فصلنامه پژوهش در علوم، مهندسی و فناوری
دوره ۳، شماره ۲، تابستان ۱۳۹۶
صفحات ۸۱-۸۹
ISSN: 2476-5082
www.uctjournals.com

مدلسازی کاهش بهینه تاخیر با استفاده از جمع کننده گزینشی رقم نقلی (CSLA)

سید علی هاشمی[1]* و دکتر کوروش منوچهری کلانتری[2]

۱ کارشناسی ارشد - رشته معماری سیستم های کامپیوتری - گروه کامپیوتر - دانشگاه صنعتی امیرکبیر واحد گرمسار
*نویسنده مسئول
۲استادیار - دکترای معماری سیستم های کامپیوتری - دانشگاه صنعتی امیرکبیر واحد گرمسار - گروه کامپیوتر

چکیده

جمع کننده گزینش رقم نقلی[1](CSLA) در بسیاری از سیستم‌های محاسباتی کاربرد دارد تا مشکل تأخیر پخش رقم نقلی با تولید جداگانه چند رقم نقلی و سپس گزینش یک رقم نقلی برای تولید جمع رفع شود. با این حال، CSLA از نظر فضا کارآمد نیست زیرا از چند زوج جمع کننده رقم نقلی پله‌ای[2](RCA) برای تولید جمع و رقم نقلی جزئی با درنظرگیری ورودی رقم نقلی استفاده می‌کند و سپس جمع و رقم نقلی نهایی به وسیله تسهیم کننده‌ها (mux) انتخاب می‌شود. ایده اولیه این تحقیق استفاده از مبدل دوتایی به اضافی-۱[3](BEC) بجای RCA در CSLA منظم جهت رسیدن به سرعت است.

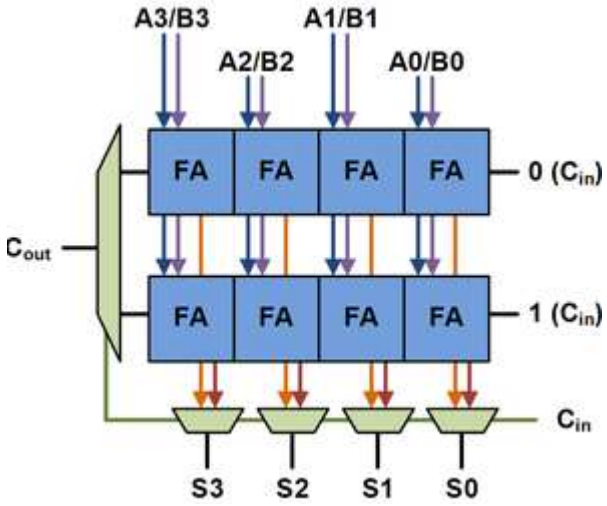
واژه‌های کلیدی: جمع کننده، گزینش رقم نقلی ، مبدل دوتایی اضافی-۱[4]BEC)

[1] Carry select adder
[2] Ripple Carry Adder
[3] Binary to Excess-1 Converter
[4] Binary to Excess-1 Converter

