

طراحی بهینه‌ی مدارهای اتاماتای سلولی کوانتومی بر مبنای تعداد سلول‌های کوانتومی

به کمک الگوریتم‌های ژنتیک

محبوبه هوشمند^۱، منیره هوشمند^۲، راضیه رضایی^۳ و سید مجتبی روحانی^۴

^۱دانشگاه فردوسی مشهد، Ma.Hooshmand@stu-mail.um.ac.ir

^۲دانشگاه امام رضا (ع) مشهد، Monirehhooshmand@gmail.com

^۳دانشگاه فردوسی مشهد، Razie.Rezaei@stu-mail.um.ac.ir

^۴دانشگاه آزاد اسلامی-واحد گناباد، m.rouhani@icee.org

چکیده- اتاماتای سلولی کوانتومی، یک تکنولوژی نوین برای پیاده‌سازی مدارات منطقی، در سطح نانو می‌باشد. در این مقاله، از الگوریتم‌های ژنتیک برای طراحی و بهینه‌سازی اتاماتای سلولی کوانتومی، استفاده شده‌است. بلوکهای پایه در اتاماتای سلولی کوانتومی، گیت اکثریت و گیت معکوس‌کننده هستند، که به ترتیب با پنج و یازده سلول کوانتومی قابل تحقق می‌باشند. روش‌های قبلی، در پیاده‌سازی مدارات بولی تک خروجی، بهینه‌سازی را بر حسب گیت‌های اکثریت و معکوس‌کننده انجام داده‌اند. با توجه به تعداد متفاوت سلول‌های کوانتومی مورد نیاز برای پیاده‌سازی گیت اکثریت و گیت معکوس‌کننده، مشخص است بهینه‌سازی تعداد گیت‌ها الزاماً به بهینه‌سازی تعداد سلول‌های کوانتومی نمی‌انجامد. ما در مقاله خود بهینه‌سازی را در سطح سلول‌های کوانتومی تشکیل‌دهنده‌ی این گیت‌ها انجام می‌دهیم. در گام اول، با تغییر یک روش قبلی، که بهینه‌سازی را بر حسب مجموع گیت‌های اکثریت و معکوس‌کننده انجام می‌داد، بهینه‌سازی برای مدارات تک خروجی بر مبنای تعداد سلول‌های کوانتومی ارائه می‌شود. در گام دوم روش گام اول توسعه‌دهنده شده و تعداد سلول‌های کوانتومی برای مدارات با دو خروجی بهینه می‌شود. شبیه‌سازی‌ها نشان می‌دهند که روش پیشنهادی ما تعداد سلول‌های کوانتومی را برای مدارات با یک و دو خروجی کاهش می‌دهد.

کلید واژه- اتاماتای سلولی کوانتومی، سلول کوانتومی، بهینه‌سازی، گیت اکثریت و گیت معکوس‌کننده.

۱- مقدمه

استفاده از $p=+1$ برای نمایش '1' منطقی و $p=-1$ برای نمایش '0' منطقی، اطلاعات دودویی می‌توانند انکد شوند. دوگیت اصلی منطقی در QCA، گیت اکثریت و گیت معکوس‌کننده، می‌باشد. یک گیت اکثریت متشکل از ۵ سلول کوانتومی است. با فرض A, B, C به عنوان ورودیهای گیت اکثریت، تابع منطقی گیت اکثریت به صورت زیر است:

$$M(A, B, C) = AB + AC + BC$$

با ثابت نگهداشتن یکی از ورودی‌ها، در منطق ۰ یا ۱ به ترتیب می‌توانیم گیت and و or بسازیم:

$$M(A, B, 1) = A + B, \quad M(A, B, 0) = AB$$

در گیت معکوس‌کننده، سلولهایی که در زاویه ۴۵ درجه نسبت به همدیگر قرار دارند، دارای قطبیت مخالف همدیگر هستند. این گیت با ۱۱ سلول کوانتومی قابل تحقق است [۶].

با توالی سلول‌های کوانتومی، سیم QCA را خواهیم داشت. در یک سیم QCA، یک سیگنال دودویی از ورودی به خروجی به علت دافعه کولنی بین سلول‌ها منتقل می‌شود. شکل ۲ سیم QCA، گیت‌های اکثریت و معکوس‌کننده را نشان می‌دهد.

