

بررسی و مقایسه‌ی اثر کدینگ‌های تصحیح و تشخیص خطا و کدینگ‌های اجتناب از همشنوایی در افزایش قابلیت اطمینان شبکه‌های بر تراشه

سپیده صفری^۱، محسن انصاری^۲، زهرا شیرمحمدی^۳، سید قاسم میرعمادی^۴

^۱ دانشجوی کارشناسی ارشد، دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف
تهران، ایران ssafari@ce.sharif.edu

^۲ دانشجوی کارشناسی ارشد، دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف
تهران، ایران mansari@ce.sharif.edu

^۳ دانشجوی دکترای تخصصی، دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف
تهران، ایران shirmohammadi@ce.sharif.edu

^۴ استاد، دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف
تهران، ایران miremadi@sharif.edu

چکیده

در شبکه‌های روی تراشه انتقال داده از طریق سیم‌های مابین هسته‌های پردازشی صورت می‌گیرد. خازن‌های جفت‌شدگی بین این سیم‌های مجاور و موازی، شبکه‌های روی تراشه را مستعد بروز اشکال همشنوایی کرده و قابلیت اطمینان این شبکه‌ها را با چالش جدی مواجه ساخته‌است. با کاهش اندازه‌ی فناوری در سال‌های اخیر، احتمال بروز همشنوایی در شبکه‌های روی تراشه در حال افزایش است. اگرچه روش‌های مختلفی در سطوح مختلف تجرید جهت مقابله با اشکال همشنوایی ارائه شده‌است، ولی کدینگ تصحیح و تشخیص خطا و کدینگ‌های اجتناب از همشنوایی با داشتن هزینه‌ی کمتر از روش‌های مؤثر مقابله با اشکال همشنوایی به شمار می‌روند؛ همین مسأله شبکه‌های روی تراشه را نیازمند استفاده از کدینگ‌های قدرتمندتری می‌کند که قابلیت‌های متفاوتی در افزایش قابلیت اطمینان شبکه‌های روی تراشه دارند. در این مقاله مهم‌ترین کدینگ‌های تصحیح و تشخیص خطا و اجتناب از همشنوایی از دیدگاه قابلیت افزایش قابلیت اطمینان و سربارها را مورد بررسی قرار می‌دهیم.

واژه‌های کلیدی

شبکه‌های روی تراشه، اشکال همشنوایی، کدینگ‌های اجتناب از همشنوایی – کدینگ‌های تشخیص و تصحیح خطا

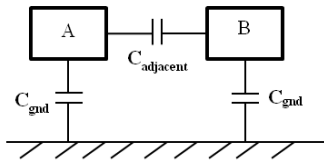
۱- مقدمه

کوچک‌تری به نام فلیت^۲ تقسیم شوند. فلیت واحدی از داده است که با توجه به پهنای بیتی کانال ارتباطی، طی یک یا چند سیکل ساعت از روی کانال عبور داده می‌شود [1]. از جمله مهم‌ترین منابع اشکالی که قابلیت اطمینان یک فلیت و سیستم بر تراشه مبتنی بر شبکه را با چالش جدی مواجه می‌سازد، اشکال همشنوایی^۳ است [2]، زیرا با کاهش اندازه‌ی فناوری، ضخامت سیم نسبت به طول و عرض آن سریع‌تر کاهش پیدا می‌کند و این عامل اثر خازن جفت‌شدگی بین سیم‌ها را افزایش می‌دهد و به این ترتیب نسبت خازن جفت‌شدگی به خازن کل افزایش خواهد یافت.

پیشرفت روزافزون فناوری ساخت قطعات نیمه‌هادی، طراحان را قادر ساخته‌است که بتوانند میلیاردها ترانزیستور را در یک تراشه به عنوان یک تراشه‌ی چند هسته‌ای مجتمع کنند. هدف از چنین تراشه‌هایی به دست آوردن قدرت پردازشی بالا می‌باشد. شبکه‌های روی تراشه^۱ به عنوان راه-حلی مقیاس‌پذیر و کارا جهت ارتباط این هسته‌های پردازشی ارائه شدند. در انتقال داده مابین این هسته‌های پردازشی هر پیام به بسته‌هایی با اندازه‌ی مساوی تقسیم می‌شود و بسته‌ها نیز ممکن است به واحدهای

² Flit
³ Crosstalk

¹ Network on Chip



شکل ۱: خازن تشکیل شده بین یک سیم با سیم مجاور آن و سیم با

زمین [4]

اثر همشنوایی در میان سیم‌های همسایه، در ارتباطات قسمت‌های مختلف شبکه روی تراشه، یک مشکل جدی به حساب می‌آید. این اشکال خود را به صورت بروز تغییرات ناخواسته در ولتاژ یک سیم، تسریع ناخواسته در لبه‌ی مثبت یا منفی^۴ سیگنال در یک سیم و یا به تعویق افتادن لبه‌ی مثبت یا منفی^۵ سیگنال در یک سیم نشان می‌دهد [5]. از آن-جا که همشنوایی وابسته به طول مسیرهای ارتباطی، الگوی داده‌های ارسالی خط، مباحث فیزیکی و غیره می‌باشد، انواع کدینگ‌ها جهت مقابله با اشکالات همشنوایی و تشخیص و تصحیح خطاهای آن مطرح شده‌است. با کاهش اندازه‌ی گیت‌ها، تأخیر گیت‌ها کاهش یافته ولی تأخیر گذرگاه عمومی در حال افزایش است. تأخیر سیم نام در یک گذرگاه m بیتی طبق فرمول ۱ محاسبه می‌گردد [6]:

$$T_i = \begin{cases} \tau_0[(1 + \lambda)\Delta_1^2 - \lambda\Delta_1\Delta_2], & i = 1 \\ \tau_0[(1 + 2\lambda)\Delta_i^2 - \lambda\Delta_i(\Delta_{i-1} + \Delta_{i+1})], & i \neq 1, m \\ \tau_0[(1 + \lambda)\Delta_m^2 - \lambda\Delta_m\Delta_{m-1}], & i = m \end{cases} \quad (1)$$

در این رابطه λ نسبت خازن جفت‌شده‌ی بین دو سیم به خازن بار (بدنه) و τ_0 تأخیر گذر در سیم عاری از همشنوایی می‌باشند. همچنین هرگاه تغییر از "۰" به "۱" و از "۱" به "۰" وجود داشته باشد Δ_i به ترتیب برابر با "۱" و "۰" و اگر بر روی سیم نام تغییری وجود نداشته باشد Δ_i برابر با صفر خواهد بود. در جدول ۱ پنج کلاس مختلف همشنوایی به همراه الگوها و تأخیرهای مرتبط با هر یک ذکر شده‌اند. در این جدول از نمادهای \uparrow و \downarrow به ترتیب برای نمایش گذارهای "۰" به "۱" و "۱" به "۰" استفاده شده-است [6].

