



Article Info

Received: 2020-7-28
Accepted: 2021-5-7

Keywords

Multiprocessor embedded systems, Lifetime, Power consumption, Task scheduling and mapping, Multi-objective optimization, Particle swarm optimization

How to Cite this article

Atena Abdi, Hamid Reza Zarandi and Shahrokh Jalilian, "Lifetime Enhancement of Embedded Systems During Task Scheduling Through a Multi-Objective Particle Swarm Optimization Approach", *Journal of Space Science, Technology and Applications*, vol 1 (1), p.: 92-107, 2021.

Lifetime Enhancement of Embedded Systems During Task Scheduling Through a Multi-Objective Particle Swarm Optimization Approach

Atena Abdi¹, Hamid Reza Zarandi^{*2} and Shahrokh Jalilian³

1. Department of Computer Engineering, Amirkabir University of Technology,
atena_abdi@aut.ac.ir

2. Department of Computer Engineering, Amirkabir University of Technology,
h_zarandi@aut.ac.ir

3Khaje Nasir Toosi, Tehran, Iran, shjalilian@gmail.com

*Corresponding author

Abstract

In this paper, a task scheduling and mapping method based on multi-objective particle swarm optimization (MOPSO) algorithm is presented to improve lifetime reliability of multiprocessors systems on chip. This method considers power consumption temperature and performance along with the lifetime reliability due to the antagonistic relations of these parameters. These antagonistic and dependent relations make the design space exploration and optimization processes complex. The proposed method is based on MOPSO algorithm and starts with an initial population of candidate solutions which are generated randomly and represents valid task scheduling and mapping solutions. By changing the scheduling and mapping parameters during the MOPSO algorithm, new solutions are produced and the design space is explored based on the objective of the target problem of this method. Several experiments on random and real-life benchmarks are performed to verify the effectiveness of our proposed method. The results demonstrate the capability of the proposed method in effective design space exploration and generating the Pareto front. Moreover, comparisons to the related research show 35%, 23%, 19% and 3% improvements in performance, lifetime reliability, temperature, and power consumption on average.



دسترس پذیر در نشانی:
Journal.isrc.ac.ir

بهبود طول عمر سیستم‌های نهفته در حین زمان بندی وظایف بر اساس بهینه سازی چندهدفی ازدحام ذرات

آتنا عبدی^۱، حمیدرضا زرنندی^{۱*}، شاهرخ جلیلیان^۲

۱. دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران. Email: Atena_abdi@aut.ac.ir

۲.*. دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران. Email: h_zarand@aut.ac.ir (نویسنده مسئول)

۳. دانشگاه خواجه نصیرالدین طوسی، تهران، ایران، sh.jalilian@isrc.ac.ir

دو فصلنامه

علوم، فناوری و

کاربردهای فضایی

سال اول، شماره ۱، صفحه ۹۲-۱۰۷
بهار و تابستان ۱۴۰۰

Available in:
Journal.isrc.ac.ir/article_
137480.html

DOI:

10.22034/JSSTA.2021.241516.100

تاریخچه داوری

دریافت: ۱۳۹۹/۰۵/۰۷

پذیرش: ۱۴۰۰/۰۲/۱۷

واژه‌های کلیدی

سیستم چندپردازنده‌ای نهفته، طول عمر، توان مصرفی، کارایی، زمان بندی و نگاهت وظایف، بهینه‌سازی چندهدفی، بهینه‌سازی ازدحام ذرات

نحوه استناد به این مقاله

آتنا عبدی، حمیدرضا زرنندی، شاهرخ جلیلیان، "بهبود طول عمر سیستم‌های نهفته در حین زمان بندی وظایف بر اساس بهینه‌سازی چندهدفی ازدحام ذرات"، دو فصلنامه علوم، فناوری و کاربردهای فضایی، جلد اول، شماره اول، صفحات ۹۲-۱۰۷، ۱۴۰۰.

چکیده

این مقاله به بررسی تاثیر زمان بندی وظایف بر روی طول عمر یک سیستم بی درنگ سخت که از مخزن انرژی مرکب متشکل از باتری و ابرخازن و برداشت گر انرژی خورشیدی برای تامین انرژی خود استفاده می‌کند، می‌پردازد. منظور از طول عمر سیستم در این مستند، لحظه شروع به کار سیستم تا لحظه مختل شدن وظایف آن به دلیل نبود انرژی است. با توجه به خواص غیرخطی باتری و ابرخازن که موجب می‌شود شارژ داخلی آن‌ها در دو بخش در دسترس (IAC) و غیرقابل دسترس (IUC) تقسیم شود، طول عمر چنین سیستمی کاملاً به الگوی شارژ و دشارژ مخزن انرژی وابسته است زیرا این الگو در نهایت منجر به میزان شارژ ذخیره شده در بخش IUC و میزان شارژ استخراج شده از این بخش می‌شود. بنابراین، با مدیریت الگوی شارژ/دشارژ مخازن انرژی می‌توان روی طول عمر سیستم و افزایش آن تاثیرگذار بود. از آنجایی که الگوی رسیدن انرژی از محیط خارج از کنترل سیستم است، ایده اصلی این مقاله تاثیرگذاری بر روی الگوی شارژ/دشارژ مخزن از طریق تنظیم الگوی مصرف انرژی است تا در نهایت طول عمر سیستم بهبود یابد. در این راستا، ابتدا دو الگوریتم زمان بندی MCF و MGF که الگوی زمان بندی در اجرای پرمصرف‌ترین و کم مصرف‌ترین وظیفه حاضر در سیستم هستند، ارائه می‌شوند. سپس الگوریتم MCG مورد بررسی قرار می‌گیرد که طبق آن الگوریتم، در هر برهه از زمان با توجه به شرایط موجود، در مورد استفاده از یکی از الگوریتم‌های مذکور تصمیم‌گیری می‌شود. نتایج آزمایش‌ها نشان می‌دهد که این الگوریتم بین ۵٪ تا ۱۶٪ طول عمر سیستم را افزایش می‌دهد. با توجه به این که در سال‌های اخیر موضوع استفاده از ابرخازن در کنار باتری و سلول‌های خورشیدی در سامانه‌های فضایی مطرح شده است، از این‌رو، نتایج این تحقیق می‌تواند برای استفاده در ماهواره‌ها نیز بررسی شود.

پارامترهای طراحی را در حین فرایند نگاشت تکنولوژی در سطح RTL بهبود می‌دهند. گرچه اغلب روش‌های بهبود، به دلیل سربار زمانی کم و انعطاف‌پذیری بالا در سطح سیستم ارائه شده‌اند [۶-۹].

تعیین زمان و مکان اجرای هر یک از اجزای برنامه کاربردی، یکی از گام‌های اصلی در فرآیند طراحی سیستم‌های نهفته است که «زمان‌بندی و نگاشت وظایف» نامیده می‌شود. این فرایند یکی از روش‌های اساسی سطح سیستم به‌منظور بهبود چالش‌های طراحی سیستم‌های نهفته است که به دلیل هزینه و سربار کم بسیار مورد استفاده قرار می‌گیرد [۷، ۱۰، ۱۱]. مطالعات نشان می‌دهند که بهبود چالش‌های طراحی در حین فرایند زمان‌بندی و نگاشت وظایف از نظر پیچیدگی، مشابه مسئله تخصیص درجه دوم بوده و در رده مسائل NP-hard قرار می‌گیرد [۵]. همچنین طراحی فرایند زمان‌بندی وظایف، یکی از مراحل اساسی در ایجاد واحد مدیریت فرمان و داده در سامانه‌های ماهواره‌ای است که عملکرد مناسب و کارایی این واحد را برحسب محدودیت‌های موجود تضمین می‌کند.

در نظر داشتن تمامی چالش‌ها در حین طراحی سیستم، به دلیل رابطه ناهم‌سوی بین عوامل موثر بر آن‌ها بسیار چالش‌برانگیز است. به‌عنوان مثال، افزایش ولتاژ و فرکانس عملیاتی سیستم منجر به بهبود کارایی شده و از سوی دیگر، توان مصرفی، دما و نرخ وقوع خطاهای سخت را افزایش می‌دهد [۱۲-۱۴]. به دلیل پیچیدگی بیشتر سیستم‌های چندپردازنده‌ای روی تراشه ناهمگن، ناهم‌سویی و موازنه بین چالش‌های طراحی در آن‌ها شدت بیشتری یافته و در نظر داشتن آن‌ها دشوارتر می‌شود [۴، ۵، ۱۵]. بهینه‌سازی چالش‌های سیستم‌های چندپردازنده‌ای روی تراشه در حین زمان‌بندی و نگاشت وظایف، راهکاری کم‌هزینه و مناسب است که با گسترش به‌کارگیری این سیستم‌ها در کاربردهای نهفته و بی‌درنگ اهمیت زیادی یافته است.

با توجه به پیچیدگی این مسئله، راهکار قطعی برای حل آن در زمان چندجمله‌ای وجود ندارد و روش‌های مکاشفه‌ای و فرامکاشفه‌ای با هدف زمان‌بندی و نگاشت بهینه وظایف ارائه شده‌اند که به طور عمده بر یک یا دو چالش اساسی متمرکز بوده‌اند. به دلیل رابطه ناهم‌سوی این چالش‌ها و

همزمان با پیشرفت فناوری و پیچیده شدن کاربردها، بهبود کارایی و استفاده مطمئن از سامانه‌های کامپیوتری به یکی از اساسی‌ترین چالش‌های طراحی تبدیل شده است. یکی از کاربردهای سیستم‌های کامپیوتری، در سامانه‌های فضایی و ماهواره‌ای است. سیستم‌های کامپیوتری تعبیه شده، در واحد مدیریت فرمان و داده ماهواره‌ها با هدف دریافت و پردازش داده‌ها و ارسال فرامین صحیح برحسب شرایط به‌کار گرفته می‌شوند. سرعت و دقت پردازش سیستم نهفته این واحد بر کارایی ماهواره تاثیر زیادی داشته و به همین دلیل، استفاده از سیستم‌های چندپردازنده‌ای در این حوزه توصیه می‌شود. سیستم‌های چندپردازنده‌ای روی تراشه که به اختصار MPSoC نامیده می‌شوند، بر این اساس عمل کرده و تمامی اجزای سیستم شامل چندین واحد پردازشی را بر روی یک تراشه مجتمع می‌کنند. برحسب نوع عناصر پردازشی، این سیستم‌ها به دو دسته همگن و ناهمگن تقسیم می‌شوند. سیستم‌های چندپردازنده‌ای ناهمگن به دلیل قابلیت‌های کارکردی و عملکردی بیشتر نسبت به سیستم‌های پردازشی همگن، استاندارد غالب و پذیرفته شده در طراحی سیستم‌های نهفته شناخته شده‌اند [۵-۸].