اتاماتای سلولی کوانتومی (QCA) [۱-۵] انقلابی را در تکنولوژی-های محاسباتی در سطح نانو ایجاد کرده است. به علت مشکلات موجود در کاهش سایز مدارهای ترانزیستوری، QCA به عنوان جایگزینی برای چنین مدارهایی پیشنهاد شده‌است. واحد اصلی مدارهای QCA، سلول کوانتومی است. همانطور که در شکل ۱ مشخص است، یک سلول کوانتومی، متشکل از چهار نقطه کوانتومی می‌باشد که در چهار گوشه یک مربع قرار دارد.



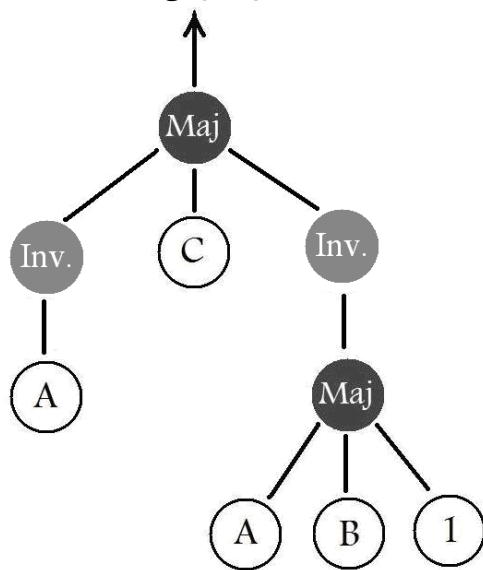
شکل ۱- یک سلول کوانتومی

هر سلول، با دو الکترون اضافی شارژ می‌شود که می‌توانند بین نقاط داخل سلول تونل‌زنی کنند. به علت دافعه کولنی، دو الکترون اضافی، راس‌های یکی از دو قطر را اشغال می‌کنند. این امر دو قطبیت مختلف ($p=+1$ و $p=-1$) را برای هر سلول ایجاد می‌کند. با

۲- روش پیشنهادی

۱-۲ بهینه‌سازی بر مبنای تعداد سلول‌های کوانتومی برای مدارات با یک خروجی

در [۱۰]، روشی برای بهینه‌سازی مدارات QCA بر مبنای الگوریتم‌های ژنتیک ارائه شده‌است. این روش را گیت تک خروجی می‌نامیم. این روش مجموع تعداد گیت‌های اکثریت و گیت‌های معکوس‌کننده را در پیاده‌سازی یک تابع بولی کاهش می‌دهد. بر خلاف اکثر روش‌هایی که از ساختار ماتریسی برای بهینه‌سازی مدارات منطقی استفاده می‌کنند، روش گیت تک خروجی از ساختار درختی برای نمایش کروموزوم‌ها استفاده می‌نماید. گره‌های داخلی درخت، گیت‌های اکثریت و معکوس‌کننده هستند و برگ‌های درخت یا شامل "۱" منطقی و یا متغیر بولی هستند. برای مثال، شکل ۳، کروموزوم مورد نظر برای ایجاد تابع معکوس‌کننده $M(A', C, M(A, B, 1))$ را نشان می‌دهد.

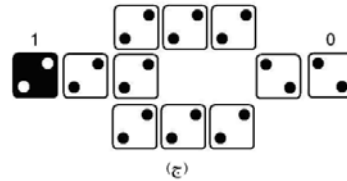
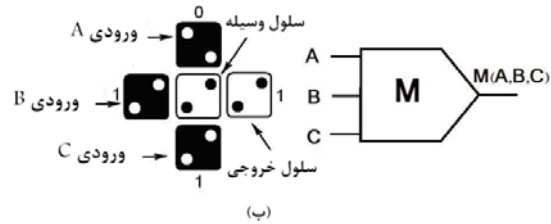
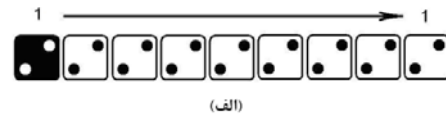


