

## تخصیص پیوسته و مهاجرت تلفیقی پردازنده‌ها جهت بهبود کارایی در نگاشت برخط شبکه روی تراشه دوبعدی

مهناز رفیعی<sup>۱</sup>اکرم رضا<sup>۲</sup>

استادیار دانشگاه آزاد اسلامی واحد رامهرمز

استاد یار دانشگاه آزاد اسلامی واحد شهر قدس

(دریافت: ۱۴۰۰/۰۵/۲۲، پذیرش: ۱۴۰۰/۰۸/۱۷)

### چکیده

شبکه روی تراشه‌ها به‌عنوان راه‌حلی جهت بهبود ارتباط بین اجزا شبکه به کار می‌روند. از آنجایی که یکی از ابعاد طراحی در این شبکه‌ها مفهوم نگاشت می‌باشد، لذا در این مقاله مفاهیم و پارامترهای مختلف در نگاشت برخط برای کارهای متعدد در شبکه روی تراشه مورد بررسی قرار گرفته است. در این راستا سه‌گام ضروری در نگاشت برخط جهت کارهای مختلف در شبکه روی تراشه در نظر گرفته شده است که شامل یافتن اندازه مناسب زیر توری برای کار ورودی، یافتن جایگاه مناسب زیر توری در توری جهت تخصیص برخط کار و یافتن محل اصلی در توری است. در ضمن، مدل‌های مؤثر قبلی جهت انتخاب ابعاد زیر توری،  $MD$ ،  $MT\&MPN$ ، روش‌های تخصیص پردازنده قبلی،  $TCB$ ،  $TRB$  و روش‌های مهاجرت پردازنده پیشین مبتنی بر دو مرز ستونی، دو مرز سطری، فشردگی‌سازی چپ - راست محدودشده و فشردگی‌سازی بالا - پایین محدود شده برای توپولوژی توری در نظر گرفته شده است و با مهاجرت‌های تلفیقی پیشنهادی مقایسه شده است. الگوریتم‌های مزبور استفاده شده‌اند تا پیوستگی افزایش یابد و تأخیر در سیستم‌های چندپردازنده‌ای کاهش یابد. از آنجایی که هدف اصلی در توسعه روش‌های پیشنهادی دستیابی به بیشینه کارایی است، لذا در این روال، تأثیر پارامترهای کارایی مختلف در مقابل مکانیزم‌های قبلی مقایسه شده است. در این مقاله، شش الگوریتم که کارایی بهتری را کسب نموده‌اند با ۴۵ مورد دیگر مقایسه شده است. استفاده از روش‌های مهاجرت تلفیقی ما را قادر می‌سازد تا تعداد مهاجرت‌های پردازنده را محدود نماییم. افزون بر این، طرح  $MT\&MPN/TCB/HCM$  نتایج بهتری را در مقایسه با سایر الگوریتم‌ها با در نظر گرفتن متوسط زمان اجرای کار با  $38.0248\%$ ، متوسط زمان پاسخ کار با  $99.5387\%$  و متوسط بهره‌وری سیستم با  $48.0239\%$  کسب نموده است. نتایج شبیه‌سازی‌ها نشان می‌دهد که روش  $MPN/TRB/HBM$  بالاترین نتایج را در متوسط مصرف توان و متوسط زمان اجرای کار به ترتیب با  $6.68\%$  و  $38.21\%$  دارد.

کلیدواژه‌ها: تخصیص، تکه‌تکه‌شدن، چندپردازنده‌ای‌ها، زیر توری، مهاجرت، شبکه روی تراشه، مهاجرت، نگاشت

## Continuous allocation and hybrid migration of processors to improve the performance of online mapping on a 2-D network on chip

M.Rafie, A.Reza

Faculty member Islamic Azad University of Ramhormoz, Ramhormoz, Iran,

Faculty member Islamic Azad University of Shahr e Qods, Shar e Qods, Iran,

(Received: 2022/August/13; Accepted: 2022/November/8)

### Abstract

The network on chips was introduced as a solution to improve coherence between the network components. As one aspect of the network on chip design is mapping, so, in this paper, we have presented different concepts and parameters in the online mapping for different jobs in the network on chips. Thus, three essential steps are considered in the online mapping for different jobs in the network on chips which are finding the appropriate size of sub-mesh for input job, finding a sub-mesh place in integrating the mesh for online job allocation and finding the main place in sub-mesh. Also, previous efficient models to select the dimensions of the sub-mesh,  $MD$ ,  $MPN$  and  $MT\&MPN$ , previous processor allocation mechanisms,  $TRB$  and  $TCB$ , and the traditional processor migration methods based on two-column boundary, two-row boundary, limited left-right compaction, and limited top-down compaction for mesh topology are considered and compared against the proposed hybrid migrations. The mentioned algorithms are used to increase continuity and decrease latency in multiprocessor systems. Since the main goal in the development of the proposed methods is to achieve the maximum performance, so in this process, the impact of different performance parameters will be compared against the previous mechanisms. In this paper, 6 algorithms, which have achieved better performance, have been selected among the 45 ones. We have demonstrated that using hybrid migration strategies enable us to limit the number of processors migrations. Moreover,  $MT\&MPN/TCB/HCM$  has revealed better results among these algorithms with the average job execution time  $38.0248\%$ , average job response time  $99.5387\%$ , and average system utilization  $48.0239\%$ . Indeed, simulation results show that  $MPN/TRB/HBM$  has maximum results of the average power consumption  $6.68\%$ , and the best results of the average job execution time  $38.21\%$ .

**Keywords:** Allocation, Fragmentation, Migration, Multiprocessors, Network on Chip (NoC), Mapping, Sub-mesh.

## ۱. مقدمه

در صورتی که تمام وظایف یک برنامه بر روی هسته‌های مجاور هم نگاشت شود تخصیص پیوسته است، اما در صورتی که تمام هسته‌های تخصیص داده شده به وظایف یک برنامه در مجاورت هم نباشد تخصیص ناپیوسته است. در تخصیص پیوسته امکان تکه‌تکه شدن پردازنده‌ها در سطح تراشه وجود دارد و به دلیل پیوسته نبودن هسته‌های آزاد در سطح تراشه با وجود بیشتر بودن تعداد هسته‌های آزاد نسبت به تعداد وظایف یک برنامه امکان اجرای برنامه وجود ندارد و این امر منجر به افت بهره‌وری، افزایش زمان انتظار برنامه‌ها برای اجرا و افت کارایی می‌شود [۹-۴].

برای حل مشکل تکه‌تکه شدن هسته‌های آزاد در سطح تراشه از روش‌های مهاجرت وظایف در حال اجرا استفاده می‌شود به شرط اینکه تخصیص منابع به وظایف، غیرانحصاری (قابل پس گرفتن در زمان اجرا) باشد [۱۰]. این روش برای یکپارچه کردن هسته‌های آزاد سطح تراشه استفاده می‌شود.

در این مقاله به منظور حل مشکل تکه‌تکه شدن هسته‌های آزاد روی تراشه دو الگوریتم مهاجرت جهت نگاشت چند برنامه در شبکه روی تراشه پیشنهاد شده است. در بخش ۲، مطالعات انجام شده روی بهبود راندمان الگوریتم‌های انتخاب زیر توری، تخصیص و مهاجرت بررسی می‌شود و وضعیت کارایی الگوریتم‌های پیشین بیان شده‌اند. در بخش ۳، الگوریتم‌های مهاجرت تلفیقی پیشنهادی ارائه شده است. خروجی شبیه‌ساز و شبیه‌سازی در بخش ۴ تشریح شده‌اند. در نهایت، در بخش ۵، نتایج مطالعات گذشته از نقطه نظر چندین پارامتر کارایی مهم مقایسه شده است.

## ۲. بررسی نگاشت بر خط

با هدف ارائه یک الگوریتم نگاشت برخط چند برنامه بر روی شبکه روی تراشه گام‌های زیر اجرا می‌شود:

- ۱- محاسبه اندازه زیر توری بر اساس اندازه کار ورودی مانند روش‌های  $MD$  [۱۱، ۱۲]،  $MPN$  [۱۳]،  $MT\&MPN$  [۱۴].
- ۲- تخصیص زیر توری به کار و آزادسازی زیر توری بعد از اتمام کار مانند روش‌های تخصیص  $SBA$  [۱۵]،  $ISBA$  [۱۶]،  $TCB$  [۱۱] و  $TRB$  [۱۲].
- ۳- نگاشت کار بر روی فضای زیر توری مانند استفاده از نگاشت تصادفی.

با توسعه یک سیستم چندپردازنده روی تراشه ( $MPSoC$ )، شبکه روی تراشه ( $NoC$ ) که یک معماری گذرگاه چندهسته‌ای است، به طور گسترده مورد توجه و استفاده قرار گرفته است [۱، ۲]. با این حال، تأخیر و قدرت طراحی  $NoC$  به چالشی برای طراحان تبدیل شده است. در یک  $MPSoC$  مبتنی بر  $NoC$  تعداد زیادی انتقال داده وجود دارد. بنابراین، نتایج نگاشت تأثیر زیادی بر روی میزان مصرف انرژی، میانگین تأخیر و سایر پارامترهای کیفیت سرویس دارد. اهمیت این موضوع در این است که در تکنولوژی نانو کاهش میزان مصرف انرژی مهم‌ترین اولویت طراحی است و یک نگاشت مناسب می‌تواند ما را در رسیدن به این هدف یاری نماید. تاکنون الگوریتم‌های متعددی برای این منظور ارائه شده است. مشکل اصلی این الگوریتم‌ها زمان اجرای آنها است. زیرا در صورت وجود  $n$  کاشی<sup>۱</sup> در شبکه، تعداد حالات مختلفی که برای نگاشت مطرح می‌شود،  $n!$  خواهد بود. بنابراین ارزیابی همه حالات غیرممکن است. معمولاً این مسئله توسط روش‌های ابتکاری قابل حل می‌باشد [۳، ۴]. الگوریتم‌های نگاشت می‌توانند قبل از شروع به کار برنامه انجام شوند (نگاشت برون خط) که در این صورت هدف یافتن بهترین نگاشت بر اساس هدف تعیین شده است و سربرار زمانی اجرای الگوریتم تأثیری بر روی زمان اجرای برنامه ندارد، هدف یافتن نگاشت بهینه است. دسته دیگر از الگوریتم‌های نگاشت در زمان اجرای برنامه‌ها بر روی تراشه صورت می‌گیرد (نگاشت بر خط) که زمان اجرای الگوریتم نگاشت تأثیر مستقیم بر روی زمان اجرای برنامه دارد. عموماً در الگوریتم‌های نگاشت بر خط علاوه بر تلاش به رسیدن به هدف تعیین شده، به دنبال کمینه نمودن سربرار زمان اجرای الگوریتم هستیم.

