طراحی ضرب‌کننده‌های سری تشخیص دهنده خطا در منطق برگشت‌پذیر

فرشید اسلامی چلندر[1]، مجتبی ولی‌نتاج[2]، حمید جزایری[3]

[1] کارشناسی ارشد معماری کامپیوتر، دانشگاه صنعتی نوشیروانی بابل

[2] استادیار دانشکده برق و کامپیوتر، دانشگاه صنعتی نوشیروانی بابل، m.valinataj@nit.ac.ir

[3] استادیار دانشکده برق و کامپیوتر، دانشگاه صنعتی نوشیروانی بابل

تاریخ دریافت: ۱۳۹۵/۰۴/۰۶ تاریخ پذیرش: ۱۳۹۵/۰۷/۱۷

چکیده

یکی از چالش برانگیزترین مشکلات در طراحی مدارهای الکترونیکی، مسأله‌ی اتلاف انرژی است. منطق برگشت‌پذیر یکی از راه‌های کاهش اتلاف انرژی است. در این مقاله، سه طرح برای ضرب‌کننده‌های تحمل‌پذیر اشکال از نوع سری با قابلیت تشخیص خطا پیشنهاد می‌شود. در طرح اول، ضرب‌کننده‌ی علامت‌دار مبتنی بر الگوریتم بوث پایه بررسی شده و بر اساس مدارهای مطرح شده تاکنون، طرح بهینه‌ای برای این نوع ضرب‌کننده ارائه می‌شود. در طرح دوم، ضرب‌کننده‌ی علامت‌داری بر پایه‌ی روشی موسوم به الگوریتم K ارائه می‌گردد که در آن پیچیدگی محاسباتی و هزینه‌ی کوانتومی نسبت به روش بوث تا حد زیادی کاهش می‌یابد. در آخرین طرح نیز یک ضرب‌کننده‌ی سری برای ضرب اعداد بدون علامت با روش جمع و شیفت پیشنهاد خواهد شد. مقایسه ضرب‌کننده‌های پیشنهادی با طرح‌های مشابه، با توجه به معیارهای مقایسه‌ی مدارهای برگشت‌پذیر مانند هزینه‌ی کوانتومی، تعداد خروجی‌های بلااستفاده، تأخیر، تعداد گیت‌ها و پیچیدگی محاسباتی، بهبودهای مناسبی را نسبت به ضرب‌کننده‌های موجود نشان می‌دهد.

کلیدواژه

منطق برگشت‌پذیر، ضرب کننده سری، الگوریتم بوث، تحمل‌پذیری اشکال، تشخیص خطا، نگهدارنده توازن.

مقدمه

در دهه‌های اخیر، مدارهای غیر برگشت‌پذیر که از گیت‌های غیر برگشت‌پذیر ساخته می‌شوند، شاکله‌ی مدارهای VLSI بوده‌اند. لانداور [۱] نشان داده است مدارهای غیر برگشت‌پذیر $kT \times ln2$ ژول انرژی مصرف می‌کنند که در آن k ثابت بولتزمن و T دمای مطلق است. مدارهای برگشت‌پذیر[1] راه‌حل مناسبی برای رفع مشکل اتلاف انرژی هستند؛ به طوری که در حالت ایده‌آل در این مدارها هیچ اتلاف انرژی‌ای وجود ندارد [۲]. این عدم اتلاف انرژی به دلیل وجود نگاشت یک‌به‌یک بین ورودی‌ها و خروجی‌ها و در نتیجه عدم از دست‌رفتن اطلاعات حاصل می‌شود. منطق برگشت‌پذیر در صورت رفع مشکلات مربوط به ساخت مدارهای کوانتومی، می‌تواند کاربردهای زیادی در طراحی مدارهای با توان پایین، محاسبات کوانتومی و فناوری نانو داشته باشد.

تحمل‌پذیری اشکال[2] عاملی مهم در طراحی مدارهای VLSI است. بدون در نظر گرفتن برگشت‌پذیری یا برگشت‌ناپذیری، مدارها باید در صورت رخداد خطا در برخی اجزا، بتوانند به عملکرد صحیح خود ادامه دهند. در مدارهای برگشت‌ناپذیر، استفاده از بیت توازن یک روش مرسوم برای تشخیص خطا است. در منطق برگشت‌پذیر، به دلیل این که فن‌آوت[3] و حلقه[4] به صورت بازخورد[5] مجاز نیست، پیاده‌سازی این روش مشکل‌تر است [۳]. به منظور تشخیص خطا در این مدارها، در [۴] ایده‌ی گیت‌های برگشت‌پذیر تحمل‌پذیر اشکال مطرح گردید که در آن‌ها وزن همینگ[6] ورودی‌ها با وزن

[1] Reversible circuits

[2] Fault tolerance

[3] Fan out

[4] loop

[5] Feedback

[6] Hamming weight

Archive of SID
www.SID.ir

همینگ خروجی‌ها برابر است. به این نوع گیت‌ها، گیت‌های نگهدارنده توازن[۷] گفته می‌شود.

در طراحی واحدهای پردازشی، عملیات جمع و ضرب به دلیل کاربرد زیادشان بسیار پر اهمیت هستند، به همین دلیل برای دستیابی به عملکردی مناسب‌تر در پردازنده‌ها، بایستی از مدارهای جمع‌کننده و ضرب‌کننده با ویژگی‌های بهتر بهره برد. در این مقاله، طرح‌هایی جدید برای ضرب‌کننده‌های سری با قابلیت تشخیص خطا ارائه می‌گردد که در مقایسه با طرح‌های موجود از ویژگی‌های مناسب‌تری برخوردارند. این مقاله شامل بخش‌های زیر است: در بخش دوم، تعاریف اولیه و گیت‌های پایه در منطق برگشت‌پذیر معرفی می‌شوند. پژوهش‌های انجام شده تاکنون برای طراحی ضرب‌کننده‌های برگشت‌پذیر تحمل‌پذیر اشکال در بخش سوم بررسی می‌شوند. در بخش چهارم، طرح‌های پیشنهادی برای ضرب‌کننده‌های سری برگشت‌پذیر با قابلیت تشخیص خطا ارائه می‌شود. در بخش پنجم، کارایی ضرب‌کننده‌های پیشنهادی در مقایسه با ضرب‌کننده‌های موجود بررسی شده و در بخش انتهایی نتیجه‌گیری مقاله بیان می‌شود.

تعاریف اولیه و گیت‌های پایه

تعاریف اولیه

در این بخش تعاریف پایه برای گیت برگشت‌پذیر، گیت تحمل‌پذیر اشکال، خروجی بلااستفاده[۸]، هزینه‌ی کوانتومی[۹]، تأخیر[۱۰] و پیچیدگی محاسباتی[۱۱] ارائه می‌گردد.

گیت برگشت‌پذیر مداری دارای n ورودی و n خروجی است (n×n) که به ازای هر بردار ورودی، بردار منحصربه‌فردی در خروجی ظاهر می‌شود. با فرض این که در یک گیت برگشت‌پذیر n×n، بردار ورودی $I_v = (I_0,I_1,\ldots,I_{n-1})$ و بردار خروجی $O_v = (O_0,O_1,\ldots,O_{n-1})$ باشد، در این صورت نگاشتی یک به یک بین بردار ورودی و بردار خروجی وجود دارد. این رابطه به صورت $I_v \leftrightarrow O_v$ قابل بیان است.

گیت تحمل‌پذیر اشکال از نوع نگهدارنده‌ی توازن، گیتی است که در آن توازن ورودی‌ها با توازن خروجی‌ها برابر است. به عبارت دیگر رابطه‌ی زیر در یک گیت نگهدارنده‌ی توازن وجود دارد:

$$I_0 \oplus I_1 \oplus \ldots \oplus I_{n-1} = O_0 \oplus O_1 \oplus \ldots \oplus O_{n-1} \quad (1)$$

ویژگی نگهدارندگی توازن باعث می‌شود که خطاهای رخداده در خروجی اصلی مدار حداقل قابل تشخیص شوند. پژوهش‌ها نشان داده است اگر یک مدار برگشت‌پذیر، تنها با گیت‌های تحمل‌پذیر اشکال طراحی شود، کل مدار تحمل‌پذیر اشکال خواهد بود [۵].

در طراحی مدار برگشت‌پذیر، ممکن است خروجی‌هایی تولید شوند که مقدارشان در ادامه کار مورد استفاده قرار نمی‌گیرد. به این خروجی‌ها، خروجی بلااستفاده می‌گویند. این خروجی‌ها تنها برای حفظ خاصیت برگشت‌پذیری مدار و یا ویژگی نگهداری توازن استفاده می‌شوند.

