بسم الله الرحمن الرحيم

دانشگاه اصفهان

دانشکده فنی و مهندسی

گروه مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد رشته‌ی مهندسی‌کامپیوتر گرایش معماری سیستم‌های کامپیوتر

ضرب‌کننده ترتیبی ۶۴ بیتی تخمینی با دقت پیش‌رونده

استاد راهنما:

دکتر هومان نیک‌مهر

پژوهشگر:

سید محمد مهدی تابعی

دی ماه ۱۳۹۱

کلیه حقوق مادی مترتب بر نتایج مطالعات، ابتکارات و نوآوری‌های ناشی از تحقیق موضوع این پایان‌نامه متعلق به دانشگاه اصفهان است.

دانشگاه اصفهان

دانشکده فنی مهندسی

گروه مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد رشته‌ی مهندسی‌کامپیوتر گرایش معماری سیستم‌های کامپیوتر

آقای سید محمد مهدی تابعی

تحت عنوان

ضرب‌کننده ترتیبی ۶۴ بیتی تخمینی با دقت پیش‌رونده

در تاریخ **۱۳۹۱/۱۰/۳۰** توسط هیأت داوران زیر بررسی و با درجه **بسیار خوب** به تصویب نهایی رسید.

۱ – استاد راهنمای پایان‌نامه **دکتر هومان نیک‌مهر** با مرتبه‌ی علمی استادیار امضا

۲ – استاد داور داخل گروه **دکتر کمال جمشیدی** با مرتبه‌ی علمی دانشیار امضا

۳ – استاد داور خارج از گروه **دکتر مهدی حبیبی** با مرتبه‌ی علمی استادیار امضا

امضای مدیر گروه

با سپاس از ...

استاد گرانقدر جناب آقای دکتر نیک‌مهر که زحمات بسیاری در این راه کشیدند و راهنمایی‌های صمیمانه و دلسوزانه ایشان چراغ راه من بود.

و

با تشکر از همسر عزیزم که همراهی و دلگرمی او انگیزه‌ی من و مایه‌ی آرامش خاطرم در پیمودن این مسیر شد.

تقدیم به مادر عزیزتر از جانم:

مادرم هستی من ز هستی توست تا هستم و هستی دارمت دوست

غمگسار جاودانی مادر است

چشمسار مهربانی مادر است

تقدیم با بوسه بر دستان پدرم:

به او که نمی‌دانم از بزرگی‌اش بگویم یا مردانگی سخاوت، سکوت، مهربانی و ...

پدرم راه تمام زندگیست

پدرم دلخوشی همیشگیست

چکیده

محاسبات تخمینی یکی از جدیدترین و پرکاربردترین شاخه‌های محاسبات دیجیتال است. افزایش سرعت، پایین آوردن توان مصرفی و کاهش مساحت واحدهای محاسباتی در کنار فراهم نمودن دقت مناسب از اهداف این ایده‌ی نوین می‌باشند که با حذف یا تغییر قسمتی از مدار واحدهای مزبور محقق می‌گردد. یکی از کلیدی‌ترین واحدهای محاسباتی ضرب‌کننده‌ها می‌باشند که نقش بسزایی در اکثر سیستم‌های دیجیتال بر عهده دارند و به همین دلیل ضرب‌کننده‌های موازی برش‌یافته که یکی از معروف‌ترین روش‌های تخمینی است در دو دهه‌ی اخیر مورد توجه فراوانی قرار گرفته‌اند. به علاوه سرعت کم ضرب ترتیبی باعث شده است تا برخلاف ساختار ساده‌تر و توان مصرفی پایین‌تر نسبت به ضرب موازی مورد توجه کمتری قرار بگیرد. بنابراین در این تحقیق سعی شده با توجه به کارایی روش برش‌یافته، ضرب‌کننده‌ای ترتیبی ارائه شود که از این ایده استفاده می‌نماید و سرعت ضرب ترتیبی را تا حدودی بهبود بدهد. در ضرب‌کننده‌ی پیشنهادی برای جبران خطای ناشی از اِعمال روش برش‌یافته، از ستون $(n-1)$ام ماتریس ضرب‌های جزیی استفاده می‌گردد تا خطای به وجود آمده در هر سیکل تا حد امکان بهبود یابد. با شبیه‌سازی و سنتز بر روی ASIC، دقت این روش و بهبودی که در زمینه‌ی سرعت داده است نیز بررسی می‌گردد. کاربرد این ضرب‌کننده در سیستم‌های تعبیه‌شده یا متحرک می‌تواند باشد که مساحت و توان مصرفی پایین در آن‌ها حیاتی است و سرعت بالاتر این روش نیز به کارایی بهتر این سیستم‌ها کمک می‌نماید.

