طراحی و شبیه‌سازی یک واحد حساب و منطق ۶۴×۶۴ بیتی با سرعت کلاک ۲ گیگا هرتز در تکنولوژی ۱۳۰ نانومتر

مریم سیستانی زاده[۱] سید رضا حسینی[۲]

۱- دانش آموخته کارشناسی ارشد- گروه مهندسی برق، واحد خوی، دانشگاه آزاد اسلامی، خوی، ایران

electronic2018.ms@gmail.com

۲- استادیار- گروه مهندسی برق، واحد خوی، دانشگاه آزاد اسلامی، خوی، ایران

hosseini@iaukhoy.ac.ir

چکیده: در این مقاله هدف طراحی یک واحد حساب و منطق ۶۴×۶۴ بیتی با توان، تأخیر پایین و سرعت بالا می باشد. واحد حساب و منطق عملیات محاسباتی نظیر جمع و ضرب را انجام می دهد. جمع کننده ها نقش مهمی در واحد حساب و منطق دارند. برای طراحی جمع کننده، از ترکیب جمع کننده های انتخاب کننده ی نقلی و جمع کننده پیش بینی کننده نقلی و همچنین از مدار "جمع کننده با یک" برای دستیابی به سرعت بالا و سخت افزار کم استفاده شده است. در طراحی ضرب کننده از الگوریتم بوث و از ساختار والاس استفاده شده است. ضرب کننده ارائه شده بر اساس تکنیک خط لوله می باشد. در ساختار والاس از کمپرسورها برای فشرده سازی حاصلضرب های جزئی استفاده شده است. استفاده از الگوریتم بوث برای تولید حاصلضرب های جزئی، منجر به بهبود سرعت ضرب کننده شده است. تأخیر و توان مصرفی بدست آمده برای جمع کننده ۶۴ بیتی در ولتاژ تغذیه ۱.۳ ولت و فرکانس ۲ گیگا هرتز به ترتیب برابر ۱۱۲ پیکو ثانیه و ۱۲ میلی وات و برای ضرب کننده، تأخیر برابر با ۲۹۱ پیکوثانیه و توان ۹۵۰ میلی وات می باشد. ساختارهای ارائه شده با استفاده از تکنولوژی 130nm CMOS پیاده سازی شده اند.

واژه‌های کلیدی: جمع کننده انتخاب کننده نقلی، جمع کننده پیش بینی کننده نقلی، ضرب کننده، الگوریتم بوث، حاصلضرب جزئی، پایپ لاین، تأخیر، توان مصرفی

نوع مقاله: پژوهشی

تاریخ ارسال مقاله : ۹۷/۰۳/۰۷

تاریخ پذیرش مشروط مقاله: ۹۷/۱۲/۲۵

تاریخ پذیرش مقاله : ۹۸/۰۴/۲۶

نام نویسنده‌ی مسئول: دکتر سید رضا حسینی

نشانی نویسنده‌ی مسئول: ایران – خوی – کیلومتر ۴ جاده خوی سلماس- دانشگاه آزاد اسلامی واحد خوی - دانشکده فنی و مهندسی- گروه مهندسی برق

۱- مقدمه

واحد حساب و منطق[۱]، یک قطعه اساسی از واحد پردازش مرکزی[۲] در کامپیوتر است. واحد حساب و منطق ترکیبی از بلوک هایی است که عملیات منطقی و محاسباتی را انجام می دهد. یک واحد حساب و منطق را می توان به طور گسترده در سه حوزه توان، تأخیر و مساحت بهینه سازی کرد. بلوک حساب، عملیات محاسباتی نظیر جمع، تفریق، ضرب و... را انجام می دهد. جمع کننده نه تنها در عمل جمع بلکه در انجام عملیات محاسباتی دیگری همچون تفریق و ضرب نیز نقش مهمی را دارد. از آنجایی که در ساختار مدار ضرب کننده نیز به یک جمع کننده ی نهایی نیاز داریم، سعی بر این است که جمع کننده ای طراحی کنیم که هر دو نیاز ما را تأمین کند [۲-۱]. با مقایسه ی جمع کننده های متفاوت [۹-۳] برای طراحی جمع کننده از ترکیب جمع کننده های انتخاب کننده ی نقلی[۳] و جمع کننده پیش بینی کننده نقلی[۴] استفاده می شود[۱۰].

به طور کلی پیاده سازی ضرب کننده به سه بخش تقسیم می شود: ۱) تولید حاصلضرب جزئی[۵] ۲)کاهش حاصلضرب های جزئی تولید شده ۳) جمع دو ردیف باقیمانده از حاصلضرب های جزئی برای بدست آوردن حاصلضرب نهایی. روش های متعددی برای این سه مرحله وجود دارد که در سال های اخیر توسعه یافته و باعث بهبود عملکرد ضرب کننده شده است. در این مقاله برای تولید حاصلضرب های جزئی از انکودر بوث اصلاح شده[۶] استفاده می شود که در آن ضرب به صورت علامت دار صورت گرفته و همچنین تعداد ردیف های حاصلضرب های جزئی در ضرب کننده n- بیتی تقریبا به نصف کاهش می یابد [۱۵-۱۱]. در مرحله دوم نیز با استفاده از ساختار های والاس[۷] وکمپرسور[۸] یا شمارنده[۹] تعداد ردیف های حاصلضرب های جزئی ایجاد شده در مرحله اول به دو ردیف کاهش می یابد[۲۲-۱۶]. در نهایت در مرحله سوم دو ردیف باقیمانده از قسمت دوم را با استفاده از جمع کننده های انتخاب کننده نقلی و پیش بینی کننده نقلی، جمع کرده و جواب نهایی ضرب تولید می شود. یکی دیگر از روش های افزایش سرعت و کارایی در پردازنده ها استفاده از تکنیک خط لوله[۱۰]است که در سالهای اخیر در بسیاری از کاربردها (آنالوگ و دیجیتال) استفاده می شود[۲۵-۲۳][۱۶]. در واقع خط لوله، مجموعه ای از عناصر(مراحل) پردازش داده است که بصورت سری به یکدیگر متصل می باشند و ورودی هر عنصر، خروجی عنصر قبلی است که باعث کاهش پیچیدگی طراحی و تأخیر آن می شود. در تکنیک خط لوله، چندین دستور می توانند در یک زمان اجرا شوند.

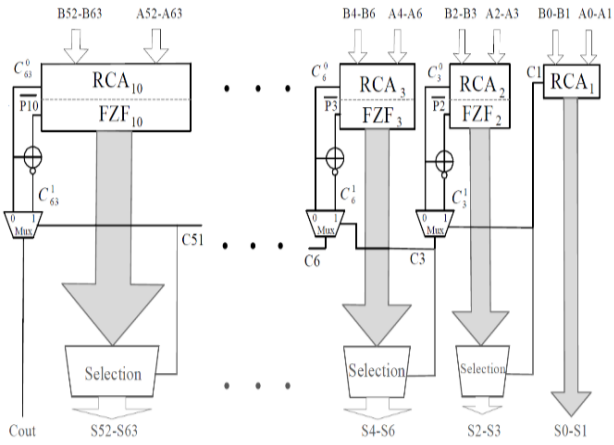
بخش ۲ این مقاله به طراحی واحد حساب و منطق و بلوک های جمع کننده و ضرب کننده می پردازد. در بخش ۳ نتایج شبیه سازی و مقایسه عملکرد جمع کننده و ضرب کننده مورد تحلیل و بررسی قرار خواهد گرفت و در بخش ۴ نتیجه گیری مقاله ارائه می گردد.

۲- طراحی واحد حساب و منطق

همانطور که پیش از این نیز گفته شد واحد حساب و منطق، عملیاتی نظیر جمع، تفریق، ضرب و... را انجام می دهد. این واحد از قسمت های جمع کننده، ضرب کننده، تفریق کننده و ... تشکیل شده است. در این بخش به طراحی جمع کننده و ضرب کننده با توان پایین و تأخیر کم پرداخته خواهد شد. هدف از این مقاله طراحی یک واحد حساب و منطق ۶۴ بیتی در تکنولوژی ۱۳۰ نانومتر CMOS و سرعت کلاک ۲ گیگا هرتز می باشد.