هزینه‌ی کوانتومی در یک مدار برگشت‌پذیر برابر است با تعداد گیت‌های کوانتومی ۱×۱ و ۲×۲ که در ساخت مدار استفاده شده است. هزینه‌ی کوانتومی هر گیت ۱×۱ و ۲×۲ برابر واحد در نظر گرفته می‌شود.

تأخیر در یک مدار برگشت‌پذیر به صورت حداکثر تعداد گیت‌های موجود در مسیر بحرانی از ورودی‌ها به خروجی‌ها تعریف می‌شود.

پیچیدگی سخت‌افزاری یا پیچیدگی محاسباتی یک مدار برگشت‌پذیر وابسته است به تعداد عملیات XOR، AND و NOT ظاهر شونده در خروجی‌های آن (البته بهتر است عملیات مشترک و یکسان در خروجی‌ها تنها یک بار شمارش شوند). با فرض این که α تعداد عملیات XOR، β تعداد عملیات AND و γ تعداد عملیات NOT باشد، پیچیدگی محاسباتی T به وسیله‌ی رابطه‌ی (۲) محاسبه می‌شود:

$$T=N(\alpha).\alpha+N(\beta).\beta+N(\gamma).\gamma \quad (2)$$

که در آن $N(*)$ بیانگر تعداد عملیات از نوع * است.

گیت‌های پایه‌ی برگشت‌پذیر نگهدارنده‌ی توازن

در این بخش گیت‌های اصلی نگهدارنده توازن خصوصاً گیت‌هایی که در طرح‌های مرتبط با مقاله مورد استفاده قرار گرفته‌اند، معرفی می‌شوند. این گیت‌ها قابلیت تشخیص خطا را به مدارهای بهره‌برداری کننده از آن‌ها می‌دهند.

۱) F2G[۱۲] [۶]: یک گیت برگشت‌پذیر ۳×۳ با هزینه‌ی کوانتومی برابر با دو است که دیاگرام بلوکی آن در شکل ۱ نشان داده شده است. از این گیت غیر از عملیات XOR می‌توان به عنوان جایگزینی برای فن‌آوت استفاده کرد. پیچیدگی محاسباتی این گیت برابر با $\alpha 2$ است.

---

۷ Parity preserving

۸ Garbage output

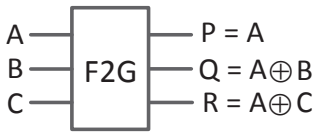
۹ Quantum cost

۱۰ Delay

۱۱ Hardware complexity

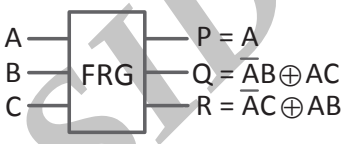
۱۲ Feynman Double Gate

Archive of SID
www.SID.ir



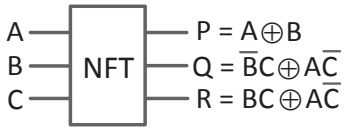
شکل ۱. دیاگرام بلوکی F2G [۶]

۲) FRG[۱۳][۷]: یک گیت ۳×۳ با هزینه‌ی کوانتومی برابر با پنج است که نمای کلی آن در شکل ۲ نشان داده شده است. اگر ورودی اول این گیت را به عنوان کنترل‌کننده در نظر بگیریم، در صورت فعال بودن این ورودی (یک بودن)، جای ورودی‌های دوم و سوم در خروجی این گیت تعویض خواهد شد. پیچیدگی محاسباتی این گیت با توجه به خروجی‌های آن، برابر با ۲α+۴β+۱γ است.



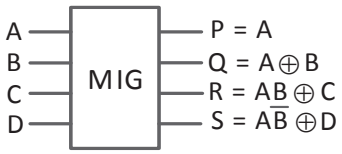
شکل ۲. دیاگرام بلوکی FRG [۷]

۳) NFT[۱۴][۸]: یک گیت تحمل‌پذیر اشکال ۳×۳ با هزینه‌ی کوانتومی پنج است که دیاگرام بلوکی آن در شکل ۳ نمایش داده شده است. پیچیدگی محاسباتی این گیت با توجه به خروجی‌های آن و شمارش تنها یک بار معادلات یکسان در خروجی، برابر با β۳+α۳+γ۲ است.



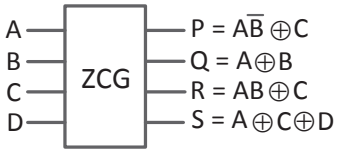
شکل ۳. دیاگرام بلوکی NFT [۸]

۴) MIG[۱۵][۹]: این گیت نگهدارنده توازن ۴×۴ که دارای هزینه‌ی کوانتومی برابر با هفت است، در صورت صفر قرار دادن ورودی‌های C و D آن به صورت یک نیم‌جمع‌کننده[۱۶]عمل می‌کند. علاوه بر این، با اتصال مناسب دو MIG می‌توان یک تمام‌جمع‌کننده[۱۷] تحمل‌پذیر اشکال بدست آورد. دیاگرام بلوکی MIG که دارای پیچیدگی محاسباتی برابر با β۲+α۳+γ۱ است، در شکل ۴ نمایش داده شده است.



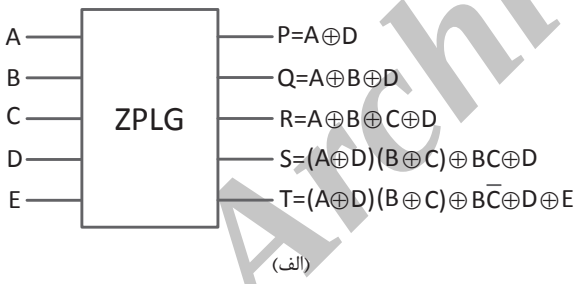
شکل ۴. دیاگرام بلوکی MIG [۹]

۵) ZCG [۱۰]: دیاگرام بلوکی این گیت ۴×۴ که دارای هزینه‌ی کوانتومی برابر با شش است، در شکل ۵ نشان داده شده است. همانند MIG با صفر قرار دادن ورودی‌های سوم و چهارم در این گیت، می‌توان از آن برای پیاده‌سازی نیم‌جمع‌کننده‌ی تحمل‌پذیر اشکال با ویژگی تشخیص خطا استفاده کرد. پیچیدگی محاسباتی این گیت برابر با ۵α+۲β+γ۱ است.

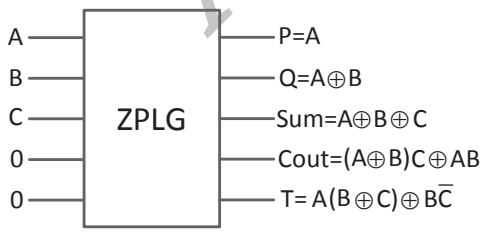


شکل ۵ .دیاگرام بلوکی ZCG [۱۰]

۶) ZPLG [۱۰]: دیاگرام بلوکی این گیت ۵×۵ که دارای هزینه‌ی کوانتومی برابر با هشت است، در شکل ۶ الف نشان داده شده است. با صفر قرار دادن ورودی‌های چهارم و پنجم، این گیت توانایی عملکرد به عنوان تمام‌جمع‌کننده‌ی تحمل‌پذیر اشکال را دارد که در شکل ۶ ب نمایش داده شده است. پیچیدگی محاسباتی این گیت برابر با β۳+α۹+γ۱ است.



(الف)



(ب)

شکل ۶. الف) دیاگرام بلوکی ZPLG [۱۰]، ب) عملکرد به صورت تمام‌جمع‌کننده

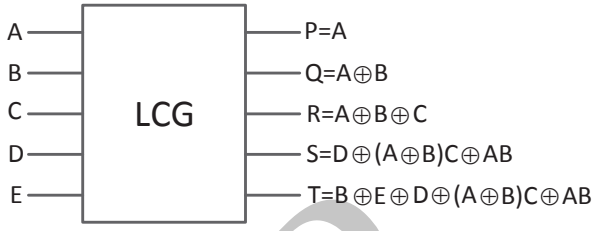
---

[۱۳] Fredkin Gate
[۱۴] New Fault Tolerant
[۱۵] Modified Islam Gate
[۱۶] Half adder
[۱۷] Full adder

Archive of SID

www.SID.ir

۷) LCG[۱۸][۱۱]: دیاگرام بلوکی این گیت ۵×۵ که دارای هزینه کوانتومی برابر با ۱۰ است، در شکل ۷ مشاهده می‌شود. عملکرد این گیت مشابه ZPLG است، با این تفاوت که دارای پیچیدگی محاسباتی کمتر و برابر با ۲β+۶α است.



