

طراحی ALU تحمل‌پذیر اشکال با روش جدید پیاده‌سازی کد برگر

احمد توحیدی گل^۱، دانشجوی دکتری؛ رضا امیدی^۲، استادیار؛ کریم محمدی^۳، استاد

۱- دانشکده مهندسی برق و کامپیوتر - دانشگاه علم و صنعت - تهران - ایران - a_towhid@elec.iust.ac.ir

۲- دانشکده فنی مهندسی - گروه برق - دانشگاه زنجان - زنجان - ایران - rezaomidi@znu.ac.ir

۳- دانشکده مهندسی برق و کامپیوتر - دانشگاه علم و صنعت - تهران - ایران - mohammadi@iust.ac.ir

چکیده: واحد محاسبه و منطق از حساس‌ترین واحدهای سازنده یک پردازنده است که اکثر دستوره‌های یک پردازنده توسط این بخش انجام می‌شود. افزونگی زمانی یکی از مناسب‌ترین روش‌های مقابله با خطای گذرا است. در اغلب روش‌های افزونگی زمانی لازم است ابتدا خطا آشکار شود، بنابراین وجود مدارهای آزمون در کنار روش‌های افزونگی زمانی ضروری است. از بزرگ‌ترین ایرادهای مدارهای آزمون سربار سخت‌افزاری بالای این مدارها است که باعث می‌شود طراحان در طراحی مدارهای کوچک مجبور به استفاده از روش‌های غیرمعمول شوند. در این مقاله روش جدیدی برای پیاده‌سازی مدار چک‌کننده برگر ارائه شده است در این روش از مدارات حالت جریان جهت پیاده‌سازی کد برگر استفاده شده است که ویژگی‌های آن سرعت بالاتر و سخت‌افزار موردنیاز کمتر است. با توجه به نتایج توان مصرفی مدار پیشنهادی نسبت به مدار دیجیتال به‌طور متوسط تا حدود ۵۱ درصد کاهش یافته است و سطح اشغالی مدار آزمون حالت جریان ۷۴/۳ درصد کمتر از سطح مصرفی مدار معادل دیجیتال است. به‌طور متوسط هزینه مدار برگر حالت جریان (حاصل ضرب توان مصرفی در تأخیر و سطح مصرفی)، ۹۱ درصد کمتر از پیاده‌سازی برگر دیجیتال معادل است.

واژه‌های کلیدی: تحمل‌پذیری اشکال، افزونگی زمانی، مدار حالت جریان، کد باقیمانده، کد برگر

Fault Tolerant ALU Designing based on New Implementation of Berger Code

A. Towhidy gol¹, PhD Student; R. Omid², Assistant professor; K. Mohammadi³, Professor

1- School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran, Email: a_towhid@elec.iust.ac.ir

2- Faculty of Engineering, University of Zanjan, Zanjan, Iran, Email: rezaomidi@znu.ac.ir

3- School of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran, Email: mohammadi@iust.ac.ir

Abstract: ALU is one of the most sensitive units of a processor, which most of the instructions of a processor are executed by this section. In most of the Architecture methods for mitigation of fault such as temporal redundancy, it is necessary to detect the errors first. Hardware overhead of test circuits is one of the most important disadvantages for testing small circuits which makes designers to use unusual methods in the design of small circuits. In this Article, a new implementation method for Berger code has been used to detect the error and compared with the previous method that is based on the Berger code. In the proposed method, the current mode circuits are employed to reduce the cost of the Berger code implementation. This circuit has higher speed and less hardware complexity than conventional implementations of Berger code. According to the result of this article, the Power of current mode Berger has been reduced at a rate of 51%, and the area of current mode Berger has been reduced at a rate of 74.3% and also the total cost of the Berger circuit (Power*Delay*Area) has been reduced at a rate of 91%.

Keywords: Fault Tolerant, Redundancy, Current Mode Circuit, Residue Code, Berger Code.

تاریخ ارسال مقاله: ۱۳۹۷/۰۶/۱۷

تاریخ اصلاح مقاله: ۱۳۹۷/۰۹/۱۳

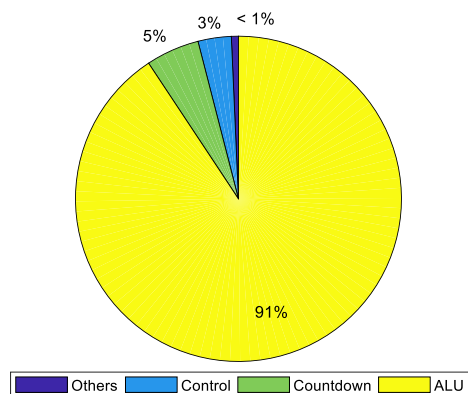
تاریخ پذیرش مقاله: ۱۳۹۷/۱۱/۲۰

نام نویسنده مسئول: رضا امیدی

نشانی نویسنده مسئول: ایران - زنجان - بلوار دانشگاه - دانشگاه زنجان

۱- مقدمه

در تحلیل انجام شده توسط Duba میزان انتشار و انتقال خرابی واحدهای مختلف مدار به دیگر واحدها نشان داده شده است که احتمال انتشار یک خطا که منشأ آن در واحد حساب و منطق است بسیار بیشتر از سایر بخش‌های یک پردازنده است به همین علت واحد محاسبه و منطق در پردازنده یکی از حساس‌ترین واحدها از نظر ایجاد خرابی در عملکرد یک پردازنده بشمار می‌رود. در شکل ۱ میزان احتمال انتقال خرابی به خروجی از واحدهای عملیاتی مختلف نشان داده شده است و همان‌طور که مشاهده می‌شود واحد حساب و منطق بحرانی‌ترین واحد پردازنده است.



شکل ۱. مقادیر احتمال انتشار خطا در پردازنده و انتقال به خروجی [۹].

روش‌های مختلفی جهت کاهش اثر خطای نرم در مدار وجود دارند که برخی از آن‌ها شامل استفاده از پکیج‌های مقاوم و روش‌هایی مانند اندازه‌گذاری دروازه‌های منطقی (تغییر نسبت طول به عرض کانال ترانزیستور جهت کاهش احتمال تولید خطا) است [۱۰]. متأسفانه استفاده از پکیج‌های مقاوم و حفاظت از مدار همیشه موفقیت‌آمیز نیست [۱۱]، بنابراین لازم است تا از روش‌هایی مانند معماری مقاوم در برابر خطا استفاده شود. کدهای آشکارسازی یا تصحیح خطا در بخش‌های مختلف مدار مانند حافظه، بخش‌های محاسباتی و... مورد استفاده قرار می‌گیرند اما یکی از چالش‌های استفاده از آن‌ها مصالحه میان سربار سخت‌افزاری و سرعت مدار است. در [۱۲] معماری خط لوله برای کد BCH پیشنهاد شده است که به بهبود سرعت مدار دیکدر و کاهش پیچیدگی سخت‌افزاری مدار کمک می‌کند.

یکی از کدهای محبوب برای تشخیص خطا در مدارهای محاسباتی کد برگر است. در کد برگر نیز سربار سخت‌افزاری و تأخیر مدار به‌عنوان چالش‌های طراحی مطرح هستند. یکی از روش‌های پیاده‌سازی این کد استفاده از مدار آستانه برنامه‌پذیر است [۱۳]. اگرچه این روش می‌تواند تا حد مناسبی پیچیدگی مدار را کاهش دهد اما فقط برای مدارهای با تعداد بیت داده کم مناسب هستند و در مدارهایی که تعداد بیت ورودی آن‌ها زیاد است از این مدار نمی‌توان استفاده کرد.

امروزه خطای نرم به یکی از بزرگ‌ترین چالش‌ها در طراحی مدارهای با کاربرد خاص مانند صنایع نظامی و صنایع فضایی تبدیل شده است بنابراین طراحی مدارها مقاوم در برابر خطا یک مسئله مهم در طراحی مدارها بشمار می‌رود. از دهه پنجاه میلادی خطای مدارهای دیجیتال در فضای خارج از جو گزارش می‌شد [۱، ۲]. تا چندین سال تصور می‌شد که این پدیده ناشی از خطای مانیتورینگ یا نمایشگرهای وضعیت باشد اما در اواسط دهه ۷۰ میلادی دلیل این پدیده توسط محققین کشف شد. در دهه ۸۰ روش‌های طراحی ماسک خطا به یکی از مسائل مهم در طراحی مدارهای الکترونیک تبدیل شد [۳]. در گذشته چندان به این مبحث پرداخته نمی‌شد اما امروزه به دلیل ظهور فرایندهای توسعه فناوری از قبیل معماری‌های سیستم بر تراشه، کاهش ابعاد ترانزیستور، و کاهش ولتاژ تغذیه حساسیت سیستم‌ها به خطای نرم افزایش یافته است. بعلاوه تحقیق‌های اخیر نشان می‌دهد که با کاهش ابعاد ترانزیستور عرض پالس تولید شده توسط خطای نرم افزایش می‌یابد [۴، ۵]. از سوی دیگر نیاز روزافزون به تولید مدارهای پردازنده با بازدهی و سرعت بالا صاحبان صنایع را مجبور به استفاده از آخرین فناوری در تراشه‌های دیجیتال کرده است.

در تحقیق انجام‌شده توسط Mahatme و همکارانش خطای SER در مدار منطقی شامل ۱۰ گیت معکوس‌کننده، در حدود ۲۰ درصد مقدار خطای مدار لچ در فرکانس ۵۰۰ مگاهرتز در فناوری ۲۰ نانومتر است در حالی که در فناوری ۴۰ نانومتر این نسبت در حدود ۱۰ درصد است. در فرکانس‌های بالاتر مقدار خطای SER در مدارهای منطقی در مقایسه خطای لچ، مقدار قابل توجهی است؛ بنابراین در فناوری‌های جدیدتر به روش‌هایی جهت کاهش خطا در مدارهای منطقی نیاز است [۶]. کاهش انرژی مصرفی مدارهای ریزپردازنده به‌عنوان مثال کاهش ولتاژ تغذیه، بر نرخ SEU تأثیر منفی دارد. روش‌هایی مانند ترانزیستورهای back-bias نیز تأثیری چندانی بر نرخ خطای SEU ندارند [۷]. در [۸] نشان داده شده است که ۸۰٪ از خطاهای پردازنده از نوع گذرا هستند.

تحلیل تجربی توسط Duba و Lyster بر روی حساسیت ریزپردازنده کنترل‌کننده موتور جت انجام شد [۹]. در این تحقیق از ابزارهایی جهت پایش و زیر نظر گرفتن سیگنال‌ها استفاده شده است و ۱۰۵۰ خطای گذرا به مدار تزریق شده است سپس نتایج خطاها بر اساس موقعیتشان و توانایی آن‌ها در تخریب داده دسته‌بندی شده‌اند. خطاهای واحد محاسبه و منطق اغلب منجر به خطا در پایه‌ها و به‌خصوص قطعه‌های جانبی پردازنده می‌شود. در تحقیق مذکور نشان داده شده است که واحد محاسبات و منطق و واحد تولید پالس ساعت بیشترین تأثیرپذیری را در برابر خطا دارند. ریزپردازنده مورد بررسی در [۹] دارای هفت بخش شامل واحد محاسبات و منطق، واحد کنترل، دیکدر، واحد تولید پالس ساعت و سیستم Watch Dog است.

محاسبه و منطق تحمل‌پذیر خطا نسبت به مدار مبتنی بر افزونگی دوتایی یا سه‌تایی مناسب‌تر است [۱۵].

کد برگر که در سال ۱۹۶۱ معرفی شد، یکی از کدهای مناسب جهت آشکار سازی خطا در واحدهای محاسباتی پردازنده است که برای عملیات منطقی و حسابی استفاده می‌شود. کد برگر کلمه‌های رمز مورد نیاز برای محافظت از ALU را کاهش می‌دهد و هزینه تمام‌شده را کم می‌کند. برای محاسبه کد برگر دو نماد B0 و B1 مورد استفاده قرار می‌گیرد. نماد B0 عبارت است از تعداد صفرهای کلمه داده که به شکل عدد دودویی نمایش داده می‌شود و نماد B1 نشان‌دهنده تعداد بیت‌های یک کلمه داده است. طول نماد آزمایش‌کننده (K) برابر است با $\log_2(N+1)$ که N برابر تعداد بیت‌های کلمه داده است. برای مثال، برای یک داده ۳۲ بیتی به ۶ بیت نماد آزمایش‌کننده برگر نیاز است. کد برگر قادر به آشکار سازی خطای تک بیت و تمام خرابی‌های یک سوپه است. خرابی‌های یک سوپه، فقط صفرها را به یک و یا یک‌ها را به صفر تبدیل می‌کند. اگر هر دو یک و صفرها تغییر کنند، لزوماً خرابی آشکار نخواهد شد. در [۱۶] کد برگر و کد همینگ با یکدیگر مقایسه شده‌اند نتایج مقایسه نشان می‌دهد که به دلیل این که کد برگر تنها خطاهای نامتقارن و تک جهته را آشکار می‌کند استفاده از آن در مدارهایی مناسب است که خروجی‌های مستقل از هم داشته باشند و کد همینگ نیز برای مدارهایی که تعداد خروجی‌های تحت تأثیر خطا زیاد است ($d \geq 3$)، مناسب نیست. بنابراین کد برگر قادر به تشخیص خطاهای نامتقارن است و در تشخیص خطای متقارن دچار مشکل خواهد شد. در [۱۷] روشی ارائه شده است که به وسیله آن مدارهای ترکیبی که با استفاده از کد برگر قابل آزمودن نیستند به مدارهایی تبدیل می‌شوند که کد برگر قادر به آزمودن آن‌ها باشد [۱۷]. با این الگوریتم می‌توان مشکل خطاهای متقارن در کد برگر برای مدارهای ترکیبی را کاهش داد. کد برگر در ارسال داده نیز بکار گرفته می‌شود بهترین حالت در استفاده از کد برگر از نظر بازده کدگذاری، زمانی است که طول داده برابر با $2^a - 1$ بیت باشد چون در این حالت بیت‌های داده کد شده به صورت چند بایت کامل در خواهد آمد و هنگام ارسال تمامی بایت‌های کد کامل خواهند بود [۱۸]. در [۱۹] کد برگر برای مقاوم سازی پردازنده DLX RISC در برابر رویداد تک خطا (SEU) برای واحد محاسبه و منطق در نظر گرفته شده است چون این کد روشی مناسب برای آشکار سازی خطا در مدارهای محاسباتی محسوب می‌شود [۱۹].

کد برگر برای بخش‌های مختلف واحد محاسبه و منطق با رابطه (۱) محاسبه می‌شود. این مثال حاصل جمع دو عدد N بیتی X و Y را محاسبه می‌کند. در این رابطه $X = (x_1, \dots, x_{N-1}, x_N)$ و $Y = (y_1, \dots, y_{N-1}, y_N)$ ورودی‌های جمع‌کننده هستند و $S = (s_0, \dots, s_{N-1}, s_N)$ خروجی جمع‌کننده است و $C = (c_1, \dots, c_{N-1}, c_N)$ رقم‌های نقلی داخلی جمع‌کننده هستند.

چون در مدار آستانه میزان تغییر ولتاژ سیگنال ناشی از خطا، با افزایش تعداد بیت داده ورودی کاهش می‌یابد و برای داده با تعداد بیت ورودی زیاد ممکن است این تغییرات ولتاژ برای مدار آستانه محسوس نباشد. در [۱۴] با استفاده از دو مدار آستانه و دو مدار تقویت‌کننده از مشکل محدودیت تعداد ورودی در مدار آستانه کاسته شده است. نقش مدارهای تقویت‌کننده یا مقایسه‌کننده در این مدار افزایش حساسیت مدار نسبت به تغییرات ولتاژ کم مدار آستانه است تا در صورت افزایش تعداد بیت ورودی و به دنبال آن کاهش تغییرات ولتاژ در مدار آستانه بازهم مدار آستانه قادر به تشخیص تغییرات ولتاژ باشد. اگرچه این روش تا حدی مشکل را برطرف می‌کند اما احتیاج به طراحی تقویت‌کننده متناسب با مدار آستانه دارد و ضریب تقویت آن وابسته به تعداد ورودی مدار آستانه است که موجب پیچیدگی مدار می‌شود. از سویی برخی ویژگی‌های تقویت‌کننده مانند تأخیر، Slew_rate، آفست، و پیچیدگی سخت‌افزاری مدار تقویت‌کننده در طراحی در نظر گرفته نشده است. در این مقاله با استفاده از مدارهای حالت جریان تلاش شده است تا مشکل وابستگی مدار به تعداد بیت‌های ورودی و پیچیدگی مدار کاهش یابد. در روش‌های پیشین ترانزیستورها نقش مقاومت‌های موازی با یکدیگر را بازی می‌کردند و ایجاد خطا در داده ورودی موجب تغییر ولتاژ گره میانی این مقاومت‌ها می‌شد که طبیعتاً وابسته به تعداد مقاومت‌ها یا به عبارتی ورودی‌ها است اما در این مقاله منابع جریان استفاده شده‌اند و جریان خروجی گره نشانگر وجود یا عدم وجود خطا است و جریان گره چون از قانون جریان کیرشهف پیروی می‌کند تغییرات آن کاملاً خطی و متناسب با عدم توازن ناشی از خطا است و وابستگی به تعداد بیت‌های ورودی ندارد.

در این مقاله مدارهای آزمون متفاوتی برای مقاوم سازی مدار ALU استفاده شده است که به کمک معماری مورد استفاده از انتشار خطا در دیگر واحدها جلوگیری می‌کنند. در بخش دوم مقاله پیش‌زمینه‌ای در رابطه با مدارهای آزمون مختلف و معماری ALU بیان شده است. در بخش سوم به روش‌های پیاده سازی کد برگر و مدار برگر حالت جریان پیشنهادی مقاله مطرح شده است. در بخش چهارم نتایج مقاله ارزیابی شده است و در نهایت در بخش پنجم نتیجه مقاله پرداخته شده است.

۲- پیش زمینه و کارهای پیشین

چهار افزونگی وجود دارد که در پیاده سازی ماسک خطا معمولاً یکی از این چهار نوع انتخاب و پیاده سازی می‌شود. چهار نوع مختلف افزونگی شامل افزونگی زمانی، افزونگی اطلاعات، افزونگی نرم‌افزاری و افزونگی سخت‌افزاری است [۳]. در اغلب الگوریتم‌های کدگذاری داده تولید شده یا ارسال شده با بیت‌های افزونگی که با داده صحیح مرتبط هستند بررسی می‌شوند و در صورتی که همچنان رابطه میان داده اصلی و داده افزونه برقرار باشد داده دریافتی صحیح است. در اکثر موارد داده افزونه نسبت به داده اصلی کوتاه‌تر است [۱۵]. به عنوان مثال استفاده از کد برگر Berger Check Prediction (BCP) جهت طراحی یک واحد

استفاده از مبنای باقیمانده ۳ می‌توان روابط را ساده کرد و به صورت نشان داده شده در جدول ۳ بازنویسی کرد.

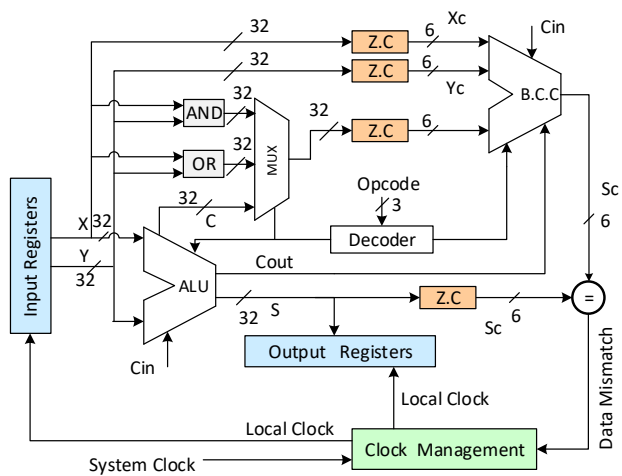
جدول ۲. روابط محاسبه باقیمانده برای اعمال ریاضی و منطقی برای مبنای باقیمانده b[۲۲].

روابط واحد محاسبه باقی‌مانده	عملیات اجرایی
$\ X\ _b + \ Y\ _b$	جمع
$\ X\ _b - \ Y\ _b$	تفریق
$\ X \text{ and } Y\ _b$	AND
$\ X\ _b + \ Y\ _b - 2\ X \text{ and } Y\ _b$	XOR
$\ X\ _b + \ Y\ _b - \ X \text{ and } Y\ _b$	OR

جدول ۳. روابط محاسبه باقیمانده خروجی واحد محاسبه و منطق از باقیمانده عملوندها

روابط واحد محاسبه باقی‌مانده	دستورهای واحد محاسبه و منطق
$- X _3$	چرخش به چپ
$ X _3$	چرخش به راست
$ X _3$	$S = X$

جهت حذف اثر خطا در خروجی مدار ALU به یک معماری مقاوم در برابر خطا احتیاج است. در این مقاله یک واحد ALU آسنکرون با قابلیت دفع اثر خطا مورد استفاده قرار گرفته است. این معماری در [۱۱] تحت عنوان Need-based-Intervention پیشنهاد شده است. در این معماری هنگامی که یک خطا آشکار شود، واحد مدیریت پالس ساعت، پالس ساعت را متوقف می‌کند تا ثبات‌ها اطلاعاتشان را تا زمان برطرف شدن خطا نگه‌دارند.



شکل ۲. نمودار بلوکی واحد محاسبه و منطق آسنکرون مبتنی بر کد برگز

(مقادیر x_i, y_i, s_i, c_i صفر یا یک هستند). مجموع i امین بیت از دو عملوند با رابطه (۱) توصیف می‌شود:

$$x_i + y_i + c_{i-1} = 2c_i + s_i = (s_i + c_i) + c_i \quad (1)$$

با فرض این که $N(X)$ نماینده تعداد یک‌ها در نمایش دودویی X و $N(x_i) = x_i$ باشد:

$$N(X) + N(Y) + c_{in} = N(S) + c_{out} + N(C) \quad (2)$$

که c_{in} رقم نقلی ورودی و c_{out} رقم نقلی خروجی است. نماد آزمایش‌کننده در ALU از نوع B0 در نظر گرفته شده است. بنابراین برای عدد N بیتی X ، نماد آزمایش‌کننده در کدگذاری B0 برابر است با $x_c = n - N(X)$ یا $N(X) = n - x_c$ با قرار دادن این عبارت در رابطه (۲)، رابطه زیر به دست می‌آید.

$$s_c = x_c + y_c - c_{in} + c_{out} - c_c \quad (3)$$

محاسبات مشابهی را می‌توان برای تفریق، عملیات منطقی، چرخش، جابجایی و ضرب آرایه‌ای انجام داد. عملیات پیاده سازی شده در ALU و الگوریتم کد برگز مربوط به آن در جدول ۱ ارائه شده است.

جدول ۱. الگوریتم‌های آزمایش‌کننده کد برگز برای ALU

روابط محاسبه کد برگز	دستورالعمل
$s_c = x_c + y_c - c_c - c_{in} + c_{out}$	جمع
$s_c = x_c - y_c - c_c - c_{in} + c_{out} + n$	تفریق
$s_c = x_c + y_c - (X \text{ or } Y)_c$	AND
$s_c = x_c + y_c - (X \text{ and } Y)_c$	OR
$s_c = x_c + y_c - 2(X \text{ and } Y)_c + n$	XOR
$s_c = x_c$	چرخش به راست
$s_c = x_c$	چرخش به چپ
$s_c = x_c$	انتقال X

۲-۱- کد باقیمانده:

یکی از اولین تحقیق‌های انجام شده برای کد باقیمانده در مدار جمع کننده، مربوط به پترسون است وی در سال ۱۹۵۸ بررسی صحت عملکرد مدار جمع کننده با استفاده از کد باقیمانده را پیشنهاد کرد [۲۰]. اگر مقدار R به صورت $R = 2^k - 1$ (برای مقادیر صحیح k) باشد، کد باقی‌مانده کم‌هزینه نامیده می‌شود، زیرا در این روش می‌توان مدار مولد باقیمانده را به صورت ساده‌تر پیاده‌سازی کرد [۲۱].

کد باقی‌مانده برای عملیات ریاضی عدد صحیح، از جمله جمع، تفریق، ضرب، تقسیم و ریشه دو قابل تعمیم است. همچنین می‌تواند برای عملیات AND، OR، XOR مورد استفاده قرار گیرد، اگر چه به دلیل ماهیت ریاضی باقیمانده، این کد گزینه مناسبی برای آزمون مدارهای منطقی بشمار نمی‌رود [۲۲]. جدول ۲ تعدادی از عملیات ریاضی و منطقی که استفاده بیشتری دارند را نشان می‌دهد. همچنین برای باقی‌دستورهای مورد استفاده مانند عملیات چرخش در صورت

دو عدد مدار آستانه استفاده شده است که هر کدام قادر است خطا در یک جهت مشخص را تشخیص دهد لذا برای تشخیص خطا باید هر دو مدار آستانه در مدار قرار گیرد تا بتواند خطا در هر دو جهت را آشکار سازد. به علاوه این مدار به یک دیگر نیاز دارد تا با توجه به مقدار کد برگر، ترانزیستور PMOS با وزن (W/L) متناسب را فعال کند. داده ورودی مقادیر X_i هستند و کد برگر مربوط به آن C_i هستند که به دیگر اعمال شده‌اند و مقادیر Q_1 و Q_2 خروجی‌های مدارهای آستانه هستند. برای پیاده‌سازی این مدار لازم است که ترانزیستورها به صورت دقیق طراحی شوند. همچنین در این مدار تعداد ورودی محدود است [۲۳]. در [۱۴] روشی جهت رفع مشکل محدودیت تعداد ورودی ارائه شده است که در آن از دو مدار آستانه قابل برنامه‌ریزی ساخته شده با ترانزیستورهای ما سفت استفاده شده است و دو عدد مقایسه کننده مقادیر ولتاژهای دو مدار را با یکدیگر مقایسه می‌کنند و عدم تطابق ولتاژ ناشی از بروز خطا در این دو مدار را آشکار می‌کنند. دو مدار آستانه استفاده شده از نظر اتصال ورودی‌ها عکس یکدیگرند بدین صورت که در یک مدار داده ورودی به PMOS و در دیگری داده ورودی به NMOS متصل است. در شکل ۴ مدار آزمون برگر مذکور نشان داده شده است. ولتاژ دو مدار باید با دو مقایسه کننده مقایسه شوند اما دقت مقایسه کننده تابعی از تعداد ورودی‌های مدار است و با افزایش تعداد بیت داده‌های ورودی میزان تغییرهای ولتاژ ناشی از ایجاد خطا کاهش می‌یابد و نیاز به مقایسه کننده دقیق‌تری خواهد بود از طرفی افزایش حساسیت مقایسه کننده منجر به افزایش ابعاد مدار مقایسه کننده و پیچیدگی طراحی آن می‌شود. در این مدار دو داده a و b با یکدیگر مقایسه می‌شوند، در صورتی که $a > b$ باشد آنگاه $V_{TH1} > V_{TH2}$ و خروجی تقویت کننده دوم فعال می‌شود همچنین اگر $a < b$ شود خروجی تقویت کننده اول فعال می‌شود و در صورتی که $a = b$ باشد خروجی هر دو تقویت کننده برابر با صفر می‌شود بنابراین خروجی گیت XOR که نشانگر خطا است، نیز صفر می‌شود. (جزئیات بیشتر در مرجع [۱۴] ارائه شده است).

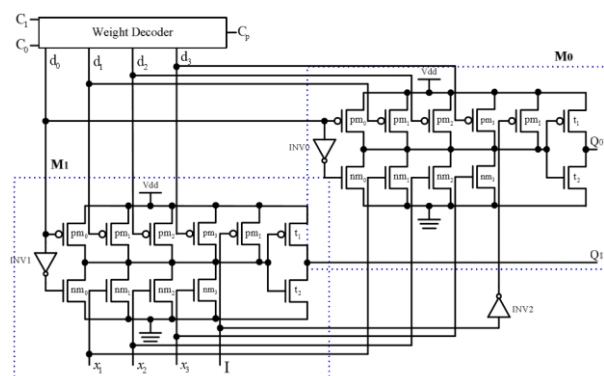
اگرچه در این روش تلاش شده است تا حاشیه نویز بهبود یابد اما تغییر ولتاژهای $V_{TH1,2}$ بر اثر خطا با تعداد ورودی مدار برگر رابطه عکس دارد، بنابراین با افزایش تعداد ورودی‌های مدار تغییر ولتاژ در ورودی‌های تقویت کننده که ناشی از عدم تساوی داده‌های ورودی است، کاهش می‌یابد. لذا بایستی با افزایش تعداد ورودی مدار بهره تقویت کننده‌ها افزایش یابد. از سویی افزایش بهره مدارهای تقویت کننده منجر به افزایش حساسیت مدار به خطای فتاوری در ترانزیستورها و همچنین نویز می‌شود، همچنین افزایش بهره تقویت کننده نیاز به یک مدار تقویت کننده پیچیده‌تر دارد که باعث پیچیدگی مدار برگر می‌شود. کد برگر به دلیل این که با واحدهای محاسباتی ریاضی مانند جمع و شمارش تولید می‌شود، این قابلیت را دارد که با استفاده از مدارهای حالت جریان پیاده‌سازی شود.

در این معماری به یک واحد آشکار ساز خطا نیاز است تا وجود خطا را تشخیص داده و فرایندهای لازم برای جلوگیری از انتشار خطا را اجرا کند. در [۱۵] واحد آشکار ساز خطا برمبنای کد برگر دیجیتال طراحی شده است. نمودار بلوکی این معماری در شکل ۲ نمایش داده شده است (واحد Z.C نشان دهنده شمارشگر صفر و واحد B.C.C واحد محاسبه کد برگر است).

بلوک محاسبه آزمون برگر، مسئول فعال کردن پرچم خطا به هنگام آشکار سازی یک خرابی نرم است. هنگام آشکار سازی یک خرابی نرم پالس ساعت کل سیستم تا زمان رفع اثر خطا خاموش می‌شود. این کار باعث می‌شود که همه ثبات‌ها مقادیرشان را برای یک دوره پالس ساعت متوقف کنند. این کار باعث می‌شود دستور اجرایی قبلی نگه داشته شود تا رخداد واحد گذرا (Single Event Transient (SET) از بین برود. اگر پالس خطای گذرا بیش تر از یک پالس ساعت طول بکشد، توقف پالس ساعت باید تا زمانی که خطا بر روی داده خروجی تأثیر می‌گذارد ادامه یابد که منجر به کاهش کارایی می‌شود که این کاهش کارایی وابسته به تعداد وقوع خرابی‌های نرم است. واحد مدیریت پالس ساعت، پالس ساعت کل سیستم را دریافت می‌کند و سیگنال‌های پالس ساعت محلی را برای مدار محافظت شده فراهم می‌کند. در این مقاله روش پیشنهادی برای شیوه پیاده سازی جدید واحد آشکار ساز خطا برمبنای کد برگر حالت جریان ارائه شده، و با پیاده‌سازی مبتنی بر کد برگر سنتی و کد باقیمانده مقایسه شده است.

۳- پیاده‌سازی کد برگر

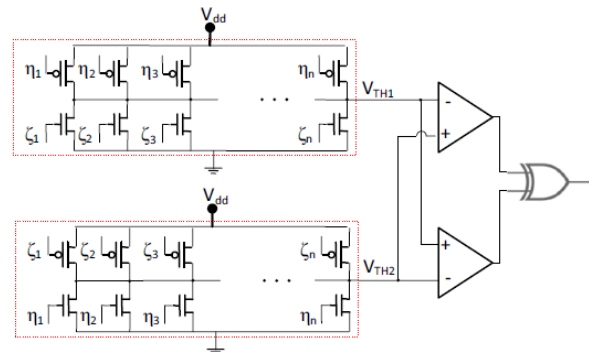
در تحقیق‌های اخیر تلاش‌هایی جهت کاهش سربار سخت‌افزاری مدار آزمون برگر صورت گرفته است [۱۳، ۱۴]. یکی از مدارهای پیشنهادی در این زمینه مبتنی بر مدارهای آستانه قابل برنامه‌ریزی است که در آن ترانزیستورهای ماسفت به گونه‌ای قرار می‌گیرند که در صورت برهم خوردن توازن کد برگر (که با ترانزیستورهای PMOS وزن دار اعمال می‌شوند) و داده اصلی که به گیت‌های NMOS اعمال شده‌اند، گیت NOT به حد آستانه برسد و خروجی تغییر وضعیت دهد.



شکل ۳. مدار آستانه برای کد برگر [۱۳، ۲۳]

در شکل ۳ شکل کلی از این مدار نشان داده شده است. با توجه به شکل

آنالوگ صحت داده دریافتی بررسی می‌شود. مدار تفریق حالت جریان را می‌توان با استفاده از منابع جریان PMOS و NMOS طراحی کرد که در حقیقت بخش PMOS را می‌توان به‌عنوان بخش مثبت و NMOS را به‌عنوان بخش منفی تساوی در نظر گرفت. منابع جریان PMOS ساطع کننده جریان و منابع جریان NMOS خورنده جریان هستند که با توجه به این ویژگی می‌توان با اتصال خروجی منابع جریان PMOS به منابع جریان NMOS حاصل تفریق دو مقدار را به دست آورد. جدول ۴ برای بلوک‌های جریانی مورد استفاده در مدار این مقاله را به همراه توضیحات و ساختار مداری آن‌ها نمایش می‌دهد. مدار آینه جریان برای تکرار جریان اصلی استفاده می‌شود و سوییچ دو حالتی متشکل از دو ترانزیستور برای کنترل جهت جریان بکار می‌رود. همچنین مدار آستانه برای مقایسه جریان با یک حد آستانه استفاده می‌شود و خروجی آن از نوع ولتاژ است.



شکل ۴. مدار برگر مبتنی بر مقایسه کننده (IBCC) [۱۴]

مدارهای حالت جریان از دیرباز در پیاده‌سازی مدارهای ریاضی مانند ضرب کننده، جمع کننده مبنای چهار، مدارهای محاسبه حداکثر و حداقل و دیگر مدارهای ریاضی مورد استفاده قرار می‌گیرند [۲۴-۲۶] و در بسیاری از موارد نسبت به هم‌تای دیجیتال خود بهتر عمل می‌کنند. در طراحی پیشنهادی مبتنی بر مدارهای حالت جریان به دلیل مقاومت بالای خروجی منابع جریان وابستگی مدار به تعداد بیت ورودی تا حد امکان کاهش می‌یابد و از سوی دیگر مدار تقویت کننده از مدار حذف می‌شود.

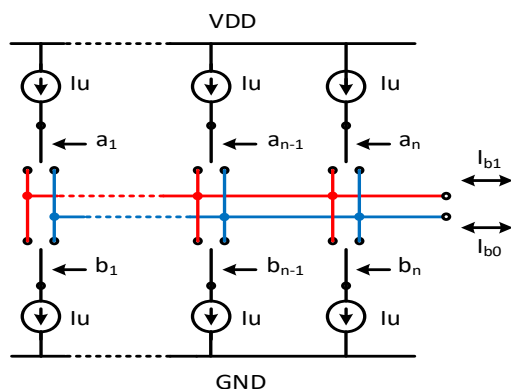
مدار پیشنهادی مقاله به‌گونه‌ای طراحی شده است که در صورت تغییر جریان خروجی مدار و تجاوز تغییرات آن از نصف دامنه جریان منابع واحد، خروجی مدار فعال شود و وجود خطا را در خروجی آشکار کند. لذا در این مدار نیز مانند مدارهای دیجیتال حاشیه یا حد نویز تعریف می‌شود. با توجه به مطالب مذکور برای مدار برگر حالت جریان نویز باید به‌گونه‌ای باشد تا از نصف جریان واحد (جریان واحد جریان هر کدام از منابع جریان است) کمتر باشد. در مدارهای برگر مبتنی بر حالت ولتاژ (دودویی حالت ولتاژ) حاشیه نویز وابسته به ولتاژ کاری مدار و ولتاژ آستانه ترانزیستور است که با پیشرفت فناوری هر دو مورد در حال کاهش هستند بنابراین شکل نویز و حاشیه نویز در مدارهای حالت ولتاژ روز به روز بیشتر می‌شود. این در حالی است که در مدار حالت جریان می‌توان با سطوح ولتاژ پایین حتی تا چندین سطح جریان قابل تفکیک تولید کرد به‌گونه‌ای که این سطوح به حد کافی از یکدیگر فاصله داشته باشند تا حاشیه نویز افزایش یابد و نویز نتواند خروجی مدار مقایسه جریان یا آستانه جریان را دچار تغییر کند. و به‌طور کلی مدار حالت جریان با ولتاژ کاری کم سازگار است و در کاهش ولتاژ کاری از نظر نویز نسبت به مدارهای حالت ولتاژ محدودیت کمتری دارد [۲۷].

۴- مدار برگر حالت جریان

در مدار برگر حالت جریان Current Mode Berger code Checker (CMBC) برای حذف بخش شمارنده بیت‌ها از مدارهای آنالوگ حالت جریان استفاده می‌شود مدار آزمون CMOS کد برگر بر اساس تساوی تعداد بیت‌های خروجی و ورودی عمل می‌کند، در این مقاله با تبدیل تساوی به تفریق و سپس بررسی صفر بودن حاصل تفریق با مدارهای

جدول ۴. بلوک‌های مورد استفاده حالت جریان

نام مدار	مدار	تابع عملکرد	نماد
کلید متقارن		If D = Active then out2 = I _r else out1 = I _r	
آینه جریان PMOS		$y_1 = \dots = y_n = -x$	
آینه جریان NMOS		$y_1 = \dots = y_n = -x$	
مدار آستانه		If $x > T$ then $y = H$ else $y = L$	



شکل ۵. مفهوم کلی مدار آزمون حالت جریان « I_{bias} » برای بایاس مدار در ولتاژ و جریان بهینه به کار گرفته شده است.

۴-۱- واحد محاسبه و منطق مقاوم شده با برگر حالت جریان

برای استفاده از کد برگر حالت جریان در مدار محاسبه و منطق لازم است تغییرهای جزئی در روابط مربوط به کد برگر صورت گیرد. به‌عنوان مثال در پیاده‌سازی کد برگر با پیاده‌سازی CMOS برای عملیات XOR از رابطه (۴) استفاده می‌شود.

$$s_c = x_c + y_c - 2(X \text{ and } Y)_c + n \quad (4)$$

در رابطه (۴) مقدار ثابت n استفاده شده است که برای پیاده‌سازی آن به‌صورت مدارهای جریانی به یک منبع جریان ثابت با وزن n نیاز است که منجر به اشغال فضای تراشه و افزایش توان مصرفی مدار می‌شود. همچنین پیاده‌سازی عملیات ضرب در ۲ در رابطه (۴) به‌صورت مدار جریانی بهینه نیست. لذا برای عملیات XOR رابطه (۴) به صورت رابطه (۵) بازنویسی می‌شود.

$$s_c = (X \text{ or } Y)_c + N(X \text{ and } Y) \quad (5)$$

کد برگر حالت جریان (CMBC) تساوی را با استفاده از صفر بودن تفریق بخش مثبت و منفی بررسی می‌کند؛ بنابراین برای نمایش بهتر رابطه (۵) را می‌توان به‌صورت رابطه (۶) بازنویسی کرد.

$$s_c - (X \text{ or } Y)_c - N(X \text{ and } Y) = 0 \quad (6)$$

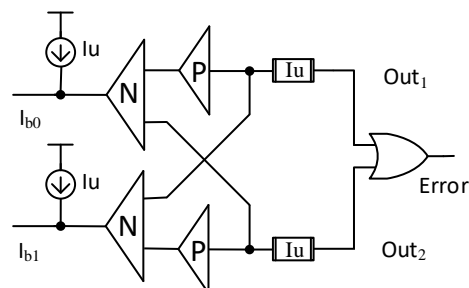
به‌عنوان مثال در رابطه (۶) $\{s_c\}$ به‌عنوان ورودی بخش مثبت مدار CMBC و $\{(X \text{ or } Y)_c, N(X \text{ and } Y)\}$ به‌عنوان ورودی های بخش منفی مدار در نظر گرفته می‌شوند. تغییرهای مشابهی برای دیگر عملیات ALU لازم است که در جدول ۵ نمایش داده شده است.

جدول ۵. الگوریتم‌های مدار آزمون حالت جریان کد برگر برای ALU

ورودی‌های مدار آزمون حالت جریان		دستورالعمل
مثبت	منفی	
s_c, c_c, c_{in}	x_c, y_c, c_{out}	جمع
s_c, c_c, c_{in}	$x_c, c_{out}, N(Y)$	تفریق
s_c	$(X \text{ and } Y)_c$	AND
s_c	$(X \text{ or } Y)_c$	OR
s_c	$(X \text{ or } Y)_c, N(X \text{ and } Y)$	XOR
s_c	x_c	چرخش
s_c	x_c	انتقال X

همچنین برای تعبیه مدار CMBC در ALU لازم است تغییراتی در مدارهای انتخاب‌گر ورودی انجام شود. در مدار پیشنهادی نمایش داده شده در شکل ۷ دو مالتی پلکسر برای انتخاب ورودی‌های بخش مثبت استفاده شده‌اند که وظیفه آن‌ها انتخاب ورودی‌ها برای روابط ریاضی و یا انتخاب خروجی توابع منطقی برای فرمان‌های منطقی است.

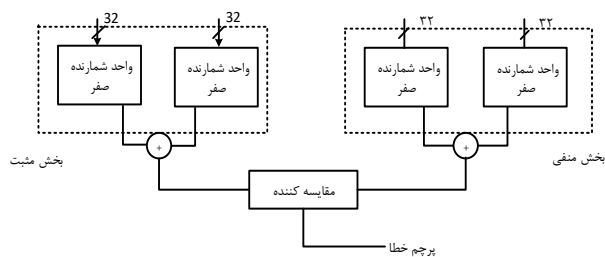
در مدار برگر حالت جریان در مسیر هر یک از منابع جریان کلیدی قرار داده شده است که به‌وسیله آن می‌توان جریان را هدایت کرد. مقادیر منابع جریان $Iu = 10uA$ است (شکل ۵). همان‌طور که گفته شد مقایسه تعداد یک‌ها و صفرها با استفاده از تفریق جریان منابع جریان مثبت و منفی (منابع جریان PMOS و NMOS) انجام می‌گیرد؛ که مقایسه تعداد صفرها و یک‌ها به دلیل وجود زوج کلیدهای متقارن به‌صورت هم‌زمان انجام می‌گیرد. در حالت عملکرد عادی مدار (داده‌های ورودی بدون خطا) تعداد منابع جریان مثبت و منفی که به گره B1 متصل‌اند برابر است و همین شرایط برای گره B0 نیز برقرار است بنابراین $I_{b1} = I_{b0} = 0$. حال اگر یک خطای صفر به یک در داده ورودی منابع جریان مثبت رخ دهد یک منبع جریان مثبت از گره B0 جدا شده و به B1 متصل خواهد شد و تعادل در دو گره مدار برهم خواهد خورد و در این حالت مقدار I_{b1} دیگر برابر صفر نخواهد بود و مقدار مثبتی خواهد داشت و I_{b0} نیز منفی خواهد بود. برای خطای صفر به یک نیز عکس این رویدادها رخ خواهد داد. در نهایت برای تولید سیگنال نشانگر خطا از یک مدار آشکار ساز جریان یا مدارهای مقایسه جریان استفاده شده است تا در صورتی که جریان‌های I_{b1} و I_{b0} برابر صفر نباشند نشانگر خطا فعال شود. در شکل ۶ ساختار مدار آشکار ساز جریان نمایش داده شده است.



شکل ۶. مدار آشکار ساز جریان

خروجی مدار مقایسه کننده در حالت عملکرد بدون خطا باید برابر صفر باشد و در صورت بروز یک خطای یک به صفر در بخش مثبت مدار مقایسه کننده و یا یک خطای صفر به یک در بخش منفی مدار (بخش مثبت منابع جریان PMOS و بخش منفی منابع جریان NMOS هستند) خروجی Out1 برابر با یک می‌شود و خروجی Out2 در حالت صفر باقی می‌ماند به همین صورت با تغییر صفر به یک در بخش مثبت مدار مقایسه کننده و یا تغییر یک به صفر در بخش منفی مدار خروجی Out2 برابر یک و خروجی Out1 برابر صفر می‌شود. برای تولید یک سیگنال برای نمایش وقوع خطا کافی است یک گیت OR یا XOR در خروجی قرار گیرد تا در صورت فعال شدن هر کدام از دو حالت بالا سیگنال نمایشگر خطا فعال شود. در صورتی که خطایی رخ ندهد اختلاف جریان‌ها برابر صفر می‌شود بنابراین ورودی و خروجی بلوک‌های آستانه نیز برابر صفر می‌شود و مقدار سیگنال نمایشگر خطا نیز برابر صفر می‌شود (مقدار جریان بایاس حداقل باید به اندازه منابع جریان واحد (Iu) باشد).

مصرفی مدار آزمون حالت جریان ۷۴/۳ در صد کمتر از سطح مصرفی مدار معادل دیجیتال است.



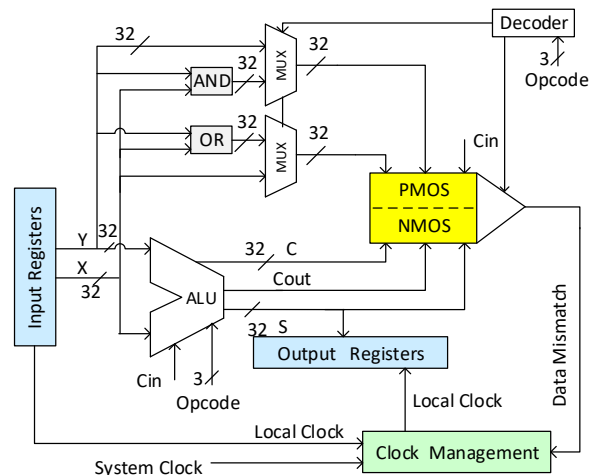
شکل ۸. نمودار بلوکی معادل مدار آزمون برگر

جدول ۶. نتایج سنتز و شبیه‌سازی مدار آزمون برگر حالت جریان و مدار دیجیتال معادل آن.

برگر حالت جریان	مدار برگر آستانه [۲۳]	مدار CMOS	برگر حالت جریان	ساختار مداری
۲	۲	۳۲	۳۲	تعداد بیت عملیات تکنولوژی (um)
۰/۱۸	۰/۳۵	۰/۱۸	۰/۱۸	ولتاژ (V)
۱/۸	۳/۳	۱/۸	۱/۸	تعداد ترانزیستور
۰/۶۶	--	۳/۴۳	۲/۴	تأخیر (ns)
۰/۷۰	۴/۸۳	۳/۱۳	۱/۵۳	توان مصرفی (mW)
--	--	۳۹۵۶۱	۳۴۷۳	سطح اشغالی* توان مصرفی* تأخیر

با توجه به جدول ۶ مشاهده می‌شود که تعداد ترانزیستور مورد نیاز مدار برگر مبتنی بر مدار آستانه (که در مرجع [۲۳] برای یک جمع کننده دو بیتی استفاده شده است) بسیار بیشتر از تعداد ترانزیستور مورد نیاز جهت پیاده‌سازی همان مدار با برگر حالت جریان است و تعداد ترانزیستور مورد نیاز مدار جریان تا ۵۴ درصد کمتر از مدار آستانه است. توجه شود که برای عملیات جمع دو بیتی هر داده ورودی، خروجی و بیت‌های نقلی داخلی مدار جمع کننده دو بیت هستند که با در نظر گرفتن مقادیر نقلی ورودی و خروجی مدار برگر در مجموع ۱۰ بیت داده را مدار آزمون دریافت می‌کند.

در شکل ۹ (الف) تأثیر تغییرات فرایند بر تأخیر مدار برگر حالت جریان ۳۲ بیتی نمایش داده شده است. برای ارزیابی تأثیر تغییرات فرایند از شبیه‌سازی مونت کارلو استفاده شده است و در شبیه‌سازی از مقادیر استاندارد تعریف شده در کتابخانه TSMC18 برای تکنولوژی ۱۸۰ استفاده شده است. همان‌طور که مشاهده می‌شود تغییرات تأخیر خروجی مدار نسبت به تغییرات فرایند در بازه ۲/۱۵ تا ۲/۵۵ نانوثانیه با میانگین ۲/۴ و انحراف معیار استاندارد برابر ۸۹/۶ پیکو قرار می‌گیرد. در شکل ۹ (ب) منحنی انتقال خروجی به ازای تغییرات یک بیت از داده ورودی (تغییر یک بیت در حالتی که منجر به تغییر مقدار خروجی شود) و تأثیر تغییرات فرایند بر این منحنی نمایش داده شده است.



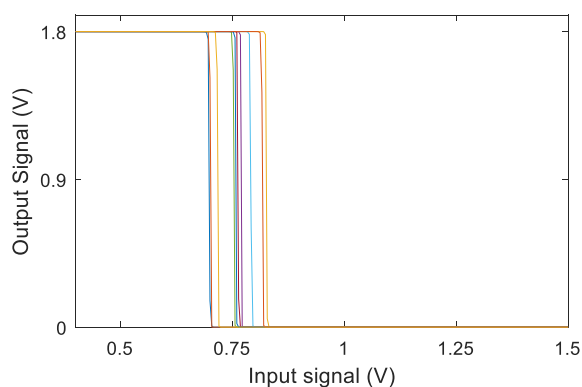
شکل ۷. نمودار بلوکی واحد محاسبه و منطق آسنکرون مبتنی بر کد برگر حالت جریان

۵- ارزیابی مدار

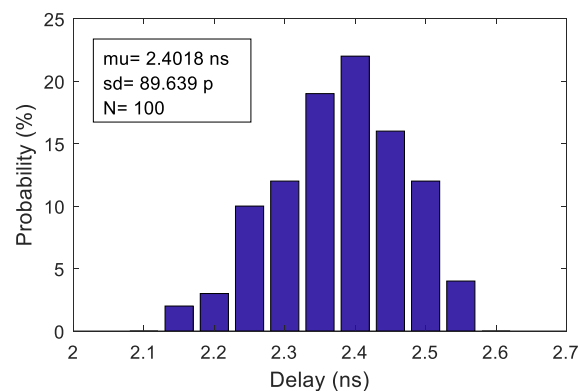
ارزیابی مدارهای مقاوم‌سازی شده شامل ارزیابی مدار از نظر هزینه پیاده‌سازی و مقدار مقاومت مدار در برابر خطا می‌شود. عملکرد این مدار شباهت زیادی به عملیات شمارش تعداد صفر برای چهار ورودی و تفریق دودسته منفی از دودسته مثبت است و در نهایت حاصل تفریق با مقدار صفر مقایسه می‌شود تا در صورت وجود خطا، پرچم نشانگر خطا فعال شود. مدار معادل مدار آزمون برگر حالت جریان با استفاده از چهار مدار شمارنده، دو عدد جمع کننده و در نهایت یک مقایسه کننده قابل پیاده‌سازی است. نمودار بلوکی مدار معادل CMOS برای مدار آزمون برگر ۳۲ بیتی حالت جریان در شکل ۸ نشان داده شده است.

جهت مقایسه عملکرد مدار آزمون برگر حالت جریان با مدارهای دیجیتال، مدار دیجیتال معادل آن به صورت شکل ۸ پیاده‌سازی شده است و نتایج تحلیل سخت‌افزاری آن با مدار آزمون حالت جریان مقایسه شده است. در جدول ۶ نتایج شبیه‌سازی و سنتز این مدارها در فرکانس ۱۰۰ مگاهرتز گزارش شده است (مدار حالت جریان با استفاده از نرم‌افزار Cadence و مدار معادل دیجیتال آن با استفاده از نرم‌افزار Design_compiler با تکنولوژی ۱۸۰ نانومتر سنتز شده‌اند).

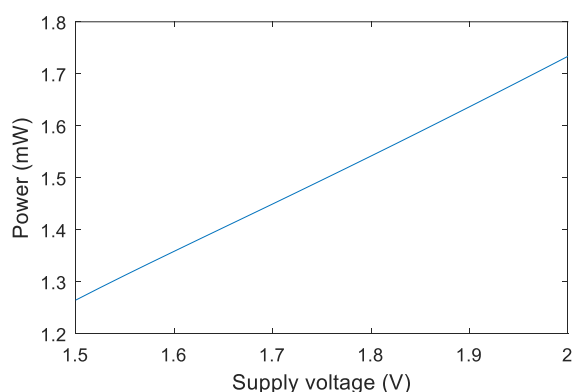
همان‌طور که در جدول ۶ مشاهده می‌شود مقدار تأخیر مدار آزمون حالت جریان به‌طور متوسط ۳۰ درصد کمتر از مدار آزمون دیجیتال معادل است (لازم به ذکر است مدار آزمون دیجیتال معادل با مدار حالت جریان از نظر تعداد بلوک سازنده تفاوت چندانی با مدار آزمون دیجیتال مورد استفاده واحد محاسبه و منطق که در بخش‌های قبل به آن پرداخته شد، ندارد چراکه برای آزمون مدار ALU بخش‌های قبل نیز چهار واحد شمارنده به همراه واحدهای جمع کننده و مقایسه کننده به کار گرفته می‌شود و در این مدار تنها چینش آن‌ها نسبت به یکدیگر اندکی تغییر کرده است تا نتایج مقایسه عادلانه‌تر شود). از نظر توان مصرفی، توان مصرفی مدار حالت جریان نسبت به مدار دیجیتال به‌طور متوسط تا حدود ۵۱ درصد کاهش یافته است. در تحلیل سطح



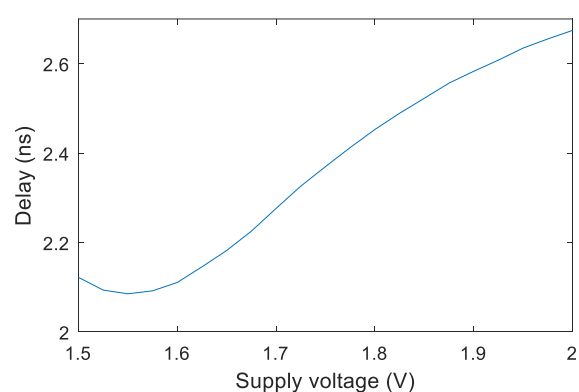
(ب)



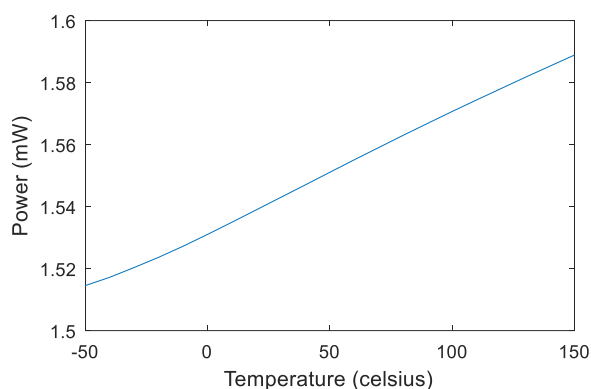
(الف)



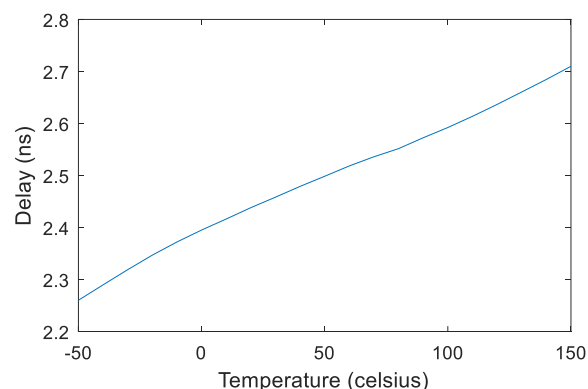
(ت)



(پ)



(ج)



(ث)

شکل ۹. تأثیر تغییرات فرایند، ولتاژ تغذیه و دما بر تأخیر و توان مصرفی مدار الف) تغییرات تأخیر بر اثر تغییرات فرایند ب) تغییرات منحنی انتقال در تحلیل DC مدار بر اثر تغییرات فرایند پ) تغییرات تأخیر با تغییر ولتاژ تغذیه ت) تغییرات توان مصرفی با تغییرات ولتاژ تغذیه ث) تغییرات تأخیر با تغییر دما ج) تغییرات توان مصرفی با تغییرات دما.

اینکه جریان مدار مشخص و ثابت است تأخیر شارژ و تخلیه سیگنال‌های داخلی مدار افزایش می‌یابد. در شکل ۹ (ت) تأثیر تغییرات ولتاژ تغذیه بر توان مصرفی مدار نمایش داده شده است. توان مصرفی مدار حالت جریان برخلاف مدارهای CMOS به صورت خطی با ولتاژ تغذیه افزایش می‌یابد. در شکل ۹ (ث) و (ج) تغییرات تأخیر به‌ازای

همان‌طور که مشاهده می‌شود با تغییرات فرایند مدار از عملکرد صحیح خود خارج نمی‌شود. در شکل ۹ (پ) تأثیر تغییرات ولتاژ تغذیه بر تأخیر مدار نشان داده شده است. با افزایش ولتاژ تغذیه برخلاف مدارهای حالت ولتاژ تأخیر مدار افزایش می‌یابد چون افزایش ولتاژ منجر به افزایش سطح ولتاژ سیگنال‌های داخلی مدار می‌شود و بدلیل

که ALU با مدار آزمون برگر حالت جریان در مقایسه با ALU با مدارهای آزمون کد باقیمانده، کد برگر دیجیتال، افزونگی دو تایی و افزونگی سه تایی به ترتیب ۳۲، ۲۵، ۳۲ و ۵۴ درصد فضای کمتری از تراشه را اشغال می‌کند. تأخیر ALU با مدار آزمون برگر حالت جریان در مقایسه با ALU با مدارهای آزمون کد باقیمانده و کد برگر دیجیتال به ترتیب ۲/۱٪ و ۷/۲٪ درصد کمتر است اما در مقایسه با افزونگی دو تایی و سه تایی تأخیر اندکی بیشتر است که دلیل آن ساده بودن مدار Voter است که مقایسه نتایج را در افزونگی سه تایی را بر عهده دارد. حاصل ضرب توان در تأخیر مدار مبتنی بر برگر حالت جریان در مقایسه با ALU با مدارهای آزمون کد باقیمانده، کد برگر دیجیتال، افزونگی دو تایی و افزونگی سه تایی به ترتیب ۲۳٪، ۲۳٪، ۴٪ و ۳۸٪ کمتر است.

جدول ۷. نتایج سنتز و شبیه‌سازی مدار ALU با مدارهای آزمون متفاوت (تکنولوژی ۱۸۰ نانومتر و ولتاژ ۱/۸ ولت)

روش مقاوم‌سازی	سطح اشغالی (mm ²)	تأخیر (ns)	توان مصرفی (mW)	PDP
کد باقیمانده	۶۲۱۶۱۷	۱۰/۷۷	۱۷/۳۴	۱۸۶/۷۶
کد برگر مبتنی بر [۱۵]	۵۶۰۲۵۷	۱۱/۳۶	۱۶/۴۵	۱۸۶/۸۷
افزونگی دو تایی	۶۱۹۵۹۵	۹/۸۹	۱۵/۱۸	۱۵۰/۱۳
افزونگی سه تایی	۹۱۵۱۶۹	۹/۹۴	۲۳/۱۲	۲۲۹/۸۱
CMBC	۴۲۰۸۴۲	۱۰/۵	۱۳/۶۱	۱۴۳/۴۵
بدون مقاوم‌سازی	۳۶۲۱۱۸	۹/۸۴	۹/۷	۹۵/۴۵

جهت بررسی میزان مقاومت واحد محاسبه و منطق در برابر خطا، ۱۰۰۰ خطا در زمان اجرای دستورها به نقاط مختلف مدار محاسبه و منطق تزریق شده است. در [۲۸] تحقیق و بررسی بر روی برنامه‌های محک انجام گرفته است و در آن نویسنده به دسته‌بندی دستورها از نظر نوع عملوندها، نوع دستورها و میزان پیچیدگی دستورها پرداخته است. در دسته‌بندی نوع دستورها نتایج نشان می‌دهد تعداد دستورهایی ریاضی در برنامه محک تا پنج برابر بیشتر از دستورهایی منطقی است. در این بخش جهت اعتبارسنجی مدار محاسبه و منطق دستورهایی تصادفی تولید شده و به مدار محاسبه و منطق اعمال می‌شوند و تعداد دستورهایی ریاضی (جمع و تفریق) در حدود چهار تا پنج برابر دستورهایی منطقی در نظر گرفته شده است. در مدار محاسبه و منطق آسنکرون با ایجاد هر خطا، مدار مجبور است حداقل به مدت یک سیکل کامل از پالس ساعت متوقف بماند تا اثر خطا برطرف گردد که این امر موجب ایجاد تأخیر در عملکرد مدار می‌شود لذا مدار ALU در شرایطی که احتمال بروز خطای با دوره زمانی طولانی در مدار کمتر است، عملکرد بهتری دارد. در شکل ۱۰ درصد خطاهای منجر به توقف پالس ساعت نسبت به کل خطاهای تزریق شده نشان داده شده است. همان‌طور که مشاهده می‌شود تعداد توقف‌های پالس ساعت با کاهش

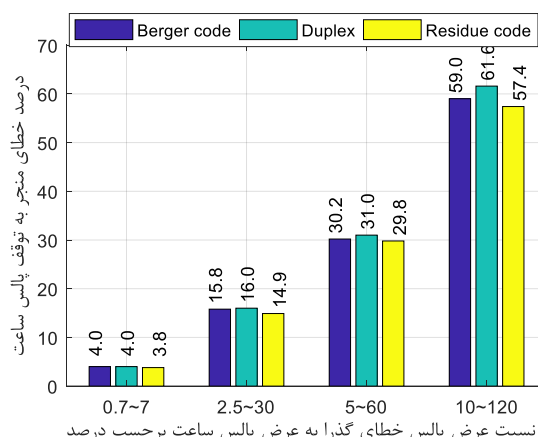
تغییرات دما نشان داده شده است. نتایج نشان می‌دهد که افزایش دما منجر به افزایش تأخیر و توان مصرفی مدار می‌شود.

در مرجع [۱۵] معماری Need-based-Intervention برای مدار محاسبه و منطق بر روی تراشه FPGA خانواده Cyclone II پیاده‌سازی شده است و مقدار فرکانس کاری برای مدار مقاوم سازی شده ۳۸/۶۶ و مدار بدون مقاوم‌سازی ۴۰/۸۸ به دست آمده است و همچنین مقدار المان‌های منطقی مورد استفاده در مدار مقاوم سازی شده ۷۳۷ عدد و برای مدار بدون مقاوم‌سازی ۲۷۸ گزارش شده است [۱۵]. در این تحقیق جهت مقایسه عادلانه تمامی مدارها با استفاده از تکنولوژی ۱۸۰ نانومتر پیاده‌سازی شده‌اند و همچنین مدار مورد استفاده در مرجع [۱۵] نیز با استفاده از تکنولوژی ۱۸۰ نانومتر پیاده‌سازی و با نتایج مقاله مقایسه شده است. در ادامه نتایج سنتز مدار محاسبه و منطق بدون مقاوم‌سازی، مدارهای مبتنی بر کد برگر، مدارهای مبتنی بر کد باقی مانده، مدار افزونگی پیمان‌های دو تایی و مدار افزونگی پیمان‌های سه تایی مورد بررسی قرار گرفته‌اند.

در مدار محاسبه و منطق بدون مقاوم‌سازی ثبات‌های ورودی و خروجی وجود دارند اما واحدهای آشکار سازی و تصحیح خطا استفاده نشده‌اند، در مدار مبتنی بر افزونگی پیمان‌های دو تایی آشکار سازی خطا با استفاده از مقایسه خروجی‌های دو واحد محاسبه و منطق صورت می‌گیرد و تصحیح آن با روش‌های مورد استفاده در کد برگر و کد باقی‌مانده (Need-based-Intervention) یکسان است. در مدار افزونه سه تایی آشکار سازی و تصحیح خطا با استفاده از مدار (محاسبه اکثریت) (Voter) انجام می‌گیرد. با توجه به نتایج به دست آمده از تحلیل مدار کاملاً مشهود است که افزونگی پیمان‌های سه تایی در مقایسه با دیگر روش‌های آشکار سازی و تصحیح خطا سربار سخت‌افزاری بالایی را می‌طلبد در نتایج این مقدار در حدود ۲۵۰ درصد به دست آمده است که اندکی کمتر از مقدار تئوری ۳۰۰ درصد است که می‌تواند به دلیل بهینه‌سازی انجام شده توسط ابزار سنتز و همچنین وجود اجزای مستقل از ساختار افزونگی پیمان‌های سه تایی مانند ثبات‌ها باشد چرا که با ایجاد افزونگی در مدار محاسبه و منطق که یک بخش ترکیبی در مدار است، بخش ترتیبی مدار دچار تغییر نشده است. همچنین با توجه به جدول ۷ مدار مقاوم شده با برگر حالت جریان کمترین سربار را به خود اختصاص داده است.

روش مبتنی بر افزونگی پیمان‌های دو تایی مقادیری قابل قیاس با مدار باقیمانده دارد که دلیل آن کوچک بودن و عدم پیچیدگی مدار محاسبه و منطق است که باعث می‌شود افزونگی پیمان‌های دو تایی و کد باقیمانده سربار مشابهی داشته باشند. در مدار افزونگی پیمان‌های سه تایی تصحیح خطا توسط مدار محاسبه اکثریت انجام می‌گیرد در این مدار به دلیل کم بودن تأخیر مدار محاسبه اکثریت و همچنین عملکرد هم‌زمان سه واحد افزونه تأخیر مدار افزونگی پیمان‌های سه تایی نسبت به مدار محاسبه و منطق بدون مقاوم‌سازی ناچیز است. با مقایسه نتایج به دست آمده از سنتز مدارهای ALU در جدول ۷ مشاهده می‌شود

ماندگاری خطا، کاهش می‌یابد که دلیل آن کاهش احتمال قرارگیری اثر خطا در لبه پالس ساعت ثبات‌ها است.



شکل ۱۰. درصد خطاهای منجر به توقف پالس ساعت به ازای مقادیر مختلف عرض پالس خطا در مدارهای مداخله در صورت نیاز.

۶- نتیجه‌گیری

روش جدید پیاده‌سازی مدار محاسبه و منطق مقاوم شده با کد برگر حالت جریان با سربرای سخت‌افزاری کمتر از انتشار خطا در مدار پردازنده جلوگیری می‌کند. هدف این مقاله طراحی روشی جهت پیاده‌سازی کد برگر با سرعت بالاتر، تأخیر کمتر و سطح اشغالی پایین‌تر است. در مقایسه با روش‌های قدیمی پیاده‌سازی کد برگر که احتیاج به فضای سخت‌افزاری زیاد داشتند و یا قابلیت توسعه برای عرض داده بزرگ را نداشتند مدار کد برگر حالت جریان قابل‌تعمیم به تعداد بیت بالاتر است و با افزایش عرض داده از دقت آن کاسته نمی‌شود.

مراجع

[7] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in *Reliability Physics Symposium (IRPS), 2011 IEEE International*, 2011, pp. 5B. 4.1-5B. 4.7: IEEE.

[8] R. K. Iyer and D. J. Rossetti, "A measurement-based model for workload dependence of CPU errors," *IEEE Transactions on Computers*, vol. 100, no. 6, pp. 511-519, 1986.

[9] P. Duba and R. Lyer, "Transient fault behavior in a microprocessor-A case study," in *IEEE International Conference on Computer Design* 1988, pp. 272-276.

[۱۰] ثابت سروستانی، محمدامین؛ بهنام قوامی و محسن راجی «کاهش نرخ خطای نرم چندگانه مدارهای ترکیبی مبتنی بر اندازه‌گذاری دروازه‌ها بر مبنای پارامتر حساسیت»، فصلنامه مهندسی برق دانشگاه تبریز، دوره ۴۷، شماره (۲)، صفحه ۴۴۵-۴۵۴، تابستان ۱۳۹۶.

[11] R. H. Maurer, M. E. Fraeman, M. N. Martin, and D. R. Roth, "RHarsh Environments: Space Radiation," *Johns Hopkins APL technical digest*, vol. 28, no. 1, p. 17, 2008.

[۱۲] نبی پور، سعیده؛ جواد جاویدان و غلامرضا زارع فتین «طراحی یک دیکدر BCH بهینه جهت افزایش اطمینان در ذخیره‌سازی اطلاعات و تصحیح خطا در حافظه‌های فلش»، فصلنامه مهندسی برق دانشگاه تبریز، دوره ۴۶، شماره (۳)، صفحه ۳۱۹-۳۳۱، پاییز ۱۳۹۵.

[13] X. Kavousianos, D. Nikolos, G. Foukarakis, and T. Gnardellis, "New efficient totally self-checking Berger code checkers," *INTEGRATION, the VLSI journal*, vol. 28, no. 1, pp. 101-118, 1999.

[14] R. Omidi and H. Zarrabi, "New Protection Technique Against Unidirectional MEUs for FIR Filters," *Circuits, Systems, and Signal Processing*, pp. 1-16, 2017.

[15] E. Ossi, D. Limbrick, W. Robinson, and B. Bhuvu, "Soft-error mitigation at the architecture-level using berger codes and instruction repetition," in *Proceedings of the IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE'09)*, 2009.

[16] V. Sapozhnikov, V. Sapozhnikov, D. Efanov, and A. Blyudov, "Analysis of error-detection possibilities of CED circuits based on Hamming and Berger codes," in *2013 11th East-West Design and Test Symposium (EWDTS)*, 2013, pp. 1-8: IEEE.

[17] V. Sapozhnikov, V. Sapozhnikov, and D. Efanov, "Search algorithm for fully tested elements in combinational circuits, controlled on the basis of berger codes," in *2017 IEEE East-West Design & Test Symposium (EWDTS)*, 2017, pp. 1-10: IEEE.

[18] F. Huemer and A. Steininger, "Advanced Delay-Insensitive 4-Phase Protocols," in *2018 Austrochip Workshop on Microelectronics (Austrochip)*, 2018, pp. 50-55: IEEE.

[19] G. P. Acharya and M. A. Rani, "Berger code based concurrent online self-testing of embedded processors," *Journal of Semiconductors*, vol. 39, no. 11, p. 115001, 2018.

[20] W. W. Peterson, "On checking an adder," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 166-168, 1958.

[21] S. J. Piestrak, "Self-testing checkers for arithmetic codes with any check base A," in *IEEE International*

[1] J. F. Ziegler et al., "IBM experiments in soft fails in computer electronics (1978-1994)," *IBM journal of research and development*, vol. 40, no. 1, pp. 3-18, 1996.

[2] F. Wang and V. D. Agrawal, "Single event upset: An embedded tutorial," *IEEE 21st International Conference on VLSI Design, VLSID*, pp. 429-434, 2008.

[3] W. Heidergott, "SEU tolerant device, circuit and processor design," in *Design Automation Conference, 2005. Proceedings. 42nd*, 2005, pp. 5-10: IEEE.

[4] B. Narasimham et al., "Characterization of digital single event transient pulse-widths in 130-nm and 90-nm CMOS technologies," *IEEE Transactions on Nuclear Science*, vol. 54, no. 6, pp. 2506-2511, 2007.

[5] M. Santarini, "Cosmic radiation comes to ASIC and SOC design-As 1C-process geometries shrink, single-event effects, such as soft errors and latch-ups, will soon become primary concerns for designers of ASICs and," *Edn*, vol. 50, no. 10, pp. 46-60, 2005.

[6] N. Mahatme et al., "Impact of technology scaling on the combinational logic soft error rate," in *Reliability*

- [25] A. Saed, M. Ahmadi, and G. A. Jullien, "Arithmetic circuits for analog digits," in *IEEE 29th International Symposium on Multiple-Valued Logic*, , 1999, pp. 186-191.
- [26] T. Temel and A. Morgul, "Implementation of multi-valued logic gates using full current-mode CMOS circuits," *Analog Integrated Circuits and Signal Processing*, vol. 39, no. 2, pp. 191-204, 2004.
- [27] F. Yuan, "Voltage-Mode Versus Current-Mode: A Critical Comparison," *CMOS Current-Mode Circuits for Data Communications*, pp. 1-12, 2007.
- [28] J. Engblom, "Why SpecInt95 should not be used to benchmark embedded systems tools," *ACM SIGPLAN Notices*, vol. 34, no. 7, pp. 96-103, 1999
- [22] I. Sayers and D. Kinniment, "Low-cost residue codes and their application to self-checking VLSI systems," *IEEE Proceedings (Computers and Digital Techniques)*, vol. 132, no. 4, pp. 197-202, 1985.
- [23] S.-H. Shieh and W.-S. Tong, "Berger Code Totally Self-Checking Checker Design for Embedded Adder Cores," *Proceedings of the 5th Symposium on Smart Life Science and Technology (Part 1)*, 2010.
- [24] N. Homma, T. Aoki, and T. Higuchi, "Algorithm-level interpretation of fast adder structures in binary and multiple-valued logic," in *IEEE 36th International Symposium on Multiple-Valued Logic, 2006. (ISMVL 2006)*, , 2006, pp. 2-2.