شکل ۳- یک کروموزوم برای ایجاد تابع $M(A', C, M(A, B, 1))$

هر کروموزوم، در این نمایش، یک مدار منطقی را نمایش می‌دهد. تعداد متغیرها را n تابع منطقی که قرار است پیاده‌سازی شود را F و C را یک کروموزوم تحت بررسی در نظر بگیرید. تابع برازندگی، بر مبنای شباهت هر کروموزوم با F به صورت زیر تعریف می‌گردد:

$$Fit(C) = \frac{N(F, C)}{2^n} \quad (1)$$

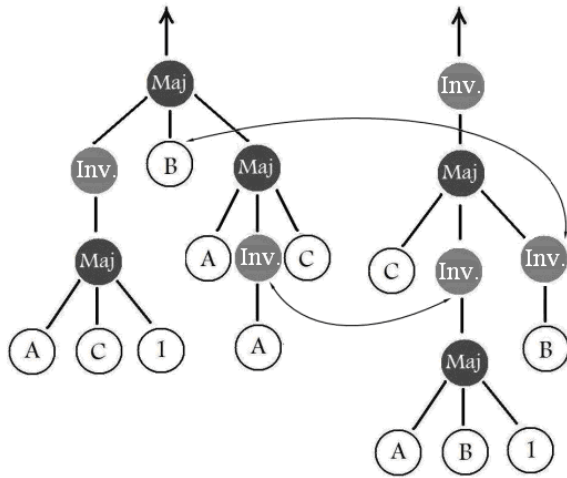
که $N(F, C)$ تعداد مینترم‌های یکسان بین این کروموزوم و تابع F است. اگر یک کروموزوم مینترم‌های یکسانی با تابع F داشته باشد، مقدار تابع برازندگی که توسط (۱) تعریف می‌شود، برابر با ۱ خواهد



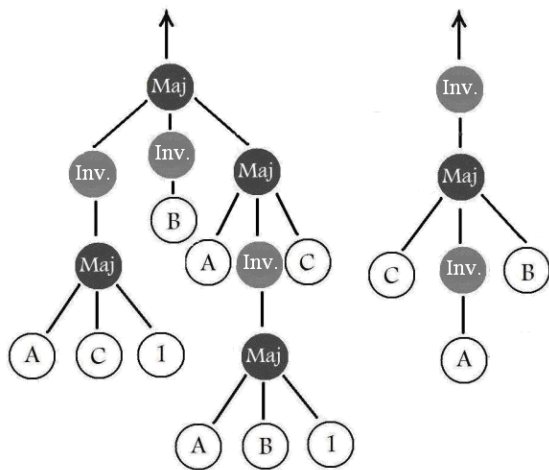
شکل ۲- الف) سیم QCA (ب) یک گیت اکثریت QCA (ج) یک گیت معکوس‌کننده

بهینه‌سازی، یک گام ضروری در طراحی مدارات منطقی است. نمایش بهینه‌ی مدارات منطقی بر حسب گیت‌های اکثریت و معکوس‌کننده، به علت گیت‌های اکثریت چند سطحی، امری دشوار است. در این حوزه، پژوهش‌هایی برای مدارات QCA انجام شده است [۷-۱۰] که همگی بهینه‌سازی در سطح گیت را مبنای قرار داده‌اند.

ما در مقاله خود، بهینه‌سازی سلول‌های کوانتومی تشکیل‌دهنده‌ی این گیت‌ها را مبنای قرار می‌دهیم، چون تعداد سلول‌های کوانتومی برای پیاده‌سازی گیت معکوس‌کننده بیش از تعداد لازم برای پیاده‌سازی گیت اکثریت است، بهینه‌سازی تعداد گیت‌ها الزاماً به بهینه‌سازی تعداد سلول‌های کوانتومی تشکیل‌دهنده‌ی آن‌ها نمی‌انجامد. ما در گام اول، یک روش قبلی را [۱۰] که بهینه‌سازی را بر مبنای مجموع تعداد گیت‌های اکثریت و معکوس‌کننده برای مدارات QCA با یک خروجی انجام می‌دهد، تغییر داده و بهینه‌سازی را بر مبنای تعداد سلول‌های کوانتومی تشکیل‌دهنده‌ی آن‌ها انجام می‌دهیم. سپس در گام بعد، روش بیان‌شده در گام اول را گسترش داده و این بهینه‌سازی را برای مدارات با دو خروجی انجام می‌دهیم. ساختار این مقاله به صورتی که در ادامه می‌آید است: در بخش ۲ در ابتدا روش پیشنهادی برای بهینه‌سازی مدارات با یک خروجی بر مبنای تعداد سلول‌های کوانتومی تشکیل‌دهنده‌ی گیت‌ها ارائه می‌شود. سپس این روش برای مدارات با دو خروجی گسترش داده می‌گردد. در بخش ۳ نتایج شبیه‌سازی‌ها آمده‌است. بخش ۴ به نتیجه‌گیری مقاله می‌پردازد.