اگر تأخیر در بدترین حالت را برابر با $\tau_0[(1 + 4\lambda)]$ در نظر بگیریم، راه‌حل‌های بسیاری برای کاهش این تأخیر پیشنهاد شده‌اند که در بین آن-ها کدهای مقابله با همشنوایی راه‌حلی نویدبخش و مناسب هستند. این کدها نه تنها وابسته به فناوری نیستند بلکه تأخیر ناشی از همشنوایی را کاهش می‌دهند و در عین حال مساحت کمتر و توان مصرفی کمتری نسبت به روش استفاده از محافظ دارند.

جدول ۱: کلاس‌های همشنوایی [6]

Transition Class	Patterns	Delay
0C	$\uparrow\uparrow\uparrow \downarrow\downarrow\downarrow$	π_0
1C	$-\uparrow\uparrow \downarrow\downarrow \uparrow\uparrow- \downarrow\downarrow-$	$(1 + \lambda)\pi_0$
2C	$-\uparrow- \downarrow-$	$(1 + 2\lambda)\pi_0$
3C	$-\uparrow\downarrow \uparrow\downarrow- \uparrow\downarrow- \uparrow\downarrow$	$(1 + 3\lambda)\pi_0$
4C	$\uparrow\downarrow\uparrow \downarrow\uparrow\downarrow$	$(1 + 4\lambda)\pi_0$

مقابله با اشکال همشنوایی در سطوح مختلف تجزید صورت می‌گیرد. این روش‌ها شامل روش‌های سطح فیزیکی^۱، ترانزیستور^۲ و انتقال ثبات^۳ هستند. عمده‌ی روش‌های دو سطح اول دارای سربار بالایی از نظر کارایی و مساحت اشغالی می‌باشند. در سطح انتقال ثبات روش کدینگ‌های اجتناب از همشنوایی و کدینگ‌های تشخیص و تصحیح خطا از جمله مؤثرین راه-کارهای مقابله با اشکال همشنوایی هستند [3]. در روش کدینگ اجتناب از همشنوایی، با جلوگیری از ظهور رشته بیت‌های خاصی، احتمال رخداد گذارهای مخاطره‌آمیز در کانال‌های شبکه‌های روی تراشه به صفر می‌رسد و این در حالی است که سربار چندانی را به سیستم اعمال نمی‌کنند. کدینگ‌های تشخیص و تصحیح خطا هنگام دریافت داده در طرف گیرنده اگر خطایی وجود داشته باشد، آن را کشف کرده و تصحیح می‌کنند و اینکار منجر به اضافه شدن سربار مساحت به سیستم می‌شود. بنابراین با توجه به اهمیت سربار مساحت و توان مصرفی در هر یک از روش‌های پیشنهادی، در این مقاله مقایسه‌ای بین این روش‌ها انجام گرفته است.

در بخش دوم از این مقاله به بیان انگیزه از مطالعه‌ی اشکال همشنوایی و مشکلات ناشی از آن اشاره شده‌است. در بخش سوم با نگاهی به کارهای پیشینی که در زمینه‌ی کدینگ‌های اجتناب از همشنوایی و کدینگ‌های تشخیص و تصحیح خطا انجام شده‌اند، به مقایسه‌ی هر یک از روش‌ها اشاره شده‌است. حاصل شبیه‌سازی‌های انجام شده نیز در قسمت چهارم مورد ارزیابی قرار گرفته‌است. در نهایت نیز در فصل پنجم به نتیجه-گیری از بررسی تمامی کدینگ‌ها خواهیم پرداخت.

۲- انگیزش و اهمیت اشکال همشنوایی

اشکال همشنوایی به دلیل وجود خازن‌های همشنوایی در خطوط طولانی و موازی گذرگاه‌های ارتباطی در شبکه‌های روی تراشه رخ می‌دهد و باعث رخداد تغییرات ناخواسته در زمان بندی گذار سیگنال‌ها و همچنین ایجاد گذارهای ناخواسته در گذرگاه روی تراشه می‌شود. برای روشن شدن هرچه بهتر موضوع به شکل ۱ توجه نمایید. در شکل ۱ سیم‌ها با سیم‌های مجاور و با زمین دارای خازن هستند. هنگامی که سیم A تغییر مقدار می-دهد، به دلیل وجود خازن تزویجی بین دو سیم، این تغییر می‌تواند روی سیم B تأثیر بگذارد که این اثر را همشنوایی گویند. در صورتی که سیم B نیز همزمان با A تغییر کند، ممکن است تأخیر سوئیچینگ، کاهش یا افزایش یابد. اگر B تغییر نکند، ممکن است همشنوایی روی سیم B نویز ایجاد کند.

اثر همشنوایی به نسبت خازن سیم همسایه به کل خازن بستگی دارد. باید در نظر داشت که خازن بار در خازن کل محاسبه می‌شود، بنابراین برای سیم‌های کوتاه و بارهای بزرگ، خازن بار نقش عمده‌ای داشته و همشنوایی قابل صرف نظر کردن است، اما برای سیم‌های بلند همشنوایی اثری بسیار مهم به حساب می‌آید [4].

⁴ Rising/Falling Speed-Up

⁵ Rising/Falling Delay

¹ Layout level

² Transistor level

³ RTL level

۳- کارهای پیشین

۳-۱-۱- کدهای گذار ممنوعه

یک کلمه‌ی کد، عاری از الگوی ممنوعه نامیده می‌شود هرگاه هیچ الگوی بیتی ممنوعه‌ای در هیچ‌یک از سه بیت متوالی آن مشاهده نشود. کدهای الگو ممنوعه، الگوهای نامعتبر ۱۰۱ و ۰۱۰ را حذف کرده تا الگوهای گذار ممنوعه‌ی $\uparrow\downarrow$ ، $\downarrow\uparrow$ ، $\uparrow\downarrow$ ، $\downarrow\uparrow$ ، $\uparrow\downarrow$ ، $\downarrow\uparrow$ ، $\uparrow\downarrow$ ، $\downarrow\uparrow$ در فلیت‌های اطلاعات کانال‌ها از بین بروند و به این ترتیب سرعت کانال‌ها و صحت داده‌های ارسالی افزایش یابد. نماد \uparrow و \downarrow به ترتیب نشان‌دهنده‌ی تغییر از "۰" به "۱" و "۱" به "۰" هستند، از علامت - هم برای نشان دادن عدم وجود گذار در سیستم استفاده می‌شود. کدگذار-کدگشایی که بر این اساس کار می‌کنند با سربار مساحت، تأخیر و توان مصرفی همراه هستند.

