

## کاهش انرژی مصرفی استاتیک با استفاده از سیستم چند FPGA ناهمگن

محسن کیانی<sup>۱</sup> و عبد الله چاله چاله<sup>۲</sup>

۱- دانشجوی دکتری، دانشکده فنی و مهندسی، گروه کامپیوتر- دانشگاه رازی کرمانشاه- کرمانشاه- ایران

Kiani.mohsen@razi.ac.ir

۲- استادیار، دانشکده فنی و مهندسی، گروه کامپیوتر- دانشگاه رازی کرمانشاه- کرمانشاه- ایران

chalechale@razi.ac.ir

**چکیده:** امروزه انرژی مصرفی در سیستم های مبتنی بر FPGA از پارامترهای مهم به شمار می آید. این پارامتر در برخی کاربردها با منبع محدود انرژی اهمیت بیشتری می یابد. انرژی مصرفی در یک سیستم شامل انرژی مصرفی استاتیک و دینامیک است. به دلیل محدودیت های یک تراشه FPGA در برخی کاربردها، از چند تراشه در کنار هم استفاده می شود. در این مقاله برای کاهش انرژی مصرفی استاتیک، استفاده از معماری ناهمگن پیشنهاد شده است و با استفاده از الگوریتم کلونی مورچه ها، وظایف بلادرنگ در یک سیستم نمونه برای هر دو حالت همگن و ناهمگن زمان بندی شده اند و نتایج هر کدام از نظر انرژی مصرفی، با تخمین از روی تعداد بلاک و زمان هر وظیفه، با هم مقایسه شده اند. برای حالتی که تعداد وظایف در هر برهه زمانی ثابت نیست، سیستم ناهمگن بطور میانگین ۶/۴۴ درصد در مصرف انرژی نسبت به سیستم همگن صرفه جویی داشته است.

**واژه های کلیدی:** انرژی مصرفی، زمان بندی وظایف، سیستم ناهمگن، کلونی مورچه ها، FPGA

### ۱- مقدمه

بازپیکربندی شوند، ممکن شده است. چنین سیستمی را می توان با پردازنده های همه منظوره نیز پیاده سازی نمود اما FPGA دارای مزیت سرعت بالاتر است. در چنین سیستم هایی، یک سری وظایف در هر برهه از زمان وارد سیستم می شوند که باید اجرا شوند. هر وظیفه یک مقدار مشخص فضا نیاز دارد، و برای مدتی اجرا می شود. سیستم می تواند بلادرنگ باشد، بدین معنی که انجام یک سری وظیفه باید در زمان محدودی انجام شود. یک محدودیت در استفاده از FPGA، محدودیت در سرعت پیکربندی آن است. از آنجا که سرعت درگاه پیکربندی آن محدود است، نمی توان بصورت همزمان تعداد زیادی وظیفه به آن فرستاد. یک راه حل برای این مسئله استفاده همزمان از چند FPGA است. یک سیستم معروف از این نوع سیستم BEE2 [۱] است. چند FPGA موجود در سیستم می توانند همسان باشند که سیستم همگن نامیده می شود، یا دارای قابلیت های متفاوت باشند، که به آن سیستم نامتقارن یا ناهمگن می گویند. با

در سال های اخیر استفاده از FPGA ها به خاطر قابلیت های مناسب آنها از نظر حجم، کارایی انعطاف پذیری و قیمت بیش از پیش مورد توجه قرار گرفته است. یک قابلیت مهم آنها، که باعث انعطاف بیش از پیش آنها شده است، قابلیت بازپیکربندی بصورت کلی یا جزئی است. با قابلیت بازپیکربندی جزئی می توان قسمتی از سخت افزار را بازپیکربندی نمود. بنابراین استفاده از آنها برای ایجاد سیستم هایی که بتوانند بصورت دینامیک

<sup>۱</sup> تاریخ ارسال مقاله : ۱۳۹۲/۱۲/۱۷

تاریخ پذیرش مقاله : ۱۳۹۳/۰۱/۱۸

نام نویسنده مسئول : محسن کیانی

نشانی نویسنده مسئول : ایران - کرمانشاه - دانشگاه رازی کرمانشاه - دانشکده فنی و مهندسی - گروه کامپیوتر

است. در این مقاله به بررسی زمان بندی در یک سیستم FPGA متشکل از چند FPGA ناهمگن پرداخته شده است. در چنین سیستمی، FPGA ها از نظر حجم، سرعت و دیگر قابلیت ها مشابه یکدیگر نیستند. دو حالت کلی متصور است. حالت اول اینکه قابلیت محاسباتی آنها یکسان باشد بدین معنی که تمامی وظایف در یک فرکانس یکسان روی همه آنها بصورت یکسان اجرا شود. در حالت دوم، ممکن است هر کدام از آنها برای یک سری از وظایف بهینه شده باشند و بنابراین هر وظیفه در همان سرعت بطور متفاوتی روی آنها اجرا شود. در حالت دوم، می توان هم از نظر سرعت و هم از نظر انرژی به بهبود بالاتری نسبت به حالت همگن دست یافت.

در این مقاله انرژی مصرفی دو سیستم همگن و ناهمگن، با استفاده از زمان بندی به کمک الگوریتم کلونی مورچه ها مورد بحث قرار گرفته است. تابحال در هیچ مقاله ای به بحث انرژی مصرفی در یک سیستم چند FPGA ناهمگن پرداخته نشده است. نتایج شبیه سازی نشان می دهد با استفاده از سیستم ناهمگن، در اکثر حالات انرژی استاتیک مصرفی سیستم کاهش می یابد.

سازمان این مقاله بدین صورت است که در بخش دوم، مسئله استفاده از سیستم های ناهمگن تشریح شده است. در بخش ۳، مدل ساده توان و انرژی استفاده شده در شبیه سازی معرفی شده است. بخش ۴ به معرفی الگوریتم کلونی مورچه ها که در این مقاله برای زمان بندی استفاده شده اختصاص یافته است. در بخش ۵ نحوه انطباق الگوریتم کلونی مورچه ها برای زمان بندی ارائه شده است. بخش ۶ به نحوه پیاده سازی الگوریتم زمان بندی و ارائه نتایج اختصاص یافته است و این نتایج در بخش ۷ مورد بحث قرار گرفته اند. در نهایت در بخش ۸ نتیجه گیری به عمل آمده است.

