Archive of SID

بررسی روش‌های مقابله با اشکال همشنوایی در سطح انتقال ثبات

زهرا شیرمحمدی [1]، سید قاسم میرعمادی [2]

[1] دانشجوی دکترای تخصصی، دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف
تهران، ایران shirmohammadi@ce.sharif.edu

[2] استاد، دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف
تهران، ایران miremadi@sharif.edu

با کوچک‌تر شدن اندازه فناوری ساخت، سرعت کلاک بیشتر می‌شود و تاخیر اتصال‌ها به طور فزاینده‌ای افزایش می‌یابد. به خصوص تاخیر انتشار گذرگاه‌های طویل در برخی طراحی‌ها باعث ایجاد محدودیت در سرعت می‌شود. یکی از مشکلات اصلی درطراحی‌های سابمایکرومتر عمیق گذرگاه-های سریع، تاخیر انتشار ناشی از تاثیر اشکال همشنوایی است. درمیان راهکارهای مختلف ارائه‌شده برای کاهش تاثیر اشکال همشنوایی، برخی از آن‌ها بر کاهش مصرف انرژی و برخی بر کاهش تاخیر متمرکز می‌شوند. بسیاری از راهکارهای کاهش اشکال همشنوایی شامل حدف یا کاهش الگوهای ناخواسته یا تحمل سربار مساحت ناشی از افزودن سیم‌ها در گذرگاه، مدارهای اضافی و... است. تاثیر ناشی از افزودن راهکارهای مقابله با اشکال همشنوایی نباید تنها با افزایش کارایی مقایسه شود؛ بلکه عواملی مثل سربار مساحت نیز مهم است و باید مورد بررسی قرارگیرند. در این مقاله برآنیم تا مروری بر موثرترین روش‌های کدگذاری در سطح انتقال ثبات که برای مقابله و کاهش اشکال همشنوایی ارائه شدند، بپردازیم.

کلمات کلیدی

اشکال همشنوایی، کد اجتناب از همشنوایی، کد غیرهم پوشان، کد یک لاندا، کد الگو ممنوعه

۱- مقدمه و تعریف مسئله

در سال‌های اخیر اشکال همشنوایی ما بین سیم‌های موازی و مجاور به یکی از مهم‌ترین چالش‌های قابلیت اطمینان در تراشه‌ها مبدل شده است [1]. روش‌های حفاظ‌گذاری [2] در سطح فیزیکی که یکی از ساده‌ترین تکنیک‌ها برای از بین بردن تاخیر ناشی از تداخل اشکال همشنوایی است، با وارد کردن سیم‌های محافظ بین سیم‌های کانال سعی در کاهش ظرفیت‌های جفت‌شدگی دارند اما، این روش نیاز به دو برابر کردن تعداد سیم‌ها دارد. بنابراین با ۲ برابر کردن تعداد سیم‌ها می توان تاخیر گذرگاه را تا ۵۰٪ کاهش داد. اما این روش دارای ۱۰۰٪ سربار فضایی است. البته حفاظ گذاری دو نوع فعال و غیرفعال دارد که سربار فضایی ناشی از نوع غیرفعال آن برای کاهش ۷۵٪ تاخیر گذرگاه، ۲۰۰٪ یا بیشتر می‌باشد [2]. روش‌های مختلفی برای مقابله با اشکال همشنوایی وجود دارد. در روش کدگذاری داده، کلمه داده توسط فرستنده به کد گذار ارسال می‌شود و عمل کدگشایی در کدگشا انجام می‌شود و کلمه داده تبدیل به کلمه کد می‌شود. کلمه کد بر روی گذرگاه انتقال می‌یابد و به دست گیرنده می‌رسد و سپس عمل کدگشایی در سمت گیرنده انجام می‌شود و کلمه داده حاصل می‌شود.