شکل ۷. دیاگرام بلوکی LCG [۱۱]

تحقیقات مرتبط

با توجه به اهمیت بسیار بالای ضرب‌کننده در طراحی پردازنده‌ها، تاکنون طرح‌های متنوعی برای انواع ضرب‌کننده‌های برگشت‌پذیر ارائه شده است. عملیات ضرب با دو ساختار کلی سری و موازی قابل پیاده‌سازی است. در کاربردهایی که کم بودن هزینه پیاده‌سازی مهمتر از سرعت اجرای عملیات است، ضرب سری و در کاربردهایی که سرعت انجام کار مهمتر از هزینه پیاده‌سازی است، ضرب موازی مقبولیت پیشتری دارد. در این میان، الگوریتم بوث[۱۹] برای ساختار سری و روش ضرب آرایه‌ای برای ساختار موازی مشهورترند.

تاکنون ضرب‌کننده‌های موازی تحمل‌پذیر اشکال متنوعی از نوع برگشت‌پذیر مانند [۱۲،۱۳،۱۴،۱۵،۱۶] ارائه شده‌اند، اما تنها در [۱۷] طراحی ضرب‌کننده‌های سری تحمل‌پذیر اشکال از نوع برگشت‌پذیر مورد بررسی قرار گرفته است.

الگوریتم بوث که برای ضرب سری اعداد علامت‌دار به کار می‌رود، در پنج مرحله به صورت زیر انجام می‌شود [۱۸]. در این روش به ازای شروع یک رشته بیت حاوی یک‌ها در عملوند دوم، یک عملیات تفریق و به ازای انتهای این رشته، یک عملیات جمع مطابق با معادله (۳) انجام می‌شود:

$$2^J + 2^{J-1} + 2^{J-2} + \ldots + 2^K = 2^{J+1} - 2^K \qquad (3)$$

مرحله ۱: تولید یک کپی از بیت‌های عملوند اول

مرحله ۲: محاسبه مکمل دو عملوند اول

مرحله ۳: استفاده از مالتی‌پلکسر که در هر مرحله بر اساس مقادیر هر دو بیت کنار هم در عملوند دوم، به نحو مقتضی از عملوند اول در عملیات استفاده می‌کند (جدول ۱).

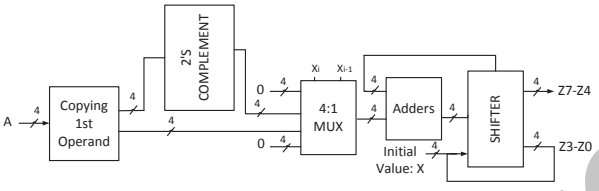
جدول ۱. توالی بیت‌های عملوند دوم و عملیات متناظر در روش بوث

x_i	x_{i-1}	y_i	عملیات
0	0	0	وسط رشته ۰ ها، بدون عملیات
0	1	1	پایان رشته ۱ ها، جمع با عملوند اول
1	0	-1	شروع رشته ۱ ها، تفریق عملوند اول
1	1	0	وسط رشته ۱ ها، بدون عملیات

مرحله ۴: استفاده از جمع‌کننده‌ها

مرحله ۵: انتقال‌دهنده‌ی موازی[۲۰]. از انتقال‌دهنده‌ی موازی برای انتقال‌به‌راست حسابی[۲۱] بیت‌های تولید شده استفاده می‌شود.

طرح این ضرب‌کننده در شکل ۸ نشان داده شده است.



شکل ۸. طرح ضرب‌کننده‌ی بوث چهار بیتی

در [۱۷]، طرحی با ساختار سری برای ضرب‌کننده‌ی علامت‌دار برگشت‌پذیر تحمل‌پذیر اشکال با ویژگی تشخیص خطا، بر اساس روش بوث برای ضرب اعداد چهار بیتی ارائه شده است. پنج مرحله‌ی روش اولیه بوث در این مرجع به صورت زیر پیاده‌سازی شده است:

(۱) برای کپی بیت‌های عملوندها از چهار F2G استفاده شده است. اولین کپی از عملوند اول به بخش تولید مکمل ۲ و دومین کپی به مالتی‌پلکسر ارسال می‌شود. این بخش با توجه به استفاده از چهار F2G دارای هزینه کوانتومی برابر با هشت است.

(۲) برای تولید مکمل ۲ عملوند اول، ابتدا بیت‌ها با عبور از گیت NOT معکوس می‌شوند، سپس نتیجه بدست آمده، با استفاده از نیم‌جمع‌کننده‌ها با عدد یک جمع می‌شود تا مکمل ۲ عملوند اول تولید شود. در این مرحله از چهار گیت NOT با هزینه‌ی کوانتومی کلی چهار و برای انجام جمع با عدد یک و انتشار بیت نقلی از چهار نیم‌جمع‌کننده از نوع MIG با هزینه‌ی کلی ۲۸ استفاده شده است

[۱۸] Low-Cost Gate

[۱۹] Booth's algorithm

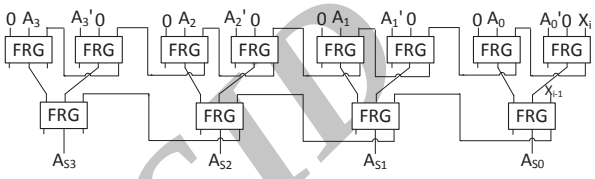
[۲۰] Shifter

[۲۱] Arithmetic right shift

Archive of SID
www.SID.ir

که در نتیجه هزینه‌ی کوانتومی کل این بخش برابر با ۳۲ شده است.

(۳) برای طراحی هر مالتی‌پلکسر چهار به یک برگشت‌پذیر، از سه مالتی‌پلکسر دو به یک استفاده شده است. هر مالتی‌پلکسر دو به یک به‌وسیله‌ی یک FRG قابل پیاده‌سازی است. بنابراین برای چهار بیت، از ۱۲ FRG با هزینه کوانتومی ۶۰ استفاده شده است. مدار این بخش که شامل چهار مالتی‌پلکسر چهار به یک است، در شکل ۹ نمایش داده شده است.



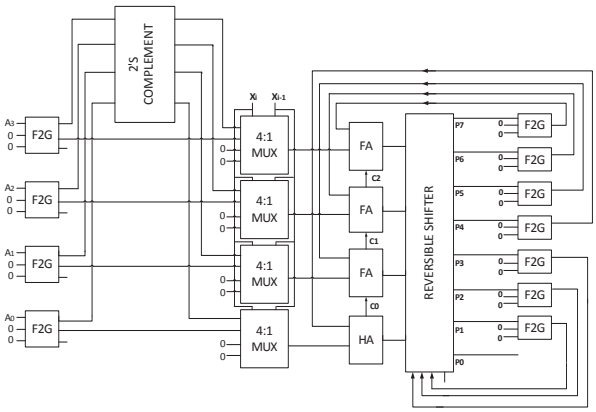
شکل ۹. مدار بخش مالتی‌پلکسرهای چهار به یک مطابق با [۱۷]

(۴) برای پیاده‌سازی جمع‌کننده‌ها از یک MIG به عنوان نیم‌جمع‌کننده و دو MIG به عنوان تمام‌جمع‌کننده استفاده می‌شود. بنابراین، با توجه به چهاربیتی بودن ضرب‌کننده و نیاز به هفت MIG، هزینه‌ی کوانتومی این بخش برابر با ۴۹ است.

(۵) انتقال‌دهنده‌ی موازی به‌وسیله‌ی هفت گیت FRG و یک گیت F2G پیاده‌سازی شده است. بنابراین، هزینه کوانتومی این بخش برابر با ۳۷ است. (هفت گیت از نوع F2G نیز برای بازگرداندن نتایج به قبل از این بخش در مقابل این بخش قرار داده شده‌اند (شکل ۱۰).

طرح ضرب‌کننده‌ی بوث چهار بیتی ارائه شده در [۱۷] شامل ۴۶ گیت با هزینه‌ی کوانتومی برابر با ۲۰۰ است. طرح کلی این ضرب‌کننده در شکل ۱۰ مشاهده می‌شود. در این شکل، عملوند اول (مضروب) درون A ($A_3A_2A_1A_0$) و عملوند دوم (مضروب‌فیه) درون X ذخیره شده است. X یک آرایه‌ی هشت بیتی است که مقدار اولیه‌ی چهار بیت کم‌ارزش‌تر آن ($X_3X_2X_1X_0$) برابر با چهار بیت عملوند دوم است.

