

طراحی ضرب‌کننده‌های ممیز- شناور با قابلیت کار در مدهای عادی و تحمل‌پذیر اشکال با استفاده از کاهش دقت محاسبات

مریم مهاجر و مجتبی ولی‌نجاج

محسوب می‌گردد.

بسیاری از برنامه‌های کاربردی ممیز- شناور مانند کاربردهای چندرسانه‌ای به دقت کاملی که مطابق استانداردهای IEEE تعریف شده است نیازی ندارند. این برنامه‌ها اگرچه تحمل‌پذیر اشکال^۲ نیستند اما می‌توانند محاسبات تقریبی (یا نادقیق)^۳ را تحمل کنند و در نتیجه یک مقدار تخمینی برای خروجی آنها کافی است [۳] و [۴]. بنابراین می‌توان این نوع عملیات ممیز- شناور را با دقت کاهش‌یافته^۴ اجرا نمود تا با کوچک‌تر شدن مدارهای لازم مساحت مورد نیاز و توان مصرفی کاهش یابد. اما هدف از این مقاله، استفاده‌ای متفاوت از این ویژگی و در واقع بهره‌برداری از واحدهای آزادشده به همراه واحدهای جدید لازم، برای انجام محاسبات افزونه به منظور دستیابی به قابلیت اطمینان به صورت تشخیص یا تصحیح خطا است. معماری ضرب‌کننده پیشنهادی در این مقاله مبتنی بر عملیات ممیز- شناور ۳۲ بیتی مطابق با قالب نمایش دقت ساده در استاندارد IEEE-۷۵۴ [۵] است. ویژگی بعدی معماری پیشنهادی قابلیت عمل در دو مد کاری عادی و تحمل‌پذیر اشکال است تا با توجه به شرایط کاری محیطی که مدار ضرب‌کننده در آن قرار دارد، قابلیت انتخاب نحوه عمل وجود داشته باشد.

با توجه به این که در طرح‌های پیشنهادی، مد تحمل‌پذیر اشکال بر اساس محاسبات با دقت کاهش‌یافته قابل استفاده است برخی خطاهای محاسباتی در خروجی تولید می‌شود که با توجه به نتایج ارائه‌شده در [۶] در مورد دقت لازم و محدوده خطاهای محاسباتی برای کاربردهای گوناگون، قابل قبول خواهند بود. بنابراین در این مقاله با پذیرفتن درصدی ناچیز از خطای محاسباتی در خروجی به خاطر محاسبات تقریبی و تغییر ساختار ضرب‌کننده اصلی درونی، دو معماری ضرب‌کننده ممیز- شناور یکی با ویژگی تشخیص خطا و دیگری با ویژگی تصحیح خطا برای مقابله با انواع اشکال‌های دائمی و گذرا پیشنهاد می‌شود که با توجه به محدودبودن خطای محاسباتی، خروجی آنها برای بسیاری از برنامه‌های کاربردی چندرسانه‌ای رضایت‌بخش است، چون بسیاری از اعوجاج‌ها در خروجی به دلیل محدودیت حواس بینایی و شنوایی انسان به طور طبیعی پوشش داده می‌شوند.

بخش‌های بعدی مقاله به صورت زیر سازماندهی شده‌اند: در بخش دوم تحقیقات پیشین و مرتبط و در بخش سوم یک پیش‌زمینه از استاندارد IEEE برای اعداد ممیز- شناور و ضرب‌کننده ممیز- شناور با دقت ساده ارائه می‌شود. در ادامه در بخش چهارم معماری‌های پیشنهادی برای ضرب‌کننده ممیز- شناور در دو مد کاری عادی و تحمل‌پذیر اشکال، یکی با قابلیت تشخیص خطا و دیگری با قابلیت تصحیح خطا توضیح داده می‌شوند. سپس در بخش پنجم، طرح‌های پیشنهادی از نظر تأخیر، توان

چکیده: عملیات ضرب یکی از مهم‌ترین محاسبات مورد استفاده در انواع پردازش‌های سیگنال خصوصاً صوت و تصویر محسوب می‌شود. با این حال، ضرب‌کننده‌ها به عنوان مدارهای دیجیتال به خاطر وجود عوامل محیطی گوناگون مانند انواع نویزها مستعد تولید خروجی‌های نادرست هستند. در این مقاله، روشی جدید برای طراحی ضرب‌کننده ممیز- شناور ۳۲ بیتی ارائه می‌شود که می‌تواند با توجه به شرایط محیطی که در آن استفاده می‌شود، در دو مد کاری عادی یا تحمل‌پذیر اشکال عمل کند. در مد تحمل‌پذیر اشکال، با کاهش دقت محاسبات و قبول مقدار ناچیزی خطای محاسباتی در خروجی، بخشی از مدار اولیه آزاد شده و برای فراهم کردن محاسبات افزونه به منظور تشخیص یا تصحیح خطاهای ناشی از اشکال‌ها استفاده می‌شود. بدین روش، دو معماری ضرب‌کننده با قابلیت تشخیص یا تصحیح خطا پیشنهاد می‌شوند که در مد کاری تحمل‌پذیر اشکال، دارای قابلیت اطمینان مناسبی در برابر انواع اشکال‌های دائمی و گذرا هستند. نتایج پیاده‌سازی نشان می‌دهد که در مد تحمل‌پذیر اشکال به جای ۲۳ بیت ماتیس اولیه، حفظ ۱۳ بیت برای دست‌یافتن به ضرب‌کننده با قابلیت تشخیص خطا و حفظ ۱۱ بیت برای دست‌یافتن به ضرب‌کننده با قابلیت تصحیح خطا، با سربار مساحت و توان قابل قبول که از ۱۲٪ تا ۲۶٪ خواهد بود و همچنین حفظ دقت مورد نیاز برای اکثر کاربردها، مناسب است.

کلیدواژه: دقت کاهش‌یافته، تحمل‌پذیری اشکال، تشخیص خطا، تصحیح خطا، ضرب‌کننده ممیز- شناور.

۱- مقدمه

الگوریتم‌های پردازش سیگنال، کاربردهای چندرسانه‌ای و بسیاری از سیستم‌های نهفته به انجام محاسبات ریاضی خصوصاً عملیات ضرب وابسته هستند [۱]. در این راستا انواع پردازش‌های سیگنال، ویدئو، صوت و تصویر به محاسبات حاوی ضرب ممیز- شناور وابسته‌اند. همچنین برنامه‌های کاربردی علمی که نیازمند شبیه‌سازی انواع پدیده‌های فیزیکی هستند از محاسبات ممیز- شناور بهره می‌برند. علاوه بر این، کاهش اندازه ترانزیستورها منجر به بروز مشکلات مرتبط با قابلیت اطمینان^۱ مدارهای دیجیتال شده است [۲]. با توجه به اهمیت این نوع از محاسبات، بروز خطا حین انجام کار این نوع از مدارها می‌تواند نتیجه عملیات را بسیار تغییر داده و مشکل‌ساز گردد. بنابراین علاوه بر توان مصرفی کم، قابلیت اطمینان نیز از اهداف مهم در طراحی سیستم‌های محاسباتی امروزی و در نتیجه واحدهای عملیاتی ممیز- شناور حاوی ضرب‌کننده‌ها

این مقاله در تاریخ ۱۲ مهر ماه ۱۳۹۶ دریافت و در تاریخ ۱۶ اردیبهشت ماه ۱۳۹۷ بازنگری شد.

مریم مهاجر، کارشناس ارشد معماری سیستم‌های کامپیوتری، دانشگاه صنعتی نوشیروانی بابل، بابل، (email: maryam.mohajer@stu.nit.ac.ir).

مجتبی ولی‌نجاج، استادیار دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی نوشیروانی بابل، بابل، (email: m.valinataj@nit.ac.ir).

1. Reliability

2. Fault-Tolerant

3. Approximate (or Inexact) Computations

4. Reduced Precision

نشان ندادند و حتی خروجی برخی از برنامه‌ها با کمتر از ۵ بیت مانیتیس نیز قابل قبول بوده است. در نتیجه، بسیاری از برنامه‌ها که با حواس انسان سروکار دارند مانند برنامه‌های مرتبط با پردازش تصویر، صوت و ویدئو، با کاهش دقت محاسبات هنوز می‌توانند خروجی‌های مناسبی تولید نمایند.

در [۲۵] یک روش تشخیص خطا برای جمع‌کننده ممیز- شناور ارائه شده که از یک واحد جمع واریسی‌کننده^۵ با دقت کاهش‌یافته برای تعیین این که آیا نتیجه در محدوده خطای قابل قبول درست است یا خیر استفاده می‌کند. در این روش با کاهش تعدادی از بیت‌های مانیتیس، مساحت و توان مصرفی کاهش می‌یابد. محاسبه کامل جمع به موازات محاسبه افزونه که در واحد جمع واریسی‌کننده انجام می‌شود و در آن فقط بیت‌های پرارزش عملوندها دریافت می‌شود، انجام می‌گردد. در مرحله آخر، نتایج این دو واحد با هم مقایسه شده و وقوع خطا تشخیص داده می‌شود. در [۲۶] تکنیک واریسی مبتنی بر کاهش دقت (RPC) مشابه [۲۵] به ضرب ممیز- شناور اعمال شده تا خطاها را تشخیص دهد. با توجه به این طرح، تشخیص خطا در ضرب‌کننده ممیز- شناور با هزینه معقول امکان‌پذیر است در حالی که امکان تصحیح خطا وجود ندارد.

