پیاده‌سازی خودکار مدارهای کوانتومی روی QFPGA با هدف همانندسازی

مصطفی حیدرزاده و محمد دانایی‌فر

چکیده: در این مقاله ابتدا به تعریف یک معماری بهینه برای FPGA با استفاده از روش‌های دقیق پرداخته شده و برای نیل به این هدف، جایابی و مسیریابی بهینه با استفاده از برنامه‌ریزی خطی به طور دقیق تعریف شده است. پس از بازتعریف معماری داخل سلول‌های منطقی، مدارهای کوانتومی توسط یک الگوریتم مکاشفه‌ای با هدف استفاده حداکثری از منابع داخل سلول‌های منطقی و کاهش تأخیر مسیرهایی که کیوبیت‌ها در مدار طی می‌کنند، افراز می‌شوند. نتایج به دست آمده پس از تعریف معماری FPGA نشان می‌دهد که تأخیر مسیرهای بحرانی در برخی مدارهای کوانتومی به کمتر از نصف کاهش می‌یابد و تعداد کانال‌های مصرف‌شده برای مسیریابی در معماری جدید تا حد قابل توجهی کاهش یافته است. همچنین نتایج نشان می‌دهد افزایش تعداد ورودی‌های سلول‌های منطقی از ۱۲ کیوبیت به ٤ کیوبیت، می‌تواند تعداد کانال‌های مصرفی و تأخیر مدارها را تا حد زیادی کاهش دهد.

کلید واژه: محاسبات کوانتومی، شبیه‌سازی مدارهای کوانتومی، همانندسازی مدارهای کوانتومی، گیت‌های کوانتومی، افراز، جایابی، مسیریابی، معماری QFPGA.

۱- مقدمه

در سال ۱۹۶۵ "گوردن مور" اظهار کرد که تعداد ترانزیستورهای موجود بر روی تراشه‌ها هر سال دو برابر خواهد شد. البته این نرخ رشد، کند شد و به دو برابر شدن به ازای هر ۱۸ تا ۲۴ ماه انجامید. این تصاعد هندسی در طول ۴۰ سال اخیر پا بر جا بوده و در تمام این سال‌ها، تلاش عمده در جهت افزایش قدرت و سرعت عملیاتی در کنار کوچک‌سازی زیرساخت‌ها و اجزای بنیادی بوده است.

از ابتدای دهه ۸۰ میلادی و با سرعت‌گرفتن این پیشرفت‌ها، شبهات و پرسش‌هایی در مورد نهایت این کوچک‌سازی‌ها مطرح شد. کوچک‌کردن ترانزیستورهای CMOS و مجتمع‌کردن آنها در فضای کمتر نمی‌تواند تا ابد ادامه داشته باشد، زیرا در حدود ابعاد نانومتری اثرات کوانتومی از قبیل تونل‌زنی الکترونی[1] بروز می‌کنند. در نتیجه بسیاری از دانشمندان در زمینه‌های مختلف به فکر رفع این مشکل افتادند [۱].

از طرف دیگر می‌توان فیزیک کوانتومی را یکی از مهم‌ترین دستاوردهای علم بشری در توصیف طبیعت دانست. به بیان دقیق‌تر، مکانیک کوانتومی مجموعه‌ای از قوانین، روابط ریاضی و مفاهیم فلسفی است که توصیف‌کننده رفتار ذرات بنیادین تشکیل‌دهنده عالم است. البته با تعمیم همین قوانین و روابط، می‌توان رفتار تمام سیستم‌های فیزیکی را که پیش از آن ارائه شده بودند، بررسی و تعیین کرد.

محاسبات مبتنی بر مکانیک کوانتوم یا "محاسبات کوانتومی" به مطالعه سیستم‌های پردازش اطلاعات با استفاده از سیستم‌های مکانیک کوانتوم اطلاق می‌شود. این عرصه از علم که به عنوان یک زمینه تحقیقاتی جدید در هزاره سوم مطرح شده است، در حقیقت مجموعه‌ای از سه علم کلاسیک نظریه اطلاعات، علوم کامپیوتر و فیزیک کوانتوم است و انتظار می‌رود که بتواند تحولات شگرفی را در زمینه‌های مختلف نظیر انتقال امن داده‌ها، افزایش چشم‌گیر سرعت پردازش اطلاعات و نیز رفع محدودیت‌های مجتمع‌سازی در سالیان آتی ایجاد کند.

در واقع هدف محاسبات کوانتومی یافتن روش‌هایی برای طراحی مجدد ادوات شناخته‌شده محاسبات (مانند گیت‌ها و ترانزیستورها) به گونه‌ای است که بتواند تحت اثرات کوانتومی که در محدوده ابعاد نانومتری و کوچک‌تر بروز می‌کنند، کار کنند. در این مباحث می‌توان گفت الکترون‌ها و فوتون‌ها مثالی معمول از ذرات کوانتومی هستند [۲].

مکانیک کوانتوم در برخی موارد با بینش ما در تضاد است. به صورت کلی می‌توان گفت که برای تفسیر مسایل در مکانیک کوانتوم باید از جبر خطی استفاده کرد و برای این منظور از مفاهیم مطرح در اعداد مختلط استفاده می‌شود. همچنین بسیاری از الگوریتم‌های کوانتومی معادل کلاسیک خود را به صورت موازی انجام می‌دهند که در محاسبات کلاسیک غیر ممکن است. این الگوریتم‌ها از خواص فیزیکی نظیر برهم‌نهی[2] و درهم‌تنیدگی[3] برای افزایش سرعت استفاده می‌کنند که این خواص در فصل دوم این رساله توضیح داده شده‌اند. در حال حاضر، برخی اثرات کوانتومی به طور موفقیت‌آمیز در کاربردهای عملی نظیر رمزنگاری و ارتباطات استفاده می‌شود. مطالب مطرح‌شده در این بخش از [۳] تا [۵] انتخاب شده‌اند و جهت بررسی بیشتر می‌توانید به این مراجع رجوع کنید.

به هر حال ایجاد کامپیوترهای کوانتومی بزرگ‌تر و عملی‌تر از برخی نمونه‌های خاص ۱۲ کیوبیتی[4] هنوز میسر نشده است. در نتیجه برای گسترش الگوریتم‌های کوانتومی، مدل‌هایی برای شبیه‌سازی ضروری هستند. از طرفی، فینمن اعلام کرده که کامپیوتر کوانتومی به طور کارا فقط با یک کامپیوتر کوانتومی دیگر قابل مدل‌کردن است [۶] اما در غیاب ماشین‌های کوانتومی بزرگ، الگوریتم‌های کوانتومی توسط کامپیوترهای کلاسیک شبیه‌سازی می‌شود.

یکی از روش‌ها برای مدل‌کردن فرایندهای کوانتومی، روش نرم‌افزاری است [۷] تا [۱۱] و این در حالی است که با استفاده از شبیه‌سازی نرم‌افزاری الگوریتم‌های کوانتومی از قابلیت موازی‌سازی موجود در آنها به

این مقاله در تاریخ ۷ آذر ماه ۱۳۹۰ دریافت و در تاریخ ۲ تیر ماه ۱۳۹۲ بازنگری شد.

مصطفی حیدرزاده، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، (email: heydarzadeh.mostafa@yahoo.com).

محمد دانایی‌فر، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، (email: mdana222@gmail.com).

1. Electronic Tunneling
2. Super-Position
3. Entanglement
4. Qubit

Base (b)	Complexity (i)	Sign (s)

شکل ۱: بیت‌های اضافه‌شده به هر ضریب یک کیوبیت [۱۲].

صورت کارامد نمی‌توان استفاده کرد. روش دیگر برای مدل‌کردن فرایندهای کوانتومی، استفاده از شبیه‌سازی سخت‌افزاری یا همانندسازی[1] است. این همانندسازها نیز مثل شبیه‌سازهای نرم‌افزاری اثرات کوانتومی را تخمین می‌زنند ولی همانندسازی طبیعت موازی محاسبات کوانتومی، کاراتر از معادل نرم‌افزاری آنها انجام می‌شود.

برای شبیه‌سازی یا همانندسازی الگوریتم‌های کوانتومی معمولاً از مدل مداری این الگوریتم‌ها استفاده می‌شود که در آنها محل قرارگیری گیت‌های کوانتومی و ترتیب آنها مشخص شده است. مدارهای کوانتومی[2] از نظر قابلیت موازی‌سازی دارای شباهت‌هایی با بیت‌ها و گیت‌های دیجیتال هستند. این اجزا می‌توانند روی FPGA همانندسازی شوند که در نتیجه وظایف محاسبات موازی را کاراتر از شبیه‌سازهای نرم‌افزاری انجام می‌دهند.

بسیاری از الگوریتم‌های کوانتومی معادل کلاسیک خود را به صورت موازی انجام می‌دهند که در محاسبات کلاسیک غیر ممکن است. این الگوریتم‌ها از خواص فیزیکی نظیر برهم‌نهی و درهم‌تنیدگی برای افزایش سرعت استفاده می‌کنند. در حال حاضر برخی اثرات کوانتومی به طور موفقیت‌آمیز در کاربردهای عملی نظیر رمزنگاری و ارتباطات استفاده می‌شود [۱]، [۲] و [۶]. برای گسترش الگوریتم‌های کوانتومی، مدل‌هایی برای شبیه‌سازی ضروری است و کامپیوتر کوانتومی به طور کارا فقط با یک کامپیوتر کوانتومی دیگر قابل مدل‌کردن است [۱۳] اما در غیاب ماشین‌های کوانتومی بزرگ، الگوریتم‌های کوانتومی توسط کامپیوترهای کلاسیک شبیه‌سازی می‌شود. یکی از روش‌ها به منظور مدل‌کردن فرایندهای کوانتومی، روش نرم‌افزاری است [۳] تا [۵]، [۱۴] و [۱۵]. این در حالی است که با استفاده از شبیه‌سازی نرم‌افزاری الگوریتم‌های کوانتومی نمی‌توان از قابلیت موازی‌سازی موجود در آنها به صورت کارامد استفاده کرد. روش دیگر برای مدل‌کردن فرایندهای کوانتومی، استفاده از شبیه‌سازی سخت‌افزاری یا همانندسازی است [۷] تا [۱۰] و [۱۲]. این همانندسازها نیز مثل شبیه‌سازهای نرم‌افزاری اثرات کوانتومی را تخمین می‌زنند، ولی همانندسازی طبیعت موازی محاسبات کوانتومی، کاراتر از معادل نرم‌افزاری آنها انجام می‌شود.