به منظور بهبود طرح پایه ضرب‌کننده‌ی مبتنی بر الگوریتم بوث، در [۱۷] الگوریتمی مناسب‌تر با نام الگوریتم K[۲۲]پیشنهاد شده است. در روش بوث، با توجه به جدول ۱، سه حالت مختلف برای مقداری که به خروجی می‌رود، وجود دارد. به همین دلیل، نیاز به مالتی‌پلکسر چهار به یک وجود دارد. در الگوریتم K برای ساده‌تر شدن عملیات، با تغییر در طراحی مدار، جدول ۱ به جدولی ساده‌تر تبدیل شده است که در آن تنها دو حالت برای مقداری که به جمع‌کننده‌ها ارسال می‌شود وجود دارد. مقادیر اصلاح‌شده در جدول ۲ نشان داده شده است. در الگوریتم K مکمل دو عملوند اول محاسبه نمی‌شود تا به عنوان یک انتخاب به مالتی‌پلکسر ارسال شود؛ بلکه تنها از بین دو مقدار صفر و یا عملوند اول، عمل انتخاب صورت می‌گیرد. در این حالت دیگر احتیاجی به مالتی‌پلکسر چهار به یک نیست و با استفاده از مالتی‌پلکسر دو به یک با خط کنترل $x_i \oplus x_{i-1}$ عمل انتخاب صورت می‌گیرد.



شکل ۱۰. طرح ضرب‌کننده‌ی بوث چهار بیتی مطابق با توضیحات ارائه شده در [۱۷]

جدول ۲. توالی بیت‌های عملوند دوم و عملیات متناظر مطابق الگوریتم K [۱۷]

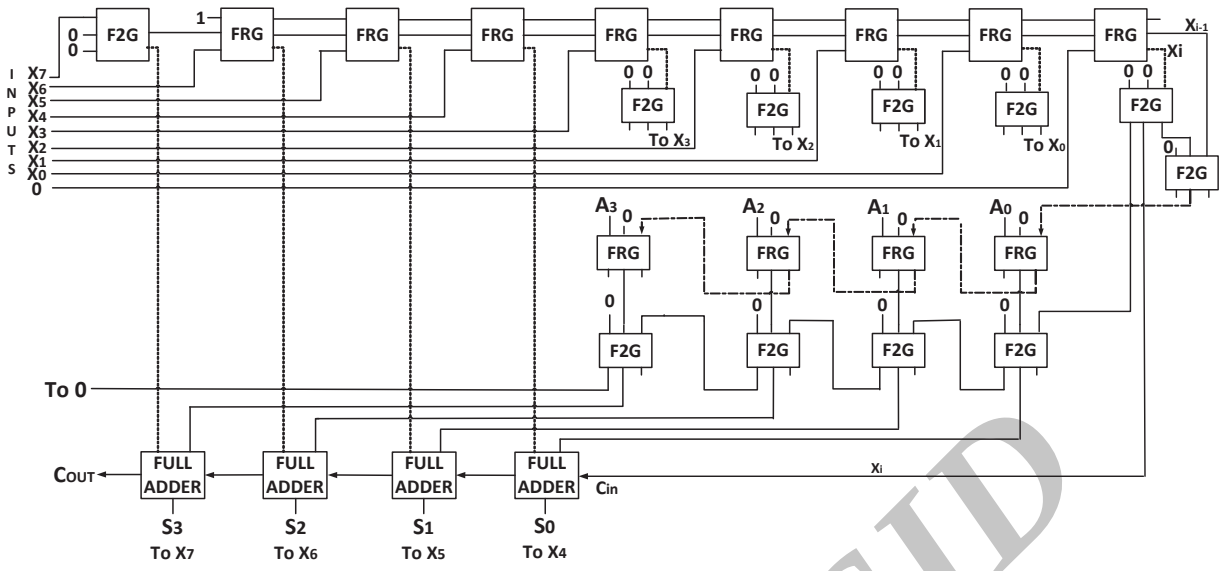
x_i	x_{i-1}	y_i	عملیات
0	0	-	وسط رشته ۰ ها، بدون عملیات
0	1	0	پایان رشته ۱ ها، جمع با عملوند اول
1	0	1	شروع رشته ۱ ها، جمع با عملوند اول
1	1	-	وسط رشته ۱ ها، بدون عملیات

برای پیاده‌سازی مدار ضرب‌کننده‌ی پیشنهادی در [۱۷]، از کم‌هزینه‌ترین تمام‌جمع‌کننده‌ی آن زمان یعنی SNFA [۱۹] با هزینه‌ی کوانتومی برابر با ۱۱ استفاده شده است. پیچیدگی محاسباتی این تمام‌جمع‌کننده برابر با $\beta 3+\alpha 9+\gamma 2$ است. همچنین، برای پیاده‌سازی عملیات XOR از F2G استفاده شده است.

ضرب‌کننده‌ی چهار بیتی ارائه شده در [۱۷] در شکل ۱۱ نشان داده شده است. در این شکل، عملوند اول درون A و عملوند دوم در شروع کار در چهار بیت کم‌ارزش X قرار دارد. این مدار حاوی ۳۹ گیت شامل ۱۲ گیت FRG، ۲۳ گیت F2G و چهار گیت NFT است. در نتیجه هزینه‌ی کوانتومی آن برابر با ۱۲۶ است.

[۲۲] Keshuv algorithm

Archive of SID
www.SID.ir



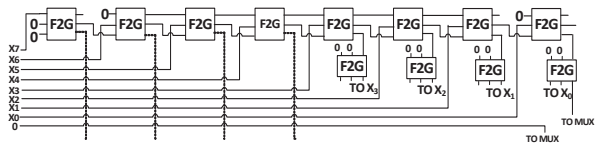
شکل ۱۱. ضرب‌کننده‌ی چهار بیتی مبتنی بر الگوریتم K ارائه شده در [۱۷]

ضرب‌کننده‌های پیشنهادی تشخیص دهنده خطا با ساختار سری

ضرب‌کننده‌های پیشنهادی برای ضرب اعداد علامت‌دار

منظور از ویژگی تشخیص خطا در ضرب‌کننده‌های پیشنهادی، داشتن ویژگی نگهداری توازن به علت استفاده از گیت‌های نگهدارنده توازن است. بنابراین، در طرح‌های پیشنهادی فقط از این نوع گیت‌ها استفاده می‌شود. اولین طرح پیشنهادی در این مقاله برای ضرب اعداد علامت‌دار، مبتنی بر روش ضرب بوث است. تفاوت این طرح با مدار ارائه شده در [۱۷] در بخش انتقال دهنده موازی است. در مدار پیشنهادی برای طراحی این بخش، به جای استفاده از گیت‌های از نوع FRG، از گیت‌های از نوع F2G و تغییر در نحوه ارسال ورودی‌ها به این گیت‌ها مطابق شکل ۱۲ استفاده شده است. بنابراین، با چینش جدید گیت‌ها و جایگزینی گیت FRG با گیت F2G می‌توان همان عملیات را با هزینه‌ی کوانتومی کمتری انجام داد. همچنین، برای پیاده‌سازی بخش جمع‌کننده‌ها، از ZCG به عنوان نیم‌جمع‌کننده و از ZPLG به عنوان تمام‌جمع‌کننده استفاده شده است. بدین ترتیب، هزینه‌ی کوانتومی ضرب‌کننده‌ی جدید مبتنی بر بوث نسبت به ضرب‌کننده‌ی اولیه کاهش می‌یابد. هزینه کوانتومی مدار پیشنهادی اول که در شکل ۱۳ نشان داده شده، برابر با ۱۵۰ است.

دومین طرح پیشنهادی در این مقاله برای ضرب اعداد علامت‌دار، مبتنی بر پیاده‌سازی بهبودیافته الگوریتم K معرفی شده در [۱۷]



شکل ۱۲. مدار پیشنهادی بخش انتقال‌دهنده‌ی موازی در ضرب‌کننده‌ی چهار بیتی مبتنی بر روش بوث

است. در این ضرب کننده، برای پیاده‌سازی مراحل سری الگوریتم ضرب بوث به کمک الگوریتم K به صورت زیر عمل نموده‌ایم:

۱. برای پیاده‌سازی بخش انتقال‌دهنده‌ی موازی، از ۱۲ گیت از نوع F2G استفاده شده است. بنابراین، هزینه‌ی کوانتومی این بخش برابر با ۲۴ است.

۲. برای پیاده‌سازی مالتی‌پلکسرهای دو به یک از چهار گیت از نوع FRG استفاده شده است. بنابراین، هزینه‌ی کوانتومی این بخش برابر با ۲۰ است.

۳. برای تولید معکوس شده‌ی عملوند اول از دو گیت از نوع F2G با هزینه‌ی کوانتومی کلی چهار استفاده شده است.