در نگاشت بر اساس تعداد هسته‌های پردازشی و تعداد وظایف برنامه‌ها دو حالت نگاشت محدود و نگاشت نامحدود ایجاد می‌شود. در نگاشت محدود، زمانی که تعداد هسته‌های پردازشی کمتر از تعداد وظایف باشد، بیش از یک وظیفه روی یک هسته نگاشت می‌شود و مسئله زمان بندی در سطح هسته هم مطرح می‌شود. در نگاشت نامحدود تعداد هسته‌های پردازشی بزرگ‌تر و یا مساوی با تعداد وظایف است.

الگوریتم‌های نگاشت عموماً برای نگاشت یک برنامه کاربردی با چندوظیفه پیشنهاد شده‌اند، اما اخیراً الگوریتم‌های نگاشت جهت نگاشت هم‌زمان چندین برنامه کاربردی با تعداد بالای وظایف روی یک تراشه مطرح شده‌اند. در نگاشت چندین برنامه کاربردی دو حالت تخصیص پیوسته و ناپیوسته وجود دارد،

<sup>۱</sup> Tile

توری‌ها به صورت هم‌زمان در ناحیه توری جستجو می‌شود، بنابراین زمان انتظار کاهش می‌یابد [۱۴].

## ۲-۲. تخصیص پیوسته

در گام تخصیص به منظور کاهش تأخیر ارتباطی وظایف از تخصیص پیوسته برای اجرای هر برنامه استفاده شده است تا توری از هسته‌های آزاد با ابعاد تعیین شده در گام قبلی بر روی NoC تعیین شود [۱۱]. همان‌طور که ذکر شد، روش‌های تخصیص پردازنده به دو دسته پیوسته و ناپیوسته تقسیم می‌شود.

### ۲-۲-۱. الگوریتم تخصیص مبتنی بر پشته (SBA):<sup>۴</sup>

در این روش یک پشته از هسته‌های آزاد به صورت مستطیل نگهداری می‌شود، هر زیر توری شامل مختصات گره پایه (گره پایین سمت چپ) و مختصات گره پایانی (گره بالا سمت راست) است. برای تخصیص زیر توری آزاد بر اساس ابعاد تعیین شده یک جستجو در این پشته برای ابعاد بزرگ‌تر یا مساوی انجام می‌شود [۱۵].

### ۲-۲-۲. الگوریتم مبتنی بر پشته بهبود یافته (ISBA)<sup>۵</sup>

الگوریتم تخصیص مبتنی بر پشته ISBA، بهبود یافته الگوریتم تخصیص مبتنی بر پشته SBA است. برای یافتن هسته‌های پردازشی آزاد پیوسته با ابعاد زیر توری، توری تعیین شده در گام قبل استفاده می‌شود. در این الگوریتم علاوه بر ابعاد زیر توری بهینه (p,q) سایر ابعاد کاندید مانند (q,p) و زیر توری با ابعاد دیگر که با اندازه برنامه مطابقت دارد به ترتیب در پشته زیر توری‌های آزاد مورد جستجو قرار می‌گیرند. ترتیب جستجو در پشته بر اساس ابعاد بهینه است [۱۶].

### ۲-۲-۳. الگوریتم تخصیص دو مرز ستونی (TCB):<sup>۶</sup>

در این روش بر اساس ابعاد زیر توری تعیین شده و ابعاد زیر توری‌های اشغال، مختصات گره پایه از زیر توری کاندید بر اساس روابط (۱) و (۲) محاسبه می‌شود. سپس در صورتی که بیش از یک گره پایه کاندید داشته باشیم، گره پایه‌ای انتخاب می‌شود که کمینه فاصله را از ستون‌های مرزی (سمت چپ ترین ستون و راست ترین ستون) داشته باشد.

$$\text{Endnode. } x_i = \text{basenode. } x_i + p$$

$$\text{Endnode. } y_i = \text{basenode. } y_i + q \quad (1)$$

$$b.x_i = \text{basenode. } x_i - 0$$

$$e.x_i = m - \text{basenode. } x_i$$

۴- مهاجرت (جابجایی) زیر توری‌ها در زمان رخ دادن تکه-تکه شدن خارجی<sup>۱</sup> در تخصیص مانند روش‌های مهاجرت تلفیقی و روش‌های ODC-SC [۱۰]، ODC-FC [۱۰]، TCB [۱۱]، TRBMA [۱۴]، LTDC [۱۲]، LLRC [۱۳]. جهت کاهش زمان سربار نگاشت برخط، مراحل دوم و سوم می‌بایست به طور هم‌زمان اجرا شوند [۱۱، ۱۲]. این مفاهیم در این بخش توضیح داده شده‌اند.

## ۱-۲. محاسبه اندازه زیر توری مناسب برای کار ورودی

### ۱-۱-۲. الگوریتم MD

در این روش هدف کاهش میزان اتلاف در زیر توری است که در این راستا قطر زیر توری می‌بایست کمینه باشد [۱۱]. برای تعیین اندازه زیر توری (اندازه ابعاد هسته‌های مورد نیاز) مناسب برای برنامه ورودی، زیر توری مستطیل در نظر گرفته شده است که طول و عرض مستطیل تعداد هسته‌ها در هر بعد است و حاصل ضرب طول×عرض بزرگ‌تر مساوی تعداد وظایف برنامه است. در بین ابعاد تعیین شده برای مستطیل‌ها، مستطیل با کمترین قطر انتخاب می‌شود [۱۲].

### ۲-۱-۲. الگوریتم انتخاب زیر توری با استفاده از

#### کوچکترین عدد اول (MPN)<sup>۲</sup>

برای افزایش پیوستگی در چند پردازنده‌ای‌ها روش انتخاب زیر توری با استفاده از کوچک‌ترین عدد اول مطرح شده است. هدف این روش به دست آوردن کوچک‌ترین عدد اولی است که قابل تقسیم بر تعداد هسته‌هاست. این عدد می‌تواند به عنوان تعداد سطرها در زیر توری مطرح شود. برای مثال اگر تعداد هسته‌ها برابر ۱۶ باشد زیر توری دارای ۲ سطر و ۸ ستون می‌باشد [۱۳].

### ۳-۱-۲. الگوریتم ترکیب حداقل آستانه و متدولوژی

#### کوچکترین عدد اول (MT&MPN)<sup>۳</sup>

در این روش هدف اصلی کمینه نمودن تأخیر و افزایش پیوستگی است. همچنین اصل بر حداقل آستانه و کوچک‌ترین عدد اول می‌باشد. در این روش آستانه عدد ۹ در نظر گرفته شده است. مقدار آستانه مبتنی بر اندازه توری می‌باشد؛ لذا می‌بایست بیش از نصف ابعاد توری باشد. با فرض اینکه مقدار آستانه را برابر ۹ در نظر بگیریم، ابعاد زیر توری از (۱،۹) یا (۹،۱) برای کار ورودی از تعداد ۹ هسته در نظر گرفته می‌شود. چنانچه اندازه کار ورودی بیش از ۹ هسته باشد به عنوان مثال برای ۱۲ هسته نتایج برابر با (۲،۶)، (۳،۴)، (۴،۳) و (۶،۲) می‌باشد. این ابعاد از زیر

<sup>۴</sup> stack-based allocation

<sup>۵</sup> Improved Stack Based Algorithm

<sup>۶</sup> Two Column Boundary

<sup>۱</sup> External fragmentation

<sup>۲</sup> Minimum prime number

<sup>۳</sup> Minimum threshold and prime number

### ۳-۲. بررسی الگوریتم‌های مهاجرت گراف وظیفه

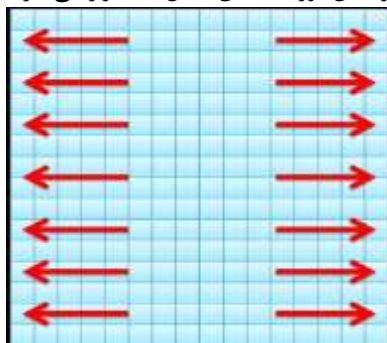
#### ۳-۲-۱. الگوریتم‌های مهاجرت استاندارد

روش‌های مهاجرت وظیفه شامل طرح‌هایی همچون تراکم پویای برخط: انتقال به سمت نزدیک‌ترین گوشه توری (ODC-FC) [۱۰]، الگوریتم مهاجرت دو مرز ستونی (TCB) [۱۱]، الگوریتم فشرده‌سازی بالا - پایین محدود شده [۱۲]، الگوریتم فشرده‌سازی چپ-راست محدود شده [۱۳] و الگوریتم مهاجرت دو مرز سطری (TRBMA) می‌باشند [۱۴] که روش‌های استاندارد نامیده می‌شوند. در ادامه برخی از الگوریتم‌های مزبور که نتایج بهتری را کسب نموده اند مطرح می‌گردند.

#### ۳-۲-۱-۱. تراکم پویای برخط: انتقال به سمت نزدیک‌ترین گوشه توری (ODC-FC)

در این روش برای مهاجرت و یکپارچه کردن فضای هسته‌های آزاد، زیر توری‌های اشغال به طرف یکی از چهار گوشه در توری مهاجرت داده می‌شود، تا گره‌های آزاد به صورت پیوسته در مرکز توری قرار گیرند. در این الگوریتم، وظایف به سمت چهار گوشه از توری به جای تنها به سمت یک گوشه از توری مهاجرت داده می‌شوند. در این راستا، توری به چهار بخش پایین سمت چپ، پایین سمت راست، بالا سمت راست و در نهایت بخش بالا سمت چپ تقسیم می‌گردد. هر زیر توری تخصیص یافته در یکی از چهار بخش مذکور قرار می‌گیرد [۱۰].

۳-۲-۱-۲. الگوریتم مهاجرت دو مرز ستونی (TCB):  
در این استراتژی زیر توری به سمت نزدیک‌ترین مرز چپ و راست توری جابه‌جا می‌شود. در این روش در ابتدا نزدیک‌ترین مرز چپ و راست توری برای تمامی زیر توری‌ها محاسبه می‌شود، سپس حداقل فاصله توسط برداری از زیر توری محصور می‌شود. بعد از آن زیر توری با کمترین فاصله و دارای مقدار غیرصفر از اتصالات آزاد و گره‌های پایه یا گره‌های پایانی زیر توری نهایی برای مهاجرت را انتخاب می‌کنند. در نهایت زیر توری باید به سمت نزدیک‌ترین بردار ستونی توری جابه‌جا شود. چنان‌که گره‌های آزاد در میان توری مطابق شکل (۳) قرار می‌گیرند [۱۱].