راهکارهای کدگذاری گذرگاه می‌تواند به همان مقدار راهکار حفاظ گذاری غیرفعال، با سربار فضایی کمتر باعث بهبود در تاخیر گذرگاه شود. این کدها با عنوان کدهای اجتناب از اشکال همشنوایی شناخته می‌شوند. کدگذاری‌های اجتناب از اشکال همشنوایی می‌توانند ۱- بدون حافظه [3] ۲- مبتنی بر حافظه باشند [3] . کدگذاری مبتنی بر حافظه کلمه کد را بر مبنای انتقال پیشین کلمه کد کنونی و قبلی که می‌خواهد منتقل شود، تولید می‌کنند. اگرچه این نوع کدگذاری کمتر نیاز به افزودن سیم در گذرگاه دارد، اما پیچیدگی این روش بیشتر است. کدگذاری بدون حافظه از یک روش کدگذاری ثابت برای ایجاد کلمه کد بر مبنای کلمه داده کنونی و کدگذاری بدون حافظه از کتاب کد متغیر با گذر زمان استفاده می‌کند.

ادامه این مقاله به شرح زیر است. در بخش دوم از این مقاله به بیان انگیزه از مطالعه‌ی اشکال همشنوایی و مشکلات ناشی از آن اشاره شده‌است. در بخش سوم و چهارم به بررسی روش‌های کدگذاری مبتنی بر حافظه و کدینگ بدون حافظه پرداخته‌ایم. نتیجه‌گیری نیز دربخش پنجم انجام شده‌است.

۲- اهمیت اشکال همشنوایی

همانطور که در [4] آورده شده است، تاخیر ناشی از i امین سیم که با T نمایش‌داده می‌شود، از رابطه ۱ به صورت زیر محاسبه می‌شود:

www.SID.ir

Archive of SID

جدول ۱: الگوهای کلاس‌های انتقال

کلاس انتقال	الگوهای انتقال
0C	↑↑↑ ↓↓↓
1C	-↑↑ -↓↓ ↑↑- ↓↓-
2C	-↑- -↓-
3C	-↑↓ ↑↓- ↓↑- ↑↓-
4C	↑↓↑ ↓↑↓

$$T = \tau_0\left[(1+2\lambda)\Delta_i^2 - \lambda\Delta_i(\Delta_{i-1}+\Delta_{i+1})\right]$$

بطوریکه λ برابر است با نسبت ظرفیت خازنی بین سیم‌های مجاور و خازن لود بین سیم i ام و زمین، τ_0 برابر است با تاخیر انتقال در سیم واحد.

Δ_i= ۱ (به ازای انتقال از ۰ به ۱)

Δ_i= -۱ (به ازای انتقال از ۱ به ۰)

Δ_i= ۰ (عدم وجود انتقال در سیم)

از آنجایی که فناوری‌های ساخت عمیق زیر-میکرون توسعه پیدا-کرده‌است، ظرفیت خازنی بین‌سیم (C_I) از ظرفیت خازنی سیم به زیرلایه (C_L) مهم‌تر شده‌است[4] اگر فرض کنیم یک گذرگاه گروهی متشکل از ۳ سیم داریم که به صورت $b_{i-1}b_i\ b_{i+1}$ نمایش داده - می‌شود، ظرفیت خازنی کل برای b_i به حالات دو سیم کناری بستگی دارد. در بهترین حالت که هر سه انتقال در یک سو است، ظرفیت خازنی $C_{min}=C_L$ و در بدترین حالت که دو انتقال b_{i-1} و b_{i+1} در خلاف جهت انتقال b_i است، ظرفیت خازنی $C_{max}=4\times C_I+C_L$ می-باشد. این نشان می‌دهد که تاخیر سیگنال‌های گذرگاه وابسته به الگوهای داده‌ای گذرگاه است. اشکال همشنوایی به گذارهای انتقالی بر روی سیم‌ها وابسته است که کلاس بندی و شدت تاخیر آنها به ترتیب در جدول شماره ۱ و جدول شماره ۲ آورده شده است.