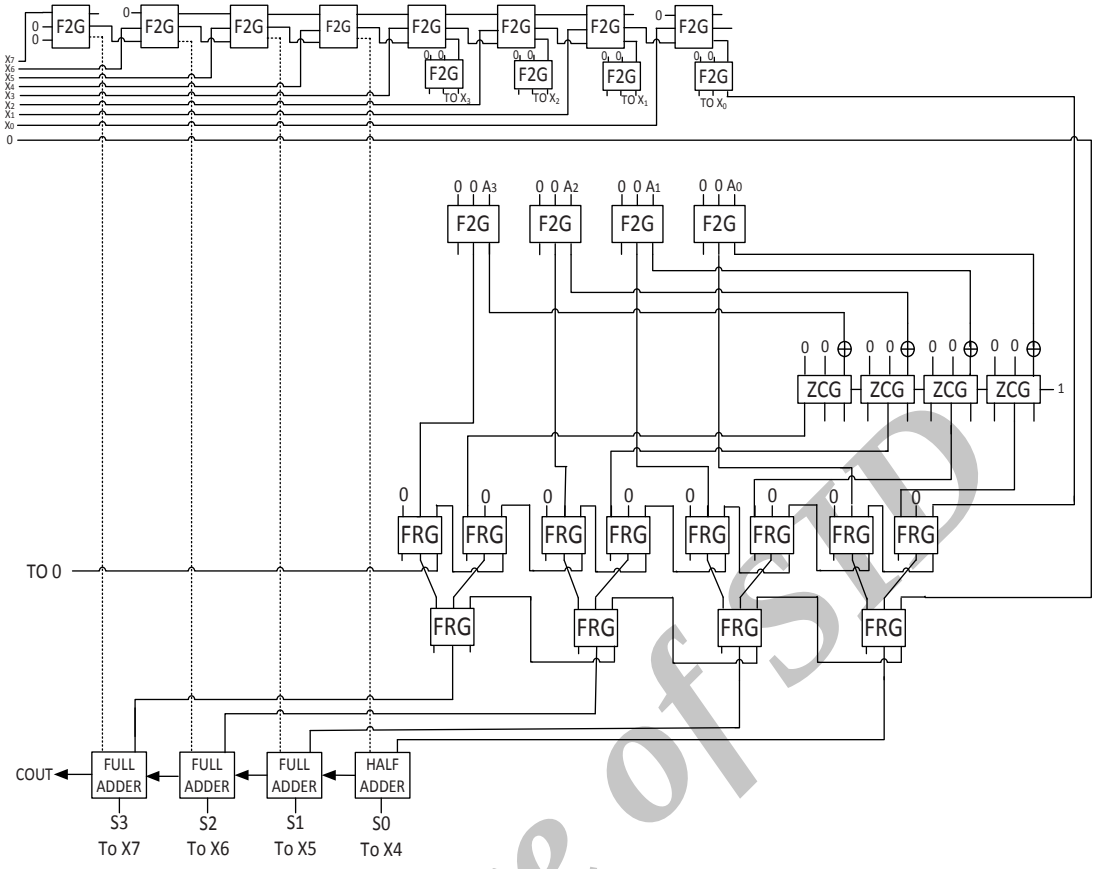
۴. عملیات جمع به‌وسیله چهار گیت از نوع ZPLG با هزینه‌ی کوانتومی کلی ۳۲ پیاده‌سازی شده است.

۵. کنترل‌کننده مالتی‌پلکسرهای دو به یک به‌وسیله یک F2G با هزینه‌ی کوانتومی دو پیاده‌سازی شده است.

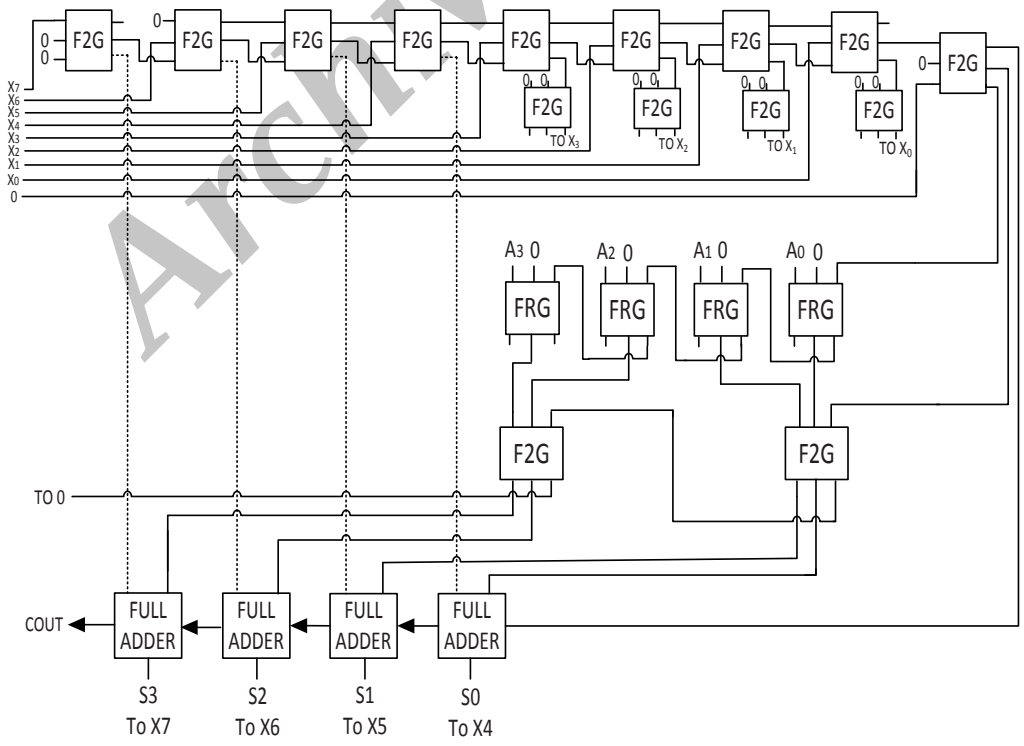
مدار پیشنهادی دوم که در شکل ۱۴ نشان داده شده است، حاوی ۲۳ گیت با هزینه‌ی کوانتومی برابر با ۸۲ است. در این ضرب‌کننده ۱۸ خروجی بلااستفاده نیز تولید می‌شود.

Archive of SID

www.SID.ir



شکل ۱۳. مدار پیشنهادی برای ضرب‌کننده‌ی چهار بیتی علامت‌دار تشخیص دهنده خطا مبتنی بر روش بوث



شکل ۱۴. مدار پیشنهادی برای ضرب‌کننده‌ی چهار بیتی علامت‌دار تشخیص دهنده خطا مبتنی بر الگوریتم K

Archive of SID

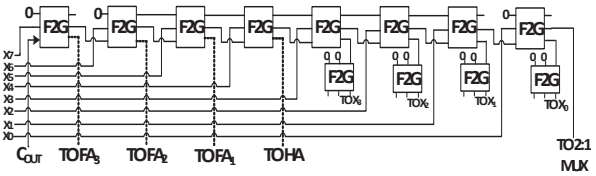
www.SID.ir

ضرب‌کننده‌ی پیشنهادی برای ضرب اعداد بدون علامت

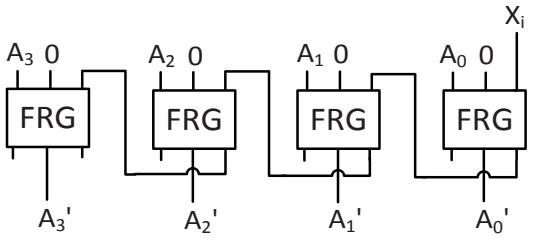
مدار پیشنهادی سوم در این مقاله، برای ضرب سری اعداد بدون علامت و مبتنی بر روش جمع و شیفت[1]است. این روش به خاطر بدون علامت بودن عملوندها نسبت به الگوریتم K پیچیدگی کمتری دارد. در این روش به ازای کم‌ارزش‌ترین بیت، تنها دو حالت وجود دارد: اگر برابر با صفر باشد، صفر به جمع‌کننده‌ها ارسال می‌شود؛ اما اگر برابر با یک باشد، خود عملوند اول به جمع‌کننده‌ها ارسال می‌شود.

در طرح ضرب‌کننده‌ی پیشنهادی برای پیاده‌سازی بخش انتقال‌دهنده‌ی موازی از ۱۲ گیت F2G استفاده شده است که با توجه به این که هزینه‌ی کوانتومی هر F2G برابر با دو است، هزینه‌ی کوانتومی این بخش برابر با ۲۴ می‌گردد. در این بخش ۱۱ خروجی بلااستفاده نیز تولید می‌شود. مدار این بخش در شکل ۱۵ نمایش داده شده است.