## ۲- تشریح مسئله

در این مقاله هدف اصلی مقایسه انرژی مصرفی در دو سیستم دارای چند FPGA همگن و ناهمگن است. برای مقایسه این دو سیستم، در ادامه مثالی آورده شده است. دو سیستم FPGA مفروض است. سیستم اول متشکل

وجود چند تراشه FPGA و رسیدن تعدادی وظیفه، یک مسئله مهم نگاشت یا زمان بندی وظایف برای اجرا روی FPGA ها است. اگر تعداد وظایف زیاد باشد، این مسئله دشوار خواهد بود و یک راه حل آن استفاده از الگوریتم های تکاملی مانند کلونی مورچه هاست.

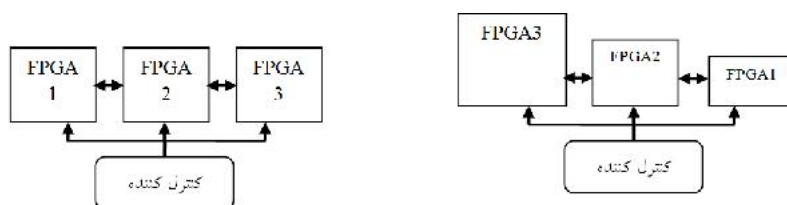
پژوهش های در حوزه زمان بندی وظایف جهت اجرا روی FPGA انجام شده است. در اکثر پژوهش ها، مسئله بصورت یک گراف بدون حلقه جهت دار (DAG) در نظر گرفته شده و برای آن زمان بندی انجام شده است. در برخی از آنها هدف تنها افزایش سرعت و در برخی پارامترهایی مانند توان یا انرژی مد نظر بوده است. در [۲] روشی برای زمان بندی ارائه شده که سربار بازپیکربندی را نیز در نظر می گیرد. در بسیاری از مقالات از این سربار صرف نظر شده است. در مقاله حاضر، این سربار در نظر گرفته شده است. در [۳] نیز یک روش زمان بندی ارائه شده است که در آن سعی شده سربار زمان بندی به حداقل برسد.

توان و انرژی مصرفی یکی از چالش های سیستم های مبتنی بر FPGA است. بویژه در کاربردهایی که انرژی از طریق باتری تامین می شود، مانند گره ها در شبکه های حسگر بیسیم، این مسئله اهمیت بیشتری می یابد. برای کاهش توان مصرفی در FPGA، روش هایی ارائه شده است. در [۴] روشی برای کاهش انرژی مصرفی در FPGA های دارای چند زمینه پرداخته شده است. چنین FPGA هایی دارای چند زمینه پیکربندی اند و بنابراین می توانند وظایف را بصورت مستقل و همزمان گرفته و اجرا کنند. فرض شده است که FPGA می تواند در سطوح مختلف سرعت کار کند و نیز با تکنیک Clock-gating آنرا به حالت Idle برد. T وظیفه که همگی در زمان ۰ می رسند و برای آنها یک سررسید<sup>۱</sup> برابر D وجود دارد. برای کاهش توان مصرفی از مقیاس بندی ولتاژ بصورت دینامیک<sup>۲</sup> استفاده شده است.

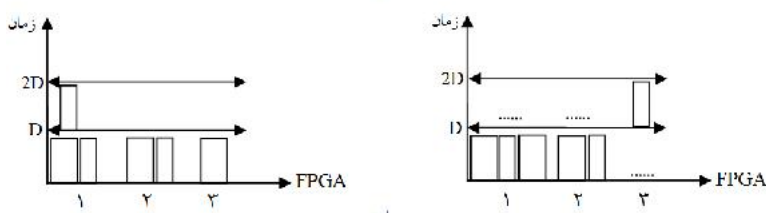
در [۵] نیز یک الگوریتم زمان بندی برای کاهش انرژی مصرفی در یک سیستم دارای چهار FPGA همگن ارائه شده است که این کار با تغییر فرکانس انجام شده است. برای زمان بندی از الگوریتم کلونی مورچه ها استفاده شده

باید روشن و آنرا اجرا نماید. اما در حالت ناهمگن کافی است FPGA کوچک روشن شود و طبیعتاً انرژی مصرفی در سیستم ناهمگن (استاتیک) کاهش می یابد (شکل ۱-ب). یک حالت دیگر برای سیستم ناهمگن متصور است که در آن هر FPGA برای پردازش تعدادی از وظایف بهینه است. اما در این مقاله همه FPGA ها تنها از نظر حجم ترانزیستور متفاوت اند.

از سه عدد FPGA یکسان یا همگن. بدین معنی که این FPGA ها همگی از نظر تکنولوژی ساخت، حجم ترانزیستور و سرعت یکسان هستند. سیستم دوم شامل سه FPGA با تکنولوژی ساخت و سرعت یکسان اما حجم ترانزیستور متفاوت است. تصویر این سیستم ها در شکل ۱-الف نشان داده شده است. اگر تعداد وظایفی که به سیستم وارد می شوند نیازمند حجم کمی بلاک منطقی باشند، در حالت سیستم همگن یک FPGA نسبتاً بزرگ



الف



ب

شکل (۱): الف) سیستم همگن در مقابل ناهمگن، ب) مثالی از زمان بندی وظایف در دو سیستم

## ۲-۱- مدل توان و انرژی در FPGA

توان مصرفی در یک FPGA از دو قسمت توان مصرفی استاتیک ناشی از جریان نشتی ترانزیستور و توان مصرفی دینامیک تشکیل شده است. پژوهش های بسیار زیادی در زمینه کاهش مصرف توان در FPGA و مدارات برپایه تکنولوژی CMOS انجام شده است که برخی در سطوح پایین در سطح ساخت هستند که عمدتاً شامل کاهش ولتاژ تغذیه، کوچک کردن ترانزیستور و ... می شوند. در سطح مدار نیز تمرکز بر روی کاهش ولتاژ تغذیه، ظرفیت خازنی و فرکانس است [۶-۸]. در سطح بالاتر، یعنی RTL، می توان به روش های برپایه Clock gating اشاره کرد [۹]. در سطح سیستم و الگوریتم، مسئله انتخاب الگوریتم مناسب و نیز استفاده از مدیریت توان در هنگام کار با روشن و

در این مقاله وظایف بصورت بلادرنگ در نظر گرفته شده اند و فرض شده است که در هر برهه از زمان تعدادی وظیفه وارد سیستم شده و باید اجرا شوند. تعداد این وظایف دارای یک حداقل و یک حداکثر است و وابستگی بین آنها وجود ندارد. هر وظیفه دارای یک زمان پیکربندی، یک زمان اجرا و تعداد بلاک مورد نیاز است. در شبیه سازی انجام شده از آنجا که یافتن زمان بندی بویژه برای تعداد وظیفه زیاد دشوار است، از الگوریتم کلونی مورچه ها استفاده شده است. دو حالت همگن و ناهمگن با یکدیگر مقایسه شده اند.