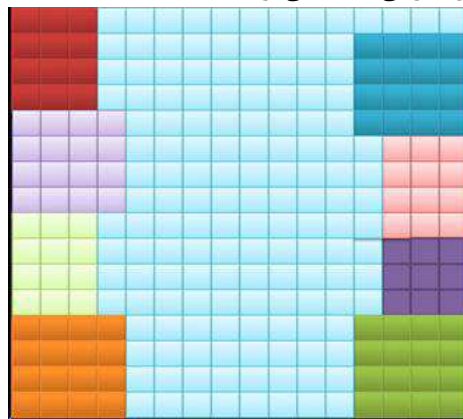
شکل (۳). یک مثال از الگوریتم مهاجرت TCB [۱۱]

$$\min.d_i = \min(b.x_i, e.x_i)$$

$$m.d = \min(\text{all } \min.d_i)$$

(۲)

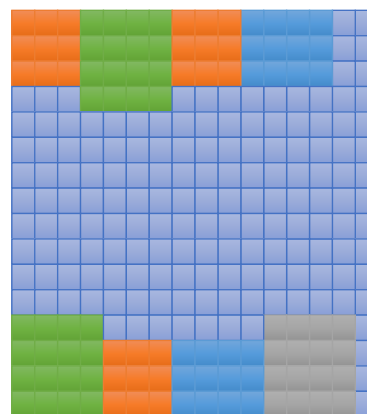
همان‌طوری که در شکل (۱) نشان داده شده است در این روش گره‌های آزاد به مرکز توری هدایت می‌شوند لذا مشکل تکه-تکه شدن خارجی کمینه می‌گردد [۱۱].



شکل (۱). الگوریتم تخصیص TCB [۱۱]

#### ۳-۲-۱-۴. الگوریتم تخصیص دو مرز سطری (TRB)

در این الگوریتم نیز همانند الگوریتم TCB بر اساس ابعاد زیر توری تعیین شده و لیست زیر توری‌های اشغال بر اساس روابط (۱) و (۲) به دنبال یافتن گره پایه (زیر توری) کاندید است. در صورتی که تعداد گره‌های پایه کاندید بیش از یک گره باشد گره-ای انتخاب می‌شود که فاصله کمینه از گره‌های مرزی در سطر اول (پایین‌ترین سطر) و سطر آخر (بالا‌ترین سطر) را داشته باشد. در این حالت تخصیص، تلاش بر این است که گره‌های آزاد به طور یکپارچه در سطرهای میانی قرار گیرند و احتمال تکه-تکه شدن خارجی کمینه شود که در شکل (۲) نشان داده شده است [۱۲].



شکل (۲). الگوریتم تخصیص TRB [۱۲]

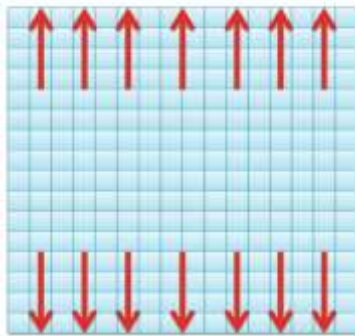
<sup>2</sup> Online Dynamic Compaction Four Corner

<sup>3</sup> Two column boundary

<sup>1</sup> Two Row Boundary

۲-۳-۱-۵. الگوریتم مهاجرت دو مرز سطری  
(<sup>۲</sup>TRBMA)

در این استراتژی هدف هدایت زیر توری به سمت نزدیک-ترین مرز بالا و پایین توری است. لذا در ابتدا نزدیک‌ترین مرز بالا و پایین توری برای تمامی زیر توری‌ها محاسبه می‌شود، تا اینکه گره‌های آزاد در وسط توری متمرکز شوند. سپس حداقل فاصله توسط برداری از زیر توری محصور می‌شود. بعد از آن زیر توری با کمترین فاصله و دارای مقدار غیرصفر از اتصالات آزاد و گره‌های پایه یا گره‌های پایانی زیر توری نهایی برای مهاجرت را انتخاب می‌کنند. در نهایت زیر توری باید به سمت نزدیک‌ترین بردار توری جابه‌جا شود [۱۴]. مثالی از این روش در شکل (۶) آمده است:



شکل (۶). یک مثال از TRBMA [۱۴]

۲-۳-۲. الگوریتم‌های مهاجرت ترکیبی

تخصیص هسته‌ها به وظایف به دو صورت انحصاری و غیرانحصاری می‌تواند صورت گیرد، در روش تخصیص انحصاری بعد از تخصیص امکان مهاجرت برنامه در حال اجرا وجود ندارد، اما برای تخصیص غیرانحصاری امکان استفاده از روش مهاجرت برای حل مشکل تکه‌تکه شدن خارجی وجود دارد. استراتژی‌هایی همچون TCB&ODC-FC [۱۱]، LTDC&TRBMA [۱۴]، مزایای LTDC&ODC-FC [۱۴] و TRBMA&ODC-FC [۱۴] هر دو الگوریتم مهاجرت را دارا می‌باشند. در هر تکرار یکی از الگوریتم‌ها اجرا می‌شود. به‌عنوان مثال، چنانچه تعداد مهاجرت‌های روش TRBMA&ODC-FC مساوی با بیست باشد، الگوریتم‌های مهاجرت TRBMA [۱۴] و ODC-FC [۱۰] هر کدام ده بار اجرا خواهند شد. در این رویه، یک متغیر جهت کنترل اجرای هر یک از دو الگوریتم مهاجرت در نظر گرفته شده است. در ضمن، ترکیب الگوریتم‌های دیگر نیز رویه مشابهی را بردارند.

۲-۳-۱-۳. الگوریتم فشرده‌سازی بالا-پایین محدود شده<sup>۱</sup> (LTDC)

در این الگوریتم برای پیوسته کردن هسته‌های آزاد با پیچیدگی کمتر، زیر توری‌های اشغال به سمت لبه‌های مرزی در بالاترین سطر و پایین‌ترین سطر مهاجرت داده می‌شوند، اولویت با سطر مرزی بالا است. فاصله گره پایه تا مختصات y کمینه توری و گره پایانی تا y بیشینه توری محاسبه می‌شود و کمینه فاصله در توری‌های کاندید انتخاب می‌شود. شبه کد الگوریتم LTDC در شکل (۴) آمده است [۱۲].

الگوریتم ۱: روش مهاجرت LTDC

**Input:** mesh topology & busy list

**Output:** migrate selected sub-mesh

1. Target-sub-mesh = (minimum distance with top or down side of the selected sub-mesh!=0)

//Find allocated sub-mesh in the top or down side

2. Stop job running in selected sub-mesh

3. Call de-allocation mechanism

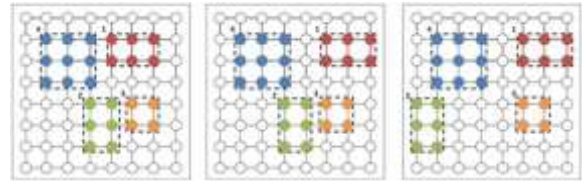
//migrate selected sub-mesh

4. Call allocation algorithm (job<sub>i</sub>)

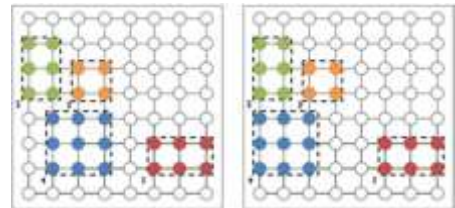
شکل (۴). شبه کد الگوریتم مهاجرت LTDC [۱۲]

۲-۳-۱-۴. الگوریتم فشرده‌سازی چپ-راست محدود شده<sup>۲</sup> (LLRC)

در این روش کارها به سمت چپ و راست توری مهاجرت می‌نمایند. اگر گره آزاد در سمت چپ کارها بود، زیر توری به سمت چپ توری مهاجرت می‌کند، در غیر این صورت به سمت راست توری مهاجرت می‌نماید. در واقع ابتدا سمت چپ زیر توری بررسی می‌شود. بنابراین فضای پیوسته وسیعی از گره‌های آزاد در توری ایجاد می‌شود که در شکل (۵) نیز قابل مشاهده است [۱۳].



c- مهاجرت وظیفه ۲ b- مهاجرت وظیفه ۱ a- قبل از مهاجرت وظایف



e- مهاجرت وظیفه ۴ d- مهاجرت وظیفه ۳

شکل (۵). استراتژی مهاجرت، بعد از تخصیص و آزادسازی در ساختار

یک توری  $M(8,8)$

<sup>3</sup> Two-row boundary migration algorithm

<sup>1</sup> Limited Top Down Compaction

<sup>2</sup> Limited Left Right Compaction



## ۲-۴. نگاشت در شبکه روی تراشه

نگاشت نحوه تخصیص هسته‌ها به هریک از وظایف یک برنامه است و یکی از گام‌های مؤثر طراحی در معماری شبکه روی تراشه است. الگوریتم نگاشت بر اساس زمان اجرا و خروجی تأثیر بسزایی بر روی پارامترهای کارایی مانند تأخیر ارتباطی، گذردهی و... و همچنین پارامترهای هزینه مانند توان مصرفی، دما و بهره‌وری منابع دارد. با توجه به این‌که یکی از چالش‌های اساسی طراحی تراشه‌ها میزان توان مصرفی و دمای تراشه است، الگوریتم‌های نگاشت زیادی برای کاهش دما و توان مصرفی تراشه ارائه شده‌اند [۳]. با توجه به زمان اجرای الگوریتم نگاشت، الگوریتم‌های نگاشت به دو دسته پویا (برخط) و ایستا (برون‌خط) تقسیم می‌شوند مهم‌ترین چالش در نگاشت ایستا بهینه بودن پاسخ است درحالی‌که در نگاشت پویا علاوه بر چالش فوق، زمان اجرای الگوریتم نگاشت یکی دیگر از چالش‌هاست، زیرا زمان اجرای الگوریتم به‌عنوان سربار زمان اجرای برنامه در نظر گرفته می‌شود، و منجر به افزایش تأخیر، توان مصرفی و افت کارایی می‌شود [۴].

در [۱۷] یک نگاشت برنامه کاربردی بلادرنگ بر روی NoC با استفاده از الگوریتم نگاشت دقیق مبتنی بر شاخه و حد (BB) توضیح می‌دهد. این رویکرد مصرف انرژی و تأخیر را کاهش می‌دهد. با این حال، توان عملیاتی را افزایش می‌دهد. این الگوریتم ادعا می‌کند که تا ۱۹،۹۳ درصد کاهش در مصرف انرژی و ۶۱،۱۰ درصد کاهش در تأخیر شبکه برای توپولوژی‌های توری و توریس دارد. در [۱۸] روشی را برای نگاشت چندین برنامه کاربردی در NoC توضیح می‌دهد که با انجام تجزیه و تحلیل در مناطق مختلف NoC شروع می‌شود. با استفاده از یک الگوریتم ژنتیک، وظایف برنامه بر روی مناطق بالقوه نگاشت شده و عملکرد آنها تجزیه و تحلیل می‌شود. در مرحله دوم، یک الگوریتم بازپخت شبیه‌سازی شده مبتنی بر الگوریتم درخت تصمیم برای استنباط نگاشت به روز شده وظایف استفاده می‌شود. با استفاده از نتایج تجربی، نویسندگان ادعا می‌کنند که ۲۳،۴۵٪ کاهش در انرژی مصرفی و ۲۴،۴۲٪ کاهش در تأخیر برای برنامه‌های کاربردی انتخاب شده دارند. در [۱۹] یک مرحله پیش محاسباتی را برای نگاشت جزئی برنامه و نگاشت نهایی در زمان اجرا برای برنامه‌های کاربردی سخت بلادرنگ پیشنهاد می‌کند. بنابراین، برنامه‌ها این امکان را دارند که نگاشت خود را بر اساس منابع پردازنده تطبیق دهند. نویسندگان ادعا می‌کنند که ۱۳ درصد کاهش در مصرف انرژی در مقایسه با رویکردهای پیشرفته دارند.