در این ضرب‌کننده، تنها به چهار مالتی‌پلکسر دو به یک نیاز است. مدار این بخش در شکل ۱۶ نشان داده شده است. در این مدار از چهار گیت FRG استفاده شده است. بنابراین، هزینه‌ی کوانتومی این بخش از مدار ضرب‌کننده برابر با ۲۰ می‌شود. همچنین، پنج خروجی بلااستفاده نیز تولید می‌گردد.

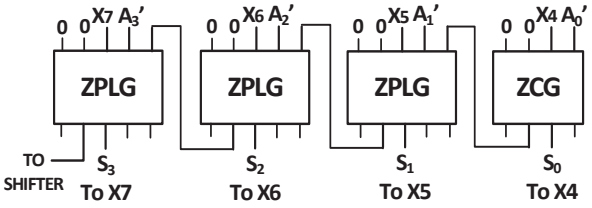


شکل ۱۵. مدار انتقال‌دهنده‌ی موازی در ضرب‌کننده‌ی پیشنهادی مبتنی بر روش جمع و شیفت



شکل ۱۶. مدار مالتی‌پلکسرهای دو به یک در ضرب‌کننده‌ی چهار بیتی پیشنهادی مبتنی بر روش جمع و شیفت

در بخش جمع‌کننده‌ها به یک نیم‌جمع‌کننده و سه تمام‌جمع‌کننده نیاز است که از ZCG به عنوان نیم‌جمع‌کننده با هزینه‌ی کوانتومی شش و ZPLG به عنوان تمام‌جمع‌کننده با هزینه‌ی کوانتومی هشت استفاده می‌شود. بنابراین، این بخش از مدار که حاوی چهار گیت است، دارای هزینه‌ی کوانتومی برابر با ۳۰ خواهد شد. علاوه بر این، ۱۱ خروجی بلااستفاده نیز در این بخش از مدار تولید می‌شود. در این بخش، آخرین رقم نقلی تولیدی به عنوان ورودی به بخش انتقال‌دهنده‌ی موازی ارسال می‌شود. مدار این بخش در شکل ۱۷ نمایش داده شده است.



شکل ۱۷. بخش جمع‌کننده‌ها در ضرب‌کننده‌ی چهار بیتی پیشنهادی مبتنی بر روش جمع و شیفت

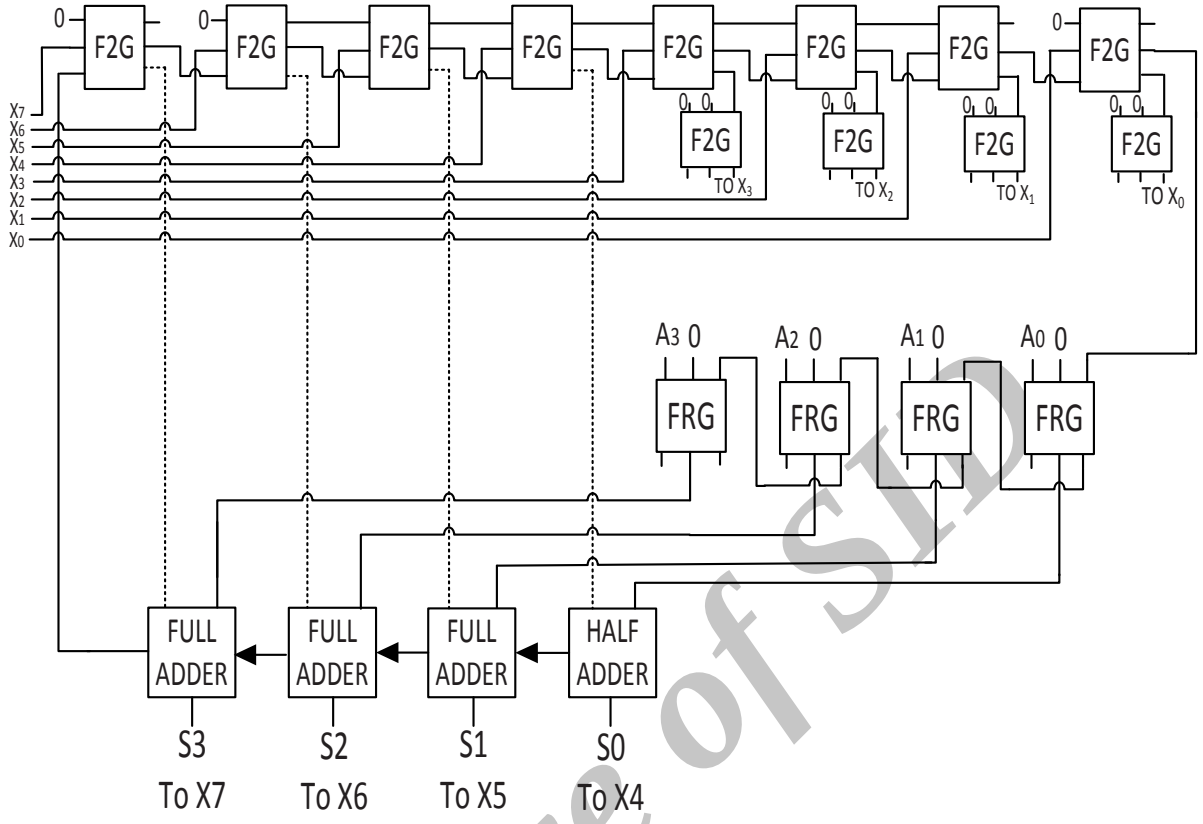
طرح کامل ضرب‌کننده‌ی سری پیشنهادی مبتنی بر روش جمع و شیفت در شکل ۱۸ نشان داده شده است. این طرح شامل ۲۰ گیت و هزینه‌ی کوانتومی آن برابر با ۷۴ است. در این مدار، ۲۷ خروجی بلااستفاده نیز تولید می‌شود.

ارزیابی و مقایسه

تنها ضرب‌کننده‌های سری برگشت‌پذیر تحمل‌پذیر اشکال طراحی شده تاکنون، مدارهای ارائه شده در [۱۷] هستند. بنابراین، مقایسه ضرب‌کننده‌های پیشنهادی در این مقاله را با آن مدارها انجام می‌دهیم. جدول ۳، مقایسه‌ی ضرب‌کننده‌های پیشنهادی چهار بیتی در این مقاله را با مدارهای قبلی از نظر پارامترهای اصلی طراحی مدارهای برگشت‌پذیر شامل تعداد گیت‌ها، هزینه کوانتومی، تأخیر، تعداد خروجی بلااستفاده و پیچیدگی محاسباتی نشان می‌دهد. مطابق با این جدول، ضرب‌کننده‌های پیشنهادی در این مقاله در تمام پارامترهای اصلی طراحی مدارهای برگشت‌پذیر از نمونه‌های مشابه قبلی خود بهتر بوده و بهبودهای مناسبی را نشان می‌دهند. همان طور که قبلاً بیان گردید، مدارهای ارائه شده در [۱۷] که در شکل‌های ۱۰ و ۱۱ نشان داده شده‌اند، ضرب‌کننده‌های سری تشخیص‌دهنده خطای مبتنی بر الگوریتم‌های بوث و K هستند. اما در این مقاله ضرب‌کننده‌های مبتنی بر هر دو الگوریتم را با استفاده از چینش جدید گیت‌های برگشت‌پذیر موجود و همچنین، استفاده از گیت‌هایی مناسب‌تر، بهبود داده‌ایم. ضرب‌کننده‌ی پیشنهادی اول که مبتنی بر الگوریتم بوث است (شکل ۱۳)، با توجه به این که حاوی چهار گیت NOT، ۱۶ گیت از نوع F2G، ۱۲ گیت از نوع FRG، پنج گیت از

[1] Add & Shift

Archive of SID
www.SID.ir



شکل ۱۸. مدار پیشنهادی برای ضرب‌کننده‌ی چهار بیتی بدون علامت تشخیص دهنده خطا مبتنی بر روش جمع و شیفت

جدول ۳. مقایسه انواع ضرب‌کننده‌های سری برگشت‌پذیر با قابلیت تشخیص خطا

ضرب‌کننده‌ی چهار بیتی	نوع الگوریتم	علامت‌دار	تأخیر	تعداد گیت	پیچیدگی محاسباتی	تعداد خروجی بلااستفاده	هزینه کوانتومی
مدار ۱ [۱۷] (شکل ۱۰)	Booth	بله	>۷۲	۴۶	۹۹α+۹۸β+γ۳۰	۶۰	۲۰۰
مدار پیشنهادی ۱ (شکل ۱۳)	Booth	بله	۴۸	۴۰	۱۱۲α+۶۷β+۲۰γ	۴۷	۱۵۰
مدار ۲ [۱۷] (شکل ۱۱)	K-alg.	بله	۷۲	۳۹	۸۲α+۶۰β+γ۲۰	۳۴	۱۲۶
مدار پیشنهادی ۲ (شکل ۱۴)	K-alg.	بله	۳۲	۲۳	۷۴α+۲۸β+γ۸	۲۸	۸۲
مدار پیشنهادی ۳ (شکل ۱۸)	Add & Shift	خیر	۲۴	۲۰	۶۴α+۲۷β+γ۸	۲۹	۷۴

