# الگوریتم مسیریابی با قابلیت تحمل‌پذیری اشکال برای شبکه روی تراشه سه‌بعدی

مصطفی تقی‌زاده فیروزجایی، مجتبی ولی‌نتاج و مجتبی منصوری

**چکیده: کارایی شبکه‌های روی تراشه به طور گسترده‌ای به الگوریتم‌های مسیریابی به کار رفته در آنها وابسته است. در سالیان اخیر، الگوریتم‌های مسیریابی زیادی برای شبکه‌های روی تراشه دوبعدی و سه‌بعدی طراحی‌شده است. شبکه روی تراشه سه‌بعدی که برای افزایش کارایی شبکه روی تراشه دوبعدی معرفی گردیده، از ترکیب مفاهیم شبکه روی تراشه و مجتمع‌سازی سه‌بعدی به وجود آمده است. در این گونه مدارها عناصر نیمه‌هادی به روشی خاص به صورت پشته‌ای روی یکدیگر قرار می‌گیرند. به دلیل تأثیرات قابل توجهی که اشکال‌های لینک‌ها یا گره‌های شبکه روی تراشه بر عملکرد مدار می‌گذارند، الگوریتم‌های مسیریابی بایستی روش‌هایی را به کار گیرند تا از تأثیرات اشکال جلوگیری نمایند. این ویژگی خصوصاً در شبکه روی تراشه سه‌بعدی که احتمال رخداد اشکال در لینک‌های عمودی آن قابل توجه است، اهمیت بیشتری دارد. در این مقاله، یک روش جدید برای مسیریابی در شبکه روی تراشه سه‌بعدی به نام FT-ZXY معرفی می‌شود که بدون استفاده از کانال‌های مجازی و در نتیجه با سربار سخت‌افزاری ناچیز، قابلیت تحمل اشکال‌های منفرد در لینک‌های افقی و اشکال‌های چندگانه در لینک‌های عمودی را دارد. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم مسیریابی پیشنهادی از نظر پارامترهای ارزیابی مانند تأخیر، قابلیت اطمینان، سربار سخت‌افزاری و توان مصرفی، عملکرد بهتری نسبت به الگوریتم‌های مطرح‌شده قبلی دارد.**

***کلیدواژه:*** **شبکه روی تراشه، شبکه روی تراشه سه‌بعدی، الگوریتم مسیریابی، تحمل‌پذیری اشکال، قابلیت اطمینان.**

## ۱- مقدمه

با افزایش چگالی تعداد ترانزیستورها در سطح تراشه، امکان به کارگیری ده‌ها و صدها هسته پردازشی که شامل واحدهای پردازشی و حافظه است، در سامانه روی تراشه که تمامی مؤلفه‌های مداری را بر روی یک تراشه یک‌پارچه می‌کند، به وجود آمده است و لذا صنعت ریزپردازنده‌ها در حال حرکت از پردازنده‌های تک‌هسته‌ای قبلی به پردازنده‌های چندهسته‌ای کنونی [۱] و [۲] و نهایتاً به پردازنده‌های بسیارهسته‌ای شامل چندصد تا چندین هزار هسته پردازشی است. همچنین با پیشرفت روزافزون فناوری ساخت مدارهای مجتمع، این تعداد هسته‌های پردازشی و واحدهای وابسته به آن نیز افزایش خواهد یافت. بدین ترتیب با توجه به عدم مقیاس‌پذیری گذرگاه‌های سنتی و افزایش حجم و پیچیدگی، یک الگوی ارتباطی درون‌تراشه‌ای معرفی شده که به این زیرساخت کارا و مقیاس‌پذیر برای ارتباطات روی تراشه، شبکه روی تراشه گفته می‌شود [۱] و [۳].

روند رو به رشد فناوری و افزایش تعداد ترانزیستورها منجر به ظهور مدارهای مجتمع سه‌بعدی شده که این نوع مدارها نسبت به نوع دوبعدی خود مزیت‌هایی از قبیل نداشتن مشکل سیم‌های بلند عمومی و کاهش تأخیر در انتقال داده‌ها و همچنین پشتیبانی از هم‌بندی‌های نامتجانس را دارا است. در این فناوری عناصر نیمه‌هادی از طریق TSV[1] روی یکدیگر به صورت پشته قرار می‌گیرند.

در سال‌های اخیر با ترکیب دو مفهوم شبکه روی تراشه و مجتمع‌سازی سه‌بعدی، مفهوم جدیدی به نام شبکه روی تراشه سه‌بعدی معرفی شده است. یکی از مهم‌ترین موضوعاتی که در شبکه روی تراشه سه‌بعدی مطرح می‌شود طراحی الگوریتم مسیریابی کارا برای این نوع از شبکه‌ها است. طراحی یک الگوریتم مناسب می‌تواند تأخیر را کاهش دهد و همچنین الگوریتم مسیریابی تأثیر مستقیمی بر میزان توان مصرفی دارد، چرا که سادگی الگوریتم مسیریابی، سادگی مسیریاب شبکه روی تراشه را در پی دارد و این امر سربار سخت‌افزاری را به طور قابل ملاحظه‌ای کاهش می‌دهد.

