کارشناسی ارشد و دکتری به ترتیب در سال‌های ۱۳۸۱، ۱۳۸۳ و ۱۳۸۸ از دانشگاه صنعتی شریف، دانشگاه تهران و دانشگاه صنعتی شریف در رشته مهندسی صنایع به پایان رسانده است و هم‌اکنون استادیار دانشکده مهندسی صنایع و سیستم‌ها در دانشگاه صنعتی اصفهان می‌باشد. زمینه‌های تحقیقاتی مورد علاقه نامبرده عبارتند از: تحقیق در عملیات، بهینه‌سازی ترکیبی در مسائل برنامه‌ریزی تولید و موجودی‌ها.

محمدحسین منشی مدرک کارشناسی و کارشناسی ارشد مهندسی برق را در سال‌های ۱۳۷۶ و ۱۳۷۹ از دانشگاه صنعتی اصفهان دریافت نمود. ایشان کارشناسی ارشد دوم خود را در رشته مهندسی کامپیوتر از دانشگاه نیس سوفا-آنتیپولیس فرانسه در سال ۱۳۸۱ دریافت کرد. پس از آن، موفق به دریافت بورس تحصیلی دولت فرانسه گردید و تحصیلات خود را در مقطع دکتری در موسسه تحقیقاتی علوم کامپیوتر و اتوماسیون (INRIA) در سال ۱۳۸۴ به پایان رسانید. ایشان پیش از پیوستن به دانشگاه صنعتی اصفهان به عنوان استادیار، در دانشگاه پلی تکنیک لوزان (EPFL) سوئیس، به عنوان محقق و مدرس ارشد مشغول به کار بوده‌اند. دکتر منشی از سال ۱۳۹۰ در دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی اصفهان مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشکده می‌باشد. زمینه‌های علمی مورد علاقه نامبرده شامل موضوعاتی مانند شبکه‌های کامپیوتری، امنیت و حریم خصوصی، نظریه بازی‌ها و طراحی مکانیزم می‌باشد.

اهمیت هر یک از این اهداف را تعیین نماید. نتایج نشان داد که این روش می‌تواند دمای بیشینه را حداکثر تا ۲۸٪ برای محکی با بیشترین تعداد وظایف و بیشترین تعداد لایه‌های عمودی کاهش دهد، ضمن این که پراکندگی دمایی هسته‌های پردازشی را نیز کاهش دهد. به عنوان کارهای آتی می‌توان تأثیر تغییر پارامترهای الگوریتم SA و همچنین ضرایب موجود در تابع هزینه را به ازای کاربردهای مختلف مورد بررسی قرار داد.

مراجع

- [1] C. Seiculescu, S. Murali, L. Benini, and G. Micheli, "3D network on chip topology synthesis: designing custom topologies for chip stacks," in *3D Integration for NoC-based SoC Architectures, Integrated Circuits and Systems*, Springer New York, pp. 193-223, 2011.
- [2] D. Jayasimha, B. Zafar, and Y. Hoskote, On-Chip Interconnection Networks: Why They Are Different and How to Compare Them, Technical Report, Intel Corp, 2006.
- [3] C. H. Chao, K. Y. Jheng, H. Y. Wang, J. C. Wu, and A. Y. Wu, "Traffic- and thermal-aware run-time thermal management scheme for 3D NoC systems," in *Proc. 4th ACM/IEEE Int. Symp. on Networks-on-Chip*, pp. 223-230, May 2010.
- [4] Y. R. Huang, J. H. Pan, and Y. C. Lu, "Thermal-aware router-sharing architecture for 3D network-on-chip designs," in *Proc. IEEE Asia Pacific Conf. on Circuits and Systems*, pp. 1087-1090, 6-9 Dec. 2010.
- [5] H. Wang, Y. Fu, T. Liu, and J. Wang, "Thermal management via task scheduling for 3D NoC based multi-processor," in *Proc. Int. SoC Design Conf., ISOC'10*, pp. 440-444, 22-23 Nov. 2010.
- [6] Y. Cui, W. Zhang, and H. Yu, "Distributed thermal-aware task scheduling for 3D network-on-chip," in *Proc. 30th IEEE Int. Conf. on Computer Design, ICCD'12*, pp. 494-495, 30 Sep.-3 Oct. 2012.
- [7] C. Addo-Quaye, "Thermal-aware mapping and placement for 3-D NoC designs," in *Proc. of the IEEE Int. Conf. on SOC*, pp. 25-28, Sep. 2005.
- [8] P. Hamedani, S. Hessabi, H. Sarbazi-Azad, and N. Enright Jerger, "Exploration of temperature constraints for thermal aware mapping of 3D networks on chip," *International Journal of Adaptive, Resilient and Autonomic Systems*, vol. 4, no. 3, pp. 42-60, Jul. 2012.
- [9] J. Donald and M. Martonosi, "Techniques for multicore thermal management: classification and new exploration," in *Proc. 33rd Int. Symp. on Computer Architecture, ISCA'06*, pp. 78-88, Jun. 2006.
- [10] F. Galea and R. Sirdey, "A parallel simulated annealing approach for the mapping of large process networks," in *Proc. of 26th IEEE Int. Parallel and Distributed Processing Symp. Workshops & PhD Forum, IPDPSW'12*, pp. 1787-1792, Shanghai, China, 21-25 May 2012.
- [11] Z. Lu, L. Xia, and A. Jantsch, "Cluster-based simulated annealing for mapping cores onto 2D mesh networks on chip," in *Proc. 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems, DDECS'08*, 6 pp., Apr. 2008.
- [12] N. Hassanpour, S. Hessabi, and P. K. Hamedani, "Temperature control in three-network on chips using task migration," *IET*