در [۲۰] مقایسه‌ای از نگاشت برنامه کاربردی NoC برای بهینه‌سازی ارتباطات، مصرف انرژی و تأخیر برای برنامه‌های کاربردی VOPD و MPEG4 را توصیف می‌کند. شایان ذکر است که تمرکز اصلی بر روی کاوش نگاشت اِپلیکیشن برای فناوری‌های طراحی جدید است. یک رویکرد جالب در [۲۱] مورد

بحث قرار گرفته است که نویسندگان یک الگوریتم نگاشت برای کاهش تأخیر و مصرف انرژی در معماری سخت افزار را مورد بحث قرار می‌دهند. برنامه در نظر گرفته شده، شبکه‌های عصبی اسپکینگ (SNN) است که باید روی سخت افزار نورومورفیک چند هسته‌ای نگاشت شوند. با این حال، این تکنیک نیاز به بهبود دارد و نمونه‌ای از نگاشت مستقیم برنامه است. در [۲۲] بر کاوش فضای طراحی برای شبکه‌های عصبی عمیق (DNN) با تمرکز بر فضای طراحی سلسله مراتب حافظه تمرکز دارد. آنها ادعا می‌کنند که ۶۴٪ عملکرد بهتری نسبت به راه‌حل‌های پیشرفته دارند، اما این کار را می‌توان با تمرکز بر جنبه ارتباطی بیشتر بهبود بخشید. در [۲۳] یک تکنیک دو مرحله‌ای را پیشنهاد می‌کند که بر اساس الگوریتم ژنتیک برای نگاشت برنامه‌های کاربردی بر روی معماری NoC است. روش پیشنهادی نسبت به الگوریتم ژنتیک سنتی از نظر تأخیر و مصرف توان نتیجه بهتری دارد.

## ۳. روش پیشنهادی

این مقاله مقایسه‌ای روی کارهای انجام شده پیشین در محدوده نگاشت برخط می‌پردازد که در شکل (۷) روش‌های پیشین و الگوریتم‌های مهاجرت پیشنهادی مطرح شده‌اند. الگوریتم‌های مهاجرت HBM و HCM روش‌های جدیدی هستند که در این بخش به شرح آنها می‌پردازیم. در ضمن، جهت نشان دادن برتری راهکار پیشنهادی آن را با روش‌های مهاجرت پیشین در نگاشت برخط مورد ارزیابی قرار داده‌ایم.



شکل (۷). روش‌های پیشنهادی و پیشین در فرایند تخصیص غیر-انحصاری

### ۳-۱-۱. الگوریتم مهاجرت ترکیبی گوشه‌ای (HCM)

الگوریتم مهاجرت HCM از مزایای هر دو روش LLRC و LTDC استفاده می‌نماید. در این روش هنگامی که تکه‌تکه شدن خارجی اتفاق می‌افتد، در گام اول، الگوریتم LLRC فراخوانی می‌شود. سپس چنانچه باز هم تکه‌تکه شدن خارجی اتفاق بیفتد، الگوریتم مهاجرت LTDC فراخوانی می‌شود. کنترل اجرای هر الگوریتم توسط متغیر پرچم انجام می‌شود. شبه کد الگوریتم ترکیبی مزبور در شکل (۸) نمایش داده شده است. شایان ذکر است که مسئله تکه‌تکه شدن پس از تعدادی تخصیص و از حالت تخصیص خارج نمودن اتفاق می‌افتد. بنابراین به شکل (۹)-الف می‌پردازیم و فرض می‌کنیم که تکه‌تکه شدن رخ داده است. همان‌طوری که ملاحظه می‌شود در این شکل تعداد هفت زیر توری در نظر گرفته شده است. در شکل (۹)-ب مشاهده می‌شود که تکه‌تکه شدن خارجی در مرحله اول با استفاده از الگوریتم

HBM در شکل (۱۰) نمایش داده شده است. همان‌طور که در شکل (۱۱) -ب نمایش داده شده است، تکه‌تکه شدن خارجی با استفاده از مهاجرت زیر توری در سطریهای مرزی از بین رفته است و گره‌های آزاد را به مرکز توری هدایت می‌کند. حل این مشکل هنگام استفاده از الگوریتم‌های TCB و TRBMA امکان‌پذیر تر می‌شود که در شکل (۱۱) قابل مشاهده است. این الگوریتم زیر توری‌ها را به چهار مرز توری هدایت می‌کند، بنابراین گره‌های آزاد در مرکز توری متمرکز می‌شوند.

هدف از بیان الگوهای فوق کاهش تعداد مهاجرت زیر توری‌ها در مقایسه با الگوریتم ODC-FC می‌باشد. این امر با استفاده از یک‌روند دومرحله‌ای ناپیوسته انجام می‌شود. علاوه بر آن تکه‌تکه شدن خارجی به اندازه الگوریتم ODC-FC کاهش می‌یابد. الگوریتم مهاجرت پیشنهادی، سطر و ستون زیر توری را شبیه الگوریتم ODC-FC جابه‌جا می‌کند. سربار زمانی و تکه‌تکه شدن خارجی با کنترل تعداد مهاجرت‌های زیر توری کاهش پیدا می‌کند. با توجه به شکل ۱۱-ب، تعداد مهاجرت زیر توری‌ها با استفاده از TRBMA و الگوریتم‌های ترکیبی در بدترین حالت به ترتیب ۶ و ۹ می‌باشند. ابعاد بزرگ‌ترین زیر توری آزاد در الگوریتم‌های TRBMA و HBM به ترتیب ۷×۱۰ و ۱۱×۹ می‌باشد.

**الگوریتم ۳: روش HBM**

- 1: *If Flag = 0 then*
- 2: *Call TRB migration algorithm.*
- 3: *Flag ← 1*
- 4: *Else*
- 5: *Call TCB migration method.*
- 6: *Flag ← 0*
- 7: *End if*

شکل (۱۰). فرآیند مهاجرت کار با استفاده از روش HBM



الف- قبل از مهاجرت  
ب- مهاجرت کار از طریق روش TRBMA  
ج- مهاجرت کار از طریق روش HBM

شکل (۱۱). شبه کد الگوریتم HBM

**۳-۳. بررسی الگوریتم‌های مهاجرت کار ترکیبی و استاندارد با ذکر مثال**

ارزیابی الگوریتم‌های مهاجرت به شکل ترکیبی و غیر ترکیبی در شکل‌های (۱۲) و (۱۳) شرح داده شده است. مقایسه

LLRC تا حدودی برطرف می‌شود. همان‌طور که در شکل (۹) -ج نیز شرح داده شده است این مشکل با استفاده از هر دو الگوریتم LLRC و LTCD برطرف می‌شود. به‌طور کلی این الگوریتم، زیر توری‌ها را به سمت گوشه چپ بالای توری و گره‌های آزاد را به سمت راست توری هدایت می‌نماید. هدف اصلی این الگوریتم کاهش تعداد مهاجرت زیر توری‌ها در مقایسه با الگوریتم ODC-FC می‌باشد که از طریق استفاده از مهاجرت دومرحله‌ای به دست می‌آید. در ضمن، در این روش تعداد محاسبات در مقایسه با روش‌های TRBMA و TCB کاهش می‌یابد. همچنین، تکه‌تکه شدن خارجی در مقایسه با الگوریتم‌های LLRC و LTCD به شدت افت می‌کند. با توجه به شکل ۸-ب، تعداد مهاجرت در شیوه LLRC چهار است. در بدترین شرایط هنگامی که هر دو شیوه LLRC و LTCD لازم می‌باشد می‌بایست بر روی زیر توری‌ها ۱۱ مهاجرت انجام شود. در واقع اولین زیر توری بزرگ به ابعاد ۱۶ و ۹ می‌باشد و دومین زیر توری بزرگ به ابعاد ۶ و ۷ می‌باشد. با مقایسه شکل (۹) -ب و (۹) -ج می‌توان پیوستگی گره‌های آزاد را با استفاده از الگوریتم‌های ترکیبی فوق به شدت افزایش داد.

**الگوریتم ۲: روش HCM**

- 1: *If Flag = 0 then*
- 2: *Call LLRC method.*
- 3: *Flag ← 1*
- 4: *Else*
- 5: *Call LTDC method.*
- 6: *Flag ← 0*
- 7: *End if*

شکل (۸). شبه کد الگوریتم HCM

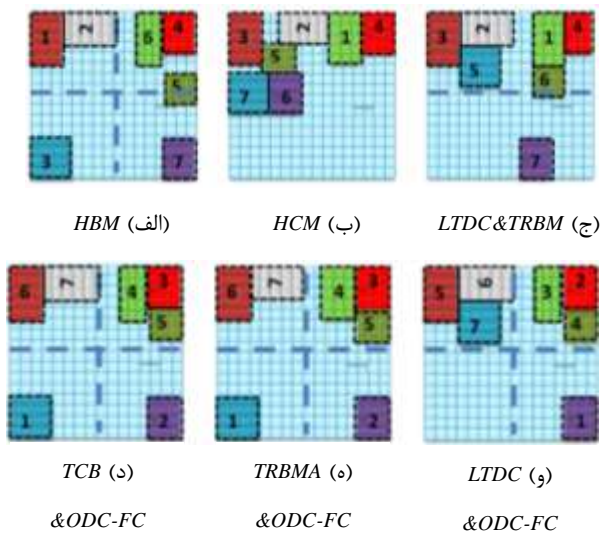


الف- قبل از مهاجرت  
ب- مهاجرت کار با استفاده از روش LLRC  
ج- مهاجرت کار با استفاده از روش HCM

**۳-۲. الگوریتم مهاجرت مرزی ترکیبی (HBM)**

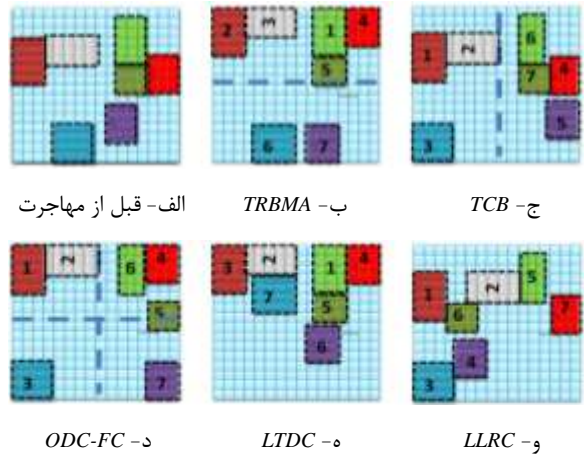
الگوریتم مهاجرت مرزی ترکیبی از مزیت‌های هر دو الگوریتم مهاجرت TRBMA و TCB برخوردار می‌باشد. در مرحله اول هنگام وقوع تکه‌تکه شدن خارجی الگوریتم TRBMA فراخوانی می‌شود. در مرحله دوم پس از وقوع تکه‌تکه شدن خارجی الگوریتم مهاجرت TCB فراخوانی می‌شود. متغیر پرچم تعیین می‌کند که کدام یک از دو الگوریتم اجرا شوند. شبه کد الگوریتم

شکل (۱۲). ارزیابی الگوریتم‌های مهاجرت به صورت انفرادی



شکل (۱۳). بررسی الگوریتم‌های مهاجرت کار ترکیبی

مختصری میان الگوریتم‌های مهاجرت با توجه به مثال ۱۲-الف در جدول (۱) انجام شده است باید خاطر نشان ساخت که  $N$  تعداد مهاجرت‌های انجام شده با اولین الگوریتم و  $n'$  تعداد مهاجرت‌های انجام شده با دومین الگوریتم می‌باشد.