روش همانندسازی ارائه‌شده در [۱۲] یک روش همانندسازی مدارهای کوانتومی مبتنی بر دو نوع پاولی و دورانی است. ویژگی اصلی گیت‌های پاولی این است که فقط ضرایب کیوبیت‌ها را جابه‌جا می‌کنند و مقادیر این ضرایب را تغییر نمی‌دهند. در این روش، عملیات اعمال گیت به جای جابه‌جا کردن ضرایب که توسط روش‌های همانندسازی دیگر انجام می‌شدند و مستلزم استفاده از تعداد زیادی رجیسترهای میانی بود، توسط افزودن کدهایی به ضرایب کیوبیت‌ها انجام می‌شود. به هر ضریب کیوبیت یک کد سه‌بیتی به صورت شکل ۱ اضافه می‌شود.

در این کد، b نشان‌دهنده پایه آن ضریب کوانتومی است که می‌تواند صفر برای نشان‌دادن پایه $|0\rangle$ و یک برای نشان‌دادن پایه $|1\rangle$ باشد. اگر بیت i یک شود به این معناست که عدد مختلط "j" باید در ضریب مختلط مربوط ضرب شود و اگر بیت s یک شود به این معناست که ضریب مختلط باید در عدد "۱-" ضرب شود.

گیت‌های دورانی مقدار ضرایب را تغییر می‌دهند. بنابراین باید ضرایب این کیوبیت‌ها با تخمینی داخل سخت‌افزار نگهداری شوند تا بتوان عملیات گیت‌ها را روی آنها اعمال کرد. به همین دلیل در این روش از نمایش اعداد اعشاری ممیز ثابت برای این کار استفاده شده است. در این روش برای همانندسازی گیت‌های دورانی، الگوریتمی با نام QCORDIC ارائه شده و از الگوریتم CORDIC که روشی برای پیاده‌سازی دوران بردار $[X_.,Y_.]$ با زاویه θ می‌باشد، استفاده شده است. در [۱۲] پیاده‌سازی QCORDIC به صورت خط لوله‌ای پیشنهاد شده که در شکل ۲ قابل مشاهده است.

از آنجایی که در روش پیشنهادشده برای همانندسازی گیت‌های پاولی جای ضرایب عوض نمی‌شود، زمانی که در مدارهای کوانتومی بعد از گیت‌های پاولی، گیت‌های دورانی باشد، باید ضرایب مرتب شوند. در این مدل همانندسازی برای حل این مشکل از تعدادی سوئیچ استفاده شده که بعد از گیت‌های پاولی و قبل از گیت‌های دورانی قرار می‌گیرند و نقش مرتب‌کردن ضرایب را بر عهده دارند.

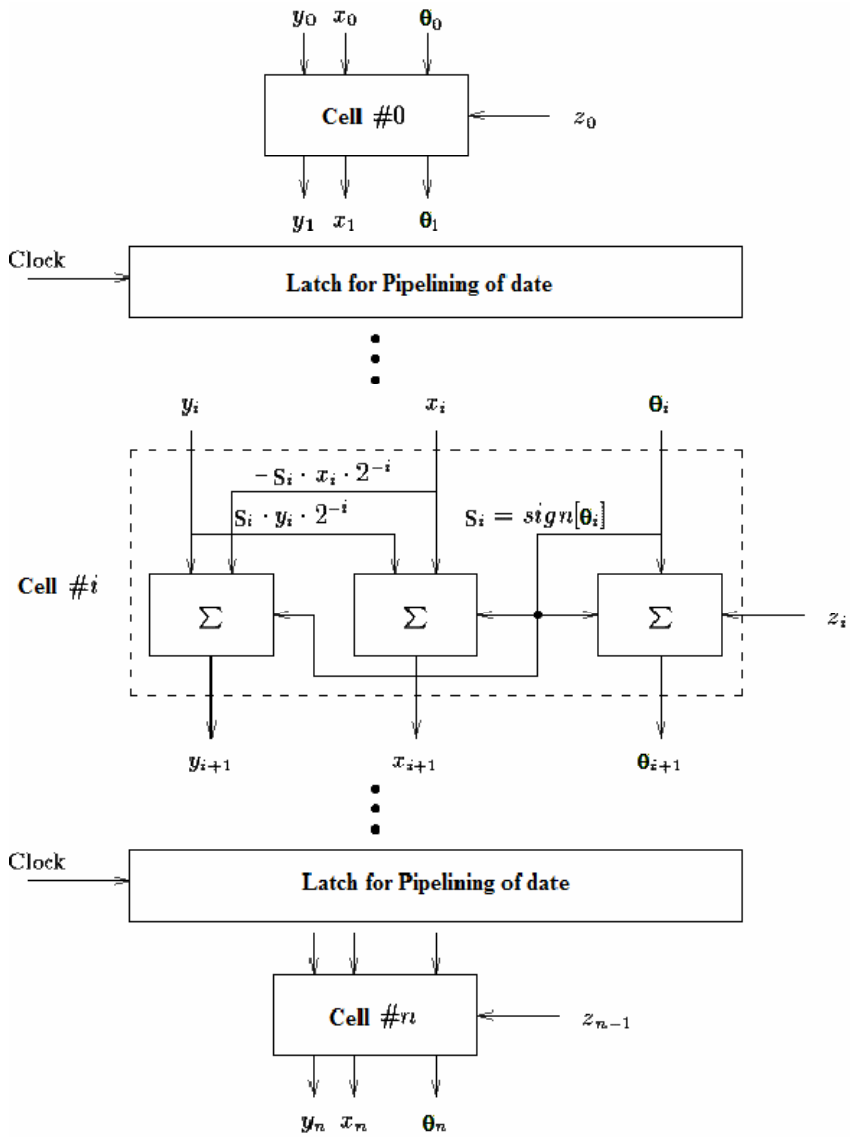
در این مدل همانندسازی، در نهایت یک FPGA که مختص همانندسازی مدارهای کوانتومی است، ارائه شده که در این مقاله با نام QFPGA از آن یاد می‌شود. برای پیاده‌سازی گیت‌های پاولی در QFPGA از بلوک‌هایی شامل تعدادی LUT استفاده شده که هر بلوک LUT قادر به پیاده‌سازی گیت‌های پاولی و کنترلی آنها با حداکثر چهار کیوبیت ورودی می‌باشد. از این بلوک‌ها در این رساله با نام بلوک‌های پاولی یاد می‌شود و برای پیاده‌سازی گیت‌های دورانی از معماری خط لوله‌ای استفاده شده است.

همان طور که در [۱۲] قابل مشاهده است، ساختار نهایی معماری پیشنهادی برای پیاده‌سازی این روش به این صورت است که داخل هر بلوک منطقی در QFPGA، از سمت چپ به ترتیب یک بلوک پاولی، یک مرتب‌کننده و در نهایت دو بلوک دورانی قرار داده شده است.

نتایج به دست آمده از همانندسازی مطرح‌شده در [۱۲] که مبتنی بر گیت‌های پاولی و دورانی است، نسبت به روش‌های همانندسازی دیگر مطلوب‌تر است چرا که به دلیل جامع‌بودن گیت‌ها و عدم استفاده زیاد از منابع در این روش، محدودیت‌های روش‌های دیگر را ندارد. در ضمن گیت‌های دورانی در محاسبات کوانتومی از اهمیت بسیار زیادی برخوردار هستند چرا که برخی تکنولوژی‌هایی که دارای چشم‌اندازی امیدبخش می‌باشند (مانند Ion - Trap) تنها قادر به پیاده‌سازی گیت‌های دورانی هستند [۹].

در این مقاله قرار است یک معماری بهینه برای QFPGA ارائه‌شده در [۱۲] ارائه شود. برای اثبات بهینه‌بودن این معماری باید از بین روش‌های دقیق و مکاشفه‌ای، از روش‌های دقیق استفاده کرد چرا که نتایج آنها دقیق و قابل استناد هستند. از طرفی برای به دست آوردن معماری دقیق، از بین الگوریتم‌های طراحی خودکار (CAD)، الگوریتم‌های جایابی و مسیریابی که به نوع معماری بستگی دارند، باید به صورت دقیق محاسبه شوند و با استفاده از این الگوریتم‌ها معماری بهینه استخراج شود و در نهایت افزار مدارهای کوانتومی بر اساس معماری بهینه طراحی گردد. به همین دلیل ابتدا الگوریتم‌هایی برای جایابی و مسیریابی مدارهای کوانتومی بر اساس FPGA ارائه‌شده در [۱۲] با استفاده از برنامه‌ریزی خطی به طور دقیق ارائه می‌شود، سپس بر اساس این الگوریتم‌ها معماری بهینه برای QFPGA ارائه می‌شود و در نهایت افراز مدارهای کوانتومی انجام می‌شود.

1. Emulation
2. Quantum Circuits



شکل ۲: پیاده‌سازی QCORDIC به صورت خط لوله‌ای [۱۲].

۲- جایابی مدار کوانتومی

با توجه به معماری ارائه‌شده QFPGA در [۱۲] و کوچک‌بودن سلول‌های منطقی و مدارات کوانتومی، جایابی به این صورت تعریف می‌شود که هر افراز مدار کوانتومی، دقیقاً توسط کدام سلول منطقی اجرا شود به گونه‌ای که حداکثر مجموع طول سیم، حداقل شود و در نتیجه فرکانس کاری مدار افزایش یافته و کانال‌های مصرف‌شده در QFPGA به حداقل برسد. از آنجایی که دو نوع گیت توسط دو نوع بلوک پیاده‌سازی می‌شوند، ابتدا دو متغیر دودویی، یکی برای تخصیص گره‌های دورانی و دیگری برای تخصیص گره‌های پاولی تعریف شده است.