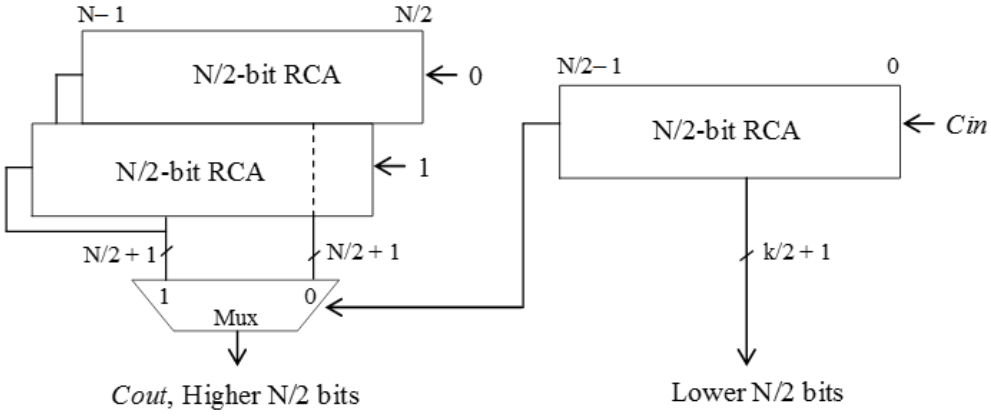
۱- مقدمه

جمع کننده ها یکی از بلوک های اصلی در پردازش های دیجیتال به شمار می آیند. روش های بسیار متنوعی به منظور پیاده سازی و بهبود سطح مصرفی، توان مصرفی، سرعت و ... برای این مدارات مطرح شده است. دو جمع کننده اصلی که در این پروژه با آن ها آشنا می شویم عبارتند از : جمع کننده گزینشی رقم نقلی[5] و جمع کننده رقم نقلی موج گونه[6].

همانطور که در [۱]اشاره شد، یک جمع کننده جمع کننده رقم نقلی موج گونه به علت تأخیر در انتشار مقدار بیت نقلی، سرعت پایینی دارد. در جمع کننده گزینشی رقم نقلی سعی شده به نحوی این مشکل برطرف شود؛ بدین صورت که ابتدا حاصل جمع به ازای هر دو حالت ممکن بیت نقلی ورودی (صفر یا یک)، با استفاده از دو جمع کننده رقم نقلی موج گونه به صورت موازی محاسبه می شود. در این حالت نیازی نیست که صبر شود تا بیت نقلی ورودی از طبقه قبل برسد تا عمل جمع انجام شود. پس از محاسبه دو مقدار ممکن جمع، با استفاده از یک مالتی پلکسر و بیت نقلی ورودی (که اکنون مقدار آن در طبقه قبل محاسبه شده و به این طبقه رسیده) حاصل جمع صحیح انتخاب می شود. در این صورت مقدار تأخیر کاهش یافته و سرعت محاسبه حاصل جمع، بیشتر از حالت جمع کننده رقم نقلی موج گونه خواهد بود.



شکل۱-۱ : نمایش شماتیک جمع کننده گزینشی رقم نقلی



شکل۲-۱ : جمع کننده گزینشی رقم نقلی k بیتی

[5] Carry Select Adder
[6]Carry Ripple Adder

۲- کارهای مرتبط

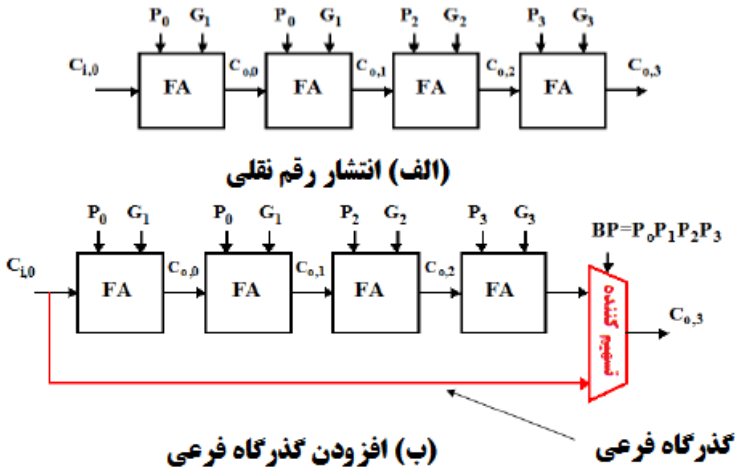
در سال ۱۹۶۲ بدریج[7] مدار جمع کننده دیجیتالی سریعی را به همراه انتخاب مجموع و رقم نقلی ضرب پذیر مبنایی تفسیر نمود. میزان تاخیر منطقی برای یک جمع کننده رقم نقلی موج گونه ۱۰۰ بیتی و یک مدل انتخابی توسط او مقایسه شد . چانگ و همکاران در عوض استفاده از جمع کننده رقم نقلی دوبل ، یک جمع کننده رقم نقلی انتخابی را با استفاده از مدار جمع پذیر برای جایگزینی با جمع کننده موج گونه با ۲۹,۲ درصد ترانزیستور و جریمه سرعت ۵,۹ درصد برای طول بیت n=64 نیاز داشت. اگر سرعت برای این جمع کننده رقم نقلی انتخابی می تواند با طرح ۶,۳ درصد ذخیره ناحیه کارآمد با همان سرعت پیشنهاد شود.