در ادامه در ابتدا مدل توان و انرژی مورد استفاده معرفی و سپس به معرفی و نحوه انطباق الگوریتم کلونی مورچه ها برای حل مسئله زمان بندی پرداخته شده است.

که این نسبت برای خانواده Virtex II با تکنولوژی nm ۱۵۰، ۵ تا ۲۰ درصد است. البته در این مقاله تراشه ها nm ۶۵ در نظر گرفته شده اند و این نسبت می تواند بالاتر هم باشد. همچنین برای حالت همگن چهار FPGA شرکت Xilinx به شماره XC5VLX155 در نظر گرفته شده است که دارای ۱۲۱۶۰ بلاک منطقی قابل پیکربندی<sup>۲</sup> یا بطور ساده بلاک است. کل سیستم دارای چهار عدد از این تراشه است که معادل ۴۸۶۴۰ بلاک می شود. از سوی دیگر برای سیستم ناهمگن نیز چهار تراشه به شماره های XC5VLX50، XC5VLX85، XC5VLX330 و XC5VLX155 به ترتیب با ۳۶۰۰، ۶۴۸۰، ۱۲۱۶۰ و ۲۵۹۲۰ بلاک، جمعاً ۴۸۱۶۰ بلاک در نظر گرفته شده است [۱۵] که در این مقاله به ترتیب تراشه اول، تراشه دوم، تراشه سوم و تراشه چهارم نامیده شده اند. همگی این تراشه ها جزء خانواده Virtex-5 هستند. زمانی که کار یک تراشه تمام شد به حالت غیرفعال می رود و بنابراین توان مصرفی آن به حدود صفر می رسد [۱۰]. اما در حالت فعال، که در حال انجام کار است، هم در حال مصرف توان دینامیک است و هم استاتیک. نکته حائز اهمیت اینست که در این مقاله اعدادی که در جدول ۱ نمایش نشان داده شده اند یک عدد (جمع انرژی دینامیک ضرب در ۱۰، به علاوه انرژی استاتیک) نشان دهنده تخمینی از انرژی مصرفی اند که و می توان این مقادیر را در انرژی مصرفی هر بلاک ضرب کرد تا تخمین مقدار واقعی حاصل شود.

لازم به ذکر است که اندازه و تعداد هر FPGA باید با توجه به ملزومات کاربرد یا حوزه کاربردی بصورت مناسب تعیین شود. همچنین، با توجه به یکسان بودن تکنولوژی FPGA های در نظر گرفته شده در مقاله و مجهز بودن تمامی آنها به رابط پیکربندی یکسان، تفاوت چندانی از نظر سرعت بین دو سیستم همگن و ناهمگن ایجاد نمی شود. هرچند، ممکن است زمان پیکربندی کمی متفاوت باشد اما در مقاله وظایف بلادرنگ بوده و با الگوریتم کلونی مورچه ها که در بخش ۲-۳ تشریح شده است زمان بندی مناسب بصورتی محاسبه شده است که در هر مرحله ضرب العجل مورد نظر نقض نشود.

خاموش کردن یا کم و زیاد کردن فرکانس است. برای مثال در برخی تراشه ها می توان با بردن تراشه به حالت معلق، ۹۹ درصد توان مصرفی استاتیک آنرا کم کرد [۱۰]. در [۱۱] مروری بر کارهای انجام شده در زمینه کاهش توان در سطح سیستم انجام شده است.

برخی از پژوهش های انجام شده، هدف کاهش توان مصرفی دینامیک بوده [۱۲] و برخی دیگر توان استاتیک [۱۳ و ۱۴]. گرچه در گذشته به توان مصرفی دینامیک به عنوان توان غالب نگریسته می شد، اما با کاهش اندازه ترانزیستور در FPGA های جدید که باعث افزایش جریان نشتی و توان استاتیک شده است، نسبت توان استاتیک به توان دینامیک رو به فزونی گذاشته و بنابراین توان مصرفی استاتیک یا توان نشتی نیز بیش از پیش مورد توجه قرار گرفته است.

توان مصرفی در یک FPGA را می توان با معادله (۱) مدل کرد.

$$P = P_S + P_D \quad (1)$$

که در آن  $P$  توان مصرفی کل،  $P_S$  توان مصرفی استاتیک و  $P_D$  توان مصرفی دینامیک است. توان مصرفی دینامیک با فرکانس و مربع ولتاژ کاری و نیز بار خازنی در تراشه ارتباط دارد. به همین صورت می توان انرژی مصرفی را از جمع دو انرژی مصرفی استاتیک و دینامیک بدست آورد (معادله (۲)). با داشتن زمان و ضرب آن در مقدار توان، مقدار انرژی حاصل می شود. در این مقاله برای مقایسه مصرف انرژی در دو حالت سیستم FPGA همگن و ناهمگن، با توجه به وظایف و تعداد بلاک مورد نیاز انرژی دینامیک، و با توجه به اندازه تراشه و فعال بودن یا نبودن آن، انرژی استاتیک تخمین زده می شود. از برخی سربارها مانند سربار تغییر از حالت غیرفعال به فعال در هر دو سیستم صرف نظر شده است.

$$E = E_S + E_D \quad (2)$$

در این مسئله فرض شده است که نسبت توان استاتیک به دینامیک یک به ده است. در [۱۲] نشان داده شده است