واژگان کلیدی:

محاسبات تخمینی، ضرب‌کننده‌ی ترتیبی، روش برش‌یافته، جبران خطا، رقم نقلی، تأخیر، توان مصرفی، مساحت

فهرست مطالب

عنوان **صفحه**

فصل اول- مقدمه

۱-۱ معرفی موضوع ۱

فصل دوم- واحدهای محاسباتی تخمینی

۲-۱ مقدمه ۷

۲-۲ جمع‌کننده ۷

۲-۳ جمع‌کننده‌های تخمینی ۱۰

۲-۳-۱ جمع‌کننده‌ی تقریبی ۱۰

۲-۳-۲ جمع‌کننده تخمینی نقلی ۱۱

۲-۳-۲ جمع‌کننده‌ی چند عملوندی تخمینی ۱۵

۲-۴ ضرب‌کننده ۱۸

۲-۴-۱ ضرب‌کننده موازی ۱۹

۲-۴-۲ ضرب‌کننده ترتیبی ۲۰

۲-۴-۳ ضرب‌کننده‌ی برش‌یافته ۲۴

۲-۴-۴ ضرب‌کننده‌ی برش‌یافته موازی ۲۵

۲-۴-۵ روش‌های تصحیح ثابت ۲۶

۲-۴-۶ روش‌های تصحیح متغیر ۲۸

۲-۴-۷ ضرب‌کننده‌ی برش‌یافته ترتیبی ۳۵

۲-۵ روش‌های تخمینی دیگر ۳۶

۲-۵-۱ ضرب‌کننده‌ی تقریبی علامت‌دار ۳۶

۲-۵-۲ ضرب‌کننده‌ی بلوکی تخمینی ۳۸

۲-۶ نتیجه‌گیری ۴۰

عنوان **صفحه**

فصل سوم- ضرب‌کننده‌ی برش‌یافته‌ی ترتیبی با قابلیت جبران خطای متغیر

۳-۱ مقدمه ۴۱

۳-۲ مفاهیم و کلیات ۴۱

۳-۳ ضرب‌کننده‌ی ترتیبی برش‌یافته ۴۳

۳-۳-۱ منابع ایجاد خطا ۴۶

۳-۴ طراحی روش پیشنهادی ۴۷

۳-۵ روش بالا به پایین ۴۸

۳-۵-۱ روش‌های جبران خطا ۵۰

۳-۵-۲ بهبود روش پیشنهادی ۶۰

۳-۵-۳ جبران خطا مستقل از LSP ۶۳

۳-۶ روش پایین به بالا ۶۶

۳-۶-۱ روش‌های جبران خطا ۶۸

۳-۷ نتیجه‌گیری ۷۳

فصل چهارم- آنالیز و تحلیل خطای روش پیشنهادی

۴-۱ مقدمه ۷۵

۴-۲ انتشار رقم نقلی ۷۵

۴-۳ تحلیل خطای ارزیابی ۷۶

۴-۳-۱ خطا در ضرب‌کننده‌ی ترتیبی برش‌یافته‌ی بدون قسمت جبران خطا ۷۶

۴-۳-۲ خطا در ضرب‌کننده‌ی ترتیبی برش‌یافته‌ی با قسمت جبران خطا ۸۲

۴-۳-۳ به دست آوردن مقدار خطا برای ضرب‌کننده‌های با ابعاد مختلف ۸۵

۴-۴ نتیجه‌گیری ۸۶

فصل پنجم- ارزیابی روش پیشنهادی

عنوان **صفحه**

۵-۱ مقدمه ۸۷
۵-۲ ارزیابی دقت ضرب‌کننده‌ی برش یافته‌ی ترتیبی ۸۸
۵-۲-۱ ضرب‌کننده‌ی برش‌یافته‌ی ترتیبی ۸ بیتی ۸۸
۵-۲-۲ ضرب‌کننده‌ی برش‌یافته‌ی ترتیبی ۱۶ بیتی ۹۵
۵-۲-۳ ضرب‌کننده‌ی برش‌یافته‌ی ترتیبی ۳۲ بیتی ۹۹
۵-۲-۴ ضرب‌کننده‌ی برش‌یافته‌ی ترتیبی ۶۴ بیتی ۱۰۲
۵-۳ ارزیابی مساحت و سرعت ۱۰۴
۵-۳-۱ ارزیابی تقریبی ۱۰۴
۵-۳-۲ ارزیابی دقیق ۱۰۷
۵-۴ نتیجه‌گیری ۱۰۹