در کنار قابلیت‌های کارکردی و عملکردی بسیار زیاد این سیستم‌ها، پیچیدگی‌های ذاتی آن‌ها محدودیت‌ها و چالش‌هایی را نیز برای طراحان ایجاد نموده است. مسأله توان مصرفی، نقاط داغ سطح تراشه و قابلیت اطمینان، مهم‌ترین چالش‌هایی است که با کاهش اندازه ترانزیستورها و افزایش تراکم آن‌ها در تراشه‌های امروزی شدت یافته‌اند. در نظر داشتن این چالش‌ها در حین طراحی و به‌کارگیری سیستم‌های چندپردازنده‌ای روی تراشه، یکی از الزامات طراحی آن‌ها به‌شمار می‌آید [۱-۳]. با توجه به اهمیت مسئله طول عمر، صحت اطلاعات و همچنین مدیریت توان مصرفی در سامانه‌های ماهواره‌ای، در نظر داشتن تمامی این عوامل در حین فرایند طراحی از اهمیت به‌سزایی برخوردار است.

به‌منظور مواجهه با این چالش‌ها و در نظر داشتن موازنه ذاتی بین آن‌ها، روش‌های گوناگونی در سطوح مختلف تجزیه ارائه شده‌اند. به‌عنوان نمونه دسته‌ای از روش‌ها سعی در بهبود قابلیت اطمینان و توان مصرفی از طریق تنظیم اندازه گیت ترانزیستورها در سطح مدار دارند. روش‌های دیگر،

- بهینه‌سازی توأم طول عمر، توان مصرفی، دمای تراشه و کارایی در سیستم‌های نهفته چندپردازنده‌ای به‌کار رونده در کاربردهای فضایی و سامانه‌های ماهواره‌ای در حین فرایند سطح سیستمی زمان‌بندی و نگاشت وظایف
- ارائه موتور خودکار جستجوی فضای طراحی مبتنی بر الگوریتم فرامکاشفه‌ای MOPSO برای حل مسئله بهینه‌سازی تعریف شده و تطبیق دادن عملگرهای الگوریتم با مسئله با هدف رعایت تنوع در جمعیت، اجتناب از نقاط بهینه محلی و هم‌گرایی زودهنگام
- مدل‌سازی جامع طول عمر با در نظر گرفتن تمامی مکانیزم‌های خرابی تاثیرگذار بر سیستم‌های چندپردازنده‌ای روی تراشه در حین بهینه‌سازی
- گسترش فضای طراحی با در نظر گرفتن روش‌های بهبود توان مصرفی و دمای تراشه در حین بهینه‌سازی با هدف جستجوی موثرتر و بهبود کیفیت مجموعه جواب نهایی

در ادامه و در بخش ۲، پژوهش‌های پیشین مشابه مورد بحث و بررسی قرار می‌گیرند. در بخش ۳، مفاهیم اولیه و فرضیات روش پیشنهادی ارائه شده و در بخش ۴ جزئیات روش فرامکاشفه‌ای پیشنهادی شرح داده می‌شوند. در بخش ۵ به آزمایشات تجربی و بررسی کارایی روش ارائه شده پرداخته شده و جمع‌بندی و روال آتی در بخش ۶ ارائه شده‌اند.

۲- پژوهش‌های پیشین

مسئله زمان‌بندی و نگاشت وظایف در سیستم‌های چندپردازنده‌ای روی تراشه با هدف بهینه‌سازی پارامترهای طراحی بسیار پیچیده بوده و در رده NP-hard طبقه‌بندی می‌شوند. به دلیل پیچیدگی، حل قطعی این مسئله میسر نبوده و برای بهینه‌سازی توابع هدف گوناگون در طی این فرایند، روش‌های مکاشفه‌ای و فرامکاشفه‌ای ارائه می‌شود [۵، ۱۲، ۱۳، ۱۶، ۱۸]. در حالت کلی، این مسئله به دو صورت بدون محدودیت و با محدودیت تعریف و حل می‌شود. در حالت بدون محدودیت، تمرکز بر بهینه‌سازی کارایی سیستم بوده و در نوع با محدودیت، مهم‌ترین چالش‌های سیستم‌های چندپردازنده‌ای روی تراشه مانند قابلیت اطمینان، توان مصرفی و دمای تراشه در حین زمان‌بندی و نگاشت وظایف در نظر گرفته می‌شوند [۶، ۱۲، ۱۳، ۱۵]. بیشتر پژوهش‌های پیشین ارائه شده در دسته روش‌های مکاشفه‌ای حل مسئله قرار دارند و یک یا دو

همچنین موازنه بین آن‌ها، در نظر داشتن هم‌زمان تمامی این عوامل بسیار دشوار است. به دلیل کاربرد گسترده این سیستم‌ها در کاربردهای نهفته، پارامتر کارایی که بر رعایت موعد وظایف تاثیرگذار است بسیار حائز اهمیت است. بهبود کارایی بر سایر پارامترهای طراحی موثر است و می‌تواند منجر به تشدید بحران در آن‌ها شود. به‌عنوان مثال در سیستم‌های چندپردازنده‌ای روی تراشه، توان پردازشی پردازنده‌ها یکسان نبوده و تخصیص وظایف به پردازنده سریع‌تر به‌منظور بهبود کارایی؛ توازن بار کاری در سطح تراشه را از بین برده و می‌تواند منجر به تسریع پیری تراشه و کاهش قابلیت اطمینان شود. از این رو توجه هم‌زمان به تمامی چالش‌های طراحی و در نظر داشتن آثار ناهمسوی آن‌ها یکی از نیازهای طراحان است [۲، ۶]. از سوی دیگر روش‌های فرامکاشفه‌ای به دلیل سادگی، انعطاف‌پذیری و جستجوی موثرتر فضای جواب راه‌حل‌های مناسب‌تری را در زمان بیشتری نسبت به الگوریتم‌های مکاشفه‌ای ارائه می‌دهند [۱۶].

در این مقاله، روشی فرامکاشفه‌ای به منظور بهینه‌سازی هم‌زمان طول عمر، توان مصرفی، دمای تراشه و کارایی به‌عنوان مهم‌ترین چالش‌های طراحی سیستم‌های چندپردازنده‌ای روی تراشه نهفته ارائه شده است. روش ارائه شده مبتنی بر الگوریتم بهینه‌سازی چند هدفی ازدحام ذرات (MOPSO) است که با توجه به موثر بودن این الگوریتم در حل مسائل بهینه‌سازی چند هدفی و جستجوی سریع و موثر فضای طراحی انتخاب شده است [۱۲، ۱۶، ۱۷]. در این روش، طول عمر به‌عنوان اساسی‌ترین چالش طراحی سیستم‌های چندپردازنده‌ای روی تراشه در نظر گرفته شده است همچنین کارایی، توان مصرفی و دمای تراشه به دلیل رابطه متقابل و موثر بر طول عمر، در حین بهینه‌سازی لحاظ شده‌اند. طول عمر سیستم توسط حاصل جمع نرخ بروز خطاهای دائمی رایج سیستم‌های کامپیوتری تخمین زده شده و توان مصرفی به‌صورت مجموع عوامل ایستا و پویا در نظر گرفته شده است. همچنین به‌منظور گسترش دادن فضای طراحی و یافتن جواب‌های بهینه، روش‌های بهبود مقیاس پویای ولتاژ و فرکانس کاری و افزودن بازه‌های زمانی خنک‌کننده با هدف کاهش توان مصرفی و دمای تراشه در روش پیشنهادی در نظر گرفته شده است. نوآوری‌های روش پیشنهادی این مقاله را می‌توان در موارد زیر خلاصه نمود:

محدودیت از چالش‌های اساسی سیستم‌های چندپردازنده‌ای روی تراشه را در نظر گرفته‌اند. همچنین غالب روش‌های فرامکاشفه‌ای ارائه شده در حوزه حل مسئله زمان‌بندی و نگاشت بدون محدودیت هستند که پیچیدگی کمتری دارند. از سوی دیگر، بیشتر روش‌های ارائه شده در بستر سیستم‌های چندپردازنده‌ای همگن بوده و به معماری ناهمگن که در طراحی سیستم‌های ناهمگن اهمیت زیادی دارند کمتر پرداخته شده است.

مسئله توان مصرفی به‌عنوان یکی از اساسی‌ترین چالش‌های سیستم‌های چندپردازنده‌ای که با پیشرفت فناوری و افزایش چگالی توان اهمیت یافته است، هدف بسیاری از پژوهش‌های حیطة زمان‌بندی و نگاشت بوده است. در [۱۹] روش مکاشفه‌ای جهت زمان‌بندی و نگاشت وظایف با هدف کمینه کردن زمان اجرا و رعایت محدودیت مشخص شده توان مصرفی سیستم ارائه شده است. در این روش، پارامترهای توان مصرفی و زمان اجرا به‌صورت گوسی مدل شده‌اند و عملیات زمان‌بندی و نگاشت بر اساس کمینه کردن زمان اجرا و رعایت آستانه توان مصرفی طبق جدولی که به‌صورت ایستا ذخیره‌سازی شده انجام می‌گیرد. در [۲۰] روش زمان‌بندی مبتنی بر اجرای تکراری وظایف با هدف کاهش ارتباطات بین هسته‌ای و کمینه‌سازی توان مصرفی ارائه شده است. در [۲۱]، مجموعه‌ای از روش‌های مکاشفه‌ای به منظور بهینه‌سازی توان ایستای سیستم‌های چندپردازنده‌ای در حین زمان‌بندی و نگاشت وظایف ارائه شده است. در یکی از روش‌های مکاشفه‌ای بر اساس مطالعات دمایی هسته‌های پردازشی و بررسی انتشار گرما در سطح تراشه، وظایف به عناصر پردازشی نگاشت می‌شوند. همچنین به‌دلیل موثر بودن دما بر توان محاسباتی ایستا، در روش دیگر معیار انتخاب، مناسب‌ترین هسته بر اساس دمای فعلی آن‌ها است.