## ۲-۲- روش Power gating

روش Power gating (قطع متناوب توان) موثرترین روش شناخته شده برای کاهش مصرف توان استاتیک است. در این روش، واحدهای عملیاتی استفاده نشده به حالتی با غیرفعال با مصرف انرژی پایین برده می شوند. این کار با یک مدار کنترلر به نام کنترل کننده خواب، شبکه توزیع سیگنال خواب و ترانزیستور خواب انجام می شود. از جمله معایب این روش این است که مدار استفاده شده انرژی مصرف می کند. دو نوع Power gating وجود دارد. درشت دانه [۱۶] و ریز دانه [۱۷]. در حالت درشت دانه از یک کنترلر خواب برای تعداد زیادی LUT استفاده می شود تا سربار توانی و فضایی اشغالی آن کم باشد اما در این روش اگر تنها یک LUT مورد کنترل توسط کنترلر خواب، فعال باشد، نمی توان هیچ کدام از دیگر LUT های تحت کنترل را با همان ترانزیستور خواب به حالت خواب برد. مشکل دیگر آنها توان مصرفی دینامیک بالا برای توزیع سیگنال خواب است. در روش ریزدانه هر LUT دارای ترانزیستور و کنترل کننده خواب مختص خود است و اگر LUT ای غیرفعال بود می توان آنرا به سرعت به حالت خواب برد. در این روش توان استاتیک کمتری مصرف می شود. عیب این روش بالا بودن فضای اشغالی کنترل کننده های خواب و توان دینامیک مصرفی آنهاست.

در روش ارایه شده در مقاله، هدف استفاده بهینه سازی توان در سطح بالاتری یعنی تراشه انجام می شود که هزینه کمتری دارد. از سوی دیگر می توان روش های Power gating را متعادل با روش ارایه شده در این مقاله دانست و می توان در هر زمان، در تراشه های فعال از این روش ها برای خاموش کردن واحدهای بلااستفاده استفاده کرد تا توان مصرفی استاتیک کاهش بیشتری داشته باشد.

## ۲-۳- الگوریتم کلونی مورچه ها

Caro و Dorigo [۱۸] الگوریتم کلونی مورچه ها را به عنوان یک روش اکتشافی بهینه سازی و با الهام از الگوی زیست شناسی مورچه ارایه کردند. این الگوریتم را می توان برای کاربردهای مختلف بهینه سازی از جمله مسئله مرد دوره گرد استفاده نمود. در این روش، تعدادی مورچه

مصنوعی ایجاد و هر کدام به دنبال مقدار بهینه جستجو را آغاز می کنند. در نهایت پس از اینکه همه مورچه ها به اصطلاح یک دو کامل زدند، یا به عبارتی پس از چند تصمیم گیری محلی یک جواب تولید نمودند، مورچه با بهترین مقدار انتخاب می شود، ماتریسی به نام ماتریس فرومون که در محاسبات استفاده می شود به روز می شود و این فرایند تکرار می شود. بعنوان مثال در مسئله مرد دوره گرد، هر مورچه به تصادف شهری را انتخاب می نماید، سپس در مورد انتخاب شهر بعدی تصمیم گیری می کند و در نهایت پس از گذر از تمامی شهرها به شهر اول باز می گردد تا یک دور کامل شود. یکی از مسائل مهم، نحوه تصمیم گیری است. یک رابطه برای این کار وجود دارد که از دو قسمت فرومون، و پارامتر تجربی تشکیل شده است. در مسئله مرد دوره گرد، برای انتخاب شهر بعدی (گذر از شهر فعلی به شهر بعدی) برای هر شهر ممکن با این رابطه یک احتمال محاسبه و سپس، شهر با بالاترین احتمال انتخاب می شود (با درجه ای از تصادف). در محاسبه احتمال، یک پارامتر تاثیر گذار، فرومون موجود است. فرومون بالاتر نشان می دهد که در پاسخ های بهینه قبلی از این شهر به شهر مورد نظر گذر شده است. پارامتر تجربی دیگر بسته به کاربرد انتخاب می شود. در مسئله مرد دوره گرد معمولاً معکوس فاصله بین دو شهر در نظر گرفته می شود. پس از عبور از تمامی شهرها، فاصله طی شده توسط هر مورچه برای عبور از تمامی شهرها محاسبه و مورچه با کمترین فاصله بعنوان بهترین مورچه در تکرار انتخاب می شود. سپس ماتریس فرومون به روز می شود. به روز رسانی شامل دو مرحله است. مرحله اول تبخیر تمامی ماتریس است. یعنی تمامی درایه های این ماتریس، با یک ضریبی پس از هر تکرار کم می شوند. سپس بسته به نوع الگوریتم درایه هایی از ماتریس فرومون با ضریبی افزایش می یابند. لازم به ذکر است که الگوریتم های مختلف دیگری مبتنی بر روش اولیه ارایه شده توسط Dorigo و همکارانش ارایه شده است. در روش اولیه پیشنهادی، تمامی مورچه ها ماتریس فرومون را با نسبت معکوس فاصله خود

یک دور، با توجه به زمان فعال بودن هر FPGA، توان استاتیک به آن اضافه می شود. در نهایت مورچه با کمترین میزان انرژی انتخاب می شود و ماتریس فرامون تخییر و تنها توسط بهترین جواب به روز می شود. این فرایند به تعداد تکرار تعیین شده انجام می شود. سپس به ازای همان وظایف تولید شده بصورت تصادفی، زمان بندی برای سیستم همگن نیز اجرا می شود و نتایج آنها برای مقایسه قابل مشاهده است.

به روز می کند. بنابراین هرچه مسیر طی شده بهتر باشد، ضریب افزایش فرامون آن بالاتر است. در یک روش دیگر، تنها بهترین جواب تولیدی تاکنون ماتریس رابه روز می کند یا به روز رسانی می تواند توسط تعدادی از مورچه ها با کیفیت جواب بالاتر انجام شود. در واقع یکی از اصلی ترین تفاوت ها در روش های مختلف برگرفته از الگوریتم کلونی مورچه ها، نحوه به روز رسانی ماتریس است.