در این مقاله انتقال از ۰ به ۱ بصورت ↑ ، انتقال از ۱ به ۰ بصورت ↓ و عدم تغییر در انتقال بصورت – نشان داده شده‌است.. همان گونه که اشاره شد کدینگ‌های اجتناب از همشنوایی در سطح انتقال ثبات از جمله بهترین روش‌های مقابله با اشکال همشنوایی هستند.

۳- کدینگ‌های حاقظه دار

در ادبیات موضوعی کدینگ حافظه دار در دو دسته کدینگ هرس نشده و کدینگ هرس شده دسته بندی می‌شوند. در ادامه با این دو دسته از کدینگ‌ها آشنا می‌شویم.

جدول ۲: مقادیر خازن کل به ازای کلاس‌های مختلف[3]

کلاس انتقال	C_T^{max}
0C	C_L
1C	$(1+\lambda)C_L$
2C	$(1+2\lambda)C_L$
3C	$(1+3\lambda)C_L$
4C	$(1+4\lambda)C_L$

۱-۲ کد هرس نشده حافظه دار

اگر کلمه داده b بیتی باشد و تمام الگوها برای کلمه کد مجاز باشند، به تعداد 2^n کلمه کد n بیتی برای این کلمه داده وجود دارد. اگر کد حافظه‌دار باشد، لزوما تمام کلمه‌های کد در طول زمان ثابت نخواهند بود. کلمه کد می‌تواند هر داده n بیتی باشد و روی گذرگاه قرار بگیرد. بیشترین تعداد سمبل‌ها برای کلمه کد توسط کلمه کد با کمترین درجه محدود می‌شود. بنابراین اولین گام محاسبه درجه کلمه کد و سپس مشخص کردن کلمه کد با کمترین درجه می‌باشد.

کلمه کد از کلاس ۱، کلمه کدی است که به طور ترتیبی بیت‌های ۰ و ۱ دارد. برای مثال برای کلمه کد 5- بیتی ۱۰۱۰۱ و ۰۱۰۱۰ را داریم.

در کلمه کد دو نوع مرز وجود دارد که عبارتند از: مرز مستقل و مرز وابسته. مرز مستقل در کلمه کد بین دو بیت همسایه با مقادیر مساوی و مرز وابسته بین دو بیت همسایه با مقادیر نامساوی رخ می‌دهد. وقوع مرز مستقل کلمه کد را بخش‌بندی می‌کند که هر بخش از کلاس ۱ می‌باشد. همچنین نوع کلاس مربوط به کلمه کد برابر با تعداد بخش-بندی‌های آن می‌باشد. برای محاسبه درجه کلمه‌های داده متفاوت در کلاس ۱ روابط زیر را داریم [3]:

قابل ذکر است ϕ تعداد سمبل‌ها می‌باشد و d_n درجه کلمه کد

$$d_n = \frac{1}{\phi+1/\phi}\left(\phi^{n+2} - (-\phi)^{-(n+2)}\right), \quad \phi = \frac{1+\sqrt{5}}{2} \qquad (۲)$$

$$d_n = \begin{cases} \frac{2}{\phi+1/\phi}\cosh\left((n+2)\ln(\phi)\right), & odd\ n \\ \frac{2}{\phi+1/\phi}\sinh\left((n+2)\ln(\phi)\right), & even\ n \end{cases}$$

n-بیتی است. افزایش n باعث می‌شود تا تعداد سمبل‌ها در ϕ یا ۱.۶۲ ضرب شود. بنابراین افزودن یک سیم فیزیکی امکان افزایش اطلاعات به اندازه $log_2^{1.62}$ را فراهم می‌کند.

۲-۲ کد هرس شده حافظه‌دار

کد هرس‌نشده[۲۰] ، با درجه کلمه کد کلاس ۱ محدود می-شود. بنابراین اگر کلمه‌های کد کلاس ۱ از کتاب کد حذف شوند، کارایی ضعیف آن‌ها باعث ایجاد محدودیت نمی‌شود. برای هرس کردن کتاب کد از الگوریتم زیر استفاده می‌شود[3]:

الگوریتم ۱: درحالی که کلمه‌های کد مجاز وجود دارند،

- گروهی از کلمه‌های معتبر با کمترین درجه پیدا شود.

برای هر کلمه کدW در گروه:

- W را از گروه کلمه‌های معتبر حذف کنید.
- درجه هر کدام از همسایه‌های W را کاهش دهید.

www.SID.ir

Archive of SID

با پیشرفت الگوریتم درجه محدودکننده افزایش می‌یابد، به مقدار ماکزیمم می‌رسد و با تهی شدن کتاب کد، کاهش پیدا می‌کند.
هرس با استفاده از الگوریتم ۱ فرآیندی تجربی است و محاسبه آن برای n بزرگ دشوار است (الگوریتم تنها برای nهای کوچکتر از ۲۳ قابل اجرا است). الگوریتم زیربهینه[۲۱] هرس وجود دارد که تقریب نزدیکی از بهینه[۲۲] است. اما وابستگی به کلاس در این الگوریتم کاملا قابل رویت است. الگوریتم ۲ اوضاع بهتری دارد.

الگوریتم ۲: به ازای هر c از 1 تا n:

- کلمه‌های کد از کلاس c از میان کلمه‌های کد حذف شود.
- درجه برای کلمه‌های کد برجای‌مانده بار دیگر محاسبه شود.

بار دیگر درجه محدودکننده افزایش می‌یابد ، به مقدار ماکزیمم می‌رسد و کاهش پیدا می‌کند. با این‌حال مشخص شده‌است که نتایج حاصل از این الگوریتم به ازای $n<۱۰$ با نتایج حاصل از الگوریتم بهینه کاملا هماهنگ است و خطا کمتر از ۱۰٪ به ازای $n<۲۳$ برایش ثابت است. پس این الگوریتم تقریب نسبتا خوبی است و چون زیربهینه است، نتایج آن همیشه قابل دسترسی هستند.

واضح است که یک گذرگاه ۳۲- بیتی می‌تواند تنها با ۴۰ سیم پیاده‌سازی شود. در این حالت سربار ناشی از سیم تنها ۲۵٪ در مقایسه با ۴۴٪ سربار ناشی از کد هرس‌نشده و سربار ۹۷٪ ناشی از حفاظ گذاری عادی می‌باشد.[3]

۴- کدینگ بدون حافظه

در این بخش به بررسی انواع کدینگ بدون حافظه می‌پردازیم.

۴-۱ کدینگ مبتنی بر گراف

کدهای گذشته به دیکدر و انکدر برای حفظ حالت‌ها نیاز داشتند؛ زیرا که کتاب کد وابسته به مقادیر قبلی کانال بود. در این حالت هدف دستیابی به بزرگترین کتاب کد می‌باشد. در نظریه گراف یک دسته از گراف بدون جهت به عنوان زیرگراف که در آن هر جفت از گره‌ها با لبه به هم متصل هستند، تعریف می‌شود. بنابراین اگر انتقال‌های مجاز در کتاب کد، معادل با لبه‌های گراف فرض شود، مشکل یافتن بیشترین مقدار کتاب کد بدون حافظه تبدیل به مشکل یافتن بزرگترین دسته در گراف می‌شود[3]. گروه همسایگی کتاب کد در کلاس‌های $c>1$ دسته نیست و نیز دسته بزرگتر از d_n وجود ندارد. بنابراین از الگوریتم ۳ استفاده می‌شود:

الگوریتم ۳: $x=y=1$

- برای هر مرز از ۱ تا n-1:
- اگر مرز از نوع 01 است، $y=x+y$
- اگر مرز از نوع ۱۰ است، $x=x+y$