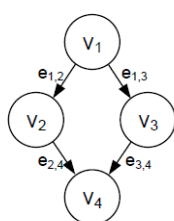
مسئله طول عمر یکی از مهم‌ترین چالش‌های سیستم‌های چندپردازنده‌ای روی تراشه است که با پیشرفت فناوری و کاهش سایز ترانزیستورها اهمیت فراوانی یافته است و هدف بسیاری از پژوهش‌های بهینه‌سازی سطح سیستم بوده است. پژوهش‌های این حوزه را می‌توان به دو دسته تقسیم نمود: روش‌هایی که طول عمر را از طریق کنترل دما به‌عنوان موثرترین عامل تاثیرگذار بر خطاهای سخت بهینه می‌کنند و روش‌هایی که این عامل را به طور مستقیم هدف قرار می‌دهند و اثربخشی بیشتری دارند. در [۲۲]، روشی فرامکاشفه‌ای بر پایه الگوریتم شبیه‌سازی ذوب به منظور بهینه‌سازی طول عمر سیستم ارائه شده است. در

این روش، خطای electro migration به‌عنوان مدل طول عمر سیستم در نظر گرفته و بهینه شده است. دیگر روش ارائه شده در این زمینه [۵] است که نگاشت وظایف آگاه از طول عمر بر اساس الگوریتم کلونی مورچگان است. در این پژوهش، مورچه‌ها شبیه‌سازی نگاشت وظایف مختلف هستند و اطلاعات را از سایر کلونی‌ها از طریق فرومون‌ها عبور می‌دهند به‌نحوی که الگوریتم جستجو به حدی از جواب که طول عمر سیستم را بهینه می‌کند، همگرا شود [۵]. در [۲۳]، روشی دو مرحله‌ای برای بهبود طول عمر ارائه نموده است. در حین طراحی، پارامتر متوسط زمان خرابی سیستم با در نظر داشتن محدودیت زمان اجرا و موعد وظایف بهینه می‌شود. به‌منظور حل مسئله بهینه‌سازی و استخراج محدوده جواب در این پژوهش، از الگوریتم مکاشفه‌ای شبیه‌سازی ذوب پذیرش آستانه استفاده شده است که نسبت به روش شبیه‌سازی ذوب سربار محاسباتی کمتری دارد.

در [۲۴] توان مصرفی پویا و طول عمر سیستم با در نظر داشتن خطای electro migration در رویکرد زمان‌بندی بر اساس الگوریتم شبیه‌سازی ذوب بهینه شده است. در [۱۲]، روش بهینه‌سازی مبتنی بر الگوریتم NSGA-II به‌منظور بهینه‌سازی طول عمر، نرخ خطاهای گذرا و توان مصرفی ایستا ارائه شده است. در این روش، طول عمر سیستم محدود به خطای electro migration بوده و توان مصرفی پویا هدف قرار داده شده است. لازم به‌ذکر است که رابطه متخاصم توان مصرفی و قابلیت اطمینان از نظر سطح ولتاژ و فرکانس کاری سیستم در این پژوهش به‌خوبی مطالعه و بررسی شده است.

با توجه به مرور پژوهش‌های صورت گرفته می‌توان گفت، بهینه‌سازی جامع تمامی چالش‌های اساسی طراحی سیستم‌های چندپردازنده‌ای روی تراشه، هدف پژوهش‌های بسیار کمی بوده است که بر پایه ارائه روش‌های مکاشفه‌ای هستند [۲۵، ۲۶]. این در حالی است که، با وجود سرعت روش‌های مکاشفه‌ای، دقت آن‌ها در جستجوی فضای طراحی و یافتن نقاط بهینه، نسبت به روش‌های فرامکاشفه‌ای بسیار کمتر است. روش‌های مکاشفه‌ای به‌علت دید محدود ناشی از ساده‌سازی مسئله و همچنین عدم امکان بازگشت به عقب در حین حل مسئله، نسبت به روش‌های فرامکاشفه‌ای از دقت کافی برخوردار نیستند. همچنین ویژگی‌های ذاتی روش‌های فرامکاشفه‌ای در جستجوی فضای طراحی و اجتناب از بهینه‌های محلی با حفظ تنوع در جمعیت و انعطاف‌پذیری، آن‌ها را به انتخاب بهتری برای حل

تعریف می‌شود که در آن گره‌های V_{app} وظایف و یال‌های E_{app} وابستگی داده‌ای آن‌ها را نمایش می‌دهد. پارامتر W_{app} هزینه اتصال هر دو وظیفه را به صورت وزن یال ارتباطی بین آن‌ها نشان می‌دهد. همچنین زمان اجرای هر وظیفه در مجموعه E_{app} بر اساس بدترین زمان اجرا و فرکانس نامی پردازنده تعیین شده و در حین اجرا بر حسب فرکانس عملیاتی سیستم مقیاس می‌شود. در پایان، پارامتر H_{app} ناهمگنی در ویژگی‌ها و شرایط تخصیص وظایف نشان داده شده و مجموعه هسته‌های پردازشی مناسب برای اجرای هر وظیفه از برنامه کاربردی را معین می‌نماید. نمونه‌ای از گراف وظایف در شکل ۲ ارائه شده است [۱۸، ۲۹].



شکل ۲. نمونه‌ای از گراف وظایف برنامه کاربردی

۳-۳- مدل سازی طول عمر

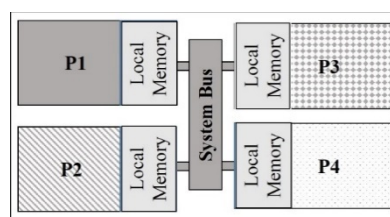
در سیستم چندپردازنده‌ای روی تراشه، احتمال خرابی تمامی واحدهای پردازشی و اتصالات وجود دارد و این خرابی به دو صورت تعریف می‌شود: خرابی‌های گذرا که بر اثر بروز خطاهای نرم رخ می‌دهند و خرابی‌های دائمی که در اثر وقوع خطاهای سخت ایجاد می‌شوند. خطاهای سخت اثر دائمی داشته و به دلیل آسیب‌های فیزیکی رخ می‌دهند که منجر به بروز خرابی‌های دائمی شده و عموماً قابل بازیابی نیستند [۲۲، ۳۰]. مهم‌ترین انواع مکانیزم‌های خرابی دائمی که بر طول عمر سیستم‌های چندپردازنده‌ای تاثیرگذار بوده و روابط محاسبه آن‌ها در سطح سیستم، در جدول ۱ خلاصه شده است.

مسئله زمان‌بندی و نگاشت و بهینه‌سازی چالش‌های طراحی تبدیل می‌کند [۱۶، ۲۷، ۲۸].

۳- کلیات و مفاهیم اولیه

۳-۱- مدل معماری

معماری سیستم چندپردازنده‌ای روی تراشه مفروض در این مقاله، متشکل از هسته‌های پردازشی متصل به یک شبکه میان‌ارتباطی است که دارای قابلیت‌های متفاوت در اجرای وظایف هستند. ناهمگنی هسته‌های پردازشی در مدل مفروض، به صورت سطوح ولتاژ و فرکانس کاری متفاوت در هسته‌های پردازشی تعریف شده است. همچنین در این مدل، هر یک از هسته‌های پردازشی توانایی اجرای تمامی وظایف برنامه کاربردی را نداشته و وظایف بر حسب نوع و ویژگی به پردازنده‌های متفاوتی نگاشت می‌شوند. شبکه میان‌ارتباطی سیستم چندپردازنده‌ای در روش‌های پیشنهادی به صورت باس در نظر گرفته شده است. این معماری به صورت گراف $Arc = \{P_{arc}, L_{arc}\}$ تعریف می‌شود که در آن P_{arc} گره‌ها و نشانگر مجموعه هسته‌های پردازشی بوده و L_{arc} یال‌ها و بیانگر اتصالات بین هسته‌ای است. نمونه‌ای از این معماری در شکل ۱ نشان داده شده است که در آن سیستم شامل چهار واحد پردازشی ناهمگن است که از طریق شبکه باس به یکدیگر متصل شده‌اند [۱۲، ۲۹]. برحسب کاربرد مورد نظر، جزئیات سخت‌افزاری سیستم چندپردازنده‌ای مانند شیوه اشتراک منابع، معماری حافظه‌ها و سلسله مراتب آن‌ها در مدل‌سازی لحاظ می‌شوند.



شکل ۱. مدل معماری سیستم چندپردازنده‌ای روی تراشه

۳-۲- مدل برنامه کاربردی

بخش نرم‌افزاری سیستم چندپردازنده‌ای روی تراشه، شامل برنامه کاربردی اجرا شونده بر روی بستر سخت‌افزاری است. برنامه کاربردی به صورت مجموعه‌ای از وظایف مستقل که توسط گراف جهت‌دار بدون دور مدل‌سازی می‌شود، در نظر گرفته شده است. این گراف به صورت $G_{app} = \{V_{app}, E_{app}, W_{app}, H_{app}\}$

یافته‌اند. توان مصرفی سیستم چندپردازنده‌ای روی تراشه بر حسب توان ایستا و پویا به صورت (۲) محاسبه می‌شود:

$$P_{system} = P_{dynamic} + P_{static} = C_{eff} \times V^2 \times f + \alpha \times T(t) + \beta \quad (2)$$

در این رابطه، C_{eff} ظرفیت خازنی، V و f ولتاژ و فرکانس عملیاتی سیستم است. همچنین α و β ضرایب وابسته به معماری هستند که بر اساس ویژگی‌های سیستم تعریف می‌شود و T دمای تراشه در زمان t بر حسب درجه کلوین است [۳۷].

همچنین دمای لحظه‌ای سیستم کامپیوتری به توان مصرفی و تغییرات دمای آن در طی زمان وابسته است. معادله دیفرانسیلی محاسبه دمای یک سامانه کامپیوتری به صورت (۳) تعریف می‌شود [۳۹]:

$$C \cdot \frac{dT}{dt} + G(T - T_{amb}) = P(t) \quad (3)$$

در این رابطه، C و G ثابت‌های وابسته به معماری و نشان‌دهنده ظرفیت خازنی و هدایت گرمایی بوده، T ، t ، T_{amb} و P به ترتیب دما، زمان، دمای محیط و توان مصرفی لحظه‌ای سیستم طبق (۲) هستند.