### ۳- پیاده سازی

مسئله زمان بندی شبیه به مسئله مرد دوره گرد است. تعداد  $n$  وظیفه وارد سیستم می شوند و هر وظیفه باید روی یکی از  $m$  تراشه FPGA موجود نگاشت یابد. برای انطباق الگوریتم کلونی مورچه ها برای زمان بندی وظایف در FPGA، برابر با تعداد وظایف تولیدی، مورچه ایجاد می شود و هر مورچه بطور مستقل یک زمان بندی ایجاد می کند. بدین معنی که وظایف را روی FPGA نگاشت می دهد و این کار را تکرار می کند تا وظایف تمام شوند. بنابراین در بخش تصمیم گیری، مورچه باید تصمیم بگیرد که وظیفه  $i$  روی کدام یک از  $m$  عدد FPGA موجود نگاشت یابد و این کار را تا اتمام تمامی وظایف تکرار کند. برای تصمیم گیری در حالت سیستم ناهمگن، از دو پارامتر فرامون، مانند حالت کلی، و یک پارامتر تجربی استفاده شده است. این پارامتر تجربی بسیار مهم و تاثیرگذار بوده و یافتن یک پارامتر مناسب بسیار دشوار است. از آنجا که بلاک های مورد نیاز تمامی وظایف و نیز تعداد بلاک موجود در هر FPGA مشخص است، جمع تمامی بلاک های مورد نیاز برای اجرای کل وظایف محاسبه و با توجه به آن با یک سری مقایسه، پارامتر تجربی تنظیم شده است. بدین صورت که اگر تعداد بلاک های مورد نیاز کوچکتر از بلاک های موجود در یک یا چند FPGA بود، به آنها پارامتر بزرگتری اختصاص می یابد. این مسئله در بخش های پیش رو بیشتر تشریح شده است.

پس از هر مرحله، با توجه به اختصاص وظیفه، پارامتر انرژی آن مورچه به روز می شود. در نهایت پس از انجام

#### ۳-۱- ارایه نتایج

در این قسمت به ارایه نتایج حاصل از شبیه سازی پرداخته شده است. شبیه سازی با استفاده از زبان C++ انجام شده است. همانطور که اشاره شد، در کد نوشته شده، در ابتدا تعدادی وظیفه در یک محدوده، بصورت تصادفی تولید می شود. حداقل عدد ۱ و حداکثر عدد ۲۰ برای تعداد وظایف در نظر گرفته شده است. همچنین، سررسید برابر حداکثر ۱۰ ثانیه در نظر گرفته شده است. حداکثر زمان هر وظیفه نیز برابر ۱۰ ثانیه در نظر گرفته شده است. تعداد بلاک نیز بین ۱۰۰ تا ۲۲۰۰ در نظر گرفته شده است.

به هر وظیفه بصورت تصادفی یک زمان (اجرا+پیکربندی) و تعدادی بلاک مورد نیاز اختصاص می یابد. از آنجا که بیشینه سرعت ICAP برابر با ۳/۲ Gbps است، طبق اطلاعات داده شده در [۱۹]، زمان بازپیکربندی که نسبت به تعداد بلاک خطی است، برای هر بلاک ۱ میکروثانیه در نظر گرفته شده است. در زمان بندی، وظایف بصورت سریال ارسال می شوند و بنابراین زمان پیکربندی آنها با هم جمع می شود. این مسئله در شکل ۲ برای سه وظیفه نشان داده شده است. وظایف تولید شده، سپس در گام اول با الگوریتم کلونی مورچه ها برای سیستم ناهمگن زمان بندی می شوند (با ۲۰۰ تکرار). سپس وظایف تولید شده، برای سیستم همگن زمان بندی می شوند (با همان تعداد تکرار). در نهایت نتایج چاپ می شوند.

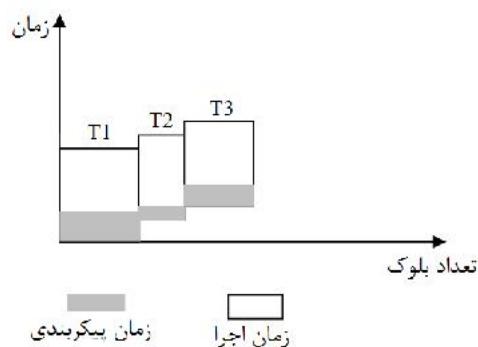
سیستم ناهمگن نیز تراشه با همان حجم فعال می شود. در صورتی که تعداد بلاک مورد نیاز دو برابر حجم یک تراشه FPGA سیستم همگن باشد، در این سیستم دو تراشه فعال می شوند. در سیستم ناهمگن بزرگترین تراشه فعال می شود که حجم آن تفاوت بسیار کمی با جمع دو تراشه همگن دارد و بنابراین اگر از سربارها صرف نظر شود، سیستم همگن اندکی کمتر انرژی مصرف می کند. در صورتی که حجم مورد نیاز سه برابر یک تراشه همگن باشد نیز شرایط به همین صورت است (تراشه سوم و چهارم در ناهمگن فعال می شوند). این حالات کم پیش می آیند و بنابراین در اکثر اوقات، و بطور میانگین سیستم ناهمگن انرژی مصرفی کمتری دارد.

لازم به ذکر است که از نظر پیچیدگی، زمان بندی سیستم ناهمگن بطور میانگین پیچیدگی بیشتری دارد. با سنجش زمان اجرای برنامه مشخص شد که اختلاف زمان اجرای دو سیستم اندک است.

در جدول ۱، نتایج برای ۱۰ بار اجرا با تعدادی وظیفه تصادفی ارائه شده است. ستون آخر نشان دهنده درصد کاهش انرژی سیستم ناهمگن نسبت به همگن است. سپس به ازای تعداد وظایف ثابت، اما زمان های متغیر، نتایج ارائه شده است.