رامکومار[8] در سال ۲۰۱۰ یک روش ساده بهبود سرعت افزوده پیشنهاد داد مبنی بر این روش برای ۱۶،۳۲ و ۶۴ بیتی معماری جمع کننده توسعه داده شد.تعدادی از دانشمندان برای بهبود عملکرد CSLA تلاش کردند در مرجع ۱ فرمولاسیون منطقی برای CSLA با حذف همه عملیات منطقی اضافه از طراحی CSLA مرسوم انجام دادند در این طراحی عملیات قبل محاسبه SUM نهایی برنامه ریزی می شود در مرجع [۳] معماری مختلف CSLA ارائه شد که آنالیز معماری ارائه شده برای سرعت و ناحیه کارآمد بررسی شدند. طراحی گیت اصلاح شده با ناحیه توان کارآمد در [۸و۴] انجام شد که با مینیمم سازی عملیات منطقی در مقایسه با طراحی CSLA مرسوم بود. تحلیل CSLA ، ۱۶ بیتی مرسوم و باینری برای مبدل اضافی-۱ (BEC) CSLA در [۷] ارائه شد .معماری اپتیمایز شده با ناحیه تاخیر از ۱۶، ۳۲ و ۶۴ بیتی جمع کننده CLSA در [۵و۶] پیشنهاد شد مرجع [۱۶] شبیه سازی و عملکرد ارزیابی معماری ۱۶ بیتی اصلاح یافته از ریشه مجذورCSLA (SQRT-CSLA) ارائه میدهد. شبیه سازی مبنی بر توان – تاخیر – مساحت از طراحی اصلاح یافته منطق اضافه از CSLA با توجه به مرجع [۱۵]مورد استفاده معماری مالتی پلاکسر صورت نگرفت. یک واحد منطقی LCU مبنی بر معماری اصلاح یافته از جمع کننده پیشنهاد شد که برای اپتیمم سازی پارامتر توان – تاخیر – مساحت بود.

۳- الگوریتم مرسوم

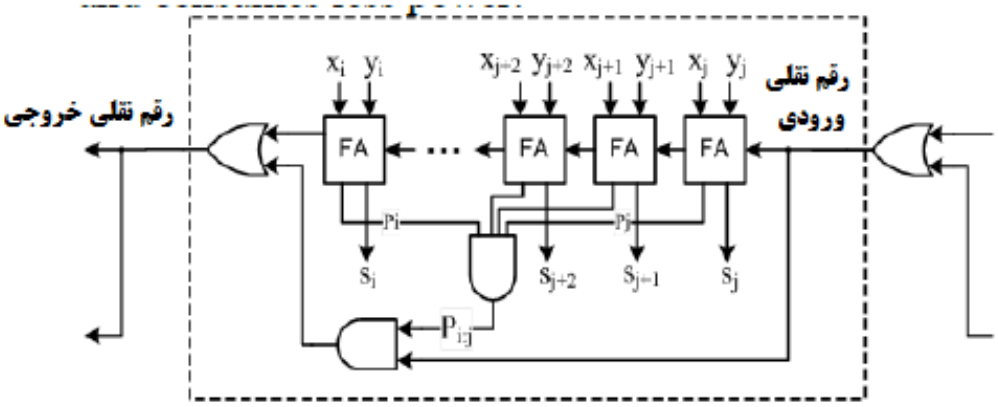
یک جمع کننده جهش رقم نقلی از یک جمع کننده رقم نقلی پله‌ای ساده با یک زنجیره رقم نقلی سرعتی ویژه بنام زنجیره جهش تشکیل شده است. جمع کننده جهش رقم نقلی زمانی که تعداد زیادی بیت ایجاد می‌شود، نسبت به جمع کننده رقم نقلی پله‌ای جمع کننده‌ای سریع محسوب می‌شود؛ جمع کننده جهش رقم نقلی که دارای تأخیر $O(\sqrt{n})$ است، سازش خوبی از نظر تأخیر علاوه بر جانمایی ساده و منظم فراهم می‌کند. این زنجیره توزیع بلوک‌های رقم نقلی پله‌ای را تعریف می‌کند که جمع کننده جهش رقم نقلی را می‌سازد. یک جمع کننده جهش رقم نقلی به این منظور طراحی شده است که سرعت جمع کننده عریض را با کمک به پخش یک بیت رقم نقلی حول بخشی از کل جمع کننده افزایش دهد. در حقیقت جمع کننده رقم نقلی پله‌ای برای مقادیر کم N سریع‌تر است. با این حال، تقاضای صنعت امروزی که اغلب رایانه‌های دسکتاپی همانند پردازنده‌های چندرسانه‌ای از طول کلمه ۳۲ بیت استفاده می‌کنند سبب می‌شود ساختار جمع کننده جهش رقم نقلی به ساختاری جالب‌تر تبدیل گردد. ساختار اولیه جمع کننده جهش رقم نقلی در شکل ۳-۱ مشاهده می‌شود.