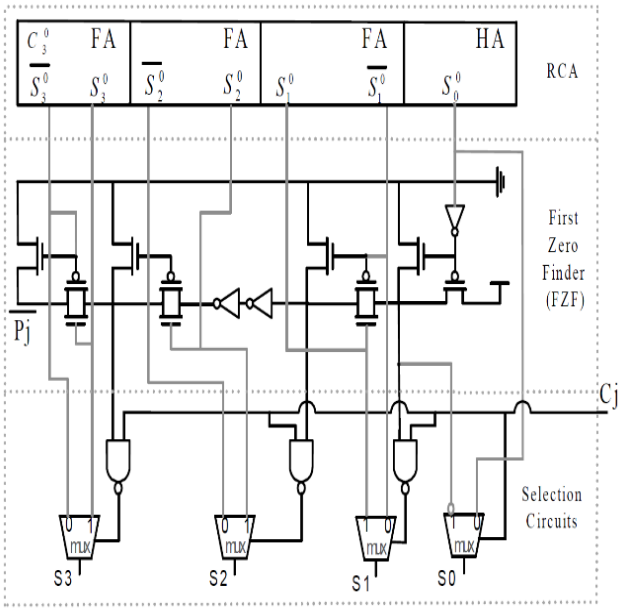
۲-۱- جمع کننده

جمع کننده بخش اساسی واحد حساب و منطق می باشد و نه تنها در جمع بلکه در اجرای بسیاری از عملیات محاسباتی اساسی همچون تفریق و ضرب نقش مهمی ایفا می کند. بدین ترتیب طراحی یک جمع کننده کارآمد برای عملکرد بهتر واحد حساب و منطق مورد نیاز است. از میان جمع کننده های مختلف، جمع کننده ی انتخاب کننده نقلی دارای مزایای بسیاری است. ساختار جمع کننده انتخاب کننده نقلی از بلوک های متفاوت تشکیل شده است که هر بلوک دو جمع را برای دو حالت ممکن بیت نقلی ورودی ۰ و ۱ اجرا می کند . حاصل جمع بدست آمده به یک مالتی پلکسر داده می شود که بسته به بیت نقلی ورودی که از طبقه قبل آمده است، خروجی صحیح را انتخاب می کند. شکل ۱ ساختار جمع کننده ارائه شده در [۹] را نشان می دهد که دارای ترانزیستورهای کمتری نسبت به جمع کننده انتخاب کننده نقلی مرسوم می باشد. با توجه به شکل ۱، عمل جمع در ساختار فوق توسط جمع کننده نقلی موج دار انجام می گیرد که با جایگزینی جمع کننده پیش بینی کننده نقلی بجای جمع کننده نقلی موج دار و ادغام دو جمع کننده پیش بینی کننده نقلی و جمع کننده انتخاب کننده نقلی، سرعت این ساختار افزایش می یابد. همچنین در شکل ۱ بجای عمل جمع دوم از یک مدار " جمع کننده با یک[۱۱] " استفاده شده است که دارای سخت افزار و توان مصرفی کوچکتر می باشد.



شکل(۱): بلوک دیاگرام جمع کننده انتخاب کننده نقلی [۹]

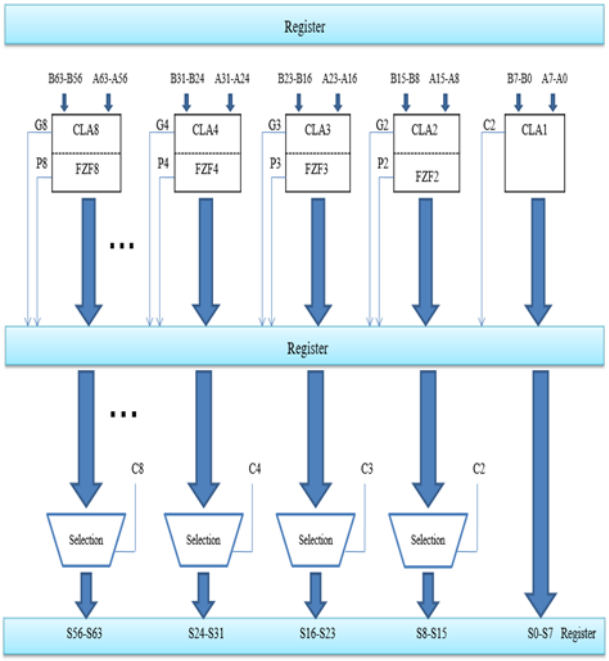
با توجه به شکل های ۱ و ۲، تاخیر جمع کننده نهایی شامل: ۱- تأخیر جمع کننده ی بلوک اول که بیت نقلی ورودی بلوک دوم را تولید می کند ۲- تأخیرِ انتشار بیت نقلی تولید شده در بلوک اول به بلوک آخر ۳- انتخاب بیت جمع نهایی با سیگنال های خروجی مدارهای "جمع کننده با یک" و جمع کننده نقلی موج دار، می باشد. می توان با اعمال تغییرات در بخش هایی از ساختارهای فوق الذکر، مثل ادغام دو جمع کننده پیش بینی کننده نقلی و جمع کننده انتخاب کننده نقلی، سرعت مدار را بهبود داد.



شکل(۲): بلوک ۴ بیت انتخاب کننده نقلی[۹]

شکل ۳ نمای کلی جمع کننده ۶۴ بیتی را نشان می دهد. همانطور که در شکل مشخص است، ۶۴ بیت به ۸ بلوک مساوی (بلوک های ۸ بیتی) تقسیم شده است که هر بلوک یک جمع کننده ی جزئی ۸ بیتی است. در طبقه ی اول این جمع کننده، هر یک از جمع کننده های جزئی با رقم نقلیِ ورودیِ صفر به همراه یک مدار "جمع کننده با یک" بکار برده شده اند. در این جمع کننده، بجای جمع کننده ی نقلی موج دار از یک جمع کننده ی پیش بینی کننده ی نقلی برای سرعت بیشتر استفاده شده است. ورودی های مدار "جمع کننده با یک" به جای اینکه به خروجی های تولید شده در جمع کننده های جزئی وصل شوند، به خروجی های گیت های XOR واقع در جمع کننده های جزئی متصل می شوند که با این کار ورودی های مدار "جمع کننده با یک" زودتر و همزمان اعمال می شوند و سرعت کل جمع کننده بالا می رود. مهمتر اینکه در این جمع کننده از یک مدار دیگر برای تولید بیت نقلی هر بلوک استفاده شده است و بر خلاف روش هایی که بیت نقلی هر بلوک را مستقیماً از ورودی های جمع کننده محاسبه می کنند[۹و۲۶]، سخت افزار کوچکتری دارد. در ادامه برای توضیح ساختار جمع کننده، اجزای هر یک از طبقات با جزئیات شرح داده خواهد شد.

در این جمع کننده، درسیکل اول ورودی ها وارد ۸ بلوک از مدارهای جمع کننده پیش بینی کننده نقلی و جمع کننده انتخاب کننده نقلی می شود و خروجی آن ها به رجیستر می رود و در سیکل دوم با توجه به مقادیری که وارد رجیستر شده است، مدار تولید کننده ی بیت نقلی بلوک ها و مدار انتخاب کننده ی نتیجه ی جمع، نتیجه نهایی را بر روی رجیستر قرار می دهد.