جدول (۱): درصد کاهش انرژی سیستم همگن نسبت به ناهمگن برای چند بار اجرا و تولید تصادفی وظایف

ردیف	تعداد وظیفه	تعداد کل بلاک ها	انرژی همگن	انرژی ناهمگن	% کاهش انرژی
۱	۲	۳۰۰۷	۶۷۲۸	۵۲۰۳	۲۲/۶۶
۲	۳	۴۶۷۳	۱۱۳۱۸	۹۹۳۵	۱۲/۲۲
۳	۵	۶۴۳۳	۱۵۶۲۳	۱۴۲۴۸	۸/۸
۴	۷	۷۶۲۱	۱۵۰۹۷	۱۳۶۲۵	۹/۷۵
۵	۱۰	۱۲۳۹۱	۲۰۲۲۸	۱۸۵۵۹	۸/۲۵
۶	۱۱	۱۶۰۶۱	۳۳۱۴۹	۳۱۲۹۲	۵/۶
۷	۱۳	۱۲۰۳۳	۲۶۴۲۳	۲۶۴۲۳	۰
۸	۱۴	۲۰۲۴۳	۳۰۹۰۷	۲۹۶۹۴	۳/۹۲
۹	۱۷	۱۹۰۱۱	۳۶۸۷۷	۳۵۸۱۳	۲/۸۸
۱۰	۱۹	۲۲۷۴۲	۳۹۳۰۹	۴۰۴۹۵	-۳
	میانگین	...	...	...	۷/۱



شکل (۲): زمان پیکربندی و زمان اجرای در وظایف

در زمان تصمیم گیری برای انتخاب تراشه، آنچه که در نظر گرفته می شود، میزان فرمون، و تعداد کل بلاک های مورد نیاز و یک عدد تصادفی است. تعداد کل بلاک ها با ظرفیت هر تراشه و تمام ترکیب های جمع آنها مقایسه و مناسب هر کدام که بود، به آن تراشه یا تراشه ها اختصاص می یابد. برای مثال در سیستم ناهمگن اگر تمامی وظایف نیاز به ۱۵۰۰۰ بلاک دارند، به دو تراشه اول و سوم پارامتر تجربی بزرگتری اختصاص می یابد. در این حالت برای زمان بندی وظایف در سیستم همگن نیاز به دو تراشه است. در سیستم ناهمگن تراشه های شماره اول و سوم انتخاب می شوند بنابراین انرژی مصرفی استاتیک در این سیستم کمتر خواهد بود چرا که سیستم همگن حجم بلوک بیشتری فعال شده در حالی که همگی آنها استفاده نخواهند شد. در سیستم ناهمگن پیشنهادی در اینجا، با چهار تراشه، ۱۶ حالت یا آستانه وجود دارد. این محدوده ها عبارتند از ۱ تا ۳۶۰۰۱، ۳۶۰۰۱ تا ۶۴۸۰۱، ۶۴۸۰۱ تا ۱۰۰۸۰۱ و ۱۰۰۸۰۱ تا ۱۲۱۶۰۱. اما در سیستم همگن چهار محدوده وجود دارد. در محدوده اول سیستم همگن که ۱ تا ۱۲۱۶۰ است، در سیستم ناهمگن، چهار محدوده ذکر شده در بالا قرار می گیرند که در سه محدوده اول سیستم ناهمگن انرژی استاتیک کمتری مصرف می کند و در چهارمین محدوده، انرژی مصرفی دو سیستم یکسان است. در برخی حالات، بسته به تعداد بلاک سیستم همگن انرژی استاتیک کمتری مصرفی می کند. یک نمونه آن ۲۴۰۰۰ بلاک است. در ادامه چند حالت خاص بررسی می شود. در صورتی که تعداد بلاک مورد نیاز دقیقاً برابر با اندازه یک تراشه FPGA سیستم همگن باشد، تنها آن تراشه فعال می شود و در

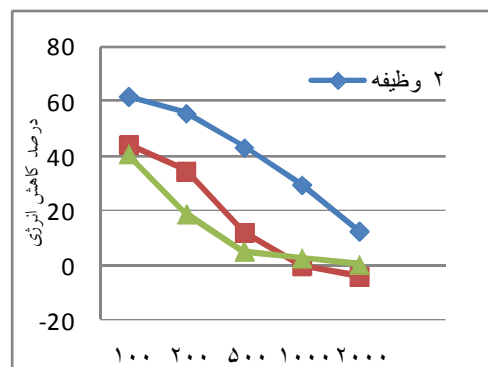
سیستم همگن و ناهمگن، چنین گراف هایی بصورت تصادفی ایجاد و زمان بندی شده اند. تعداد مراحل گراف ۲۰ در نظر گرفته شده است و در هر مرحله تعداد ۱ تا ۲۰ وظیفه بصورت تصادفی ایجاد و زمان بندی شده است. پس از اجرای آنها، انرژی مصرفی در دو حالت همگن و ناهمگن مقایسه شده اند. برای اطمینان از قابل استناد بودن و همگرایی نتایج، شبیه سازی به تعداد ۱۰۰۰ بار تکرار شده است. انرژی مصرفی هر تکرار (یک گراف با ۲۰ سطح و هر سطح بین ۱ تا ۲۰ وظیفه) محاسبه و در نهایت از تمامی آنها میانگین گیری شده است. نتیجه شبیه سازی کاهش ۶/۴۴ درصدی در انرژی مصرفی بوده است.

### ۳-۳- بررسی نتایج

در اکثر حالات سیستم ناهمگن دارای انرژی مصرفی کمتری است. اگر تعداد وظایف کم باشند، تنها تراشه کوچک فعال شده و وظایف را انجام می دهد. اما در سیستم همگن، تراشه نسبتاً بزرگ باید روشن شود و انرژی مصرفی استاتیک آن بالاتر است. اگر وظایف بطوری باشند ظرفیت مورد نیازشان کمی بیشتر از ظرفیت یک تراشه در سیستم همگن باشد، نیاز است تا دو تراشه فعال شوند. اما در حالت ناهمگن، تراشه کوچک و یک تراشه دیگر فعال می شود و بنابراین انرژی مصرفی کمتر خواهد بود. از آنجا که ترکیب ظرفیت های ایجاد شده در سیستم همگن تنها چهار حالت است، اما این ترکیب برای حالت ناهمگن ۱۶ حالت است، بنابراین توانایی این سیستم در سازگاری با تعداد وظایف بالاتر است و این مسئله کاهش انرژی را به دنبال دارد.