الف) پیش از عمل تقاطع



ب) پس از عمل تقاطع

شکل ۴- عمل تقاطع الف- قبل از عمل تقاطع ب- پس از عمل تقاطع

عمل انتخاب توسط الگوریتمی دلخواه در [۱۱] می‌تواند انجام شود.

۲-۲ بهینه‌سازی بر مبنای تعداد سلول‌های کوانتومی برای مدارات با دو خروجی

روش سلول_تک_خروجی برای مدارات با یک خروجی کارآمد است. اما در محیط‌های واقعی، اغلب مدارات بیش از یک خروجی دارند. بهینه‌سازی مدارات با چند خروجی به نحوی که هر خروجی به‌طور مستقل بهینه گردد، الزاماً به یک پاسخ کلی بهینه نمی‌انجامد. چرا که در پیاده‌سازی هم‌زمان چند خروجی، ممکن است بتوان گیت‌های مشترکی را پیدا کرد.

در این گام ما روش سلول_دو_خروجی را برای بهینه‌سازی مدارات با دو خروجی گسترش می‌دهیم.

فرض کنید قصد داریم که تعداد سلول‌های کوانتومی یک مدار با دو خروجی را کمینه نماییم. خروجی‌های مدار را $out1$ و $out2$ در

شد. در این صورت در روش گیت_تک_خروجی، تابع برازندگی دیگری، مطابق (۲) استفاده می‌شود:

$$Fit(C) = 1 + \frac{1}{Nodes(C)} \quad (2)$$

که $Nodes(C)$ تعداد گره‌ها را در C نشان می‌دهد.

ما در این گام، در روش پیشنهادی خود، که آن را سلول_تک_خروجی می‌نامیم، قصد داریم تعداد سلول‌های کوانتومی تشکی-دهنده‌ی گیت‌ها را کمینه کنیم. لذا، تابع برازندگی را متفاوت با (۲) و به صورت (۳) تعریف می‌کنیم.

$$Fit(C) = 1 + \frac{1}{(\alpha.No_{maj} + \beta.No_{inv})(C)} \quad (3)$$

که No_{maj} تعداد گیت‌های اکثریت و No_{inv} تعداد گیت‌های معکوس‌کننده C را نشان می‌دهد. α و β در بازه‌ی $[0, 1]$ ، وزن-های نرمال‌شده‌ای هستند که به ترتیب به گیت‌های اکثریت و معکوس‌کننده منتسب می‌کنیم. همانطور که در مقدمه مطرح شد، چون، تعداد سلول‌های کوانتومی برای تحقق گیت اکثریت برابر با ۵ و تعداد سلول‌های کوانتومی برای پیاده‌سازی گیت معکوس‌کننده

۱۱ است، لذا α را $\frac{5}{16}$ و β را $\frac{11}{16}$ در نظر می‌گیریم. توابع

برازندگی (۱) و (۳) با یکدیگر، کروموزوم‌ها را هم بر مبنای شباهت با تابع بولی F و هم تعداد سلول‌های کوانتومی رتبه‌دهی می‌کنند.

برای کروموزوم‌هایی که تابع منطقی مورد نظر را پیاده‌سازی می‌کنند، مقدار تابع برازندگی ۱ یا بیشتر، وابسته به تعداد سلول‌های کوانتومی است. هرچه قدر تعداد سلول‌های کوانتومی کمتر باشد، مقدار تابع برازندگی بیشتر می‌شود.

این روش با یک جمعیت اولیه تصادفی شروع می‌کند. تکامل با جهش و تقاطع انجام می‌گردد. جهش، یک یا تعدادی ژنوم را در جمعیت با تعدادی دیگر از فضای مساله عوض می‌کند. عمل جهش برای ساختار درختی عمل پیچیده‌ای است. برای مثال، حالتی را در نظر بگیرید که بخواهیم گره‌ی معکوس‌کننده را با گره اکثریت تعویض کنیم. به این علت که تعداد ورودی‌ها در ساختار معکوس-کننده و اکثریت، متفاوت است، مساله بغرنج می‌گردد.