فصل ششم- نتیجه‌گیری و پیشنهادات

۶-۱ نتیجه‌گیری ۱۱۱
۶-۲ پیشنهادات ۱۱۲

منابع و مآخذ ۱۱۳

فهرست شکل‌ها

عنوان **صفحه**

شکل ۲-۱ جمع‌کننده‌ی ۳۲ بیتی CRA ۸

شکل ۲-۲ مدار تمام جمع‌کننده ۹

شکل ۲-۳ زنجیره‌های نقلی و طول آن‌ها ۹

شکل ۲-۴ جمع‌کننده‌ی ۳۲ بیتی تخمینی ۱۱

شکل ۲-۵ جمع‌کننده‌ی ۳۲ بیتی انتخاب‌گر رقم نقلی ۱۲

شکل ۲-۶ یک 32-bit ESTC ۱۳

شکل ۲-۷ مدار کنترلی با استفاده از ۲ بیت ۱۴

شکل ۲-۸ مدار کنترلی ۳ بیتی ۱۵

شکل ۲-۹ شمارنده‌های ۲:۲ و ۳:۲ و کمپرسورهای ۳:۲ ..... ۱۶

شکل ۲-۱۰ کمپرسورهای متداول با استفاده از شمارنده ۳:۲ ..... ۱۶

شکل ۲-۱۱ شماتیک الف) یک شمارنده ۳:۲ ب)یک شمارنده ۴:۲ اشباع‌کننده ۱۷

شکل ۲-۱۲ شمارنده‌های ۶:۲و۷:۲ با استفاده از شمارنده تخمینی ۱۸

شکل ۲-۱۳ ساختار کلی ضرب‌کننده‌های موازی ۱۹

شکل ۲-۱۴ ضرب‌کننده‌ی درختی ۷×۷ با استفاده از درخت CSA ۲۰

شکل ۲-۱۵ ساختار ضرب‌کننده‌ی آرایه‌ای ۵×۵ با استفاده از بلوک‌های FA ۲۰

شکل ۲-۱۶ پیاده‌سازی ضرب ترتیبی با استفاده از روش جابجایی به راست ۲۱

شکل ۲-۱۷ پیاده‌سازی ضرب ترتیبی با استفاده از روش جابجایی به چپ ۲۲

شکل ۲-۱۸ مدار تولید کننده‌ی کد بندی بوث در مبنای ۴ ..... ۲۳

شکل ۲-۱۹ دیاگرام بلوکی مدار ضرب‌کننده‌ی ترتیبی با استفاده از CSA ۲۳

شکل ۲ – ۲۰ قسمت‌های مختلف ماتریس ضرب‌های جزیی برای یک ضرب‌کننده ۸×۸ ..... ۲۵

شکل ۲-۲۱ شمای کلی یک ضرب‌کننده‌ی برش‌یافته با تصحیح متغیر ۲۹

شکل ۲ – ۲۲ پیاده‌سازی روش [۴۰] برای ضرب‌کننده‌ی آرایه‌ای ۵×۵ ..... ۳۰

عنوان **صفحه**

شکل ۲ – ۲۳ یک سلول AO ۳۱

شکل ۲-۲۴ ضرب‌کننده‌ی برش‌یافته برای اعداد بدون علامت $n = 6$ ۳۱

شکل ۲ – ۲۵ ضرب‌کننده‌ی برش‌یافته برای اعداد مکمل دو $n = 6$ ۳۲

شکل ۲-۲۶ پیاده‌سازی سخت‌افزاری ضرب‌کننده‌ی ترتیبی برش‌یافته ۳۶

شکل ۲-۲۷ الف) ضرب‌کننده‌ی علامت‌دار ۸×۸ درختی با استفاده از شمارنده ۳:۲ ب) ضرب‌کننده‌ی تخمینی علامت‌دار ۸×۸ درختی با استفاده از شمارنده ۴:۲ ..... ۳۷

شکل ۲-۲۸ نقشه کارنو ضرب‌کننده‌ی تخمینی۲×۲ ..... ۳۸

شکل ۲-۲۹ مدار ضرب‌کننده‌ی تخمینی۲×۲ ..... ۳۹

شکل ۲-۳۰ مدار ضرب‌کننده‌ی دقیق۲×۲ ..... ۳۹

شکل ۲-۳۱ طراحی ضرب‌کننده با ابعاد بزرگ با استفاده از بلوک‌های کوچک‌تر ۳۹

شکل ۳-۱ جمع یک ضرب‌کننده ترتیبی ۴ بیتی به صورت نمایش نقطه‌ای ۴۳

شکل ۳-۲ ماتریس ضرب‌های جزیی دو عملوند n بیتی A و B ۴۴

شکل ۳-۳ ماتریس ضرب‌های جزیی مربوط به ضرب‌کننده‌ی n بیتی برش‌یافته ۴۶

شکل ۳-۴ مدار سخت‌افزاری روش پیشنهادی ۴۹

شکل ۳-۵ مقداردهی اولیه ثبات EX ۵۰

شکل ۳-۶ نمایش نقطه‌ای ضرب‌کننده‌ی n بیتی به همراه وزن بیت‌های جواب نهایی ۵۳

شکل ۳-۷ عملیات جمع مربوط به سیکل iام ماتریس ضرب‌های جزیی ۵۴

شکل ۳-۸ پیاده‌سازی سخت‌افزاری ضرب‌کننده‌ی پیشنهادی به همراه مدار جبران خطا ۵۷

شکل ۳-۹ پیاده‌سازی قسمت جبران خطا ۵۹

شکل ۳-۱۰ پیاده‌سازی مدار ضرب‌کننده‌ی پیشنهادی به همراه مدار جبران خطا ۵۹

شکل ۳-۱۱ الف)مسیر بحرانی ضرب‌کننده‌ی دقیق ب)مسیر بحرانی ضرب‌کننده‌ی برش‌یافته ۶۱

شکل ۳-۱۲ الف)مدار جبران خطا با استفاده از $AV_{n-i,i-1}$ ب) مدار جبران خطا با استفاده از $pp_{n-i-1,i}$ ... ۶۳