جدول (۱). ارزیابی الگوریتم‌های مهاجرت کار

الگوریتم‌های مهاجرت کار	ابعاد بزرگ‌ترین زیر توری آزاد با در نظر گرفتن شکل (۱۲)-الف	ابعاد دومین بزرگ‌ترین زیر توری آزاد با در نظر گرفتن شکل (۱۲)-الف	تعداد مهاجرت‌های زیر توری در بهترین حالت	تعداد مهاجرت‌های زیر توری در بدترین حالت
TRBMA	7x10	3x12	$O(N)$	$O(N)=6$
TCB	7x9	3x10	$O(N)$	$O(N)=2$
LLRC	7x10	3x10	$O(N)$	$O(N)=4$
LTDC	9x9	3x12	$O(N)$	$O(N)=7$
ODC-FC	9x11	4x7	$O(N)$	$O(N)=9$
LTDC&TRBMA	9x9	3x12	$O(N)$	$O(N+n')=8$
TRBMA&ODC-FC	9x11	4x7	$O(N)$	$O(N+n')=10$
LTDC&ODC-FC	9x13	3x5	$O(N)$	$O(N+n')=12$
TCB & ODC-FC	9x11	4x7	$O(N)$	$O(N+n')=9$
HCM	9x11	6x7	$O(N)$	$O(N+n')=11$
HBM	9x11	4x7	$O(N)$	$O(N+n')=9$

الگوریتم‌هایی که به صورت تکی به کار برده شده‌اند دارای بیش-ترین تعداد مهاجرت زیر توری است. در ضمن، روش مزبور به بزرگ‌ترین زیر توری آزاد با ابعاد 9x11 دست می‌یابد. روش‌های LTDC و ODC-FC در میان الگوریتم‌های مهاجرت ترکیبی با توجه به بیش‌ترین زیر توری آزاد بهترین نتیجه را از خود نشان داده‌اند که به ابعاد 9x13 می‌باشد. اما روش فوق در بدترین حالت دارای بیش‌ترین تعداد مهاجرت زیر توری که همان ۱۲ می‌باشد

با توجه به جای‌گذاری زیر توری‌ها در شکل (۱۲)-الف و بررسی روش‌های منفرد در جدول (۱)، مشاهده می‌شود که الگوریتم‌های TCB و LLRC غالباً کم‌ترین تعداد مهاجرت زیر توری را دارند که برای هر کدام به ترتیب تعداد ۲ و ۴ جابه‌جایی بیش‌ترین محدوده آزاد برای تخصیص این دو روش با ابعاد 7x9 و 7x10 می‌باشد که نسبت به اکثر روش‌های دیگر کوچک-ترین ابعاد را دارند. همچنین، الگوریتم ODC-FC میان



### ۳-۴-۳. میانگین بهره‌وری سیستم (ASU)

به میانگین پردازش پردازشگرها در سیستم بسیار شلوغ می‌گویند. این مقدار بین ۰ و ۱ می‌باشد و با معادله ۵ تخمین زده می‌شود.

$$SystemUtilization = \sum_{i=1}^t \frac{w \times h - n_i}{(w \times h) \times t} \quad (5)$$

در رابطه (۵)،  $n_i$  تعداد پردازنده‌های آزاد این سیستم در زمان  $i$ ،  $t$  کل زمان صرف شده و  $w \times h$  تعداد پردازنده‌های سیستم می‌باشد [۲۵]. در معادله یک تا سه  $N$  تعداد کل کارها می‌باشد [۲۶].

### ۳-۴-۴. میانگین تعداد مهاجرت کارها

به میانگین تعداد کارهایی که از طریق استفاده از روش‌های مهاجرت جابه‌جا می‌شوند گفته می‌شود. پس از هر طرح انتخاب زیر توری، تخصیص و مهاجرت محاسبه می‌شود و بر تعداد دوره‌هایی که هر طرح اجرا می‌شود تقسیم می‌گردد.

### ۳-۵. نتایج آزمایشگاهی و مشاهدات

در این بخش نتایج شبیه‌سازی‌هایی که برای ارزیابی کارایی الگوریتم‌های مهاجرت ترکیبی پیشنهادی انجام می‌شوند با شیوه‌های پیشین مقایسه شده است. بنابراین  $ODC-FC$  [۱۰]،  $TCB$  [۱۱]،  $TRBMA$  [۱۴]،  $LLRC$  [۱۳]،  $LTDC$  [۱۲]،  $TCB \& ODC-FC$  [۱۱]،  $ODC-FC \& TRBMA$  [۱۴]،  $LTDC \& ODC-FC$  [۱۴]،  $LTDC \& TRBMA$  [۱۴] و الگوریتم‌های مهاجرت ترکیبی پیشنهادی در نظر گرفته شده‌اند. این روش‌ها با شبیه‌ساز  $OM$  پیاده‌سازی شده‌اند. شایان‌ذکر است که شکل‌های (۱۴) تا (۲۱) از شبکه توری با ابعاد  $W=L=16$  یا  $W=L=10$  استفاده می‌کنند. این الگوریتم‌ها با روش‌های انتخاب زیر توری  $MD$ ،  $MPN$ ،  $MT \& MPN$  و روش‌های تخصیص  $TCB$  یا  $TRBMA$  کار می‌کنند. این نتایج با استفاده از ۷۰ کار کسب شده‌اند.

### ۳-۵-۱. ارزیابی میانگین زمان اجرای کار

همان‌طور که در شکل (۱۴) و جدول (۳) شرح داده شده است طرح  $MPN$  [13]/ $TRB$  [12]/ $HBM$  از میانگین زمان اجرای کار بهتری برای ۷۰ کار برخوردار می‌باشد. الگوریتم مزبور زیر توری‌ها را به مرز سطری توری هدایت می‌کند و آنها را برای مهاجرت به مرز سیستم توری هدایت می‌نماید و در نتیجه یک فضای آزاد بسیار وسیعی در مرکز توری به وجود می‌آورد. بنابراین در مقایسه با روش [14] ( $MT \& MPN/ISBA/...$ ) از توانایی بسیار زیادی برخوردار می‌باشد و ۳۸٫۲۲٪ بهتر از آنها عمل می‌کند.

است. سپس روش  $HCM$  با اولین و دومین ابعاد بزرگ زیر توری آزاد  $9 \times 11$  و  $6 \times 7$  بهترین نتیجه را کسب کرده است که بیشترین پیوستگی گره‌های آزاد در این الگوریتم را نشان می‌دهد. تعداد مهاجرت در این روش ۱۱ می‌باشد. در نهایت، روش  $HBM$  با ۹ مهاجرت زیر توری و اولین و دومین ابعاد بزرگ زیر توری آزاد با ابعاد  $9 \times 11$  و  $4 \times 7$  نسبت به الگوریتم‌های دیگر بهترین نتیجه را دارد. به‌طور کلی از فرآیندهای  $HBM$  و  $TCB \& ODC-FC$  برای تنظیم بیشتر نتایج مهاجرت زیر توری استفاده می‌شود.

### ۳-۴. تفسیر نتایج خروجی

تحقیق فعلی مبتنی بر استفاده از شبیه‌ساز  $OM$  می‌باشد. این شبیه‌ساز از طریق  $C\#$  توسعه یافته است. پارامترهای شبیه‌ساز در جدول (۲) نمایش داده شده‌اند. سیستم چندپردازنده‌ای شبیه‌سازی شده شامل چندپردازنده‌ای‌های با ابعاد  $W \times L$  در سراسر شبکه توری مربعی دوبعدی  $L$  و  $W$  می‌باشد. در این تحقیق  $W=L=16$  در نظر گرفته شده است. در واقع، برای تعیین سطوح مناسب دقت، هر آزمایش شبیه‌سازی بیش از ۱۰ بار تکرار می‌شود. خروجی شبیه‌سازی شامل میانگین زمان اجرای کار، میانگین زمان پاسخ کار، میانگین بهره‌وری سیستم و میانگین تعداد مهاجرت کار می‌باشد.

### ۳-۴-۱. میانگین زمان اجرای کار (AJET)

از زمان تخصیص تقاضای کار تا لحظه اتمام اجرای کار را میانگین زمان اجرای کار می‌گویند که با معادله ۳ محاسبه می‌شود.

$$AJET = \frac{\sum_{i=1}^N JET(P_i)}{N} \quad (3)$$

زمان اجرای کار  $i$ ام با عبارت  $JET(P_i)$  تعریف می‌شود [۲۴].

### ۳-۴-۲. میانگین زمان پاسخ کار (AJRT)

زمان تخصیص تقاضا تا اولین پاسخ واقعی کار را  $AJRT$  می‌گویند که با معادله ۴ محاسبه می‌شود.

$$AJRT = \frac{\sum_{i=1}^N JRT(P_i)}{N} \quad (4)$$

زمان پاسخ  $i$  امین کار با  $JRT(P_i)$  نمایش داده می‌شود [۲۴].