جدول ۳: مقایسه سربار ناشی از کدها

سربار	تعداد سیم مورد نیاز	طول گذرگاه	کد
٪۴۴	۴۶	۳۲ بیت	کد هرس نشده حافظه دار
٪۲۵	۴۰	۳۲ بیت	کد هرس شده حافظه دار
٪۴۴	۴۶	۳۲ بیت	کد بدون حافظه
٪۹۷	۶۳	۳۲ بیت	حفاظ گذاری معمولی

- اندازه دسته نخست برابر است با: $x+y$

ماکزیمم کارایی کد خود محافظ بدون حافظه برابر $Log_2(d_n)$ بیت بر سیم می‌باشد.
جدول شماره ۳ مقایسه کدهای بررسی شده تا به حال را نشان می‌دهد.

۴-۲ کدهای اجتناب از اشکال همشنوایی

کدهای اجتناب از همشنوایی عبارتند از: کد غیرهم پوشان، کد یک لاندا، کد الگو ممنوعه. در هریک از این کدها سعی شده‌است تا تاخیر و یا مساحت بهینه شود که برحسب نیاز هریک از این کدها را می‌توان استفاده‌کرد. در ادامه به توضیح مشروحی از این روش‌های کدینگ خواهیم پرداخت.

۴-۲-۱ کدینگ یک لاندا

کدهای یک لاندا دارای حداکثر تاخیر $\tau_0 (1+\lambda)$ می‌باشد. همچنین انتقال‌های ۱۰→۰۱ و ۰۱→۱۰ در کدهای یک لاندا غیرمجاز است[5]. دو شرط اصلی برای این نوع کد عبارتند از:

۱- مرزهای آن از دو نوع ۱۰ , ۰۱ تناوبی تشکیل شده‌است.

۲- الگوهای بیتی ۰۱۱۰، ۱۰۰۱ ،۱۰۱۰ ،۱۰۱ در آن غیرمجاز می‌باشند.

۴-۲-۲ کد الگو ممنوعه و گذار ممنوعه

کدهای الگو ممنوعه و گذار ممنوعه دارای حداکثر تاخیر $\tau_0 (1+2\lambda)$ می‌باشند[8]-[6]. این دو نوع بیشتر از بقیه مورد مطالعه قرار گرفته‌اند. انتقال‌های ۱۰ → ۰۱ و ۰۱→۱۰در الگو ممنوعه غیرمجاز است. حداکثر اندازه کتاب کد در کدهای گذار ممنوعه n-بیتی F_{n+2} است که F_n دنباله فیبوناچی با مقادیر پایه $F_1=F_2=1$ و $F_n= F_{n-1}+ F_{n-2}$ به ازای $n>2$ می‌باشد. الگوهای بیتی ۰۱۰ و ۱۰۱در الگو ممنوعه غیرمجاز است. حداکثر اندازه کتاب کد در الگو ممنوعه n-بیتی، $2F_{n+1}$

www.SID.ir

Archive of SID

است که این مقدار بزرگتر از حداکثر اندازه متن کدهای گذار ممنوعه است. از آنجایی که اندازه متن مورد نیاز توانی از ۲ است، متغیر ۱ به صورت $F_{n+2}< 2^{l} \leq 2F_{n+1}$ مورد استفاده قرارمی‌گیرد کد الگو ممنوعه بر پایه دنباله عددی فیبوناپی رمزگذاری می‌شود. اما تعداد اعداد صحیحی که قابل نمایش با دنباله عددی فیبوناپی هستند برابر است با $\sum_{i=1}^{n}(F_i) = F_{n+2} + 1$ که کمتر از $2F_{n+2}$ که ماکزیمم کاردینالیتی کتاب کد الگو ممنوعه است، می‌باشد. بنابراین تنها می‌توان بخشی از کتاب کد الگو ممنوعه را توسط دنباله فیبوناچی کدگذاری کرد.