شکل ۳-۱۳ ماتریس ضرب‌های جزیی MSP ۶۴

عنوان **صفحه**

شکل ۳-۱۴ مدار ضرب‌کننده‌ی برش‌یافته‌ی مستقل از LSP در روش بالا به پایین ۶۵
شکل ۳ – ۱۵ ماتریس ضرب‌های جزیی برش‌یافته برای روش پایین به بالا ۶۶
شکل ۳-۱۶ مدار پیشنهادی برای پیاده‌سازی ضرب‌کننده‌ی برش‌یافته به روش پایین به بالا ۶۷
شکل ۳-۱۷ ماتریس ضرب‌های جزیی برش‌یافته روش بالا پایین به صورت برعکس ۶۹
شکل ۳-۱۸ عملیات جمع مربوط به سیکل iام ماتریس ضرب‌های جزیی در روش پایین به بالا ۶۹
شکل ۳-۱۹ مدار جبران خطا با استفاده از گیت‌های AND..... ۷۰
شکل ۳-۲۰ الف) پیاده‌سازی ضرب‌کننده‌ی ترتیبی برش‌یافته‌ی پایین به بالا به همراه مدار جبران خطا ب) مدار جبران خطا مربوط به روابط ۳-۲۹و ۳-۳۰..... ۷۱
شکل ۳-۲۱ الف) مدار جبران خطا رابطه‌ی ۳-۲۱ ب) مدار جبران خطا رابطه‌ی ۳-۲۲..... ۷۲
شکل ۳-۲۲ مدار ضرب‌کننده‌ی برش‌یافته‌ی مستقل از LSP در روش پایین به بالا ۷۳
شکل ۴-۱ جمع‌کننده‌ی زنجیره‌ای به همراه مسیر بحرانی ۷۶
شکل ۴-۲ ماتریس ضرب‌های جزیی مربوط به LSP..... ۷۷
شکل ۴-۳ عملیات جمع دو عددn بیتی ۸۱
شکل ۵-۱ نمودار ستونی خطای نسبی و مطلق برای ضرب‌کننده XOR_AND ۹۵
شکل ۵-۲ نمودار ستونی خطای نسبی و مطلق برای ضرب‌کننده AND ۹۵
شکل ۵-۳ نمودار ستونی خطای نسبی و مطلق برای ضرب‌کننده ی ۱۶ بیتی با AND_XOR ۹۸
شکل ۵-۴ نمودار ستونی خطای نسبی و مطلق برای ضرب‌کننده ی ۳۲ بیتی با AND_XOR ۱۰۱
شکل ۵-۵ نمودار ستونی خطای نسبی و مطلق برای ضرب‌کننده ی ۶۴ بیتی با AND_XOR ۱۰۴
شکل ۵-۶ تصویر از گزارش زمانی یک جمع‌کننده‌ی ۸ بیتی ۱۰۸