از آنجایی که دو نوع گیت توسط دو نوع بلوک پیاده‌سازی می‌شوند، ابتدا دو متغیر دودویی، یکی برای تخصیص گیت‌های دورانی و دیگری برای تخصیص گیت‌های پاولی تعریف شده است. برای این منظور متغیر دودویی[1] $X_{i,j}$ تعریف می‌شود که اندیس i نشان‌دهنده گیت‌های پاولی و اندیس j نشان‌دهنده بلوک‌های پاولی داخل QFPGA می‌باشند. اگر گیت پاولی i توسط بلوک پاولی j پیاده‌سازی شود، این متغیر یک می‌شود وگرنه مقدار متغیر صفر خواهد بود. همچنین متغیر دودویی $Y_{i,j}$ تعریف می‌شود که اندیس i نشان‌دهنده گیت‌های دورانی و اندیس j نشان‌دهنده بلوک‌های دورانی داخل QFPGA می‌باشد. مقدار متغیر دودویی $Y_{i,j}$ یک خواهد بود اگر گیت دورانی i توسط بلوک دورانی j پیاده‌سازی شود، در غیر این صورت مقدار این متغیر صفر است. در اینجا فرض بر این است که منابع به صورت اشتراکی استفاده نمی‌شوند، یعنی هر بلوک (پاولی یا دورانی) به طور خاص فقط برای پیاده‌سازی یک نوع گیت استفاده می‌شود. همان طور که در (۱) و (۲) مشاهده می‌شود، برای هر بلوک پاولی مجموع گیت‌های پاولی تخصیص داده شده و نیز برای بلوک‌های دورانی، مجموع گیت‌های دورانی پیاده‌سازی شده برای هر بلوک دورانی باید برابر یک باشد

$$\forall j : \sum_{i=1}^{k} X_{ij} = 1 \qquad (1)$$

$$\forall j : \sum_{i=1}^{l} Y_{ij} = 1 \qquad (2)$$

تعداد بلوک‌های پاولی و بلوک‌های دورانی به ترتیب m و n است و تعداد گیت‌های پاولی و گیت‌های دورانی به ترتیب k و l است. هر گیت دورانی فقط در یک بلوک دورانی باید پیاده‌سازی شود و هر گیت پاولی توسط فقط یک بلوک پاولی باید پیاده‌سازی شود. روابط (۳) و (۴) اشاره به

1. Binary

این نکته دارد که به ازای هر گیت پاولی، مجموع بلوک‌های پاولی و به ازای هر گیت دورانی، مجموع بلوک‌های دورانی برابر یک است

$$\forall i: \sum_{j=1}^{m} X_{ij} = 1 \qquad (3)$$

$$\forall i: \sum_{i=1}^{n} Y_{ij} = 1 \qquad (4)$$

ضمناً همان طور که اشاره شد، متغیرهای x و y دودویی هستند و این مطلب باید صراحتاً در برنامه ذکر شود. به دلیل این که تعداد اتصالات بین گیت‌ها در تخصیص منابع مهم است و سه نوع یال در QDFG وجود دارد (اتصالات بین گیت‌های مختلف باید جداگانه بررسی شود)، سه ماتریس برای مشخص‌کردن تعداد هر نوع ارتباط تعریف می‌شود. برای تعریف پارامتر فاصله بین بلوک‌ها، سه ماتریس D_P2P، D_R2R و D_P2R که به ترتیب نشانگر فاصله بین بلوک‌های پاولی، فاصله بین بلوک‌های دورانی و در نهایت فاصله بین بلوک‌های پاولی و دورانی هستند، تعریف می‌شوند که این فاصله‌ها به معماری داخلی سلول منطقی بستگی دارد. با توجه به معماری QFPGA که هر سلول منطقی به ترتیب یک بلوک پاولی و دو بلوک دورانی را شامل می‌شود، فاصله بین سلول‌های منطقی به مراتب بیشتر از فاصله بین بلوک‌های داخل یک سلول منطقی در نظر گرفته می‌شود، چرا که سیم‌های[1] داخل کوتاه‌تر و کم‌هزینه‌تر و سریع‌تر هستند [۱۰]. اگر فقط یک نوع گیت پاولی و یک نوع بلوک پاولی موجود بود، (۵) تابع هدف را تعریف می‌کرد که باید به حداقل می‌رسید

$$\min: \sum_{i=1}^{k}\sum_{j=1}^{m}\sum_{a=1}^{l}\sum_{b=1}^{n} X_{ij} \times X_{ab} \times n\_P2P_{ia} \times D\_P2P_{jb} \qquad (5)$$

مفهوم (۵) این است که اگر قرار باشد گیت پاولی i در بلوک پاولی j قرار بگیرد، و گیت پاولی a در بلوک پاولی b، تعداد اتصالات بین این دو گیت در QDFG تعیین می‌کند که فاصله بلوک‌های تخصیص داده شده به هر کدام از این دو گیت زیاد یا کم باشد، به طوری که مجموع طول سیم‌ها در QFPGA به حداقل برسد. یعنی سعی شود گیت‌های با تعداد اتصالات زیاد در بلوک‌های نزدیک هم قرار بگیرند. همچنین اگر فقط یک نوع گیت دورانی وجود داشت، تابع هدف به صورت (۶) می‌شد و در نهایت تابع هدف برای مسئله مورد نظر به صورت (۷) می‌شود

$$\min: \sum_{i=1}^{k}\sum_{j=1}^{m}\sum_{a=1}^{l}\sum_{b=1}^{n} Y_{ij} \times Y_{ab} \times n\_R2R_{ia} \times D\_R2R_{jb} \qquad (6)$$

$$\begin{aligned}\min: &[\sum_{i=1}^{k}\sum_{j=1}^{m}\sum_{a=1}^{l}\sum_{b=1}^{n} X_{ij} \times X_{ab} \times n\_P2P_{ia} \times D\_P2P_{jb}] \\ &+[\sum_{i=1}^{k}\sum_{j=1}^{m}\sum_{a=1}^{l}\sum_{b=1}^{n} Y_{ij} \times Y_{ab} \times n\_R2R_{ia} \times D\_R2R_{jb}] \\ &+[\sum_{i=1}^{k}\sum_{j=1}^{m}\sum_{a=1}^{l}\sum_{b=1}^{n} X_{ij} \times Y_{ab} \times n\_P2R_{ia} \times D\_P2R_{jb}]\end{aligned} \qquad (7)$$

در (۷) به دلیل ضرب دو متغیر دودویی در هم، مسأله از حالت خطی خارج شده و جزء مسایل غیر خطی محسوب می‌شود. برای تبدیل الگوریتم‌های غیر خطی به خطی روش‌های متفاوتی وجود دارد که از بین این روش‌ها، روش ارائه‌شده در [۱۱] که (۸) از آن استخراج شده، ساده‌تر است و در این مقاله از آن استفاده شده است. در این رابطه، یک متغیر به نام z که ماتریسی است با ابعادی برابر مجموع ابعاد ماتریس‌های (۷) تعریف می‌شود که از مجموع متغیرهای دودویی X باید بزرگ‌تر باشد که باعث می‌شود متغیرها از حالت غیر خطی خارج شوند

$$\begin{aligned}&\forall i,j,a,b: Z_{i,j,a,b} \geq X_{i,j} + X_{a,b} - 1 \\ &\forall i,j,a,b: Z_{i,j,a,b} \leq \frac{1}{2}(X_{i,j} + X_{a,b})\end{aligned} \qquad (8)$$

۳- مسیریابی مدارهای کوانتومی

قبل از اجرای مسیریابی به طور دقیق مشخص شده است که هر افراز در کدام بلوک QFPGA اجرا می‌شود و حال باید مشخص کرد که اتصالات بین بلوک‌ها به چه ترتیبی مسیریابی شوند. برای مسیریابی نیز همانند جایابی، از الگوریتم‌های دقیق مانند برنامه‌ریزی خطی صحیح استفاده شده است که در ادامه توضیح داده می‌شود.

برای مسیریابی می‌توان ریزدانگی مسیریابی را در سطح بلوک‌ها و یا در سطح سلول‌های منطقی فرض کرد. در بخش معماری پیشنهادی کانال‌های داخل سلول منطقی بیان شده است که بین بلوک‌های داخل سلول‌ها سوئیچ وجود دارد تا کیوبیت‌ها بعد از اعمال حتی یک گیت بر آنها بتوانند به راحتی از سلول خارج شوند و به هر کدام از بلوک‌ها بدون درگیر شدن در بقیه بلوک‌های آن سلول منطقی، می‌توانند دسترسی داشته باشند. پین‌های ورودی هر سلول منطقی در سمت چپ سلول منطقی و پین‌های خروجی در سمت راست سلول منطقی قرار دارند. در نتیجه برای مسیریابی، ریزدانگی در سطح بلوک‌ها در نظر گرفته می‌شود چرا که از یک بلوک دورانی در سلول منطقی می‌توان به صورت مستقیم به بیرون مسیر داشت. بعد از تعیین ریزدانگی، مسیریابی به این صورت انجام می‌شود که ابتدا یک Grid روی QFPGA فرض می‌شود به این صورت که هر خانه[2] از Grid شامل یک سلول منطقی (اگر مسیریابی در سطح سلول‌های منطقی باشد) و یا شامل یک بلوک (اگر مسیریابی در سطح بلوک‌ها باشد) باشد. با توجه به نکته اشاره‌شده در بالا، در این مقاله هر خانه از Grid شامل یک بلوک است. در مسیریابی تفاوت بین بلوک‌های پاولی و دورانی وجود ندارد.

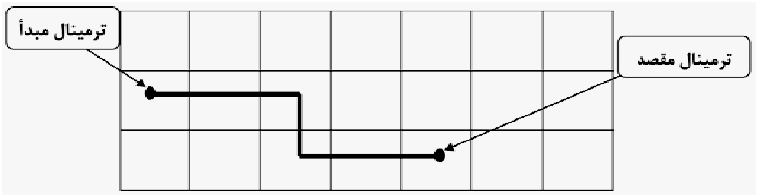
برای مسیریابی به دو روش کلی در حالت برنامه‌ریزی خطی صحیح می‌توان رابطه‌ها را نوشت. یک روش مبتنی بر مسیرهای موجود و روش دیگر مبتنی بر حرکت و بدون در نظر گرفتن مسیرهای از پیش تعیین شده است. در مدل مسیریابی مبتنی بر حرکت، مسیرهای ممکن به الگوریتم داده نمی‌شود و با حرکت خانه به خانه یا کانال به کانال از مبدأ یک Grid، مسیر بهینه تا مقصد توسط الگوریتم پیدا می‌شود.