در [۲۷] دو طرح ساده برای تشخیص و تصحیح خطا در ضرب‌کننده ممیز- شناور ارائه شده که با توجه به کوچک‌تر شدن مدار به خاطر استفاده از محاسبات تقریبی، افزونگی‌های لازم به نحوی اعمال شده‌اند که مساحت یا توان مصرفی مدار جدید در همان حدود مدار اولیه باقی بماند. طرح‌های پیشنهادی در [۲۷] همانند روش‌های قبلی تنها در یک مد کاری قابل استفاده هستند.

با بررسی کارهای گذشته می‌توان دریافت که طرحی مشابه طرح‌های پیشنهادی در این مقاله که قابلیت عمل در دو مد کاری عادی و تحمل‌پذیر اشکال را داشته باشد وجود ندارد. علاوه بر این، غیر از یکی از طرح‌های ارائه‌شده در [۲۷] طرحی مخصوص ضرب‌کننده ممیز- شناور با قابلیت تصحیح خطا در کارهای پیشین وجود ندارد.

۳- مبانی

۳-۱ استاندارد IEEE برای اعداد ممیز- شناور

معمول‌ترین نمایش‌هایی که برای اعداد حقیقی استفاده می‌شوند، روش‌های نمایش ممیز- ثابت و ممیز- شناور هستند [۲۸]. نمایش ممیز- شناور محدوده پویای وسیع‌تر و دقت بالاتری را در مقایسه با نمایش ممیز- ثابت، فراهم اما مساحت و توان مصرفی بیشتری را نیز تحمیل می‌کند. نمایش معمول اعداد ممیز- شناور مطابق استاندارد IEEE-۷۵۴ [۵] در دو قالب نمایش دقت ساده^۶ و دقت مضاعف^۷ انجام می‌شود. قالب [۵] قالب نمایش دقت ساده ۳۲بیتی است که شامل ۱ بیت علامت، ۸ بیت نما^۸ و ۲۳ بیت مانیتیس است. در این مقاله، تمرکز روی قالب نمایش دقت ساده است. شکل ۱ نمایش دودویی قالب دقت ساده و (۱) نحوه محاسبه مقدار عدد را نشان می‌دهد

$$Number = (-1)^{Sign} \times 1.Mantissa \times 2^{Exponent-127} \quad (1)$$

مصرفی و مساحت مورد ارزیابی قرار گرفته و در پایان در بخش ششم، نتیجه‌گیری نهایی ارائه می‌گردد

۲- تحقیقات مرتبط

با توجه به پیچیدگی واحدهای محاسباتی ممیز- شناور از جمله ضرب‌کننده‌های ممیز- شناور، تحمل‌پذیر کردن آنها در برابر اشکال‌ها و خرابی‌های^۱ احتمالی یکی از مسایل مهمی است که باید توجه دقیقی به آن نمود. چندین روش پایه برای بهبود قابلیت اطمینان وجود دارد اما این روش‌ها سربار توان و مساحت زیادی را تحمیل می‌کنند. روش‌های سنتی تشخیص یا تصحیح خطا مانند دونسخه‌سازی و مقایسه^۲، TMR^۳ و افزونگی زمانی (با اجرای مجدد) که در پردازنده‌های ممیز- شناور استفاده شده‌اند، اغلب دارای سربار مساحت و توان زیادی هستند.

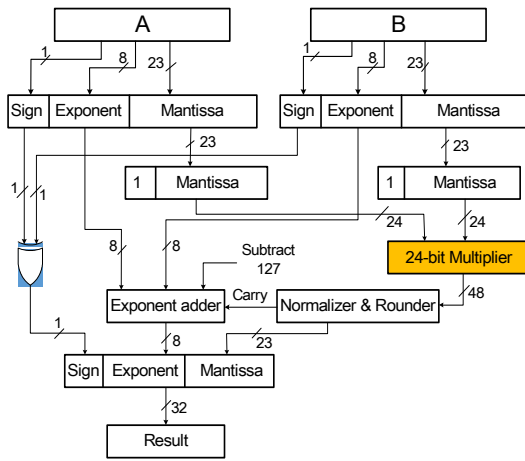
تاکنون واحدهای محاسباتی ممیز- ثابت متنوعی به صورت تحمل‌پذیر اشکال طراحی شده‌اند مانند [۷] تا [۱۱] اما به طراحی واحدهای محاسباتی ممیز- شناور توجه کمتری شده است. در [۱۲] یک واحد محاسباتی ممیز- شناور تشخیص‌دهنده خطا که از کدهای مانده کم‌هزینه استفاده می‌کند پیشنهاد شده است. اما با توجه به ماهیت کدهای مانده، طرح پیشنهادی تنها برای تشخیص خطا می‌تواند به کار رود و کمکی به تصحیح خطا نمی‌کند. در [۱۳] یک روش فقط برای حفاظت مدار کنترلی واحد ممیز- شناور با بررسی بخش نمای عدد ممیز- شناور ارائه شده است. این روش با توجه به عدم بررسی خطا در مانیتیس^۴، تعداد خطاهای کمتری را نسبت به [۱۲] تشخیص می‌دهد و علاوه بر این، فقط برای تشخیص خطاهای گذرا در مدار کنترلی واحد ممیز- شناور مناسب است.

در زمینه محاسبات تقریبی در ضرب‌کننده‌های ممیز- شناور که کاهش دقت محاسبات در آنها با حذف بیت‌های کم‌ارزش‌تر مانیتیس انجام می‌شود، کارهای متنوعی انجام شده است. اما بیشتر روش‌های قبلی مبتنی بر محاسبات تقریبی مانند [۱۴] تا [۱۹] با هدف کاهش توان مصرفی بخش اصلی ضرب‌کننده که ضرب مانیتیس‌ها را انجام می‌دهد، ابداع شده‌اند؛ بدین دلیل که این بخش، بیشترین توان مصرفی (حدود ۸۰٪) را به خود اختصاص داده است [۲۰]. بعضی از روش‌های مرتبط با محاسبات تقریبی نیز بر روی کاهش مساحت مصرفی و افزایش سرعت تمرکز داشته‌اند مانند [۲۱] و [۲۲]. همچنین روش‌های دیگری مبتنی بر محاسبات تقریبی وجود دارند که به صراحت برای کاربردهای مقاوم در برابر خطا ابداع شده‌اند مانند [۲۳] و [۲۴]. اما بایستی توجه داشت که خود این روش‌ها از نوع تحمل‌پذیر خطا یا اشکال نیستند، بلکه روش‌های ابداع‌شده برای کاربردهایی مناسب هستند که مقداری خطای محاسباتی ناشی از محاسبات تقریبی در خروجی آنها قابل قبول است.

نتایج ارائه‌شده در [۶] نشان می‌دهد که همه برنامه‌های کاربردی برای رسیدن به نتایج معتبر، به دقت اولیه فراهم‌شده توسط سخت‌افزار ممیز- شناور نیاز ندارند و می‌توانند تعداد بیت‌های کمتری از مانیتیس و نما را در سخت‌افزار مربوطه به کار گیرند و در نتیجه، مساحت و توان مصرفی می‌تواند کاهش یابد. طبق نتایج ارائه‌شده در [۶]، با کاهش عرض بیت مانیتیس از ۲۳ بیت به ۱۱ بیت در نمایش ممیز- شناور ۳۲بیتی، هیچ یک از برنامه‌های ممیز- شناور تحت آزمون کاهش قابل توجهی در دقت را

5. Checker Adder
6. Reduced Precision Checking
7. Single Precision
8. Double Precision
9. Exponent

1. Failure
2. Duplication and Comparison
3. Triple Modular Redundancy
4. Mantissa



شکل ۲: ضرب‌کننده ممیز- شناور ۳۲بیتی پایه.

Sign (1-bit)	Biased Exponent (8-bit)	Mantissa (23 bit)
--------------	-------------------------	-------------------

شکل ۱: نمایش دودویی قالب دقت ساده.

جدول ۱: مقادیر ممیز- شناور به ازای مقادیر مختلف نما و مانتیس.

$E = 0, M = 0$	عدد صفر
$E = 0, M \neq 0$	اعداد ناهنجار
$0 < E < 255$	اعداد هنجار ممیز- شناور
$E = 255, M = 0$	بی‌نهایت
$E = 255, M \neq 0$	اعداد غیر ممیز- شناور

طبق استاندارد IEEE-754 بخش مانتیس یک عدد ممیز- شناور همیشه کوچک‌تر از یک است اما همواره یک بیت ضمنی '۱' در سمت چپ نقطه اعشار در نظر گرفته می‌شود و همراه با مانتیس، significant ۲۴بیتی را تشکیل می‌دهد. بنابراین significant عددی است که در محدوده (۱, ۲] قرار دارد و به عدد ممیز- شناور حاوی چنین significantی عدد هنجار می‌گویند. بیت علامت، علامت significant را نشان می‌دهد در حالی که علامت نما به طور ضمنی درون آن قرار دارد. در واقع، نما به شکل افزون بر ۱۲۷ نمایش داده شده که با توجه به غیر منفی بودن نما هنگام ذخیره‌شدن (از ۰ تا ۲۵۵) با کم کردن عدد ۱۲۷ از آن نمای واقعی منفی یا مثبت می‌تواند ساخته شود. مقادیر ممیز- شناور به ازای مقادیر مختلف نما و مانتیس در جدول ۱ نشان داده شده است.