برای مثال در جدول ۱، در سطر اول از آنجا که تعداد بلاک کمی مورد نیاز است، در سیستم ناهمگن تراشه کوچک فعال می شود و بنابراین کاهش مصرف قابل توجه است. در سطرها دوم و سوم تراشه دوم در سیستم ناهمگن و یکی از تراشه ها در سیستم همگن فعال می شود و باز کاهش نسبتاً زیاد است. در سطر پنجم، بلاک های مورد نیاز کمی از حجم یک تراشه همگن بیشتر است و بنابراین دو تراشه فعال می شوند اما در سیستم ناهمگن تراشه اول و سوم فعال می شوند. اگر سربار تغییر حالت از غیرفعال به

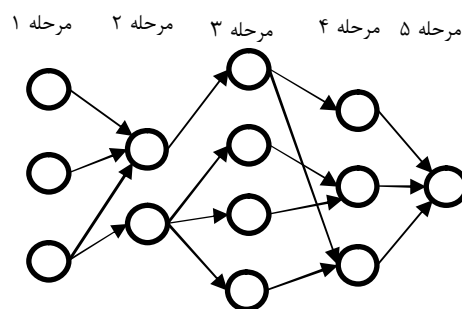
در شکل ۳، به ازای تعداد وظایف ثابت برابر ۲، ۱۲ و ۲۰، کاهش انرژی سیستم ناهمگن نسبت به همگن برای تعداد بلاک ثابت ۱۰۰، ۲۰۰، ۵۰۰، ۱۰۰۰ و ۲۰۰۰ بلاک، (در تمامی وظایف ثابت و یکسان) نتایج ارایه شده است.



شکل (۳): نتایج برای تعداد بلوک و وظیفه ثابت

### ۳-۲- زمان بندی DAG

در مسائل زمان بندی، معمولاً وظایف بصورت یک DAG<sup>۲</sup> برای وظایف ایجاد می شود که دارای تعدادی گره و تعدادی لبه است و هر گره مبین یک وظیفه بوده و لبه ها نشان دهنده وابستگی بین وظایف و زمان اجرای هر وظیفه اند. سپس با توجه به این گراف، در هر مرحله، تعدادی وظیفه موجود است که با یکدیگر وابستگی نداشته و تمام وظایفی که بدانها وابسته اند اجرا شده اند. این وظایف می توانند بصورت موازی اجرا شوند.



شکل (۴): یک DAG نمونه با ۵ مرحله و تعدادی وظیفه در هر مرحله، برای سادگی زمان وظایف نشان داده نشده است

در آخرین سناریو برای مقایسه انرژی مصرفی بین دو



کاهش مصرف انرژی سیستم ناهمگن نسبت به همگن اثر دارد. بنابراین اگر این پارامترها یا محدوده آنها از قبل مشخص هستند، می توان سیستم مناسب را انتخاب کرد که در برخی حالات سیستم همگن مناسب تر است. اما اگر مانند بسیاری از کاربردها تعداد وظایف در هر برهه از زمان متفاوت با زمان های قبلی و بعدی است، سیستم ناهمگن می تواند انرژی مصرفی را پایین بیاورد که این نتیجه شبیه سازی DAG با تکرار بسیار بالا اثباتی بر این مدعا است. میزان این کاهش نیز به پارامترهای برشمرده بستگی دارد.

#### ۴- نتیجه گیری

در این مقاله جهت کاهش مصرف انرژی استاتیک، پیشنهاد شد بجای استفاده از چند FPGA همگن، از چند FPGA ناهمگن استفاده شود. با استفاده از الگوریتم کلونی مورچه ها وظایفی که بصورت تصادفی تولید شده در هر دو حالت همگن و ناهمگن زمان بندی شد که نتایج تاییدی بر موثر تر بودن سیستم ناهمگن در اکثر حالات است. این سیستم با داشتن ترکیب های متنوع، قابلیت بالاتری در انطباق با تعداد وظایف دارد و حداقل منابع مورد نیاز فعال می شوند تا انرژی استاتیک کمتری مصرف شود بنابراین در کاربردهایی که تعداد وظایف مشخص نیست یا در هر برهه از زمان متفاوت است، استفاده از سیستم ناهمگن به جای همگن ارجحیت دارد. نتایج شبیه سازی برای تعداد وظایف مجزا نشان داد با سیستم ناهمگن، مصرف انرژی بطور میانگین ۷/۱ نسبت به سیستم همگن کاهش می یابد. همچنین نتایج شبیه سازی یک DAG با تکرار ۱۰۰۰ بار، کاهش میانگین ۶/۴۴ انرژی مصرفی را نشان داد. هزینه زمان بندی سیستم همگن کمی بالاتر است، اما این تفاوت قابل چشم پوشی است. در آینده نیاز است تا تمامی سربارها و نیز محدودیت های فیزیکی در بازپیکربندی در هر دو سیستم مورد ملاحظه قرار گیرد و در صورت امکان، بصورت عملی با استفاده از تجهیزات مناسب، مستقیماً توان و انرژی محاسبه شود. بکارگیری چنین سیستمی در یک کاربرد خاص که انرژی مصرفی در آن اهمیت دارد نیز موضوع مناسبی برای کار در آینده است. در این مقاله فرض بر مستقل بودن وظایف بود که وظایف وابسته نیز یک

فعال در نظر گرفته می شد، در برخی حالات مانند سطر دهم حالت مصرف انرژی سیستم ناهمگن افزایش می یافت و در برخی حالات برعکس که از این مسئله صرف نظر شده است. همچنین تعداد تراشه نیز می تواند اثر داشته باشد که البته اثر آن ناچیز است و صرف نظر شده است. در این حالت، دو تراشه در سیستم همگن و سه تراشه در سیستم ناهمگن فعال شده اند. در سطر هفتم از آنجا که در هر دو سیستم تراشه ای با حجم یکسان فعال می شود، انرژی مصرفی یکسان است. در سطر دهم، دو تراشه از سیستم همگن و تراشه چهارم که مجموع بلاک آن از دو تراشه همگن بیشتر است فعال شده است. از سوی دیگر به دلیل سربارهای پیکربندی وظایف که همگی بطور سریال باید در یک تراشه پیکربندی شوند، انرژی سیستم ناهمگن از همگن بیشتر بوده است. مصرف انرژی در سیستم ناهمگن در اکثر اوقات و بطور میانگین کمتر از حالت همگن است و البته این مقدار به وظایف (تعداد وظایف و بلاک ها و زمان اجرا) بستگی دارد. کاهش انرژی بطوری میانگین برای تعدادی وظیفه تولید شده بصورت تصادفی حدود ۷/۱ بوده است. این مقدار در تعداد وظایف کوچکتر بیشتر است.