[7] O. J. Bedrij, “Carry select adder,” IRE Trans. Electron. Comput . , pp.340 –344 , 1962.
[8] B. Ramkumar, H.M. Kittur, and P. M. Kannan, “ASIC Implementation Of Modified Faster Carry Save Adder,” Eur. J. Sci. Res. , vol. 42, no. 1, pp. 53 –58, 2010.



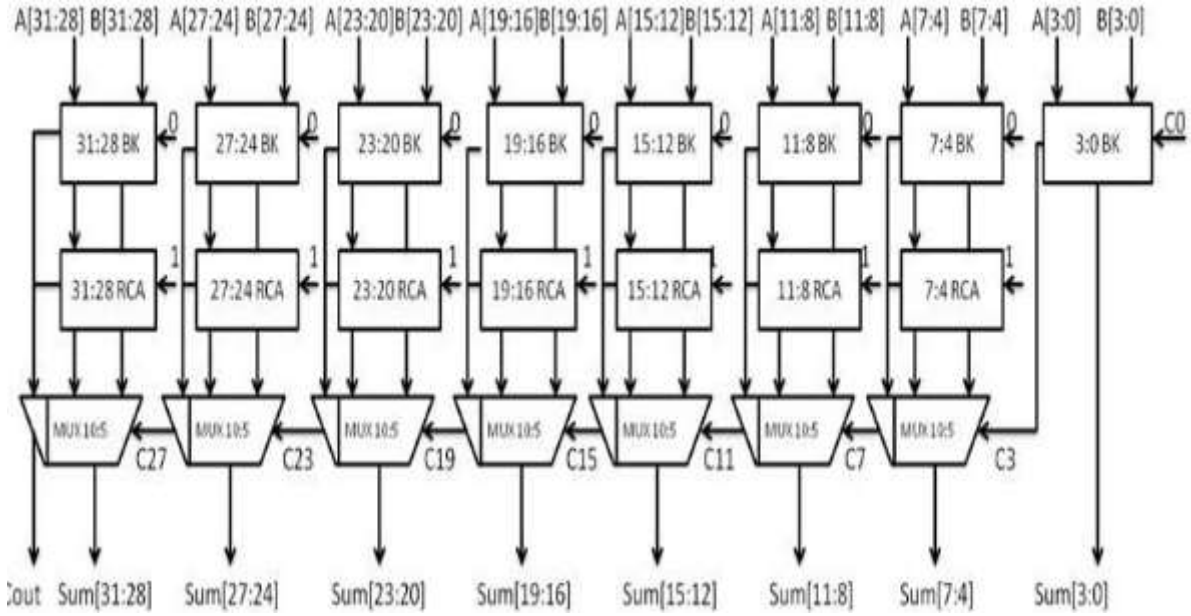
شکل ۳-۱. ساختار جمع کننده جهش رقم نقلی-مفهوم اولیه

نقطه تلاقی بین جمع کننده رقم نقلی پله‌ای و جمع کننده جهش رقم نقلی به ملاحظات فناوری وابسته است و معمولاً ۴ تا ۸ بیت فضا می‌گیرد. مداربندی جهش رقم نقلی از دو گیت منطقی تشکیل شده است. گیت AND بیت رقم نقلی داخلی را می‌پذیرد و آن را با سیگنال پخش گروه با استفاده از مقادیر پخش جداگانه مقایسه می‌کند. یک جمع کننده جهش رقم نقلی با جهش از گروه‌های مراحل جمع کننده متوالی سبب کاهش زمان پخش رقم نقلی می‌شود. جمع کننده جهش رقم نقلی معمولاً از نظر سرعت با شیوه جمع کننده با پیش بینی رقم نقلی قابل مقایسه است. اما به فضای کمتری برای تراشه نیاز داشته و برق کمتری مصرف می‌کند.



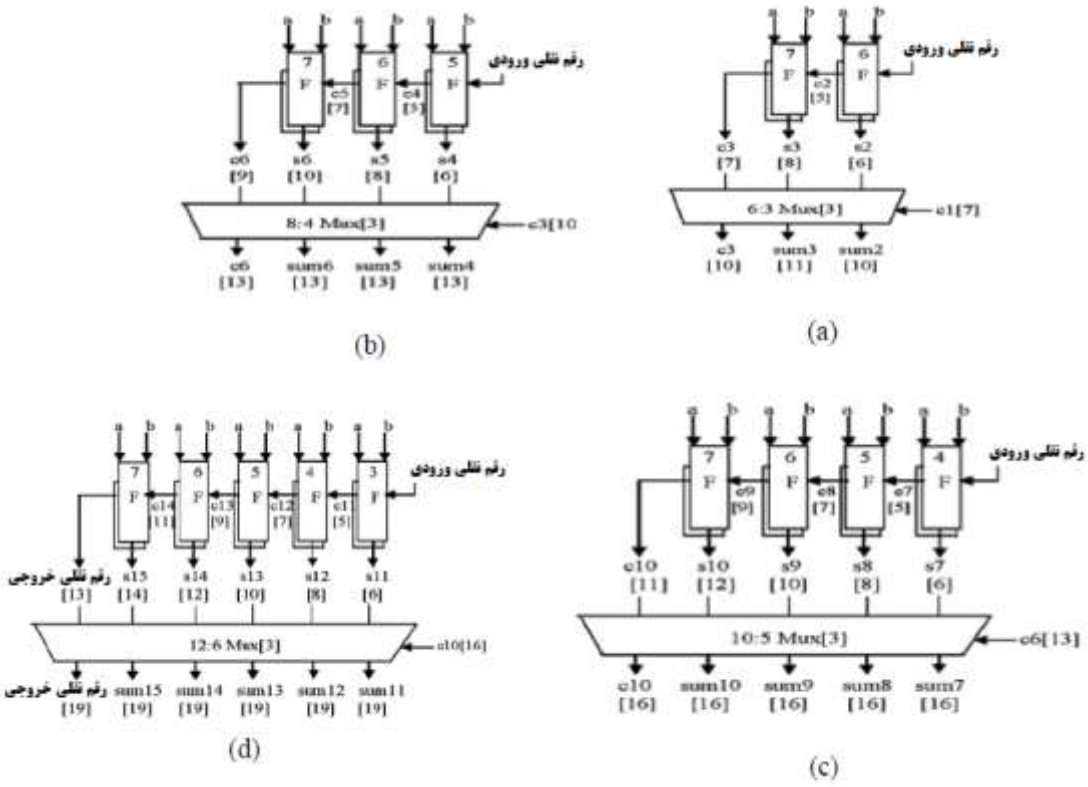
شکل ۳-۲. جمع کننده جهش رقم نقلی

در جمع کننده جهش رقم نقلی، از هر مرحله جمع کننده که برای آن $Pm = xm \text{ exor } ym = 1$ باشد، می‌توان جهش کرد که در اینجا P_m نشان دهنده m امین پخش رقم نقلی است. ساختار این جمع کننده به بلوک‌های مراحل متوالی با یک طرح رقم نقلی پله‌ای ساده تقسیم می‌شود. هر بلوک نیز یک سیگنال پخش رقم نقلی بلوک را تولید می‌کند که مساوی ۱ است مشروط براینکه کلیه مراحل داخلی بلوک رابطه $P_m=1$ را ارضا کنند. این سیگنال برای آنکه به یک رقم نقلی ورودی اجازه داده شود تا که از کلیه مراحل در بلوک جهش کند و یک رقم نقلی خارجی بلوک را تولید کند، قابل استفاده است. شکل ۳-۳ یک بلوک نمونه را نشان می‌دهد که از k محل بیت $j, j+1, \ldots, j+k-1$ تشکیل شده است.



شکل ۳-۳. جمع کننده گزینش رقم نقلی ۳۲ بیتی

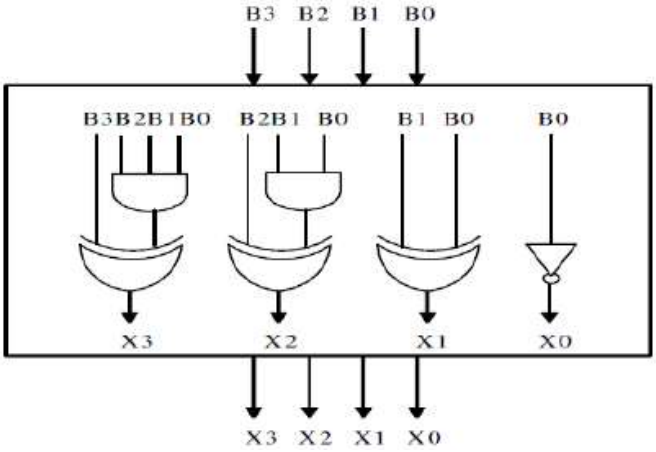
ساختار جمع کننده گزینش رقم نقلی ۳۲ بیتی در شکل ۵ مشاهده می‌شود. این ساختار از ۵ گروه جمع کننده رقم نقلی پله‌ای با سایزهای مختلف تشکیل شده است.



شکل ۳-۴. محاسبه تأخیر و فضای SQRT CSLA منظم: الف. گروه ۲؛ ب. گروه ۳؛ ث. گروه ۴ و د. گروه ۵. F یک تمام جمع کننده است.

۴-مدل پیشنهادی

محدودیت سرعت اصلی در هر جمع کننده در تولید ارقام نقلی دیده می‌شود و بسیاری از مؤلفین مسئله جمع را درنظر گرفته‌اند. ایده اولیه تحقیق پیشننهادی استفاده از مبدل‌های BEC جهت افزایش سرعت عمل جمع است. این منطق را می‌توان با جمع کننده گزینش رقم نقلی اجرا کرد تا ساختاری کم مصرف و با فضای کارآمد حاصل شود. جمع کننده گزینش رقم نقلی ۳۲ بیت پیشنهادی با جمع کننده جهش رقم نقلی[9](CSKA) و جمع کننده گزینش رقم نقلی ۳۲ بیتی منظم مقایسه می‌شود. CSLA در بسیاری از سیستم‌های محاسباتی استفاده می‌شود تا مشکل تأخیر پخش رقم نقلی با تولید جداگانه چند رقم نقلی و سپس انتخاب یک رقم نقلی برای تولید جمع برطرف شود. با این حال، CSLA از نظر فضا کارآمد نیست زیرا از چند زوج RCA برای تولید جمع و رقم نقلی جزئی با درنظرگیری ورودی رقم نقلی Cin=0 و Cin=1 استفاده می‌کند و سپس جمع و رقم نقلی نهایی به وسیله تسهیم کننده‌ها انتخاب می‌شوند. کل کار با استفاده از مبدل BEC بجای RCA با Cin=1 در CSLA منظم انجام می‌شود تا مصرف انرژی کمتری حاصل آید. مزیت اصلی منطق این BEC آن است که از تعداد کمتری گیت منطقی بجای n RCA بیتی حاصل شده است. یک ساختار BEC ۴ بیتی و جدول حقیقی به ترتیب در شکل ۴-۱ و جدول ۴-۱ نشان داده شده‌اند.

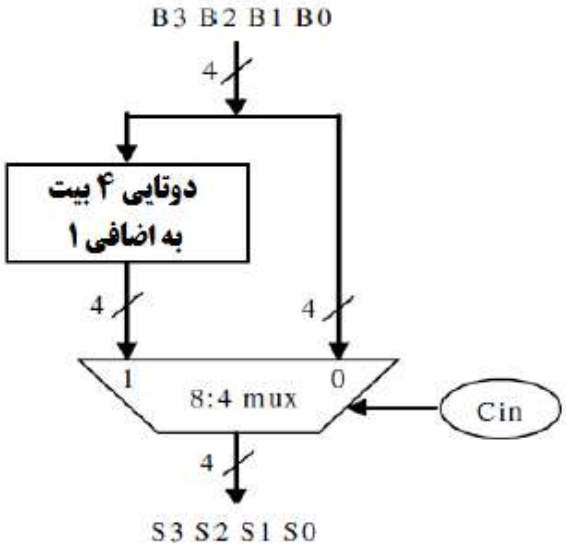


شکل ۴-۱. یک مبدل ۴ بیتی دوتایی به اضافی-۱ (BEC)

جدول ۴-۱. جدول کارکردی BEC ۴ بیتی

B[3:0]	X[3:0]
0000	0001
0001	0010
⋮	⋮
1110	1111
1111	0000

[9] Carry Skip Adder



شکل ۴-۲. BEC ۴ بیتی دارای 8:4 MUX

هدف جمع سریع با استفاده از BEC در کنار یک تسهیم کننده (mux) که در شکل ۴-۱ و ۴-۲ نشان داده شده است، عملی می‌شود، یک ورودی 8:4 mux زمانی که ورودی (B3,B2,B1,B0) داده می‌شود حاصل می‌شود و ورودی دیگر MUX خروجی BEC است. این جمع دو حاصلضرب جزئی محتمل را به طور موازی نتیجه می‌دهد و MUX ها برای انتخاب خروجی BEC یا ورودی‌های مستقیم براساس Cin سیگنال کنترل استفاده می‌شوند.

۵- ارزیابی نتایج

CSLA در سه متد مختلف مورد ارزیابی قرار گرفت و در مدل شبیه سازی شده تاخیر در موج گونه ۰,۷ و در همراه با RCA معادل ۰,۶ و در مدل با BEC برابر ۰,۵ بوده است لذا با این وجود CSLA های معمولی به دلیل ساختار دوتایی جمع کننده ی متناوب RCA، فضا و حجم زیادی مصرف می کنند. در این مقاله با جایگزین کردن یکی از RCA ها با مبدل کد BEC می توان مزایای زمان را بهبود بخشیده و با تحلیل نتایج نشان داد که ساختارهای بهبود یافته خیلی کارآمدتر و موثرتر از CSLA معمولی می باشند.

جدول ۵-۱:

Before Place & Route	8-Bit Adder	32-Bit Adder
	Timing (in ns)	Timing (in ns)
Carry Ripple Adder	0.70	2.50
Carry Select Adder with RCA	0.60	1.20
Carry Select Adder with BEC	0.50	0.70

جدول ۵-۲:

After Place & Route	8-Bit Adder	32-Bit Adder
	Timing (in ns)	Timing (in ns)
Carry Ripple Adder	1.19	3.97
Carry Select Adder with RCA	0.97	1.85
Carry Select Adder with BEC	0.84	1.43

۶- نتیجه گیری

عمل جمع مرسوم‌ترین و پرکاربردترین عمل حسابی در ریزپردازنده‌ها، پردازنده‌های سیگنال دیجیتال و به ویژه رایانه‌های دیجیتال است. هم چنین، عمل جمع به عنوان یک بلوک سازنده برای ترکیب کلیه اعمال حسابی دیگر بکار می‌رود. بنابراین، درمورد پیاده سازی یک واحد منطق حسابی بطور کارآمد، ساختارهای جمع کننده واحد سخت افزاری بسیار مهم و حیاتی محسوب می‌شوند. جمع کننده‌های دیجیتال بلوک اصلی پردازنده‌های DSP هستند. ساختار جمع کننده پخش رقم نقلی (CPA) نهایی بسیاری از جمع کننده‌ها، تأخیر زیادش در پخش رقم نقلی ایجاد می‌کنند و این تأخیر سبب کاهش عملکرد کلی پردازنده DSP می‌شود. در این پروژه، محاسبات کمی جمع کننده CSLA با و بدون معماری‌های BEC ارائه گردید. در مورد زمان تأخیر می‌توان این طور نتیجه گرفت که پیاده سازی CSLA دارای BEC کارآمد است. مزیت اصلی این منطق BEC از شمار کمتر گیت‌های منطقی نسبت به ساختار تمام جمع کننده (FA) n بیتی نشأت می‌گیرد.

منابع

[1]Z. Navabi, VHDL Analysis & Modeling of Digital System. MC Grow Hill, 1998.
[2]Basant Kumar Mohanty and Sujit Kumar Patel, "Area Delay Power Efficient Carry Select Adder", IEEE Transactions on Circuits and Systems-II, Volume-61 No.-6, 2014.
[3]Gurpreet Kaur, Loveleen Kaur and Navdeep Kaur, "Reduced Area Carry Select Adder with Low Power Consumptions", International Journalof Emerging Engineering Research and Technology, Volume-3 Issue-3, 2015.
[4] S. A. Mashankar, R. D. Kadam and P. R. Indurkar, "Power Efficient Carry Select Adder using D-Latch" International Research Journal of Engineering and Technology, Volume-2 Issue-3, 2015.
[5] M. D. Bobade and m. n. Kakatkar, "Implementation on FPGA Area-Delay Efficient Architecture of CSLA" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Volume-4 Issue-6, 2015.
[6] Priya H. Agrawal and Prashant R. Rothe, "Implementation of Area Delay and Power Efficient Carry Select Adder", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, Volume-3 Issue-7, 2015.
[7] Theegala Ravinder Reddy and P. Anjaiah, "Design of High Speed Hybrid Carry Select Adder", International Journal and Magazine of
Engineering, Technology, Management and Research, Volume-2 Issue-7, 2015.

[8] Lingappagari Raju and Tipparti Anil Kumar, "FPGA Implementation of Efficient Carry-Select-Adder using Verilog HDL", International Journal and Magazine of Engineering, Technology, Management and Research, Volume-2 Issue-6, 2015.
[9] Korra Ravi Kumar, Santosh Kumar Allenki and G. Ramesh, "FPGA Implementation of Area Delay and Power Efficient Carry Select Adder Architecture Design", International Journal on Recent and Innovation Trends in Computing and Communication, Volume-3 Issue-5, 2015.
[10]Anna Johnson and Rakesh S., "High Speed Non-Linear CSA used in Wallace Tree Multiplier and in Radix-4 Booth Recorded Multiplier", International Journal of Innovative Science, Engineering and Technology, Volume-2 Issue-3, 2015.
[11] S. Banu Priya and G. Lingeswaran, "Area Delay Power Efficient and Implementation of Modified Square-Root Carry Select Adder Architecture", International Journal for Trends in Engineering and Technology, Volume-4 Issue-2, 2015.
[12] Priya Meshram, Mithilesh Mahendra and Parag Jawarkar, "Designed Implementation of Modified Area Efficient Enhanced Square-Root Carry Select Adder", International Journal for Research in Emerging Science and Technology, Volume-2 Issue-5, 2015.
[13] S. Murugeswari and S. Kaja Mohideen, "Design of optimized Low-Power and Area-Efficient Digital FIR Filter using Modified Group Structures based Square-Root Carry Select Adder", Research Journal of Applied Sciences, Engineering and Technology, 2015.
[14] Priya H. Agrawal and Prashant R. Rothe, "Implementation of Fast Low-Power and Area Efficient Carry Select Adder", International Journal on Recent and Innovation Trends in Computing and Communication, Volume-3 Issue-4, 2015.
[15] Deepak Kumar Patel, Raksha Chouksey and Minal Saxena, "An Efficient VLSI Architecture for Carry Select Adder without Multiplexer", International Journal of Computer Applications, Volume-127 No.-9, 2015.
[16] Madhurima Bose and Sourabh Sharma, "Implementation and Analysis of High-Speed and Area-Efficient 64-bit Carry Select Adder" International Journal of Engineering Science and Innovative Technology, Volume-4 Issue-4, 2015.