۳-۲ ضرب‌کننده ممیز- شناور با دقت ساده

ضرب‌کننده ممیز- شناور ۳۲بیتی عمل ضرب دو عملوند ورودی با دقت ساده را انجام داده و حاصل ضرب را نیز در دقت ساده ۳۲بیتی تولید می‌کند. الگوریتم ضرب دو عدد ممیز- شناور طبق ۶ مرحله انجام می‌شود: مرحله ۱: ضرب مانتیس‌ها به همراه بیت ضمنی (ضرب‌کننده ۲۴بیتی) مرحله ۲: هنجارسازی و گرد کردن مانتیس نتیجه مرحله ۳: جمع نماها مرحله ۴: محاسبه علامت نتیجه مرحله ۵: بررسی وقوع سرریز (یا فروریز) مرحله ۶: ساختن نتیجه استاندارد

شکل ۲ طرح کلی ضرب‌کننده ممیز- شناور ۳۲بیتی را نشان می‌دهد. با توجه به این که در این طرح، بلوک ضرب‌کننده اصلی ۲۴بیتی است، خروجی این ضرب‌کننده عددی ۴۸بیتی است اما طبق استاندارد برای قالب با دقت ساده، پس از جداسازی بیت ضمنی تنها به ۲۳ بیت از آن برای مانتیس نتیجه نیاز است. علاوه بر این، خروجی ۴۸بیتی ضرب‌کننده از یک هنجارساز و گردکننده عبور می‌کند تا در صورت نیاز باعث تغییر مقدار نمای نهایی شود.

۴- طرح‌های پیشنهادی

با توجه به این که در بسیاری از کاربردها قابلیت تحمل محاسبات تقریبی وجود دارد، در این مقاله از روش کاهش دقت محاسبات از طریق حذف بیت‌های کم‌ارزش‌تر استفاده می‌شود تا با توجه به نیاز به سخت‌افزاری کمتر بتوان یک سری افزونگی را برای دستیابی به قابلیت اطمینان از نوع تشخیص یا تصحیح خطا در مد کاری تحمل‌پذیر اشکال اعمال نمود. در واقع ذکر این نکته ضروری است که گرچه بسیاری از کاربردها می‌توانند محاسبات تقریبی را تحمل کنند، اما انحراف در خروجی

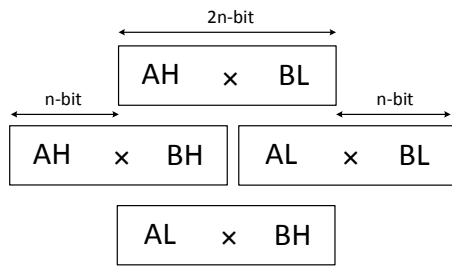
و نتایج کاربردهای گوناگون به دلیل رخداد خطاهای تصادفی هنگام انجام محاسبات، اکثراً غیر قابل قبول هستند. در این مقاله منظور از انحراف در خروجی، خطاهای تولیدشده در خروجی است که به علت رخداد اشکال‌های گذرا یا دائمی درون مدار اصلی، ایجاد می‌شوند. همچنین مدل اشکال فرض‌شده مبتنی بر تغییرات رخ داده در سطح منطقی و صفر یا یک‌شدن سیگنال‌ها است.

ضرب‌کننده ۲۴بیتی نشان داده شده در شکل ۲ که برای ضرب مانتیس‌ها به کار می‌رود، بلوک اصلی ضرب‌کننده ممیز- شناور ۳۲بیتی است. این ضرب‌کننده حدود ۸۰٪ مساحت و توان مصرفی کل ضرب‌کننده ممیز- شناور را به خود اختصاص داده است. بنابراین احتمال رخداد خطا در آن بسیار بیشتر از سایر بلوک‌های درونی است. به همین دلیل، قابلیت تشخیص یا تصحیح خطا را اساساً روی این بلوک اعمال می‌نماییم.

۴-۱ معماری ضرب‌کننده ممیز- شناور با قابلیت تشخیص خطا

ضرب‌کننده مانتیس استفاده‌شده در معماری پیشنهادی، یک ضرب‌کننده با ساختار ماجولار است. این نوع ضرب‌کننده بر اساس جمع حاصل ضرب‌های پاره‌ای با ساختارهای تکرارشونده عمل می‌کند. بنابراین اگر ضرب‌کننده مورد نظر $2n \times 2n$ بیتی باشد (یعنی عملوندهای آن $2n$ بیتی باشند) آن گاه می‌توانیم هر عملوند ورودی را به دو نیمه n بیتی مرتبه بالا و مرتبه پایین تقسیم نموده و ضرب‌کننده را بر اساس ضرب $n \times n$ بیتی طراحی نماییم [۲۸]. برای مثال اگر عملوند $2n$ بیتی A را به صورت $A = A_H . A_L$ نشان دهیم، بدین معنا است که دو بخش مرتبه بالا و مرتبه پایین A_H و A_L ، n بیتی هستند. بدین ترتیب می‌توان ضرب‌کننده $2n$ بیتی را با استفاده از ماجول‌های کوچک‌تر ضرب‌کننده n بیتی طراحی نمود. به عنوان مثال، روند انجام عملیات ضرب میان دو عملوند $2n$ بیتی A و B که به دو نیمه بالا و پایین تقسیم شده‌اند در شکل ۳ نشان داده شده است. تقسیم کردن هر عملوند به دو نیمه، چهار حاصل ضرب پاره‌ای را تولید می‌کند. در این شکل، ضرایب 2^n و 2^{2n} برای هم‌تراز کردن وزن حاصل ضرب‌های پاره‌ای استفاده شده‌اند. شکل ۴ نیز به نحوی تفاوت وزن حاصل ضرب‌های پاره‌ای و نیاز به هم‌ترازی آنها را قبل از جمع نمودن آنها با هم نشان می‌دهد. برای افزایش سرعت جمع و انجام آن به طور موازی، هرگاه تعداد عملوندهای جمع‌شونده بیش از دو باشد، استفاده از جمع‌کننده‌ای با نام CSA^۱ سودمند خواهد بود. شکل ۵

1. Carry Save Adder



شکل ۵: سازماندهی مناسب حاصل ضرب‌های پاره‌ای برای انجام ضرب $2n$ بیتی $A \times B$.

و تحمل‌پذیر اشکال با قابلیت تشخیص خطا در شکل ۶ نشان داده شده است. در این شکل، بخش‌های رنگی نشان‌دهنده بلوک‌های تغییر یافته ساختار اولیه ضرب‌کننده ممیز- شناور یا بلوک‌های جدیدی هستند که به ساختار اولیه اضافه شده‌اند. با توجه به شکل ۶، دو حاصل ضرب $2m$ بیتی تولیدی از بلوک‌های هنجارساز و گردکننده عبور کرده و حاصل آنها به k بیت گرد می‌شود. اگر خروجی‌های متناظر با دو ضرب‌کننده m بیتی با هم برابر نباشند یا رقم‌های نقلی تولید شده توسط هنجارسازها با هم برابر نباشند، سیگنال خطا برابر یک شده و بروز خطا اعلام می‌شود. در طرح پیشنهادی شکل ۶ فرض بر این است که اگر دو عملوند ورودی یک مقایسه‌کننده با هم برابر نباشند، خروجی آن برابر یک می‌شود. ضمناً در این شکل، سیگنال ورودی mode با مقادیر صفر یا یک به ترتیب تعیین‌کننده مدهای کاری عادی یا تحمل‌پذیر اشکال است.

پیاده‌سازی هر کدام از دو نسخه ضرب‌کننده مورد نیاز با دقت کاهش یافته m بیتی ($m = 12 + t$) طبق شکل ۷ انجام می‌شود. با توجه به حاصل ضرب‌های پاره‌ای تولید شده در این ضرب‌کننده، هر ضرب‌کننده m بیتی به یک ضرب‌کننده ۱۲ بیتی (12×12)، دو ضرب‌کننده $12 \times t$ و یک ضرب‌کننده t بیتی ($t \times t$) احتیاج دارد. از طرف دیگر، همان طور که قبلاً بیان شد، ضرب‌کننده ۲۴ بیتی اصلی را می‌توان با توجه به شکل ۵ با استفاده از چهار ضرب‌کننده ۱۲ بیتی ساخت. علاوه بر این، دو ضرب‌کننده $12 \times t$ را تا زمانی که t بزرگ‌تر از ۶ نباشد می‌توان به طور کامل با یک ضرب‌کننده ۱۲ بیتی پیاده‌سازی کرد زیرا هر ضرب‌کننده ۱۲ بیتی خود می‌تواند از چهار ضرب‌کننده ۶ بیتی تشکیل شود (هر ضرب‌کننده $12 \times t$ از دو ضرب‌کننده $6 \times t$ تشکیل می‌شود). در نتیجه، مقدار t در رابطه $m = 12 + t$ حداکثر برابر ۶ انتخاب می‌شود ($t \leq 6$). مسلماً برای t های بزرگ‌تر امکان پیاده‌سازی وجود دارد اما به علت عدم وجود مدار اولیه برای انجام کامل آن، سربار زیادی تحمیل خواهد شد.

در ادامه، معماری پیشنهادی برای تجزیه ضرب‌کننده اصلی ۲۴ بیتی به ضرب‌کننده‌های کوچک‌تر برای استفاده در دو مد کاری شرح داده می‌شود. ساختار ضرب‌کننده اصلی ۲۴ بیتی برای استفاده در مدهای کاری عادی و تحمل‌پذیر اشکال با قابلیت تشخیص خطا به صورت شکل ۸ پیشنهاد می‌شود. در این شکل، نسخه اول ضرب‌کننده با دقت کاهش یافته m بیتی به کمک دو ضرب‌کننده ۱۲ بیتی بالایی و سمت راست و نسخه دوم ضرب‌کننده با دقت کاهش یافته m بیتی به کمک دو ضرب‌کننده ۱۲ بیتی پایینی و سمت چپ پیاده‌سازی شده است. با توجه به این که ضرب‌کننده تغییر یافته بایستی قابلیت اجرا در هر کدام از مدهای کاری را داشته باشد، یک سری مالتی‌پلکسر برای انتخاب ورودی‌های متفاوت، درون ضرب‌کننده قرار داده شده‌اند. در هر نسخه ضرب‌کننده m بیتی یک ضرب‌کننده ۱۲ بیتی (در نسخه اول ماجول $L \times L$ و در نسخه دوم ماجول $H \times H$) بدون تغییر در ساختار درونی برای پیاده‌سازی ضرب‌کننده 12×12 شکل ۷ استفاده می‌شود. اما در هر دو نسخه، ضرب‌کننده

$$\begin{aligned} P &= A \times B = A_H \cdot A_L \times B_H \cdot B_L \\ &= A_L \times B_L + A_H \times B_L \times 2^n \\ &\quad + A_L \times B_H \times 2^n + A_H \times B_H \times 2^{2n} \end{aligned}$$

شکل ۳: ضرب دو عملوند A و B با تجزیه به ضرب‌های کوچک‌تر.