پیشرفت صنعت نانوالکترونیک پیامدهایی از جمله کوچک‌تر شدن ظرفیت خازن‌های مدار، افزایش فرکانس کاری، کاهش سطوح ولتاژ منبع تغذیه و غیره را به همراه دارد. این پیامدها، حساسیت گیت‌ها، لینک‌های ارتباطی و واحدهای حافظه مورد استفاده را در برابر انواع نویزهای محیطی که می‌توانند منجر به ایجاد اشکال‌های گذرا و دائمی شوند، افزایش داده است [۴]. به همین دلیل طراحی سامانه‌های تحمل‌پذیر اشکال که بتوانند مانع از تولید نتایج نادرست یا آثار مخرب شوند و یا با استفاده از روش‌هایی بتوانند خود را از وضعیت اشکال ایجادشده در سامانه یا شکست کامل سامانه بازنمایی نمایند، اهمیت ویژه‌ای یافته است [۵].

طراحی الگوریتم‌های مسیریابی با قابلیت تحمل‌پذیری اشکال در شبکه روی تراشه یکی از کارامدترین روش‌ها برای افزایش قابلیت اطمینان تراشه‌ها است. در این مقاله، یک الگوریتم مسیریابی کارا برای شبکه روی تراشه معرفی شده که دارای قابلیت تحمل‌پذیری اشکال است. هدف این است که الگوریتم مسیریابی پیشنهادی در عین سادگی و بهره‌گیری از سربار سخت‌افزاری اندک، کارایی قابل قبولی را فراهم آورد، لذا برای اجتناب از سربار سخت‌افزاری زیاد، عدم رخداد بن‌بست بدون استفاده از کانال مجازی و با بهره‌گیری از یک الگوریتم مبتنی بر چرخش فراهم خواهد شد، چرا که کانال مجازی سربار سخت‌افزاری زیادی را تحمیل می‌کند و همچنین برای پایین‌آوردن متوسط تأخیر بسته، سعی می‌شود کوتاه‌ترین مسیر به سمت مقصد پیدا شود.

در ادامه و در بخش دوم، مقالات پیشین در زمینه الگوریتم‌های

این مقاله در تاریخ ۲۰ دی ماه ۱۳۹۳ دریافت و در تاریخ ۹ مهر ماه ۱۳۹۴ بازنگری شد.

مصطفی تقی‌زاده فیروزجایی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی نوشیروانی بابل، بابل، (email: m.taghizade@nit.ac.ir).

مجتبی ولی‌نتاج، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی نوشیروانی بابل، بابل، (email: m.valinataj@nit.ac.ir).

مجتبی منصوری، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی نوشیروانی بابل، بابل، (email: mansoori@nit.ac.ir).

1. Through Silicon Via

Archive of SID

www.SID.ir

مسیریابی در شبکه روی تراشه سه‌بعدی مورد بررسی قرار می‌گیرد. در بخش سوم، الگوریتم مسیریابی پیشنهادی با قابلیت تحمل‌پذیری اشکال برای شبکه روی تراشه سه‌بعدی معرفی می‌شود. در بخش چهارم، روش جلوگیری از بن‌بست برای الگوریتم پیشنهادی تشریح شده و در بخش پنجم، نتایج ارزیابی‌ها و مقایسه با دیگر الگوریتم‌ها بیان می‌شود. در انتها در بخش ششم نتیجه‌گیری مقاله ارائه می‌شود.

## ۲- کارهای مرتبط

به طور کلی الگوریتم‌های مسیریابی برای شبکه روی تراشه دوبعدی و سه‌بعدی از جنبه‌های متفاوتی تقسیم‌بندی می‌شوند. این الگوریتم‌ها از نظر تطبیق‌پذیری به سه دسته تطبیقی، نیمه‌تطبیقی و قطعی و از نظر تصمیم مسیریابی به دو دسته توزیع‌شده و متمرکز و همچنین از نظر پیاده‌سازی به دو دسته مسیریابی مبتنی بر جدول و مسیریابی مبتنی بر ماشین حالت تقسیم‌بندی می‌شوند [۱]. ویژگی مهم دیگری که بر پایه آن می‌توان الگوریتم‌های مسیریابی برای شبکه روی تراشه را تقسیم‌بندی کرد، روش جلوگیری از رخداد بن‌بست است. بر این مبنا می‌توان الگوریتم‌ها را به دو دسته مبتنی بر چرخش و الگوریتم‌هایی که از کانال مجازی استفاده می‌کنند، تقسیم‌بندی نمود.