در شکل ۳، همانطور که مشاهده می شود، برای تعداد وظیفه ثابت، و نیز تعداد بلاک ثابت در هر وظیفه با یک زمان تصادفی، نتایج ارایه شده است. برای هر کدام از این حالات، برنامه ۵۰ بار اجرا شده و از نتایج میانگین گیری شده است. نکته جالب توجه اینست که در حالت ۱۲ وظیفه، کاهش انرژی در تعداد بلاک ۲۰۰۰ منفی است. یعنی سیستم ناهمگن انرژی بیشتری مصرف می کند. دلیل آن نیز همانطور که گفته شد، تطابق بیشتر تعداد بلاک مورد نیاز با سیستم همگن است. بعنوان مثال تعداد بلاک های مورد نیاز برای ۱۲ وظیفه هرکدام با ۲۰۰۰ بلاک برابر ۲۴۰۰۰ بلاک می شود که در سیستم همگن، نیاز به فعال شدن دو تراشه است که اتفاقاً تعداد بلاک هایشان تقریباً همین مقدار می شود اما در سیستم ناهمگن، نیاز به فعال شدن تراشه چهارم است در این حالت بطور میانگین انرژی مصرفی سیستم ناهمگن ۴ درصد بیشتر از سیستم همگن است.

در جمع بندی چنین نتیجه گیری می شود که هم زمان وظایف، هم تعداد بلاک های آنها و هم تعداد وظایف در

- Low Power Design", International Conference on Intelligent Control and Information Processing August 13-15, Dalian, China, 2010.
- [11] Czapski P. P., Sluzek A., "A Survey on System-Level Techniques for Power Reduction in Field Programmable Gate Array (FPGA)-Based Devices", IEEE SENSORCOMM '08. Second International Conference on, pp. 319 – 328 , 25-31 Aug. 2008.
- [12] Shang L., Kaviani A. S., Bathala K., "Dynamic Power Consumption in Virtex-II FPGA Family", in: Proc. of the 2002 ACM/SIGDA 10th International Symposium on Field-Programmable Gate Arrays, ACM Press, pp. 157 – 164, 2002.
- [13] Gayasen A., Tsai Y., Vijaykrishnan N., Kandemir M., Irwin M.J., Tuan T., "Reducing Leakage Energy in FPGAs Using Region-Constrained Placement", FPGA'04, pp. 51-58, February 22-24, Monterey, California, USA, 2004.
- [14] Anderson J. H., Najm F. N., "Active Leakage Power Optimization for FPGAs", IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 25, No. 3, pp. 423 - 437 MARCH 2006.
- [15] Virtex-5 Family Overview, Technical Report, DS100 (v5.0) February 6, 2009. [http://www.xilinx.com/support/documentation/data\\_sheets/ds100.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf)
- [16] Hoo C. H., Ha Y., Kumar A., "A directional coarse-grained power gated FPGA switch box and power gating aware routing algorithm", in: IEEE 23rd International Conference on Field Programmable Logic and Applications (FPL), 2013.
- [17] Ishihara S., Hariyama M., Kameyama M., "A low-power FPGA based on autonomous fine-grain power gating", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 19, No. 8, pp. 1394-1406, 2011.
- [18] Dorigo M., Mauro B., Thomas S., "Ant colony optimization", Computational Intelligence Magazine, IEEE, Vol. 1, No. 4, pp. 28-39., Nov. 2006.
- [19] Partial Reconfiguration User Guide, Xilinx Inc. UG702, Ver. 12.1, May 2010. [http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx14\\_1/ug702.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/ug702.pdf)
- مسئله مهم در کارهای بعدی است که باید بدان پرداخته شود.
- ### مراجع
- [1] Chang C., Wawrzynek J., Brodersen R.W., "BEE2: a high-end reconfigurable computing system", IEEE Design & Test of Computers Vol 22, No. 2, pp.114–125, 2005.
- [2] Lu Y., Marconi T., Bertels K., Gaydadjiev G., "Online task scheduling for the FPGA-based partially reconfigurable systems", in: Proc. International Workshop on Reconfigurable Computing: Architectures, Tools and Applications, 2009.
- [3] Gu Z., Yuan M., He X., "Optimal static task scheduling on reconfigurable hardware devices using model-checking", in: Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2007.
- [4] Perng N. C., Chen J. J., Yang C. Y., Kuo T. W., "Energy-efficient scheduling on multi-context FPGA's", in: Proc. IEEE International Symposium on Circuits and Systems (ISCAS), 2006.
- [5] Jing C., Zhu Y., Li M. "Energy-efficient scheduling on multi-FPGA reconfigurable systems", Microprocess. Microsyst. Vol. 37, No. 6, pp. 590 – 600, 2013.
- [6] Li F., Lin Y., He L., "Vdd Programmability to Reduce FPGA Interconnect Power", in: Proc. of the ACM/SIGDA 12th international symposium on FPGA (FPGA '04), ACM New York, USA 2004.
- [7] Khandelwal V., Davoodi A., Srivastava A., "Simultaneous V/sub t/ Selection and Assignment for Leakage Optimization", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 13, No. 6, pp. 762 - 765 June 2005.
- [8] Li F., Lin Y., He L., "FPGA Power Reduction Using Configurable Dual-Vdd", in: Proc. of the 41st annual design automation, ACM New York, USA 2004.
- [9] Wang Q., Gupta S., Anderson J. H., "Clock Power Reduction for Virtex-5 FPGAs", in: Proc. of the ACM/SIGDA international symposium on FPGA (FPGA '09), ACM New York, USA 2009.
- [10] Sun F., Wang H., Fu F., Li X., "Survey of FPGA

---

<sup>1</sup> Deadline

<sup>2</sup> Dynamic Voltage Scaling (DVS)

<sup>3</sup> Configurable Logic Block (CLB)

<sup>4</sup> Directed Acyclic Graph