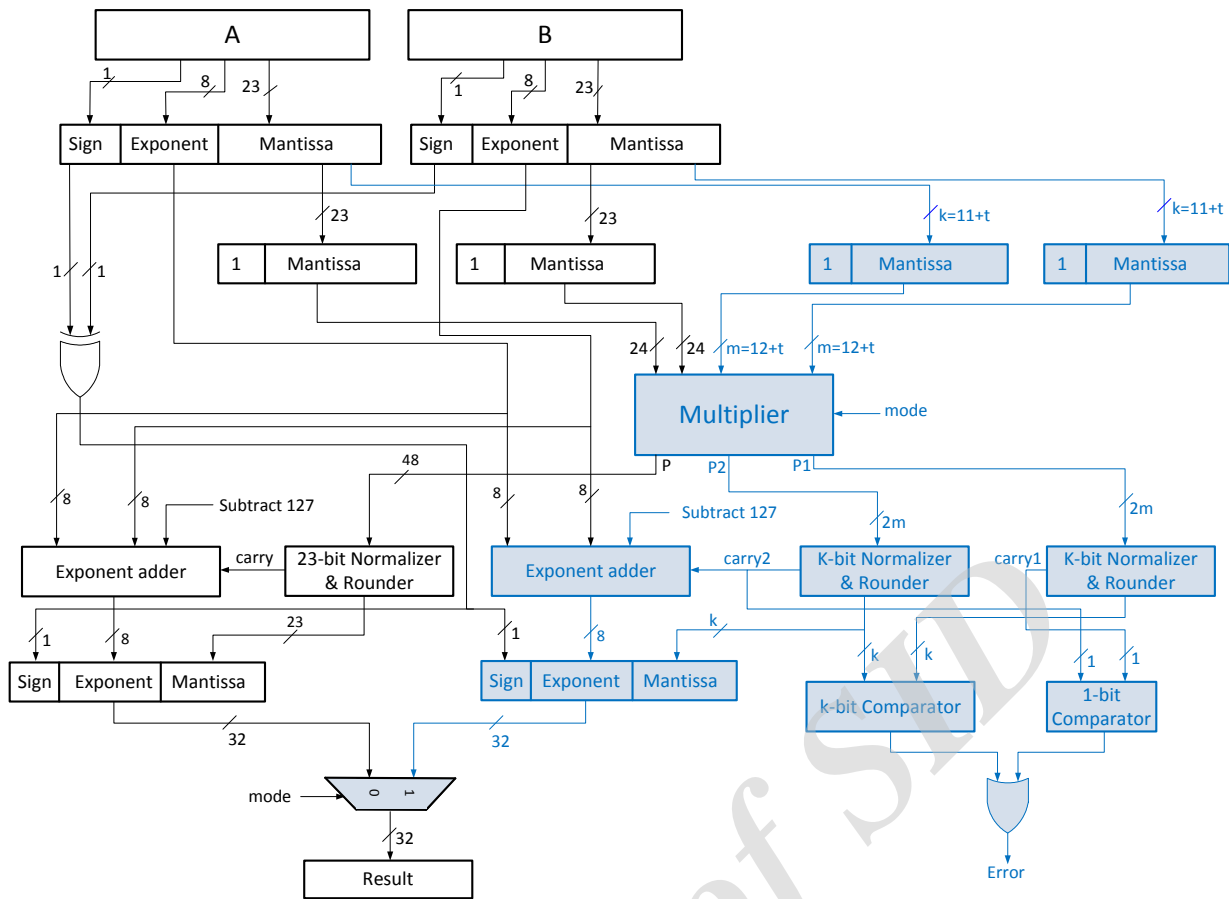
$$\begin{array}{r} A_H \cdot A_L \\ \times B_H \cdot B_L \\ \hline A_L \times B_L \\ + A_H \times B_L \\ + A_L \times B_H \\ + A_H \times B_H \\ \hline \text{Product} \end{array}$$

شکل ۴: ضرب $A \times B$ و تولید حاصل ضرب‌های پاره‌ای با وزن‌های متفاوت.

سازماندهی مناسب چهار حاصل ضرب پاره‌ای تولید شده را برای کاهش تعداد سطوح در استفاده از CSA نشان می‌دهد. با این ساختار عمل ضرب در دو مرحله انجام خواهد شد. در مرحله اول، چهار ضرب‌کننده کوچک‌تر m بیتی حاصل ضرب‌های پاره‌ای را تولید می‌کنند. سپس در مرحله دوم حاصل ضرب‌های پاره‌ای با استفاده از چینش نشان داده شده در شکل ۵ با هم جمع شده و نتیجه نهایی $4m$ بیتی تولید می‌شود.

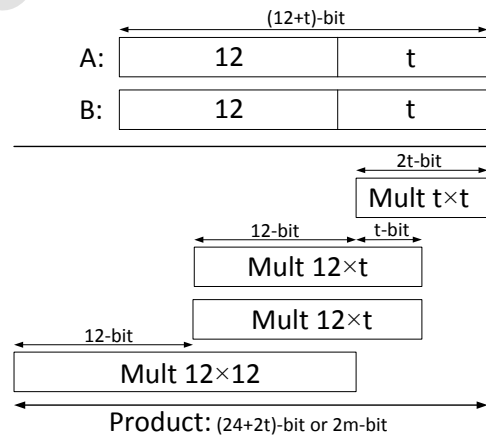
اعمال معماری ضرب‌کننده ماجولار توضیح داده شده به بخش اصلی ضرب‌کننده ممیز- شناور با دقت ساده که در شکل ۲ با رنگ متفاوت نشان داده شده است امکان‌پذیر است. بنابراین با انجام این کار امکان استفاده از ضرب‌کننده‌های کوچک‌تر درونی برای انجام محاسبات افزونه برای دسترسی به قابلیت اطمینان در مد تحمل‌پذیر اشکال فراهم خواهد شد. بنابراین ضرب‌کننده ۲۴ بیتی در شکل ۲ که برای ضرب دو significant عملوندهای ورودی به کار می‌رود، می‌تواند با استفاده از چهار ضرب‌کننده ۱۲ بیتی مشابه شکل ۵ پیاده‌سازی شود.

ویژگی ضرب‌کننده ممیز- شناور پیشنهادی در این بخش این است که می‌تواند در مد کاری عادی (با دقت کامل شامل یک ضرب‌کننده ۲۴ بیتی) یا مد کاری تحمل‌پذیر اشکال با قابلیت تشخیص خطا (با دقت کاهش یافته شامل دو ضرب‌کننده کوچک‌تر m بیتی) کار کند. با توجه به وجود تعداد کافی ضرب‌کننده ۱۲ بیتی، حداقل مقدار m را برابر با ۱۲ در نظر می‌گیریم که به معنای مانیتیس ۱۱ بیتی و حذف ۱۲ بیت کم‌ارزش‌تر مانیتیس عملوندهای ورودی است. در مد تحمل‌پذیر اشکال با توجه به این نوع حذف، برخی از ضرب‌کننده‌های ۱۲ بیتی آزاد می‌شوند که می‌توانند برای انجام محاسبات افزونه با هدف تشخیص (یا تصحیح خطا) به کار روند. بنابراین برای دستیابی به قابلیت تشخیص خطا در ضرب‌کننده ممیز- شناور، به جای یک ضرب‌کننده ۲۴ بیتی از دو ضرب‌کننده با دقت کاهش یافته m بیتی ($m = 12 + t$ و $t \leq 6$ که دلیل آن در ادامه توضیح داده خواهد شد) طبق مفهوم روش دونسخه‌سازی و مقایسه استفاده می‌نماییم. بنابراین هر مانیتیس ۲۳ بیتی از عملوندهای ورودی به یک مانیتیس k بیتی جدید ($k = 11 + t$) تبدیل می‌شود (یعنی حذف $23 - k$ بیت کم‌ارزش‌تر مانیتیس) و این مانیتیس به همراه یک بیت ضمنی، یک significant m بیتی ($m = k + 1$) را تشکیل می‌دهد. در معماری با قابلیت تشخیص خطا به جای یک ضرب‌کننده ۲۴ بیتی با خروجی ۴۸ بیتی، دو ضرب‌کننده با دقت کاهش یافته m بیتی استفاده می‌شود تا دو خروجی $2m$ بیتی تولید شده بعد از عبور از واحد هنجارساز و گردکننده توسط مقایسه‌کننده با هم مقایسه شوند تا خطای احتمالی تشخیص داده شود. ساختار کلی ضرب‌کننده ممیز- شناور پیشنهادی در دو مد کاری عادی



شکل ۶: ساختار ضرب‌کننده پیشنهادی ممیز- شناور در دو مد کاری عادی و تحمل‌پذیر اشکال با قابلیت تشخیص خطا.