۴- شرح روش زمان‌بندی و نگاشت پیشنهادی

در این بخش، به جزئیات مسئله مورد نظر در این مقاله و روش زمان‌بندی و نگاشت ارائه شده پرداخته می‌شود. مسئله مورد نظر در این مقاله، بهبود طول عمر سیستم‌های نهفته چندپردازنده‌ای روی تراشه ناهمگن یا در نظر داشتن کارایی، توان مصرفی و دمای تراشه در حین فرایند زمان‌بندی و نگاشت وظایف است. بدین منظور داده‌ها و ورودی‌های مسئله عبارتند از:

- مدل معماری سیستم چندپردازنده‌ای روی تراشه ناهمگن
- گراف برنامه کاربردی مورد نظر و زمان اجرای هر وظیفه و اتصال در معماری سیستم
- بیشینه مقادیر آستانه طول عمر، توان مصرفی، دمای سیستم چندپردازنده‌ای مورد نظر و همچنین موعد اجرای برنامه کاربردی
- اهدافی که در این مسئله دنبال می‌شوند عبارتند از:

جدول ۱. مکانیزم‌های خرابی تأثیرگذار بر طول عمر و روابط آن‌ها

مکانیزم خرابی	رابطه سطح سیستم	تعریف پارامترها
مهاجرت الکترون (EM)	$MTTF_{EM} = \frac{A_{EM}}{j^n} e^{\frac{E_{EM}}{kT}}$	n ثابت تجربی، J چگالی جریان، E_{aEM} انرژی فعال‌سازی، K ثابت بولتزمن، T دما
مهاجرت بار (SM)	$MTTF_{SM} = A_{SM} (\Delta T)^{-n} e^{\frac{E_{SM}}{kT}}$	n ثابت تجربی، ΔT اختلاف دما با حالت اولیه، E_{aSM} انرژی فعال‌سازی، K ثابت بولتزمن، T دما
شکستگی زمانی دی‌الکتریک (TDDB)	$MTTF_{TDDB} = A_{TDDB} \left(\frac{1}{V}\right)^{a-bT} e^{\frac{x+y+ICT}{ZT}}$	a, b, x, y, Z پارامترهای تناسب، V ولتاژ کاری، T دما
چرخه دمایی (TC)	$MTTF_{TC} = A_{TC} \left(\frac{1}{\Delta T}\right)^q$	ΔT اختلاف دما با محیط، q ثابت وابسته به جنس
ناپایداری یاباس منفی دما (NBTI)	$MTTF_{NBTI} = \frac{A_{NBTI}}{V_{GS}} e^{\frac{E_{NBTI}}{kT}}$	V_{GS} ولتاژ گیت-سورس، E_{aNBTI} انرژی فعال‌سازی، K ثابت بولتزمن، T دما

همان‌گونه که در روابط ذکر شده مشاهده می‌شود، دمای تراشه (T) عامل موثر و مشترک در بروز تمامی مکانیزم‌های خرابی شرح داده شده است. به منظور مدل‌سازی طول عمر سیستم، در [۳۴] معیاری بر اساس جمع نرخ‌های خرابی ارائه شده که طول عمر سیستم را بر اساس جمع نرخ خرابی تمامی اجزای آن و بر اثر مکانیزم‌های خرابی مختلف به صورت زیر تخمین می‌زند:

$$MTTF_{SYSTEM} = \frac{1}{\lambda} = \frac{1}{\sum_{p=1}^j \sum_{l=1}^k \lambda_{pl}} \quad (1)$$

در این رابطه، λ_{pl} نرخ خطای واحد پردازشی p به علت رخداد مکانیزم خرابی l را نشان می‌دهد. در این مقاله به منظور بهبود طول عمر سیستم‌های چندپردازنده‌ای روی تراشه ناهمگن، این پارامتر توسط معیار جمع نرخ‌های خطا و همچنین مدل‌سازی جداگانه آن‌ها در فرایند بهینه‌سازی در نظر گرفته شده است.

۳-۴- مدل‌سازی توان مصرفی و دمای تراشه

توان مصرفی اجرای یک وظیفه یا اتصال برنامه کاربردی بر بستر سخت‌افزاری سیستم شامل دو جزء اساسی است: توان پویا و توان ایستا [۳۶]. توان پویا به ولتاژ و فرکانس کاری سیستم وابسته بوده و توان ایستا از جریان نشتی که به طور عمده بر اثر دمای تراشه ایجاد می‌شود تأثیر می‌گیرد. با وجود آنکه در بسیاری از پژوهش‌های پیشین از توان ایستا تراشه صرف‌نظر شده است، مطالعات نشان می‌دهند که این نوع توان در طی زمان و با پیشرفت فناوری اهمیت زیادی

وظایف ارائه شده‌اند که بیشتر روش‌ها به بهینه‌سازی زمان اجرا پرداخته‌اند و سایر پارامترهای طراحی را لحاظ ننموده‌اند [۱۶]. در روش پیشنهادی این مقاله به منظور حل مسئله شرح داده شده از ساختار حل مسائل بهینه‌سازی چندهدفی MOPSO [۱۷] که بر پایه الگوریتم بهینه‌سازی ازدحام ذرات استفاده شده است. این موتور جستجو، مجموعه جواب نزدیک به بهینه را در قالب پوسته Pareto استخراج نموده و اصول رتبه‌دهی و شایسته‌سالاری در جمعیت (مجموعه جواب‌های کاندید) را به منظور حفظ تنوع در جواب‌ها و اجتناب از بهینه‌های محلی در نظر می‌گیرد. بدین ترتیب و با به‌کارگیری عملگر جهش با هدف اجتناب از هم‌گرایی زودهنگام و جواب‌های بهینه محلی، مجموعه جواب نهایی مسئله به دست می‌آید.

روش پیشنهادی، یک موتور جستجوی خودکار فضای طراحی بر اساس الگوریتم MOPSO است که طول عمر، توان مصرفی، دمای تراشه و کارایی سیستم را به طور توأمان بهینه‌سازی می‌کند. کارایی سیستم به صورت بیشینه زمان اجرای وظایف برنامه کاربردی بر هسته‌های پردازشی تخمین زده می‌شود. طول عمر سیستم بر حسب نرخ مکانیزم‌های خرابی EM، SM، TDDDB، TC و NBTI که در بخش پیشین شرح داده شد، محاسبه می‌شود و توان مصرفی حاصل جمع مولفه‌های ایستا و پویا است که به سطوح ولتاژ و فرکانس و جریان نشتی سیستم وابسته است. در نهایت دمای تراشه به عنوان عامل تاثیرگذار بر چگالی توان مصرفی و طول عمر سیستم در کنار سایر پارامترها در نظر گرفته می‌شود. همچنین به منظور جستجوی موثرتر فضای جواب و گسترش دادن محدوده آن، مقیاس پویای ولتاژ و فرکانس کاری سیستم و افزودن بازه‌های زمانی خالی به اجرا، با هدف کاهش توان مصرفی پویا و دمای سیستم در استخراج جواب توسط روش پیشنهادی در نظر گرفته شده است.

روش پیشنهادی با مجموعه‌ای از جواب‌های اولیه در قالب ذرات مورد نیاز روش ازدحام ذرات شروع به کار می‌کند. هر یک از این ذرات یک جواب معتبر مسئله زمان‌بندی و نگاشت هدف در این مقاله است که محدودیت‌های تعریف شده را رعایت می‌نماید. هر جواب شامل زمان‌بندی اجرای وظایف برنامه کاربردی بر هسته‌های پردازشی و اطلاعات نگاشت آن‌ها به هسته‌های پردازشی ناهمگن و سطوح ولتاژ و فرکانس اجرایی است. در زمان‌بندی مشخص شده برای ذرات باید محدودیت‌هایی از جمله ترتیب اجرای وظایف و تقدم و تاخر تعریف شده بر اساس گراف برنامه کاربردی رعایت شده باشد.

○ تعیین زمان و مکان اجرای هر وظیفه از برنامه کاربردی در هسته‌های پردازشی سیستم مورد نظر به صورت ایستا به نحوی که:

- طول عمر، توان مصرفی، دمای تراشه و کارایی سیستم بهینه شوند
- رابطه ناهم‌سوی میان پارامترهای طراحی در نظر گرفته شود
- توزیع متناسب وظایف بر هسته‌های پردازشی با هدف اجتناب از پیری زودرس هسته‌های پردازشی رعایت شود
- رابطه تقدم و تاخیر میان وظایف برنامه کاربردی رعایت شود
- حدود آستانه در نظر گرفته شده برای پارامترهای طراحی سیستم رعایت شود.

● قیدهایی که در این مسئله در نظر گرفته شده‌اند عبارتند از:

- تخصیص هر وظیفه به یک هسته پردازشی و یک سطح ولتاژ و فرکانس کاری
- تخصیص وظیفه به هسته پردازشی مناسب بر حسب نوع پردازش و ناهمگن بودن هسته‌های پردازشی
- رعایت تقدم و تاخر در اجرای وظایف بر اساس وابستگی‌های داده‌ای
- رعایت محدودیت‌های خاص سیستم مانند موعد زمانی اجرا، توان مصرفی و دمای تراشه

۴-۲- شرح روش پیشنهادی

روش پیشنهادی در این مقاله با هدف حل مسئله شرح داده شده در سطح سیستم ارائه شده است. بدین منظور پارامترهای اساسی طراحی در حین زمان‌بندی و نگاشت وظایف در قالب یک مسئله بهینه‌سازی چندهدفی در نظر گرفته می‌شوند. زمان‌بندی و نگاشت وظایف با هدف بهینه‌سازی چالش‌های طراحی مسئله‌ای با پیچیدگی NP-hard است که جواب‌های نزدیک به بهینه آن توسط راهکارهای مکاشفه‌ای و فرامکاشفه‌ای استخراج می‌شود. با توجه به دقت و انعطاف‌پذیری بیشتر روش‌های فرامکاشفه‌ای در جستجوی فضای طراحی و استخراج مجموعه جواب بهینه، روش پیشنهادی این مقاله بر پایه این دسته از الگوریتم‌ها است. روش‌های فرامکاشفه‌ای زیادی مانند الگوریتم ژنتیک، مورچه‌ها، PSO و ... برای حل مسئله زمان‌بندی و نگاشت

چنانچه زمان‌بندی مشخص شده این ترتیب را رعایت نکند، ذره مورد نظر معتبر نبوده و از مجموعه جواب ورودی مسئله حذف می‌شود. همچنین اطلاعات نگاشت هر وظیفه به هسته‌های پردازشی برحسب تعریف ناهمگونی در معماری سیستم چندپردازنده‌ای مشخص بوده و همچنین ولتاژ و فرکانس اجرای آن و زمان‌های بی‌کاری در نظر گرفته شده با هدف مدیریت دما نیز به‌طور تصادفی تخصیص داده شده‌اند. در نتیجه هر ذره شامل اطلاعات زمان‌بندی و نگاشت وظایف برنامه کاربردی و یک جواب معتبر بالقوه از مسئله تعریف شده است. در روش پیشنهادی این مقاله، به‌منظور اجتناب از هم‌گرایی زودهنگام به نقاط بهینه محلی و حفظ تنوع، جمعیت اولیه ورودی به‌صورت تصادفی تولید شده است.