در شبکه روی تراشه سه‌بعدی، یکی از الگوریتم‌های مسیریابی تحمل‌پذیر اشکالی که از کانال مجازی استفاده می‌کند، الگوریتم RILM است که در [۶] معرفی شده است. این الگوریتم یک الگوریتم مسیریابی قابل اطمینان مناسب برای شبکه روی تراشه سه‌بعدی با هم‌بندی‌های ناهمگن در لایه‌های مختلف است که برای مهارکردن اشکال در لینک‌های عمودی ضعیف به کار می‌رود. این الگوریتم مستقل از این که در لایه‌های افقی چه الگوریتم مسیریابی و یا چه همبندی به کار رفته است، اشکال‌های چندگانه را در لینک‌های عمودی تحمل می‌کند. الگوریتم دیگر، FTDR سه‌بعدی که در [۷] معرفی شده و یک الگوریتم تحمل‌پذیر اشکال با سربار کم است که در آن از یک جدول برای لایه توری دوبعدی (افقی) و دو بردار وضعیت TSV به جای مسیریابی عمومی (یک جدول برای کل شبکه) استفاده شده است. در این الگوریتم از همبندی Nostrum استفاده شده و اشکال‌ها به دو دسته اشکال‌های افقی و اشکال‌های عمودی (TSV) تقسیم‌بندی شده‌اند. همچنین از یک بردار شش‌بیتی برای وضعیت اشکال شش لینک متصل به مسیریاب استفاده شده است. الگوریتم دیگر که در [۸] به نام AdaptiveZ معرفی شده است از ساختار NOC-Bus ۳D Hybrid استفاده می‌کند که در آن مسیریاب‌ها در یک ساختار سه‌بعدی قرار دارند. لینک‌های عمودی این ساختار سه‌بعدی از نوع گذرگاه است. لذا اشکال در این لینک عمودی برای همه لایه‌ها در نظر گرفته می‌شود. در این الگوریتم تمهیداتی برای تعادل ترافیکی و تعادل گرمایی نیز به کار گرفته شده است.

الگوریتم‌های مسیریابی تحمل‌پذیر اشکال مبتنی بر چرخش که برای شبکه روی تراشه سه‌بعدی در سال‌های گذشته معرفی شده‌اند، شامل الگوریتم مسیریابی AFRA است که در [۹] معرفی شده و تحمل‌پذیری اشکال را در لینک‌های عمودی شبکه با ساختار توری سه‌بعدی بررسی می‌کند. این الگوریتم اشکال‌های دایمی را بر روی لینک‌های عمودی در نظر گرفته و بر این حقیقت متکی است که نرخ اشکال در لینک‌های عمودی بیشتر از نرخ اشکال در لینک‌های افقی است. این الگوریتم مسیریابی در غیاب اشکال از الگوریتم ZXY و در حضور اشکال از الگوریتم XZXY استفاده می‌کند. الگوریتم مبتنی بر چرخش دیگر به نام HAMFA که در [۱۰] معرفی شده است، بدون استفاده از کانال مجازی و بسته‌های افزونه، اشکال‌های یک‌جهته را در لینک‌های عمودی و افقی تحمل می‌کند. این الگوریتم از مسیر هامیلتونی استفاده می‌کند و مسیریابی را بدون نیاز به جدول و یا اطلاعات اضافی در سرآیند بسته انجام می‌دهد و تنها نیاز به اطلاعات اشکال در لینک‌های مجاور خود دارد. این مسأله کارایی این الگوریتم را بالا برده و سربار سخت‌افزاری آن را کاهش داده است.

## ۳- الگوریتم مسیریابی پیشنهادی

در این بخش، یک الگوریتم مسیریابی به نام FT_ZXY معرفی می‌شود که قابلیت تحمل اشکال‌های چندگانه در لینک‌های عمودی و اشکال منفرد در لینک‌های افقی را دارد. این الگوریتم مبتنی بر مدل چرخش بوده که با محدودکردن برخی از چرخش‌ها از وقوع بن‌بست جلوگیری می‌کند.

یک از مشکلات مهمی که در شبکه روی تراشه وجود دارد وقوع بن‌بست در شبکه است. برای جلوگیری از بن‌بست دو روش عمده وجود دارد، اول استفاده از کانال مجازی و دوم روش‌های مبتنی بر مدل چرخش (روش مسیریابی انحرافی نیز وجود دارد که جزء روش‌های عمده محسوب نمی‌شود). از آنجایی که استفاده از تعداد کانال مجازی بیشتر منجر به پیچیدگی بیشتر شبکه و هزینه‌های مساحت و توان مصرفی بیشتر می‌گردد، الگوریتم مسیریابی که به تعداد کانال مجازی کمتری احتیاج دارد و یا به هیچ کانال مجازی احتیاج ندارد سودمندتر است. این مسأله در مورد الگوریتم‌های مسیریابی تحمل‌پذیر اشکال اهمیت بیشتری دارد، زیرا از یک طرف این الگوریتم‌ها خود پیچیده بوده و از طرف دیگر برای تحمل حالت‌های گوناگون خرابی در شبکه به کانال‌های مجازی بیشتری احتیاج دارند. بنابراین روش‌های مبتنی بر مدل چرخش برای شبکه روی تراشه سودمند هستند اگرچه نسبت به روش‌های مبتنی بر کانال مجازی تطبیق‌پذیری کمتری دارند.