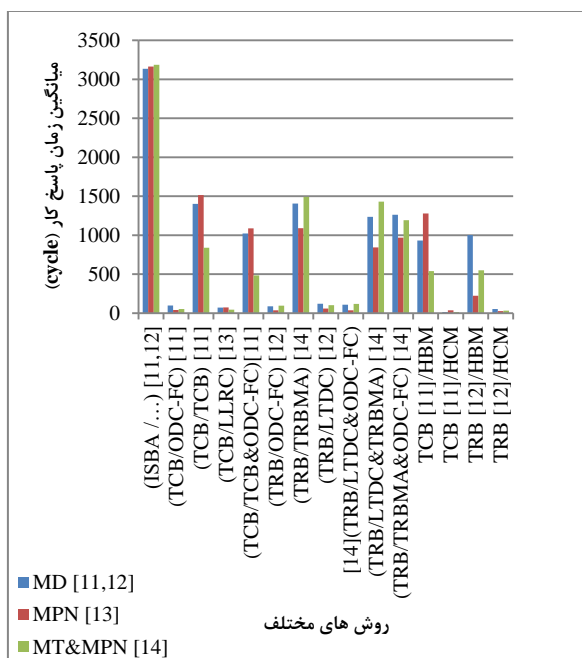
جدول (۲). پیکربندی شبیه‌ساز OM [۱۱، ۱۲]

مقادیر	پارامترهای شبیه‌سازی
۱۶×۱۶	اندازه NoC
۱ تا ۱۰۰۰ بیت در ثانیه	نرخ ارتباطات
ویدئو و رسانه	نوع کار
تصادفی بین ۹ تا ۳۲ هسته	اندازه کار
تصادفی بین ۱۰۰،۰۰۰ تا ۱،۰۰۰،۰۰۰	زمان بقای کار
تصادفی بین ۰ تا ۳۰۰،۰۰۰	زمان ورود کار
۲۰ میلیون سیکل	کل زمان شبیه‌سازی
تصادفی بین ۵۰ تا ۲۰۰	تعداد کارها

جدول (۳). نتیجه عملکرد میانگین زمان اجرای کار با توجه به طرح‌های مختلف و  $W=l=16$ 

ردیف	مهاجرت / تخصیص	MD (%)	MPN (%)	MT&MPN (%)
۱	(TCB/ODC-FC [10]) [11]	35.4090	37.8566	35.0291
۲	(TCB/TCB) [11]	36.3702	33.9135	37.7994
۳	(TCB/LLRC) [11]	36.2500	35.8621	37.3943
۴	(TCB/TCB&ODC-FC [10]) [11]	37.3874	34.1652	37.9756
۵	(TRB/ODC-FC [10]) [12]	37.0430	37.8497	36.1665
۶	(TRB/TRBMA) [14]	37.7410	37.1792	37.1643
۷	(TRB/LTDC) [12]	36.3347	35.8255	35.2076
۸	(TRB/LTDC&ODC-FC) [14]	35.7946	37.5774	35.0875
۹	(TRB/LTDC&TRBMA) [14]	35.7958	37.7982	34.7797
۱۰	(TRB/TRBMA&ODC-FC) [14]	36.0910	38.2159	36.6437
۱۱	TCB [11] /HBM	36.6402	36.5647	36.9309
۱۲	TCB [11] /HCM	37.6689	36.1734	38.0248
۱۳	TRB [12] /HBM	36.9480	38.2182	36.3908
۱۴	TRB [12] /HCM	35.2488	37.5179	36.6917

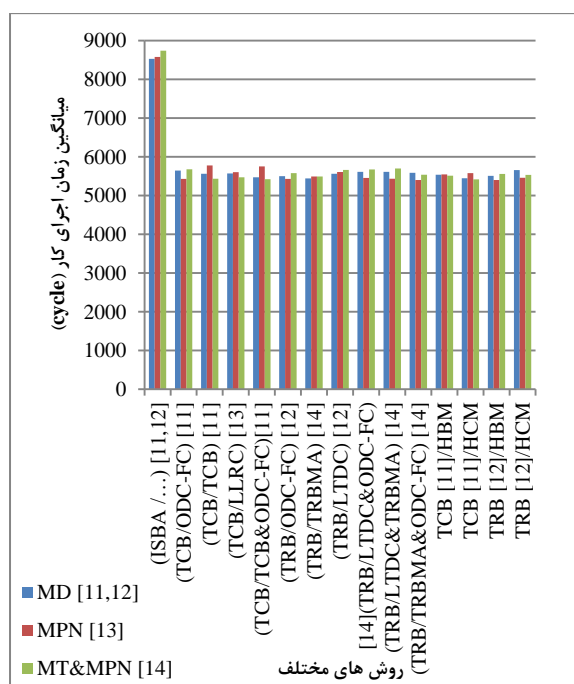
برخوردار می‌باشد و همان طوری که در جدول (۴) مشاهده می‌شود در مقایسه با الگوریتم [14]  $MT&MPN/ISBA/...$  می‌توان مشاهده کرد که در تمامی روش‌هایی که در آن‌ها از مهاجرت‌های مبتنی بر دو مرز ستونی یا دو مرز سطری استفاده نمی‌شود و روش‌های مبتنی بر الگوریتم‌های  $LLRC$ ،  $OCD-FC$  و  $LTDC$ ، میانگین زمان پاسخ کار به حداقل مقدار خود می‌رسد. در واقع، روش  $OCD-FC$  مبتنی بر هدایت زیرتوری‌ها به گوشه توری می‌باشد و الگوریتم‌های  $LLRC$  و  $LTDC$  غالباً زیرتوری‌ها را به یک مرز توری هدایت می‌نمایند. بنابراین، تخصیص، فضای آزادتری را فراهم می‌نماید. لذا در چنین روش‌هایی، میانگین زمان پاسخ کار به شکل چشمگیری کاهش می‌یابد.



شکل (۱۶). میانگین زمان پاسخ کار برای ۷۰ کار متفاوت در ابعاد مختلف طرح‌های تخصیص/مهاجرت/زیر توری و  $W=L=16$ .

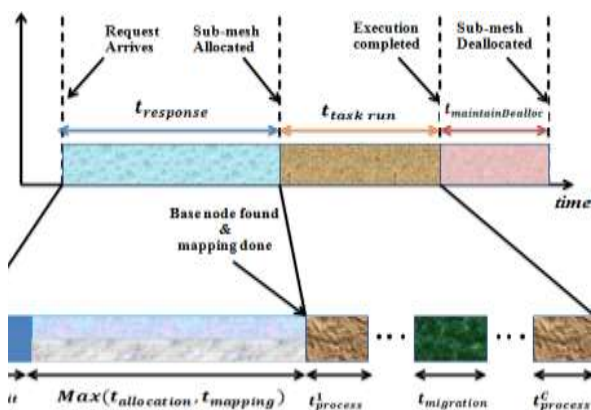
### ۳-۵-۳. ارزیابی میانگین بهره‌وری سیستم

همان‌طور که در شکل (۱۷) و جدول (۵) نشان داده شده است طرح  $MT&MPN/TRB/HBM$  دارای میانگین بهره‌وری سیستم بهتری برای ۷۰ کار ورودی می‌باشد و در مقایسه با روش  $(MPN/ISBA / ...)$  [۱۳] به مقدار ۴۹،۵۹۴۶ درصد بهتر عمل می‌نماید.



شکل (۱۴). میانگین زمان اجرای کار برای ۷۰ کار مختلف با در نظر گرفتن ابعاد مختلف طرح‌های زیر توری/تخصیص/مهاجرت و  $W=L=16$

عناصر زمانی در طول پاسخ به کار در شکل (۱۵) آمده است.



شکل (۱۵). اجزای زمانبندی در طول پاسخ به کار

زمان اجرا با معادله ۶ محاسبه می‌شود.

$$t_{execution} = t_{wait} + \max(t_{mapping}, t_{allocation}) + t_{migration} + t_{process} + t_{maintaindeallocation} \quad (6)$$

باید خاطر نشان کرد که شبیه‌ساز  $OM$  در زمان  $t_{maintaindeallocation}$  صفر در نظر گرفته شده است.

### ۳-۵-۲. ارزیابی میانگین زمان پاسخ کار

شکل (۱۶) نشان می‌دهد که طرح  $MT&MPN[14]/TCB[11]/HCM$  برای ۷۰ کار ورودی از میانگین زمان پاسخ بهتری در مقایسه با الگوریتم‌های دیگر

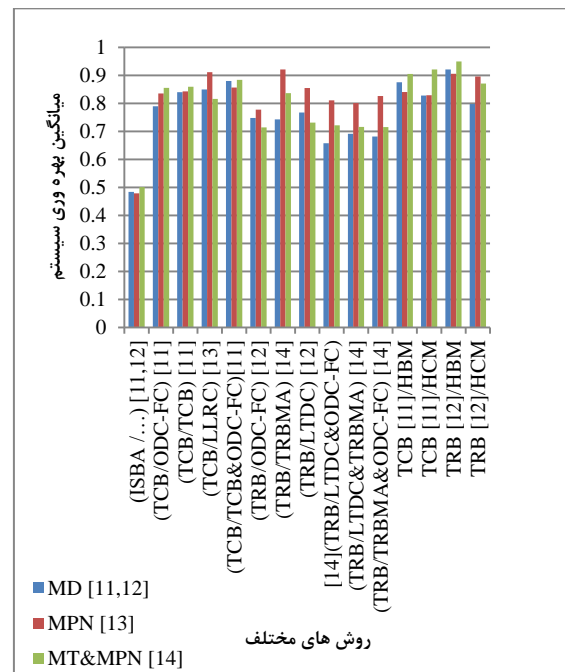
طرح متفاوت شامل انتخاب زیر توری متفاوت، تخصیص و مهاجرت می توان مشاهده کرد که شش روش نسبت به الگوریتم های دیگر با در نظر گرفتن سه پارامتر نتایج بهتری خواهند داشت که این نتایج در جدول (۶) فهرست شده اند. طبق جدول (۶) می توان مشاهده کرد که الگوریتم های مهاجرت ترکیبی پیشنهاد شده غالباً نسبت به روش های دیگر نتایج بهتری را در بر خواهند داشت.

### ۳-۵-۵. ارزیابی تعداد مهاجرت کارها

شکل (۱۸) تعداد مهاجرت کار در نگاشت برخط را نشان می دهد. استفاده از *ISBA* با الگوریتم های مهاجرت متفاوت، منجر به هیچ مهاجرتی نمی شود. زیرا پیدا کردن یک زیر توری مناسب برای تخصیص زمان زیادی را نیاز دارد. لذا در شکل (۱۸) مقداری برای آن در نظر گرفته نشده است. همچنین این شیوه طبق شکل (۲۴) کارایی کمتر از ۰/۵ دارد و نشان می دهد که بیش از نیمی از پردازنده ها بی کار هستند. بنابراین در این روش نیازی به مهاجرت کارها نمی باشد. بنابراین طبق شکل (۱۸) و جدول (۶) ملاحظه می شود که الگوریتم های مهاجرت ترکیبی برای کاهش تأثیر تکه تکه شدن خارجی و اختصاص محدوده بیشتر جهت تخصیص زیر توری ها غالباً منجر به کاهش تعداد مهاجرت کارها می شود.