نوع ZCG و سه گیت از نوع ZPLG است و این گیت‌ها به ترتیب دارای هزینه کوانتومی برابر با یک، دو، پنج، شش و هشت هستند، دارای هزینه‌ی کوانتومی برابر با ۱۵۰=۸×۶+۳×۵+۵×۲+۱۲×۱+۱۶×۴ می‌شود که بهبود مناسبی را در مقایسه با ضرب‌کننده‌ی مبتنی بر روش بوث ارائه شده در [۱۷] نشان می‌دهد. از طرف دیگر، ضرب‌کننده‌ی پیشنهادی مبتنی بر الگوریتم K (شکل ۱۴) با توجه به چینش جدید گیت‌های برگشت‌پذیر موجود و همچنین، استفاده از تمام‌جمع‌کننده‌های کم‌هزینه‌تر، به بهبودهای قابل توجهی نسبت به بهترین مدار ارائه شده تاکنون (شکل ۱۱) رسیده است. با توجه به این که ضرب‌کننده پیشنهادی دوم حاوی ۱۵ گیت از نوع F2G، چهار گیت از نوع FRG و چهار گیت از نوع ZPLG است، در نتیجه دارای هزینه‌ی کوانتومی برابر با ۸۲=۸×۴+۵×۴+۲×۱۵ می‌شود که بهبود قابل ملاحظه‌ای را در مقایسه با ضرب‌کننده‌ی مشابه ارائه شده در [۱۷] (شکل ۱۱) نشان می‌دهد.

Archive of SID
www.SID.ir

مدار پیشنهادی سوم در این مقاله (شکل ۱۸) ضرب‌کننده سری مبتنی بر روش پایه جمع و شیفت است و بنابراین برای ضرب اعداد بدون علامت به‌کار می‌رود. اما با توجه به این که بهبودهای انجام شده در مدار پیشنهادی دوم در این ضرب‌کننده نیز مورد استفاده قرار گرفته و همچنین، با توجه به این که ضرب بدون علامت ساده‌تر از ضرب علامت‌دار است، پارامترهای مداری مناسب‌تری نسبت به همه مدارهای قبلی دارد که در جدول ۳ نیز مشاهده می‌شود. به عنوان نمونه، هزینه کوانتومی این ضرب‌کننده به خاطر استفاده از ۱۲ گیت از نوع F2G، چهار گیت از نوع FRG، یک گیت از نوع ZCG و سه گیت از نوع ZPLG برابر با ۷۴ است که از همه ضرب‌کننده‌ها کمتر است.

برای محاسبه تأخیر کل هر مدار با توجه به این که همه ضرب‌کننده‌ها چهار بیتی هستند، تأخیر اجرای هر مرحله از عملیات ضرب، در عدد چهار که همان اندازه ضرب‌کننده و برابر با تعداد مراحل ضرب سری است، ضرب شده است. تأخیر اجرای هر مرحله از عملیات ضرب نیز با توجه به حداکثر تعداد گیت‌های موجود در مسیر ورودی‌ها به خروجی‌ها بدست می‌آید. در این محاسبه، مشابه کارهای گذشته برای سادگی، تأخیر هر گیت برابر واحد در نظر گرفته می‌شود. مطابق با جدول ۳، کاهش تأخیر یا افزایش سرعت مدارهای پیشنهادی در مقایسه با مدارهای قبلی قابل مشاهده است.

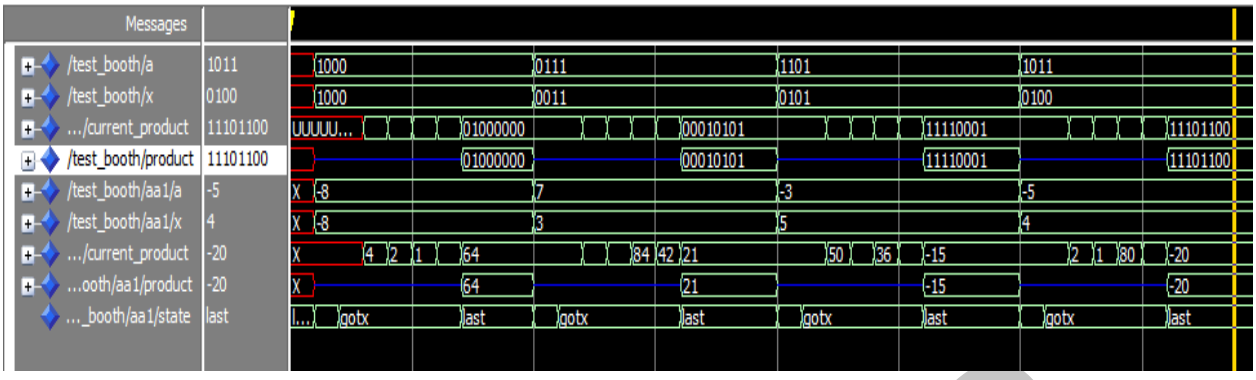
تعداد خروجی بلااستفاده در مدارهای پیشنهادی با شمارش خروجی‌های بلااستفاده همه گیت‌های مورد استفاده در مدار بدست می‌آید. در واقع، خروجی‌های میانی یا نهایی که مقدارشان در ادامه مدار مورد استفاده قرار نمی‌گیرد، به عنوان خروجی‌های بلااستفاده در نظر گرفته می‌شوند. بدین ترتیب مطابق جدول ۳، بهبودهای مناسبی نسبت به مدارهای مشابه قبلی بدست آمده است.

همان گونه که از جدول ۳ برمی‌آید، پیچیدگی محاسباتی ضرب‌کننده‌های پیشنهادی نیز بهبود مناسبی یافته‌اند. در این جدول، پیچیدگی محاسباتی هر مدار ضرب‌کننده با جمع کردن پیچیدگی محاسباتی همه گیت‌های آن مدار بدست آمده است. در واقع، با توجه به این که پیچیدگی محاسباتی گیت‌های مورد استفاده در بخش‌های اولیه مقاله ذکر شده است، پیچیدگی محاسباتی مدارهای مورد نظر در جدول ۳ به راحتی محاسبه می‌شود. اما نکته‌ای که در مورد نحوه‌ی یافتن پیچیدگی محاسباتی این مدارها بایستی بدان توجه نمود، این است که همانند روش ارائه شده در [۲۰]، در محاسبه‌ی پیچیدگی محاسباتی مربوط به هر کدام از گیت‌های مورد استفاده، معادلات یا عملیات مشترک در خروجی‌های هر گیت تنها یک بار شمارش می‌شوند. این نحوه شمارش با طرح کوانتومی داخلی گیت‌ها سازگاری دارد، زیرا اجزای کوانتومی مورد استفاده در گیت‌ها در صورت نیاز، به طور مشترک در تعیین مقدار خروجی گیت‌ها مورد استفاده قرار می‌گیرند.