۳-۱-۲- سیستم عددی دودویی بر پایه‌ی فیبوناچی

روش کدگذاری الگوی ممنوعه بر مبنای سیستم عددی فیبوناچی، از دنباله اعداد فیبوناچی به عنوان پایه‌های خود استفاده می‌کند؛ ایده‌ی اصلی این است که هر الگوی بیتی ۰۱۱ با ۱۰۰ قابل جابه‌جایی است. بر این اساس برخی الگوهای نامعتبر حذف می‌شوند. نمایش فیبوناچی یک عدد و تعریف تابع فیبوناچی به ترتیب در فرمول ۳ و ۴ نشان داده شده‌اند.

$$F = \sum_{k=1}^m d_k f_k, d_k = 0 \text{ or } 1 \quad (3)$$

$$f_m = \begin{cases} 0 & \text{if } m = 0 \\ 1 & \text{if } m = 1 \\ f_{m-1} + f_{m-2} & \text{if } m \geq 2 \end{cases} \quad (4)$$

کدهای مبتنی بر فیبوناچی دو ضعف عمده دارند. اولین ضعف این است که کدگذار-کدگشایی که برای این روش استفاده می‌شوند دارای پیچیدگی زیادی هستند و این امر منجر شده است تا ایده‌ی ساده‌سازی و کاهش پیچیدگی آن‌ها بسیار مهم تلقی شود. دومین ضعف نیز این است که سیستم اعداد مبتنی بر فیبوناچی دارای ابهام است [5].

۳-۱-۳- کدهای غیرهمپوشان

در این روش دو سیم مجاور که در حال عبور از یک کلمه‌ی کد به کلمه‌ی دیگری هستند، مجاز به تغییر در خلاف جهت یکدیگر نیستند. یعنی از گذارهایی به شکل $10 \leftarrow 01$ و $01 \leftarrow 10$ جلوگیری می‌شود. فرض کنید که بین دو سیم همسایه یک کرانه وجود داشته باشد. اگر در تمامی کلمه‌های کد بین این دو کد فقط کلمه‌های کد ۱۰، ۰۰ و ۱۱ وجود داشته باشد، از آن به عنوان یک کرانه‌ی ۰۱ نام برده می‌شود. در غیر این صورت اگر بین دو کد فقط کلمه‌های کد ۰۱، ۰۰ و ۱۱ وجود داشته باشد به آن یک کرانه‌ی نوع ۱۰ می‌گویند. تمامی کرانه‌های موجود در یک کلمه کد غیرهمپوشان از نوع ۱۰ یا ۰۱ هستند. در کتابچه‌های کد غیرهمپوشان با اندازه‌ی بیشینه همواره دو شرط صادق است: (۱) کتابچه-های کد غیرهمپوشان دارای کرانه‌های یک در میان ۱۰ و ۰۱ هستند. (۲) الگوهای بیتی به شکل "۰۱۰"، "۱۰۱"، "۱۰۰۱" و "۰۱۱۰" در آن-ها ظاهر نمی‌شود [5].

در ادبیات موضوعی روش‌های مختلفی جهت مقابله با اشکال همشنوایی ارائه شده‌اند. این روش‌ها در سطوح مختلف تجرید صورت می‌گیرند، که شامل روش‌های سطح فیزیکی، ترانزیستور و انتقال ثبات هستند. با توجه به سطوح مختلف انتزاع از روش‌های اجتناب از همشنوایی متفاوتی استفاده می‌شود. در پایین‌ترین سطح انتزاع یعنی سطح فیزیکی با استفاده از روش محافظ‌گذاری^۱ برای سیم‌ها و به کارگیری تکنیک‌های خاص مسیریابی، برای کاهش نرخ وقوع اشکال همشنوایی تلاش می‌شود. روش محافظ‌گذاری با وجود کاهش تأخیر گذرگاه، افزایش شدید مساحت را به همراه دارد. همچنین این روش نیازمند اعمال تغییرات در فرآیند ساخت است که زمان طراحی و هزینه را افزایش می‌دهد. بنابراین سربار بیش از حد این روش باعث شده است تا طراحان و محققان به سمت استفاده از آن نروند [5]. در سطح ترانزیستور نیز با استفاده از روش‌های تغییر زمان‌بندی تغییرات سیگنال‌های همسایه، می‌توان تأخیر ناشی از اثر همشنوایی را کاهش داد؛ ولی این روش فقط در کانال‌های ارتباطی قابل تکرار، انجام‌پذیر است. در سطح انتقال ثبات نیز از کدینگ داده برای کاهش اشکالات همشنوایی در کانال‌های شبکه روی تراشه استفاده می‌شود. کدینگ‌های اجتناب از همشنوایی نمونه‌ای از این کدها می‌باشند. برای مثال کدینگ-های مبتنی بر دنباله‌ی اعداد فیبوناچی می‌توانند از وقوع دنباله‌های عددی نامناسب و غیر مجاز جلوگیری نمایند، ولی این روش به خصوص وقتی پهنای کانال ارتباطی افزایش می‌یابد، دارای پیچیدگی الگوریتمی زیادی است [3]. کدهای همشنوایی را می‌توان بر اساس میزان کاهش تأخیر دسته‌بندی نمود که در ادامه هر یک از آن‌ها شرح داده شده‌اند.

۳-۱-۳- کدینگ‌های اجتناب از همشنوایی

کدینگ‌های اجتناب از همشنوایی در خانواده‌های مختلفی قرار می‌گیرند که عبارتند از: (۱) کدگذاری‌های غیر همپوشان^۲ (۲) کدگذاری‌های الگوی ممنوعه^۳ و کدگذاری‌های گذار ممنوعه^۴ (۳) کدگذاری‌های یک-لاندا^۵ [7]. کدگذاری‌های اجتناب از همشنوایی به دو دسته‌ی با حافظه و بدون حافظه تقسیم می‌گردند. در کدگذاری‌های با حافظه، کلمه‌ی کد^۶ بر اساس کلمه‌ی کد قبلی و کلمه‌ی داده‌ای^۷ که باید در سیکل جاری ارسال شود، تولید می‌گردد. اما روش‌های بدون حافظه، کتابچه‌ی کد^۸ ثابتی دارند و برای تولید کلمه‌های کد فقط به داده‌هایی که هم اکنون باید ارسال شوند توجه می‌کنند. کدگذار-کدگشایی با حافظه نسبت به کدگذار-کدگشایی بدون حافظه پیچیده‌ترند و سربار اعمالی آن‌ها به سیستم بالاتر است [5].

¹ Shielding

² Forbidden Overlap Codes (FOCs)

³ Forbidden Pattern Code (FPCs)

⁴ Forbidden Transition Codes (FTCs)

⁵ One Lambda Code (OLCs)

⁶ Code Word

⁷ Dataword

⁸ Codebook

۳-۱-۴- کدگذاری‌های یک-لاندا

لاندا با پیچیدگی مرتبه‌ی دوم و دارای معماری ساده و منظم و امکان خط-لوله شدن، استفاده شده‌است.