لذا روش نوینی برای عمل جهش برای این ساختار کروموزومی ارائه گشته‌است. در این روش یک کروموزوم تصادفی تولید می‌گردد و به جای کروموزومی که بدترین برازندگی را در جمعیت دارد به جمعیت اضافه می‌شود. این عمل با احتمال مشخصی انجام می‌گیرد. برای انجام عمل تقاطع، یک گره تصادفی و زیر درخت آن با یک گره تصادفی دیگر و زیر درخت آن در کروموزوم دیگری جابجا می‌گردد. شکل ۴ عمل تقاطع را نشان می‌دهد. عمل تقاطع نیز با احتمال مشخصی صورت می‌پذیرد.

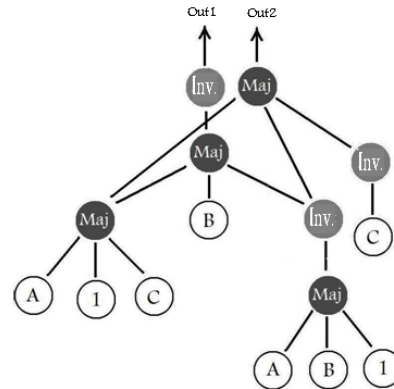
اجرا می‌نماییم. شکل ۷ و شکل ۸ به ترتیب فلوچارت و شبه کد این الگوریتم را نشان می‌دهند.

در این الگوریتم، مجموعه‌ی کروموزوم‌ها برای خروجی ۲، $Chromosome2[i,j]$ خوانده می‌شود و z امین کروموزوم در i امین نسل را نشان می‌دهد.

یک $FinalChromosome2[i,j]$ با ادغام یک $TrueChromosome1$ با یک $Chromosome2[i,j]$ ایجاد می‌شود. سپس، $FinalChromosome2$ با بهترین برازندگی ذخیره می‌شود. هر یک از این کروموزوم‌ها را $TrueFinalChromosome2$ می‌نامیم.

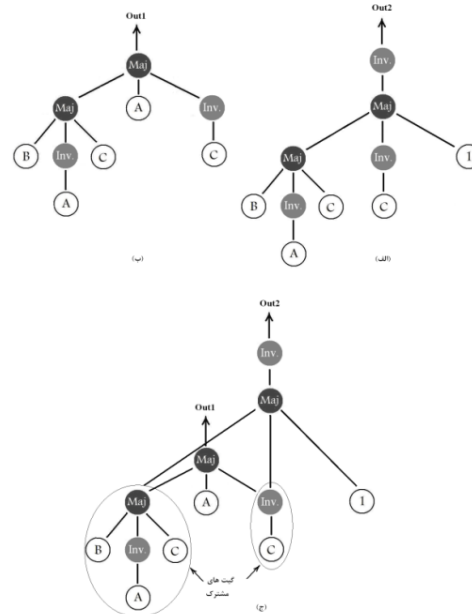
پس از اجرای الگوریتم، برای هر یک از $TrueFinalChromosome1$ ها، بهترین کروموزوم در بین $TrueFinalChromosomes2$ به‌عنوان پاسخ نهایی انتخاب می‌شود.

نظر بگیرید. نمایش یک کروموزوم به‌صورت یک ساختار درختی با دو خروجی را در شکل ۵ مشاهده می‌کنید.