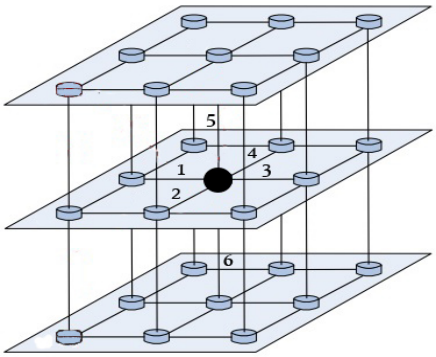
در این مقاله فرض شده که همبندی شبکه، توری سه‌بعدی و اشکال‌هایی که در آن رخ می‌دهند از نوع اشکال‌های دایمی است. مدل اشکال مورد استفاده در این الگوریتم بر پایه این فرض است که هر گاه هر نوع اشکال دایمی در هر کدام از جهت‌های یک لینک دوجهته رخ دهد، کل آن لینک خراب فرض می‌شود. علاوه بر این فرض شده است که یک روش تست، مانند مکانیزم BIST [۱۲]، لینک‌های معیوب را تشخیص می‌دهد.

مسیریاب‌های استفاده‌کننده از الگوریتم FT_ZXY برای مسیریابی بسته‌ها باید از اطلاعات اشکال در لینک‌های مجاور مطلع باشند. برای ذخیره وضعیت لینک‌های مجاور از یک ثبات بازپیکربندی استفاده می‌شود. اطلاعات اشکال به صورت یک گام است یعنی هر گره، تنها از وضعیت لینک‌های متصل به خود باخبر است. به عبارت دیگر اطلاعات ثبات بازپیکربندی شامل W,E,D,U,S,N است که در آن '۱'شدن N به معنی قرارداشتن گره در شمال لینک خراب و '۱'شدن D به معنی قرارداشتن گره در پایین لینک خراب متصل به آن است. همچنین '۱'شدن W به معنی خرابی لینک شرقی متصل به گره است (S، U و E هم به همین صورت). شکل ۱ یک مسیریاب را به صورت گره مرکزی به همراه شش لینک در همسایگی آن نشان می‌دهد.

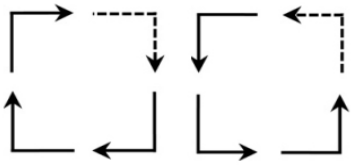
روش ارائه‌شده بر پایه الگوریتم شناخته‌شده قطعی ZXY است. به عبارتی دیگر این الگوریتم در نبود اشکال مانند الگوریتم ZXY عمل می‌کند. در الگوریتم ZXY اگر مبدأ و مقصد در دو لایه مجزا باشند، ابتدا بسته در جهت z به سمت لایه مقصد ارسال می‌شود و سپس بسته در

Archive of SID

www.SID.ir




شکل ۱: ۶ لینک در همسایگی یک مسیریاب (گره پر).



شکل ۲: چرخش‌های ممنوع (شرق- جنوب و شمال- غرب).

مسیر x فرستاده می‌شود تا زمانی که فاصله گره فعلی و گره مقصد در جهت x برابر صفر شود و سپس در جهت y به سمت مقصد فرستاده می‌شود. اما در حضور اشکال بر حسب این که اشکال در لینک‌های افقی یا عمودی رخ دهد، الگوریتم به دو بخش الگوریتم مسیریابی درون‌لایه‌ای و الگوریتم مسیریابی بین‌لایه‌ای تقسیم می‌شود.

### ۳-۱ الگوریتم مسیریابی درون‌لایه‌ای

هنگامی که یکی از لینک‌های مجاور گره مبدأ خراب باشد، اگر این لینک در مسیر ارسال بسته از مبدأ به سمت مقصد باشد باید یک مسیر جایگزین پیدا شود و این مسیر نباید به صورتی باشد که بن‌بست ایجاد شود. برای جلوگیری از بن‌بست، چرخش‌های شرق-جنوب و شمال-غرب ممنوع شده که مشابه روش Negative-First ارائه‌شده در [۱۱] است. در شکل ۲ چرخش‌های ممنوع نشان داده شده است. این محدودیت‌ها این اطمینان را می‌دهد که در گراف شبکه حلقه ایجاد نمی‌شود که نتیجه آن عدم رخداد بن‌بست است.

جدول ۱ بر حسب موقعیت مقصد نسبت به گره فعلی و همچنین وضعیت ثبات اشکال، پورت خروجی را مشخص می‌کند. در صورتی که پورت خروجی مورد نظر وجود نداشته باشد یا به عبارتی دیگر اگر گره فعلی در لبه توری قرار داشته باشد، از پورت جایگزین مشخص‌شده استفاده می‌شود.

پیروی از قوانین این جدول این اطمینان را می‌دهد بدون این که بن‌بست اتفاق بیفتد، وجود یک اشکال به طور کامل تحمل شود. شبه‌کد این الگوریتم در شکل ۳ نشان داده شده است.