پس از تولید جمعیت اولیه، الگوریتم بهینه‌سازی ازدحام ذرات چندهدفی بر آن‌ها اعمال شده و مجموعه غیرمغلوب از جواب‌های جدید تولید شده ذخیره می‌شوند. در هر مرحله هر ذره از جمعیت بر اساس سه معیار سرعت پیشین، بهترین خاطره شخصی ذره و بهترین جواب پیدا شده بر اساس بهینه‌سازی توأم اهداف مسئله را با هدف کشف تمام فضای طراحی حرکت داده می‌شود. انتخاب بهترین خاطره شخصی و بهترین جواب پیدا شده بر اساس معیار غلبه و فاصله نقاط صورت می‌گیرد. معیار غلبه بر اساس تابع هزینه تعریف شده مسئله مورد نظر در این مقاله با هدف بهینه‌سازی هم‌زمان طول عمر، توان مصرفی، کارایی و دمای تراشه به‌صورتی که در (۴) آمده تعریف می‌شود:

$$\forall \text{ core in multiprocessor system,} \quad (4)$$

$$\text{Cost} = \text{Max}(SL), \text{Max}(\Lambda), \text{Max}(P), \text{Max}(\theta)$$

در این رابطه، پارامترهای SL ، P و Λ به ترتیب نشانگر زمان اجرای وظایف، نرخ خطاهای موثر بر طول عمر، توان مصرفی و دمای هر هسته پردازشی هستند. در این رابطه، می‌توان Λ را به‌صورت نرخ بروز هر یک از مکانیزم‌های خرابی یا مجموع آن‌ها در نظر گرفت. به‌منظور تعریف تابع هزینه ارائه شده بر بستر سیستم چندپردازنده‌ای روی تراشه، بیشینه مقدار هر یک از پارامترها در هسته‌های پردازشی در نظر گرفته می‌شود. کمینه کردن این تابع هزینه در حین فرایند زمان‌بندی و نگاشت وظایف منجر به بهینه‌سازی هم‌زمان پارامترهای طراحی مدنظر می‌شود. لازم به ذکر است که رابطه محاسبه هر یک از پارامترهای استفاده شده

در رابطه (۷) در بخش ۳ از مقاله بیان شده و بر آن اساس در الگوریتم پیشنهادی به‌کار گرفته می‌شود. موتور جستجوی فضای طراحی مبتنی بر بهینه‌سازی چندهدفی ازدحام ذرات استفاده شده در این مقاله، برای هر ذره از فضای جواب، سه معیار در نظر می‌گیرد: (۱) سرعت ذره، (۲) بهترین جوابی که ذره در طی مسیرش در فضای جواب یافته است و (۳) بهترین جوابی که در کل فرایند توسط ذرات مختلف کشف شده است. بهترین جواب‌های محلی و کلی هر ذره بر اساس تابع هزینه تعریف شده در (۷) بر اساس مسئله مورد نظر در این مقاله و معیار پراکندگی مشخص می‌شوند. در هر مرحله از الگوریتم، هر ذره در جهت سه معیار ذکر شده (سرعت، بهترین جواب محلی ذره و بهترین جواب کلی الگوریتم تاکنون) حرکت داده شده و نقطه جدیدی از فضای جواب کشف می‌شود. این نقطه با بهترین جواب محلی ذره مقایسه شده و در صورتی که بتواند بر آن غلبه کند جایگزین می‌شود و در غیر این صورت الگوریتم ادامه پیدا می‌کند. همچنین با توجه به چندهدفی بودن مسئله بهینه‌سازی موردنظر در این مقاله، بهترین جواب‌های فضای طراحی در یک مجموعه ذخیره شده و برای اعمال بر هر ذره بر اساس معیار پراکندگی و به‌صورت تصادفی انتخاب می‌شود. بدین ترتیب انتخاب بهترین جواب کلی برای هر ذره، از مجموعه نقاط نامغلوب و بر اساس معیار بولتزمان به‌صورت زیر و با ناحیه‌بندی فضای کشف شده انجام می‌شود.

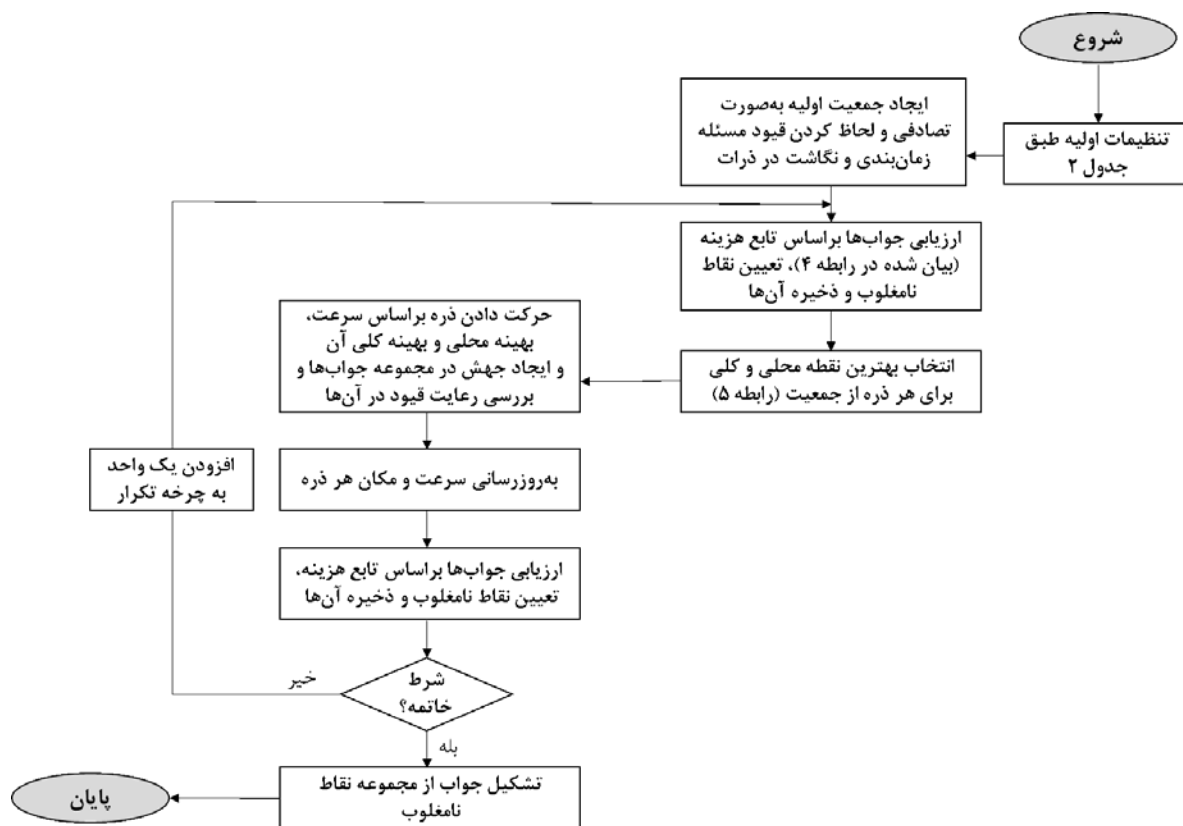
$$P_i = \frac{e^{-\beta n_i}}{\sum_j e^{-\beta n_j}} \quad (5)$$

در این رابطه، P_i احتمال انتخاب شدن ذره از ناحیه i مجموعه نقاط نامغلوب، n_i تعداد نقاط این ناحیه و β فشار انتخاب در روش بولتزمان را نشان می‌دهد. بدین ترتیب نقاطی که در نواحی کم‌جمعیت‌تر مجموعه ذرات نامغلوب قرار گرفته‌اند با اولویت بیشتری انتخاب شده و معیار پراکندگی به‌خوب بودن جواب‌ها افزوده می‌شود.

جدول ۲. تنظیمات اولیه پارامترهای الگوریتم پیشنهادی

اندازه جمعیت	۲۰۰
تعداد تکرارها	۴۰۰
تعداد هسته‌های پردازشی	۴
تعداد سطوح ولتاژ و فرکانس	۳
احتمال جهش	۰/۴
ضریب اینرسی سرعت	۰/۹۹
ضریب تاثیر بهینه محلی	۱
ضریب تاثیر بهینه کلی	۲

با تکرار الگوریتم، در هر مرحله ذرات جدیدی ایجاد شده و مجموعه جواب‌های بهینه مسئله در قالب یک پوسته Pareto ذخیره می‌شوند.



شکل ۳. فلوچارت شیوه عملکرد و جزئیات روش فرامکاشف‌های زمان‌بندی و نگاشت پیشنهادی

طراحی شده، در محیط MATLAB 2014 پیاده‌سازی شده است. همچنین محک‌های تصادفی و واقعی متعددی از برنامه‌های کاربردی که عموماً در سیستم‌های نهفته اجرا می‌شوند از مجموعه‌های TGFF و E3S در آزمایشات و شبیه‌سازی‌ها در نظر گرفته شده است [۴۰، ۴۱]. لازم به ذکر است که آزمایشات انجام شده در این بخش بر بستر یک سیستم مجهز به پردازنده Intel Core i5 با چهار هسته پردازشی و حافظه RAM با ظرفیت ۶۴ گیگابایت انجام شده است.