شکل (۳): نمای کلی جمع کننده ی ۶۴ بیتی

۲-۱-۱- جمع کننده ی پیش بینی کننده ی نقلی

جمع کننده پیش بینی کننده نقلی با ترکیبی از منطق C-CMOS و گیت انتقال[۱۲] طوری طراحی شده که علاوه بر سرعت بالا و توان مصرفی کم، قابلیت تحقق در سطح چیپ کوچک و چینش منظم را داشته باشد[۱۰]. در شکل ۴ و روابط زیر نحوه ی تولید بیت نقلی و متمم بیت نقلی نشان داده شده است:

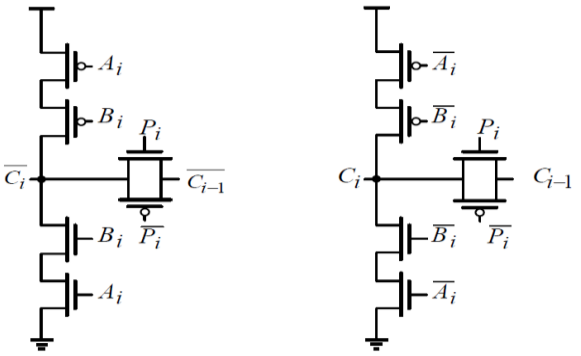
$$C_i = A_i.B_i + P_iC_{i-1} \qquad (۱)$$

$$\overline{C_i} = \overline{A_i}.\overline{B_i} + P_i\overline{C_{i-1}} \qquad (۲)$$

جمله ی اول این دو رابطه را می توان به کمک منطق C-CMOS (توسط دو NMOS سری و دو PMOS سری) و جمله ی دوم این دو رابطه را توسط منطق گیت انتقال تحقق داد.

با توجه به این که هر بلوک جمع کننده ی جزئی در ساختار استفاده شده ۸ بیتی است، سری کردن تعداد زیاد از مدارهای شکل۴-الف و یا شکل ۴-ب باعث می شود چندین گیت انتقال پشت سرهم قرار گیرد و چون گیت انتقال قدرت درایو کردن بالایی ندارد، بایستی از اینورتر در میان طبقات به عنوان بافر استفاده گردد.

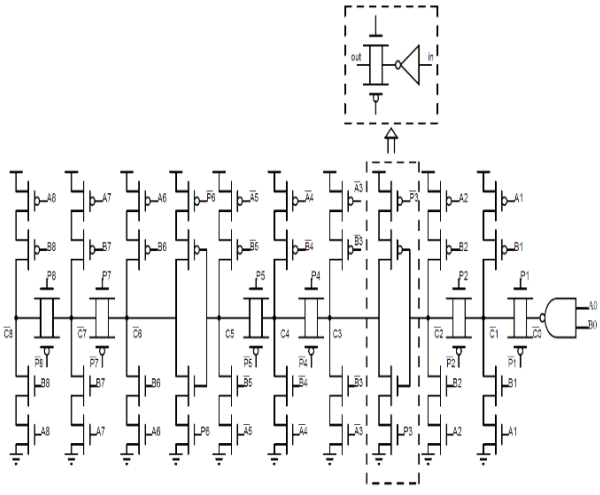
در نتیجه پس از هر سه بیت یک اینورتر قرار داده و پس از آن مدار متمم مدارِ قبل از اینورتر را قرار می دهیم. همانطور که در شکل ۵ مشخص است می توان بجای سری کردن اینورتر و گیت انتقال از ساختار نقطه چین مشخص شده در شکل ۵ استفاده کرد که سبب چینش منظم تر و تأخیر کمتر می شود. مدار نشان داده شده در شکل ۵ در تکنولوژی CMOS 130nm با $W_{p\text{-}ch}$=1µm و $W_{n\text{-}ch}$=3µm پیاده سازی شده است.



شکل (۴): الف) سلول تولید کننده ی متمم بیت نقلی($\overline{C_{in}}$) و ب) سلول تولید کننده ی بیت نقلی (C_{in})[۱۰]

۲-۱-۲- مدار "جمع کننده با یک"

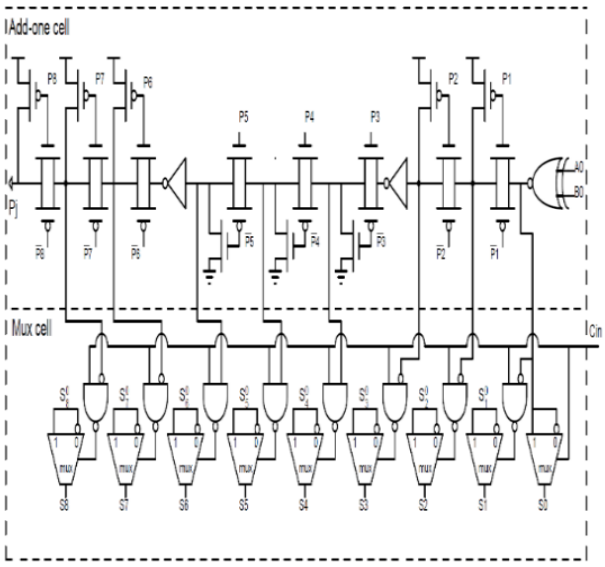
در مدار جمع کننده انتخاب کننده نقلی دو بلوک جمع کننده بصورت همزمان عمل می کنند، بطوریکه در یکی بیت نقلی ورودی صفر و در دیگری بیت نقلی ورودی یک است. بنابراین اختلاف حاصل این دو بلوک، ۱ خواهد بود. با توجه به این نکته می توان نتایج بلوک اول (ورودی بیت نقلی صفر است) را با ۱ جمع کرد و این عمل جمع توسط مدار "جمع کننده با یک" انجام می گیرد که سخت افزار و توان مصرفی کوچکتری دارد. این موضوع اساس مدار "جمع کننده با یک" است که اولین بار توسط Chang ارائه شد و سپس راهکارهایی برای بهبود این مدار ارائه گردید [۲۸-۲۷][۹].



شکل (۵): مدار جمع کننده پیش بینی کننده نقلی ۸-بیتی

شکل ۶ مدار "جمع کننده با یک" که در این مقاله استفاده شده است را نشان می دهد. جمع کننده در تکنولوژی CMOS 130nm با $W_{p\text{-}ch}$=1µm و $W_{n\text{-}ch}$=3µm پیاده سازی شده است. ورودی های مدار "جمع کننده با یک" باید بجای اتصال به خروجی های جمع کننده، به ($A_i \oplus B_i$) متصل شوند زیرا در جمع کننده های جزئی $C_{in}=0$ می باشد و تا زمانی که $A_i \oplus B_i = 1$ باشد، $S_i^0 = 1$ (خروجی جمع کننده به ازای $C_{in}=0$) خواهد بود و اولین $S_k^0 = 0$ زمانی رخ می دهد که $B_k \oplus A_k = 0$ باشد. بعد از این که اولین صفر رخ می دهد، خروجی مدار "جمع کننده با یک" برای بیت k و بیت های بعدی یک خواهد بود.

با توجه به شباهتی که بین مدار جمع کننده پیش بینی کننده نقلی و مدار "جمع کننده با یک" ارائه شده وجود دارد، می توان انتظار داشت که ماکزیمم تأخیر هر دو مدار یکی باشد.



شکل(۶): مدار های "جمع کننده با یک" و انتخاب کننده ی نهایی

۲-۱-۳- مدار تولید کننده ی بیت نقلی بلوک ها

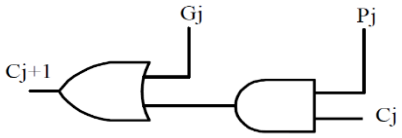
نحوه پخش بیت نقلی بین بلوک ها نقش مهمی در تعیین سرعت جمع کننده انتخاب کننده نقلی دارد. شکل ۷ مدار تولید کننده ی بیت نقلی بلوک ها را نشان می دهد که بیت نقلی ورودی هر بلوک توسط بلوک قبلی طبق رابطه ی زیر تولید می شود:

$$C_{j+1} = G_j + P_j C_j \qquad (۳)$$

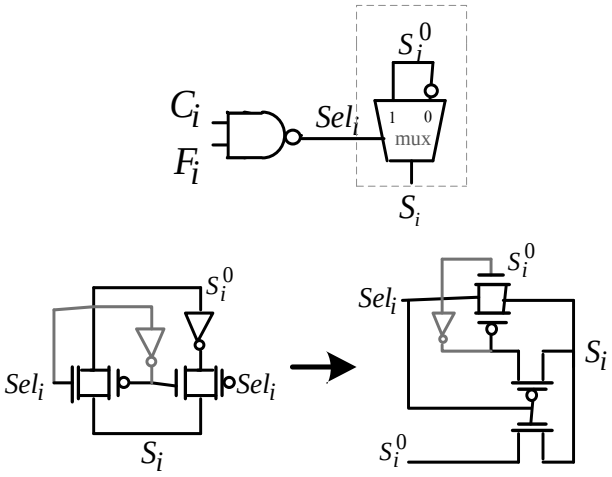
که j شماره ی هر بلوک، C_j بیت نقلی ورودی بلوک j-ام، G_j آخرین بیت نقلی خروجی تولید شده توسط مدار جمع کننده پیش بینی کننده نقلی بلوک j-ام و P_j آخرین خروجی مدار "جمع کننده با یک" بلوک j-ام است.

پس از تولید بیت نقلی ورودی هر بلوک، نتیجه ی نهایی عملیات جمع توسط بیت نقلی تولید شده، انتخاب می شود. شکل ۸ مدار ارائه شده برای این قسمت را نشان می دهد[۱۰-۹]. با توجه به شکل ۸ چون دو

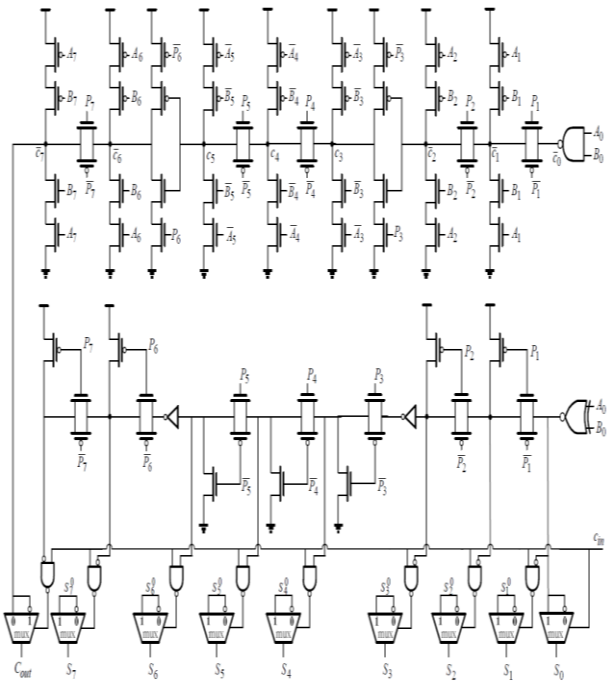
ورودی مالتی پلکسر قرینه ی همدیگرند، می توان بجای مالتی پلکسر از گیت XOR استفاده کرد که سبب کاهش توان مصرفی و تعداد ترانزیستورها می شود.



شکل (۷): مدار تولید کننده ی بیت نقلی بلوک ها در سطح گیت منطق [۱۰]



شکل (۸) : مدار انتخاب کننده ی نهایی[۱۰]

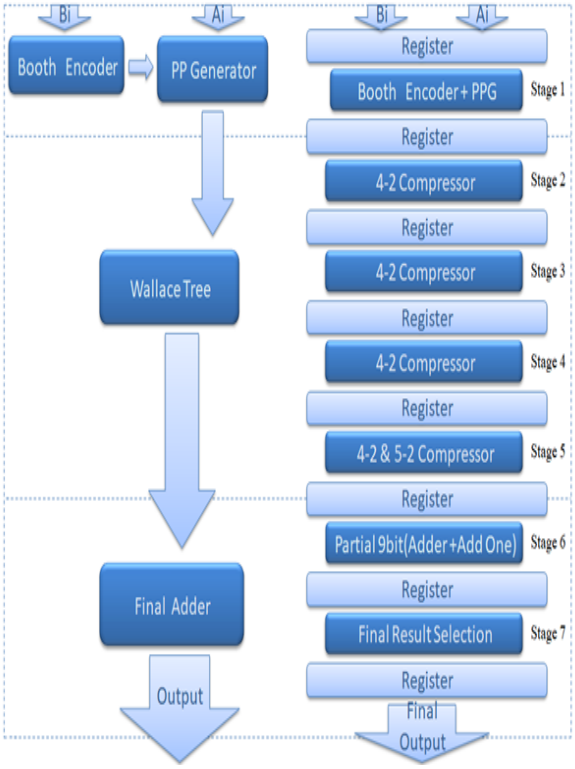


شکل (۹): ساختار کامل جمع کننده ۸-بیتی

شکل ۹ مدار طراحی شده نهایی برای جمع کننده را نشان می دهد که در آن از ترکیب دو ساختار جمع کننده پیش بینی کننده نقلی و جمع کننده انتخاب کننده، به کار رفته است. استفاده از ترکیب منطق C-CMOS و گیت انتقال و نیز استفاده از گیت XOR در مدار انتخاب کننده نهایی، سرعت بالا، توان مصرفی کم، قابلیت تحقق در سطح چیپ کوچک، چینش منظم و کاهش تعداد ترانزیستور ها را فراهم نموده است.

۲-۲- ضرب کننده

ضرب کننده از بخش های مهم در طراحی یک واحد حساب و منطق می باشد. همچنین الگوریتم های مختلفی با ویژگی های متفاوت در زمینه ی سرعت، پیچیدگی مداری، مساحت و توان مصرفی ارائه شده است. امروزه، نیاز به پردازش های سریعتر و کاراتر بیش از پیش اهمیت پیدا کرده است. در چنین پردازنده هایی، اغلب از ضرب کننده های موازی استفاده می شود که نسبت به الگوریتم های دیگر سخت افزار بزرگتر و توان مصرفی بیشتری دارند ولیدارای سرعت بالاتری هستند. شکل ۱۰ ساختار کلی ضرب کننده طراحی شده را نشان می دهد. همانطور که در این شکل مشخص است، این ضرب کننده شامل هفت سیکل زمانی است. از این هفت سیکل، یک سیکل برای تولید حاصلضرب های جزئی می باشد که با استفاده از انکودر بوث اصلاح شده، صورت می گیرد. در ساختار ارائه شده، ضرب به صورت علامت دار صورت می گیرد و همچنین تعداد ردیف ها در یک ضرب کننده - n بیتی را به $n/2+1$ کاهش می دهد (سیکل اول). چهار سیکل برای درخت والاس لحاظ می شود تا حاصلضرب های جزئی با ارزش یکسان را بصورت عمودی جمع کند. این مساله باعث کاهش تعداد ردیف های ایجاد شده، به دو ردیف می شود. این قسمت عمدتا توسط کمپرسورها صورت می گیرد (سیکل های دوم، سوم و چهارم و پنجم). دو سیکل برای جمع کننده ی نهایی می باشد که دو ردیف باقی مانده از قسمت قبل را جمع کرده و جواب نهایی ضرب را تولید می کند. جمع کننده های پیش بینی کننده نقلی و انتخاب کننده نقلی جمع کننده های متعارفی هستند که در این قسمت به کار گرفته می شوند (سیکل های ششم و هفتم). بهبود در هر یک از این سه قسمت اصلی می تواند قابلیت کلی ضرب کننده را بالا ببرد. علاوه بر این، یکی از روش های افزایش سرعت و کارایی در پردازنده ها استفاده از تکنیک خط لوله است. برای این منظور سعی شده با اعمال تغییرات در برخی ساختارهای قبلی و ارائه ساختار جدید، توازن در تأخیر طبقات برقرار شود تا ماکزیمم بهره برداری از تکنیک خط لوله صورت گیرد.. در این بخش به توضیح مراحل ضرب کننده استفاده شده در این مقاله خواهیم پرداخت.



شکل (۱۰) : ساختار کلی ضرب کننده ۶۴ بیتی

۲-۲-۱- تولید حاصلضرب های جزئی

اولین مرحله درفرآیند ضرب، تولید حاصلضرب های جزئی می باشد. در این مقاله، برای تولید حاصلضرب های جزئی از انکودر بوث اصلاح شده به دلیل کارکرد ساده در ایجاد و دوبرابر کردن ضرب شونده[۱۳] استفاده شده است. در بلوک انکودر بوث، تولید حاصلضرب های جزئی در تعیین میزان توان مصرفی و مساحت نقش مهمی دارد.

چون ساختار ضرب کننده ی مورد نظر از نوع خط لوله است، هریک از طبقات ضرب کننده باید سرعت بالایی داشته باشند. به طور معمول الگوریتم بوث اصلاح شده j- امین حاصلضرب جزئی در ردیف i-ام را مطابق با رابطه زیر محاسبه می کند:

$$PP_{ij} = (A_j.One_i + A_{j-1}.Two_i) \oplus Neg_i \qquad (۴)$$

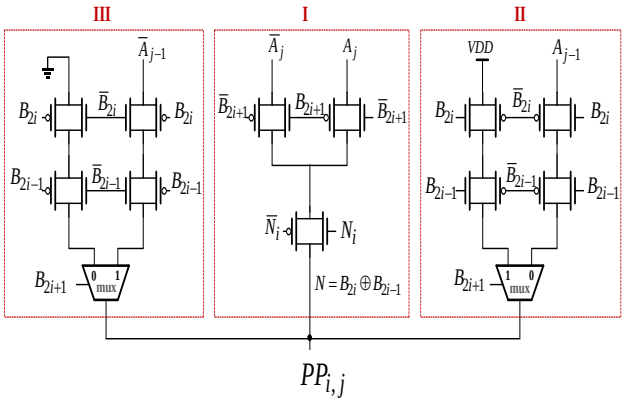
در عبارت فوق A_j و A_{j-1} ورودی های ضرب شونده به ترتیب برابر با 2^j و 2^{j-1} می باشند. One_i و Two_i تعیین می کنند که ضرب شونده باید دو برابر شود یا خیر. Neg_i رقمی است که مشخص می کند ضرب شونده باید معکوس شود. جدول ۱، جدول درستی برای مدار طراحی شده را نشان می دهد که با ایجاد تغییراتی در رابطه فوق و با توجه به جدول ارائه شده در [۲۹] بدست آمده است.

جدول (۱): جدول درستی انکودر بوث اصلاح شده

	B_{2i+1}	B_{2i}	B_{2i-1}	OP	D_i	N_i	T_i	M	Neg_i
III	0	0	0	+0	0	0	1	0	0
I	0	0	1	+A	1	1	0	0	0
I	0	1	0	+A	1	1	0	0	0
II	0	1	1	+2A	-2	0	0	1	0
III	1	0	0	-2A	-2	0	1	0	1
I	1	0	1	-A	-1	1	0	0	1
I	1	1	0	-A	-1	1	0	0	1
II	1	1	1	-0	-0	0	0	1	1

شکل ۱۱ مدار بوث ارائه شده را نشان می دهد. برای طراحی مدار، سه حالت متمایز که با اعداد رومی مشخص شده اند، در نظر گرفته شده است. در هر لحظه فقط یک شاخه روشن می باشد. با توجه به جدول ۱ می توان گفت: قسمت اول(I)، شامل ۴ حالت است که در آنها $B_{2i} \neq B_{2i-1}$ و $N_i = 1$ است، در این وضعیت مقدار D_i برابر با 1 و یا 1- است. بنابراین وقتی $N_i = 1$ است، اگر $Neg_i = 0$ (بیت علامت و برابر با B_{2i+1}) باشد، بیت j-ام ضرب شونده و اگر $Neg_i = 1$ باشد، مکمل بیت j-ام از ضرب شونده به خروجی می رود. با استفاده از چند گیت انتقال، این مسیر از تولید حاصلضرب های جزئی تحقق می یابد. تأخیری که برای این شاخه داریم برابر با تأخیر دو گیت انتقال است. برای این قسمت داریم:

$$N_i = B_{2i} \oplus B_{2i-1} \qquad (۵)$$



شکل (۱۱): مدار مربوط به الگوریتم بوث ارائه شده که در تکنولوژی CMOS 130nm با $W_{p\text{-}ch}$=4.8 µm و $W_{n\text{-}ch}$=2.7 µm پیاده سازی شده است

در قسمت دوم (II)، $B_{2i} = B_{2i-1} = 1$ است که با پارامتر M_i مشخص شده است و مقدار D_i برابر 0- یا 2 است . برای $D_i = -0$، با توجه به مقدار Neg_i که برابر 1 است، خروجی باید 1 شود و برای $D_i = 2$، ضرب شونده دو برابر می شود. برای این کار برای دو برابر کردن ضرب شونده، بیت A_{j-1} را به خروجی ارسال می شود (یک بیت شیفت به چپ بیت های ضرب شونده). در قسمت سوم (III) $B_{2i} = B_{2i-1} = 0$ می باشد که با پارامتر T_i مشخص شده است و مقدار

D_i برابر با 0 و یا 2- است. اگر $D_i = 0$ باشد، با توجه به $Neg_i = 0$، مقدار 0 به خروجی می رود و اگر $D_i = -2$ باشد، دو کار باید انجام گیرد: ابتدا دو برابر کردن و دیگری با توجه به $Neg_i = 1$، مکمل کردن.

شکل ۱۲ نمودار نقطه ای حاصلضرب های جزئی ضرب کننده بوث بکار رفته در این مقاله را نشان می دهد. بیت های حاصلضرب جزئی در ستون هایی برای جمع شدن، جهت تشکیل حاصلضرب نهایی مرتب شده اند. هر نقطه نمایش دهنده ی یک بیت حاصلضرب است. با توجه به شکل، بیت علامت با هر ردیف حاصلضرب جزئی جمع می شود. این کار، عمل مکمل-۲ را در الگوریتم بوث کامل می کند. جمع کردن بیت علامت در ردیف آخر مثل این است که یک ردیف اضافی به ردیف های حاصلضرب جزئی افزوده شود. یک راه برای رسیدن به سرعت بالاتر و آرایش بهتر در نمودار نقطه ای، منظم تر کردن حاصلضرب های جزئی می باشد. برای رسیدن به این هدف، کم ارزشترین بیت در هر ردیف از حاصلضرب های جزئی با بیت علامت که زیر کم ارزشترین بیت قرار گرفته، ادغام می شود (شکل ۱۳-الف). بنابراین بیت کم ارزش جدید به نام K_i و یک بیت جدید به نام H_i در هر ردیف با آرایش مکانی جدید و به جای بیت کم ارزش اصلی و بیت علامت، جایگزین می شود(شکل ۱۳- ب). می توان مشاهده کرد که K_i، حاصل XOR دو بیت نشان داده شده درشکل ۱۳- الف و H_i، حاصل AND این دو بیت می باشد. درنتیجه، تعداد عملیات جمع و تعداد بیت های باقیمانده برای عمل جمع نهایی در آخرین سیکل، کاهش می یابد.

معادلات منطقی K_i و H_i در شکل ۱۲ از معادلات زیر برگرفته شده است[۱۲][۳۰-۳۱]:

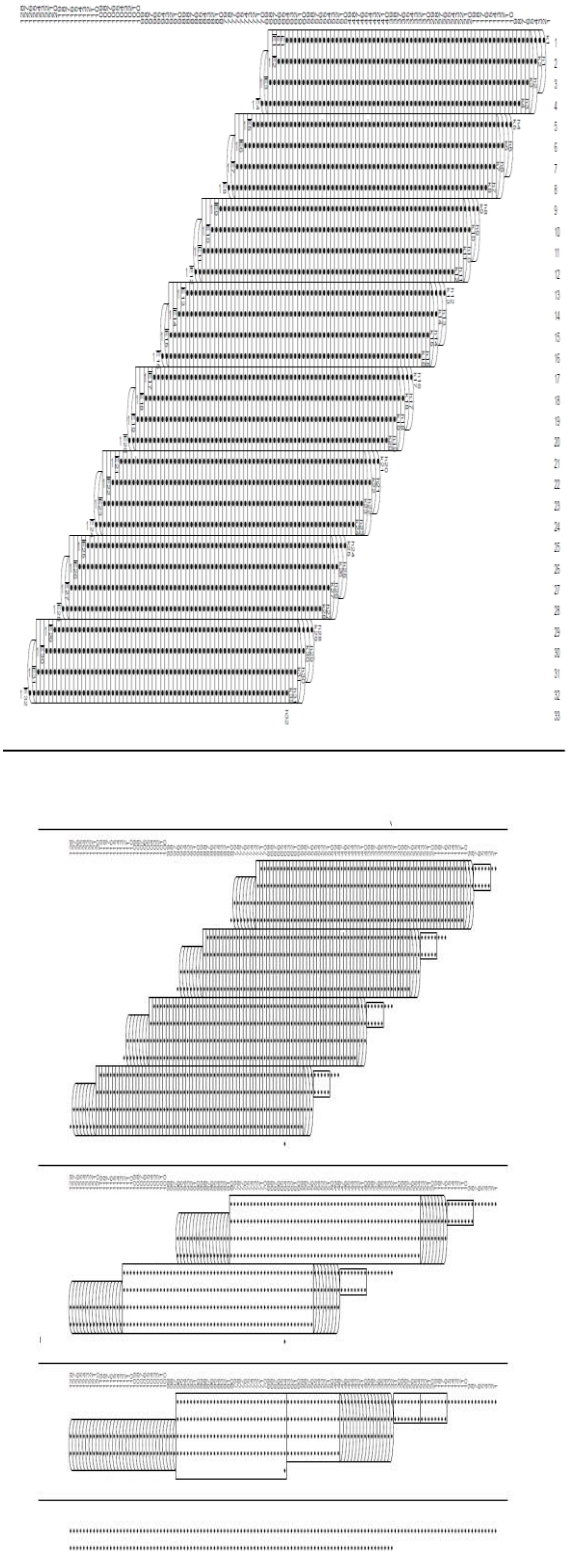
$$k_i = A_0.N_i \qquad (۶)$$

$$H_i = Neg_i.(\overline{N_i} + \overline{A_0}) = \overline{\overline{Neg_i} + N_i.A_0} \qquad (۷)$$

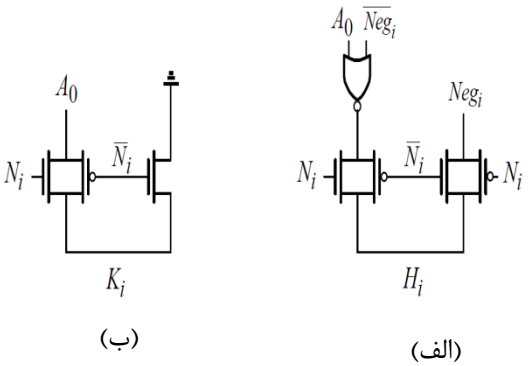
شکل ۱۳-الف و ۱۳-ب به ترتیب مدارهای تولید K_i و H_i را نشان می دهد.

۲-۲-۲- مدل های فشرده سازی

برای انجام عملیات جمع حاصلضرب های جزئی و کاهش طبقات آن در ضرب کننده، از ساختار والاس استفاده شده است. ساختار والاس به دلیل انجام موازی عمل جمع دارای سرعت بالایی می باشد و تأخیر کاهش حاصلضرب های جزئی آن بصورت لگاریتمی با تعداد بیت ضرب کننده متناسب است.



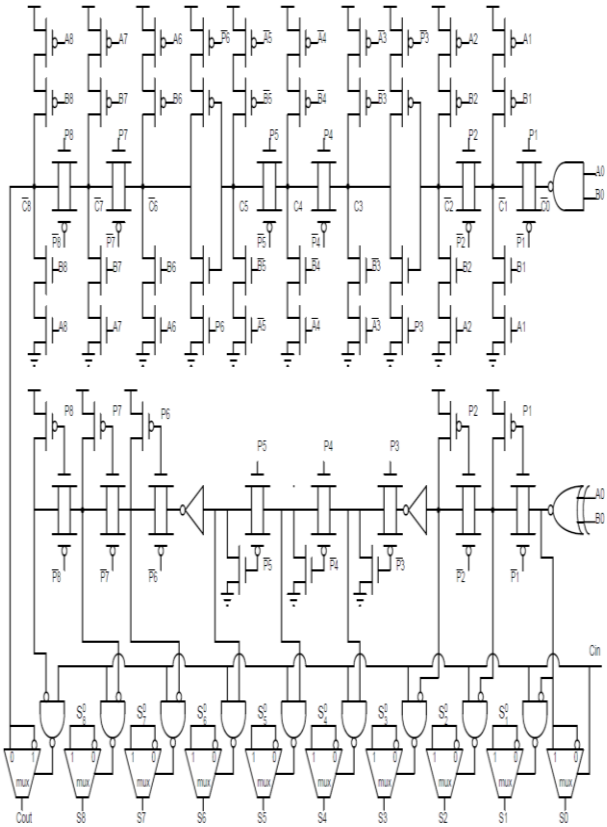
شکل (۱۲): ساختار حاصلضرب های جزئی ضرب کننده ۶۴ بیتی



شکل (۱۳): الف) مدارهای تولید کننده H_i ب) K_i [۳۱]

از این رو این ساختار برای ضرب کننده های با بیت های بالا مناسب تر است. استفاده از کمپرسورها بصورت موازی در این ساختار به دلیل سرعت بالا و سخت افزار کوچک رواج بیشتری دارد[۲۰][۱۶]. با توجه به شکل ۱۲، تعداد ردیف های حاصلضرب های جزئی تولید شده با استفاده از الگوریتم بوث در طبقه ی اول شامل ۳۳ (N/۲+۱) ردیف است (ردیف ۳۳ با یک بیت که بیت علامت ردیف ۳۲ می باشد). این ۳۳ ردیف باید با کمپرسورهای ۴:۲ و ۵:۲ به ۲ ردیف کاهش یابد تا جمع کننده ی نهایی دو ردیف را جمع کرده و نتیجه ی نهایی ضرب را محاسبه کند.

با توجه به شکل های ۱۰ و۱۲ این کار در چهار سیکلِ دوم، سوم، چهارم و پنجم صورت می گیرد. بطوریکه درسیکل دوم، به صورت همزمان ۸ ردیف کمپرسور ۴ به ۲، ۳۲ ردیف را به ۱۶ ردیف تبدیل می کنند. در سیکل سوم، ۴ ردیف کمپرسور ۴به ۲، تعداد ردیف ها را از ۱۶ردیف به ۸ ردیف کاهش می دهند و در سیکل چهارم، ۲ ردیف کمپرسور ۴ به ۲ تعداد ردیف ها را از ۸ ردیف به ۴ ردیف کاهش می دهند و در سیکل چهارم، ۱ردیف کمپرسور ۴ به ۲ و کمپرسور ۵ به ۲، تعداد ردیف ها را از ۴ ردیف به ۲ ردیف کاهش می دهند. علاوه بر موارد فوق، در بعضی قسمت ها که دو ردیف وجود دارد، از جمع کننده های پیش بینی کننده نقلی ۵-بیتی یا ۸-بیتی استفاده می شود تا در مراحل بعدی رجیستر و سخت افزار کمتری لازم باشد و همچنین اندازه ی جمع کننده ی نهایی کوچکتر شود. بعد از طبقه ی پنجم، دو ردیف باقی می ماند که ردیف دوم از بیت با ارزش مکانی ۳۲ شروع شده و با این حساب از این بیت تا بیت ۱۲۸- اُم باید عمل جمع نهایی توسط جمع کننده ۹- بیتی صورت گیرد. کمپرسور ۴:۲ بهینه شده دارای تاخیر سه گیت XOR وکمپرسور ۵:۲ بهینه شده دارای تاخیر ۴ گیت XOR می باشد.



شکل (۱۴): ساختار کامل جمع کننده ۹ بیتی

۲-۲-۳- جمع کننده نهایی

از مهمترین قسمت های یک ضرب کننده، جمع کننده ی نهایی آن است که تأثیر قابل توجهی بر تأخیر کلی ضرب کننده دارد. در طراحی جمع کننده، ساختاری به کار رفته است که برای جمع کننده ی نهاییِ ضرب کننده نیز قابل استفاده باشد. با توجه به توضیحات قسمت قبل، بعد از طبقه ی پنجم، دو ردیف باقی می ماند که ردیف دوم از بیت با ارزش مکانی ۳۲ شروع شده و با این حساب از این بیت تا بیت ۱۲۸ اُم، عمل جمع نهایی توسط جمع کننده ۹- بیتی صورت می گیرد. در این جمع کننده از ۱۱ جمع کننده جزئی ۹- بیتی استفاده شده است. شکل ۱۴ ساختار کامل جمع کننده ۹- بیتی طراحی شده را نشان می دهد.

۳ - نتایج شبیه‌سازی

حال ساختار ضرب کننده و جمع کننده ارائه شده در این مقاله با نتایج بدست آمده از تحقیقات دیگر مقایسه می شود. شبیه سازی های جمع کننده و ضرب کننده با استفاده از نرم افزار HSPICE در تکنولوژی TSMC 130nm CMOS ، ولتاژ تغذیه ۱/۳ ولت و فرکانس کلاک ۲ گیگا هرتز انجام شده است.

در جدول ۲ نتایج بدست آمده از سایر جمع کننده ها با جمع کننده ارائه شده مقایسه شده است. استفاده از ترکیب دو ساختار جمع کننده پیش بینی کننده نقلی و جمع کننده انتخاب کننده نقلی سبب کاهش تأخیر و بالا رفتن سرعت جمع کننده شده است. مدار "جمع کننده با یک" نیز دارای سخت افزار کم و توان مصرفی پایین است. همچنین ورودی های مدار "جمع کننده با یک" به جای اینکه به خروجی های تولید شده در جمع کننده های جزئی وصل شوند، به خروجی های گیت های XOR واقع در جمع کننده های جزئی متصل می شوند که با این کار ورودی های مدار "جمع کننده با یک" همزمان اعمال می شوند و سرعت کل جمع کننده بالا می رود. سوم و مهمتر اینکه در این جمع کننده از یک مدار دیگر برای تولید بیت نقلی هر بلوک استفاده شده است و بر خلاف روش هایی که بیت نقلی هر بلوک را مستقیماً از ورودی های جمع کننده محاسبه می کنند، سخت افزار کوچکتری دارد. همچنین استفاده از گیت XOR در مدار انتخاب کننده نهایی سبب کاهش توان مصرفی و تعداد ترانزیستورها شده است. بنابراین جمع کننده طراحی شده داری تأخیر و توان پایین می باشد. حاصل ضرب توان– تأخیر[۱۴] هم معیاری برای ارزیابی کیفیت یک افزاره ی سوئیچ شونده است. مطابق با جدول ۲، حاصلضرب توان- تأخیر جمع کننده ارائه شده نسبت به نمونه های دیگر کمتر می باشد. در جدول ۳ چند نمونه از ضرب کننده ها با هم مقایسه شده اند. تعداد ردیف های حاصلضرب های جزئی تولید شده در مرحله اول ضرب کننده با استفاده از الگوریتم بوث تقریبا به نصف کاهش یافته است که این مساله باعث افزایش سرعت ضرب کننده شده است. همچنین استفاده از ساختار والاس و کمپرسور ها که جمع حاصلضرب های جزئی درآن به صورت موازی صورت می گیرد دارای سرعت بالا و سخت افزار کوچک می باشند. جمع کننده نهایی طراحی شده نیز که در مرحله آخر ضرب کننده استفاده شده است، طبق توضیحات قسمت های قبل دارای سرعت بالایی می باشد. همچنین استفاده از روش خط لوله سرعت ضرب کننده ارائه شده را بهبود داده است.

جدول (۲) : مقایسه جمع کننده های سریع با جمع کننده ارائه شده

	[32]	[33]	[10]	This work
Process(µm)	0.09	0.065	0.18	0.13
Input bits	64×64	64×64	64×64	64×64
Supply voltage(V)	1	1	1.8	1.3
Delay(ns)	0.203	0.148	0.592	0.112
Power(mW)	9.58	135	--	12
PDP(pJ)	1.94	19.98	--	1.34

از این رو سرعت نمونه برداری بالا و در عین حال تأخیرکلی کم (تعداد طبقات کم) از ویژگی های بارز ضرب کننده ی ۶۴ بیتی ارائه شده در این مقاله می باشد که با تکنولوژی ۱۳۰ نانومتر CMOS و با سرعت کلاک ۲ گیگا پیاده سازی شده است.

جدول (۳): مقایسه ضرب کننده های سریع با ضرب کننده ارائه شده

	[34]	[31]	[35]	This work
Process(µm)	0.25	0.35	0.18	0.13
Input bits	16×16	16×16	32×32	64×64
Supply voltage(V)	2.5	3.3	1.8	1.3
Frequency(MHz)	50	1600	1000	2000
Delay(ns)	10.59	3.1	0.38	0.291
Power(mW)	17.12	169	543	950
PDP(pJ)	181.30	523.9	206.34	276.45

۴- نتیجه‌گیری

در این مقاله یک واحد حساب و منطق ۶۴ بیتی ارائه شد که در آن جمع کننده از ترکیب جمع کننده های انتخاب کننده نقلی و پیش بینی کننده نقلی و ضرب کننده با استفاده الگوریتم بوث، در تکنولوژی ۱۳۰ نانومترCMOS و سرعت کلاک ۲ گیگا هرتز طراحی و شبیه سازی گردید. استفاده از جمع کننده های پیش بینی کننده نقلی و جمع کننده انتخاب کننده نقلی و ترکیب گیت انتقال و C-CMOS باعث دستیابی به جمع کننده ای با سرعت بالا و توان کم شد. همچنین در طراحی ضرب کننده، استفاده از الگوریتم بوث و تکنیک خط لوله، سرعت ضرب کننده را افزایش داد که در ۷ سیکل عمل ضرب را انجام می دهد.. تعداد حاصلضرب های جزئی در ضرب کننده ۶۴×۶۴ بیتی ۴ برابر ضرب کننده ۳۲×۳۲ بیتی و ۱۶ برابر ضرب کننده ۱۶×۱۶ بیتی می باشد که این معیار مناسبی برای مقایسه توان مصرفی و تعداد ترانزیستورهای ضرب کننده ها می باشد. از نتایج بدست آمده مشخص گردید ضرب کننده ۶۴×۶۴ بیتی ارائه شده دارای تأخیر کمتر و حاصل ضرب توان- تأخیر بهبود یافته ای می باشد.