کاری، نیازی به استفاده از مالتی‌پلکسر در ورودی‌های آن نیست. با توجه به قابلیت پیاده‌سازی ضرب‌کننده $12 \times t$ با استفاده از دو ضرب‌کننده عیبی، پیاده‌سازی دو ضرب‌کننده $12 \times t$ مورد نیاز در هر نسخه $(A_H \times B_{LF})$ و $(B_H \times A_{LF})$ ، به چهار ضرب‌کننده عیبی نیاز دارد. از طرف دیگر، هر ضرب‌کننده 12 بیتی خود می‌تواند از چهار ضرب‌کننده عیبی تشکیل شود. بنابراین ضرب‌کننده 12 بیتی شماره ۱ (ماجول $H \times L$) مطابق شکل ۹ به چهار ضرب‌کننده عیبی تجزیه می‌شود تا برای تولید دو حاصل ضرب $12+t$ بیتی (خروجی‌های ضرب‌کننده‌های شماره ۴ (ماجول $L \times H$) نیز انجام می‌گردد و هر دو نسخه ضرب‌کننده m بیتی ضرب‌های درونی خود را به طور موازی انجام می‌دهند. در نتیجه، همه حاصل ضرب‌های پاره‌ای مورد نیاز طبق شکل ۷ برای هر دو نسخه فراهم می‌گردند که پس از آن با استفاده از جمع‌کننده از نوع CSA با هم جمع شده و برای هر نسخه یک حاصل $2m$ بیتی تولید می‌شود. ادامه کار مطابق شکل ۶ انجام خواهد شد.



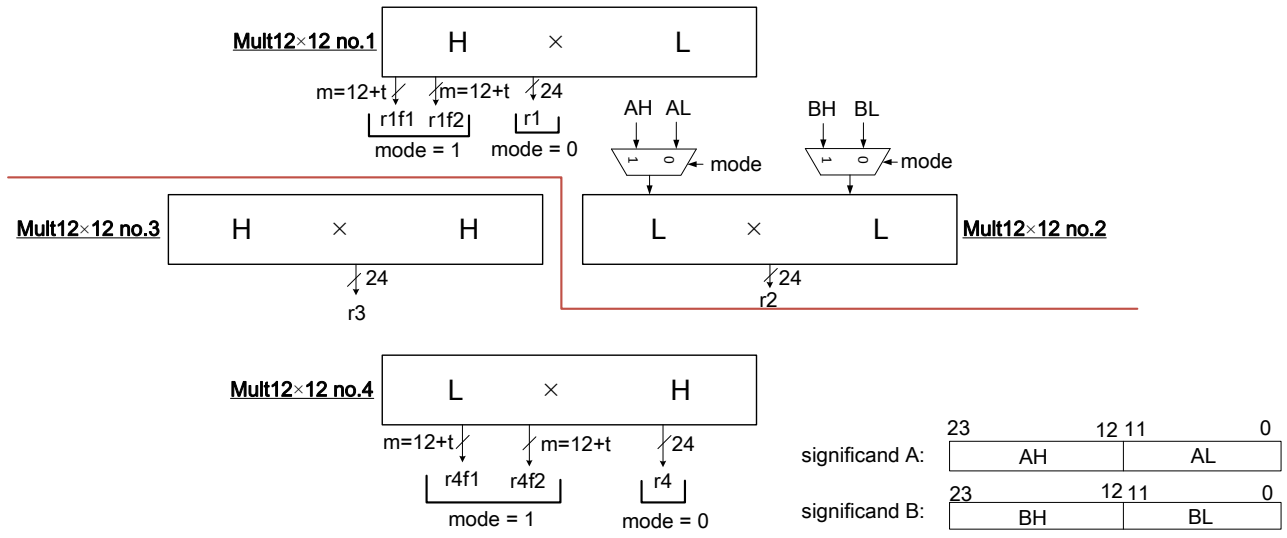
شکل ۷: نحوه انجام ضرب m یا $(12+t)$ بیتی.

12 بیتی باقیمانده (ماجول‌های $L \times H$ و $H \times L$) باید مطابق شکل ۹ برای پیاده‌سازی دو ضرب‌کننده $12 \times t$ درونی طراحی شود. البته شکل ۹ جزئیات معماری درونی ضرب‌کننده $12 \times t$ بیتی را نشان می‌دهد که قابل تعمیم به ضرب‌کننده $12 \times H$ است. بدین ترتیب تنها ضرب‌کننده t بیتی $(t \times t)$ مورد نیاز درون هر ضرب‌کننده m بیتی (کوچک‌ترین حاصل ضرب پاره‌ای در شکل ۷) بایستی به طور جداگانه طراحی شده و به مدار اضافه گردد.

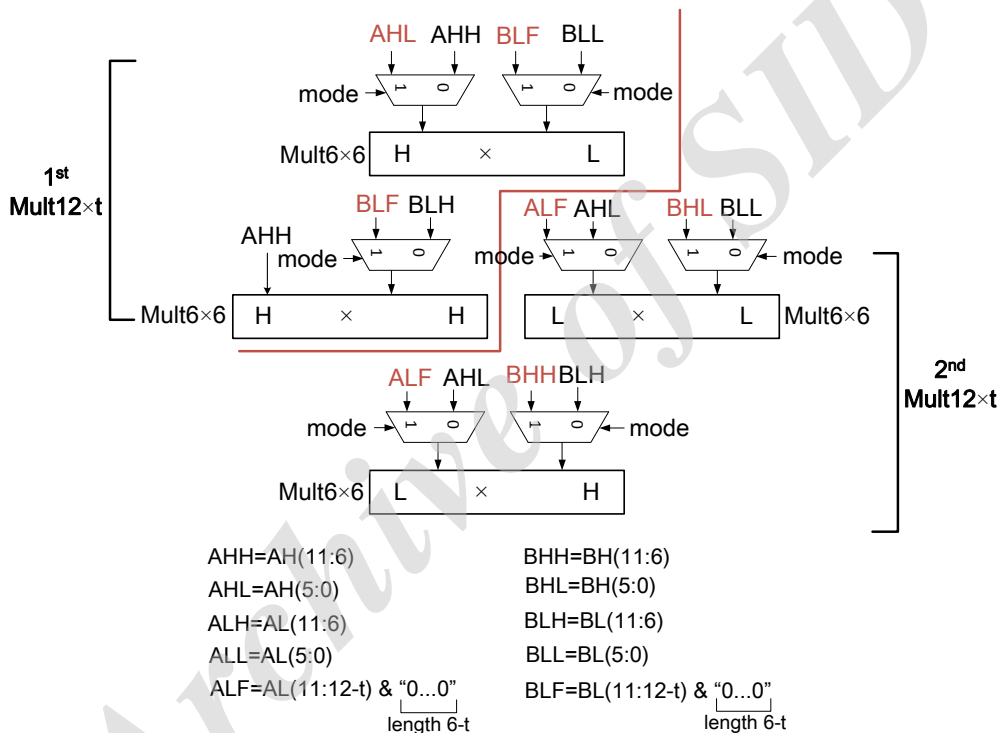
با توجه به شکل ۸، ضرب‌کننده 12 بیتی شماره ۲ (ماجول $L \times L$)، در مد تحمل‌پذیر اشکال حاصل ضرب 24 بیتی $A_H \times B_H$ را برای نسخه اول ضرب‌کننده m بیتی تولید می‌کند (r_1)، در حالی که در نسخه دوم، ضرب‌کننده 12 بیتی شماره ۳ (ماجول $H \times H$) همین عملیات ضرب را انجام می‌دهد که با توجه به نیاز به اجرای این عملیات در هر دو مد

۲-۴ معماری ضرب‌کننده ممیز- شناور با قابلیت تصحیح خطا

طرح کلی پیشنهادی برای ضرب‌کننده ممیز- شناور در دو مد کاری عادی و تحمل‌پذیر اشکال با قابلیت تصحیح خطا در شکل ۱۰ آمده است. در این شکل، بخش‌های رنگی نشان‌دهنده بلوک‌های تغییر یافته ساختار اولیه ضرب‌کننده ممیز- شناور یا بلوک‌های جدیدی هستند که به ساختار اولیه اضافه شده‌اند. در این طرح، برای دست‌یافتن به قابلیت تصحیح خطا در مد تحمل‌پذیر اشکال، ضرب‌کننده 24 بیتی اولیه که خروجی 48 بیتی P را تولید می‌کند به گونه‌ای تنظیم می‌شود که سه عملیات ضرب با



شکل ۸: ساختار ضرب کننده اصلی ۲۴ بیتی پیشنهادی برای استفاده در مدهای کاری عادی و تحمل پذیر اشکال با قابلیت تشخیص خطا.