اولین روش مسیریابی به این صورت است که برای یک مسیر دو ترمینالی، مسیرهای موجود بین دو ترمینال به عنوان ورودی به برنامه داده می‌شوند تا الگوریتم با انتخاب یکی از این مسیرها، مسیر بهینه بین دو ترمینال را تعیین کند. در این حالت اگر تعداد خانه‌های شبکه ۲×۲ فرض شود، تعداد مسیرهای ممکن بین خانه‌های این Grid (مسیرهای موجود)، برای مسیرهای دو ترمینالی ۱۲ عدد خواهد بود. اما اگر تعداد خانه‌های Grid زیاد شود، تعداد مسیرهای بین خانه‌ها بسیار زیاد شده و نمی‌توان همه آنها را برای Grid بزرگ‌تر به دست آورد و نرم‌افزارهای حل‌کننده برنامه‌ریزی خطی صحیح[3] قادر به حل آن نخواهند بود. در حالی که در روش مبتنی بر حرکت، مسیرها به عنوان ورودی به الگوریتم داده نمی‌شود و خود الگوریتم مسیرهای بهینه را پیدا می‌کند و از نظر زمان اجرا و تعداد

1. Wire-Segment

2. Bin

3. ILP Solver



شکل ۳: حرکت از مبدأ تا مقصد بین خانه‌های Grid.

۱	۲	۳	۴	۵	۶	۷
۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴
۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱

شکل ۴: حفظ شرط پیوستگی و استفاده یک یال در دو جهت.

ورودی‌های مسأله بهتر است.

۳-۱ مسیریابی مبتنی بر حرکت بین خانه‌های Grid

همان طور که در شکل ۳ نشان داده شده است، یک Grid ۳×۷ بر روی QFPGA فرض می‌شود. اگر یک مسیر در یک خانه از Grid ترمینالی داشته باشد، محل ترمینال در مرکز این خانه از Grid فرض می‌شود. در این روش مسیر بین مبدأ و مقصد یک اتصال، با حرکت بین خانه‌های Grid ایجاد می‌شود.

در این روش برای نشان‌دادن محل هر خانه Grid، از یک عدد استفاده می‌کنیم، به این صورت که در یک Grid با m سطر و n ستون، شماره خانه‌ای در سطر i و ستون j از $(i-۱)\times n+j$ به دست می‌آید. متغیر دودویی X با ۳ آرگومان به این صورت تعریف می‌شود که اگر مسیر k–ام از لبه بین دو خانه i و j استفاده کند (از خانه i به خانه j برود)، مقدار $X_{k,i,j}$ برابر یک خواهد بود و در غیر این صورت مقدار آن صفر است. در واقع اگر خانه i و j مجاور باشند، ۱بودن مقدار $X_{k,i,j}$ به این مفهوم است که اتصال k در مسیر حرکت خود از خانه i به خانه j رفته است و کانال بین این دو خانه را اشغال کرده است. باید از حرکت از مبدأ و خارج‌شدن از خانه مبدأ شبکه مطمئن شد، در غیر این صورت خروجی برنامه نامعتبر است چرا که ممکن است مسیری تشکیل شود که این مسیر یک حلقه باشد یا این که مسیر تشکیل‌شده با مبدأ پیوستگی نداشته باشد. برای اِعمال این محدودیت از (۹) استفاده می‌شود. اندیس $k_{(۱)}$ در این رابطه شماره خانه مبدأ را نشان می‌دهد

$$\forall k: \sum_{j=۱}^{m\times n} X_{k,k(۱),j} + \sum_{i=۱}^{m\times n} X_{k,i,k(۱)} = ۱ \qquad (۹)$$

محدودیت ذکرشده در (۹) این مفهوم را هم می‌رساند که مسیر از خانه‌ای که به عنوان مبدأ حرکت تعریف شده باید خارج شود و دیگر این مسیر به این خانه وارد نشود. یعنی مجموع حرکت از مبدأ به خانه‌های مجاور با حرکت از خانه‌های مجاور به مبدأ باید یک باشد تا مسیر بسته در مبدأ شکل نگیرد. در مورد مقصد همانند مبدأ باید دو نکته را در نظر گرفت. اول این که یک مسیر حتماً شامل خانه مقصد باشد یعنی این مسیر نهایتاً به مقصد برسد و در ضمن مسیر از خانه‌های مجاور به خانه مقصد وارد شود و جمع حرکت از خانه‌های مجاور به مقصد و حرکت از مقصد به خانه‌های مجاور آن یک باشد که در این خانه مسیر بسته شکل نگیرد. این محدودیت در (۱۰) آمده است. در این رابطه اندیس $k_{(۲)}$ شماره خانه مقصد را نشان می‌دهد

$$\forall k: \sum_{i=۱}^{m\times n} X_{k,i,k(۲)} + \sum_{j=۱}^{m\times n} X_{k,k(۲),j} = ۱ \qquad (۱۰)$$

نکته بعدی که باید در نظر گرفت این است که پیوستگی مسیر حفظ شود، به این ترتیب که اگر مسیری به خانه‌ای میانی وارد شد حتماً باید از آن خانه خارج شود. در روش مسیریابی مبتنی بر مسیرهای موجود، لازم نبود شرط پیوستگی در نظر گرفته شود چرا که به صورت ضمنی مسیرهای موجود پیوسته هستند. خانه‌های میانی یک مسیر همه خانه‌های Grid به جز مبدأ و مقصد آن مسیر هستند چرا که در مورد مبدأ و مقصد فقط خارج‌شدن یا واردشدن به آن خانه باید حفظ شود. رابطه (۱۱) شرط پیوستگی را به این صورت حفظ می‌کند که اگر مسیری به خانه‌ای وارد شد، حتماً از آن خانه خارج شود

$$\forall k, \forall j: \sum_{i=۱}^{m\times n} X_{k,i,j} = \sum_{i=۱}^{m\times n} X_{k,j,i} \qquad (۱۱)$$

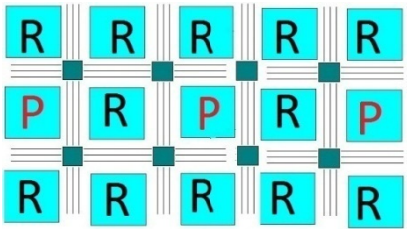
نکته‌ای که باید مد نظر قرار بگیرد، این است که همان طور که در شکل ۴ نشان داده شد، مسیری از خانه شماره ۱۷ به خانه شماره ۱۰ وارد شده و دوباره از خانه شماره ۱۰ به خانه شماره ۱۷ خارج می‌شود، چرا که یک یال دو بار و در دو جهت طی شده است که در این صورت شرط پیوستگی برقرار است ولی یال‌های اضافی در مسیر طی می‌شود.

برای حل مشکل ذکرشده، باید با اعمال محدودیتی یال‌ها را به صورت یک‌طرفه تعریف کرد تا اگر مسیری به خانه‌ای وارد شد، برای خارج‌شدن از آن خانه از همان یال استفاده نکند. اگر در (۱۲) شرط مساوی صدق کند، گویای این مطلب است که اگر یک مسیر از خانه i به خانه j آمد، از خانه j به دلیل شرط پیوستگی باید خارج شود ولی نمی‌تواند به خانه i برگردد. به دلیل این که یک مسیر از تمام خانه‌های یک Grid عبور نمی‌کند، حاصل جمع در محدودیت لحاظ‌شده می‌تواند کمتر از یک باشد

$$\forall k, \forall i, \forall j: [X_{k,i,j} + X_{k,j,i}] \le ۱ \qquad (۱۲)$$

تعداد کانال‌های بین سلول‌های منطقی محدود است، بنابراین با اعمال محدودیتی استفاده از این سیم‌ها باید کنترل شود. پارامتر E_C معرف ظرفیت کانال‌هاست. از آنجایی که مقدار ظرفیت بین تمام خانه‌های Grid باید لحاظ شود، این پارامتر به صورت یک ماتریس تعریف می‌شود که ابعاد آن $(m\times n)\times(m\times n)$ است. این ماتریس یک ماتریس خلوت[1] است چرا که بین هر خانه‌ای فقط با خانه‌های مجاورش کانال وجود دارد. رابطه (۱۳) موظف به کنترل این محدودیت است

1. Sparse



شکل ۵: معماری پیشنهادی داخلی سلول منطقی.

$$\forall i, \forall j : [\sum_{k=1}^{l} X_{k,i,j} + \sum_{k=1}^{l} X_{k,j,i}] \le E\_C_{i,j} \qquad (۱۳)$$

در نهایت مسیرها در مسیریابی باید به نحوی انتخاب شوند که طول سیم بین ترمینال‌های یک مسیر به حداقل برسد. تابع هدف که در (۱۴) آمده است، سعی در به حداقل‌رساندن تعداد سیم‌های استفاده‌شده برای تمام نت‌ها را دارد

$$\min : \sum_{k=1}^{l} \sum_{i=1}^{m \times n} \sum_{j=1}^{m \times n} X_{k,i,j} \qquad (۱۴)$$

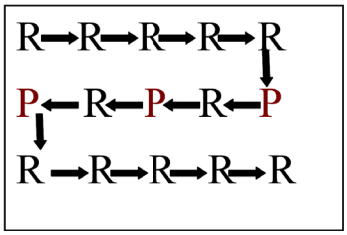
٤- طراحی معماری QFPGA

٤-١ معماری سلول‌های منطقی

منظور از بهینه‌کردن معماری این است که تعیین کنیم در یک سلول منطقی در QFPGA چه تعدادی بلوک پاولی و چه تعدادی بلوک دورانی قرار بگیرد و برای رسیدن به این هدف، الگوریتم‌های جایابی و مسیریابی را برای تعدادی مدار محک پیاده کردیم. با توجه به محدودیت تعداد بلوک‌های درون یک سلول که حداکثر ۱۵ یا ۱۶ بلوک می‌تواند در یک سلول قرار بگیرد چرا که نرم‌افزارهای حل مسایل خطی محدودیت تعداد ورودی دارند. با توجه به نتایج به دست آمده مشخص شد که کدام ترکیب از بلوک‌ها در یک سلول بهترین حالت است.