آزمایشات تجربی بر اساس اجرای گراف وظایف برنامه‌های کاربردی بر یک سیستم چهارپردازنده‌ای روی تراشه ناهمگن با اتصالات گذرگاه (مشابه مدلسازی ارائه شده در بخش سوم) انجام شده‌اند. با توجه به ناهمگن بودن سیستم هدف، هر هسته پردازشی قابلیت اجرای برخی از وظایف گراف برنامه کاربردی را دارد. همچنین فرض شده که هر هسته پردازشی دارای سه سطح ولتاژ و فرکانس کاری است که با توجه به ناهمگن بودن سیستم چندپردازنده‌ای هدف این مقادیر متفاوت نیستند. محدوده سطوح ولتاژ و فرکانس پردازشی هسته‌ها بین ۱ تا ۱/۶ ولت و ۳۰۰ تا ۹۰۰ مگاهرتز در نظر گرفته شده که این مقادیر از طراحی پردازنده‌های سیستم‌های نهفته ARM CORTEX الهام گرفته شده‌اند [۱۲]. برنامه‌های کاربردی توسط گراف‌های DAG با ویژگی‌های اجرایی مشخص ذکر شده در بخش ۳، برای وظایف و اتصالات به صورت تصادفی یا نمونه‌های واقعی مدل‌سازی شده‌اند.

روش پیشنهادی این مقاله، زمان‌بندی و نگاشت وظایف را با هدف بهینه‌سازی طول عمر، توان مصرفی، کارایی و دمای تراشه بر اساس الگوریتم MOPSO انجام می‌دهد. تنظیمات اولیه این الگوریتم در تعیین جواب نهایی مسئله بسیار موثر بوده و مقادیر در نظر گرفته شده در پیاده‌سازی و آزمایشات این بخش در جدول ۲ نمایش داده شده است. طول عمر، توان مصرفی و دمای تراشه به عنوان اهداف مسئله بهینه‌سازی مورد مطالعه در این مقاله به صورتی که در بخش ۳ شرح داده شد در قالب یک تابع هزینه مدل‌سازی شده‌اند. همچنین پارامتر کارایی سیستم به صورت بیشینه زمان اجرا در هر یک از هسته‌های پردازشی محاسبه می‌شود.

آزمایش‌های انجام گرفته و ارائه شده در این بخش به منظور بررسی موثر بودن روش پیشنهادی و قابلیت آن در بهینه‌سازی هم‌زمان پارامترهای طراحی به سه دسته تقسیم می‌شوند. دسته

تکرار الگوریتم تا تحقق شرط خاتمه به صورت تکراری ادامه می‌یابد. شرط خاتمه بر اساس هم‌گرایی مجموعه جواب‌ها تعریف شده است، بدین صورت که اگر مجموعه جواب نهایی با ۱ درصد تکرارهای قبلی میانگین تفاوت بسیار اندکی داشته باشند به هم‌گرایی رسیده‌ایم. به عبارتی عدم بهبود محسوس در مجموعه جواب نهایی در طی تکرارهای متوالی به معنای ایجاد هم‌گرایی و خاتمه فرایند بهینه‌سازی است. همچنین بررسی تداوم این روند در چندین تکرار متوالی، منجر به تمایز از بهینه‌های محلی می‌شود.

لازم به ذکر است که در حین اجرای الگوریتم بهینه‌سازی چندهدفی ازدحام ذرات، به منظور اجتناب از نقاط بهینه محلی، از عملکرد جهش نیز استفاده شده است. در این حالت، هر ذره با احتمال مشخصی به صورت گوسی جهش داده شده و نقطه حاصل شده به فضای جواب افزوده می‌شود. بدین ترتیب خروجی نهایی مسئله یک پوسته Pareto متشکل از بهترین جواب‌های کشف شده در حین جستجو است.

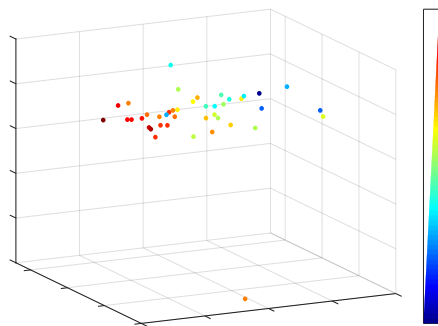
در مجموعه جواب حاصل شده در قالب پوسته Pareto تمامی اعضا نقاط بهینه بوده و هیچ‌یک نسبت به دیگری برتری ندارد. انتخاب هر یک از جواب‌های پوسته Pareto طبق نیاز کاربر و محیط به‌کارگیری بر حسب رابطه ناهمسوی اهداف مسئله بهینه‌سازی مورد نظر قابل انجام است. لازم به ذکر است که جستجوی دقیق فضای طراحی توسط روش فرامکاشف‌های بهینه‌سازی ازدحام ذرات پیشنهادی، دقت و سرشار زمانی زیادی داشته و جهت اعمال در زمان طراحی و پیش از اجرای سیستم در نظر گرفته شده است. تنظیمات اولیه الگوریتم فرامکاشف‌های پیشنهادی و مقادیر اولیه پارامترهای مورد نیاز آن در جدول ۲ نشان داده شده است. شکل ۳ عملکرد روش پیشنهادی جهت بهینه‌سازی هم‌زمان طول عمر، توان مصرفی، کارایی و دمای تراشه سیستم‌های چندپردازنده‌ای ناهمگن در حین زمان‌بندی و نگاشت وظایف را نشان می‌دهد.

۵- نتایج تجربی

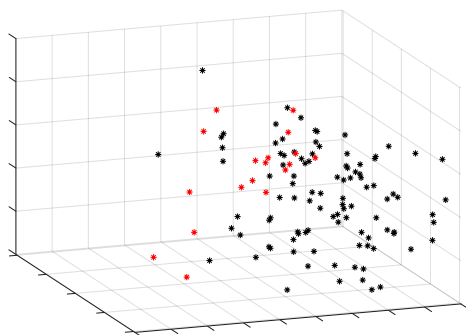
در این بخش، به بررسی کارایی روش پیشنهادی و قابلیت‌های آن در بهینه‌سازی هم‌زمان پارامترهای طراحی و رعایت رابطه متخاصم بین آن‌ها که هدف اصلی مسئله تعریف شده در این مقاله است می‌پردازیم. بدین منظور، موتور جستجوی فضای طراحی پیشنهادی که بر اساس الگوریتم MOPSO

بیشتر دمای کمتری دارند و بدین ترتیب رابطه ناهمسوی این دو پارامتر تأیید می‌شود.

همچنین به منظور درک بهتر روابط متقابل و تغییرات اهداف بهینه‌سازی مسئله تعریف شده، مجموعه جواب حاصل شده پس از حل در سه بعد با ثابت در نظر گرفتن دمای تراشه در شکل ۵ نشان داده شده است. در این شکل نقاط مشکی تمامی نقاط کشف شده فضای طراحی در طی اجرای الگوریتم MOPSO بوده و نقاط قرمز نشان‌دهنده مجموعه جواب بهینه مسئله و نقاط نامغلوب است.



شکل ۴. پوسته Pareto حاصل از حل مسئله بهینه‌سازی هدف توسط روش زمان‌بندی و نگاشت ارائه شده



شکل ۵. مجموعه کشف شده از حل مسئله در سه بعد (نقاط مشکی) و مجموعه جواب (نقاط قرمز)

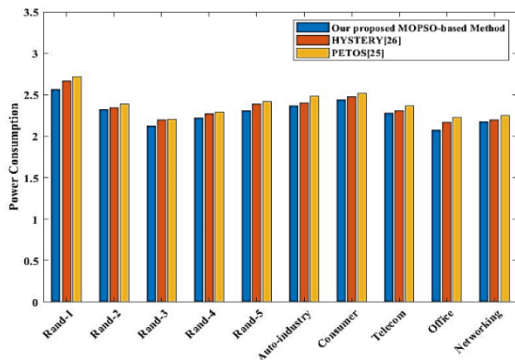
ب) مقایسه روش پیشنهادی با پژوهش‌های پیشین از لحاظ کیفیت جواب‌های مسئله بهینه‌سازی

دسته دوم از آزمایشات با هدف مقایسه عملکرد و کارایی روش زمان‌بندی ارائه شده با سه پژوهش پیشین و مرتبط انجام شده است. روش مکاشفه‌ای [۲۵] با هدف بهینه‌سازی هم‌زمان توان مصرفی، کارایی و دمای سیستم‌های چندپردازنده‌ای ارائه شده است. در این روش جمع وزن‌دار اهداف بهینه‌سازی به‌عنوان

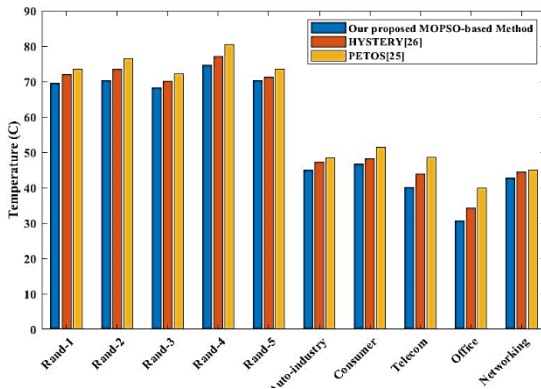
اول به بررسی کارایی روش پیشنهادی در جستجوی موثر فضای طراحی و برقراری توازن بین پارامترهای متخاصم می‌پردازد. بدین منظور پوسته Pareto حاصل از جستجوی فضای طراحی و مجموعه جواب‌های بهینه نهایی ارائه می‌شوند. آزمایشات دسته دوم، روش پیشنهادی را با دو روش مکاشفه‌ای مرتبط پیشین [۲۶، ۱۲] و یک روش فرامکاشفه‌ای مبتنی بر الگوریتم SPEA2 از لحاظ بهبود پارامترهای هدف بهینه‌سازی (طول عمر، توان مصرفی، کارایی و دمای تراشه) مقایسه می‌کنند. معیار مقایسه در آزمایش‌های دسته دوم، بهیگی نقاط خروجی الگوریتم‌ها از حیث طول عمر، توان مصرفی، دمای تراشه و زمان اجرا و همچنین قابلیت هریک در جستجوی فضای جواب است. بدین منظور مقدار هر یک از پارامترهای طراحی در روش‌های مورد نظر به ازای اجرای محک‌های واقعی و تصادفی در بستر شبیه‌سازی استخراج و با یکدیگر مقایسه شده‌اند. در آزمایش دسته سوم سربار زمانی و حافظه‌ای روش پیشنهادی نسبت به پژوهش‌های پیشین مورد مطالعه قرار گرفته است.