در [12] خانواده‌ای از کدهای مبتنی بر کدهای اشتراک ممنوعه برای کاهش سربارهای ناشی از کدگذار-کدگشا ارائه شده‌است. برتری دو کد پیشنهاد شده نسبت به کدهای قبلی، در این است که میزان مساحت و انرژی مصرفی آن‌ها کمتر می‌باشد و همچنین با افزایش سرعت سیستم، کارایی مناسبی را ایجاد می‌کند. کدهای مطرح شده نسبت به حالتی که از محافظ استفاده می‌کنند، سربارهای کمتری را در توان، مساحت و تأخیر به سیستم اعمال می‌کنند. در [13] با طراحی یک تکنیک هم‌گذار و عکس عمل هم‌گذار، اثرات نامطلوب هم‌شنوایی روی گذرگاه انتقال را کاهش داده و با از بین بردن برخی از الگوهای انتقال داده، اثر هم‌شنوایی را به حداقل رسانده‌است. در واقع یک طرح کدینگ گذرگاه برای معماری گذرگاه‌های عریض در پردازنده‌های با کارایی بالا ارائه داده‌است. در [14] یک تکنیک کدینگ جدید برای به حداقل رساندن فعالیت خازن‌های گذار و جفت‌شده در خطوط گذرگاه با استفاده از طراحی کدگذار-کدگشای تمام کدهای اجتناب از هم‌شنوایی مبتنی بر سیستم اعداد دودویی و افزودنی سه‌بعدی پیشنهاد شده‌است. طراحی‌های کدگذار-کدگشاهای ارائه شده با پیچیدگی مساحت و تأخیری که با اندازه‌ی گذرگاه با نسبت درجه دو افزایش می‌یابد، همراه است در حالی که نرخ کد آن بهینه و یا نزدیک به بهینه است. برای مثال در این روش کلمه‌ی کد ۰۱۰۱ در گذرگاه چهاربیتی تبدیل به ۱۰۰۰ می‌شود. در منبع [15] یک الگوریتم مبتنی بر نمودار تصمیم‌گیری دودویی ترتیبی کاهش‌یافته^۴ برای محاسبه‌ی سربار مساحت کدگذار-کدگشا ارائه شده‌است. الگوریتم مذکور با استفاده از ساختار یک کدگذار-کدگشای عاری از گذار، الگوی KC که مقدار الگو می‌باشد را برای پهنای باند واقعی n تولید می‌کند. همچنین با استفاده از این کدگذار-کدگشاهای حذف‌کننده‌ی هم‌شنوایی باحافظه، می‌توان به یک گذرگاه با بیش از شش برابر سرعت دست یافت و این در حالی است که سربار اندازه‌ی گذرگاه و مساحت کل، کمتر از سربارهای مشابه در کدگذار-کدگشای بدون حافظه است.

در منبع [16] یک کدگذار-کدگشای مبتنی بر کد اجتناب از هم‌شنوایی گذار ممنوعه ارائه شده‌است. کدینگ مطرح شده مبتنی بر سیستم اعداد فیبوناچی است که می‌تواند تمام اعداد را بوسیله‌ی بردارهای FTF در سیستم اعداد فیبوناچی نمایش دهد. طراحی کدگذار-کدگشای پیشنهاد شده بسیار کارآمد است و به راحتی می‌تواند با یک تکنیک افزاز گذرگاه ترکیب شود، همچنین پیچیدگی منطقی این روش بسیار کمتر از روش‌های پیشین است و حدود ۱۷٪ کاهش پیچیدگی منطقی را در بر دارد. طراحی الگوریتم کدینگ که هر مقدار ورودی را به یک کلمه‌ی کد FTF در سیستم اعداد فیبوناچی نگاشت می‌کند، بدین صورت است که می‌تواند ظهور الگوهای ممنوعه‌ی "۰۱" و یا "۱۰" را بر روی دو بیت پر ارزش حذف کند.

در کدهای یک-لاندا از تغییر کلمه‌ی کد "۰۱" به "۱۰" و همچنین "۰۱۰" به "۱۰۱" و مواردی مشابه آن که سیم‌های همسایه در طی تغییر کلمه‌ی کد در جهت مخالف یکدیگر حرکت می‌کنند، اجتناب می‌شود.

۳-۱-۵- مروری بر سایر کدینگ‌های اجتناب از هم‌شنوایی

در [8] برای اولین بار یک الگوی کلی برای طراحی کدگذار-کدگشای انواع کلاس‌های کدهای اجتناب از هم‌شنوایی مبتنی بر سیستم اعداد دودویی مبتنی بر فیبوناچی ارائه شده‌است. در طراحی‌های کدگذار-کدگشاهای قبل از [8] با افزایش اندازه‌ی گذرگاه، پیچیدگی مساحت و تأخیر با درجه‌ی دو افزایش می‌یافت، ولی در [8] سعی شده‌است تا با استفاده از روش‌های مبتنی بر سیستم اعداد دودویی فیبوناچی، کدگذار-کدگشاهایی طراحی شوند که از نظر مساحت دارای نسبت خطی با پهنای باند گذرگاه باشند. کدگذار-کدگشاهای موردنظر از نظر مداری ساده و منظم هستند و از معماری خطلوله برای بالا بردن توان عملیاتی و از تکنیک‌های کدینگ‌های جزئی برای مقابله با تأخیر هم‌شنوایی استفاده می‌کنند. تکنیک [9] بسیار شبیه به تکنیک کدینگ فیبوناچی نرمال^۱ است. این روش اصلاً نیازی به حذف کردن گذارهای هم‌شنوایی ندارد. در تکنیک کدینگ فیبوناچی نرمال اعداد فیبوناچی به عنوان عناصر پایه برای تولید m بیت کلمه‌ی کد در نظر گرفته می‌شوند. در تکنیک [9] که به نام فیبوناچی افزونه نام‌گذاری شده‌است، اعداد فیبوناچی به عنوان عناصر پایه در نظر گرفته می‌شوند. تکنیک دیگر [9] فیبوناچی افزونه‌ی مکمل^۲ است که در آن کلمه‌های کد، مکمل هر کلمه‌ی کد در روش فیبوناچی افزونه هستند. ایده-ی اصلی [10] ارائه‌ی سیستم عددی مبتنی بر فیبوناچی سه‌مقدار با اعداد فیبوناچی متفاوت با وزن‌های وابسته به موقعیت در ارقام مختلف است. در کلمه‌ی کد معتبر توزیع نامتقارن سه مقدار ۰، ۱ و ۲ در نمایش اعداد با یک جفت مکمل متوالی، هرگز دیده نمی‌شود. در این سیستم وقتی طول داده بزرگ است احتمال وجود اعداد در هر کلمه‌ی کد به صورت مجانبی در می‌آید و برای اعداد ۰، ۱ و ۲ به ترتیب برابر با ۴۲.۲٪، ۳۲.۲٪ و ۱۹.۶٪ است. این در حالی است که احتمال رخداد هر کدام از اعداد مذکور در سیستم‌های سه‌تایی متداول در هر کلمه‌ی کد برابر با ۳۳.۳۳٪ است. خانواده‌ی جدید کدهای اجتناب از هم‌شنوایی مطرح شده در [7] شامل برخی از کدهای قدیمی از جمله کدهای لاندا است. برای مثال دو نوع از کدهای جدید تولیدی بر مبنای کلاس‌های هم‌شنوایی (2C، 1C) و (3C، 4C) هستند. در [11] ابتدا یک سیستم عددی مبتنی بر فیبوناچی و برایش-یافته^۳ ارائه شده و سپس الگوی کدگذاری مبتنی بر کدهای الگو ممنوعه بر روی آن به کار برده شده‌است. الگوی کدینگ پیشنهاد شده دارای نرخ کد یکسان با کدگذار-کدگشای مبتنی بر کدهای الگو ممنوعه‌ی بهینه است. همچنین یک سیستم عددی برای کدهای یک-لاندا ارائه شده‌است و از آن سیستم عددی برای ارائه‌ی اولین کدگذار-کدگشای مبتنی بر کدهای یک-