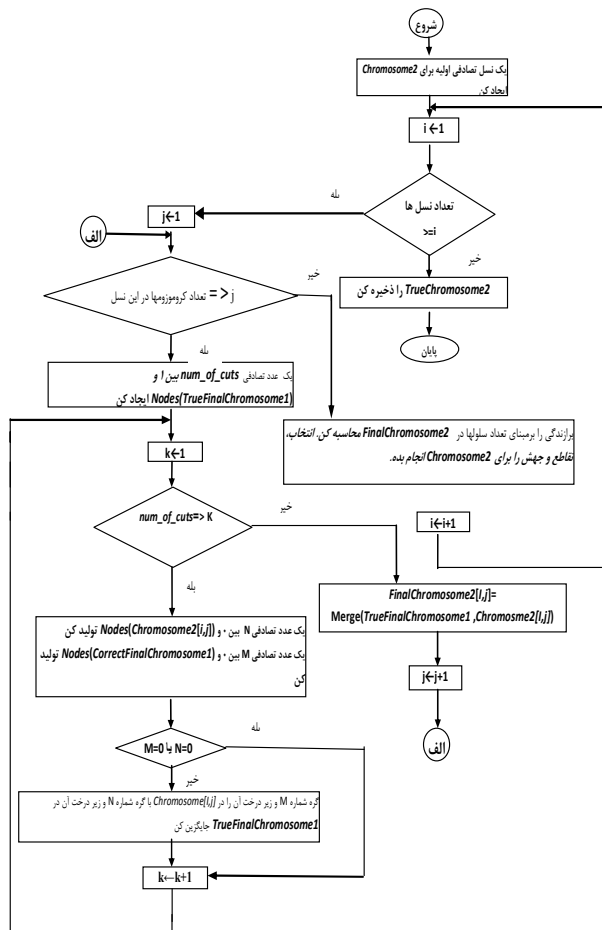


شکل ۵- یک کروموزوم با دو خروجی

پیش از ادامه‌ی مبحث، لازم است دو تابع را تعریف کنیم. تابع $Nodes(C)$ تعداد گره‌های کروموزوم C را بر می‌گرداند و تابع $merge(ch1, ch2)$ دو کروموزوم $ch1$ و $ch2$ را به‌عنوان ورودی می‌گیرد و یک کروموزوم با دو خروجی را که ادغامی از این دو کروموزوم است را بر می‌گرداند. شکل ۶ با مثال عملکرد این تابع را نشان می‌دهد.



شکل ۶- الف) $(ch1)$ ، ب) $(ch2)$ ، ج) کروموزوم ادغامی



شکل ۷- فلوچارت الگوریتم ارائه‌شده

در ابتدا روش سلول تک خروجی برای خروجی اول ($Out1$) اجرا می‌کنیم. این روش را آنقدر اجرا می‌نماییم تا کروموزوم‌هایی با برازندگی بیشتر از ۱ کسب کنیم و آن‌ها را ذخیره می‌نماییم. این کروموزوم‌ها را $TrueFinalChromosome1$ می‌نامیم. سپس برای هر یک از $TrueFinalChromosome1$ الگوریتم پیشنهادی خود را

۱- اولین نسل تصادفی را برای $Chromosome2$ تولید کن

۲- برای $i=1$ تا $number\ of\ generations$ تکرار کن

۱-۲- برای $j=1$ تا $number\ of\ chromosomes$ در این نسل تکرار کن

۱-۱-۲- عدد تصادفی num_of_cuts را بین ۱ و $Nodes(TrueFinalChromosome1)$ تولید کن

// عمل ادغام (شکل ۹)

۱-۱-۱-۲- برای $k=1$ تا num_of_cuts تکرار کن

۱-۱-۱-۱-۲- عدد تصادفی M را بین ۰ و $Nodes(Chromosome2[i,j])$ تولید کن

۱-۱-۱-۲- عدد تصادفی N را بین ۰ و $Nodes(TrueFinalChromosome1)$ تولید کن

۱-۱-۱-۲-۳- اگر یکی از متغیرهای M, N است، کاری انجام نده،

در غیر این صورت N شماره گره و زیر درخت آن را در $TrueFinalChromosome1$ جایگزین گره با شماره M و زیر درخت آن در

$Chromosome2[i,j]$ کن. // توجه کنید که $TrueChromosome1$ تغییری نمی‌کند.

۱-۱-۱-۲- $FinalChromosome2[i,j] = Merge(TrueFinalChromosome1, Chromosome2[i,j])$

۱-۱-۱-۲-۳- عمل تقاطع، جهش و انتخاب را برای $Chromosome2$ ، مشابه حالت مبتنی بر سلول تک‌خروجی انجام بده. محاسبه‌ی برازندگی (۳) برای

$FinalChromosome2$ ، به‌صورتی که در ادامه می‌آید انجام می‌شود

$Fit(FinalChromosome2) =$

$$1 + \frac{1}{(5.No_{maj} + 11.No_{inv})(TrueChromosome1) + (5.No_{maj} + 11.No_{inv})(Chromosome2[i,j]) - (5.No_{Common_maj} + 11.No_{Common_inv})}$$

No_{Common_inv} و No_{Common_maj} به ترتیب تعداد گیت‌های مشترک اکثریت و معکوس‌کننده بین $Chromosome2[i,j]$ و $FinalChromosome2$ را نشان می‌دهد.