۳-۲-۴ کد غیرهم پوشان

واضح است که می‌توان تاخیر $(1+4\lambda)$ را حذف کرد بطوری‌که در خروجی از دو انتقال برای سه سیم مجاور پرهیز شود[9]. قابل ذکر است کلمه کدی که این ویژگی را داشته‌باشد، به عنوان‌کد غیرهم پوشان شناخته می‌شود. کد غیرهم پوشان دارای حداکثر تاخیر $\tau 0(1+3\lambda)$ می‌باشد. متن کد نمی‌تواند دو الگوی بیتی ۰۱۰ و ۱۰۱ که در اطراف موقعیت‌های بیتی ظاهر می‌شوند را دارا باشد. ماکزیمم اندازه غیرهم پوشان n-بیتی با T_n نمایش داده‌می‌شود.

$T_n= T_{n-۱}+ T_{n-۲}+ T_{n-۳}$; $n\geq ۴$

$T_۱=۲$, $T_۲=۴$, $T_۳=۷$

سیستم عددی یک سیستم زبان‌شناسی و نمادگذاری ریاضی برای نمایش عددهای یک گروه است. سیستم عددی‌هایی که بیشتر استفاده می‌شوند، سیستم عددی وابسته به موقعیت هستند. به ازای یک عدد طبیعی b، یک رشته $(a_n,\ldots,a_2,a_1)_b$ عدد $\sum_{i=1}^{m}(a_i b^{i-1})$ را نمایش می‌دهد. نوع بهینه کد غیرهم پوشان در [9] مطرح شده‌است که این مشکل را با استفاده از منطق اضافی برای رمزگذاری اعداد صحیح که نمی‌توانند با دنباله فیبوناچی کدگذاری شوند، حل می‌کند. در [12] از تغییر یافته دنباله فیبوناچی برای رمزگذاری کد غیرهم پوشان استفاده شده‌است. ماکزیمم تعداد اعداد صحیحی که قابل نمایش با تغییر یافته دنباله فیبوناچی هستند، برابر است با $2F_{m+2}$.

۵- جمع‌بندی و نتیجه‌گیری

در این مقاله انواع راه‌های مقابله با اشکال همشنوایی بررسی شد که با توجه به سربار قابل تحمل در شرایط متفاوت، امکان استفاده از هر روش برای کاهش یا حذف اشکال همشنوایی وجود دارد. از جمله روش‌های مقابله با اشکال همشنوایی حفاظ گذاری معمولی یا قرار دادن سیم‌ها بین سیم‌های گذرگاه بود اما این روش سربار زیادی داشت و همانطور که در جدول ۳ آورده شده‌است، سربار ناشی از آن ۹۷٪ می‌باشد. از جمله روش‌های دیگر برای مقابله با اشکال همشنوایی استفاده از عمل کدگذاری داده برای ارسال و کدگشایی آن در سمت گیرنده بود. برای این روش کدهای متفاوتی تعریف‌شده‌اند که در دو نوع حافظه‌دار و بدون حافظه هستند. انواع کدهای حافظه‌دار به کلمه-های کد فرستاده‌شده قبلی و کلمه‌های کد موجود روی باس کنونی وابسته‌اند بنابراین برای محافظت از مقادیر قبلی نیاز به حافظه دارند که این خود سربار سخت‌افزاری است. اما انواع بدون حافظه تنها به کلمه‌های کد موجود روی گذرگاه کنونی برای ارسال نیاز دارند. پس این نوع از کدها سربار سخت افزاری حافظه ندارند. هرکدام از انواع کدهای مبتنی‌برحافظه و بدون حافظه خود به دسته‌های متفاوتی تقسیم می‌شوند. انواع حافظه‌دار عبارتند از: ۱- کد هرس‌نشده حافظه-دار: که بیشترین تعداد کلمه کد آن با $Log_2(d_n)$ قابل محسابه است. پیاده‌سازی گذرگاه ۳۲-بیتی در آن نیاز به ۴۶ سیم دارد که به عبارتی سربار ناشی از آن ۴۴٪ است. ۲- کد هرش‌شده حافظه‌دار: که الگوریتم اصلی آن تنها برای $n<23$ قابل پیاده‌سازی است و به ازای $n>23$ از الگوریتم زیربهینه استفاده می‌شود. گذرگاه ۳۲ بیتی قابل پیاده‌سازی با ۴۰ سیم است به عبارتی سربار ناشی از آن ۲۵٪ می‌باشد که در مقایسه با سربار ۹۷٪ ناشی از حفاظ گذاری و سربار ۴۴٪ ناشی از کد هرس‌نشده حافظه‌دار، سربار کمتری است. کدهای بررسی شده بدون حافظه عبارتند از: ۱-کد خودمحافظ که برای پیاده‌سازی گذرگاه ۳۲ بیتی دارای ۴۴٪ سربار است که این سربار برابر با سربار ناشی از کد هرس‌نشده حافظه‌دار است.۲- کد غیرهم پوشان با استفاده از دنباله اعداد فیبوناچی که در این حالت تنها بخشی از کتاب کد الگو ممنوعه قابل رمزگذاری با دنباله فیبوناچی می‌باشد که محدودیت بزرگی است. ۲- الگو ممنوعه با استفاده از دنباله فیبوناچی بهبود یافته که در این حالت مشکل حالت قبلی رفع می‌شود ولی دارای سربار سخت‌افزاری برای بهش دیکدر است. ۳- کد کد غیرهم پوشان با انتقال C که در مقایسه با کد غیرهم پوشان معمولی، پیچیدگی سخت افزاری کمتر و نرخ کد بالاتری دارد. تفاوت اصلی این روش با کد **کد** غیرهم پوشان معمولی در این است که در خروجی‌های زوج دیکدر از معکوس‌کننده‌ها استفاده شده است.