¹ Normal-Form Fibonacci

² Complement Redundant Fibonacci

³ Modified Fibonacci Numerl System (MFNS)

⁴ ROBDD (Reduced Ordered Binary Decision Diagram)

۳-۲- کدینگ‌های تشخیص و تصحیح خطا

شیفت کرانه‌ای همانند دوگان‌سازی و توازن است. به این معنی که در ابتدا یک نسخه از روی داده‌ی ورودی تهیه می‌شود و این نسخه به همراه بیت-های توازن به داده‌ی اصلی اضافه می‌گردد. بنابراین برای اینکه بتوان از تغییر بیت‌های همسایه در خلاف جهت یکدیگر جلوگیری کرد، داده‌ی کدگذاری شده قبل از ارسال، به تعداد چرخه‌های فرد به سمت راست انتقال چرخشی می‌یابند [19]. کدگذاری‌های شیفت کرانه‌ای دارای قابلیت اجتناب از همسایه‌ی هستند و با داشتن حداقل فاصله‌ی همینگ برابر با سه، می‌توانند خطاهای تک‌بیتی را تصحیح نمایند و همچنین در این روش خازن جفت‌شدگی سیم گذرگاهی که برای ارسال کلمه‌ی کُد با روش کدگذاری شیفت کرانه‌ای استفاده می‌شود، در بدترین حالت برابر با $2C$ خواهد بود که این مقدار مشابه با خازن جفت‌شدگی سیم گذرگاه استاندارد در بدترین حالت است؛ ولی این روش نیازمند سیم‌های مضاعف است و تأخیر بیش‌تری را در سیستم ایجاد می‌کند [18]. کدگذاری دوگان-سازی و توازن همراه با معکوس نمودن گذرگاه^۷ تلفیقی از کدهای دوگان-سازی و توازن و برخی مشخصه‌های کدهای کم‌توان است. در این کدینگ اگر کلمه‌ی کدی که هم‌اکنون بر روی گذرگاه قرار است ارسال شود با کلمه‌ی کدی که قبل از آن ارسال شده‌است، در بیش از نیمی از بیت‌ها تفاوت داشته باشد، کلمه‌ی کُد جاری معکوس می‌گردد. ویژگی‌های کم‌توان بودن این نوع کدگذاری منجر می‌شود تا تعداد تغییرات داده، در گذرگاه داده کم شود و به این ترتیب در میزان توان مصرفی صرفه‌جویی گردد [18]. کدهای تصحیح خطای چندگانه^۸ ترکیبی از کدهای همینگ و کدگذاری‌های دوگان‌سازی و توازن می‌باشند، که می‌توانند همزمان از اثر همسایه‌ی جلوگیری کنند و همچنین دو خطا را نیز تصحیح کنند^۹. لازم به ذکر است که این کدگذاری علی‌رغم داشتن فاصله‌ی همینگ هفت، به منظور ساده‌سازی و کم کردن سربار توان و مساحت، به جای تصحیح سه خطا، تنها قادر به تصحیح دو خطا است [18]. کدگذاری‌های تلفیقی با استفاده از کدهای مقابله با همسایه‌ی و تصحیح خطای سه‌گانه^{۱۰} بر مبنای کدهای تصحیح خطای چندگانه می‌باشند. ماهیت کدگذار این طرح از این حقیقت پیروی می‌کند که کمترین فاصله‌ی همینگ بین دو کلمه‌ی کد که دارای کد همینگ با قابلیت تصحیح یک خطا می‌باشد، سه است. همچنین با استفاده از دوگان‌سازی می‌توان از ایجاد بدترین حالت همسایه‌ی بین سیم‌های همسایه اجتناب نمود و فاصله‌ی همینگ را به شش رساند. در نهایت با اضافه نمودن بیت توازن کلی فاصله‌ی همینگ برابر با هفت بیت خواهد شد. بنابراین کد تولید شده با فاصله‌ی همینگ هفت، قابلیت تصحیح سه خطا را خواهد داشت. انجام عملیات دوگان‌سازی علاوه بر آنکه منجر به زیاد شدن فاصله‌ی همینگ می‌شود، مانع از تغییر بیت‌های همسایه در جهت مخالف یکدیگر می‌گردد و به این ترتیب الگوهای ممنوعه رخ نمی‌دهند [20]. با تولید چنین کُدی تأخیر یک کانال در بدترین شرایط از $C_L(1+4\lambda)$ به $C_L(1+2\lambda)$ خواهد رسید. ولی باید در نظر داشت

برای بالا بردن کارایی و قابلیت اطمینان شبکه‌های روی تراشه، علاوه بر استفاده از روش‌هایی مانند حفاظ‌گذاری یا کدینگ‌های اجتناب از همسایه‌ی، از کدینگ‌های تصحیح خطا^۱ استفاده می‌شود. بنابراین می‌توان در اتصالات و بلوک‌های منطقی موجود در شبکه‌های روی تراشه که در معرض انواع نویزها قرار دارند، از این کدها استفاده نمود. در اغلب کدینگ-های تشخیص و تصحیح خطا از مفهوم فاصله‌ی همینگ استفاده می‌شود. این مفهوم بدین صورت است که اگر فاصله‌ی همینگ میان کدها برابر با d باشد، می‌توان $d-1$ خطا را تشخیص داد و $\left\lfloor \frac{d-1}{2} \right\rfloor$ خطا را تصحیح نمود.

کدهای همینگ نمونه‌ای از کدهای بلوکی خطی می‌باشند که کمترین فاصله‌ی همینگ آن‌ها برابر با سه است. برای یک عدد صحیح مثبت $(r \geq 3)$ کد همینگ (n, k) طبق روابط زیر وجود دارد:

$$n = 2^r - 1$$

$$k = 2^r - 1 - r$$

کد همینگ نرمال، دارای انعطاف‌پذیری بالایی است و هر پهنای داده-ی ورودی را پشتیبانی می‌کند [17]. با اضافه نمودن یک بیت توازن سراسری می‌توان کد همینگ را گسترش داد. کد همینگ گسترش‌یافته‌ی $(n+1, k)$ با رعایت نیازمندی‌های زیر به دست می‌آید:

$$n = 2^{r-1} - 1$$

$$k = 2^{r-1} - r$$

چنین کد همینگ گسترش‌یافته‌ای دارای فاصله‌ی همینگ چهار است و می‌تواند همزمان یک بیت خطا را تصحیح و دو بیت خطا را کشف کند و تحت عنوان کدهای SEC-DED^۲ شناخته می‌شود [17]. کدینگ دوگان-سازی و توازن^۳ از روش تکرار استفاده می‌کند، به این معنی که می‌توان یک نسخه‌ی دیگر از داده‌ها تهیه نمود و در کنار آن قرار داد. این کدگذاری با داشتن فاصله‌ی همینگ سه می‌تواند خطاهای تک‌بیتی را تصحیح و خطاهای دو بیتی را کشف کند. همچنین این روش می‌تواند اثر خازن‌های جفت‌شدگی را نیز کاهش دهد و تأخیر بدترین حالت را به مقدار $\tau_0[(1+2\lambda)]$ برساند [17] [18]. ایده‌ی اصلی کدینگ ریل دوگانه‌ی بهبودیافته^۴ همانند دوگان‌سازی و توازن است. در این روش برای یک کلمه‌ی داده‌ی K بیتی، تعداد $m=K+1$ بیت به عنوان بیت‌های واریسی^۵ به کلمه‌ی داده‌ی اصلی اضافه می‌شود. بنابراین با در نظر گرفتن بیت‌های واریسی، عرض کلمه‌های کد برابر با $n=2K+1$ خواهد شد. در این روش دو نسخه از بیت توازن در کنار بیت‌های کلمه‌ی کد شده قرار می‌گیرند و به این ترتیب باعث می‌شود تا اشکال همسایه‌ی، بیش‌تر کاهش بیابد. این روش می‌تواند دو خطا را کشف و یک خطا را تصحیح کند. [5]. کدگذاری شیفت کرانه‌ای^۶ نوعی از کدها می‌باشد که در آن بیت‌های همسایه نمی‌توانند همزمان در جهت مخالف یکدیگر تغییر کنند. نحوه‌ی کدگذاری

¹ Error Correcting Codes

² Single-Error-Correction, Double-Error-Detection

³ Duplicated-Add-Parity (DAP)

⁴ Modified Dual Rail (MDR)

⁵ Check Bits

⁶ Boundary Shift Code (BSC)

⁷ Duplicate Add Parity Bus Invert (DAPBI)

⁸ Multiple Error Correction Codes (MECC)

⁹ Crosstalk Avoidance Double Error Correcting (CADEC) Code

¹⁰ Joint Crosstalk Avoidance and Triple Error Correction (JTEC)

اطمینان بالایی در طول شبکه انتقال داده می‌شوند. در این طرح ثبات‌ها و حالات در مسیر یاب‌ها توسط افزونگی سه پیمان‌های محافظت می‌شوند. در مقایسه با سایر روش‌های مقابله با همشنوایی پیشنهاد شده، روش کدهای مقابله با همشنوایی انتخاب شده، سربار سیم، اتلاف توان و تأخیر کل را کاهش می‌دهد و همچنین کارایی بالاتری دارد و نسبت به نوسانات خطاهای تک بیتی نیز مطمئن‌تر است. اگر از روش SCAC در شبکه‌های روی تراشه استفاده شود می‌تواند تا حدود ۲۰٪ مساحت را ذخیره کند و همچنین به میزان ۴۹٪ اتلاف توان را کاهش دهد.

۴- تحلیل کدها

به منظور مشخص کردن میزان کارایی کدینگ‌های تشخیص و تصحیح خطا در شبکه‌های روی تراشه و با توجه به بررسی‌های انجام گرفته، نتایج پیاده‌سازی برخی از کدینگ‌ها از لحاظ انرژی مصرفی در جدول ۲ گردآوری شده است. لازم به ذکر است که تمامی پیاده‌سازی‌ها در فناوری ۹۰ نانومتر و بر اساس سلول‌های استاندارد کتابخانه‌ی CMP انجام گرفته‌اند. با توجه به نتایج این جدول و بررسی‌های انجام شده مشخص می‌شود که هرچه قابلیت کشف و تصحیح خطا بالاتر می‌رود، میزان انرژی مصرفی در کدگذار و کدگشای این کدینگ‌ها نیز افزایش می‌یابد [21]. به طوریکه کدینگ JTEC-SQED که قابلیت کشف چهار خطا و تصحیح سه خطا را دارد دارای بالاترین انرژی مصرفی می‌باشد. در بین سایر کدینگ‌هایی که تنها قابلیت تصحیح یک خطا را دارند، کدینگ شیفت کرانه‌ای دارای بالاترین مساحت، تأخیر و توان مصرفی است [22].

حاصل مقایسه‌ی کدینگ‌های اجتناب از همشنوایی از لحاظ توان، تأخیر، مساحت و انرژی مصرفی در جدول ۳ آورده شده است. پیاده‌سازی این کدینگ‌ها در فناوری ۱۳۰ نانومتر و برای یک گذرگاه ۳۲ بیتی صورت گرفته است. در این جدول لیستی از تعداد سیم‌ها، میزان تأخیر و سربارهای توان مصرفی برای کدهای مختلف برای گذرگاه داده‌ی ۳۲ بیتی به نمایش درآمده است. در جدول ۳ میزان سربار تعداد سیم‌های اضافه شده بیانگر آن است که استفاده از کدگذار-کدگشاهای مبتنی بر کدهای اجتناب از همشنوایی میزان تأخیر و مساحت مصرفی به همراه توان مصرفی میانگین را افزایش داده است ولی در طرف مقابل از داده‌های ارسالی به نحو مناسبی محافظت کرده است [12]. با توجه به مطالعات انجام گرفته تمامی این کدینگ‌ها میزان تأخیر گذرگاه را به مقدار $1+2\lambda$ می‌رسانند و همانطور که مشخص است کدگذار ممنوعه نسبت به سایر کدینگ‌های اجتناب از همشنوایی به تعداد سیم‌های بیش‌تری نیاز دارد ولی در عین حال مساحت، تأخیر و انرژی مصرفی کدگذار و کدگشای آن کمینه می‌باشد. همچنین لازم به ذکر است که میزان توان مصرفی هر روش وابسته به توزیع آماری داده‌های ارسالی با کدهای مذکور است. ولی در عین حال روش‌های مختلفی برای کاهش توان مصرفی خطوط ارتباطی بر روی یک تراشه وجود دارد، که معروف‌ترین آن‌ها کدگذاری خط با استفاده از اختلاف بیتی بین داده‌ی قبلی ارسال شده با داده‌ی جدیدی است که قرار است بر روی خط