مثالی از الگوریتم درون‌لایه‌ای در شکل ۴ ارائه شده است. همان طور که مشاهده می‌شود، مقصد در شمال شرق مبدأ قرار دارد و همچنین لینک بین گره‌های ۹ و ۱۰ خراب است، لذا مسیر منحرف می‌شود و این انحراف نباید به صورتی باشد که چرخش ممنوع ایجاد شود.

### ۳-۲ الگوریتم مسیریابی بین‌لایه‌ای

در این قسمت هدف این است که بسته به لایه مقصد فرستاده شود. روش ارائه‌شده به گونه‌ای عمل می‌کند که هم از بن‌بست جلوگیری شود و هم مسیری با کوتاه‌ترین اندازه برای ارسال بسته‌ها به مقصد پیدا شود. بنابراین روش کار الگوریتم مسیریابی بین‌لایه‌ای بدین صورت است: ابتدا

جدول ۱: تعیین خروجی بر حسب موقعیت مقصد و وضعیت اشکال.

| موقعیت مقصد | وضعیت ثبات اشکال ('۱' شدن) | پورت خروجی | در لبه توری |
|---|---|---|---|
| NW | به غیر از E | W | – |
| NW | E | S | N |
| SW | به غیر از E | W | – |
| SW | E | S | – |
| NE | به غیر از W | E | – |
| NE | W | N | – |
| SE | به غیر از W | E | – |
| SE | W | S | – |
| E | به غیر از W | E | – |
| E | W | S | N |
| W | به غیر از E | W | – |
| W | E | S | N |
| N | به غیر از S | N | – |
| N | S | W | E |
| S | به غیر از N | S | – |
| S | N | W | E |

بر حسب این که مقصد در لایه بالایی یا لایه پایینی نسبت به لایه فعلی قرار داشته باشد، بسته در مسیر بالا و یا پایین گره فعلی فرستاده می‌شود. اما اگر لینک بالایی یا پایینی خراب باشد، اگر مختصات بعد y گره فعلی فرد باشد، بسته به سمت شرق و در غیر این صورت به سمت غرب فرستاده می‌شود. در صورتی‌که بسته به لبه شبکه برسد، اگر مختصات بعد x فرد باشد بسته به سمت جنوب و در غیر این صورت، به سمت شمال فرستاده می‌شود. بدین ترتیب مسیری مناسب برای بسته پیدا می‌شود. شبه‌کد الگوریتم بین‌لایه‌ای در شکل ۵ نشان داده شده است.

در FT_ZXY اگر اشکال در چهار گوشه توری رخ دهد، پشتیبانی نمی‌شود. به عنوان مثال در شکل ۶ که مثالی از مسیریابی بین‌لایه‌ای را نشان می‌دهد، اشکال در لینک‌های بالا و پایین گره‌های ۹، ۱۵، ۱۷ و ۱۱ پشتیبانی نمی‌شود. این مشخصه در قسمت ارزیابی قابلیت اطمینان الگوریتم لحاظ شده است.

## ۴- بن‌بست

بن‌بست وقتی اتفاق می‌افتد که دو یا چند بسته منتظر منبعی باشند که به وسیله یکدیگر اشغال شده‌اند. این مسأله ممکن است منجر به حلقه انتظار نامحدود شود که باعث کاهش شدید در کارایی شبکه می‌شود.

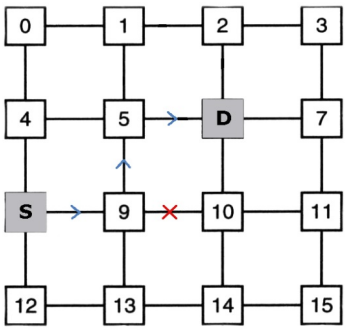
در شبکه روی تراشه سه‌بعدی، بن‌بست می‌تواند به دو صورت بن‌بست صفحه‌ای (حلقه در لایه همسان اتفاق می‌افتد) یا بن‌بست سه‌بعدی (حلقه شامل لینک عمودی است) وجود داشته باشد. شکل ۷ این دو نوع بن‌بست را نشان می‌دهد [۱۳].

بن‌بست می‌تواند به وسیله یکی از دو روش شناسایی و ترمیم و یا اجتناب مدیریت شود. تکنیک شناسایی و ترمیم اجازه ایجاد بن‌بست در شبکه را می‌دهد ولی با یک روش شناسایی، بن‌بست را یافته و آن را ترمیم می‌کند. یک مثال از این تکنیک از بین بردن بسته‌ای که به مدت طولانی در بافر مانده و ارسال دوباره آن بعد از یک زمان مشخص است. روش دیگر استفاده از سخت‌افزار اضافی است تا بسته متوقف را از یک راه فرعی با استفاده از روش چندکانالی خارج کند (مانند کانال مجازی). روش شناسایی و ترمیم دارای سربار توان مصرفی، مساحت و کارایی است.