آ) بررسی موثر بودن روش زمان‌بندی و نگاشت پیشنهادی در بهینه‌سازی توام طول عمر، توان مصرفی، کارایی و دمای تراشه

در نتیجه حل مسئله بهینه‌سازی چندهدفه مورد نظر در این مقاله، مجموعه جواب خروجی به‌صورت پوسته Pareto استخراج می‌شود. در اولین آزمایش، خروجی موتور جستجوی فضای طراحی ارائه شده در این مقاله در نتیجه حل مسئله برای گراف وظایف تصادفی متشکل از ۴۰ وظیفه در شکل ۴ نشان داده شده است. با توجه به چهارهدفی بودن مسئله بهینه‌سازی مورد نظر و محدود بودن دید انسان به سه بعد، اطلاعات بعد چهارم در این شکل در قالب رنگ نقاط بهینه نمایش داده شده است. در شکل ۴ پارامترهای طول عمر، توان مصرفی و دمای تراشه در سه محور نمایش داده شده و زمان اجرا به‌عنوان معیار کارایی سیستم در تغییرات رنگ نقاط در نظر گرفته شده است. تمامی نقاط نمایش داده شده در این شکل جواب‌های بهینه حل مسئله بهینه‌سازی بوده و انتخاب بین آن‌ها برحسب نیاز کاربر و محیط عملیاتی انجام می‌گیرد. همان‌گونه که در این شکل نشان داده شده، نقاط با دما و توان مصرفی کمتر که دارای بازه‌های زمانی بیکاری بیشتر و سطح پایین‌تر ولتاژ و فرکانس کاری هستند زمان اجرای بیشتری دارند. همچنین نقاط با طول عمر



(پ)



(ت)

شکل ۶. مقایسه روش ارائه شده با [۲۶ و ۲۵] در محک‌های تصادفی و واقعی (آ) زمان اجرا، (ب) طول عمر، (پ) توان مصرفی و (ت) دما

جدول ۳ میانگین بهبود روش پیشنهادی مبتنی بر الگوریتم فرامکاشفای MOPSO در پارامترهای اساسی طراحی مد نظر نسبت به الگوریتم SPEA2 را به ازای محک‌های تصادفی و واقعی با اندازه گراف بین ۳۰ تا ۴۰ وظیفه نشان می‌دهد.

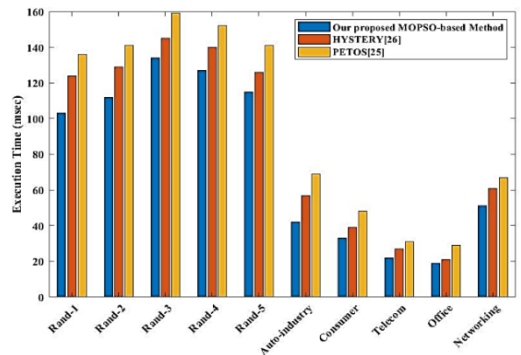
جدول ۳. مقایسه عملکرد روش پیشنهادی با الگوریتم فرامکاشفای SPEA2 در بهبود پارامترهای اساسی طراحی

دما	توان مصرفی	متوسط زمان تا خرابی (نرمال)	زمان اجرا (میلی ثانیه)	روش پیشنهادی
۶۲/۵۳	۲/۴۳	۰/۱۱	۷۶/۳۸	روش پیشنهادی
۶۱/۷۶	۲/۳۹	۰/۱۴	۷۷/۳۶	بهینه‌سازی با الگوریتم SPEA2

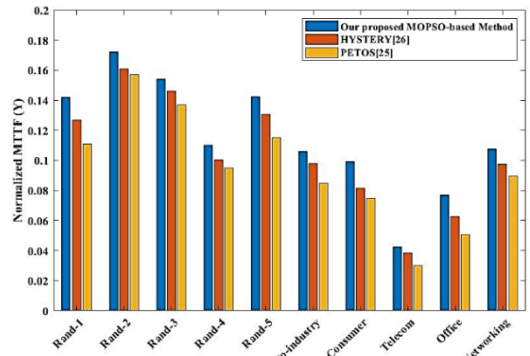
طبق آزمایشات انجام شده و نتایج ارائه شده در شکل ۶ و جدول ۳، روش زمان‌بندی مبتنی بر الگوریتم بهینه‌سازی چندهدفی ازدحام ذرات ارائه شده در این مقاله، پارامترهای طراحی هدف بهینه‌سازی را در حدود ۲۰ درصد به‌طور میانگین نسبت به روش‌های مکاشفای مرتبط پیشین بهبود می‌دهد. لازم به‌ذکر است که روش‌های مکاشفای به‌دلیل سرعت عملکرد و دقت قابل قبول، در بسیاری از کاربردها نسبت به روش‌های

تابع هدف در نظر گرفته شده و با تغییر دوره‌ای وزن‌ها بهینه شده است. از سوی دیگر در [۲۶] روش مکاشفای بر اساس الگوریتم زمان‌بندی مبتنی بر لیست به‌منظور بهینه‌سازی همزمان طول عمر، توان مصرفی، کارایی و دمای تراشه ارائه شده است. در این روش با تبدیل مسئله بهینه‌سازی چندهدفی به حالت تک‌هدفی، زمان‌بندی و نگاشت وظایف با هدف بهبود پارامترهای تعریف شده انجام می‌گیرد. سومین روش مورد مقایسه، بر پایه به‌کارگیری الگوریتم فرامکاشفای و رایج SPEA2 [۴۲] است که در حل مسئله تعریف شده است.

در این بخش به‌منظور مقایسه قابلیت روش ارائه شده با پژوهش‌های پیشین در بهبود اهداف مسئله بهینه‌سازی تعریف شده، بستر شبیه‌سازی یکسانی در نظر گرفته شده و عملکرد روش‌ها برای زمان‌بندی پنج گراف تصادفی و چهار گراف واقعی مستخرج از E3S با یکدیگر مقایسه شده‌اند. شکل ۶ نتیجه این مقایسه را به تفکیک پارامترهای مورد نظر طراحی، محک‌های مورد استفاده و روش‌های در نظر گرفته شده نشان می‌دهد. گراف‌های تصادفی برنامه کاربردی استفاده شده در این آزمایش دارای ۳۰ تا ۴۰ وظیفه هستند.



(آ)



(ب)

پیشنهادی، اصولاً فرض بر آن است که اجرای الگوریتم در زمان طراحی و با بودجه زمان و حافظه کافی صورت می‌گیرد بنابراین روش‌های فرامکاشفه‌ای در این حالت در بیشتر کاربردها به دلیل دقت بیشتر مناسب‌تر هستند. در مقایسه با روش فرامکاشفه‌ای SPEA2، روش پیشنهادی سرعت عملکرد بهتری دارد و همگرایی آن سریع‌تر است. همچنین با توجه به عمگرهای مورد استفاده حافظه مصرفی کمتری را اشغال می‌کند و از حیث دقت جواب‌های استخراج شده نیز مناسب‌تر است.

۶- نتیجه‌گیری

در این مقاله روش زمان‌بندی و نگاشت وظایف مبتنی بر الگوریتم بهینه‌سازی چندهدفی ازدحام ذرات با هدف بهبود طول عمر در سیستم‌های نهفته چندپردازنده‌ای با در نظر داشتن توان مصرفی، کارایی و دمای تراشه ارائه شد. در این روش، به‌منظور مدل‌سازی طول عمر، رابطه سطح سیستم مکانیزم‌های خرابی قابل رخداد در سیستم‌های چندپردازنده‌ای در نظر گرفته شد که جامع‌ترین رویکرد در محاسبه این پارامتر به‌عنوان یکی از مهم‌ترین چالش‌های طراحی سیستم‌های چندپردازنده‌ای روی تراشه ناهمگن است. به‌منظور حل مسئله مطرح شده در این مقاله از الگوریتم فرامکاشفه‌ای MOPSO به‌عنوان یکی از ابزارهای موثر حل مسائل بهینه‌سازی چندهدفی استفاده شده است. روش ارائه شده جواب‌های معتبر زمان‌بندی و نگاشت متنوعی را که به‌صورت تصادفی تولید شده‌اند را در غالب ذرات در نظر گرفته و برحسب تابع هزینه تعریف شده از اهداف مسئله بهینه‌سازی (طول عمر، توان مصرفی، کارایی و دمای تراشه) ارزیابی کرده و نقاط نامغلوب را ذخیره می‌کند. سپس با تعیین سرعت لحظه‌ای و بهینه محلی برای هر ذره، یکی از نقاط مجموعه نامغلوب را بر اساس معیار پراکندگی به‌عنوان بهینه کلی برمی‌گزیند. هر ذره توسط برداری متشکل از سه پارامتر سرعت، بهینه محلی و کلی حرکت داده شده و بدین ترتیب تمامی فضای جواب پوشش می‌شود. مجموعه جواب نهایی پس از تعداد مشخصی تکرار، در قالب پوسته Pareto به کاربر ارائه می‌شود. آزمایشات تجربی انجام شده در این پژوهش، کارایی روش زمان‌بندی و نگاشت ارائه شده را نشان داده و قابلیت‌های آن را تضمین نموده‌اند. به‌عنوان روال آتی، گسترش دادن روش ارائه شده به زمان اجرای سیستم (حالت برخط) با هدف مدیریت رفتار پویای سیستم و همچنین تمرکز دقیق‌تر بر هر یک از مکانیزم‌های خرابی با هدف مدل‌سازی مناسب‌تر طول عمر پیشنهاد می‌شود.