۲-۳- $TrueFinalChromosome2$ را ذخیره کن (بهترین کروموزوم در آخرین نسل)

شکل ۸- الگوریتم ارائه شده

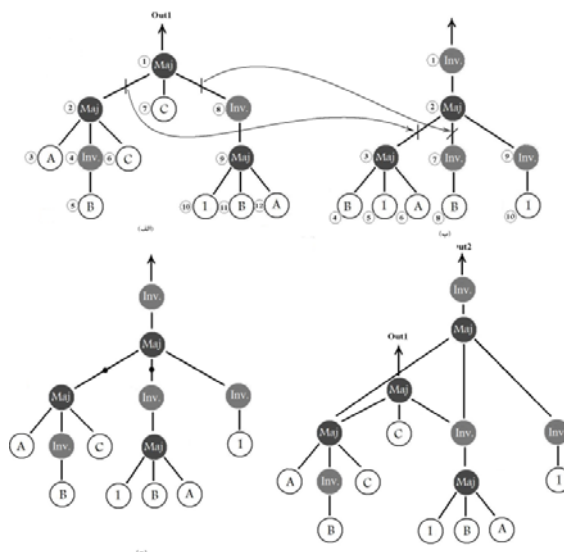
۳- نتایج شبیه‌سازی

ما روش پیشنهادی خود، سلول تک‌خروجی را برای تمام مدارات ممکن به فرم مجموع‌مینترمها با ۳ ورودی اعمال کردیم. هم‌چنین روش گیت تک‌خروجی را نیز برای این مدارات اعمال کردیم.

روش سلول تک‌خروجی در ۸۵٪ موارد تا ۱۲٪ کاهش در سلول کوانتومی نسبت به روش گیت تک‌خروجی ایجاد کرد. در بقیه موارد با روش گیت تک‌خروجی تعداد برابری سلول کوانتومی تولید کرد.

روش سلول دو‌خروجی را نیز برای ۱۰۰۰ مدار به فرم مجموع‌مینترمها با ۳ ورودی/۲ خروجی که به‌طور تصادفی ایجاد شده‌اند، اعمال کردیم. هم‌چنین روش سلول تک‌خروجی را برای هر یک از خروجی‌های این مدارات به‌طور مجزا اعمال نمودیم. روش سلول دو‌خروجی، در ۶۲٪ مدارات تا ۳۰٪ کاهش در سلول‌های کوانتومی را ایجاد می‌کرد. در بقیه موارد دو روش تعداد سلول‌های کوانتومی یکسانی تولید می‌کردند.

پارامترهای تعداد نسل‌ها را بین ۱۰۰۰ تا ۵۰۰۰، تعداد جمعیت اولیه را ۱۰۰، احتمال جهش را ۰.۰۶ و احتمال تقاطع را ۰.۷ در نظر گرفتیم. الگوریتم چرخ رولت را برای انتخاب اعمال کردیم.



شکل ۹- عمل ادغام برای $num_of_cuts=2$ و $N(1)=2$ و $N(2)=8$ (الف) یک $TrueFinalChromosome$ (ب) $Chromosome2$ (ج) $Chromosome2$ بعد از گرفتن چند گیت مشترک از $TrueChromosome1$ (د) $FinalChromosome2$ (با دو خروجی)

جدول ۱ یک نمونه از مقایسه نتایج روش سلول تک خروجی و جدول ۲- یک نمونه از مقایسه نتایج روش سلول دو خروجی و گیت تک خروجی را نشان می‌دهد. اجرای مستقل روش سلول تک خروجی را نشان می‌دهد.