که این روش برای تولید و محاسبه‌ی بیت توازن کلی به حلقه‌های طولانی از گیت‌های XOR نیاز دارند. اگر پهنای فلیت ارسالی نیز بزرگ باشد، ممکن است به سخت‌افزار پیچیده‌ای نیاز باشد که می‌تواند بر روی انرژی و زمان‌بندی سیستم و تأخیر مسیر بحرانی تأثیر منفی بگذارد [18]. روش JTEC را می‌توان به گونه‌ای تغییر داد که همزمان هم قابلیت تصحیح سه خطا را داشته باشد و هم بتواند چهار خطا را کشف کند. این روش تلفیقی از روش‌های JTEC و کشف همزمان چهار خطا^۱ می‌باشد که تحت عنوان JTEC-SQED شناخته می‌شود. در کدگذار این روش ابتدا داده‌های اصلی با استفاده از کدینگ همینگ SEC-DED کدگذاری می‌گردند. به این ترتیب کمترین فاصله‌ی همینگ بین کلمه‌های کد برابر با چهار خواهد شد. سپس یک نسخه از بیت‌های کدگذاری شده تهیه می‌شود، در نتیجه فاصله‌ی همینگ به هشت می‌رسد که به این ترتیب قابلیت کشف الگوهای خطای چهارتایی را خواهد داشت. این روش همانند روش JTEC توانایی مقابله با همشنوایی را نیز دارد. ولی در این روش علاوه بر بیت توازن اصلی، کپی آن نیز به داده‌ها اضافه می‌شود [20]. کدینگ تصحیح خطاهای چندگانه‌ی پیوسته^۲ می‌تواند خطاهای پیوسته با طول ۱۴ بیت را تصحیح کند و این درحالی است که از اشکال همشنوایی نیز جلوگیری می‌کند. علاوه بر آن، از یک الگوی کنترل خطای منطبق با روش کدگذاری ارائه شده برای بالا بردن قابلیت اطمینان ارتباطات نیز استفاده شده است. ماژول‌های اصلی این طرح شامل کد همینگ (۷،۴)، کدهای الگو ممنوعه، برگ‌برگ-ساز، کنترل‌کننده و واحد ارزیاب می‌باشد [21]. کدینگ خود تصحیح-کننده‌ی سبز^۳ می‌تواند قابلیت اطمینان بالا و مصرف پهنه‌ی انرژی در اتصالات شبکه‌های روی تراشه در مقیاس نانومتر را تضمین کند. کدینگ سبز، یک کدگذاری کم‌توان و شامل دو مرحله‌ی کدگذاری تصحیح خطای سه پیمان‌های^۴ و کدگذاری گذرگاه با استفاده از روش کدگذاری سبز است. استفاده از روش کدگذاری تصحیح خطای سه پیمان‌های منجر به بالا رفتن قابلیت اطمینان این روش می‌شود، به این صورت که فاصله‌ی همینگ هر دو کد با یکدیگر برابر با سه است و اگر بیش از دو بیت خطا در سه بیت داده وجود نداشته باشد، هر بیت می‌تواند به وسیله‌ی خودش تصحیح گردد. در واقع در این روش خطا توسط رأی اکثریت تصحیح می‌شود. لازم به ذکر است که تأخیر مسیر بحرانی کدگشای این روش برابر با بیش‌ترین تأخیر مدار رأی اکثریت آن است. پس تأخیر این روش از سایر کدینگ‌های تصحیح خطا بسیار کم‌تر است. این مدار از سایر روش‌های کدگذاری ساده-تر و کارآمدتر است و از مساحت کمتری استفاده می‌کند [22]. در [23] با استفاده از کدهایی که هم قابلیت مقابله با همشنوایی و هم تصحیح خطای تک بیتی را دارند یک طراحی شبکه روی تراشه‌ی قابل اطمینان معرفی شده است. چنین کدی، یک کد مقابله با همشنوایی انتخاب شده^۵، نامیده می‌شود، که در آن داده‌ها بر اساس الگوی کدهای مقابله با همشنوایی انتخاب شده، کدگذاری می‌شوند. به این ترتیب به سرعت و با قابلیت

¹ JTEC and Simultaneous quadruple-error-detection code (JTEC-SQED)

² Multiple-Continuos-Error-Correct-Coding (MCECC)

³ Self-Corrected Green Coding Scheme (S-C Green)

⁴ Triplication Error Correction Coding

⁵ Selected Crosstalk Avoidance Code (SCAC)

جدول ۲: مقایسه‌ی کدینگ‌های تشخیص و تصحیح خطا از لحاظ کارآیی [20]

Percentage Savings(%)	$E_{\text{savings/cycle}}$ (nJ)	$E_{\text{NoC/cycle}}$ (nJ)	$E_{\text{interface}}$ (pJ)	E_{switch} (pJ)	E_{decoder} (pJ)	E_{Link} (pJ)	Coding Scheme
۰	۰	۴.۵۲	۰	۴۰	۰	۱۷.۷۵	Uncoded
۲۹.۲	۱.۳۲	۳.۲۰	۰	۴۰	۰	۱۲.۷۸	Spacing
۱۷.۸	۰.۸۰	۳.۷۱	۰.۱۱	۴۰	۲.۳۸	۱۰.۹۳	ED
۳۳.۷	۱.۵۲	۲.۹۹	۰.۱۹	۴۰	۲.۶۱	۶.۷۳	DAP
۳۳.۶	۱.۵۲	۳.۰۰	۰.۱۹	۴۰	۲.۶۵	۶.۷۳	BSC
۳۳.۲	۱.۵۰	۳.۰۲	۰.۲۰	۴۰	۲.۶۳	۶.۸۳	MDR
۳۷.۸	۱.۷۱	۲.۸۱	۰.۲۳	۴۰	۲.۹۶	۵.۲۵	CADEC
۴۳.۶	۱.۹۷	۲.۵۵	۰.۲۳	۴۰	۳.۹۲	۲.۹۸	JTEC
۴۳.۶	۲.۱۱	۲.۴۱	۰.۲۳	۴۰	۳.۷۹	۲.۳۷	JTEC-SQED

جدول ۳: مقایسه‌ی کدینگ‌های اجتناب از هم‌شنوایی برای یک گذرگاه ۳۲ بیتی [12]

سربار کدگذار-کدگشا			گذرگاه			روش‌های کدگذاری
انرژی مصرفی (pj)	تأخیر (ps)	مساحت (um ²)	توان متوسط ($CV_{DD}^2 \times$)	تأخیر ($\tau \times$)	تعداد سیم‌ها	
۰	۰	۰	$8+15.5 \lambda$	$1+4 \lambda$	۳۲	بدون کد
۰	۰	۰	$8+15.5 \lambda$	$1+2 \lambda$	۶۳	محافظ‌گذاری
۹.۱۰	۱۷۸۴	۹۵۰۳	$13+13.0 \lambda$	$1+2 \lambda$	۵۲	کد الگو ممنوعه (۴،۵)
۵.۲۶	۲۵۰	۶۲۹۰	$13+13.0 \lambda$	$1+2 \lambda$	۵۲	کد همپوشان الگو ممنوعه (۴،۵)
۱.۶۱	۱۰۷	۱۸۳۰	$8.9+12.4 \lambda$	$1+2 \lambda$	۵۳	کد گذار ممنوعه (۴،۳)
۵.۷۸	۲۳۵	۹۷۶۰	$10.4+11.1 \lambda$	$1+2 \lambda$	۴۸	کد همپوشان گذار ممنوعه (۴،۳)