فرامکاشفه‌ای که زمان‌بر هستند ارجحیت دارند و آزمایشات ارائه شده در این بخش نشان‌دهنده عدم کارایی آن‌ها نیست. همچنین روش ارائه شده در مقایسه با روش فرامکاشفه‌ای SPEA2 در استخراج نقاط بهینه، عملکرد مشابه و مناسبی دارد که این مقایسه‌ها و نتیجه‌گیری درباره آن‌ها در آزمایشات بخش بعدی کامل‌تر خواهد شد.

پ) بررسی سربار زمان اجرا و حافظه مصرفی روش پیشنهادی

همان‌گونه که نتایج آزمایشات پیشین نشان دادند، روش‌های فرامکاشفه‌ای به دلیل انعطاف‌پذیری و ایجاد تنوع در فضای جواب، کارایی بهتری در یافتن نقاط بهینه و دوری از بهینه‌های محلی نسبت به روش‌های مکاشفه‌ای و حریم‌دارند. این دقت در ازای صرف هزینه در زمان اجرا و حافظه مصرفی است. جدول ۴، میانگین سربار زمان اجرا و حافظه مصرفی روش پیشنهادی را برای حل مسئله تعریف شده به ازای محک‌های تصادفی و واقعی نشان می‌دهد. همچنین روش پیشنهادی با پژوهش‌های پیشین ارائه شده در [۲۵، ۲۶] و همچنین الگوریتم SPEA2 از لحاظ این سربارها مقایسه شده است.

جدول ۴. مقایسه سربار زمانی و حافظه مصرفی روش پیشنهادی با

پژوهش‌های پیشین

روش پیشنهادی	زمان اجرای الگوریتم (ثانیه)	مصرف حافظه (%)
روش پیشنهادی	۲۷۰۰	۴۹
روش مکاشفه‌ای [۲۵]	۸۵	۱۷
روش مکاشفه‌ای [۲۶]	۴۰	۸
روش مبتنی بر SPEA2	۳۲۰۰	۵۳

نتایج نشان می‌دهند که نوعی مبادله و تقابل در دقت نقاط استخراج شده و سربار زمانی و حافظه مصرفی در روش‌های بهینه‌سازی حین زمان‌بندی و نگاشت وظایف وجود دارد. بدین ترتیب که روش‌های فرامکاشفه‌ای با صرف هزینه زمانی و حافظه مصرفی زیاد، نقاط بیشتر و بهتری در فضای جواب یافته و بهینه‌سازی موثرتری انجام می‌دهند. از سوی دیگر روش مکاشفه‌ای سربار زمان و حافظه بسیار کمی داشته ولی به دلیل سادگی و سرعت نقاط کمتر و بی‌دقت‌تری را استخراج می‌نماید. لازم به ذکر است که تقابل دقت و سربار بر حسب کاربرد قابل تنظیم است و برحسب نوع کاربرد هر یک از این روش‌ها مورد استفاده قرار می‌گیرد. با توجه به برون‌خط بودن روش‌های

- [12] Das, A., et al. "Combined DVFS and mapping exploration for lifetime and soft-error susceptibility improvement in MPSoCs". Design, Automation and Test in Europe Conference, 2014.
- [13] Ma, Y., et al., "Improving System-Level Lifetime Reliability of Multicore Soft Real-Time Systems". IEEE Transactions on VLSI Systems, 25(6), 2017.
- [14] Das, A., Kumar, A., Veeravalli, B., "Reliability and energy-aware mapping and scheduling of multimedia applications on multiprocessors". IEEE Transactions on Parallel and Distributed Systems, 27(3), 2016.
- [15] Zhou, J., et al., "Thermal-aware task scheduling for energy minimization in heterogeneous real-time mpsocs". IEEE Transactions on Computer-Aided, 35(8), 2016.
- [16] Akbari, M., et al., "An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems," Engineering Applications of Artificial Intelligence, 61(1), 2017.
- [17] Coello, C., et al., "Handling multiple objectives with particle swarm optimization," IEEE Transactions on evolutionary computation, 8(1), 2004.
- [18] Wolf, M., High-performance embedded computing: applications in cyber-physical systems and mobile computing, Newnes, 2014.
- [19] Nomura, K., Takashima, Y., Nakamura, Y., "PEVaS: Power and execution-time variation-aware scheduling for MPSoC," New Circuits and Systems Conference, 2016.
- [20] Ammar, M., et al., "On exploiting energy-aware scheduling algorithms for mode-based design space exploration of mpsoc," Parallel, Distributed, and Network-Based Conference, 2016.
- [21] Ananthana, G., Sarangi, S., Balakrishnan, M., "Leakage power aware task assignment algorithms for multicore platforms," VLSI Symposium, 2016.
- [22] Huang, L., Yuan, F., Xu, Q., "Lifetime reliability-aware task allocation for MPSoC platforms," Design, Automation and Test in Europe Conference, 2009.
- [23] Rathore, V., Chaturvedi, V., Srikanthan, T., "Performance constraint-aware task mapping to optimize lifetime reliability of many-core systems," Great Lakes Symposium on VLSI, 2016.
- [24] Huang, L., Xu, Q., "Energy-efficient task allocation and scheduling for multi-mode MPSoCs under lifetime reliability constraint," Design, automation and test in Europe Conference, 2010.
- [25] Sheikh, H., Ahmad, I., "Sixteen heuristics for joint optimization of performance, energy, and temperature in allocating tasks to multi-cores", ACM Transactions on Parallel Computing, 3 (2), 2016.

تعارض منافع

هیچ‌گونه تعارض منافع توسط نویسندگان بیان نشده است.

تشکر و قدردانی

بدین‌وسیله از آقای دکتر حمیدرضا زرنندی که در تعریف موضوع پژوهش و تدوین مقاله راهنمایی‌های قابل توجهی داشتند، خانم دکتر آتنا عبدی که در اجرای پژوهش و تدوین مقاله نقش اصلی را داشته‌اند و همچنین از راهنمایی‌های آقای مهندس محمدرضا امینی در بازنگری و انجام اصلاحات مقاله کمال تشکر و قدردانی را دارم.

مراجع

- [1] Wolf, W., Jerraya, A., Martin, J. "Multiprocessor system-on-chip (MPSoC) technology," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27(10), 2008.
- [2] Ahmed, J., et al., Fuzzy Logic Based Power-Efficient Real-Time Multi-Core System, Springer, 2017.
- [3] Abdallah, A.B., Advanced Multicore Systems-On-Chip. Springer, 2017.
- [4] Wolf, W. "The future of multiprocessor systems-on-chips," Design Automation Conference, 2004.
- [5] Ferrandi, F., et al., "Ant colony heuristic for mapping and scheduling tasks and communications on heterogeneous embedded systems", IEEE Transactions on Computer-Aided Design, 29(6), 2010.
- [6] Scheffer, L., Lavagno, L., Martin, G. EDA for IC system design, verification, and testing, CRC press, 2016.
- [7] Das, A., Kumar, A., Veeravalli, B., A Survey of Lifetime Reliability-Aware System-Level Design Techniques for Embedded Multiprocessor Systems. National University of Singapore, 2014.
- [8] Lee, E.A., Seshia, S.A. Introduction to embedded systems: A cyber-physical systems approach. MIT, 2016.
- [9] Ammar, M., Baklouti, M., Abid, M., "The Performance-Energy Tradeoff in Embedded Systems Design: A Survey of Existing Design Space Exploration Tools and Trends". International Journal of Computer Science and Information Security, 14(5), 2016.
- [10] Silberschatz, A., Galvin, P.B., Gagne, G. Operating System Concepts Essentials. Wiley, 2016.
- [11] Singh, A.K., et al. "Mapping on multi/many-core systems: survey of current and emerging trends". Design Automation Conference, 2013.

- [38] Liu, Y., et al. "Accurate temperature-dependent integrated circuit leakage power estimation is easy", Design, automation and test in Europe Conference, 2007.
- [39] Shah, R., Bell, K., handbook of thermal engineering, CRC Press Boca Raton, 2000.
- [40] Embedded system synthesis benchmark suite (e3s), May 2019, <http://ziyang.eecs.umich.edu/dickrp/e3s/>.
- [41] D. Rhodes, R. Dick, K. Vallerio, Task graphs for free, May 2019, <http://ziyang.eecs.umich.edu/dickrp/tgff>.
- [42] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in International Conference on Parallel Problem Solving from Nature, 2004.
- [26] Abdi, A., Zarandi. H., "HYSTERY: a hybrid scheduling and mapping approach to optimize temperature, energy and lifetime of heterogeneous multiprocessor systems," The Journal of Supercomputing, 74(5), 2018.
- [27] Omara, F., Arafa, M., "Genetic algorithms for task scheduling problem," Foundations of Computational Intelligence, 3(1), 2009.
- [28] Majd, A., et al., "Nomes: Near-optimal metaheuristic scheduling for mpsoes," Computer Architecture and Digital Systems Conference, 2017
- [29] Pilato, C., Design Methodologies for Improving Embedded Systems with Hardware Accelerators. Politecnico Di Milano, 2011.
- [30] Lienig, J., Bruemmer, H., Fundamentals of Electronic Systems Design, Springer, 2017.
- [31] Srinivasan, J., et al. "The case for lifetime reliability-aware microprocessors," ACM SIGARCH Computer Architecture News, 2004.
- [32] Failure Mechanisms and Models for Semiconductor Devices, JEP 122H, 2016.
- [33] Sathyakam, P., Mallick, P., "Towards realisation of mixed carbon nanotube bundles as VLSI interconnects: A review," Nano Communication Networks, 3(3), 2012.
- [34] Srinivasan, J., et al., "Lifetime reliability: Toward an architectural solution," IEEE Micro, 25(3), 2005.
- [35] Franco, J., Kaczer, B., Groeseneken, G., Reliability of high mobility SiGe channel MOSFETs for future CMOS applications, Springer, 2014.
- [36] Zhu, D., Melhem, R., Mossé, D. "The effects of energy management on reliability in real-time embedded systems", Computer Aided Design Conference, 2004.
- [37] Kumar, P., Thiele, L. "Thermally optimal stop-go scheduling of tasks with real-time constraints," Asia and South Pacific Design Automation Conference, 2011.



COPYRIGHTS

© 2021 by the authors. Licensee Iranian Space Research Center of Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) (<https://creativecommons.org/licenses/by/4.0/>)