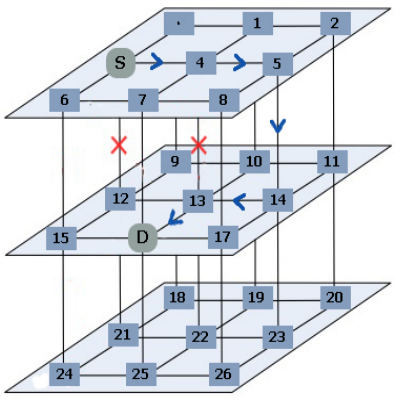
Archive of SID
www.SID.ir

```
if (destination position = Local)
    output direction = Local;
else if (destination position = south-west)
    if (the left link is not faulty)
        output direction = WEST;
    else
        output direction = SOUTH;
else if (destination position = north-west or west)
    if (the left link is not faulty)
        output direction = WEST;
    else if (current node in south border)
        output direction = NORTH;
    else
        output direction = SOUTH;
else if (destination position = north-east)
    if (the right link is not faulty)
        output direction = EAST;
    else
        output direction = NORTH;
else if (destination position = east)
    if (the right link is not faulty)
        output direction = EAST;
    else
        output direction = SOUTH;
else if (destination position = south-east)
    if (the right link is not faulty)
        output direction = EAST;
    else
        output direction = SOUTH;
else if (destination position = north)
    if (the north link is not faulty)
        output direction = NORTH;
    else if (current node in west border)
        output direction = EAST;
    else
        output direction = WEST;
else if (destination position = south)
    if (the south link is not faulty)
        output direction = SOUTH;
    else if (current node is west border)
        output direction = EAST;
    else
        output direction= WEST;
```

شکل ۳: شبه‌کد الگوریتم مسیریابی درون لایه‌ای.




شکل ۴: مسیریابی از S به D با استفاده از الگوریتم درون‌لایه‌ای.




شکل ۶: مثالی از مسیریابی بین‌لایه‌ای.

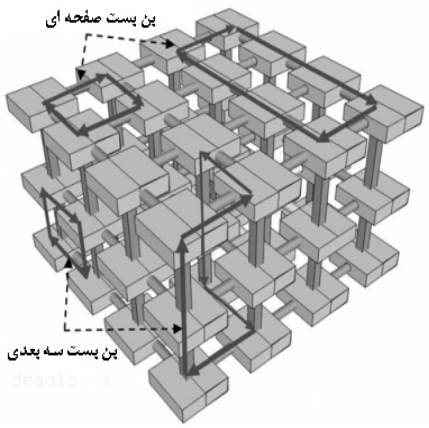
```
if (destination is in up layer) {
    if (the up link is not faulty)
        output direction = UP;
    else if (x-even & y-even) {
        if (current node is not in west border)
            output direction = WEST;
        else if (current node is not in north border)
            output direction = NORTH;
    }
    else if (x-even & y-odd) {
        if (current node is not in east border)
            output direction = EAST;
        else if (current node is not in north border)
            output direction= NORTH;
    }
    else if (x-odd & y-even) {
        if (current node is not in west border)
            output direction = WEST;
        else if (current node is not in south border)
            output direction = SOUTH;
    }
    else if (x-odd & y-odd) {
        if (current node is not in east border)
            output direction = EAST;
        else if (current node is not in south border)
            output direction = SOUTH;
    }
}
else if (destination is in down layer) {
    if (the down link is not faulty)
        output direction = DOWN;
    else if (y-even & x-even) {
        if (current node is not in west border)
            output direction = WEST;
        else if (current node is not in north border)
            output direction = NORTH;
    }
    else if (y-odd & x-even) {
        if (current node is not in east border)
            output direction = EAST;
        else if (current node is not in north border)
            output direction = NORTH;
    }
    else if (y-even & x-odd) {
        if (current node is not in west border)
            output direction = WEST;
        else if (current node is not in south border)
            output direction = SOUTH;
    }
    else if (y-odd & x-odd) {
        if (current node is not in east border)
            output direction = EAST;
        else if (current node is not in south border)
            output direction = SOUTH;
    }
}
else if (destination is in the same layer)
    output direction = direction of intra layer routing;
```

شکل ۵: شبه‌کد الگوریتم مسیریابی بین لایه‌ای.

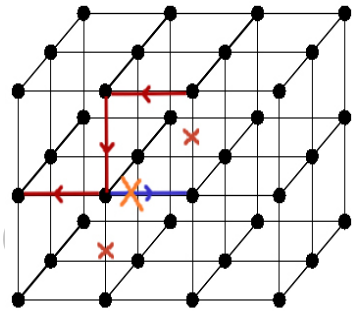
راه حل دیگر که به کلی از ایجاد بن‌بست جلوگیری می‌کند، روش اجتناب از بن‌بست است. در این روش با ممنوع‌کردن چرخش‌های معین، از ایجاد حلقه وابستگی جلوگیری می‌شود و این به معنی عدم رخداد بن‌بست است. ایجاد محدودیت به وسیله ممنوع‌کردن چرخش‌ها درجه تطبیق‌پذیری را کاهش می‌دهد که موجب کاهش کارایی می‌شود اما در حالت کلی به علت مصرف توان کمتر روش مقرون به صرفه‌تری است.