جدول (۴). نتایج کارایی میانگین زمان پاسخ کار با در نظر گرفتن طرح های مختلف و  $W=L=16$

ردیف	مهاجرت / تخصیص	MD (%)	MPN (%)	MT&MPN (%)
۱	(TCB/ODC-FC [10]) [11]	96.8887	98.7448	98.3118
۲	(TCB/TCB) [11]	56.0060	52.4696	73.6287
۳	(TCB/LLRC) [11]	97.7595	97.6654	98.5660
۴	(TCB/TCB&ODC-FC [10]) [11]	67.9145	65.8623	84.8249
۵	(TRB/ODC-FC [10]) [12]	97.2527	98.8233	96.9907
۶	(TRB/TRBMA) [14]	55.8836	65.7933	53.1913
۷	(TRB/LTDC) [12]	96.2078	98.1298	96.7867
۸	ODC-FC [14]&(TRB/ LTDC	96.6063	98.8358	96.2517
۹	(TRB/LTDC&TRBMA) [14]	61.2150	73.5095	55.0866
۱۰	(TRB/TRBMA&ODC-FC) [14]	60.3772	69.5682	62.5612
۱۱	TCB [11] /HBM	70.7669	59.8626	83.0708
۱۲	TCB [11] /HCM	99.5324	98.8327	99.5387
۱۳	TRB [12] /HBM	68.5766	92.9679	82.7193
۱۴	TRB [12] /HCM	98.3086	99.2030	99.0021



شکل (۱۷). میانگین بهره وری سیستم برای کارهای مختلف با در نظر گرفتن ابعاد مختلف طرح های زیر توری/تخصیص/مهاجرت و  $W=L=16$

### ۳-۵-۴. ارزیابی شش طرح انتخابی

با توجه به جدول های (۳)، (۴) و (۵) و بررسی چهل و پنج

جدول (۵). نتایج کارایی میانگین بهره‌وری سیستم برای کارها با در نظر گرفتن طرح‌های مختلف و  $W=L=16$

ردیف	مهاجرت / تخصیص	MD (%)	MPN (%)	MT&MPN (%)
۱	(TCB/ODC-FC [10]) [11]	39.4051	42.705	44.0313
۲	(TCB/TCB) [11]	43.0119	43.208	44.2919
۳	(TCB/LLRC) [11]	43.6691	47.4706	41.3502
۴	(TCB/TCB&ODC-FC [10]) [11]	45.6023	44.1032	45.8545
۵	(TRB/ODC-FC [10]) [12]	36.0027	38.4704	32.9646
۶	(TRB/TRBMA) [14]	35.5633	48.0352	42.8008
۷	(TRB/LTDC) [12]	37.6124	43.9986	34.5591
۸	ODC-FC [14]&(TRB/ LTDC	27.2161	40.9523	33.6889
۹	(TRB/LTDC&TRBMA) [14]	30.7837	40.2596	33.1331
۱۰	(TRB/TRBMA&ODC-FC) [14]	29.7888	42.0530	33.1144
۱۱	TCB [11] /HBM	45.3102	43.0661	47.1167
۱۲	TCB [11] /HCM	42.2069	42.2627	48.0239
۱۳	TRB [12] /HBM	48.0521	47.1634	49.5946
۱۴	TRB [12] /HCM	39.9900	46.5856	45.0212

جدول (۶). بهترین طرح‌های انتخابی برای بهبود کارایی در نگاشت بر خط کار

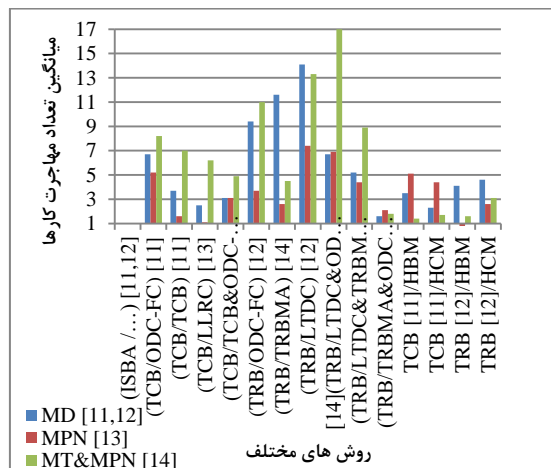
ردیف	مهاجرت / تخصیص / انتخاب زیر توری	میانگین زمان اجرای کار (%)	میانگین زمان پاسخ کار (%)	میانگین بهره‌وری سیستم (%)
۱	MT&MPN [14] /TCB [11] /HCM	38.0248	99.5387	48.0239
۲	MPN [13] /TRB [12] /HBM	38.2182	92.9679	47.1634
۳	MD [11,12] /TCB [11] /HCM	37.6689	99.5324	42.2069
۴	MT&MPN [14] /TRB [12] /HBM	36.3908	82.7193	49.5946
۵	MT&MPN [14] /TCB [11] /HBM	36.9309	83.0708	47.1167
۶	MPN [13] / (TRB/TRBMA&ODC-FC) [14]	38.2159	69.5682	42.0530

شکل (۱۸). میانگین تعداد مهاجرت کارها با در نظر گرفتن ابعاد

مختلف طرح‌های زیر توری / تخصیص / مهاجرت و  $W=L=16$ .

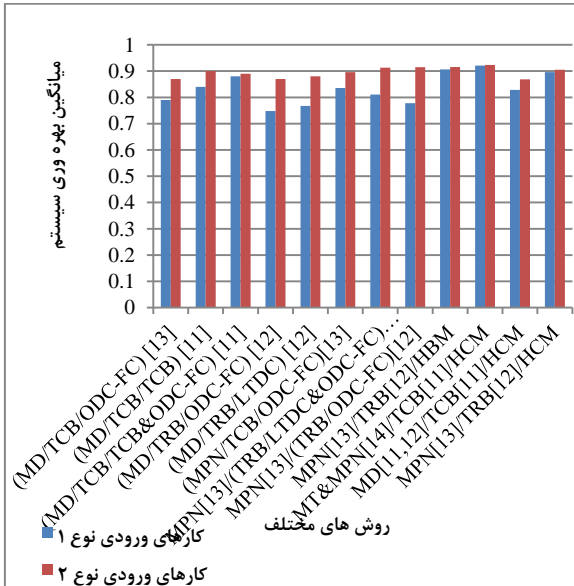
در شکل (۱۸) مشاهده می‌شود که طرح  $MPN[13]/TRB[12]/HBM$  نسبت به الگوریتم‌های انتخابی دارای کمترین مهاجرت کارها به مقدار ۰/۴ می‌باشد. همچنین طرح  $MT&MPN[14]/TCB[11]/HCM$  با در نظر گرفتن سه پارامتر میانگین زمان پاسخ کار، میانگین زمان اجرای کار و میانگین بهره‌وری سیستم به نتایج مطلوبی رسیده است.

۳-۵-۶. ارزیابی میانگین بهره‌وری سیستم برای دو اندازه مختلف کارهای ورودی با در نظر گرفتن





شکل (۱۹). میانگین بهره‌وری سیستم برای دو اندازه متفاوت از کارهای ورودی با در نظر گرفتن ابعاد مختلف طرح‌های زیر توری/ تخصیص/ مهاجرت و  $W=L=10$ .



شکل (۲۰). میانگین بهره‌وری سیستم برای دو اندازه متفاوت از کارهای ورودی با در نظر گرفتن ابعاد متفاوت طرح‌های زیر توری/ تخصیص/ مهاجرت و  $W=L=16$ .

### ۳-۵-۷. ارزیابی میانگین مصرف انرژی برای نگاشت برخط کار

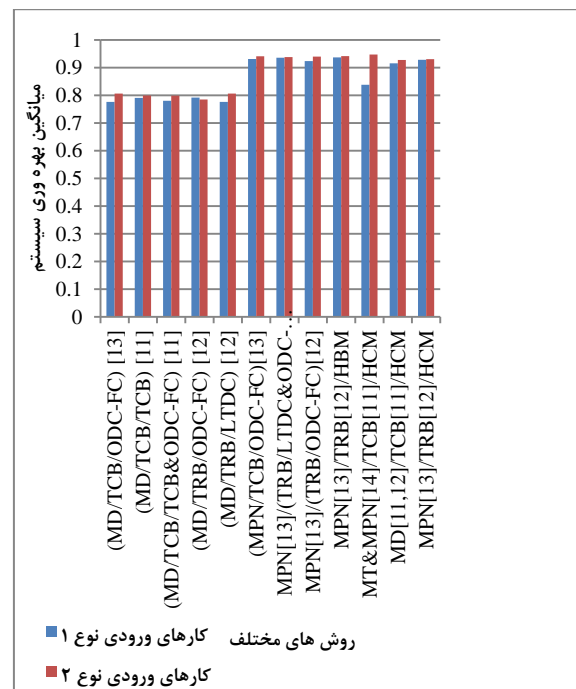
پس از شبیه‌سازی الگوهای متفاوت با استفاده از شبیه‌ساز OM، از فایل خروجی شبیه‌ساز که الگوهای ترافیک بر روی سطح توری هستند و هر فایل به‌عنوان ورودی شبیه‌ساز OMNET جهت ارزیابی میانگین مصرف انرژی برای نگاشت برخط کار استفاده می‌شود. باید خاطر نشان ساخت هر فایل شامل ارتباطات (نقطه منبع، نقطه مقصد و مقدار ترافیک) است. مصرف انرژی بر اساس ORION 2.0 برای مدل مسیریاب الکتریکی با تکنولوژی 32 nm محاسبه شده است [۲۷]. همچنین، اندازه قطعه  $2cm \times 2cm$  در نظر گرفته شده است که در جدول (۷) می‌توان آن را مشاهده نمود.

جدول (۷). پیکربندی ORION 2.0 [۲۷]

ارزش‌ها	پارامترهای شبیه‌سازی
۱۶×۱۶	اندازه توپولوژی
۲.۵ GHZ	فرکانس
۱.۰V	ولتاژ
۳۲nm	تکنولوژی
۲cm×۲cm	اندازه تراشه

### $W=L=16$ و $W=L=10$

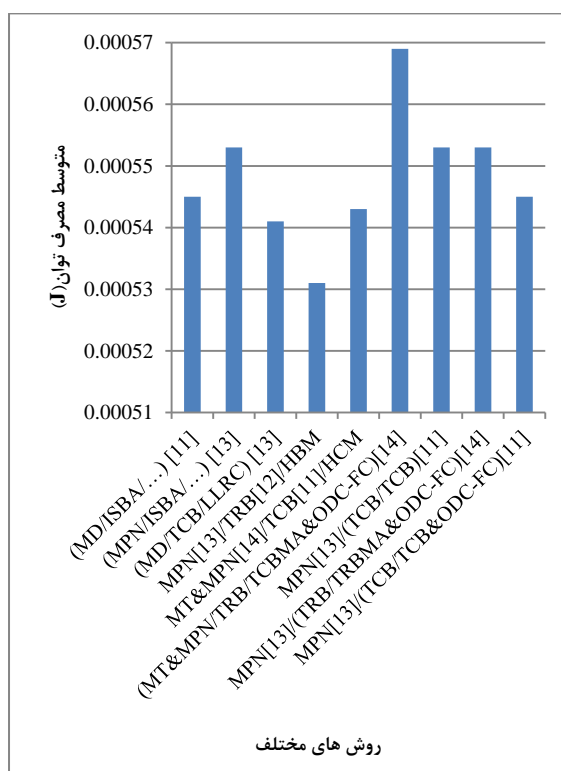
میانگین بهره‌وری سیستم برای دو اندازه مختلف از کارهای ورودی با طرح‌های مختلف انتخاب زیر توری/ تخصیص/ مهاجرت با توری‌هایی به اندازه  $10 \times 10$  و  $16 \times 16$  مورد بررسی قرار گرفته‌اند. باید خاطر نشان ساخت که تعداد کارها در کارهای ورودی نوع دوم بیشتر از کارهای ورودی نوع اول است. نتایج حاصل از شبیه‌سازی نشان می‌دهد که افزایش تعداد کارهای ورودی در سیستم منجر به افزایش میانگین بهره‌وری سیستم می‌شود که در شکل‌های (۱۹) و (۲۰) دو اندازه متفاوت از کارهای ورودی نشان داده شده است. همان‌طور که مشاهده می‌شود طرح  $MT \& MPN$  با افزایش تعداد کارهای ورودی و اندازه بندی دوباره زیر توری نسبت به روش‌های دیگر به نتایج بهتری دست می‌یابد. با توجه به شکل (۱۹) هنگامی که تعداد کارهای ورودی کم هستند بهره‌وری پایین روش مزبور را می‌توان مشابه کرد. از آنجایی که این روش، کمینه میانگین زمان پاسخ را دارد، طبق جدول (۴) و شکل (۱۶)، هنگامی که تعداد کارهای ورودی پایین هستند به کارها سریع‌تر پاسخ داده می‌شود؛ لذا زمان اجرای کار سریع‌تر می‌باشد. بنابراین، پردازنده‌های بیکار در چنین شرایطی در سیستم قابل‌دسترسی می‌باشند، در شرایطی که هنوز کار به محدوده زیر توری‌های تخصیص داده شده اختصاص نیافته است. بنابراین تعداد کارهای ورودی کم‌تر منجر به بهره‌وری پایین‌تر می‌شود (شکل ۱۹). همچنین با افزایش تعداد کارهای ورودی بیشتر پردازنده‌ها درگیر می‌شوند و بهره‌وری نیز افزایش می‌یابد.