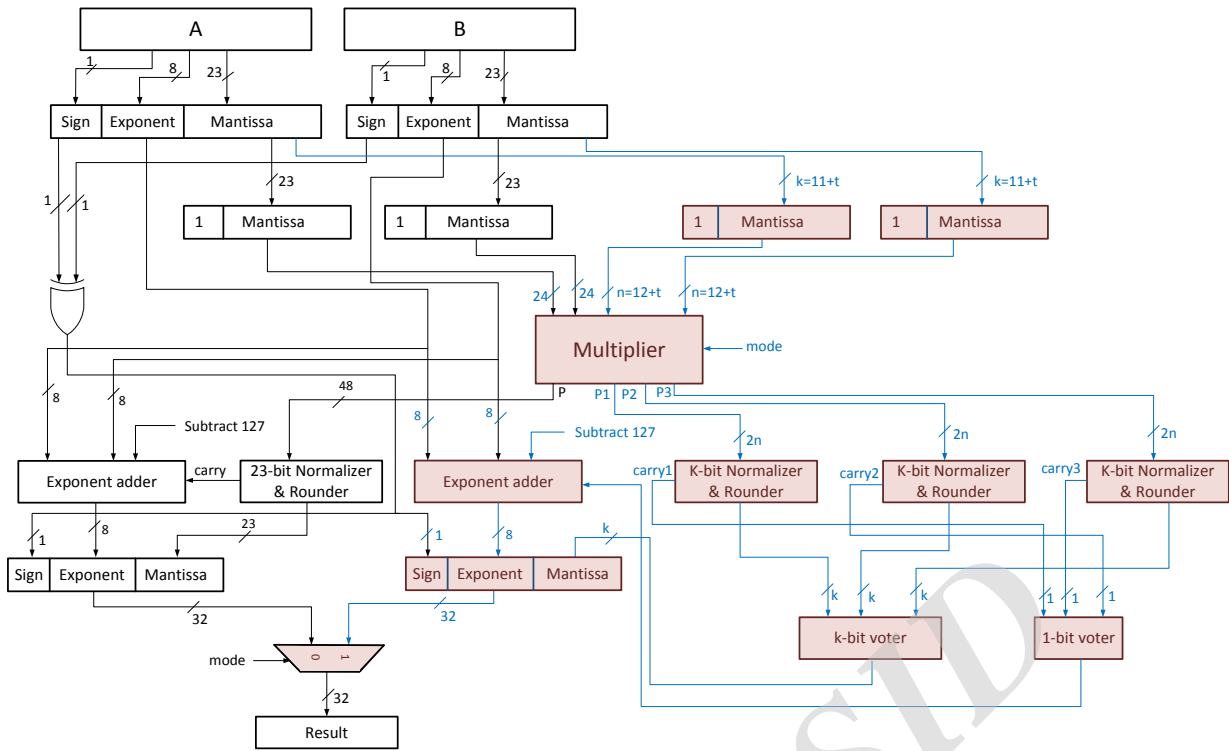


شکل ۹: جزئیات معماری درونی ضرب کننده ۱۲ بیتی ماجول $H \times L$ در شکل ۸.

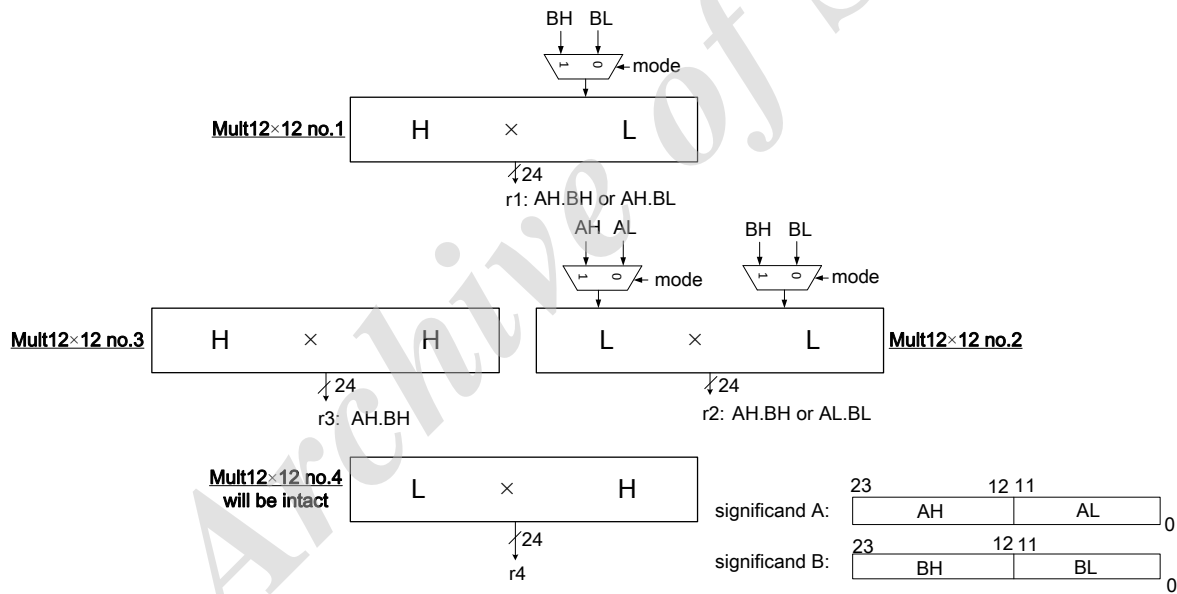
تبدیل می کند. نحوه عملکرد ساختار شکل ۱۰ در مد عادی مشابه ساختار شکل ۶ است.

اگرچه در مد تحمل پذیر اشکال از ضرب کننده ۲۴ بیتی اولیه برای تولید سه حاصل ضرب با طول مساوی استفاده می شود، اما با توجه به اندازه ضرب کننده های لازم، ممکن است به ضرب کننده های اضافه کوچک تری نیز احتیاج باشد. بنابراین اگر در ضرب کننده با دقت کاهش یافته n بیتی $(n = 12 + t)$ برابر صفر باشد، تنظیم و استفاده از ضرب کننده های ۱۲ بیتی درون ضرب کننده ۲۴ بیتی به راحتی و مطابق شکل ۱۱ انجام خواهد شد. با توجه به نیاز به سه ضرب کننده ۱۲ بیتی، یکی از ضرب کننده های درونی (شماره ۴ در شکل ۱۱) بلااستفاده خواهد ماند. واضح است که برای n های بزرگ تر از ۱۲ ($t > 0$)، ساختاری مشابه شکل ۹ اما برای سه ضرب کننده n بیتی مورد نیاز است که با توجه به نیاز به ضرب کننده های اضافه جدید، سربار کلی افزایش واضحی خواهد داشت.

دقت کاهش یافته n بیتی ($n = 12 + t$ و $t \leq 6$) را انجام دهد تا سه خروجی n بیتی P_1 ، P_2 و P_3 تولید شود. این سه خروجی بعد از عبور از بلوک های هنجارساز و گردکننده توسط رأی دهنده با هم مقایسه شده و خطای احتمالی تصحیح یا پوشانده می شود. در شکل ۱۰ برای رقم های نقلی تولید شده توسط بلوک های هنجارساز و گردکننده نیز رأی دهنده یک بیتی گذاشته شده تا به طور جداگانه تصحیح خطا در رقم نقلی را انجام دهد. در ضرب کننده جدید، هر مانتیس ۲۳ بیتی به یک مانتیس k بیتی جدید ($k = 11 + t$) تبدیل می شود که به همراه یک بیت ضمنی، یک significant n بیتی ($n = k + 1$) را تشکیل می دهد. همان طور که در شکل ۱۰ مشاهده می شود، بلوک هنجارساز و گردکننده سه تایی شده است تا قابلیت تصحیح خطا در کل طرح افزایش یابد. همچنین همانند شکل ۶، بلوک سازنده نتیجه استاندارد، نتیجه خارج شده از رأی دهنده k بیتی را با گذاشتن تعداد مورد نیاز صفر در سمت راست آن و ترکیب با توان و علامت عدد نهایی به قالب نمایش دقت ساده ۳۲ بیتی



شکل ۱۰: ساختار ضرب‌کننده پیشنهادی ممیز-شناور در دو مد کاری عادی و تحمل‌پذیر اشکال با قابلیت تصحیح خطا.



شکل ۱۱: ساختار ضرب‌کننده اصلی ۲۴ بیتی پیشنهادی برای استفاده در مدهای کاری عادی و تحمل‌پذیر اشکال با قابلیت تصحیح خطا برای significant ۱۲ بیتی.

ضرب‌کننده‌های پیشنهادی به ازای عرض بیت‌های مختلف مانتیس، در مقایسه با ضرب‌کننده پایه نشان داده شده است. با توجه به این جدول‌ها می‌توان نتیجه گرفت که انتخاب عرض مانتیس برابر با ۱۱ برای هر دو طرح پیشنهادی، سربار ناچیزی را خصوصاً برای تأخیر و مساحت مورد نیاز تحمیل می‌کند در حالی که مطابق [۶] برای این اندازه مانتیس بسیاری از برنامه‌های کاربردی دارای دقت کافی در خروجی خود خواهند بود. همچنین می‌توان برای ساختار با قابلیت تشخیص خطا عرض مانتیس برابر با ۱۲ یا ۱۳ را نیز مورد استفاده قرار داد زیرا به سربارهای متعارفی احتیاج دارند.

با توجه به نتایج ارائه‌شده در این مقاله و [۶] می‌توان دریافت که برنامه‌هایی که به نحوی با پردازش صوت، تصویر یا ویدئو در ارتباط هستند، در صورت استفاده از مانتیس حداقل ۱۱ بیتی در ضرب‌کننده

۵- نتایج پیاده‌سازی و ارزیابی

طرح‌های پیشنهادی با تمام افزودنی‌های اعمال‌شده به همراه طرح پایه با استفاده از زبان توصیف سخت‌افزار VHDL پیاده‌سازی و شبیه‌سازی شده و سپس سنتز آنها توسط ابزار synopsys انجام شده است تا مساحت و توان مورد نیاز آنها به دست آید. همچنین مساحت و توان مصرفی طرح‌های پیشنهادی به ازای عرض بیت‌های مختلف مانتیس (k) اندازه‌گیری شده است. نتایج پیاده‌سازی با استفاده از کتابخانه استاندارد CMOS ۶۵ nm LPLVT STMicroelectronics (با ولتاژ تغذیه ۱٫۲ V و دمای ۲۵°C) در جدول‌های ۲ و ۳ به ترتیب برای ساختارهای با قابلیت تشخیص خطا و تصحیح خطا آورده شده است. در این دو جدول، تأخیر، مساحت، توان و سربارهای مربوط به آنها در

جدول ۲: نتایج پیاده‌سازی طرح ضرب‌کننده ممیز- شناور پیشنهادی حاوی دو مد عادی و تحمل‌پذیر اشکال با قابلیت تشخیص خطا در مقایسه با طرح پایه.

اندازه مانیتیس بر حسب بیت در مد تحمل‌پذیر اشکال	تأخیر (نانوثانیه)	سربار تأخیر	مساحت (میکرومتر مربع)	سربار مساحت	توان (میلی‌وات)	سربار توان	احتمال تشخیص خطا
$k = 23$ (طرح پایه)	۸,۷۰	-	۱۱۲۹۰,۲	-	۵,۶۲	-	۰
$k = 11$	۹,۰۴	۳,۹%	۱۲۵۲۴,۷	۱۰,۹%	۶,۴۹	۱۵,۵%	۸۱,۵%
$k = 12$	۸,۸۷	۲,۰%	۱۳۶۴۹,۵	۲۰,۹%	۶,۵۹	۱۷,۳%	۸۳,۱%
$k = 13$	۸,۸۷	۲,۰%	۱۴۰۸۲,۶	۲۴,۷%	۷,۰۹	۲۶,۳%	۸۳,۳%
$k = 15$	۸,۸۷	۲,۰%	۱۴۶۹۴,۷	۳۰,۲%	۷,۹۰	۴۰,۶%	۸۳,۷%

جدول ۳: نتایج پیاده‌سازی طرح ضرب‌کننده ممیز- شناور پیشنهادی حاوی دو مد عادی و تحمل‌پذیر اشکال با قابلیت تصحیح خطا در مقایسه با طرح پایه.

اندازه مانیتیس بر حسب بیت در مد تحمل‌پذیر اشکال	تأخیر (نانوثانیه)	سربار تأخیر	مساحت (میکرومتر مربع)	سربار مساحت	توان (میلی‌وات)	سربار توان	احتمال تصحیح خطا
$k = 23$ (طرح پایه)	۸,۷۰	-	۱۱۲۹۰,۲	-	۵,۶۲	-	۰
$k = 11$	۹,۰۰	۳,۴%	۱۲۶۹۸,۴	۱۲,۵%	۶,۷۰	۱۹,۳%	۸۱,۶%
$k = 12$	۸,۸۸	۲,۱%	۱۶۳۶۲,۳	۴۵,۰%	۷,۷۷	۳۸,۳%	۸۵,۵%
$k = 13$	۸,۸۸	۲,۱%	۱۷۲۱۹,۳	۵۲,۵%	۸,۳۹	۴۹,۳%	۸۶,۱%
$k = 15$	۸,۸۸	۲,۱%	۱۹۱۴۹,۰	۶۹,۶%	۹,۸۱	۷۴,۶%	۸۷,۲%

جدول ۴: ضرب‌کننده‌های ممیز- شناور پیشنهادی در مقایسه با روش‌های قبلی.

طرح به همراه اندازه مانیتیس (دقت نتایج خروجی)	توانایی کار در دو مد عادی و تحمل‌پذیر اشکال	قابلیت تحمل‌پذیری اشکال	سربار مساحت	سربار توان	حداکثر خطای نسبی خروجی
DMR با مانیتیس ۲۳بیتی	خیر	تشخیص	بیش از ۱۰%	بیش از ۱۰%	۰
TMR با مانیتیس ۲۳بیتی	خیر	تصحیح	بیش از ۲۰%	بیش از ۲۰%	۰
طرح‌های ارائه‌شده در [۱۲] با مانیتیس ۲۳بیتی	خیر	تشخیص	از ۳۰% تا ۵۰%	نامعلوم	۰
طرح ارائه‌شده در [۲۶] با مانیتیس ۷بیتی	خیر	تشخیص	۱۷,۸%	۲۸%	۰,۰۰۷۸۱
طرح ارائه‌شده در [۲۷] با مانیتیس ۱۵بیتی	خیر	تصحیح	۲۸,۹%	۱۲,۶%	۰,۰۰۰۰۳
معماری پیشنهادی با مانیتیس ۱۱بیتی (شکل ۶)	بله	تشخیص	۱۰,۹%	۱۵,۵%	۰,۰۰۰۴۹
معماری پیشنهادی با مانیتیس ۱۱بیتی (شکل ۱۰)	بله	تصحیح	۱۲,۵%	۱۹,۲%	۰,۰۰۰۴۹

ممیز- شناور از کیفیت مناسبی برخوردار خواهند بود. نمونه‌های مناسبی از این برنامه‌ها که در [۶] مورد ارزیابی قرار گرفته‌اند عبارتند از برنامه شناسایی سیگنال صحبت، برنامه طبقه‌بندی کننده اثر انگشت در سطح الگو و الگوریتم‌های دوبعدی DCT^۱ و معکوس DCT که دقت خروجی همه آنها بیش از ۹۰% بوده است.

با توجه به جدول‌های ۲ و ۳، در طرح‌های پیشنهادی برای تمام عرض مانیتیس‌های ذکر شده، احتمال تشخیص یا تصحیح خطا از ۸۱% تا ۸۷% تغییر می‌کند. در اینجا احتمال تشخیص یا تصحیح خطا در کل طرح با محاسبه نسبت مساحت بلوک‌های دارای ویژگی تشخیص یا تصحیح خطا به کل مساحت مصرفی معماری پیشنهادی به دست آمده است. در واقع با توجه به این که مشخص است در چه بلوک‌هایی تشخیص یا تصحیح خطا به طور کامل انجام می‌شود و مساحت این بلوک‌ها نیز در سنتز نهایی به دست آمده است، این محاسبات انجام شده و نیازی به تزریق اشکال^۲ در معماری‌های پیشنهادی نیست. در طرح‌های پیشنهادی، مساحت کل ضرب‌کننده اصلی و اجزای آن به همراه بخش‌های تکرار شده دارای قابلیت اطمینان در برابر اشکال هستند.

با توجه به ماهیت معماری‌های پیشنهادی در این مقاله که قابلیت عمل در دو مد کاری عادی و تحمل‌پذیر اشکال (از نوع تشخیص خطا یا نوع تصحیح خطا) با مانیتیس ۱۱بیتی، دارای وضعیت مناسب‌تری نسبت به روش‌های قبلی هستند. باید توجه داشت که روش پایه دونسخه‌سازی

ممیز- شناور از کیفیت مناسبی برخوردار خواهند بود. نمونه‌های مناسبی از این برنامه‌ها که در [۶] مورد ارزیابی قرار گرفته‌اند عبارتند از برنامه شناسایی سیگنال صحبت، برنامه طبقه‌بندی کننده اثر انگشت در سطح الگو و الگوریتم‌های دوبعدی DCT^۱ و معکوس DCT که دقت خروجی همه آنها بیش از ۹۰% بوده است.

با توجه به جدول‌های ۲ و ۳، در طرح‌های پیشنهادی برای تمام عرض مانیتیس‌های ذکر شده، احتمال تشخیص یا تصحیح خطا از ۸۱% تا ۸۷% تغییر می‌کند. در اینجا احتمال تشخیص یا تصحیح خطا در کل طرح با محاسبه نسبت مساحت بلوک‌های دارای ویژگی تشخیص یا تصحیح خطا به کل مساحت مصرفی معماری پیشنهادی به دست آمده است. در واقع با توجه به این که مشخص است در چه بلوک‌هایی تشخیص یا تصحیح خطا به طور کامل انجام می‌شود و مساحت این بلوک‌ها نیز در سنتز نهایی به دست آمده است، این محاسبات انجام شده و نیازی به تزریق اشکال^۲ در معماری‌های پیشنهادی نیست. در طرح‌های پیشنهادی، مساحت کل ضرب‌کننده اصلی و اجزای آن به همراه بخش‌های تکرار شده دارای قابلیت اطمینان در برابر اشکال هستند.

با توجه به ماهیت معماری‌های پیشنهادی در این مقاله که قابلیت عمل در دو مد کاری عادی و تحمل‌پذیر اشکال (از نوع تشخیص خطا یا