در روش پیشنهادی برای این که نشان دهیم الگوریتم در حالت کلی فاقد بن‌بست است، کافی است در قدم اول اثبات کنیم مسیریابی درون‌لایه‌ای فاقد بن‌بست است و در قدم دوم اثبات کنیم مسیریابی بین‌لایه‌ای فاقد بن‌بست است.

Archive of SID
www.SID.ir




شکل ۷: انواع بن‌بست در شبکه روی تراشه سه‌بعدی [۱۳].



شکل ۸: عدم ایجاد حلقه وابستگی در صفحه عمودی.

## ٤-١ بن‌بست صفحه‌ای

همان طور که ذکر شد، مسیریابی درون‌لایه‌ای در غیاب اشکال همانند الگوریتم مسیریابی XY عمل می‌کند. در الگوریتم XY بن‌بست رخ نمی‌دهد زیرا بسته‌ها اجازه ندارند هنگامی که در بعد y حرکت می‌کنند به چپ یا راست تغییر جهت دهند، لذا مسیریابی درون‌لایه‌ای در بیرون از همسایگی لینک خراب، بدون بن‌بست است.

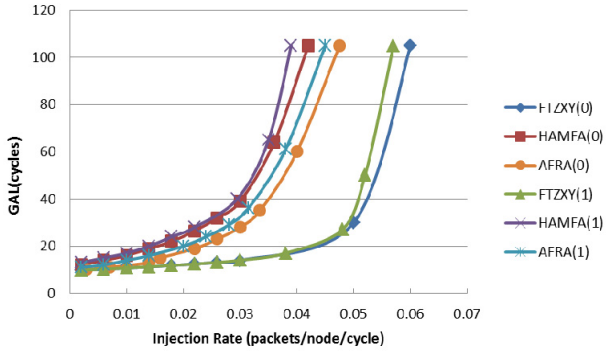
در همسایگی لینک خراب نیز این الگوریتم فاقد بن‌بست است چرا که چرخش‌های ممنوع‌شده (چرخش‌های شرق- جنوب و شمال- غرب) اجازه ایجاد وابستگی حلقوی در هنگام مسیریابی را نمی‌دهند.

## ٤-٢ بن‌بست سه‌بعدی

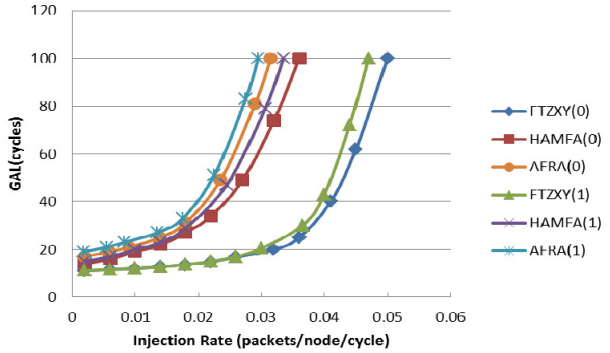
مسیریابی بین‌لایه‌ای در غیاب اشکال همانند الگوریتم ZXY عمل می‌کند که فاقد بن‌بست است، بنابراین اگر مقصد در لایه دیگر باشد، بسته ابتدا باید در بعد z حرکت کند و حق حرکت در ابعاد x و y را ندارد اما هنگامی که یک یا چند لینک عمودی خراب باشند، با توجه به ارسال بسته به سمت مقصد برحسب زوج یا فرد بودن شماره سطر یا ستون گره فعلی، بسته تنها به یک جهت (جهت شرق و یا غرب و در صورت رسیدن به لبه، جهت شمال یا جنوب) ارسال می‌شود و بدین ترتیب از ایجاد حلقه وابستگی جلوگیری خواهد شد. به عنوان مثال، همان طور که در شکل ۸ مشاهده می‌شود در یک صفحه عمودی وقتی لینک عمودی خراب باشد، اگر شماره y زوج باشد، بسته حق انحراف به سمت غرب را ندارد و این موضوع از ایجاد حلقه در صفحه عمودی جلوگیری می‌کند.

## ۵- نتایج ارزیابی

برای ارزیابی کارایی و قابلیت اطمینان الگوریتم مسیریابی پیشنهادی، از یک شبیه‌ساز توسعه داده شده بر پایه ابزار Noxim [۱۴] که مبتنی بر




شکل ۹: میانگین تأخیر بسته در شبکه ۴×۴×۴ تحت ترافیک یکنواخت.




شکل ۱۰: میانگین تأخیر بسته در شبکه ۴×۴×۴ زیرترافیک ترانهاده.