مراجع

[1] Uniyal, A., Niranjan, V. A new16-bit ALU using variable stage adder and PTL mux, IEEE International Conference on Computing, Communication and Automation, pp. 1374-1378, 2017

[2] Mukhedkar, M., Pandurang, W. A 180 nm Efficient Low Power and Optimized Area ALU design using Gate Diffusion Input technique, IEEE International Conference on Data Management, Analytics and Innovation Zeal Education Society, Pune, India, pp. 47-51, 2017

[3] Yezerla, S., Naik, B. Design and Estimation of delay, power and area for Parallel prefix adders, Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 2014

[4] Natarajan P., Ghosh S., Karthik, R. Low Power High Performance Carry Select Adder, IEEE International Conference on Electronics, Communication and Aerospace Technology, 2017
[5] Balasubramanian, P., Dang, C., Maskell, D., Prasad, K. Approximate Ripple Carry and Carry Lookahead Adders-A Comparaitive analysis, IEEE International Conference On Microelectronics, 2017
[6] Alghamdi, A., Gebali, F. Performance Analysis of 64-bit Carry Lookahead Adders Using Conventional and Hierarchical Structure Styles, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada, 2015
[7] Koyada, B., Meghana, N., Jaleel, M., Jeripotula, P. A Comparative Study on Adders, International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India, 2017
[8] Bahadori, M., Kamal, M., Afzali-Kusha, A. High-Speed And Energy-Efficient Carry Skip Adder Operating Under A Wide Range Of Supply Voltage Levels, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol, 24, No. 2, pp.421-433, 2015
[9] Jaberipur, Gh., Gorgin, S., Design and Synthesis of High Speed Low Power Signed Digit Adders, Journal of Iranian Association of Electrical and Electronics Engineers., Vol.7, No. 2, pp. 7-14, 2010.
[10] Tamar, H., Tamar, A., Hadidi, Kh., Khoei, A., Hoseini, P., High Speed Area Reduced 64-bit Static Hybrid Carry Lookahead/Carry-Select Adder, IEEE International Conference on Electronics, Circuits and Systems, Beirut, Lebanon, 2011
[11] Luu, X., Hoang, T., Bui, T., Dinh-Duc, A. A High-speed Unsigned 32-bit Multiplier Based on Booth-encoder and Wallace-tree Modifications, International Conference on Advanced Technologies for Communications, Hanoi, Vietnam, 2014
[12] Ghasemzadeh, M., Akbari, A., Hadidi, Kh. An Ultra High Speed Booth Encoder Structure for Fast Arithmetic Operations, IEEE International Conference Mixed Design of Integrated Circuits and Systems, Torun, Poland, 2015
[13] Prathiba, R., Sandhya, P., Varun, R. Design of High Performance and Low Power Multiplier using Modified Booth Encoder, IEEE International Conference on Electrical, Electronics, and Optimization Techniques, Chennai, India, 2016
[14] Qian, L., Wang, Ch., Liu, W., Lombardi, F., Han, J. Design and Evaluation of An Approximate Wallace-Booth Multiplier, IEEE International Symposium on Circuits and Systems ,Montreal, QC, Canada, 2016
[15] Bokade, S., Dakhole, P., CLA based 32-Bit Signed Pipelined Multiplier, International Conference on Communication and Signal Processing, Melmaruvathur, India, 2016
[16] Ram, G., Rani, D., Balasaikesava, R., Sindhuri, K. Design of Delay Efficient Modified 16 bit Wallace Multiplier, IEEE International Conference On Recent Trends In Electronics Information Communication Technology, Bangalore, India, 2016
[17] Asif, S., Kong, Y. Performance Analysis of Wallace and Radix-4 Booth-Wallace Multipliers, Electronic System Level Synthesis Conference, San Francisco, CA, USA, 2015
[18] Mandloi, A., Agrawal, S., Sharma, S., Shrivastava, S. High-speed, area efficient VLSI architecture of Wallace-Tree multiplier for DSP-applications , International Conference on Information, Communication, Instrumentation and Control, Indore, India , 2017
[19] Vyas, K., Jain, G., Maurya, V., , Mehra, A., Analysis of an Efficient Partial Product Reduction Technique, International Conference on Green Computing and Internet of Things, Noida, India, 2015
[20] Balobas. D., Konofaos, N., Low-power high-performance CMOS 5-2 compressor with 58 transistors, Electronics Letters, Vol 54, No 5, pp.278-280, 2017
[21] Asif, Sh., Kong, Y. Design of an Algorithmic Wallace Multiplier using High Speed Counters, International Conference on Computer Engineering & Systems , Cairo, Egypt, 2015
[22] Kumar, D., Kumar, M. Modified 4-2 Compressor Using Improved Multiplexer for Low Power Applications, IEEE International Conference on Advances in Computing, Communications and Informatics, Jaipur, India, 2016
[23] Kumar, V., Bujjibabu, P. Design and Implementation of RADIX-8 Based 32-bit Pipelined Multiplier by using CLA, International Conference on Communication and Electronics Systems, Coimbatore, India, 2017
[24] Kshirsagar, R., Aishwarya, E., Vishwanath, A., Jayakrishnan, P. Implementation of Pipelined Booth Encoded Wallace Tree Multiplier Architecture, International Conference on Green Computing, Communication and Conservation of Energy, Chennai, India, 2013.
[۲۵]فرشیدی، ا.، تیزنوبیک، م. کاهش سخت‌افزار و توان نویز کوانتیزه در مدولاتورهای دلتا-سیگمای دیجیتال و پیاده سازی توسط زبان توصیف سخت‌افزار VHDL، مجله مهندسی برق و الکترونیک ایران، جلد ۱۶، شماره ۲، ص. ۱۱۲-۱۰۱، تابستان ۱۳۹۸.
[26] Chang, T., Hsiao, M. Carry-select adder using single ripple-carry adder, Electronics Letter, Vol. 34, No. 22, pp. 2101–2103, 1998.
[27] Ruiz, G., Granda, M. An area-efficient static CMOS carry-select adder based on a compact carry look-ahead unit, Microelectronics Journal, Vol. 35, No. 12, pp. 939-944, 2004.
[28] He, Y., Chang, C. A Power-Delay Efficient Hybrid Carry Lookahead/ Carry-Select Based Redundant Binary to Two's Complement Converter, IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 55, No. 1, pp. 336 - 346, 2008.
[29] Anderson, F., Earle, J., Goldschmidt, R., Powers, D. The IBM system 360/91 floating point execution unit, IBM J. Res. Develop., vol.11, pp.34-53, Jan. 1967.
[30] Wang, L., Jou, S., Lee, C. A well-structured modified Booth multiplier design",Proc. of IEEE VLSI-DAT, pp. 85-88, April. 2008.
[31] Ghasemizadeh, H., Fathi, A., Ghasemzadeh, A., High Speed 16×16-bit Low-Latency Pipelined Booth Multiplier, Electrical and Electronic Engineering, Vol. 2, No. 3, pp. 121-127, 2012.
[32] Shieh, S., Huang, D., Chu, Y. Low Voltage and Low Power64-Bit Hybrid Adder Based on Radix-4 Prefix Tree Structure , International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 2014
[33] Chuang, P., Li, D., Sachdev, M., Gaudet, V. A 148ps 135mW 64-bit Adder with Constant-Delay Logic in 65nm CMOS, IEEE Custom Integrated Circuits Conference ,San Jose, CA, USA, 2012.
[34] Oliveira, L., Santos, C., Ferrao, D., Costa, E., Monteir, J., Martinz, J., Bampi, S., Reis, R. A Comparison of Layout Implementations of Pipelined and Non-Pipelined Signed Radix-4 Array Multiplier and Modified Booth

Multiplier Architectures, Springer, International Federation for Information Processing, Volume 240,
[35] Teymouri, Sh. Design and Implemention 1GHz , 32 Bit ALU in 0.18µm CMOS Process, MSc Thesis , RAZI University , 2014.

VLSI-SoC,2007.

زیرنویس‌ها

[1] Arithmetic Logic Unit(ALU)
[2] Central Processing Unit(CPU)
[3] Carry Select Adder (CSA)
[4] Carry Lookahead Adder (CLA)
[5] Partial Product
[6] Modified Booth Encoder (MBE)
[7] Wallace Tree
[8] Compressor
[9] Counter
[10] Pipeline
[11] Add-on
[12] Transmission gate (TG)
[13] Multiplicand
[14] Power Delay Product(PDP)