جدول ۱- یک نمونه از مقایسه نتایج روش سلول تک خروجی و گیت تک خروجی

تابع (مجموع مینترم‌ها)	روش مورد استفاده	مدار به دست آمده	تعداد گیت معکوس-کننده	تعداد گیت اکثریت	تعداد کل گیت‌ها	تعداد کل سلول‌های کوانتومی	تعداد کاهش سلول‌های کوانتومی
$F_1 = \sum m(1,3,4,6)$	گیت تک خروجی	$M(1, M(A, 1', C'), M(1, A, C'))'$	۴	۳	۷	۵۹	
	سلول تک خروجی	$M(M(1, A, C), M(1, B, 1)'), M(1, A, C)'$	۳	۴	۷	۵۳	

جدول ۲- یک نمونه از مقایسه نتایج روش سلول دو خروجی و اجرای مستقل روش سلول تک خروجی

تابع (مجموع مینترم‌ها)	روش مورد استفاده	مدار به دست آمده	تعداد گیت معکوس‌کننده	تعداد گیت اکثریت	تعداد سلول‌های کوانتومی	تعداد کاهش سلول‌های کوانتومی
$F_1 = \sum m(0,3,4)$	سلول تک خروجی به‌طور مستقل	$M(M(1, A, M(B, C, A)), M(1, B, C), A)'$	۲	۴	۴۲	
		$M(M(1, A, C), M(C, 1, B)'), B$	۱	۳	۲۶	
		تعداد کل (خروجی ۱ + خروجی ۲ - مشترک)	۱+۲=۳	۳+۴=۷	۲۶+۴۲=۶۸	
	سلول دو خروجی	$M(M(C, B, 1)', M(A, C, M(B, 1, A)), A)'$	۲	۴	۴۲	
		$M(M(A, C, M(B, 1, A)), B, M(C, B, 1)')$	۱	۴	۳۱	
		تعداد کل (خروجی ۱ + خروجی ۲ - مشترک)	۱+۲-۱=۲	۴+۴-۲=۵	۳۱+۴۲-۲۶=۴۷	

- [۴] N. Gergel, S. Craft, and J. Lach, "Modeling QCA for area minimization in logic synthesis," in *ACM Great Lakes Symposium VLSI*, 2003.
- [۵] M. Wilson, "Nanotechnology: Basic Science and Emerging Technologies," London, UK: Chapman & Hall, 2002.
- [۶] R. Farazkish, M. R. Azghadi, K. Navi, and M. Haghparast, "New Method for Decreasing the Number of Quantum Dot Cells in QCA Circuits," *World Applied Sciences Journal*, vol. 4, pp. 793-802, 2008.
- [۷] R. Zhang, P. Gupta, and N. K. Jha, "Synthesis of majority and minority networks and its applications to QCA, TPL and SET based nanotechnology," in *Conference on VLSI Design*, 2005.
- [۸] R. Zhang, K. Walnut, W. Wang, and G. Jullien, "A method of majority logic reduction for quantum cellular automata," *IEEE Trans. Nanotechnology*, vol. 3, 2004.
- [۹] Z. Huo, Q. Zhang, Z. Huo, and Q. Zhang, "Logic optimization for Majority Gate-Based Nanoelectronic Circuits," in *ISCAS*, 2006.
- [۱۰] M. R. Bonyadi, S. M. R. Azghadi, N. M. Rad, K. Navi, and E. Afjei, "Logic optimization for Majority Gate-Based Nanoelectronic Circuits Based on Genetic Algorithm," in *International Conference on Electrical Engineering, ICEE*, 2007.
- [۱۱] M. Melanie, *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts, London, England: The MIT Press, 1999.

۴- نتیجه‌گیری

در این مقاله، یک روش نوین با استفاده از الگوریتم‌های ژنتیک برای بهینه‌سازی مدارات QCA با یک خروجی بر مبنای سلول‌های کوانتومی تشکیل‌دهنده‌ی گیت‌های اکثریت و معکوس‌کننده معرفی شد. سپس این روش گسترش داده شده، و برای بهینه‌سازی مدارات QCA با دو خروجی استفاده شد. شبیه‌سازی‌ها نشان دادند که این روش برای مدارات با یک خروجی و دو خروجی سلول‌های کوانتومی تشکیل‌دهنده‌ی گیت‌های اکثریت و معکوس‌کننده کمتری تولید می‌کند.

مراجع

- [۱] C. Lent and P. Tougaw, "A Device Architecture for Computing with Quantum Dots," in *Proceeding IEEE*, 1997.
- [۲] P. D. Tougaw and C. S. Lent, "Logical devices implemented using quantum cellular automata," *Appl. Phys.*, 75 (3), pp. 1818-1825, 1994.
- [۳] I. Amlani, "Experimental demonstration of a leadless quantum-dot cellular automata cell," *Appl. Phys. Lett.*, vol. 77, pp. 738-740, 2000.