کدهای کنترل خطای قدیمی با استفاده از تصحیح خطاهای منطقی به خوبی می‌توانستند قابلیت اطمینان شبکه‌های روی تراشه را بهبود ببخشند، ولی در مقابل تأخیرهای ناشی از اثر هم‌شنوایی چندان مؤثر نبودند و همین امر منجر به پایین آمدن کارآیی سیستم و بروز خطاهای زمانی می‌شد. تمامی این عوامل سبب شد تا از کدگذاری‌هایی استفاده شود که علاوه بر داشتن قابلیت کنترل خطا، بتوانند با مشکل هم‌شنوایی نیز مقابله نمایند. در این مقاله انواع کدینگ‌ها از لحاظ توانایی کشف و تصحیح خطا، سربار توان، تأخیر و مساحت و همچنین از لحاظ نحوه‌ی کدگذاری داده‌ها با یکدیگر مقایسه شده‌اند و مشخص است که با توجه به نوع کاربرد می‌توان از کدهای مورد نظر استفاده نمود؛ ولی نکته حائز اهمیت که در تمامی کدینگ‌ها وجود دارد این است که کدینگ‌های اجتناب از هم‌شنوایی تعداد سیم‌های گذرگاه را افزایش می‌دهند ولی کدینگ‌های تصحیح و تشخیص خطا تعداد سیم‌ها را تغییر ندهد و در عوض سربار بیشتری از لحاظ توان مصرفی به سیستم اعمال می‌کنند.

۵- نتیجه‌گیری

امروزه با کاهش فناوری به مقیاس نانومتر، میلیاردها هسته‌ی پردازشی بر روی یک تراشه مجتمع گشته‌اند. شبکه روی تراشه یک راه‌حل کارآمد و مفید برای حل مشکل ارتباطات میان ماژول‌ها بر روی یک تراشه است. از طرفی با توجه به بیش‌تر شدن اثر هم‌شنوایی در شبکه‌های روی تراشه که منجر به بروز خطا و یا ایجاد تأخیر در سیم‌های گذرگاه می‌شود، مهم‌ترین چالش‌اتصالات در شبکه‌های روی تراشه قابلیت اطمینان، توان و تأخیر آن‌ها است. بنابراین از کدینگ‌های اجتناب از هم‌شنوایی و تشخیص و تصحیح خطا به منظور فراهم آوردن ارتباطات قابل اطمینان استفاده می‌شود. به دلیل کوچک شدن اندازه‌ی اتصالات، ارتباطات شبکه‌های روی تراشه در معرض خطاهای تصادفی چندگانه و یا قطاری قرار دارند. به همین علت به کدگذاری‌های کنترل خطای قدرتمندتری نیز نیاز است.

- [16] C. Duan, C. Zhu and S. P. Khatri, "Forbidden Transition Free Crosstalk," in *ACM/IEEE Design Automation Conference*, 2008.
- [17] Fu, Bo; Ampadu, Paul;, Error Control for Network-on-Chip Links, New York: Springer, 2012.
- [18] A. Garamoun, "Error Correction Techniques on NoC Protocol Layers," Stuttgart, 2009.
- [19] Ketan, Patel N; Igor, L. Markov;, "Error-Correction and Crosstalk Avoidance in DSM Busses," in *IEEE Transactions*, 2014.
- [20] Ganguly, Amlan; Pande, Partha Pratim; Belzer, Benjamin;, "Crosstalk-Aware Channel Coding Schemes for Energy Efficient and Reliable NOC Interconnects," in *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, Washington State, 2009.
- [21] Wang, Bin; Xie, Jing; Mao, Zhigang; Wang, Qin;, "Multiple Continuous Error Correct Code for High Performance Network-an-chip," in *IEEE Electronics - Asia Pacific Conference on Postgraduate Research in Microelectronics & Electronics*, 2011.
- [22] Huang, Po-Tsang; Fang, Wei-Li; Wang, Yin-Ling; Hwang, Wei;, "Low Power and Reliable Interconnection with Self-Corrected Green Coding Scheme for Network-on-Chip," in *Second ACM/IEEE International Symposium on Networks-on-Chip*, HsinChu, 2008.
- [23] Zhang, Ying; Li, Hua-Wei; Li, Xiao-Wei;, "Selected Crosstalk Avoidance Code for Reliable Network-on-Chip," *JOURNAL OF COMPUTER*, 2009.
- [1] س. کوهی, "ارزیابی توان مصرفی در شبکه های روی تراشه با همبندی توری," دانشگاه صنعتی شریف, تهران, ۱۳۸۶.
- [۲] ه. زمانی سبزی, "بهبود قابلیت اطمینان در شبکه های روی تراشه در مقابل خطای نرم با در نظر گرفتن وقوع خطای چندبیتی," دانشگاه صنعتی شریف, تهران, ۱۳۹۳.
- [3] Patooghy, Ahmad; Miremadi, Seyed Ghassem; Tabkhi, Hamed;, "A reliable and power efficient flow-control method to eliminate crosstalk faults in network-on-chips," *Elsevier*, vol. 35, no. 8, pp. 766-778, November 2011.
- [۴] وست, نیل; هریس, دیوید;, طراحی مدارهای مجتمع CMOS, تهران: مرکز چاپ و انتشارات دانشگاه شاهد, ۱۳۹۳.
- [۵] ا. پاطوقی, "کاهش اثر خطای نرم و اشکال همشنوایی در شبکه های روی تراشه," دانشگاه صنعتی شریف, تهران, ۱۳۸۹.
- [6] Shirmohammadi, Zahra; Miremadi, Seyed Ghassem;, "Crosstalk Avoidance Coding for Reliable Data Transmission of Nefwork on Chips," in *IEEE*, Tampere, 2013.
- [7] X. Wu and . Z. Yan, "CAC CODEC DESIGNS BASED ON NUMERAL SYSTEMS," in *IEEE Workshop on Signal Processing Systems*, 2009.
- [8] M. Mutyam, "Fibonacci Codes for Crosstalk Avoidance," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, 2012.
- [9] F. Shi, X. Wu and Z. Yan, "New Crosstalk Avoidance Codes Based on a Novel Pattern Classification," *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, 2012.
- [10] K. Sinha, R. Ghosh and B. P. Sinha, "A New Number System Using Alternate Fibonacci Numbers as the Positional Weights with Some Engineering Applications," Springer, 2014.
- [11] X. Wu and Z. Yan, "Efficient CODEC Designs for Crosstalk Avoidance Codes Based on Numeral Systems," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, April, 2011.
- [12] S. R. Sridhara, A. Ahmed and N. R. Shanbhag, "Area and Energy-Efficient Crosstalk Avoidance Codes for On-Chip Buses," in *Computer Design: VLSI in Computers and Processors*, 2004.
- [13] W.-W. Hsieh, P.-Y. Chen and C.-Y. Wang, "A Bus-Encoding Scheme for Crosstalk Elimination in High-Performance Processor Design," *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, vol. 26, no. 12, pp. 2222 - 2227, 2007.
- [14] K. Ramesh and E. Srinivas, "FPGA Implementation of Codec Design for Optimal," in *International Journal of Science and Research (IJSR)*, February 2013.
- [15] C. Duan, . K. Gulatit and S. P. Khatrit, "memory-based Cross-talk Canceling CODECs for On-chip Buses," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE*