Computational Intelligence in Electrical Engineering
Vol. 16, No. 3, 2025
pp. 1-18
Research Paper

Low-Power Approximate Multiplier Design Based on Incremental and Decremental Approximate Adders for Efficient Implementation of Artificial Neural Networks

Sahand Divsalar[1], Mahla Salehi Sheikhali Kelayeh [2], Shaghayegh Vahdat[*3]
[1] School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[2] School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[3] School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

Abstract:
In this paper, through simplification and approximation of the truth table of exact adders with two, three, and four inputs, two methods for approximate implementation of each adder are proposed. In one structure, the average error of the adder is positive, while in the other, it is negative. These adders are respectively named incremental and decremental structures, as they tend to produce outputs that are, on average, larger or smaller than those of an exact adder. These adders are cleverly integrated into the structure of an approximate multiplier in such a way that they cancel out each other's errors, thereby minimizing the accuracy loss in the final multiplication result. The findings show that the proposed multiplier can improve the delay and power consumption of the multiplication operation by up to 51% and 83%, respectively, compared to exact multipliers. The performance of the proposed multiplier is also examined in common applications such as image processing and artificial neural networks, and the results indicate its satisfactory effectiveness in these applications. For instance, when the proposed multiplier is used in the implementation of the trained VGG-11 model for the CIFAR-10 dataset, the energy consumption of multiplication operations can be reduced by 63%, while the network's accuracy experiences only a slight 0.77% decrease compared to the exact case.

Keywords: Incremental and Decremental Approximate Adder, Approximate Multiplier, Low-Power, Image Processing, Artificial Neural Networks.

This is an open access article under the CC BY-NC-ND/4.0/ License (https://creativecommons.org/licenses/by-nc-nd/4.0/).
https://doi.org/10.22108/isee.2025.145437.1740

مقاله پژوهشی

طراحی ضرب‌کننده تقریبی کم‌توان مبتنی بر جمع‌کننده‌های تقریبی افزایشی و کاهشی برای اجرای کارای شبکه‌های عصبی مصنوعی

سهند دیوسالار[۱]، مهلا صالحی شیخعلی کلایه [۲]، شقایق وحدت[۳*]

۱- کارشناسی ارشد، گروه الکترونیک، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران
s.divsalar96@ut.ac.ir

۲- کارشناسی، گروه سیستم‌های دیجیتال، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران
mahla.salehi@ut.ac.ir

۳- استادیار گروه الکترونیک، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران
vahdat_s@ut.ac.ir

چکیده: در این مقاله، با ساده‌سازی و اعمال تقریب در جدول درستی جمع‌کننده‌های دقیق با دو، سه، و چهار ورودی، دو روش برای اجرای تقریبی هر جمع‌کننده ارائه شده است که در یک ساختار، میانگین خطای جمع‌کننده مثبت و در ساختار دیگر، منفی است. این جمع‌کننده‌ها به ترتیب ساختارهای افزایشی و کاهشی نامیده شده‌اند که به طور میانگین، خروجی بزرگ‌تر و کوچک‌تری نسبت به جمع‌کنندهٔ دقیق تولید می‌کنند. این جمع‌کننده‌ها به گونه‌ای هوشمندانه در ساختار یک ضرب‌کنندهٔ تقریبی در کنار هم قرار می‌گیرند تا خطای تولیدی توسط یکدیگر را خنثی کنند و افت دقت حاصل ضرب نهایی را کاهش دهند. نتایج نشان می‌دهد ضرب‌کنندهٔ پیشنهادی می‌تواند تأخیر و توان مصرفی عملیات ضرب را تا ۵۱ و ۸۳ درصد نسبت به ضرب‌کنندهٔ دقیق بهبود بخشد. عملکرد ضرب‌کنندهٔ پیشنهادی در کاربردهای متداول پردازش تصویر و شبکه‌های عصبی مصنوعی نیز بررسی می‌شود و نتایج حاکی از عملکرد قابل قبول ضرب‌کنندهٔ پیشنهادی در کاربردهای یادشده است. برای مثال، با استفاده از ضرب‌کنندهٔ پیشنهادی در اجرای مدل VGG-11 آموزش‌یافته برای مجموعه دادهٔ CIFAR-10، می‌توان انرژی مصرفی عملیات ضرب را ٦۳ درصد کاهش داد؛ در حالی که صحت شبکه نسبت به حالت دقیق فقط ۰/۷۷ درصد افت پیدا می‌کند.

واژه‌های کلیدی: جمع‌کنندهٔ تقریبی افزایشی و کاهشی، ضرب‌کنندهٔ تقریبی، کم‌توان، پردازش تصویر، شبکه‌های عصبی مصنوعی.

۱- مقدمه

امروزه، بسیاری از مردم از ادوات قابل حمل مانند گوشی‌های هوشمند و رایانه‌های شخصی استفاده می‌کنند که دارای باتری‌هایی با شارژ محدود هستند. به منظور مدیریت شارژ محدود باتری این دستگاه‌ها، نیاز است راهکارهایی برای کاهش توان مصرفی ارائه شوند که از جملهٔ این راهکارها می‌توان به تنظیم پویای سطح ولتاژ و فرکانس[۱] [۱]، قطع منبع ولتاژ [۲] و قطع سیگنال ساعت[۲] [۳] اشاره کرد.

برخی از برنامه‌های پرکاربرد که بر روی دستگاه‌های قابل حمل اجرا می‌شوند، تحمل‌پذیر خطا هستند و لزومی برای انجام محاسبات دقیق در آنها وجود ندارد. از جملهٔ این کاربردها می‌توان به پردازش تصاویر و شبکه‌های عصبی مصنوعی اشاره کرد که برای محاسبهٔ خروجی آنها، نیاز به انجام تعداد زیادی از عملیات محاسباتی پایه مانند جمع و ضرب وجود دارد. برای مثال، اگر محاسبات انجام‌شده بر

[۱] تاریخ ارسال مقاله: ۱٤۰٤/۰۳/۰٦
تاریخ پذیرش مقاله: ۱٤۰٤/۰۸/۰٦
نام نویسندهٔ مسئول: شقایق وحدت
نشانی نویسندهٔ مسئول: ایران، تهران، دانشگاه تهران، دانشکده مهندسی برق و کامپیوتر، گروه الکترونیک

روی پیکسل‌های یک تصویر به صورت دقیق انجام نشوند و مقدار خروجی هر پیکسل تفاوتی اندک با مقدار دقیق داشته باشد، چشم انسان توانایی تشخیص این اختلاف را ندارد؛ بنابراین، کیفیت تصویر تولیدی افت پیدا نمی‌کند. از این رو، در کاربردهای تحمل‌پذیر خطا می‌توان از روش محاسبات تقریبی به منظور کاهش انرژی مصرفی و افزایش سرعت محاسبات استفاده کرد؛ بدون آنکه تأثیر چندانی روی کیفیت خروجی داشته باشد [٤].

همان‌طور که پیش‌تر اشاره شد، عملیات ضرب و جمع جزو عملیات پرکاربرد در پردازش تصاویر و شبکه‌های عصبی مصنوعی هستند که از بین آنها، عمل ضرب انرژی مصرفی و تأخیر بیشتری دارد. به همین دلیل، تا کنون پژوهش‌هایی متنوع در حوزهٔ طراحی ضرب‌کننده‌های تقریبی انجام شده‌اند که در ادامه، برخی از روش‌های متداول طراحی این حوزه بررسی خواهند شد.

به طور کلی، مراحل صورت‌گرفته در عمل ضرب را می‌توان به سه بخش تقسیم کرد:

۱) تولید حاصل‌ضرب‌های جزئی[۳]: در این مرحله، یک یا چند بیت از عملوند[٤] دوم در هر یک از بیت‌های عملوند اول ضرب می‌شود که به آن، حاصل‌ضرب جزئی گفته می‌شود.

۲) کاهش حاصل‌ضرب‌های جزئی[۵]: در این مرحله، حاصل‌ضرب‌های جزئی تولیدشده با هم جمع می‌شوند و این عملیات آن‌قدر ادامه پیدا می‌کند تا در نهایت دو سطر از اعداد باقی بماند.

۳) انباشت حاصل‌ضرب‌های جزئی[٦]: در این مرحله، حاصل جمع دو سطر باقی‌مانده توسط یک جمع‌کننده سریع محاسبه می‌شود تا خروجی نهایی ضرب‌کننده تولید شود.

با اعمال تقریب در هر یک از سه مرحلهٔ بالا می‌توان یک ضرب‌کنندهٔ تقریبی طراحی کرد. برای مثال، در **[۵]**، یک ضرب‌کنندهٔ بوث[۷] تقریبی ارائه شده است که در آن، مدار تولیدکنندهٔ حاصل‌ضرب‌های جزئی به صورت تقریبی طراحی شده است. استفاده از جمع‌کننده‌های تقریبی در مرحلهٔ دوم نیز می‌تواند تأثیری به‌سزا در کاهش تأخیر و توان مصرفی ضرب‌کننده داشته باشد. یکی از راهکارهای ارائه‌شده طراحی واحدهای جمع‌کنندهٔ پایه مانند تمام‌جمع‌کننده[۸] و نیم‌جمع‌کننده[۹] با تعداد ترانزیستورهای کمتر است که می‌تواند منجر به کاهش مساحت، توان مصرفی و تأخیر آنها شود [٦]؛ این در حالی است که خروجی تولیدشده تقریبی خواهد بود. استفاده از فشرده‌سازهای[۱۰] ٤:۲ تقریبی نیز یک روش متداول برای اجرای ضرب‌کننده‌های تقریبی است که در عمدهٔ این پژوهش‌ها، با اعمال تقریب به جدول درستی[۱۱] فشرده‌ساز دقیق، ساختار تقریبی با پیچیدگی کمتر استخراج می‌شود [۷]، [۸]، [۹]، [۱۰].

برای جلوگیری از افت دقت شدید در بیشتر این ضرب‌کننده‌ها، از واحدهای جمع‌کنندهٔ تقریبی برای محاسبهٔ حاصل جمع حاصل‌ضرب‌های جزئی ستون‌های کم‌ارزش‌تر استفاده می‌شود و حاصل جمع ستون‌های پرارزش‌تر با استفاده از جمع‌کننده‌های دقیق محاسبه می‌شود [۱۱]. برای جبران افت دقت ناشی از جمع‌کننده‌های تقریبی نیز می‌توان از واحدهای جبرانگر خطا[۱۲] استفاده کرد [۱۲].

یکی دیگر از راهکارهای متداول برای کاهش توان مصرفی ضرب‌کننده‌ها بریدن[۱۳] بیت‌های کم‌ارزش ورودی‌ها و انجام عملیات محاسباتی روی بیت‌های باقی‌مانده است. در این روش، هر ورودی به چندین بخش تقسیم می‌شود که عملیات محاسباتی روی برخی از این بخش‌ها انجام می‌شود. بخش‌بندی ورودی‌ها را می‌توان به صورت ایستا[۱٤] و یا پویا[۱۵] انجام داد. در روش ایستا، ارزش مکانی هر بخش از ورودی از قبل مشخص است [۱۳]، [۱٤]، [۱۵]؛ اما در روش پویا، بخش‌بندی بر اساس مکان قرارگیری بیت یک پرارزش ورودی انجام می‌شود و بسته به مقدار ورودی، می‌تواند ارزش مکانی مختلفی داشته باشد [۱٦]، [۱۷]. پس از انتخاب بخش مدنظر، عملیات محاسباتی مانند ضرب، جمع و انتقال[۱٦] روی آنها انجام می‌شود تا خروجی نهایی ضرب‌کننده تعیین شود. با توجه به اینکه در ساختارهای ایستا، ارزش مکانی هر بخش از قبل مشخص است، اجرای سخت‌افزاری واحد انتقال مورد نیاز آن ساده‌تر از ساختارهای پویاست؛ اما با استفاده از ساختارهای پویا عمدتاً می‌توان به دقت بیشتری دست یافت. به منظور برقراری مصالحه بین دقت قابل دست‌یابی و سادگی سخت‌افزار مورد نیاز، می‌توان از ترکیب ساختارهای ایستا و پویا استفاده کرد [۱۸]، [۱۹].

در این مقاله، دو روش برای اجرای واحدهای جمع‌کنندهٔ تقریبی با دو، سه، و چهار ورودی ارائه می‌شوند که این

واحدها فقط یک خروجی دارند. در یکی از این روش‌ها، به طور میانگین، خروجی‌های تولیدشده کوچک‌تر از مقدار دقیق و در روش دیگر، بزرگ‌تر هستند. در ادامه، این جمع‌کننده‌ها به گونه‌ای در ساختار یک ضرب‌کننده قرار می‌گیرند که خطاهای یکدیگر را تا حدی خوب جبران کنند و به یک ضرب‌کننده با دقت مناسب و مشخصات سخت‌افزاری مطلوب منجر شوند.

در بخش دوم مقاله، روش‌های پیشنهادی برای اجرای جمع‌کننده‌های تقریبی و چگونگی قرارگیری مناسب آنها در ساختار یک ضرب‌کنندهٔ هشت‌بیتی توضیح داده می‌شود. در بخش سوم، تحلیلی از خطای ضرب‌کنندهٔ پیشنهادی و همچنین مشخصات سخت‌افزاری آن ارائه و با برخی از ضرب‌کننده‌های تقریبی پیشین مقایسه می‌شود. در بخش چهارم، عملکرد ضرب‌کننده‌های تقریبی پیشنهادی در کاربردهای پردازش تصویر و شبکه‌های عصبی مصنوعی بررسی و با کارهای پیشین مقایسه می‌شود. در بخش پنجم، نتیجه‌گیری مقاله ارائه می‌شود.

۲- ساختار پیشنهادی

در این بخش، ابتدا دو روش برای اجرای جمع‌کننده‌های تقریبی پیشنهاد می‌شوند که در آنها فقط از دروازهٔ OR[۱۷] استفاده می‌شود. سپس، احتمال بروز خطا و میانگین خطای هر یک از این جمع‌کننده‌ها برای تولید حاصل جمع حاصل‌ضرب‌های جزئی محاسبه می‌شود. در ادامه، چگونگی قرارگیری این جمع‌کننده‌ها در ساختار یک ضرب‌کنندهٔ هشت‌بیتی بررسی می‌شود تا در نهایت، بتوان به یک ضرب‌کننده با دقت قابل قبول و مشخصات سخت‌افزاری مناسب دست یافت.

۲-۱- جمع‌کننده‌های تقریبی پیشنهادی

همان‌طور که در بخش ۱ بیان شد، جمع‌کننده‌ها واحدهای اصلی استفاده‌شده برای اجرای ضرب‌کننده‌ها هستند. به همین دلیل، طراحی کم‌توان و پرسرعت این واحدها می‌تواند تأثیری به‌سزا در بهبود مشخصات سخت‌افزاری ضرب‌کننده داشته باشد. از این رو، در این بخش، دو روش برای اجرای کارای جمع‌کننده‌های تقریبی با دو، سه، و چهار ورودی ارائه می‌شود و میزان خطای هر یک از آنها برای تولید حاصل جمع حاصل‌ضرب‌های جزئی محاسبه می‌شود. در هر دو ساختار جمع‌کننده، فقط از دروازهٔ OR برای تولید خروجی استفاده می‌شود. در ساختار اول، فقط بیت S (بیت کم ارزش خروجی) و در ساختار دوم، فقط بیت C (بیت پرارزش خروجی) تولید می‌شود و خروجی‌های دیگر در این ساختارها صفر در نظر گرفته می‌شوند و نیازی به تولید آها نخواهد بود.

با فرض آنکه احتمال صفر یا یک بودن هر بیت از ورودی‌های ضرب‌کننده برابر ۰/۵ باشد و دو ورودی نسبت به هم مستقل باشند، احتمال یک شدن هر بیت از حاصل‌ضرب‌های جزئی برابر ۰/۲۵ خواهد بود؛ زیرا هر بیت از حاصل‌ضرب جزئی با انجام عمل AND روی بیت‌های متناظر در دو ورودی ضرب‌کننده تولید می‌شود و خروجی یک دروازه AND در صورتی یک است که هر دو ورودی آن یک باشند. با فرض کاملاً تصادفی و مستقل بودن ورودی‌ها، این احتمال برابر (۰/۵×۰/۵=)۰/۲۵ خواهد بود. به همین ترتیب، احتمال صفر شدن حاصل‌ضرب جزئی برابر ۰/۷۵ می‌شود. گفتنی است، این حاصل‌ضرب‌های جزئی به عنوان ورودی به جمع‌کننده‌ها اعمال می‌شوند.

نیم‌جمع‌کنندهٔ تقریبی: یک نیم‌جمع‌کننده دارای دو ورودی و دو خروجی تک‌بیتی است. جدول درستی و خطای نیم‌جمع‌کننده‌های تقریبی پیشنهادی اول (HA_APX1) و دوم (HA_APX2) در جدول (۱) ارائه شده است. در این جدول، A و B ورودی‌ها و S و C خروجی‌های جمع‌کننده هستند و احتمال وقوع هر یک از سطرهای این جدول در ستون $Prob$ نمایش داده شده است. همان‌طور که پیش‌تر اشاره شد، در ساختار اول، فقط خروجی S با OR کردن ورودی‌ها تولید می‌شود و مقدار خروجی C صفر در نظر گرفته می‌شود. در ساختار دوم، فقط خروجی C با OR کردن ورودی‌ها تولید می‌شود و مقدار خروجی S صفر در نظر گرفته می‌شود.

بر اساس جدول (۱)، خروجی این جمع‌کننده‌ها در بیشتر حالات صحیح تولید می‌شود و فقط در یک حالت، خروجی با مقدار مورد انتظار تفاوت دارد. بنابراین، میانگین خطای

نیم‌جمع‌کنندهٔ اول برای تولید حاصل‌جمع حاصل‌ضرب‌های جزئی بر اساس رابطهٔ (۱) قابل محاسبه است:

$$Err_{HA_APX1} = \sum_{i=1}^{4} Err_i \times Prob_i = \frac{-1}{16} \quad (۱)$$

از آنجا که میانگین خطای محاسبه‌شده منفی است، این جمع‌کننده را می‌توان یک جمع‌کنندهٔ کاهشی در نظر گرفت. به عبارت دیگر، خروجی این جمع‌کننده به طور میانگین کوچک‌تر از مقدار دقیق مورد انتظار است.

میانگین خطای نیم‌جمع‌کنندهٔ دوم با استفاده از رابطهٔ (۲) قابل محاسبه است:

$$Err_{HA_APX2} = \sum_{i=1}^{4} Err_i \times Prob_i = \frac{+3+3}{16} = \frac{+6}{16} \quad (۲)$$

از آنجا که میانگین خطای این جمع‌کننده یک مقدار مثبت است، این جمع‌کننده را می‌توان یک جمع‌کنندهٔ افزایشی در نظر گرفت که به طور میانگین، منجر به تولید خروجی بزرگ‌تر از مقدار واقعی خواهد شد.

جدول (۱): جدول درستی نیم‌جمع‌کننده‌های تقریبی پیشنهادی به همراه خطای آنها

ورودی	دقیق	HA_APX1	HA_APX2	$Prob$
(A,B)	(C,S)	(Err,C,S)	(Err,C,S)	
(۰,۰)	(۰,۰)	(۰,۰,۰)	(۰,۰,۰)	$\frac{9}{16}$
(۰,۱)	(۰,۱)	(۰,۰,۱)	(+۱,۱,۰)	$\frac{3}{16}$
(۱,۰)	(۰,۱)	(۰,۰,۱)	(+۱,۱,۰)	$\frac{3}{16}$
(۱,۱)	(۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	$\frac{1}{16}$

تمام‌جمع‌کنندهٔ تقریبی: یک تمام‌جمع‌کننده دارای سه ورودی و دو خروجی تک‌بیتی است. جدول درستی و خطای تمام‌جمع‌کننده‌های تقریبی پیشنهادی در جدول (۲) ارائه شده است. در ساختار اول، فقط خروجی S و در ساختار دوم، فقط خروجی C با OR کردن ورودی‌ها تولید می‌شود. میانگین خطای هر کدام از این تمام‌جمع‌کننده‌ها در روابط (۳) و (٤) محاسبه شده است. این روابط نشان می‌دهند میانگین خطای جمع‌کنندهٔ اول (دوم) منفی (مثبت) است و بنابراین، این جمع‌کننده را می‌توان به عنوان یک جمع‌کنندهٔ کاهشی (افزایشی) در نظر گرفت:

$$Err_{FA_APX1} = \sum_{i=1}^{8} Err_i \times Prob_i = \frac{-3\times 3-2}{64} = \frac{-11}{64} \quad (۳)$$

$$Err_{FA_APX2} = \sum_{i=1}^{8} Err_i \times Prob_i = \frac{+9\times 3-1}{64} = \frac{+26}{64} \quad (٤)$$

فشرده‌ساز ٤:۲ تقریبی: یک فشرده‌ساز ٤:۲ دقیق، پنج ورودی و سه خروجی تک‌بیتی دارد. هر پنج ورودی تک‌بیتی و یکی از خروجی‌های تولیدشده (که در این مقاله با S نمایش داده می‌شود) دارای ارزش وزنی یکسان هستند؛ در حالی که دو خروجی دیگر (بیت‌های نقلی که در این مقاله، با C_{out} و C نمایش داده می‌شوند)، ارزش وزنی دوبرابر دارند. یکی از ورودی‌های فشرده‌ساز، بیت نقلی (C_{out}) ستون قبل است و بیت نقلی تولیدی توسط این فشرده‌ساز نیز به ستون بعد اعمال می‌شود. به عبارت دیگر، هر یک از فشرده‌سازها، یک بیت نقلی ورودی و خروجی دارند که بین فشرده‌سازهای ستون‌های مختلف منتقل می‌شود. به همین دلیل است که این واحدها به عنوان فشرده‌سازهای ٤:۲ نام‌گذاری شده‌اند.

جدول (۲): جدول درستی تمام‌جمع‌کننده‌های تقریبی پیشنهادی به همراه خطای آنها

ورودی	دقیق	FA_APX1	FA_APX2	$Prob$
(A,B,Cin)	(C,S)	(Err,C,S)	(Err,C,S)	
(۰,۰,۰)	(۰,۰)	(۰,۰,۰)	(۰,۰,۰)	$\frac{27}{64}$
(۰,۰,۱)	(۰,۱)	(۰,۰,۱)	(+۱,۱,۰)	$\frac{9}{64}$
(۰,۱,۰)	(۰,۱)	(۰,۰,۱)	(+۱,۱,۰)	$\frac{9}{64}$
(۰,۱,۱)	(۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	$\frac{3}{64}$
(۱,۰,۰)	(۰,۱)	(۰,۰,۱)	(+۱,۱,۰)	$\frac{9}{64}$
(۱,۰,۱)	(۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	$\frac{3}{64}$
(۱,۱,۰)	(۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	$\frac{3}{64}$
(۱,۱,۱)	(۱,۱)	(−۲,۰,۱)	(−۱,۱,۰)	$\frac{1}{64}$

در فشرده‌سازهای ٤:۲ تقریبی، از تولید و انتشار بیت‌های نقلی (C_{out}) بین فشرده‌سازها صرف‌نظر می‌شود. به همین دلیل، یک فشرده‌ساز ٤:۲ تقریبی دارای چهار ورودی و دو خروجی تک‌بیتی است. جدول درستی و خطای فشرده‌سازهای ٤:۲ تقریبی پیشنهادی در جدول (۳) ارائه شده است که در ساختارهای اول و دوم، به ترتیب، فقط خروجی‌های S و C با OR کردن ورودی‌ها تولید شده‌اند. اگر از فشرده‌سازها برای تولید حاصل جمع حاصل‌ضرب‌های جزئی استفاده شود، احتمال آنکه از میان چهار ورودی فشرده‌ساز، k ورودی مقدار صفر داشته باشند و بقیهٔ ورودی‌ها مقدار یک داشته باشند، برابر $3^k/4^4$ است. این امر نشان می‌دهد احتمال داشتن تعداد زیادی ورودی یک در یک فشرده‌ساز ٤:۲ کم است و بنابراین، انتظار می‌رود تقریب‌های پیشنهادی منجر به خطایی بزرگ در خروجی فشرده‌ساز نشوند؛ این در حالی است که تقریب پیشنهادی می‌تواند مشخصات سخت‌افزاری را نسبت به حالت دقیق به طرزی چشمگیر بهبود بخشد که در بخش بعد بررسی خواهد شد.

میانگین خطای هر یک از فشرده‌سازهای یادشده در روابط (۵) و (٦) محاسبه شده است:

$$Err_{Com_APX1} = \sum_{i=1}^{16} Err_i \times Prob_i = \frac{-9\times 6 - 6\times 4 - 3}{256} = \frac{-81}{256} \quad (۵)$$

$$Err_{Com_APX2} = \sum_{i=1}^{16} Err_i \times Prob_i = \frac{+27\times 4 - 3\times 4 - 2}{64} = \frac{+94}{256} \quad (٦)$$

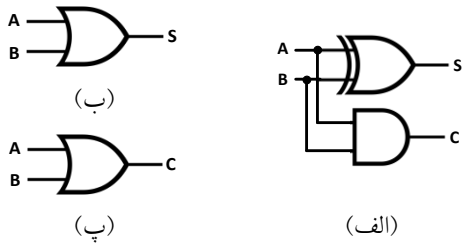
همانند سایر ساختارهای جمع‌کنندهٔ پیشنهادی قبل، فشرده‌ساز ساختار اول دارای میانگین خطای منفی و ساختار دوم دارای میانگین خطای مثبت است. بنابراین، ساختار اول را می‌توان یک جمع‌کنندهٔ کاهشی و ساختار دوم را یک جمع‌کنندهٔ افزایشی در نظر گرفت.

جدول (۳): جدول درستی فشرده‌سازهای ٤:۲ تقریبی پیشنهادی به همراه خطای آنها

$Prob$	Com_APX2	Com_APX1	دقیق	ورودی
	(Err, C, S)	(Err, C, S)	(C_{out}, C, S)	(A, B, D, E)
$\frac{81}{256}$	(۰,۰,۰)	(۰,۰,۰)	(۰,۰,۰)	(۰,۰,۰,۰)
$\frac{27}{256}$	(+۱,۱,۰)	(۰,۰,۱)	(۰,۰,۱)	(۰,۰,۰,۱)
$\frac{27}{256}$	(+۱,۱,۰)	(۰,۰,۱)	(۰,۰,۱)	(۰,۰,۱,۰)
$\frac{9}{256}$	(۰,۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	(۰,۰,۱,۱)
$\frac{27}{256}$	(+۱,۱,۰)	(۰,۰,۱)	(۰,۰,۱)	(۰,۱,۰,۰)
$\frac{9}{256}$	(۰,۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	(۰,۱,۰,۱)
$\frac{9}{256}$	(۰,۱,۰)	(−۱,۰,۱)	(۱,۰,۰)	(۰,۱,۱,۰)
$\frac{3}{256}$	(−۱,۱,۰)	(−۲,۰,۱)	(۱,۰,۱)	(۰,۱,۱,۱)
$\frac{27}{256}$	(+۱,۱,۰)	(۰,۰,۱)	(۰,۰,۱)	(۱,۰,۰,۰)
$\frac{9}{256}$	(۰,۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	(۱,۰,۰,۱)
$\frac{9}{256}$	(۰,۱,۰)	(−۱,۰,۱)	(۰,۱,۰)	(۱,۰,۱,۰)
$\frac{3}{256}$	(−۱,۱,۰)	(−۲,۰,۱)	(۱,۰,۱)	(۱,۰,۱,۱)
$\frac{9}{256}$	(۰,۱,۰)	(−۱,۰,۱)	(۱,۰,۰)	(۱,۱,۰,۰)
$\frac{3}{256}$	(−۱,۱,۰)	(−۲,۰,۱)	(۱,۰,۱)	(۱,۱,۰,۱)
$\frac{3}{256}$	(−۱,۱,۰)	(−۲,۰,۱)	(۱,۰,۱)	(۱,۱,۱,۰)
$\frac{1}{256}$	(−۲,۱,۰)	(−۳,۰,۱)	(۱,۱,۰)	(۱,۱,۱,۱)

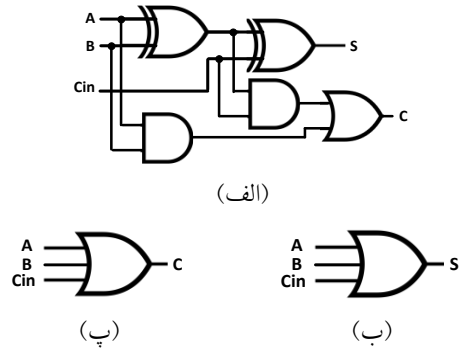
۲-۲- اجرای سخت‌افزاری واحدهای جمع‌کننده

به منظور دست‌یابی به یک درک بهتر از میزان ساده‌سازی صورت‌گرفته در اجرای واحدهای جمع‌کننده، اجرای دقیق هر ساختار در کنار اجراهای تقریبی پیشنهادی در شکل‌های (۱) تا (۳) نمایش داده شده است. همان‌طور که در شکل (۱) مشاهده می‌شود، در ساختار یک نیم‌جمع‌کنندهٔ دقیق، به یک دروازهٔ XOR و یک دروازهٔ AND دوورودی با ۱۲ و ٦ ترانزیستور نیاز خواهد بود که در ساختار تقریبی، به یک دروازهٔ OR با ٦ ترانزیستور ساده‌سازی می‌شود.



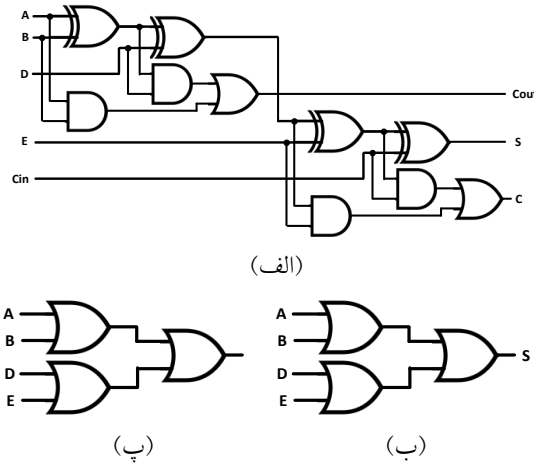
شکل (۱): اجرای نیم‌جمع‌کننده به صورت (الف) دقیق، (ب) تقریبی نوع اول (HA_APX1)، (پ) تقریبی نوع دوم (HA_APX2)

در یک تمام‌جمع‌کنندهٔ دقیق، به دو دروازهٔ XOR، دو دروازهٔ AND و یک دروازهٔ OR دوورودی نیاز خواهد بود که در ساختار تقریبی، با یک دروازهٔ OR سه‌ورودی اجرا می‌شود (به شکل (۲) مراجعه شود).



شکل (۲): اجرای تمام‌جمع‌کننده به صورت (لف) دقیق، (ب)تقریبی نوع اول (FA_APX1)، (پ) تقریبی نوع دوم (FA_APX2)

از آنجا که یک فشرده‌ساز ٤:۲ دقیق دارای پنج ورودی تک‌بیتی است، برای تولید خروجی‌های آن به دو تمام‌جمع‌کننده نیاز است که همانند بخش (الف) شکل (۳) به یکدیگر متصل شده باشند. در این ساختار، C_{in} و C_{out} نشان‌دهندهٔ بیت‌های نقلی ورودی و خروجی هستند که بین فشرده‌سازهای ستون‌های مختلف منتشر می‌شوند. یک فشرده‌ساز ٤:۲ دقیق متشکل از چهار دروازهٔ XOR، چهار دروازهٔ AND و دو دروازهٔ OR دوورودی است. در ساختار تقریبی پیشنهادی، از بیت‌های نقلی ورودی و خروجی صرف‌نظر و خروجی فشرده‌ساز با اعمال تابع OR به چهار ورودی تولید می‌شود. برای انجام عملیات OR روی چهار ورودی، می‌توان از سه دروازهٔ OR دوورودی استفاده کرد (همانند بخش‌های (ب) و (پ) شکل (۳). به منظور کاهش بیشتر تعداد ترانزیستورهای به‌کاررفته در ساختار تقریبی، می‌توان از دو دروازهٔ NOR و یک دروازهٔ NAND دوورودی استفاده کرد که عملکردی مشابه یک دروازهٔ OR چهارورودی خواهند داشت. مقایسهٔ ساختار پیشنهادی نسبت به اجرای یک فشرده‌ساز ٤:۲ دقیق، نشان‌دهندهٔ میزان چشمگیر ساده‌سازی صورت‌گرفته در اجرای سخت‌افزاری آن است که تعداد ترانزیستورهای به‌کاررفته را از ۸٤ به ۱۲ کاهش می‌دهد.



شکل (۳): اجرای فشرده‌ساز ٤:۲ به صورت (لف) دقیق، (ب)تقریبی نوع اول (Com_APX1)، (پ) تقریبی نوع دوم (Com_APX2)

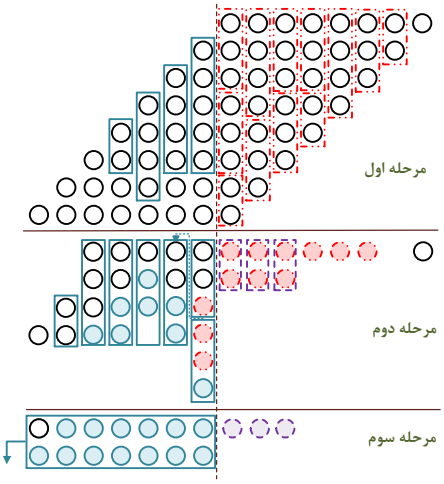
۳-۲- اجرای ضرب‌کنندهٔ تقریبی

در این بخش، چگونگی اجرای یک ضرب‌کنندهٔ هشت‌بیتی تقریبی با استفاده از جمع‌کننده‌های تقریبی پیشنهادشده در بخش قبل ارائه می‌شود. در اجرای یک ضرب‌کننده می‌توان فقط از یک نوع جمع‌کنندهٔ تقریبی یا ترکیبی از آنها استفاده کرد. همچنین، تعداد ستون‌هایی از حاصل‌ضرب‌های جزئی که به صورت تقریبی جمع می‌شوند، می‌تواند تأثیری به‌سزا روی دقت ضرب‌کننده داشته باشد. بدیهی است، با افزایش تعداد ستون‌های تقریبی، انتظار می‌رود دقت ضرب‌کننده کاهش یابد، اما مشخصات سخت‌افزاری آن بهبود یابد. بنابراین، برای طراحی ضرب‌کنندهٔ تقریبی نیاز است یک مصالحه بین دقت

ضرب‌کننده و مشخصات سخت‌افزاری آن در نظر گرفته شود و بر اساس کاربرد مدنظر، ساختار ضرب‌کننده با دقت مطلوب و مشخصات سخت‌افزاری بهتر انتخاب شود.

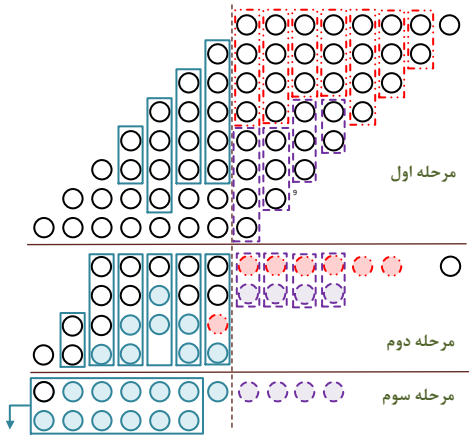
نمودار نقطه‌ای[18] یک ضرب‌کنندهٔ هشت‌بیتی پیشنهادی با هشت ستون تقریبی در شکل (٤) نمایش داده شده است. در این شکل، مستطیل‌های آبی‌رنگ نشان‌دهندهٔ جمع‌کننده‌های دقیق و مستطیل‌های خط‌چین بنفش و قرمز رنگ نشان‌دهندهٔ جمع‌کننده‌های تقریبی نوع اول و دوم هستند. دایره‌های مشکی‌رنگ نشان‌دهندهٔ حاصل‌ضرب‌های جزئی، دایره‌های آبی نشان‌دهندهٔ خروجی‌های جمع‌کننده‌های دقیق و دایره‌های بنفش و قرمز رنگ نشان‌دهندهٔ خروجی‌های جمع‌کننده‌های تقریبی نوع اول و دوم هستند. گفتنی است، جمع‌کننده‌های بنفش‌رنگ فقط خروجی S را تولید می‌کنند که با یک دایرهٔ بنفش‌رنگ در همان ستون نمایش داده شده است و جمع‌کننده‌های قرمزرنگ فقط خروجی C را تولید می‌کنند که با دایرهٔ قرمزرنگ در ستون بعدی نمایش داده شده است. گفتنی است، خروجی‌های هر مرحله در مرحلهٔ بعد توسط جمع‌کننده‌های تقریبی یا دقیق جمع می‌شوند.

در این ساختار، حاصل جمع حاصل‌ضرب‌های جزئی هشت ستون کم‌ارزش با استفاده از جمع‌کننده‌های تقریبی و سایر ستون‌ها، با استفاده از جمع‌کننده‌های دقیق محاسبه شده است. در مرحلهٔ اول کاهش حاصل‌ضرب‌های جزئی، فقط از جمع‌کننده‌های افزایشی نوع دوم استفاده شده است که بر اساس روابط (۲)، (٤)، و (٦)، دارای میانگین خطای مثبت هستند و به طور میانگین، خروجی‌های بزرگ‌تر از مقدار دقیق تولید می‌کنند. به منظور کاهش خطای ضرب‌کننده، در مرحلهٔ دوم، از جمع‌کننده‌های کاهشی نوع اول استفاده شده است که میانگین خطای منفی ایجاد می‌کنند و تا حدی می‌توانند خطای ناشی از جمع‌کننده‌های افزایشی نوع دوم را جبران کنند. همین ایده را می‌توان برای اجرای یک ضرب‌کننده با تعداد ستون‌های تقریبی متفاوت نیز استفاده کرد. برای مثال، در مرحلهٔ اول یک ضرب‌کنندهٔ هشت‌بیتی با ۱۱ ستون تقریبی (ضرب‌کنندهٔ M1_11) نیز فقط از جمع‌کننده‌های تقریبی نوع دوم و در مرحلهٔ دوم، از جمع‌کننده‌های تقریبی نوع اول استفاده می‌شود. حاصل جمع سایر ستون‌ها با استفاده از جمع‌کننده‌های دقیق محاسبه می‌شود.



شکل (٤): نمودار نقطه‌ای ضرب‌کنندهٔ هشت‌بیتی تقریبی نوع اول با هشت ستون تقریبی (M1_8)

از آنجا که در ساختار ضرب‌کنندهٔ نوع اول، عمدتاً از جمع‌کننده‌های افزایشی استفاده شده است، انتظار می‌رود خروجی تولیدشده توسط این ضرب‌کننده از مقدار دقیق بزرگ‌تر باشد. به منظور کاهش میزان خطا، ساختار دوم را پیشنهاد می‌کنیم که در مرحلهٔ اول آن، یکی از جمع‌کننده‌های هر ستون به صورت افزایشی و دیگری به صورت کاهشی اجرا شده باشد. همچنین، با توجه به آنکه اندازهٔ میانگین خطای جمع‌کننده‌های افزایشی بزرگ‌تر از جمع‌کننده‌های کاهشی است، در مرحلهٔ دوم نیز از جمع‌کننده‌های کاهشی استفاده می‌شود که بتوانند خطای بزرگ‌تر ناشی از جمع‌کننده‌های افزایشی را بهتر جبران کنند. برای مثال، نمودار نقطه‌ای ضرب‌کنندهٔ هشت‌بیتی تقریبی نوع دوم با هشت ستون تقریبی (ضرب‌کنندهٔ M2_8) در شکل (٥) رسم شده است.



شکل (٥): نمودار نقطه‌ای ضرب‌کنندهٔ هشت‌بیتی تقریبی نوع دوم با هشت ستون تقریبی (M2_8)

۳- نتایج شبیه‌سازی و بحث

در این بخش، نتایج شبیه‌سازی دو ساختار ضرب‌کنندهٔ تقریبی پیشنهادی از نظر دقت و مشخصات سخت‌افزاری ارائه و با برخی از ضرب‌کننده‌های تقریبی پیشین مقایسه می‌شود.

۳-۱- تحلیل دقت

به منظور بررسی دقت ضرب‌کنندهٔ تقریبی پیشنهادی، در ابتدا برخی از پارامترهای متداول خطا و نحوهٔ محاسبهٔ آنها توضیح داده می‌شوند. در تمامی روابط، O و O^* نشان‌دهندهٔ خروجی‌های دقیق و تقریبی ضرب‌کننده هستند و N نشان‌دهندهٔ تعداد کلیهٔ حالات اعمال ورودی‌ها به ضرب‌کنندهٔ n-بیتی است.

- خطا[۱۹]: نشان‌دهندهٔ اختلاف خروجی دقیق و تقریبی است که از طریق رابطهٔ (۷) محاسبه می‌شود:

$$Err = O^* - O \quad (۷)$$

- اندازهٔ (فاصلهٔ) خطا[۲۰]: نشان‌دهندهٔ قدرمطلق اختلاف خروجی دقیق و تقریبی است که از طریق رابطهٔ (۸) محاسبه می‌شود:

$$ED = |O^* - O| \quad (۸)$$

- میانگین خطا[۲۱]: نشان‌دهندهٔ میانگین خطای تولید شده به‌ازای اعمال کلیهٔ حالات ورودی است که از طریق رابطهٔ (۹) محاسبه می‌شود:

$$MErr = \frac{1}{N}\sum_{i=1}^{N} Err_i \quad (۹)$$

- میانگین اندازهٔ خطا[۲۲]: نشان‌دهندهٔ میانگین اندازهٔ خطای تولیدشده به‌ازای اعمال کلیهٔ حالات ورودی است که از طریق رابطهٔ (۱۰) محاسبه می‌شود:

$$MED = \frac{1}{N}\sum_{i=1}^{N} ED_i \quad (۱۰)$$

- میانگین اندازهٔ خطای نرمالیزه‌شده[۲۳]: نشان‌دهندهٔ نسبت میانگین اندازهٔ خطای ضرب‌کننده به بزرگ‌ترین خروجی ممکن تولیدشده توسط ضرب‌کننده است که از طریق رابطهٔ (۱۱) محاسبه می‌شود:

$$NMED = \frac{MED}{(2^n - 1)^2} \quad (۱۱)$$

- اندازهٔ خطای نسبی[۲۴]: نشان‌دهندهٔ اندازهٔ خطای تولیدشده نسبت به خروجی دقیق است که از طریق رابطهٔ (۱۲) محاسبه می‌شود:

$$RED = \left|\frac{ED}{O}\right| \quad (۱۲)$$

- میانگین اندازهٔ خطای نسبی[۲۵]: نشان‌دهندهٔ میانگین اندازهٔ خطای نسبی تولیدشده به‌ازای اعمال کلیهٔ حالات ورودی است که از طریق رابطهٔ (۱۳) محاسبه می‌شود:

$$MRED = \frac{1}{N}\sum_{i=1}^{N} \left|\frac{ED_i}{O_i}\right| \quad (۱۳)$$

- میانگین مربعات خطا[۲۶]: نشان‌دهندهٔ میانگین مجذور خطای تولیدشده به‌ازای اعمال کلیهٔ حالات ورودی است که از طریق رابطهٔ (۱٤) محاسبه می‌شود:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} ED_i^2 \quad (۱٤)$$

- تعداد بیت مؤثر[۲۷]: تخمینی از تعداد بیت پرارزش خروجی می‌دهد که به صورت دقیق توسط ضرب‌کننده تولید شده‌اند و از طریق رابطهٔ (۱۵) محاسبه می‌شود:

$$NoEB = 2n - \log_2\left(1 + \sqrt{MSE}\right). \quad (۱۵)$$

برای محاسبهٔ پارامترهای خطای هر ضرب‌کننده، ابتدا کد وریلاگ[۲۸] مربوط به توصیف عملکرد آن نوشته شد. سپس، تمامی حالات ممکن ورودی‌های هشت‌بیتی به ضرب‌کننده اعمال و خروجی‌ها توسط نرم‌افزار ModelSim استخراج و به صورت یک جدول ۲۵٦×۲۵٦ ذخیره شدند که در ادامهٔ شبیه‌سازی‌های مقاله نیز از این جدول استفاده شده است. با تعریف پارامترهای خطا در نرم‌افزار MATLAB و مقایسهٔ خروجی‌های ضرب‌کنندهٔ پیشنهادی با مقادیر دقیق، مقادیر خطا اندازه‌گیری شدند که نتایج آن در جدول (٤) ارائه شده است. همان‌طور که از نتایج استنباط می‌شود، میانگین خطای ($MErr$) ضرب‌کننده‌های تقریبی نوع اول مقداری مثبت و با اندازه‌ای نسبتاً بزرگ است که نشان می‌دهد خروجی‌های این

ضرب‌کننده‌ها عموماً بزرگ‌تر از مقادیر دقیق هستند که علت آن، استفاده از جمع‌کننده‌های افزایشی در مرحلهٔ اول ساختار ضرب‌کننده است. اما در ساختار دوم که از جمع‌کننده‌های کاهشی به همراه جمع‌کننده‌های افزایشی در مرحلهٔ اول استفاده شده است، میانگین خطا به‌شدت کاهش یافته و منجر به میانگین خطای نسبی کوچک‌تر ($MRED$) نیز شده است که مطلوب است. نتایج نشان می‌دهد با در نظر گرفتن یک مقدار مشخص برای تعداد ستون‌های تقریبی، با استفاده از ساختار تقریبی نوع دوم می‌توان به دقت بیشتر (خطای کوچک‌تر) نسبت به ساختار تقریبی نوع اول دست یافت. به همین دلیل، در ادامهٔ این مقاله، از ساختار دوم برای مقایسه با کارهای پیشین و همچنین بررسی عملکرد ساختار پیشنهادی در کاربردهای تحمل‌پذیر خطا استفاده خواهد شد.

جدول (٤): پارامترهای خطای ضرب‌کننده‌های تقریبی پیشنهادی

ساختار	$MErr$	MED	$NMED$ ($\times 10^{-4}$)	$MRED$ (%)	$NoEB$
M1_8	۱٦٥/۲	۱۸٥/۳	۲۹	٥/۸	۸/۲۳
M2_8	-٦۲/۲	۱٤۰/۷	۲۲	۳/۷	۸/۳۹
M1_9	۱۸٥/٦	۲۹٥/٤	٤٥	۸/۹	۷/٥۲
M2_9	-۱۱۳/٤	۲۷۳/۹	٤۲	٦/۰	۷/٤۲
M1_10	۳۹۹/۳	٥٦٥/۰	۸۷	۱٤/۱	٦/٥۹
M2_10	-۲۰۳/۳	٥۱٤/۲	۷۹	۹/٤	٦/٥۲
M1_11	۸٥۷/۰	۱۰۸۰/۰	۱٦٦	۲۱/٥	٥/٦٦
M2_11	-۱۱۰/۳	۸٥٤/۰	۱۳۱	۱۳/۹	٥/۸۰

۲-۳- استخراج مشخصات سخت‌افزاری

برای استخراج پارامترهای سخت‌افزاری، تمامی ضرب‌کننده‌ها به زبان وریلاگ توصیف شدند و سپس، با استفاده از نرم‌افزار Synopsys Design Compiler، مشخصات سخت‌افزاری هر ساختار شامل تأخیر، توان مصرفی و مساحت اشغالی به دست آمدند. فرایند سنتز ساختارها با استفاده از فناوری ٤٥ نانومتر Nangate و در شرایط گوشهٔ معمولی[۲۹]، دمای ۲٥ درجهٔ سانتی‌گراد و ولتاژ تغذیهٔ ۱/۱ ولت انجام شد. توان مصرفی هر ضرب‌کننده در بیشترین فرکانس کاری قابل دست‌یابی برای آن ساختار اندازه‌گیری شده است. نتایج به‌دست‌آمده برای ضرب‌کنندهٔ دقیق و همچنین ضرب‌کننده‌های تقریبی پیشنهادی نوع دوم در جدول (٥) ارائه شده است.

برای نمایش بهتر عملکرد ضرب‌کنندهٔ پیشنهادی در مقایسه با ضرب‌کنندهٔ دقیق، میزان بهبود مشخصات سخت‌افزاری آن، مانند تأخیر، توان مصرفی، مساحت اشغالی، انرژی مصرفی، حاصل‌ضرب انرژی در تأخیر (EDP)[30] و حاصل‌ضرب توان در تأخیر در مساحت (PDA)[31] در شکل (٦) ارائه شده است. همان‌طور که مشاهده می‌شود، با استفاده از ساختار پیشنهادی می‌توان انرژی مصرفی را تا ۹۲ درصد و تأخیر را تا ٥۱ درصد کاهش داد.

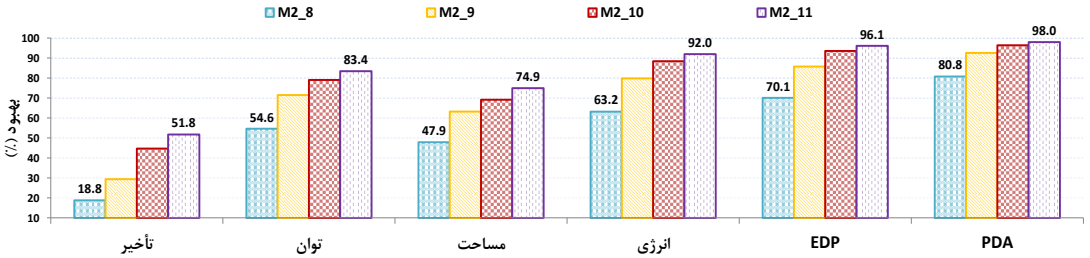
جدول (٥): مشخصات سخت‌افزاری ضرب‌کننده‌های تقریبی پیشنهادی به همراه ضرب‌کنندهٔ دقیق

ساختار	تأخیر (nS)	توان (μW)	مساحت (μm^2)	انرژی (fJ)	$MRED$ (%)
دقیق	۰/۸٥	۳٦۰	٤۱۷	۳۰٦	۰
M2_8	۰/٦۹	۱٦۳	۲۱۷	۱۱۳	۳/۷
M2_9	۰/٦۰	۱۰۳	۱٥۳	٦۲	٦/۰
M2_10	۰/٤۷	۷٥	۱۲۹	۳٥	۹/٤
M2_11	۰/٤۱	٦۰	۱۰٤	۲٤	۱۳/۹

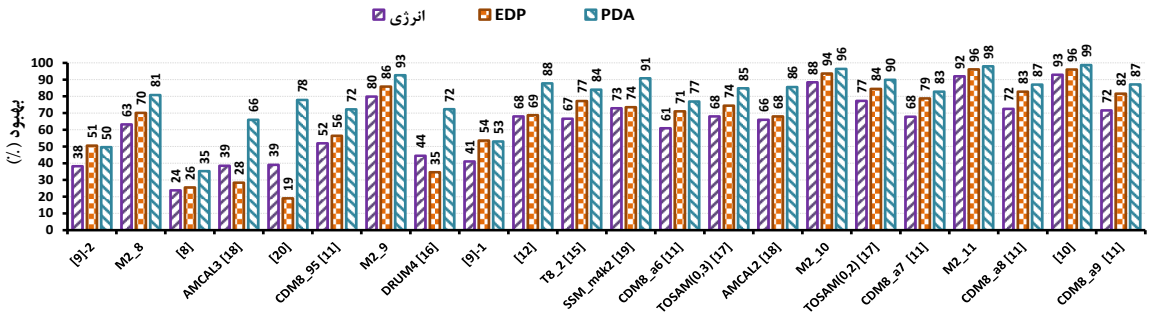
برای بررسی عملکرد ضرب‌کننده‌های پیشنهادی در مقایسه با ضرب‌کننده‌های تقریبی پیشین، میزان بهبود انرژی، EDP و PDA آنها نسبت به ضرب‌کنندهٔ دقیق در شکل (۷) نمایش داده شده است. در این شکل، ساختارهای مختلف بر اساس میزان MRED مرتب شده‌اند؛ به بیان دیگر، ساختار ۲- [9] کمترین و ساختار CDM8_a9 بیشترین MRED را دارد. نتایج نشان می‌دهد ساختار پیشنهادی نسبت به ساختارهای پیشین با مقادیر MRED مشابه، عملکردی بهتر از نظر مشخصات سخت‌افزاری از خود نشان می‌دهد. برای مثال، ساختار M2_8 با MRED=3.7% منجر به کاهش ٦۳، ۷۰ و ۸۱ درصد در انرژی، EDP و PDA نسبت به ضرب‌کنندهٔ دقیق می‌شود؛ در حالی که، ساختار ارائه‌شده در [۸] با MRED=3.8% منجر به بهبود ۲٤، ۲۷ و ۳٥ درصد در پارامترهای یادشده می‌شود و میزان بهبود برای [۱۱] CDM8_95 با MRED=5.1% برابر ٥۲، ٥۸ و ۷۲ درصد است که نشان‌دهندهٔ عملکرد بهتر ساختار پیشنهادی نسبت به

کارهای پیشین است.

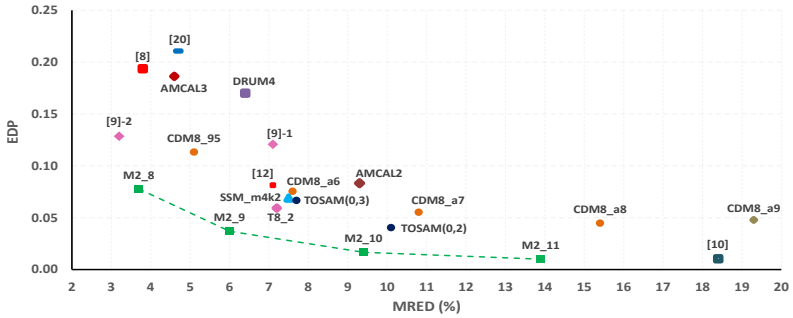
برای نمایش بهتر عملکرد ضرب‌کنندهٔ پیشنهادی در مقایسه با ساختارهای پیشین با در نظر گرفتن مشخصات سخت‌افزاری و دقت ضرب‌کننده‌ها، نمودار EDP برحسب MRED آنها در شکل (۸) رسم شده است. همان‌طور که خط‌چین سبزرنگ رسم‌شده در این شکل نشان می‌هد، ساختار پیشنهادی عملکردی بهتر نسبت به ساختارهای پیشین از خود نشان می‌دهد و می‌تواند به‌ازای میزان خطای قابل قبول، کمترین EDP را داشته باشد.



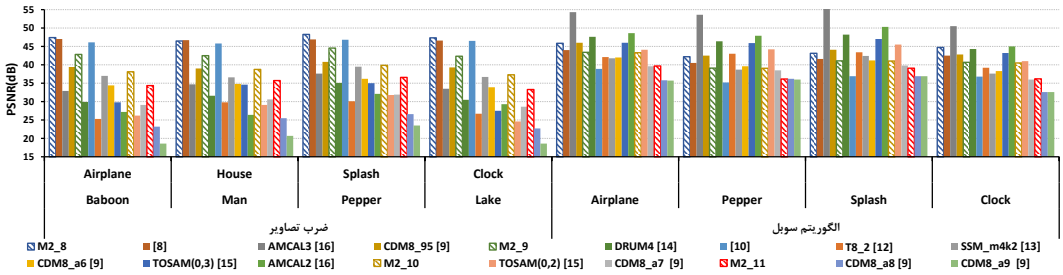
شکل (٦): میزان بهبود مشخصات سخت‌افزاری ضرب‌کنندهٔ تقریبی پیشنهادی در مقایسه با ضرب‌کنندهٔ دقیق



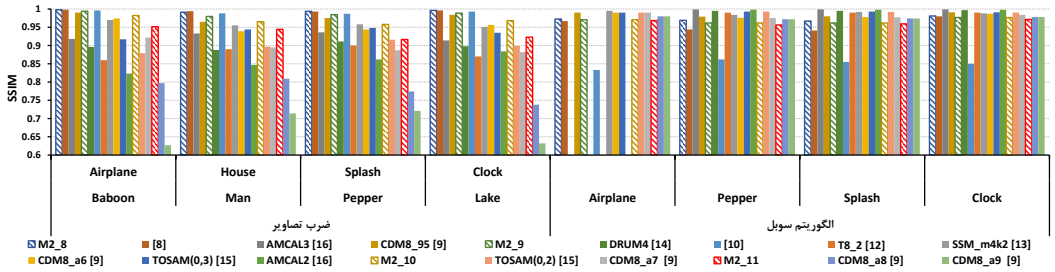
شکل (۷): میزان بهبود مشخصات سخت‌افزاری ضرب‌کنندهٔ تقریبی پیشنهادی و ضرب‌کننده‌های پیشین در مقایسه با ضرب‌کنندهٔ دقیق



شکل (۸): مقایسهٔ عملکرد ضرب‌کننده‌های تقریبی با در نظرگرفتن EDP و MRED آنها



شکل (۹): مقایسهٔ PSNR تصاویر تولیدشده با ضرب‌کننده‌های تقریبی پیشنهادی و پیشین نسبت به تصاویر دقیق



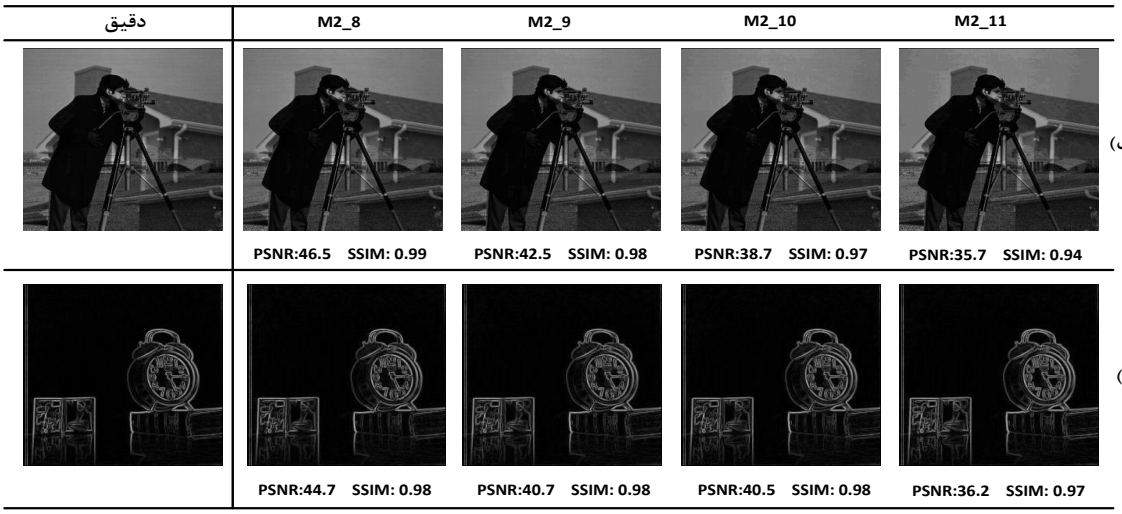
شکل (۱۰): مقایسهٔ SSIM تصاویر تولیدشده با ضرب‌کننده‌های تقریبی پیشنهادی و پیشین نسبت به تصاویر دقیق

۴- کاربردها

در این بخش، عملکرد ضرب‌کننده‌های پیشنهادی در کاربردهای پردازش تصویر (شامل ضرب تصاویر و یافتن لبهٔ تصاویر با الگوریتم سوبل[۳۲]) و همچنین طبقه‌بندی تصاویر با استفاده از شبکه‌های عصبی مصنوعی ارزیابی می‌شود.

۴-۱- پردازش تصاویر

برای بررسی کارایی ضرب‌کننده‌های پیشنهادی در کاربرد پردازش تصویر، عملکرد آنها در کاربردهای ضرب تصاویر و یافتن لبه‌های تصاویر با الگوریتم سوبل ارزیابی شد. برای این منظور، الگوریتم‌های سوبل و ضرب تصاویر در نرم‌افزار MATLAB اجرا شدند. سپس، با استفاده از یک جدول ۲۵٦×۲۵٦ که خروجی ضرب‌کننده را برای تمامی حالات ورودی هشت‌بیتی مشخص می‌کرد، تصاویر خروجی تولید شدند. سپس، پارامترهای PSNR و SSIM تصاویر تقریبی در مقایسه با تصاویر دقیق، توسط نرم‌افزار MATLAB محاسبه شدند و نتایج آن در شکل (۹) و شکل (۱۰) ارائه شده است که نشان‌دهندهٔ عملکرد بهتر ساختار پیشنهادی (در بیشتر حالات) نسبت به سایر ساختارها با MRED مشابه است. اگرچه در برخی از حالات، PSNR یا SSIM تصاویر تولیدشده با ضرب‌کننده‌های پیشنهادی کمتر از برخی از ضرب‌کننده‌های پیشین است، اما کیفیت تصاویر تا حدی زیاد است که چشم انسان توانایی تشخیص این تفاوت‌ها را نسبت به تصاویر دقیق ندارد. برای مثال، چند نمونه از تصاویر تولیدشده توسط ضرب‌کننده‌های پیشنهادی در مقایسه با ضرب‌کننده‌های دقیق در شکل (۱۱) نمایش داده شده‌اند.



شکل (۱۱): تصاویر تولیدشده توسط ضرب‌کننده‌های پیشنهادی در مقایسه با ضرب‌کنندهٔ دقیق در کاربردهای (الف) ضرب تصاویر و (ب) یافتن لبهٔ تصاویر با الگوریتم سوبل

۲-۴- شبکه‌های عصبی مصنوعی

در این بخش، کارایی ضرب‌کننده‌های پیشنهادی در کاربرد طبقه‌بندی تصاویر با استفاده از شبکه‌های عصبی مصنوعی بررسی می‌شود. برای شبیه‌سازی‌ها، از چارچوب AdaPT [۲۱] (مبتنی بر زبان پایتون) استفاده شد. به منظور ارزیابی عملکرد ضرب‌کنندهٔ پیشنهادی و سایر ضرب‌کننده‌های تقریبی در کاربرد شبکه‌های عصبی مصنوعی، برای هر ضرب‌کننده یک جدول ۲۵۶×۲۵۶ تولید شد که خروجی آن را برای تمامی حالات ورودی هشت‌بیتی نشان می‌دهد. این جدول‌ها به عنوان مدل عملکرد ضرب‌کننده‌ها در شبیه‌ساز Adapt استفاده شدند. در این شبیه‌ساز، ضرب‌کننده‌های موجود در تمام لایه‌های شبکه عصبی با جدول توصیف‌کنندهٔ ضرب‌کنندهٔ تقریبی تعویض می‌شوند و صحت[۳۳] شبکه اندازه‌گیری می‌شود. استفاده از جدول به‌جای مدل ریاضی ضرب‌کننده‌ها با هدف افزایش سرعت شبیه‌ساز انجام شده است. نتایج به‌دست‌آمده در جدول (٦) ارائه شده است.

مدل FC-3 یادشده در جدول (٦) یک مدل کوچک و دارای سه لایهٔ تماماً متصل[۳٤] با تعداد ۵۱۲، ۲۵٦، و ۱۰ نورون در سه لایهٔ متوالی است. سایر مدل‌ها نیز مدل‌های شبکه عصبی پیچشی[۳۵] شناخته‌شده هستند که برای کاربردهای طبقه‌بندی تصاویر استفاده می‌شوند. نتایج نشان می‌دهد ضرب‌کنندهٔ پیشنهادی به‌ازای تمامی زوج حالت‌های مدل مجموعه داده، عملکردی قابل قبول را از خود نشان داده است. برای مثال، بیشترین افت صحت مدل‌های ساخته‌شده با ضرب‌کنندهٔ M2_8 نسبت به ضرب‌کنندهٔ دقیق برابر ۲/۵٦ درصد است که در طبقه‌بندی تصاویر مجموعه دادهٔ CIFAR-10 با استفاده از مدل DenseNet-121 اتفاق می‌افتد.

همچنین، صحت ضرب‌کنندهٔ M2_9 به‌ازای تمامی زوج‌های مدل مجموعه داده، اختلاف بسیار کمی با صحت ضرب‌کنندهٔ M2_8 دارد و در بدترین حالت، صحت آن نسبت به M2_8 به میزان ۰/۸۳ درصد افت می‌کند؛ در حالی که همان‌گونه که در بخش ۳ بیان شد، مشخصات سخت‌افزاری هر دو ضرب‌کننده بهبودی جالب توجه را نسبت به ضرب‌کنندهٔ دقیق از خود نشان می‌دهند. مشاهدهٔ دیگر این است که تفاوت صحت ضرب‌کننده‌های تقریبی متفاوت در مدل‌های کوچک‌تر منجر به تفاوت‌هایی کمتر در صحت طبقه‌بندی شبکهٔ عصبی می‌شود، در حالی که در مدل‌های بزرگ‌تر (که برای طبقه‌بندی مجموعه دادهٔ CIFAR-10 استفاده شده‌اند) خطای زیاد ضرب‌کننده‌ها موجب افت شدید صحت طبقه‌بندی شبکه شده است؛ به طوری که دو ضرب‌کنندهٔ پیشنهادی M2_10 و M2_11 و سایر ضرب‌کننده‌های تقریبی که دارای خطای قابل مقایسه با این دو ساختار هستند، در مدل‌های بزرگ‌تر صحتی غیرقابل قبول را از خود نشان می‌دهند.

جدول (٦): دقت مدل‌های مختلف شبکهٔ عصبی برای طبقه‌بندی مجموعه داده‌های مختلف با استفاده از ضرب‌کننده‌های دقیق و تقریبی مختلف

مجموعه داده	MNIST		CIFAR-10		
مدل شبکهٔ عصبی	FC-3	LeNet-5	VGG-11	ResNet-18	DenseNet-121
ضرب‌کننده‌ی دقیق	۹۷/۸۲	۹۷/٦۳	۹۲/۳٤	۹۲/۹۲	۹۳/۹۹
[9]-2	۹۷/۷۵	۹۷/۵۲	۹۱/۵۱	۹۱/٤۷	۹۲/۲۰
[8]	۹۷/۷۷	۹۷/٤۸	۹۱/٤۸	۹۱/۵۲	۹۱/۷۳
AMCAL3 [18]	۹۷/٦۵	۹۷/۳۷	۹۰/۵۸	۹۰/۸۲	۹۱/۱۱
CDM8_95 [11]	۹۷/٦٦	۹٦/۹۳	۹۰/۹۰	۹۰/۷۵	۹۱/۱٦
DRUM4 [16]	۹۷/۵۸	۹٦/٤٤	۹۰/۳۷	۹۰/۸٤	۹۱/۰۷
[9]-1	۹۷/۵۵	۹۵/٦۸	۹۰/۳۱	۹۰/٦۰	۹۱/۱۱
[12]	۹۷/۵۰	۹۵/٤۷	۹۰/۲۷	۹۰/۵۱	۹۰/٤٦
T8_2 [14]	۹۷/٤۹	۹۵/۳۷	۹۰/۲۱	۹۰/٤۵	۹۰/۱۱
SSM_M4k2 [15]	۹۷/۵۵	۹۵/٦٤	۸۸/۱۷	۹۰/۱٦	۸۹/۸۲
CDM8_a6 [11]	۹۷/۵٦	۹۵/۱۹	۸۷/۳۱	۸۹/۸۱	۸۸/٦۲

مجموعه داده	MNIST		CIFAR-10		
مدل شبکهٔ عصبی	FC-3	LeNet-5	VGG-11	ResNet-18	DenseNet-121
TOSAM (0,3) [17]	۹۷/۱٦	۹٥/۲۱	۸٦/٤۷	۹۰/۰۲	۸۸/٤۷
AMCAL2 [18]	۹٦/۷٥	۹٥/٥۷	۸٦/۳۳	۸۸/۲٥	٤٤/۱٥
TOSAM (0,2) [17]	۹٦/٥۱	۹٥/۰۷	۸٥/٥۹	۸۸/۱٦	۳۷/۲٥
CDM8_a7 [11]	۹٦/۷۰	۹٤/٦٦	۸٥/٤۹	٦۷/۹۹	۳۳/٤۳
CDM8_a8 [11]	۹٦/٤٤	۹۰/۳۳	٤٥/۳۳	۲۱/٦۹	۹/٦۱
[10]	۹٦/۲۸	۹۰/۰۱	٤٤/٦۸	۱۹/٦۳	۱۳/۹۳
CDM8_a9 [11]	۹٦/۱۷	۸۹/۹٥	٤۳/۲٥	۱۸/۷۲	۷/۹٤
M2_8	۹۷/٥٦	۹۷/۳٦	۹۱/٥۷	۹۱/۲٥	۹۱/٤۳
M2_9	۹۷/۲٤	۹٦/٥۳	۹۰/۸۷	۹۱/۰٤	۹۱/۱۳
M2_10	۹٦/۸۹	۹٥/۷۱	۸٦/٤۱	۸۸/۳۸	٤۰/٤۱
M2_11	۹٦/٦٥	۹۰/۲٤	٤۹/٥۹	۲٥/۸۳	۱۰/٥٦

به منظور ارائهٔ یک ارزیابی بهتر از عملکرد ضرب‌کنندهٔ ارائه‌شده در این کاربرد، نتایج پارامترهای امتیاز F1[36]، دقت[37]، بازیابی[38] و ماتریس درهم‌ریختگی[39] برای حالت طبقه‌بندی مجموعه دادهٔ CIFAR-10 با استفاده از مدل VGG-11 با استفاده از چهار حالت ضرب‌کنندهٔ ارائه‌شده به تفکیک کلاس خروجی در جدول (۷) نمایش داده شده است. نتایج نشان می‌دهد امتیاز F1 و دقت برای ضرب‌کننده‌های M2_8 و M2_9 همواره بیش از ۸۰ درصد هستند. همچنین، میانگین پارامترهای امتیاز F1 و بازیابی میان ۱۰ کلاس خروجی، تقریباً با صحت ارائه‌شده برای هر ضرب‌کننده در جدول (٦) برابر است. مجموعهٔ مشاهدات بالا نشان می‌دهد ضرب‌کنندهٔ پیشنهادی در کاربرد شبکه‌های عصبی مصنوعی نیز می‌تواند با افت دقت جزئی نسبت به شبکه‌های عصبی دقیق، عملکردی بسیار خوب را از خود نشان دهد.

۵- نتیجه‌گیری

در این مقاله، دو روش برای اجرای تقریبی واحدهای جمع‌کننده ارائه شدند که در یک روش، به طور میانگین، خطای تولید خروجی‌ها مثبت و در روش دیگر، منفی بود. با استفاده از این جمع‌کننده‌ها در ساختار یک ضرب‌کنندهٔ هشت‌بیتی، میزان انرژی مصرفی و مساحت اشغالی تا ۹۲ و ۷۵ درصد نسبت به ضرب‌کنندهٔ دقیق کاهش یافت؛ در حالی که میانگین خطای نسبی ضرب‌کننده کمتر از ۱٤ درصد بود. علت خطای اندک ضرب‌کننده قرارگیری هوشمندانهٔ جمع‌کننده‌های پیشنهادی در ساختار آن بود؛ به نحوی که خطای یکدیگر را به‌خوبی خنثی کنند. با استفاده از ضرب‌کننده‌های پیشنهادی در کاربردهای تحمل‌پذیر خطا، مانند پردازش تصاویر و شبکه‌های عصبی مصنوعی، کارایی آنها در این کاربردها نیز بررسی شد. نتایج نشان داد تصاویر تقریبی تولیدشده با ضرب‌کننده‌های پیشنهادی کیفیت زیادی دارند و چشم انسان قادر به تشخیص این تفاوت‌های ناچیز نیست. همچنین، با استفاده از ضرب‌کننده‌های پیشنهادی در ساختار پنج مدل شبکهٔ عصبی مصنوعی، مشاهده شد دقت مدل‌های ساخته‌شده با ضرب‌کنندهٔ پیشنهادی بسیار به دقت مدل دقیق معادل خود نزدیک است؛ به طوری که دقت مدل‌های ساخته‌شده با ضرب‌کنندهٔ تقریبی M2_8، به‌ازای کاهش توان مصرفی به میزان ۵۵ درصد، در بدترین حالت به اندازهٔ ۲/٥٦ درصد نسبت به دقت مدل دقیق افت کرد.

جدول (۷): سایر پارامترهای شبکهٔ عصبی برای طبقه‌بندی مجموعه دادهٔ CIFAR-10 با استفاده از مدل VGG-11 و حالات مختلف ضرب‌کنندهٔ پیشنهادی

ضرب‌کننده	کلاس خروجی	F1 Score	دقت	بازیابی	ماتریس درهم‌ریختگی									
M2_8	۰	۹۳/۸۲	۹۱/۳۷	۹۶/٤	۹٦٤	٥	٤	۳	۱	۰	۱	۲	۱۲	۸
	۱	۹۷/۷۳	۹٦/٤۹	۹۹	۱	۹۹۰	۰	۰	۰	۰	۰	۰	۳	٦
	۲	۸٦/۷۹	۹۷/۰۳	۷۸/٥	٥۲	۰	۷۸٥	۳۲	۲۷	۲۸	٥۱	۸	۸	۹
	۳	۸۱/۱۷	۸٤/۸٤	۷۷/۸	۷	۳	۱۲	۷۷۸	۲٥	۷۱	۳٦	۲٥	۲۱	۲۲
	٤	۹۰/٥۸	۹۲/۳۲	۸۸/۹	۸	۰	۷	۱۸	۸۸۹	٦	۲٥	٤٥	۲	۰
	٥	۸٤/۸۰	۸۸/۱۳	۸۱/۷	۷	۹	۰	۷۳	۱۷	۸۱۷	۱۹	۳۱	۱۸	۹
	٦	۹۳/۲٦	۸۸/۲۲	۹۸/۹	۰	۰	۱	۷	۲	۰	۹۸۹	۰	۰	۱
	۷	۹۳/۷۹	۸۹/۷٦	۹۸/۲	۳	۰	۰	٦	۲	٥	۰	۹۸۲	۰	۲
	۸	۹٥/۹٥	۹۳/۷۱	۹۸/۳	۹	٥	۰	۰	۰	۰	۰	۱	۹۸۳	۲
	۹	۹٦/۱۳	۹٤/۳۲	۹۸	٤	۱٤	۰	۰	۰	۰	۰	۲	۹۸۰	
M2_9	۰	۹۱/٦۸	۸۹/۷٥	۹۳/۷	۹۳۷	۷	٥	٤	۰	۰	۰	۲	۲۹	۱٦
	۱	۹٦/٤٤	۹٦/۷۸	۹٦/۱	٥	۹٦۱	۰	۰	۰	۰	۰	۰	۷	۲۷
	۲	۸٥/۸٤	۹٦/۰٤	۷۷/٦	٥۸	۰	۷۷٦	۲٦	۳۰	۳۱	٤٥	۱۱	۱۱	۱۲
	۳	۸۲/۰۰	۸٤/٦٦	۷۹/٥	٥	۲	۹	۷۹٥	۲٥	٦٤	۳۹	۲۲	۲۳	۱٦
	٤	۸۷/٤٤	۹۲/۱٤	۸۳/۲	۱٤	۳	۱٤	۲۷	۸۳۲	۱۳	۳٤	٥۱	۷	٥
	٥	۸۷/۳۷	۸۸/٦۷	۸٦/۱	۰	۰	۰	۷٦	۱۳	۸٦۱	۱۱	۲۸	۱۰	۱
	٦	۹۳/۱۹	۸۸/۳٥	۹۸/٦	۱	۰	۳	۷	۲	۰	۹۸٦	۰	۰	۱
	۷	۹٤/۰٦	۸۹/٥۹	۹۹	٤	۰	۰	۳	۰	۲	۰	۹۹۰	۰	۱
	۸	۹٤/۰۱	۹۱/٦٤	۹٦/٥	۱۷	۹	۱	۱	۱	۰	۱	۰	۹٦٥	٥
	۹	۹٥/۱٦	۹۲/۱۳	۹۸/٤	۳	۱۱	۰	۰	۰	۰	۰	۱	۱	۹۸٤
M2_10	۰	۸۹/٥۹	۸٥/٤۰	۹٤/۲	۹٤۲	٤۳	۰	۰	۰	۰	۰	۰	٤	۱۱
	۱	۷۸/۱۱	٦٦/۲۷	۹٥/۱	٦	۹٥۱	۰	۲	۰	۰	٥	۰	۱٦	۲۰
	۲	۸٤/۸۷	۹۸/۹۳	۷٤/۳	۷۹	٥۱	۷٤۳	۰	۷	۰	۲۸	٤۰	۰	٥۲
	۳	۷۹/۸٤	۹٦/۸٦	٦۷/۹	۱٤	۱۱۳	۰	٦۷۹	۲	۹۲	۳	۰	۰	۹۷
	٤	۸۸/٥۷	۹۸/۸۹	۸۰/۲	۹	٤٤	۸	۰	۸۰۲	۱۲	۰	۱۱٤	۰	۱۱
	٥	۸٤/۰۲	۸۸/۲۲	۸۰/۲	۰	٦۲	۰	۱۸	۰	۸۰۲	۰	۹۱	۰	۲۷
	٦	۹٤/۸۸	۹٦/۲۹	۹۳/٥	٤	۳٦	۰	۰	۰	۰	۹۳٥	۱۰	۰	۱٥
	۷	۸٦/٥٦	۷۸/۸۲	۹٦/۰	۱	۲٤	۰	۰	۰	۳	۰	۹٦۰	۰	۱۲
	۸	۹۳/۸۱	۹۷/۸۳	۹۰/۱	۳٦	٥٤	۰	۰	۰	۰	۰	۰	۹۰۱	۹
	۹	۸٤/۹٥	۷۸/٤۷	۹۲/٦	۱۲	٥۷	۰	۲	۰	۰	۰	۳	۰	۹۲٦
M2_11	۰	۷۰/٤۷	٦۹/۸٤	۷۱/۱	۷۱۱	۲۲٦	٦	٤	۰	۲	۰	۱٤	٥	۳۲
	۱	۳۹/۸۳	۲٤/۹۲	۹۹/۲	۱	۹۹۲	۰	۰	۰	۰	۰	۱	۰	٦
	۲	٤۲/۸٤	۸٤/٤۱	۲۸/۷	۹٤	۳۳٥	۲۸۷	۲۷	۹	۳۲	۲٦	۱۲٤	٤	٦۲
	۳	۳۲/۸۷	٦۱/۷۱	۲۲/٤	۳٤	٤۰٥	۹	۲۲٤	۹	۹٦	۱۸	۱۱۱	۸	۸٦
	٤	۳۱/۸۹	۸۷/٤٤	۱۹/٥	۳۰	۳٤۲	۲۱	۲۸	۱۹٥	٤۹	۳۸	۲۲۲	۸	٦۷
	٥	٤۷/٥۸	٦۳/۹۱	۳۷/۹										

ماتریس درهم‌ریختگی										بازیابی	دقت	F1 Score	کلاس خروجی	ضرب‌کننده
۵۶	۱۱	۱۴۸	۱۲	۳۷۹	۵	۴۶	۵	۳۱۹	۱۹	۴۱/۲	۸۱/۴۲	۵۴/۷۱	۶	
۸۰	۲	۵۷	۴۱۲	۱۲	۵	۲۳	۱۲	۳۷۶	۲۱	۷۴/۲	۵۲/۰۰	۶۱/۱۵	۷	
۴۸	۱	۷۴۲	۰	۲۳	۰	۱۰	۰	۱۶۸	۸	۳۲/۲	۸۹/۲۰	۴۷/۳۲	۸	
۵۶	۳۲۲	۶	۰	۰	۰	۱	۰	۵۱۸	۹۷	۶۹/۵	۵۸/۵۰	۶۳/۵۳	۹	
۶۹۵	۰	۲	۰	۰	۰	۰	۰	۳۰۰	۳					

مراجع

[1] W. Xiong, J. Cao, Y. Liu, J. Wang, J. Lai, M. Huang, "A Reliable and Efficient Online Solution for Adaptive Voltage and Frequency Scaling on FPGAs", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 32, No. 6, pp. 1058-1071, June 2024. https://doi.org/10.1109/TVLSI.2024.3361459

[2] C. S. Joice, C. Jenisha, S. H. Shanitini, R. Desika, D. Kirthana, "A Comprehensive Review of Various SRAM Using Power Gating Technique", 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), Tirunelveli, India, pp. 356-361, 2024. https://doi.org/10.1109/ICDICI62993.2024.10810787

[3] S. Sathish, N. Prajwal, R. Akshaya, Kaustubha, S. Kumar, "Impact of Clock-Gating on ALU Optimized RISC-V Microarchitectures for Low Power Applications", IEEE 4th International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI SATA), Bangalore, India, pp. 1-6, 2024. https://doi.org/10.1109/VLSISATA61709

[4] J. Vafaei, O. Akbari, "HPR-Mul: An Area and Energy-Efficient High-Precision Redundancy Multiplier by Approximate Computing", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 32, No. 11, pp. 2012-2022, Nov. 2024. https://doi.org/10.1109/TVLSI.2024.3445108

[5] M. H. Haider, H. Zhang, S. -B. Ko, "Decoder Reduction Approximation Scheme for Booth Multipliers", in IEEE Transactions on Computers, Vol. 73, No. 3, pp. 735-746, March 2024. https://doi.org/10.1109/TC.2023.3343093

[6] M. Reshadinezhad, S. E. Fatemieh, Z. Davari Shalamzari, "Design and optimization of an Approximate Full-adder Based on CNTFETs and its application in image processing," Computational Intelligence in Electrical Engineering (ISEE), Vol. 11, No. 3, pp. 25-36, 2020. https://doi.org/10.22108/isee.2020.120390.1313 [In Persian]

[7] L. Sayadi, S. Timarchi, A. Sheikh-Akbari, "Two Efficient Approximate Unsigned Multipliers by Developing New Configuration for Approximate 4:2 Compressors", in IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 70, No. 4, pp. 1649-1659, April 2023. https://doi.org/10.1109/TCSI.2023.3242558

[8] Y. Zhang, X. Chen, P. Guo, G. Xie, "Design and Analysis of Approximate Multiplier of Majority-Based Imprecise 4–2 Compressor for Image Processing", IEEE 23rd International Conference on Nanotechnology (NANO), Jeju City, Korea, pp. 1-5, 2023. https://doi.org/10.1109/NANO58406.2023.10231167

[9] L. H. Krishna, A. Sk, J. B. Rao, S. Veeramachaneni, N. M. Sk, "Energy-efficient approximate multiplier design with lesser error rate using the probability-based approximate 4:2 compressor", in IEEE Embedded Systems Letters, Vol. 16, No. 2, pp. 134-137, June 2024. https://doi.org/10.1109/LES.2023.3280199

[10] S. Shirkavand Saleh Abad, M. Moaiyeri, "Hardware-accuracy trade-offs for error-resilient applications using an ultra-efficient hybrid approximate multiplier", The Journal of Supercomputing, Vol. 79, pp. 3357-3372, Sep. 2022. https://doi.org/10.1007/s11227-022-04789-6

[11] N. Amirafshar, A. S. Baroughi, H. S. Shahhoseini, N. TaheriNejad, "Carry Disregard Approximate Multipliers", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 70, No. 12, pp. 4840-4853, Dec. 2023. https://doi.org/10.1109/TCSI.2023.3306071

[12] L. Sayadi, A. Amirany, M. H. Moaiyeri, S. Timarchi, "Balancing Precision and Efficiency: an Approximate Multiplier with

truncation- and rounding-based scalable approximate multiplier", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 27, No. 5, pp. 1161-1173, May 2019. https://doi.org/10.1109/TVLSI.2018.2890712

[18] R. Zendegani, S. Safari, "AMCAL: Approximate Multiplier With the Configurable Accuracy Levels for Image Processing and Convolutional Neural Network", IEEE Access, Vol. 12, pp. 94135-94151, 2024. https://doi.org/10.1109/ACCESS.2023.3273449

[19] L. Li, I. Hammad, K. El ‑ Sankary, "Dual segmentation approximate multiplier", Electronics Letters, Vol. 57, No. 19, pp. 718-720, 2021. https://doi.org/10.1049/ell2.12243

[20] M. Taheri, N. Cherezova, S. Nazari, A. Rafiq, A. Azarpeyvand, T. Ghasempouri, M. Daneshtalab, J. Raik, M. Jenihhin, "AdAM: Adaptive fault-tolerant approximate multiplier for edge DNN accelerators", 2024 IEEE European Test Symposium (ETS), The Hague, Netherlands, pp. 1-4, 2024. https://doi.org/10.1109/ETS61313.2024.10567161

[21] D. Danopoulos, G. Zervakis, K. Siozios, D. Soudris, J. Henkel, "AdaPT: Fast Emulation of Approximate DNN Accelerators in PyTorch", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 42, No. 6, pp. 2074-2078, June 2023. https://doi.org/10.1109/TCAD.2022.3212645

Built-in Error Compensation for Error-Resilient Applications", Early Access, The Journal of Supercomputing, Vol. 81, Article Number 109, 2025. https://doi.org//10.1007/s11227-024-06563-2

[13] D. Pandey, S. Singh, V. Mishra, S. Satapathy, D. S. Banerjee, "SAM: A Segmentation Based Approximate Multiplier for Error Tolerant Applications", IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, pp. 1-5, 2021. https://doi.org/10.1109/ISCAS51556.2021.9401266

[14] I. Nunziata, E. Zacharelos, G. Saggese, A. M. G. Strollo, E. Napoli, "Approximate Recursive Multipliers Using Carry Truncation and Error Compensation", 17th Conference on Ph.D Research in Microelectronics and Electronics (PRIME), Villasimius, SU, Italy, pp. 137-140, 2022. https://doi.org/10.1109/PRIME55000.2022.9816787

[15] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, G. Saggese, G. Di Meo, "Approximate multipliers using static segmentation: error analysis and improvements", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 69, No. 6, pp. 2449-2462, June 2022. https://doi.org/10.1109/TCSI.2022.3152921

[16] S. Hashemi, R. I. Bahar, S. Reda, "DRUM: a dynamic range unbiased multiplier for approximate applications", in Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 418-425, 2015. https://doi.org/10.1109/ICCAD.2015.7372600

[17] S. Vahdat, M. Kamal, A. Afzali-Kusha, M. Pedram, "TOSAM: An energy-efficient

[1] Dynamic Voltage Frequency Scaling
[2] Clock
[3] Partial product generation
[4] Operand
[5] Partial product reduction
[6] Partial product accumulation
[7] Booth Multiplier
[8] Full adder
[9] Half adder
[10] Compressors
[11] Truth table
[12] Error compensation modules

[13] Truncation
[14] Static
[15] Dynamic
[16] Shift
[17] Gate
[18] Dot diagram
[19] Error
[20] Error Distance
[21] Mean Error
[22] Mean Error Distance
[23] Normalized Mean Error Distance
[24] Relative Error Distance
[25] Mean Relative Error Distance
[26] Mean Squared Error
[27] Number of Effective Bits
[28] Verilog
[29] Typical corner
[30] Energy-Delay Product
[31] Power-Delay-Area product
[32] Sobel edge detection algorithm
[33] Accuracy
[34] Fully-Connected
[35] Convolutional
[36] F1 Score
[37] Precision
[38] Recall
[39] Confusion Matrix