1. Discrete Cosine Transform
2. Fault Injection

- Automation Conf., DAC'12*, pp. 820-825, San Francisco, CA, USA, 3-7 Jun. 2012.
- [4] S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Computing approximately, and efficiently," in *Proc. Design, Automation & Test in Europe Conf., DATE'15*, pp. 748-751, Grenoble, France, 9-13 Mar. 2015.
- [5] IEEE Computer Society (2008), IEEE standard for floating-point arithmetic, IEEE Standard 754-2008.
- [6] J. Ying, F. Tong, D. Nagle, and R. A. Rutenbar, "Reducing power by optimizing the necessary precision/range of floating-point arithmetic," *IEEE Trans. VLSI Systems*, vol. 8, no. 3, pp. 273-286, Jun. 2000.
- [7] I. Z. Milovanovic, E. I. Milovanovic, M. K. Stojcev, and M. P. Bekakos, "Orthogonal fault-tolerant systolic arrays for matrix multiplication," *Microelectronics Reliability*, vol. 51, no. 3, pp. 711-725, Mar. 2011.
- [8] M. Fazeli, A. Namazi, S. G. Miremadi, and A. Haghdoust, "Operand width aware hardware reuse: a low cost fault-tolerant approach to ALU design in embedded processors," *Microelectronics Reliability*, vol. 51, no. 12, pp. 2374-2387, Dec. 2011.
- [9] P. Reviriego, S. Z. Can, C. Eryilmaz, J. A. Maestro, and O. Ergin, "Exploiting processor features to implement error detection in reduced precision matrix multiplications," *Microprocessors and Microsystems*, vol. 38, no. 6, pp. 581-584, Aug. 2014.
- [10] A. Mukherjee and A. S. Dhar, "Real-time fault-tolerance with hot-standby topology for conditional sum adder," *Microelectronics Reliability*, vol. 55, no. 3-4, pp. 704-712, Feb./Mar. 2015.
- [11] M. Valinataj, "Fault-tolerant carry look-ahead adder architectures robust to multiple simultaneous errors," *Microelectronics Reliability*, vol. 55, no. 12, pp. 2845-2857, Dec. 2015.
- [12] D. Lipetz and E. Schwarz, "Self checking in current floating-point units," in *Proc. 20th IEEE Symp. Computer Arithmetic*, pp. 73-76, Tubingen, Germany, 25-27 Jul. 2011.
- [13] M. Maniatakos, Y. Makris, P. Kudva, and B. Fleischer, "Exponent monitoring for low-cost concurrent error detection in FPU control logic," in *Proc. 29th IEEE VLSI Test Symp.*, pp. 235-240, Dana Point, CA, USA, 1-5 May 2011.
- [14] A. Gupta, et al., "Low power probabilistic floating point multiplier design," in *Proc. IEEE Computer Society Annual Symp. VLSI, ISVLSI'11*, pp. 182-187, Chennai, India, 4-6 Jul. 2011.
- [15] H. Zhang, W. Zhang, and J. Lach, "A low-power accuracy-configurable floating point multiplier," in *Proc. 32nd IEEE Intl. Conf. Computer Design, ICCD'14*, pp. 48-54, Seoul, South Korea, 19-22 Oct. 2014.
- [16] S. Hashemi, R. Iris Bahar, and S. Reda, "DRUM: a dynamic range unbiased multiplier for approximate applications," in *Proc. IEEE Int. Conf. on Comp. Aided Design, ICCAD'15*, pp. 418-425, Austin, TX, USA, 2-6 Nov. 2015.
- [17] S. Narayanamoorthy, H. Asghari Moghaddam, Z. Liu, T. Park, and N. Sung Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," *IEEE Trans. on VLSI*, vol. 23, no. 6, pp. 1180-1184, Jun. 2015.
- [18] V. Camus, J. Schlachter, C. Enz, M. Gautschi, and F. K. Gurkaynak, "Approximate 32-bit floating-point unit design with 53% power-area product reduction," in *Proc. 42nd European Solid-State Circuits Conf., ESSCIRC'16*, pp. 465-468, Lausanne, Switzerland, 12-15 Sept. 2016.
- [19] M. Imani, D. Peroni, and T. Rosing, "CFPU: configurable floating point multiplier for energy-efficient computing," in *Proc. 54th ACM/EDAC/IEEE Design Automation Conf., DAC'17*, 6 pp., Austin, TX, USA, 18-22 Jun. 2017.
- [20] P. Yin, C. Wang, W. Liu, and F. Lombardi, "Design and performance evaluation of approximate floating-point multipliers," in *Proc. IEEE Computer Society Annual Symp. VLSI, ISVLSI'16*, pp. 296-301, Pittsburgh, PA, USA, 11-13 Jul. 2016.
- [21] A. Sunny, B. K. Mathew, and P. B. Dhanusha, "Area efficient high speed approximate multiplier with carry predictor," *Procedia Technology*, vol. 24, pp. 1170-1177, 2016.
- [22] M. Fathi and H. Nikmehr, "Improving accuracy, area and speed of approximate floating point multiplication using carry prediction," *J. of Information Systems and Telecommunication*, vol. 5, no. 2, pp. 120-127, Apr./Jun. 2017.
- [23] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *Proc. Design, Automation & Test in Europe Conf., DATE'14*, 4 pp., Dresden, Germany, 24-28 Mar. 2014.
- [24] P. Yin, C. Wang, W. Liu, E. E. Swartzlander Jr., and F. Lombardi, "Designs of approximate floating-point multipliers with variable accuracy for error-tolerant applications," *J. of Signal Processing Systems*, vol. 90, no. 4, pp. 641-654, Apr. 2018.

و مقایسه که نام دیگر آن DMR^1 است، برای تشخیص خطا به بیش از ۱۰۰٪ سربار مساحت و توان مصرفی احتیاج دارد که مناسب بودن معماری‌های پیشنهادی را نشان می‌دهد. همچنین معماری پیشنهادی دوم در این مقاله که در مد تحمل‌پذیر اشکال دارای قابلیت تصحیح خطا است و با توجه به اندازه مانیتیس‌های مشخص شده در جدول ۳ سربارهایی را تحمیل می‌کند، در مقایسه با روش TMR که حداقل به ۲۰۰٪ سربار مساحت و توان مصرفی احتیاج دارد به سربار بسیار کمتری نیاز دارد. علاوه بر این معماری پیشنهادی نسبت به روش ارائه شده در [۲۷] که تنها در یک مد کاری عمل می‌کند و قادر به تولید نتایج با دقت کامل نیست، در وضعیت مناسبی قرار دارد.

از آنجایی که طرح‌های تحمل‌پذیر اشکال پیشنهادی برای واحدهای ممیز- شناور در این پژوهش بر اساس کاهش دقت خروجی محاسبات است بنابراین برخی خطاهای محاسباتی در خروجی تولید خواهد شد که در بسیاری از برنامه‌های کاربردی به ویژه برنامه‌هایی که با حواس انسان سروکار دارند، قابل قبول است. تخمین حداکثر خطای نسبی برای یک خروجی ممیز- شناور که در جدول ۴ نیز انجام شده ساده است و طبق رابطه زیر به دست می‌آید

$$\text{Maximum Relative Error} = 2^{-l} \quad (2)$$

در (۲)، l عرض بیت مانیتیس مورد استفاده در معماری‌های مختلف است. بنابراین در طرح‌های با مانیتیس کوچک‌تر که سربارهای کمتری تحمیل می‌شود مقدار خطای نسبی بیشتر خواهد بود.

۶- نتیجه‌گیری

در این مقاله، دو معماری برای ضرب‌کننده ممیز- شناور با قابلیت کار در دو مد کاری عادی و تحمل‌پذیر اشکال پیشنهاد گردید. اگر محیطی که در آن از ضرب‌کننده استفاده می‌شود محیطی پرنویز باشد، استفاده از مد کاری تحمل‌پذیر اشکال بسیار مفید خواهد بود. اما خروجی عملیات با توجه به هزینه‌های مورد نیاز می‌تواند دارای یکی از قابلیت‌های تشخیص یا تصحیح خطا باشد. همچنین نشان داده شد که با کاهش دقت محاسبات در محدوده قابل قبول برای بسیاری از برنامه‌های کاربردی، می‌توان به قابلیت تشخیص یا تصحیح خطا دست یافت. معماری‌های پیشنهادی پس از حذف بیت‌های کم‌ارزش‌تر عملوندهای ورودی، با استفاده از تجزیه و طراحی ماجولار ضرب‌کننده ۲۴بیتی اصلی درون طرح ضرب‌کننده ممیز- شناور به منظور انجام محاسبات افزونه، به قابلیت‌های اشاره شده می‌رسند. نتایج پیاده‌سازی نشان می‌دهد که با سربارهای معقول می‌توان به ساختارهایی رسید که دارای درجه بالایی از قابلیت اطمینان در برابر خطاهای منفرد از نوع تشخیص یا تصحیح باشند.

مراجع

- [1] C. H. Yu, K. Chung, D. Kim, and L. S. Kim, "An energy-efficient mobile vertex processor with multithread expanded VLIW architecture and vertex caches," *IEEE J. of Solid-State Circuits*, vol. 42, no. 10, pp. 2257-2269, Oct. 2007.
- [2] B. Nicolescu, N. Lgnat, Y. Savaria, and G. Nicolescu, "Analysis of real-time systems sensitivity to transient faults using MicroC kernel," *IEEE Trans. Nuclear Science*, vol. 53, no. 4, pp. 1902-1909, Aug. 2006.
- [3] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proc. 49th Annual Design*

مجتبی ولی‌نجاج تحصیلات خود را در مقاطع کارشناسی، کارشناسی ارشد و دکترای مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۷۹، ۱۳۸۱ و ۱۳۸۹ هر سه در دانشگاه تهران به پایان رسانده است. نام‌برده در سال‌های ۱۳۹۰، ۱۳۹۱ و ۱۳۹۶ پروژه‌های مشترکی را با گروه‌های تحقیقاتی در دانشگاه تورکو و دانشگاه صنعتی تاملپره در فنلاند به انجام رسانده است. ایشان از سال ۱۳۸۹ تاکنون به‌عنوان عضو هیأت علمی در دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی نوشیروانی بابل مشغول به فعالیت است. زمینه‌های پژوهشی مورد علاقه ایشان شامل موضوعاتی مانند سیستم‌های قابل اطمینان، حساب کامپیوتری، شبکه بر تراشه، منطق برگشت‌پذیر و سیستم‌های چندپردازنده‌ای است.

- [25] P. J. Eibl, A. D. Cook, and D. J. Sorin, "Reduced precision checking for a floating point adder," in *Proc. 24th IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems*, pp. 145-152, Chicago, IL, USA, 7-9 Oct. 2009.
- [26] K. Seetharam, L. C. T. Keh, R. Nathan, and D. J. Sorin, "Applying reduced precision arithmetic to detect errors in floating point multiplication," in *Proc. IEEE Pacific Rim Intl. Symp. Dependable Computing, PRDC'13*, pp. 232-235, Vancouver, BC, Canada, 2-4 Dec. 2013.
- [27] M. Mohajer and M. Valinataj, "A novel reduced-precision fault-tolerant floating-point multiplier," *Int. J. of Modern Education and Computer Science*, vol. 9, no. 6, pp. 17-24, Jun. 2017.
- [28] B. Parhami, *Computer Arithmetic, Algorithms and Hardware Designs*, 2nd Ed., Oxford University Press, 2009.

مریم مهاجر تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۹۲ و ۱۳۹۵ در دانشگاه صنعتی نوشیروانی بابل به پایان رسانده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از حساب کامپیوتری، محاسبات قابل اطمینان و سیستم‌های تحمل‌پذیر اشکال.

Archive of SID