فهرست جدول‌ها

عنوان **صفحه**

جدول ۲-۱ جدول کارنو مربوط به شمارنده‌های اشباع‌کننده و بازتابی ۴:۲ ..... ۱۷

جدول ۳-۱ تمامی حالات ممکن برای محاسبه‌ی $C_{n-i-1,i}$ ۵۵

جدول ۳-۲ تمامی حالات ممکن برای محاسبه‌ی $C_{n-i-1,i}$ ۶۲

جدول ۳-۳ تمامی حالات ممکن برای محاسبه‌ی $C_{i,n-i-1}$ ۷۰

جدول ۴-۱ تمامی حالات ممکن برای محاسبه‌ی $C_{n-i-1,i}$ ۷۸

جدول ۴-۲ مقادیر عددی احتمال بروز خطا در ضرب‌کننده‌ی ۸ بیتی ۸۵

جدول ۵-۱ نام‌گذاری اختصاری ضرب‌کننده‌های فصل ۳ ..... ۸۹

جدول ۵-۲ اختلاف بین روش دقیق و ضرب‌کننده‌های جدول ۵-۱(۸ بیتی) ۹۰

جدول ۵-۳ خطای مطلق و نسبی برای $a = 11011110$ و $b = 11110101$ ۹۲

جدول ۵-۴ میانگین خطای نسبی و مطلق برای تمامی اعداد ۸ بیتی ۹۲

جدول ۵-۵ اختلاف بین روش دقیق و ضرب‌کننده‌های جدول ۵-۱(۱۶ بیتی) ۹۶

جدول ۵-۶ میانگین خطای نسبی و مطلق برای ۱۰۰۰۰۰ عدد تصادفی ۱۶ بیتی ۹۷

جدول ۵-۷ اختلاف بین روش دقیق و ضرب‌کننده‌های جدول ۵-۱ (۳۲ بیتی) ۹۹

جدول ۵-۸ میانگین خطای نسبی و مطلق برای ۱۰۰۰۰۰ عدد تصادفی ۳۲ بیتی ۱۰۰

جدول ۵-۹ اختلاف بین روش دقیق و ضرب‌کننده‌های جدول ۵-۱(۶۴ بیتی) ۱۰۲

جدول ۵-۱۰ میانگین خطای نسبی و مطلق برای ۱۰۰۰۰۰ عدد تصادفی ۶۴ بیتی ۱۰۲

جدول ۵-۱۱ تأخیر مدارهای جبران خطا برای روش‌های مختلف بر حسب واحد تأخیر گیت ۱۰۳

جدول ۵-۱۲ تأخیر ضرب‌کننده‌های مختلف بر حسب واحد تأخیر گیت ۱۰۶

جدول ۵-۱۳ سرعت نسبی برای ضرب‌کننده‌ها با ابعاد مختلف ۱۰۶

جدول ۵-۱۴ تاخیر مربوط به عناصر تشکیل‌دهنده‌ی مسیر بحرانی بر حسب نانوثانیه ۱۰۹

جدول ۵-۱۵ سرعت نسبی برای ضرب‌کننده‌ها با ابعاد مختلف ۱۰۹

فصل اول

مقدمه

۱-۱ معرفی موضوع

امروزه مدارهای دیجیتال در پیاده‌سازی پردازنده‌ها، پردازش سیگنال‌های رقمی[1]، سری فوریه و ... کاربرد بسیار گسترده‌ی پیدا کرده‌اند. از کلیدی‌ترین عناصر تشکیل‌دهنده سیستم‌های دیجیتال واحدهای محاسباتی بوده که افزایش کارایی آن‌ها می‌تواند بر روی عملکرد این سیستم‌ها تأثیر بسزایی داشته باشد [۱]. از این رو زیرشاخه‌ی بسیار مهمی در علم معماری کامپیوتر به این موضوع اختصاص پیدا کرده است که به نام محاسبات کامپیوتری[2] یا محاسبات دیجیتال[3] شناخته می‌شود. محاسبات کامپیوتری به بحث در مورد واحدهای محاسباتی مانند جمع‌کننده و ضرب‌کننده و همچنین سیستم‌های نمایش اعداد، روش‌های تخمین خطا و غیره می‌پردازد و سعی می‌کند تا کارایی این واحدهای محاسباتی را بهبود بخشد.

به خاطر اهمیت این زمینه تحقیقات متعددی در این مورد صورت پذیرفته و همچنین کتاب‌های زیادی در مورد آن نوشته شده است. از اولین کتاب‌ها می‌توان به The Logic of Computer Arithmetic اشاره نمود که

1 Digital Signal Processing(DSP)
2 Computer Arithmetic
3 Digital Arithmetic

توسط Flores در سال ۱۹۶۳ نوشته شده است [۲]. پس از آن نیز Hawng در ۱۹۷۹ کتابی در این زمینه ارائه نمود. از دیگر پیشگامان در این زمینه می‌توان به Koren، Ercegovac و Lang اشاره نمود که کتاب‌ها و تحقیقات متعددی را در این زمینه منتشر کرده‌اند [۳] و [۴]. پروفسور پرهامی نیز کتابی جامع در این زمینه ارائه کرده که چاپ دوم آن در سال ۲۰۰۹ منتشر شده است و مرجع بسیار معتبری برای تدریس در دانشگاه‌های مختلفی در سراسر دنیا می‌باشد [۱].

همان طور که گفته شد واحدهای محاسباتی به دلیل گستردگی کاربردشان نقش کلیدی در سیستم‌های دیجیتال ایفا می‌نمایند، پس اگر بتوان کارایی این واحدها را افزایش داد عملکرد کل سیستم بهبود می‌یابد. یکی از مهم‌ترین فاکتورها در بهبود کارایی یک مدار سرعت آن است و با توجه به این که در یک سیستم دیجیتال معمولاً سرعت واحد محاسباتی کمتر از بقیه واحدها می‌باشد در نتیجه با افزایش سرعت واحدهای محاسباتی سرعت کل سیستم افزایش پیدا می‌نماید. به عنوان مثال جمع‌کننده یکی از عناصر کلیدی واحد محاسبه و منطق[۱] در پردازنده می‌باشد، از آنجا که سرعت جمع‌کننده از بقیه‌ی واحدهای تشکیل‌دهنده‌ی پردازنده کمتر است سرعت پردازنده به تناسب آن کم می‌شود [۵]. از دیگر فاکتورهای مهم در سیستم‌های دیجیتال توان مصرفی آن سیستم می‌باشد. در یک سیستم DSP از ضرب‌کننده‌های فراوانی استفاده شده است، چون این ضرب‌کننده‌ها دارای توان مصرفی قابل توجهی هستند در نتیجه توان مصرفی کل سیستم بالا می‌رود که این مسئله در برخی کاربردها می‌تواند تبعات منفی داشته باشد [۶]. فاکتور دیگر مسئله مساحت می‌باشد، در سیستم‌های دیجیتال یکی از واحدهایی که مساحت زیادی را اشغال می‌نمایند واحدهای محاسباتی هستند. به عنوان مثال در یک سیستم متحرک[۲] یا در یک سیستم جاسازی شده[۳] که یکی از دغدغه‌های اصلی قابل حمل بودن آن‌ها است، مسئله‌ی مساحت می‌تواند مشکل‌ساز باشد [۷].