با بررسی مدارهای کوانتومی این نتیجه به دست آمد که تعدادی گیت پاولی متوالی و دورانی به صورت متوالی و متناوب تکرار می‌شوند و همچنین هر بلوک پاولی قادر به پیاده‌سازی تعداد زیادی گیت‌های پاولی متوالی است و هر بلوک دورانی تنها توانایی پیاده‌سازی یک گیت دورانی را دارد. بنابراین داخل هر سلول منطقی بعد از هر بلوک پاولی چند بلوک دورانی می‌تواند قرار بگیرد. حال باید تعیین کرد که در یک سلول منطقی به ازای هر بلوک پاولی چه تعداد بلوک دورانی و به چه ترتیبی قرار گیرند. طبق نتایج به دست آمده، معمولاً در مدارهای کوانتومی به ازای هر چهار گیت دورانی یک مجموعه گیت پاولی متوالی قرار دارد و از این رو نسبت بلوک‌های پاولی به دورانی در یک سلول منطقی یک به چهار فرض می‌شود. با توجه به محدودیت افزایش تعداد ورودی در الگوریتم‌های پیشنهادی، تعداد بلوک‌های پاولی و دورانی داخل سلول منطقی به ترتیب می‌توانند ۳ و ۱۲ یا ۴ و ۱۲ در نظر گرفته شود.

بعد از تعیین تعداد بلوک پاولی و دورانی در یک سلول منطقی، باید تعیین کرد که این بلوک‌ها به چه ترتیبی قرار بگیرند. به عنوان مثال می‌توان پانزده بلوک را در یک سطر به صورت چیدمان خطی قرار داد و یا به صورت شکل ۵ بلوک‌های دورانی را در اطراف بلوک‌های پاولی قرار داد. در حالت اول برای ارتباط اولین بلوک پاولی با چهارمین بلوک دورانی باید مسیر طولانی‌تری نسبت به شکل ۵ طی کرد. با مقایسه تأخیر مدارهای محک در معماری‌های مختلف، می‌توان نتیجه گرفت که مناسب‌ترین معماری مطابق با شکل ۵ است. در این معماری با توجه به



شکل ۶: ساختار زنجیره‌ای در سلول منطقی.

نتایج جدول ۱ در بخش نتایج، بلوک‌ها در سه سطر قرار می‌گیرند. بلوک‌های پاولی در سطر دوم هستند و بلوک‌های دورانی در اطراف بلوک‌های پاولی قرار می‌گیرند.

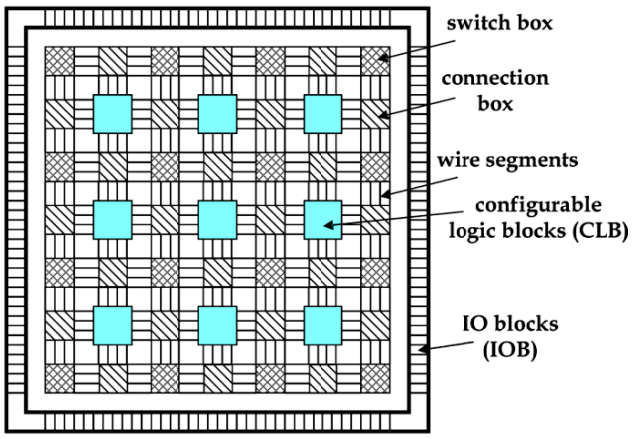
٤-٢ معماری کانال‌های ارتباطی در سلول منطقی

در بخش قبل، تعداد و نوع بلوک‌های هر سلول منطقی مشخص شده است و قرار است در این بخش نوع کانال‌های ارتباطی بین بلوک‌ها و سلول‌ها تعیین شود. برای برقراری ارتباط بین بلوک‌ها از بلوک‌های ارتباطی (CB)[1] و برای برقراری ارتباط بین سلول‌های منطقی از SB[2] استفاده می‌شود. نحوه قرارگیری سوئیچ‌ها و همچنین نحوه ارتباط بین بلوک‌ها و مسیری که کیوبیت‌ها می‌توانند در یک سلول طی کنند، می‌تواند متفاوت باشد. به عنوان مثال یک معماری می‌تواند به صورت شکل ۶ باشد که مسیر بین بلوک‌ها به صورت زنجیره‌ای تعریف شده است. در این معماری نمونه برای ارتباط اولین بلوک در سطر اول با یک بلوک در سطر دوم، کیوبیت مورد نظر باید از تمام بلوک‌های سطر اول عبور کند که باعث افزایش طول سیم و تأخیر می‌شود. اگر هر کیوبیتی بتواند پس از خروج از یک بلوک به هر کدام از بلوک‌های مجاور و یا به بلوک‌های سلول‌های منطقی دیگر بدون طی‌کردن مسیر اضافه حرکت داشته باشد، تأخیر کاهش می‌یابد. به عبارت دیگر، کیوبیت‌ها بتوانند از یک بلوک به چهار بلوک اطرف حرکت داشته باشند. برای این منظور در معماری پیشنهادی برای ارتباط بین بلوک‌ها و سلول‌های منطقی از سوئیچ استفاده می‌شود. در حالت زنجیره‌ای به وجود سوئیچ بین بلوک‌ها احتیاج نیست، اما نتایج نشان می‌دهد تأخیر به مراتب افزایش یافته است. شکل ۶ یک معماری نمونه از نحوه ارتباط بین بلوک‌های یک سلول منطقی می‌باشد. شکل‌های ۷- الف و ۷- ب ساختار داخلی یک FPGA استاندارد را نشان می‌دهد که در این مقاله از همین ساختار استاندارد استفاده شده است. ساختار معماری پیشنهادی QFPGA در شکل ۷- الف نشان داده شده که باکس‌های SB عملیات اتصال سیگنال‌های مسیریابی عمودی و افقی را انجام می‌دهد. ساختار داخلی باکس‌های CB و SB در شکل ۷- الف و ۷- ب نشان داده شده است. همان طور که ملاحظه می‌نمایید سوئیچ‌های SRAM نشان می‌دهد که بلوک‌ها و سلول‌ها می‌توانند به چهار طرف ارتباط داشته باشند.

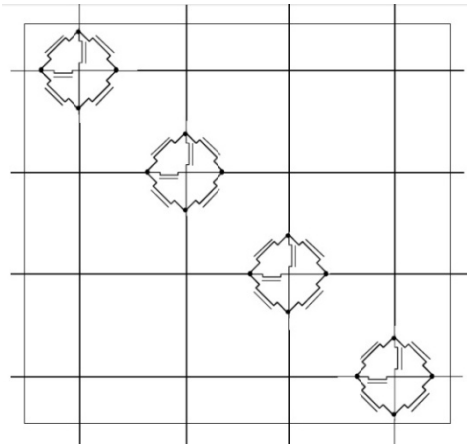
٥- افرازکردن مدار

افراز به این معنا است که کدام گیت‌ها از یک مدار بزرگ کوانتومی در یک بلوک و سلول QFPGA اجرا شوند. به دلیل ارائه معماری جدید برای سلول‌های منطقی، باید مدارها بر اساس معماری پیشنهادی افراز شوند. افرازکردن مدارها بر اساس معماری و تعداد ورودی‌های سلول‌های منطقی می‌تواند متفاوت باشد. به دلیل این که تعداد ورودی‌های بلوک‌های

1. Connection Block
2. Switch Box



(الف)



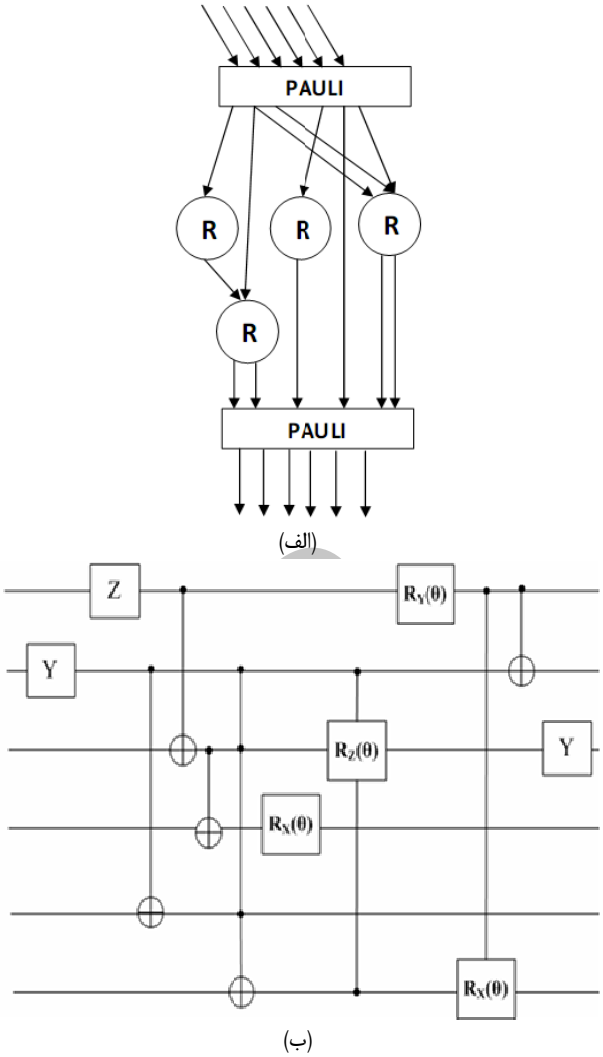
(ب)

شکل ۷: (الف) معماری کلی و درونی QFPGA و (ب) ساختار داخلی SB.

جدول ۱: مقایسه تأخیر و منابع مصرفی معماری‌های متفاوت.

	PRRR PRRR PRRR PRRR	PRRRR PRPRR PRRRR	RPRRR RPRRR RPRRR	PRRRR RRRRP PRRRR	PRRR RRRP PRRR RRRP
۱	$74\,ws+15\,sb+4\,CB$ ۳۷۳۰	$68\,ws+15\,sb+4\,CB$ ۳۶۷۹	$56\,ws+14\,sb+4\,CB$ ۳۵۲۲	$59\,ws+17\,sb+4\,CB$ ۳۷۱۷	$52\,ws+16\,sb+4\,CB$ ۳۶۰۱
۲	$65\,ws+14\,sb+4\,CB$ ۳۵۹۸	$62\,ws+14\,sb+4\,CB$ ۳۵۷۲	$64\,ws+15\,sb+4\,CB$ ۳۶۴۶	$63\,ws+15\,sb+4\,CB$ ۳۶۳۷	$55\,ws+17\,sb+4\,CB$ ۳۶۸۳
۳	$54\,ws+13\,sb+4\,CB$ ۳۴۴۹	$52\,ws+13\,sb+4\,CB$ ۳۴۳۲	$51\,ws+13\,sb+4\,CB$ ۳۴۲۳	$50\,ws+12\,sb+4\,CB$ ۳۳۵۸	$53\,ws+15\,sb+4\,CB$ ۳۵۵۳
۴	$53\,ws+13\,sb+4\,CB$ ۳۴۴۰	$51\,ws+13\,sb+4\,CB$ ۳۴۲۳	$55\,ws+15\,sb+4\,CB$ ۳۵۷۰	$55\,ws+14\,sb+4\,CB$ ۳۵۱۳	$48\,ws+13\,sb+4\,CB$ ۳۳۹۸
۵	$77\,ws+15\,sb+4\,CB$ ۳۷۵۵	$75\,ws+15\,sb+4\,CB$ ۳۷۳۸	$53\,ws+13\,sb+4\,CB$ ۳۴۴۰	$73\,ws+16\,sb+4\,CB$ ۳۷۷۸	$63\,ws+15\,sb+4\,CB$ ۳۶۳۷
۶	$63\,ws+14\,sb+4\,CB$ ۳۵۸۱	$60\,ws+13\,sb+4\,CB$ ۳۴۹۹	$58\,ws+14\,sb+4\,CB$ ۳۵۳۹	$57\,ws+12\,sb+4\,CB$ ۳۴۱۷	$62\,ws+14\,sb+4\,CB$ ۳۵۷۲
۷	$52\,ws+13\,sb+4\,CB$ ۳۴۳۲	$50\,ws+12\,sb+4\,CB$ ۳۳۵۸	$44\,ws+11\,sb+4\,CB$ ۳۲۵۱	$48\,ws+9\,sb+4\,CB$ ۳۱۷۲	$49\,ws+11\,sb+4\,CB$ ۳۲۹۳
۸	$53\,ws+13\,sb+4\,CB$ ۳۴۴۰	$51\,ws+13\,sb+4\,CB$ ۳۴۲۳	$55\,ws+13\,sb+4\,CB$ ۳۴۵۷	$51\,ws+11\,sb+4\,CB$ ۳۳۱۰	$50\,ws+12\,sb+4\,CB$ ۳۳۵۸
۹	$58\,ws+14\,sb+4\,CB$ ۳۵۳۹	$54\,ws+13\,sb+4\,CB$ ۳۴۴۹	$66\,ws+14\,sb+4\,CB$ ۳۶۰۶	$57\,ws+12\,sb+4\,CB$ ۳۴۱۷	$56\,ws+13\,sb+4\,CB$ ۳۴۶۵
۱۰	$75\,ws+15\,sb+4\,CB$ ۳۷۳۸	$72\,ws+15\,sb+4\,CB$ ۳۷۱۳	$70\,ws+15\,sb+4\,CB$ ۳۶۹۶	$65\,ws+13\,sb+4\,CB$ ۳۵۴۱	$73\,ws+16\,sb+4\,CB$ ۳۷۷۸
۱۱	$76\,ws+16\,sb+4\,CB$ ۳۸۰۳	$69\,ws+16\,sb+4\,CB$ ۳۷۴۴	$57\,ws+17\,sb+4\,CB$ ۳۷۰۰	$60\,ws+11\,sb+4\,CB$ ۳۳۸۶	$53\,ws+10\,sb+4\,CB$ ۳۲۷۱
۱۲	$65\,ws+14\,sb+4\,CB$ ۳۵۹۸	$62\,ws+13\,sb+4\,CB$ ۳۵۹۸	$65\,ws+14\,sb+4\,CB$ ۳۵۹۸	$63\,ws+10\,sb+4\,CB$ ۳۳۵۵	$56\,ws+15\,sb+4\,CB$ ۳۵۷۸
۱۳	$54\,ws+13\,sb+4\,CB$ ۳۵۰۵	$52\,ws+15\,sb+4\,CB$ ۳۵۴۵	$46\,ws+13\,sb+4\,CB$ ۳۳۸۱	$50\,ws+11\,sb+4\,CB$ ۳۳۰۲	$51\,ws+12\,sb+4\,CB$ ۳۳۶۷
۱۴	$78\,ws+15\,sb+4\,CB$ ۳۷۶۴	$75\,ws+14\,sb+4\,CB$ ۳۶۸۲	$73\,ws+14\,sb+4\,CB$ ۳۶۶۵	$68\,ws+16\,sb+4\,CB$ ۳۷۳۶	$76\,ws+17\,sb+4\,CB$ ۳۸۶۰
میانگین	$64\,ws+14\,sb+4\,CB$ ۳۵۹۸	$61\,ws+14\,sb+4\,CB$ ۳۵۵۵	$58\,ws+13\,sb+4\,CB$ ۳۵۳۵	$58\,ws+12\,sb+4\,CB$ ۳۴۷۴	$56\,ws+14\,sb+4\,CB$ ۳۵۳۰

موجود ۴ کیوبیت است، می‌توان تعداد ورودی‌های سلول‌های منطقی را ۴ کیوبیت در نظر گرفت و یا به دلیل این که هر سلول منطقی در معماری پیشنهادی شامل سه سطر از بلوک‌های ۴ ورودی است، می‌توان تعداد ورودی‌های سلول‌های منطقی را ۱۲ کیوبیت فرض کرد. برای هر دو



شکل ۸: (الف) یک نمونه از مدار کوانتومی و (ب) استخراج DFG از مدار کوانتومی.

حالت یک روش برای افرازکردن ارائه می‌شود و در فصل آینده این دو حالت با هم مقایسه شده‌اند. در ضمن در این رساله برای افرازکردن از روش‌های مکاشفه‌ای استفاده شده است.

در افراز مدارهای کوانتومی می‌توان دو هدف را در نظر گرفت که باید بین این دو هدف تعادل[1] برقرار کرد. اول این که افراز به گونه‌ای باشد که تعداد گیت‌های بیشتری داخل هر سلول منطقی پیاده‌سازی شود. با این توجیه که از کانال‌های داخل سلول منطقی که تأخیر کمتری در مقایسه با کانال‌های بین سلول‌های منطقی دارند، بیشتر استفاده شود و در نهایت مساحت مصرفی و تأخیر کمتر شوند. در این حالت ممکن است به دلیل استفاده حداکثری از منابع داخل سلول منطقی، کیوبیت‌ها از سلول‌های منطقی بیشتری عبور کنند.

ثانیاً افراز به گونه‌ای باشد که تأخیر کلی مدار به حداقل برسد. یعنی سعی شود یک کیوبیت از ابتدا تا انتهای مدار از تعداد سلول منطقی کمتری عبور کند تا تأخیر کیوبیت‌ها کمتر شود. به دلیل وجود این محدودیت که تعداد ورودی‌های یک سلول منطقی بیش از ۴ کیوبیت نباشد، از منابع موجود در سلول منطقی بهینه استفاده نمی‌شود که به افزایش مساحت[2] و تأخیر منجر می‌شود.

با حذف این محدودیت و با فرض این که تعداد ورودی هر سلول منطقی به دلیل داشتن سه سطر از بلوک‌های ۴ کیوبیتی، ۱۲ کیوبیت باشد، تعداد گیت‌هایی که داخل یک افراز قرار می‌گیرند را می‌توان افزایش داد و از منابع داخل سلول منطقی بهینه‌تر استفاده کرد. نتیجه تغییر تعداد ورودی این خواهد بود که از کانال‌های داخل سلول منطقی که نسبت به کانال‌های موجود بین سلول‌های منطقی تأخیر کمتری دارند، بیشتر استفاده می‌شود و تأخیر کاهش می‌یابد.

شکل ۸- الف نمونه‌ای از مدار کوانتومی و شکل ۸- ب نحوه استخراج DFG از مدار کوانتومی و نحوه ارتباط بین بلوک‌ها را نشان می‌دهد.

برای افراز مدارهایی با حداکثر ۱۲ کیوبیت به روش زیر می‌توان عمل کرد. در این روش افراز، حرکت در مدار به صورت افقی انجام می‌شود. علی رغم این که تعداد ورودی‌های سلول‌های منطقی ۱۲ کیوبیت در نظر گرفته شده است، ولی این محدودیت که هر بلوک داخل سلول منطقی قادر به پیاده‌سازی گیت‌های حداکثر با ۴ ورودی است، وجود دارد. به همین دلیل با توجه به مدار ورودی، سعی می‌شود تا جایی که ممکن است کیوبیت به افراز اضافه نشود و با افزایش تعداد گیت‌هایی که کیوبیت مشترک دارند، تأخیر کیوبیت‌ها به حداقل برسد. به عبارت دیگر سعی می‌شود گیت‌های۴ کیوبیتی که وارد یک بلوک در یک سلول منطقی می‌شوند، تمام بلوک‌های آن سلول را اشغال کنند.

در این روش مدار کوانتومی از ابتدا تا انتها سطح‌بندی می‌شود. سپس یک گیت ۴ کیوبیتی که در کمترین سطح قرار دارد به عنوان یک افراز در نظر گرفته می‌شود و به سمت ابتدای مدار حرکت می‌کند. در این با رسیدن افراز به یک گیت که علاوه بر ۴ کیوبیت موجود در افراز، این گیت کیوبیت دیگری را هم شامل می‌شود، گیت مذکور و کیوبیت‌های آن وارد افراز می‌شوند، البته اگر محدودیت ۱۲ کیوبیت ورودی سلول منطقی نقض نشود. با رسیدن افراز به ابتدای مدار و یا یکسان‌شدن تعداد گیت‌های داخل افراز با تعداد منابع موجود در سلول منطقی و یا رسیدن به یک افراز دیگر، حرکت به سمت ابتدای مدار پایان می‌گیرد و شرط خاتمه حرکت افراز برقرار می‌شود. اگر شرط اتمام حرکت به سمت ابتدای مدار، چیزی غیر از اتمام منابع بود، افراز از گیت ۴ کیوبیتی به سمت انتهای مدار حرکت می‌کند تا جایی که یکی از سه شرط خاتمه حرکت افراز برقرار شود. به همین ترتیب گیت‌های ۴ کیوبیتی دیگر افراز می‌شوند. بعد از این که تمام گیت‌های ۴ کیوبیتی داخل افرازها قرار گرفتند، به ترتیب گیت‌های ۳، ۲ و تک کیوبیتی که هنوز داخل افراز قرار نگرفته‌اند به همین روش افراز می‌شوند.

ممکن است یک افراز در حین حرکت به سمت ابتدا یا انتهای مدار به افراز دیگری برسد و دیگر نتواند به صورت افقی حرکت کند، ولی هنوز تعداد گیت‌های داخل افراز از تعداد بلوک‌های سلول منطقی کمتر باشد و منابع استفاده‌نشده داخل سلول منطقی موجود باشد، در این صورت می‌توان به دو صورت عمل کرد. در اولین حالت اگر نتوان به صورت افقی حرکت کرد، می‌توان افراز را به صورت عمودی به سمت بالا یا پایین گسترش داد. به عبارت دیگر با اضافه‌کردن کیوبیت‌ها و گیت‌های دیگر به افراز از منابع داخل سلول منطقی بیشتر استفاده می‌شود. در حالت دوم افراز مذکور به اتمام می‌رسد و دیگر گسترش نمی‌یابد و افزارکردن مدار ادامه می‌یابد. بعد از این که همه گیت‌ها در افراز قرار گرفتند و یک افراز ابتدایی انجام شد، در یک پردازش نهایی می‌توان این افرازها را با هم ادغام کرد، اگر مجموع گیت‌های تعدادی از افرازها از تعداد منابع سلول منطقی بیشتر نباشد. در نهایت اگر هنوز افرازهایی با تعداد گیت‌های کمی وجود داشت، با شیفت‌دادن این گیت‌ها به سمت انتهای مدار سعی بر استفاده حداکثری از منابع داخل سلول منطقی می‌شود. به عبارت دیگر

1. Trade-Off
2. Area

جدول ۲: مقایسه تأخیر و منابع مصرفی معماری‌های متفاوت.

معماری / مدار	PRRRR RRPRR RRRRP	RPRRR RRRPR RPRRR	RRRRR PRPRP RRRRR
۱	$۵۱ws+۱۳sb+۴CB$ ۳۴۲۳	$۵۴ws+۱۲sb+۴CB$ ۳۳۹۲	$۵۱ws+۱۲sb+۴CB$ ۳۳۶۷
۲	$۵۹ws+۱۴sb+۴CB$ ۳۵۴۷	$۵۳ws+۱۴sb+۴CB$ ۳۴۹۷	$۵۲ws+۱۳sb+۴CB$ ۳۴۳۲
۳	$۵۰ws+۱۲sb+۴CB$ ۳۳۵۸	$۴۷ws+۱۲sb+۴CB$ ۳۳۳۳	$۴۵ws+۱۴sb+۴CB$ ۳۴۲۹
۴	$۵۱ws+۱۴sb+۴CB$ ۳۴۸۰	$۴۸ws+۱۰sb+۴CB$ ۳۲۲۹	$۵۳ws+۱۱sb+۴CB$ ۳۳۲۷
۵	$۵۸ws+۱۳sb+۴CB$ ۳۴۸۲	$۵۸ws+۱۱sb+۴CB$ ۳۳۶۹	$۵۹ws+۱۰sb+۴CB$ ۳۳۲۱
۶	$۵۹ws+۱۱sb+۴CB$ ۳۳۷۸	$۴۶ws+۱۰sb+۴CB$ ۳۲۱۲	$۴۳ws+۹sb+۴CB$ ۳۱۳۰
۷	$۴۱ws+۱۲sb+۴CB$ ۳۲۸۳	$۳۷ws+۱۱sb+۴CB$ ۳۱۹۲	$۴۲ws+۸sb+۴CB$ ۳۰۶۵
۸	$۴۴ws+۱۳sb+۴CB$ ۳۳۶۴	$۴۶ws+۱۵sb+۴CB$ ۳۴۹۴	$۴۵ws+۱۳sb+۴CB$ ۳۳۷۳
۹	$۵۴ws+۱۴sb+۴CB$ ۳۵۰۵	$۵۵ws+۱۳sb+۴CB$ ۳۴۵۷	$۵۸ws+۱۱sb+۴CB$ ۳۳۶۹
۱۰	$۶۲ws+۱۶sb+۴CB$ ۳۶۸۵	$۶۵ws+۱۵sb+۴CB$ ۳۶۵۴	$۶۳ws+۱۲sb+۴CB$ ۳۴۶۸
۱۱	$۵۲ws+۱۵sb+۴CB$ ۳۵۴۵	$۵۵ws+۱۴sb+۴CB$ ۳۵۱۳	$۵۲ws+۱۰sb+۴CB$ ۳۲۶۲
۱۲	$۶۰ws+۱۵sb+۴CB$ ۳۶۱۲	$۵۴ws+۱۰sb+۴CB$ ۳۲۷۹	$۵۳ws+۱۱sb+۴CB$ ۳۳۲۷
۱۳	$۴۳ws+۱۳sb+۴CB$ ۳۳۵۶	$۴۲ws+۱۵sb+۴CB$ ۳۴۳۵	$۴۴ws+۱۲sb+۴CB$ ۳۳۰۸
۱۴	$۶۵ws+۱۴sb+۴CB$ ۳۵۹۸	$۶۸ws+۱۶sb+۴CB$ ۳۷۳۶	$۶۶ws+۱۳sb+۴CB$ ۳۵۵۰
میانگین	$۵۳٫۵ws+۱۲٫۵sb+۴CB$ ۳۴۷۳	$۵۲٫۳ws+۱۱sb+۴CB$ ۳۴۱۴	$۵۱ws+۱۰sb+۴CB$ ۳۳۳۸

گیت‌های این افرازها به درون افراز مجاور که در سمت انتهای مدار قرار دارد، وارد می‌شوند و به همان نسبت گیت‌هایی از انتهای افراز مجاور خارج می‌شوند تا تعداد گیت‌های موجود در افراز بیشتر از تعداد منابع سلول منطقی نشود. این کار ادامه می‌یابد تا در نهایت گیت‌هایی که هنوز افراز نشده‌اند در انتهای مدار قرار بگیرند. به عبارت دیگر بعضی از افرازها افراز مجدد[1] می‌شوند تا گیت‌های افرازنشده در انتهای مدار قرار بگیرند. در این مقاله به این حالت عمل شده است.

۶- نتایج آزمایش‌ها

برای حل برنامه‌ریزی خطی صحیح مطرح‌شده در بحث جایابی و مسیریابی، در این مقاله ابتدا از نرم‌افزار LINGO استفاده شد که به دلیل عدم کارایی در مواجهه با افزایش تعداد ورودی مسأله، کنار گذاشته شد و به جای آن از نرم‌افزار Cplex استفاده شد. لازم به توضیح است که نرم‌افزارهای حل مسایل برنامه‌ریزی خطی صحیح محدودیت افزایش تعداد ورودی دارند، ولی این محدودیت برای همه این نرم‌افزارها یکسان نیست، مثلاً در نرم‌افزار Cplex حداکثر تعداد ورودی نسبت به LINGO بالاتر است. در مورد زمان لازم برای اجرای نرم‌افزار باید اشاره کرد که حل مسأله برنامه‌ریزی خطی تا سقف ۱۶ عدد ورودی حدود ۱ ساعت طول می‌کشد که با افزایش تعداد ورودی از ۱۶، مدت خروجی اجرای برنامه پس از گذشت ۱ روز هم مشخص نشد. پس از تعریف الگوریتم‌های جایابی و مسیریابی و تعیین سقف تعداد ورودی این دو الگوریتم، چنانچه در جداول ۱ و ۲ مشاهده می‌شود، معماری‌های متفاوت با هم مقایسه شدند و در نهایت با توجه به هزینه منابع مصرفی و تأخیرهای به دست آمده از هر معماری، معماری مورد نظر مطابق آنچه در شکل ۵ آمده است به عنوان معماری بهینه بازتعریف گردید. معماری‌های مورد مقایسه معماری‌هایی هستند که نحوه قرارگرفتن گیت‌های پاولی و دورانی در یک سلول را نشان می‌دهند. جایابی و مسیریابی برای ۱۴ مدار ۴ کیوبیتی بر اساس انواع معماری‌های مختلف با الگوریتم‌های پیشنهادی انجام شد که این مدارها شامل گیت‌های پاولی و دورانی هستند. از آنجایی که تعداد مدار محکی که فقط شامل گیت دورانی و پاولی باشد اندک است، از مدارهای شامل این دو گیت که به صورت تصادفی تولید شده‌اند هم

1. Re-Partitioning

جدول ۳: مقایسه کانال‌های مصرفی معماری [۱۲] و معماری پیشنهادی با ۱۲ کیوبیت ورودی.

	RRRRR PRPRP RRRRR	PRR
۱	$۲۵۵\,ws + ۴۹\,CB + ۱۲\,SB + ۶۰\,sb$	$۳۰۰\,ws + ۱۰\,sb + ۱۲۹\,CB + ۳۶\,SB$
۲	$۲۹۷\,ws + ۵۳\,CB + ۱۷\,SB + ۷۲\,sb$	$۳۲۴\,ws + ۷\,sb + ۱۷۳\,CB + ۶۸\,SB$
۳	$۳۲۶\,ws + ۴۶\,CB + ۷\,SB + ۴۹\,sb$	$۲۹۱\,ws + ۹\,sb + ۱۴۸\,CB + ۴۶\,SB$
۴	$۲۶۳\,ws + ۶۴\,CB + ۲۵\,SB + ۵۸\,sb$	$۲۱۰\,ws + ۱۵\,sb + ۱۶۵\,CB + ۴۳\,SB$
۵	$۳۳۹\,ws + ۲۷\,CB + ۴\,SB + ۶۳\,sb$	$۲۳۴\,ws + ۱۳\,sb + ۲۳۸\,CB + ۶۴\,SB$
۶	$۳۲۱\,ws + ۴۳\,CB + ۱۲\,SB + ۴۷\,sb$	$۲۰۴\,ws + ۷\,sb + ۱۲۱\,CB + ۳۲\,SB$
۷	$۳۷۲\,ws + ۷۰\,CB + ۱۸\,SB + ۸۶\,sb$	$۳۰۳\,ws + ۶\,sb + ۱۵۰\,CB + ۳۳\,SB$
میانگین	$۳۱۰٫۴۲\,ws + ۶۲٫۱۴\,sb + ۵۰٫۲۸\,CB + ۱۳٫۵۷\,SB$	$۲۶۶٫۵۷\,ws + ۹٫۵۷\,sb + ۱۶۰٫۷۵\,CB + ۴۶\,SB$

استفاده شده است و تفاوت آنها در نوع این گیت‌ها و نحوه قرارگرفتن آنها در مدار است. نمونه‌ای از این مدارهای محک در [۱۲] آمده است. سپس تأخیر و تعداد کانال‌های مسیریابی مصرف‌شده که معرف بخشی از مساحت اشغال‌شده توسط هر مداری است، طبق جداول ۱ و ۲ به دست آمد. تعداد کانال‌های اشغال‌شده در هر مدار، به دلیل تعداد برابر بلوک‌های مصرفی هر مدار می‌تواند معیار مناسبی برای مقایسه مساحت اشغالی توسط هر مدار باشد. در این جداول ws، sb، SB و CB هر کدام به ترتیب نشان‌دهنده سیم‌ها و سوئیچ‌های داخلی یک سلول منطقی و سوئیچ‌ها و بلوک‌های اتصالی بین سلول‌های منطقی هستند و سطر اول این جداول معرف معماری‌های مختلف است. عدد اول در هر خانه از جدول تعداد منابع اشغالی هر مدار و عدد دوم معرف تأخیر مسیر بحرانی هر مدار بر حسب نانوثانیه است.

همان طور که ملاحظه می‌شود بین معماری‌های مختلف با توجه به نتایج به دست آمده می‌توان دید که در یکی از معماری‌ها تأخیر مدار برابر ۳۴۷۴ نانوثانیه است و همچنین تعداد سوئیچ‌ها و کانال‌های مصرفی نیز نسبت به دیگر معماری‌ها کمتر است. طبق نتایج به دست آمده، معماری ارائه‌شده در ستون سوم جدول ۲ تأخیر کمتر و منابع مصرفی کمتری نسبت به معماری‌های دیگر دارد.

در اینجا به بررسی نتایج مقایسه افراز بر اساس سلول‌های منطقی با ۴ و ۱۲ کیوبیت ورودی می‌پردازیم. برای تعیین تعداد ورودی، ۶ مدار هر مدار با ۱۲ کیوبیت و عبور هر کیوبیت به طور متوسط از ۲۰ گیت و ۱ مدار با ۱۲ کیوبیت و عبور هر کیوبیت به طور متوسط از ۳۰ گیت انتخاب شدند. مدارها هر کدام بر اساس معماری پیشنهادی و هم بر اساس معماری پیشنهادشده در [۱۲] در دو حالت سلول‌های منطقی ۴ و ۱۲ کیوبیتی افراز شدند. جدول ۳ مقایسه کانال‌های مصرفی معماری [۱۲] و معماری پیشنهادی با ۱۲ کیوبیت ورودی را نشان می‌دهد.

تأخیر این مدارها باید با تأخیر در همانندسازی مدارهای کوانتومی مقایسه شود و نه با تأخیر در پیاده‌سازی مدارهای دیجیتال. در مقایسه با تأخیر همانندسازی مدارهای کوانتومی مناسب به نظر می‌رسد. تکنولوژی مورد بحث و مورد پیاده‌سازی تکنولوژی‌های کوانتومی نیست، بلکه تکنولوژی‌های دیجیتال همانند CMOS است.

همان طور که ملاحظه می‌شود به طور میانگین در معماری پیشنهادی منابع مصرفی نسبت به منابع مصرفی در معماری [۱۲] کمتر است و معماری پیشنهادی بهینه‌تر عمل می‌کند.

۷- نتیجه‌گیری

در این مقاله افراز، جایابی و مسیریابی مدارهای کوانتومی توضیح داده شدند. ابتدا مدار کوانتومی توسط روش مکاشفه‌ای ارائه‌شده بر اساس تعداد بلوک‌های داخل سلول منطقی افراز می‌شود. سپس جایابی که به صورت برنامه‌ریزی خطی صحیح توضیح داده شد، بر اساس تعداد اتصالات بین افرازها و فاصله فیزیکی سلول‌های منطقی از یکدیگر، مشخص می‌کند که هر کدام از افرازها در کدام سلول منطقی قرار بگیرند. بعد از آن دوباره از برنامه جایابی استفاده می‌شود تا مشخص شود که هر گیت پاولی و دورانی به ترتیب در کدام بلوک پاولی و دورانی داخل سلول منطقی اجرا شود. در واقع از جایابی به صورت سلسله مراتبی در دو سطح با ریزدانگی در سطح سلول‌های منطقی و سپس ریزدانگی در سطح بلوک استفاده می‌شود. با این تفاوت که در حالت اول منابع فقط از یک نوع یعنی سلول منطقی و در حالت دوم منابع از دو نوع یعنی بلوک پاولی و دورانی هستند. بعد از این که مشخص شد هر گیت در کدام بلوک قرار می‌گیرد، توسط الگوریتم مسیریابی، مسیرهای بین بلوک‌های داخل سلول منطقی مشخص می‌شوند و در نهایت با استفاده از همین الگوریتم برای تعیین مسیرهای بین سلول‌های منطقی استفاده می‌شود. الگوریتم مسیریابی نیز همانند الگوریتم جایابی به صورت سلسله مراتبی در دو سطح، یکی سطح سلول منطقی و دیگری سطح QFPGA مورد استفاده قرار می‌گیرد.

مراجع

[1] S. Imre and F. Balazs, *Quantum Computing and Communications*, John Wiley & Sons, 2005.

[2] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Schor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Physical Review*, vol. 52, no. 5, pp. 3457-3467, Nov. 1995.

[3] M. Udresco, L. Prodan, and M. Vladutiu, "Using HDLs for describing quantum circuits: a framework for efficient quantum algorithm simulation," in *Proc. 1st ACM Conf. on Computing Frontiers*, vol. 44, pp. 96-110, Apr. 2004.

[4] M. Fujishima, K. Saito, M. Onouchi, and H. Hoh, "High - speed processor for quantum - computing emulation and its applications," in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 4, pp. 884-887, May 2003.

[5] M. Fujishima, "FPGA-based high - speed emulator of quantum computing," in *Proc. IEEE Int. Conf. on Field-Programmable Technology*, vol. 4, pp. 21-26, Dec. 2003.

[6] J. P. Hayes and I. L. Markov, *Simulation, Synthesis, and Testing of Quantum Circuits*, DARPA QuIST Annual Research Review, Beverly Hills, 2003.

[7] Stratix Device Handbook, vol. 1, http://www.altra.com

[8] A. Glassner, Quantum Computing Part 2, http://www.glassner.com, Sep. 2001.

[9] A. Abdollahi and M. Pedram, "Decision diagram-based representation and recursive bi-decomposition of quantum logic functions," *IEEE Trans. on Computers, Under Review*.

[10] J. Swartz, V. Betz, and J. Rose, "A fast routability - driven router for FPGAs," in *Proc. Int. Symp. on Field Programmable Gate Arrays* pp. 140-149, Monterey, CA, US, Mar. 1998.

[11] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, Springer, 3rd Edition, pp. 79-90, 2008.

[۱۲] م. امینیان، *همانندسازی مدارهای کوانتومی با استفاده از FPGA*، پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی امیرکبیر، دانشکده مهندسی کامپیوتر و فن‌آوری اطلاعات، ۱۳۸۷.

[13] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Improving gate-evel simulation of quantum circuits," *Quantum Information Processing*, vol. 2, no. 5, pp. 347-380, Sep. 2003.

[14] A. U. Khalid, Z. Zilic, and K. Radecka, "FPGA emulation of quantum circuits," in *Proc. IEEE Intl Conf. on Computer Design*, pp. 310-315, 11-13 Oct. 2004.

[15] Wikipedia, "Timeline of quantum computers," Jul. 2007.

مصطفی حیدرزاده کلهرودی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد کامپیوتر گرایش سخت افزار به‌ترتیب در سال‌های ۱۳۸۱ و ۱۳۸۷ به‌ترتیب از دانشگاه‌های شاهد و صنعتی امیر کبیر به پایان رسانده است و هم‌اکنون در حال تحصیل در مقطع دکتری فناوری اطلاعات در دانشگاه سراسری قم می‌باشد. نام‌برده از سال ۱۳۸۸ در مرکز تحقیقات مخابرات ایران (پژوهشگاه فضای مجازی) مشغول به فعالیت بوده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری و شبکه‌های نسل آینده، امنیت شبکه داده‌ها، داده‌کاوی و پردازش‌های کوانتومی.

محمد دانایی‌فر در سال ۱۳۸۳ مدرک کارشناسی مهندسی برق خود را از دانشگاه شاهد و در سال ۱۳۹۰ مدرک کارشناسی ارشد مخابرات خود را از دانشگاه صنعتی خواجه نصیرالدین طوسی اخذ نمود و هم‌اکنون دانشجوی دکتری دانشگاه خواجه نصیرالدین طوسی در رشته مخابرات و در گرایش میدان است. عمده فعالیت‌های پژوهشی ایشان در حوزه ادوات پنهان‌ساز، EMC، کوانتوم الکترونیک و استفاده از فراماد و گرافن در ادوات مخابراتی فرکانس بالا بوده است. ایشان در ارتباط با حوزه‌های مذکور ۸ مقاله در مجلات معتبر و ۳ مقاله در کنفرانس‌های داخلی منتشر و ارائه نموده است.