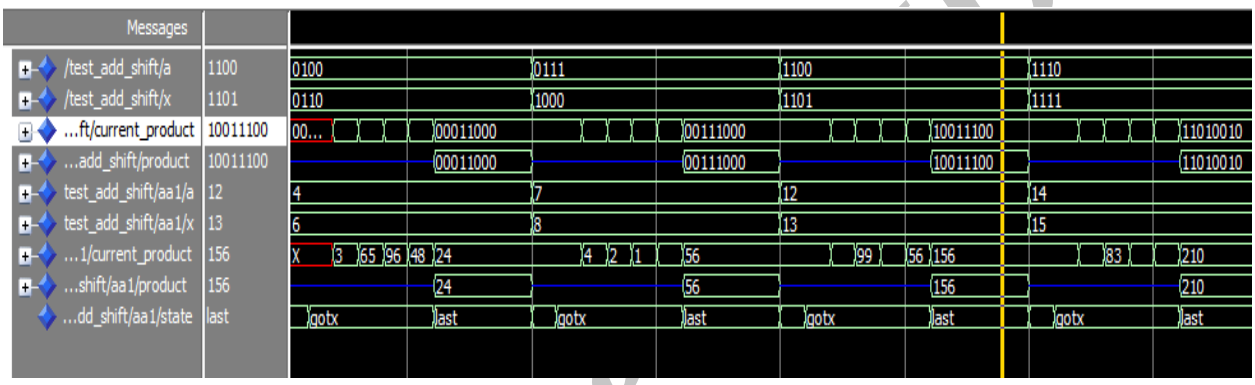
لازم به ذکر است که تأیید صحت عملکرد ضرب‌کننده‌های برگشت‌پذیر پیشنهادی به همراه دو ضرب‌کننده معرفی شده در [۱۷] از نظر عملیات منطقی مورد نظر، به کمک زبان توصیف سخت‌افزار VHDL و اجرای شبیه‌سازی‌های لازم در نرم‌افزار مدلسیم انجام شده است. با توجه به این که سطح طراحی این ضرب‌کننده‌ها یک درجه بالاتر از سطح طراحی گیت‌های برگشت‌پذیر است و همچنین، با توجه به مشخص بودن عملکرد گیت‌های مورد استفاده، تأیید صحت عملکرد نهایی مدارهای ضرب‌کننده مورد بررسی به سادگی به کمک زبان‌های توصیف سخت‌افزار و نرم‌افزارهای شبیه‌سازی مرتبط امکان‌پذیر است. برای نمونه، شکل موج خروجی مدارهای پیشنهادی ۱ و ۳ که مدارهای مبتنی بر الگوریتم بوث و روش جمع و شیفت هستند، به ترتیب در شکل‌های ۱۹ الف و ۱۹ ب نشان داده شده است. در این دو شکل، عملوندهای ضرب (A و X)، نتیجه میانی و نتیجه نهایی ضرب در دو فرمت دودویی و دهدهی نمایش داده شده‌اند. در شکل ۱۹ الف با توجه به این که ضرب از نوع علامت‌دار است، ضرب اعداد چهاربیتی علامت‌دار شبیه‌سازی شده است. برای نمونه، خط نمایش‌گر عمودی مکانی را نشان می‌دهد که دو عدد ۵- و ۴+ در فرمت نمایش مکمل ۲ در یکدیگر ضرب شده و نتیجه صحیح ۲۰- با تأخیر بدست آمده است. همچنین، در شکل ۱۹ ب با توجه به این که ضرب مبتنی بر روش جمع و شیفت از نوع بدون علامت است، ضرب عملوندهای چهاربیتی بدون علامت شبیه‌سازی شده است. در این شکل، خط نمایش‌گر عمودی مکانی را نشان می‌دهد که دو عدد ۱۲ و ۱۳ در فرمت نمایش دودویی ساده در یکدیگر ضرب شده و نتیجه صحیح ۱۵۶ در هشت بیت حاصل شده است.

Archive of SID

www.SID.ir



(الف)



(ب)

شکل ۱۹. شکل موج‌های نمونه شبیه‌سازی ضرب‌کننده‌های پیشنهادی (الف) مبتنی بر الگوریتم بوث، (ب) مبتنی بر روش جمع و شیفت

نتیجه‌گیری

در این مقاله، با توجه به مدارهای ضرب‌کننده‌ی برگشت‌پذیر سری موجود، سه ضرب‌کننده‌ی جدید با ویژگی نگهداری توازن که منجر به قابلیت تشخیص خطا می‌شود، شامل دو طرح برای ضرب اعداد علامت‌دار و یک طرح برای ضرب اعداد بدون علامت پیشنهاد شده است. برای طراحی ضرب‌کننده‌های علامت‌دار از الگوریتم بوث و الگوریتم K که بهبودیافته الگوریتم پایه بوث است و برای طراحی ضرب‌کننده‌ی بدون علامت از روش پایه جمع و شیفت استفاده گردید. در مدارهای پیشنهادی برای کاهش هزینه کوانتومی و بهبود سایر پارامترهای طراحی، هم از چینش جدید گیت‌های موجود استفاده گردید و هم از گیت‌های جدیدتر و کم‌هزینه‌تر خصوصاً برای واحدهای جمع‌کننده درون ضرب‌کننده‌ها بهره‌برداری شد. مقایسه ضرب‌کننده‌های پیشنهادی این مقاله با معدود مدارهای مشابه موجود نشان می‌دهد که در تمام پارامترهای اصلی طراحی مدارهای برگشت‌پذیر، بهبودهای مناسبی حاصل شده است. بدین ترتیب، مدار پیشنهادی دوم برای ضرب سری اعداد علامت‌دار و تنها مدار پیشنهادی برای ضرب سری اعداد بدون علامت بهترین طرح‌های موجود از این نوع برای استفاده در کامپیوترهای کوانتومی یا مدارهای برگشت‌پذیر هستند.

منابع

[1] R. Landauer, “Irreversibility and heat generation in the computing process,” *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183–191, 1961.

[2] C. Bennet, “Logical reversibility of computation,” *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, 1973.

[3] R. Feynman, “*Quantum* Mechanical Computers, ” *Optics News*, vol. 11, pp. 11–20, 1985.

[4] T. Toffoli, “Reversible Computing” *Tech. memo MIT/LCS/TM-151, MIT Lab for Computer Science*, 1980.

Archive of SID
www.SID.ir

[5] A. Peres, "Reversible logic and quantum computers," *Physical Review*, vol. 32, pp. 3266–3276, 1985.

[6] B. Parhami, "Fault-tolerant reversible circuits," *40th Asilomar Conf. on Signals, Systems and Computers*, (*ACSSC'06),* pp. 1726–1729, 2006.

[7] E. Fredkin and T. Toffoli, "Conservative logic," *Intl. Journal of Theoretical Physics*, vol. 21, pp. 219–253, 1982.

[8] M. Hagparast and K. Navi, "A Novel Fault Tolerant Reversible Gate for Nanotechnology Based System," *American Journal of Applied Sciences,* vol. 5, no. 5, pp. 519–523, 2008.

[9] M. S. Islam, M. M. Rahman, Z. Begum, and M. Z. Hafiz, "Fault Tolerant Reversible Logic Synthesis: Carry Look-Ahead and Carry-Skip Adders," *Intl. Conf. Advances in Computational Tools for Engineering Applications (ACTEA'09)*, pp. 396–401, 2009.

[10] R.-G. Zhou, Y.-C. Li, and M.-Q. Zhang, "Novel design for fault tolerant reversible binary coded decimal adders," *Intl. Journal of Electronics*, vol. 101, no. 10, pp.1336–1356, 2014.

[11] M. Valinataj, M. Mirshekar, and H. Jazayeri, "Novel low-cost and fault-tolerant reversible logic adders," *Computers and Electrical Engineering, Elsevier*, vol. 53, pp. 56–72, Jul. 2016.

[12] X. Qi and F. Chen, "Design of fast fault tolerant reversible signed multiplier," *Intl. Journal of the Physical Sciences* Vol. 7, no. 17, pp. 2506–2514, 2012.

[13] H. P. Sinha and N. Syal, "Design of fault tolerant reversible multiplier," *Intl. Journal of Soft Computing and Engineering (IJSCE)* vol. 1, no. 6, pp.120–124, 2012.

[14] L. Jamal, M. Rahman and H. Babu, "An optimal design of a fault tolerant reversible multiplier," *IEEE 26th Intl. SOC Conference (SOCC)*, pp. 37–42, 2013.

[15] P. Bhardwaj and M. Singh, "An improved design of a fault tolerant reversible binary parallel multiplier," *Intl. Journal of Engineering and Technical Research (IJETR)*, vol. 2, no. 9, pp. 72-76, 2014.

[16] A. Sahu and A. K. Sahu, "High speed fault tolerant reversible vedic multiplier," *Intl. Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 2, no. 6, pp. 71-78, 2015.

[17] K. Bhardwaj, M. Deshpande, "K-Algorithm: An Improved Booth's Recoding for Optimal Fault-Tolerant Reversible Multiplier," *26th Intl. Conf. on VLSI Design*, pp. 362–367, 2013.

[18] I. Koren, Computer Arithmetic Algorithms, 2nd Ed., A. K. Peters, Natick, MA, 2002.

[19] S. K. Mitra and A. R. Chowdhury, "Minimum cost fault tolerant adder circuits in reversible logic synthesis," *25th IEEE Intl. Conf. on VLSI Design (VLSID)*, pp. 334–339, 2012.

[20] محبوبه میرشکار، مجتبی ولی‌نتاج و حمید جزایری، "طراحی جمع‌کننده‌های BCD تحمل‌پذیر اشکال در منطق برگشت‌پذیر"، فصلنامه صنایع الکترونیک، دوره ۶، شماره ۴، زمستان ۱۳۹۴.

Archive of SID

www.SID.ir