نشان می‌دهد که روش‌های MT&MPN/TCB/HCB و MPN/TRB/HBM (ابعاد زیر توری/تخصیص/مهاجرت) در پارامترهای AJET, ATSU, AJRT و مصرف توان غالباً نتایج مطلوب‌تری دارند. افزون بر این، تحقیقات آینده در این حوزه بر روی پیاده‌سازی روش‌های پیشنهادی با معماری‌های دیگری همچون توری مدور، توری سه‌بعدی و رویکردهای نوری [۲۸]، استفاده از مکانیزم‌های انتخاب زیر توری متنوع و نیز بهره‌گیری از زیر توری‌های کوژ و ارائه الگوریتم‌های نگاشت مبتکرانه در روند پیشنهادی قابل انجام است.

### ۵. مراجع

1. A. Gangwar, Z. Xu, NK. Agarwal, R. Sreedharan, and A. Prasad, "Traffic Driven Automated Synthesis of Network-on-Chip from Physically Aware Behavioral Specification", IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 122-127, 2019.
2. W. Liao, H. Deng, Y. Luo, S. Xiao, C. Li, Z. Yu, "An Efficient and Low-Overhead Chip-to-Chip Interconnect Protocol Design for NOC", IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA), pp. 77-78, 2019.
3. WT. Shen, CH. Chao, YK. Lien, and AY. Wu, "A New Binomial Mapping and Optimization Algorithm for Reduced-Complexity Mesh-Based On-Chip Network", in Proceedings of the First International Symposium on Networks-on-Chip (NOC '7), pp. 317-322, 2007.
4. J. Janidarmian, A. Khademzadeh, and M. Tavnapour, "Onyx: A new heuristic bandwidth-constrained mapping of cores onto tile based Network on Chip", IEICE Electronics Express, Vol. 6, No. 1, pp. 1-7, 2009.
5. K. Windisch, V. Lo, and B. Bose, "Contiguous and non-contiguous processor allocation algorithms for k-ary n-cubes", Technical Report, University of Oregon, Oregon, USA, 1995.
6. S. Bani-Mohammad, and I. Ababneh, "Improving system performance in non-contiguous processor allocation for mesh interconnection networks", Simulation Modelling Practice and Theory, Vol. 80, pp. 19-31, 2018.
7. S. Bani-Mohammad, "Efficient Processor Allocation Strategies for Mesh-Connected Multicomputers", PhD Thesis, The Faculty of Information and Mathematical Sciences University of Glasgow, Glasgow, U.K, 2008.
8. S. Bani-Mohammad, M. Ould-Khaoua, I. Ababneh, and L. Mackenzie, "Comparative evaluation of contiguous allocation strategies on 3D mesh multicomputers", Journal of Systems and Software, Vol. 82, No. 2, pp. 307-318, 2009.
9. KJ. Windisch, W. Liu, and B. Nitzberg, "Non-contiguous processor allocation algorithms for mesh-connected multicomputers", IEEE Transactions on Parallel and Distributed Systems, Vol. 8, No. 7, pp. 712-726, 1997.
10. LK. Goh, and B. Veeravalli, "Design and performance evaluation of combined first-fit task allocation and migration strategies in mesh multicomputer systems", Journal of Parallel Computing, Vol. 34, No. 9, pp. 508-520, 2008.
11. A. Reza, and M. Rafie, "Improving Multi Task Running Time in Two Column Boundary Allocation Method in Mesh-based Chip Multiprocessors Using Combined Migration Mechanisms", International Journal of Computer & Information Technologies (IJOCIT), Vol. 3, No. 3, pp. 743-755, 2015.
12. A. Reza, and M. Rafie, "Performance improvement in



شکل (۲۱). مصرف توان برای نگاشت بر خط کار جهت دو طرح مختلف در شکل (۲۱)، مصرف انرژی در نگاشت بر خط کار جهت دو طرح متفاوت نشان داده شده است. رابطه مستقیمی بین مصرف توان و زمان اجرای کار وجود دارد. با توجه به شکل (۲۱)، مصرف توان در طرح  $MPN[13]/TRB[12]/HBM$  به دلیل زمان اجرای پایین بهبود یافته است. به طور کلی، روش‌های مبتنی بر الگوریتم‌های مهاجرت  $HBM$  و  $HCM$  به دلیل فرآیند آسیب پذیر برای مهاجرت زیر توری‌ها جهت کاهش زمان اجرای کار الگوریتم‌ها توانستند نتایج مطلوبی کسب نمایند.

شکل‌های (۱۴) تا (۲۱) و جدول‌های (۳) تا (۶) نشان می‌دهند که مهاجرت‌های ترکیبی غالباً عملکرد پارامترهای بهتری را به همراه دارد. بنابراین در مقایسه با الگوریتم‌های دیگر نتایج بهتری را از خود نشان می‌دهد.

### ۴. نتیجه‌گیری

در این مقاله یک نگاشت چندکاره پویا برای توپولوژی توری دوبعدی معرفی شد که شامل چهار مفهوم انتخاب زیر توری، تخصیص پردازنده، مهاجرت کار و نگاشت کار می‌باشد. به عبارت دیگر، انواع مختلف روش‌های تخصیص و مهاجرت با در نظر گرفتن سه الگوریتم انتخاب زیر توری با مکانیسم‌های مهاجرت ترکیبی پیشنهادی مقایسه شده‌اند. خروجی شبیه‌سازی

- Mapping Method for Mapping Large-Scale SNNs onto NoC-based Neuromorphic Hardware”, In Proceedings of the 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 640–647, 2020.
22. L. Mei, P. Houshmand, V. Jain, S. Giraldo, M. Verhelst, “ZigZag: Enlarging Joint Architecture-Mapping Design Space Exploration for DNN Accelerators”, IEEE Transactions on Computers, Vol. 70, No. 8, pp. 1160–1174, 2021.
23. J. Fang, H. Zong, H. Zhao, and H. Cai, “Intelligent mapping method for power consumption and delay optimization based on heterogeneous NoC platform”, Electronics, Vol. 8, No. 8, 912, 2019.
24. S. Bani-Ahmad, “On Improved Processor Allocation in 2D Mesh-based Multicomputers: Controlled Splitting of Parallel Requests”, International Conference on Communication, Computing and Security (ICCCS'11), pp. 204-209, 2011.
25. R. Zolfaghari, “Efficient and Quick Algorithm for Processor Allocation in Mesh Multi-Computers Network”, International Journal of Engineering and Advanced Technology (IJEAT), Vol. 2, No. 5, pp. 517-523, 2013.
26. N. Kumar, and DP. Vidyarthi, “A novel energy-efficient scheduling model for multi-core systems”, Cluster Computing, 2020, <https://doi.org/10.1007/s10586-020-03143-w>.
27. AB. Kahng, L. Bin, P. Li-Shiuan, and K. Samadi, “ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration”, Design, automation & test in Europe conference & exhibition, DATE'09, pp 423–428, 2009, <http://hdl.handle.net/1721.1/60547>.
28. A. Ghorbani, and H. Khodadadi, “Optimum design of WDM networks by EDFA optical amplifiers with the approach of improving communication distance and data transmission rate”, Journal of Applied Information and Communication Technology Innovations, Vol. 1, No. 1, pp. 67-73, 2021 (in Persian).
- multiprocessors using two row boundary allocation method and online dynamic compaction algorithm”, International journal of computer applications (IJCA), Vol. 123, No. 1, pp. 14-20, 2015.
13. M. Rafie, A. Khademzadeh, A. Reza, and M. Reshadi, “External Fragmentation Aware Dynamic Multi Task Mapping on Mesh Interconnection”, Journal of Computational and Theoretical Nanoscience, Vol. 13, No. 1, pp. 958-963, 2016.
14. M. Rafie, A. Khademzadeh, A. Reza, and M. Reshadi, “Performance evaluation of task migration in contiguous allocation for mesh interconnection topology”, Journal of Supercomputing, Vol. 72, No. 4, pp. 1660-1677, 2016.
15. DD. Sharma, and DK. Pradhan, “A fast and efficient strategy for submesh allocation in mesh-connected parallel computers”, Fifth IEEE Symposium on Parallel and Distributed Processing, pp. 682–689, 1993.
16. G. Chmaj, D. Zydek, and L. Koszalka, “Allocation Algorithms Problems in Mesh-Connected Systems”, 2004.
17. S. Khan, S. Anjum, U. A. Gulzari, M. K. Afzal, T. Umer, and F. Ishmanov, “An Efficient Algorithm for Mapping Real Time Embedded Applications on NoC Architecture”, IEEE Access, Vol. 6, pp. 16324–16335, 2018.
18. F. Ge, C. Cui, F. Zhou, and N. Wu, “A Multi-Phase Based Multi-Application Mapping Approach for Many-Core Networks-on-Chip”, Micromachines, Vol. 12, No. 6, 2021.
19. R. Khasanov, and J. Castrillon, “Energy-efficient Runtime Resource Management for Adaptable Multi-application Mapping”, In Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE), Grenoble, France, 9–13, pp. 909–914, March 2020.
20. W. Amin, F. Hussain, S. Anjum, S. Khan, N. K. Baloch, Z. Nain, and S. W. Kim, “Performance evaluation of application mapping approaches for network-on-chip designs”, IEEE Access, Vol. 8, pp. 63607–63631, 2020.
21. L. Zhang, S. Li, L. Qu, Z. Kang, S. Wang, J. Chen, and L. Wang, “MAMAP: Congestion Relieved Memetic Algorithm based