[1] S. Mukherjee, "Architecture Design for Soft Errors," Boston, Morgan-Kaufmann, 2008.

[2] C. Duan, B. J. LaMeres, and S. P. Khatri, "On and Off-Chip Crosstalk Avoidance in: VLSI Design", Springer Publishing, 22 (4) (2010) 223–225.

[3] B. Victor and K. Keutzer, "Bus Encoding to Prevent Crosstalk Delay," in IEEE JACM International Conference on Computer Aided Design, 2001, pp. 57-63.

[4] P. Sotiriadis, and A. Chandrakasan, "Reducing Bus Delay in Sub-micron Technology Using Coding," in Proceedings of IEEE Asia and South Pacific Design Automation Conference (ASPDAC), pp. 109-114, 2001.

[5] X. Wu and Z. Yan, "Efficient CODEC Designs for Crosstalk Avoidance Codes Based on Numerical Systems," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 19(4), pp. 548-558, April 2011.

[6] Z. Shirmohammadi, S. G. Miremadi, "On Designing an Efficient Numerical-based Forbidden Pattern Free Crosstalk Avoidance Codec for Reliable Data Transfer of NoCs," Microelectronics Reliability, 63, pp. 304-313, 2016.

[7] Z. Shirmohammadi, S. G. Miremadi, "Addressing NoC Reliability Through an Efficient Fibonacci-Based Crosstalk Avoidance Codec Design," in Proceedings of IEEE International Conference Algorithms and Architectures for Parallel Processing (ICA3PP), pp. 756-770, 2015.

www.SID.ir

Archive of SID

[8] Z. Shirmohammadi and S. G. Miremadi, “Crosstalk Avoidance Coding for Reliable Data Transmission of Network on Chips,” in Proceedings of IEEE Symposium on System-on-Chip (SoC), pp. 1-4, 2013.

[9] C. S. Chang, J. Cheng, T. K. Huang and D. S. Lee, “Constructions of Memoryless Crosstalk Avoidance Codes Via C-Transform,” in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 22(9), pp. 2030-2033, September 2014.

.

www.SID.ir