با توجه به نکات مطرح شده می‌توان چالش‌های اصلی پیش‌رو در محاسبات دیجیتال را در سه مورد خلاصه کرد: سرعت، توان مصرفی و مساحت. توان مصرفی و مساحت هزینه‌های یک سیستم به حساب می‌آیند و باید بین آن‌ها و سرعت مدار یک موازنه وجود داشته باشد. به طور کلی مداری ایده‌آل می‌باشد که دارای سرعت بالا، توان مصرفی و مساحت کمی باشد و از این رو محققان متعددی سعی کرده‌اند که این چالش‌ها را برطرف نمایند. در دهه‌های ابتدایی ظهور مدارهای دیجیتال دغدغه‌ی اصلی طراحان، سرعت پایین این مدارها بوده و تلاش آن‌ها

[1] Arithmetic Logic Unit(ALU)
[2] Mobile System
[3] Embedded System

در راستای افزایش سرعت مدارهای دیجیتال بوده است. از میان واحدهای محاسباتی جمع‌کننده و ضرب‌کننده از اهمیت خاصی برخوردار هستند که تقریباً در تمامی سیستم‌های دیجیتال به کار می‌روند و اکثر توجه طراحان به این دو واحد محاسباتی معطوف می‌باشد. عملیات جمع تقریباً مهم‌ترین عملیات حسابی است که می‌توان بقیه‌ی عملیات‌ها را توسط آن پیاده‌سازی کرد و به همین دلیل جمع‌کننده‌ها مورد توجه فراوانی قرار گرفته‌اند.

جمع‌کننده زنجیره‌ای نقلی[1] ساده‌ترین نوع جمع‌کننده است که مشکل اصلی آن انتشار رقم نقلی و در نتیجه سرعت آن پایین می‌باشد. به همین دلیل جمع‌کننده‌ی پیش‌بینی‌گر رقم نقلی[2] در سال ۱۹۵۸ ارائه شد که این مشکل را برطرف می‌نماید [۸]. از نمونه‌های دیگر می‌توان به carry select adder، carry skip adder و prefix adder اشاره نمود که هدف اصلی آن‌ها افزایش سرعت جمع‌کننده بوده است اما باعث شده است که مساحت و توان مصرفی (هزینه) این مدارها افزایش پیدا کند [۱].

ضرب‌کننده یکی از پرکاربردترین واحدهای محاسباتی به خصوص در DSP می‌باشد [۹-۱۴]، ابتدایی‌ترین الگوریتم ضرب روش جابجایی/جمع[3] می‌باشد که روشی ترتیبی بوده و در ضرب دو عدد n بیتی عملیات ضرب n سیکل به طول می‌انجامد [۱]. پیاده‌سازی این الگوریتم نشان می‌دهد که مساحت و توان مصرفی کم بوده ولی سرعت در حد مطلوبی نمی‌باشد، از این رو ضرب‌کننده‌ی موازی ارائه شد که مشکل سرعت نوع ترتیبی را برطرف می‌نمود. اولین نوع این روش توسط پروفسور C. S. Wallace در سال ۱۹۶۴ ارائه گردید. این روش سرعت جمع حاصل‌ضرب‌های جزیی را در ضرب‌کننده‌ی موازی بهبود می‌دهد [۱۵]. پس از آن Luigi Dadda این روش را در ۱۹۸۳بهبود داد[۱۶]. این گروه ضرب‌کننده‌ها، به ضرب‌کننده‌های درختی نیز معروفند. همچنین نمایش اعداد در مبناهای بالاتر منجر به کاهش ارقام عددهای نمایش داده شده در آن مبنا می‌باشد. با علم به این موضوع، ضرب‌کننده‌هایی ارائه شده‌اند که عملیات ضرب را در مبناهای بالا مانند ۴ یا ۸ انجام می‌دهند و باعث بالاتر رفتن سرعت ضرب‌کننده می‌شوند[۱]. یکی دیگر از مواردی که تحول شگرفی در زمینه بهبود ساختار ضرب‌کننده‌ها ایجاد کرده است، ضرب به روش بوث است که توسط Andrew Booth ارائه شد. این روش با یک کد بندی جدید، سرعت ضرب‌کننده را زیاد می‌کند[۱۷]. از دیگر ایده‌های مطرح در این زمینه، ضرب در اعداد ثابت است که کاربردهای فراوانی از جمله در فیلتر پاسخ ضربه محدود[4]، پردازش سیگنال، پردازش تصویر، کنترل و ارتباط داده دارند. با توجه به ثابت بودن مضروب یا مضروب‌فیه الگوریتم‌های مختلفی در این

[1] Carry Ripple Adder(CRA)
[2] Carry Lookahead Adder(CLA)
[3] Shift/Add
[4] Finite Impulse Response(FIR)

زمینه ارائه شده است که سرعت ضرب‌کننده‌ها را بهبود می‌بخشند [۱۲] و [۱۸]. عملیات ضرب در اصل از حاصل‌جمع ضرب‌های جزیی بدست می‌آید در نتیجه مشکل انتشار رقم نقلی که در جمع وجود دارد به ضرب‌کننده نیز سرایت می‌نماید. به همین دلیل روش‌هایی برای از بین بردن مشکل رقم نقلی موجود است که مهم‌ترین آن‌ها پیاده‌سازی ضرب با استفاده از جمع‌کننده‌ی ذخیره‌کننده رقم نقلی[1] است. در این جمع‌کننده، رقم نقلی دیگر به مکان بعد منتقل نشده و در همان مکان ذخیره می‌شود. در انتها، پس از پایان عمل جمع دوباره حاصل این جمع‌کننده توسط یک جمع‌کننده‌ی زنجیره‌ای نقلی به حالت عادی باز می‌گردد، این ضرب‌کننده از نوع ترتیبی می‌باشد [۱].

با توجه به انواع مختلف ضرب‌کننده و جمع‌کننده که در فوق مطرح شد می‌توان به این نتیجه رسید که بهبود یکی از فاکتورهای مزبور باعث افزایش مشکلات بقیه فاکتورها می‌باشد، مثلاً در ضرب‌کننده‌های موازی علیرغم افزایش سرعت، مساحت و توان مصرفی آن نیز بالا می‌باشد که در بعضی سیستم‌های دیجیتال می‌تواند تبعات منفی داشته باشد. اخیراً ایده‌ای مشهور به محاسبات تخمینی[2] مطرح شده که تلاشی برای یافتن یک نقطه بهینه در تبادل[3] مابین سه ویژگی فوق است. این نوع محاسبات که یکی از جدیدترین شاخه‌های محاسبات کامپیوتری محسوب می‌شود با حذف و/یا دستکاری بخشی از مدارهای محاسباتی و در نتیجه با به کنار گذاشتن برخی از محاسبات میانی پیاده‌سازی می‌گردد [۲۰]. حذف این محاسبات می‌تواند باعث افزایش سرعت، کاهش توان مصرفی و کاهش سطح سیلیکون گردیده، هر چند که از سوی دیگر، منجر به کاهش دقت محاسباتی مدارهای مزبور می‌شود. ویژگی‌های مثبت این روش به خوبی با نیازهای برخی سیستم‌های دیجیتال همچون DSP، جاسازی شده و موبایل هم‌خوانی دارد به شرطی که این ابزارها بتوانند کاهش دقت و بروز خطای محدود در محاسبات را تحمل نمایند.

محاسبات تخمینی همچنین می‌تواند از دیدگاهی متفاوت مورد توجه قرار گیرد. ممکن است پس از آزمون نهایی در کارخانه معلوم شود که تراشه‌های تولید شده صد درصد سالم نیستند و خطایی تولید می‌نمایند. این مسئله در دهه‌ی گذشته با برگشت دادن و نابود کردن تراشه‌های معیوب حل می‌شد اما محققان دریافتند که بعضی سیستم‌های دیجیتال مثل DSPها می‌توانند از این تراشه‌های خراب استفاده کنند. این موضوع باعث شد تا طراحان به این نتیجه برسند که اگر می‌شود از تراشه‌های معیوب استفاده نمود، چرا نتوانیم مدارهایی طراحی کنیم که به

[1] Carry Save Adder(CSA)
[2] Estimating Arithmetic
[3] Trade off

صورت عمدی ناقص بوده و خطا تولید کنند و در عوض سرعت بالاتر، سطح سیلیکون و توان مصرفی کمتری داشته باشند [۲۱].

در میان واحدهای محاسباتی، جمع‌کننده‌ها و ضرب‌کننده‌ها را با استفاده از تکنیک محاسبات تخمینی پیاده‌سازی کرده‌اند که در این زمینه تحقیقات متعددی انجام شده و نتایج قابل توجهی به دست آمده است. این تکنیک که تقریباً از دهه‌ی ۹۰ میلادی شکل گرفته، مورد استقبال فراوانی در بین طراحان مدارهای دیجیتال قرار گرفته است. ضرب‌کننده‌های برش‌یافته‌ی[1] موازی اولین نمونه از پیاده‌سازی این تکنیک بود که در ۱۹۹۲ ارائه گردید [۲۲] و تاکنون مورد استقبال شایان توجهی قرار گرفته است. مبنای این روش بر حذف قسمت کم‌ارزش حاصل‌ضرب استوار است، چون در بسیاری از کاربردهای دیجیتال نیاز است که مساحت تراشه کم باشد در نتیجه می‌توان از $2n$ بیت نتیجه (نتیجه‌ی ضرب دو عدد n بیتی برابر $2n$ بیت است)، تنها n بیت پرارزش را نگه داشت و n بیت کم‌ارزش را حذف کرد. بنابراین مدار ضرب‌کننده به صورتی طراحی می‌شود که قسمت کم‌ارزش را محاسبه نکند. ولی حذف n بیت کم‌ارزش، باعث به وجود آمدن خطا می‌شود و ممکن است این خطا به اندازه‌ای باشد که قابل صرف‌نظر کردن نباشد. از این‌رو محققان روش‌های متنوعی را برای برطرف کردن این خطا پیشنهاد کرده‌اند که تاکنون نیز این تحقیقات در حال بررسی و انجام است [۲۳-۲۸].

نمونه‌ی دیگری از ضرب‌کننده‌ی تخمینی موازی ارائه شده که از بلوک‌های ضرب‌کننده‌ی تخمینی ۲×۲ ساخته شده است. این ضرب‌کننده دارای دو خروجی است درحالی‌که ضرب‌کننده‌ی ۲×۲ استاندارد سه خروجی دارد. حال می‌توان با بلوک‌های تخمینی ارائه شده ضرب‌کننده‌های بزرگ‌تری ساخت و در عمل ضرب‌کننده‌های تخمینی با سرعت بالاتری تولید نمود[۲۹].

در مورد ضرب تخمینی ترتیبی به دلیل سرعت پایین آن، تحقیقات متنوعی مانند ضرب موازی انجام نگرفته است و از معدود ضرب‌کننده‌های تخمینی که به صورت ترتیبی عمل می‌کنند می‌توان به ضرب‌کننده‌ی برش‌یافته‌ی تخمینی اشاره کرد که همانند ضرب‌کننده‌ی برش‌یافته‌ی موازی، خروجی n بیتی دارند. این روش الهام گرفته از روش برش‌یافته به صورت موازی است [۳۰].

تا کنون جمع‌کننده‌ها و مشکلات آن‌ها به صورت گسترده‌ای مورد بحث و بررسی قرار گرفته و تکنیک محاسبات تخمینی در موردشان به صورت متنوعی بکار رفته که نمونه‌هایی از آن‌ها در مقالات مختلفی مشاهده می‌شود[۳۱-۳۳]. برای مثال یک جمع‌کننده‌ی تخمینی ارائه شده است که برای شمارش مورد استفاده قرار

[1] Truncated Multiplier

می‌گیرد. در این مدار، تنها در زمانی که هر چهار ورودی یک باشند یک خطا رخ می‌دهد ولی در عوض سرعت مدار و سطح سیلیکون آن بهبود چشمگیری یافته است [۳۱]. در فصل دوم در مورد جمع‌کننده‌های تخمینی مطرح شده بررسی انجام خواهد شد.

کارایی سیستم‌های بهره گرفته از روش تخمینی بیانگر این است که این روش می‌تواند به خاطر حذف قسمتی از مدار و در نتیجه کوتاه‌تر کردن مسیر بحرانی (طولانی‌ترین مسیر از ورودی تا خروجی)، سرعت آن را افزایش داده و همچنین مساحت آن را کاهش داده و توان مصرفی نیز خودبه‌خود کمتر می‌گردد. اما ممکن است دقت مدار تا حدودی کم شود که در مقابل نکات مثبت این روش، قابل صرف‌نظر کردن است و به همین دلیل جمع‌کننده‌ها و ضرب‌کننده‌های مختلفی با این تکنیک پیاده‌سازی شده‌اند. در این میان ضرب ترتیبی مورد توجه کمتری قرار گرفته است در صورتی که در برخی سیستم‌ها توان مصرفی و/یا مساحت، جزء پارامترهای حیاتی بوده و باید تا حد امکان آن‌ها را کاهش داد. بنابراین برای برآورده شدن این فاکتورها در این سیستم‌ها از ضرب ترتیبی استفاده می‌گردد، حال باید برای سرعت کم ضرب ترتیبی نیز چاره‌ای اندیشید.

با توجه به موارد ذکر شده، طراحی یک ضرب‌کننده ترتیبی با سرعت قابل قبول با استفاده از تکنیک محاسبات تخمینی را می‌توان به عنوان هدف اصلی این تحقیق معرفی کرد. همچنین ذکر این نکته نیز حائز اهمیت است که چون مدار طراحی شده از نوع ترتیبی است می‌توان با انجام یک قسمت از ضرب در هر سیکل، خطای به وجود آمده‌ی ناشی از حذف برخی از قسمت‌های مدار را تا حدودی برطرف کرد. در نتیجه مشکل ایجاد خطا که به صورت اجباری و پس از حذف برخی سیگنال‌های میانی بروز کرده است تا حدودی رفع می‌گردد. در اصل سیستم ضرب‌کننده به گونه‌ای طراحی می‌شود که بتواند اندازه‌ی خطا را در هر سیکل پیش بینی کرده و در رفع آن تا حد ممکن موفق باشد.

شبیه‌سازی این تحقیق توسط نرم‌افزارهای شبیه ساز و پیاده‌سازی سخت افزاری این مدار برای مشخص نمودن کارایی و مقایسه ضرب‌کننده‌ی طراحی شده با دیگر ضرب‌کننده‌ها هدف بعدی این تحقیق می‌باشد.

نتیجه‌ای که می‌توان برای این تحقیق متصور بود استفاده از این ضرب‌کننده در سیستم‌هایی مانند موبایل و جاسازی‌شده می‌باشد که ذخیره‌ی توان مصرفی و کاهش مساحت در آن‌ها اولویت دارد و همچنین نیازی به جواب نهایی به صورت کامل (با طول $2n$) ندارند. در نتیجه می‌توان سرعت این سیستم‌ها را که به خاطر استفاده از ضرب‌کننده‌ی ترتیبی کاهش یافته بود را بهبود بخشید.