جدول ۲: تنظیمات پارامترهای ابزار شبیه‌سازی.

| پارامترها | تنظیمات |
|---|---|
| اندازه بافر ورودی | ۸ فلیت |
| طول بسته | ۸ فلیت |
| طول فلیت | ۳۲ بیت |
| زمان راه‌اندازی | ۱۰۰۰ سیکل |
| مدت شبیه‌سازی | ۵۰۰۰۰ سیکل |
| همبندی | توری سه‌بعدی ۴×۴×۴ و ۶×۶×۴ |

کدهای SystemC است، استفاده کرده و قابلیت مدیریت و تزریق اشکال را به آن افزودیم.

الگوریتم‌های مورد مقایسه با الگوریتم پیشنهادی، الگوریتم HAMFA و الگوریتم AFRA است که آنها را به همراه الگوریتم پیشنهادی با استفاده از کدهای SystemC پیاده‌سازی کردیم. الگوریتم‌های مسیریابی HAMFA و AFRA به این دلیل انتخاب شده‌اند که از روش‌های مشابه قبلی کارایی بالاتری دارند. الگوهای ترافیکی که در ارزیابی‌ها استفاده شده است، ترافیک‌های یکنواخت، ترانهاده و نقطه‌داغ است که بر دو ساختار توری سه‌بعدی ۴×۴×۴ و ۶×۶×۴ اعمال شده است.

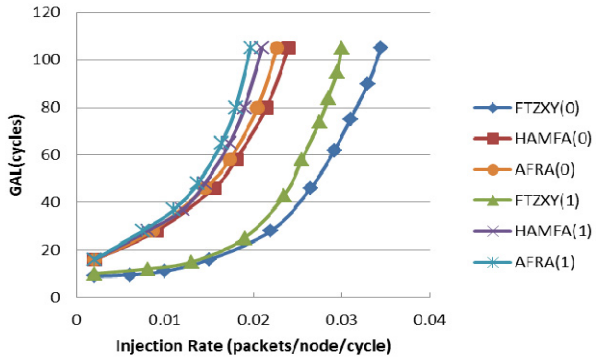
جدول ۲ تنظیمات عمومی پارامترهای ابزار شبیه‌سازی را نشان می‌دهد.

در شبیه‌سازی، هر تکرار به مدت ۵۰۰۰۰ سیکل اجرا شده و در هر بار تکرار، یک یا چند اشکال تصادفی به شبکه تزریق شده است.

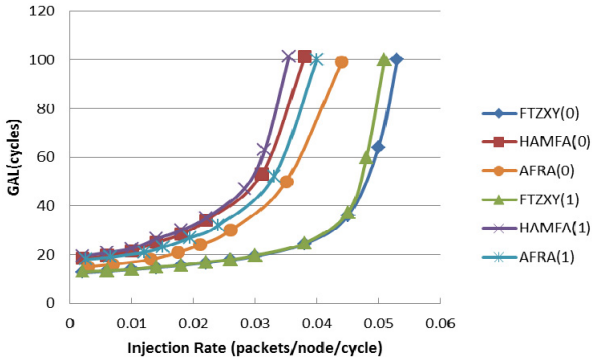
## ۵-۱ ارزیابی میانگین تأخیر بسته‌ها

میانگین تأخیر بسته یکی از مهم‌ترین پارامترهایی است که کارایی الگوریتم مسیریابی را مشخص می‌کند. روند افزایش میانگین تأخیر بسته زمانی که نرخ تزریق بسته را افزایش می‌دهیم به وسیله یک منحنی نشان داده می‌شود. همان طور که در شکل‌های ۹ تا ۱۴ مشاهده می‌شود،

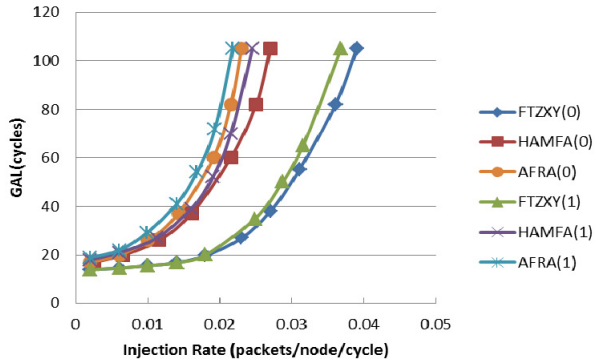
Archive of SID
www.SID.ir



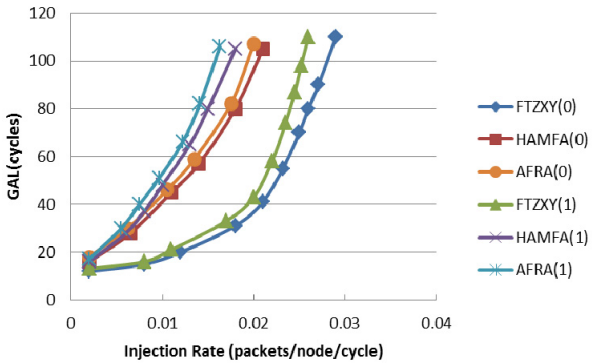

شکل ۱۱: میانگین تأخیر بسته در شبکه ۴×۴×۴ زیرترافیک نقطه‌داغ.



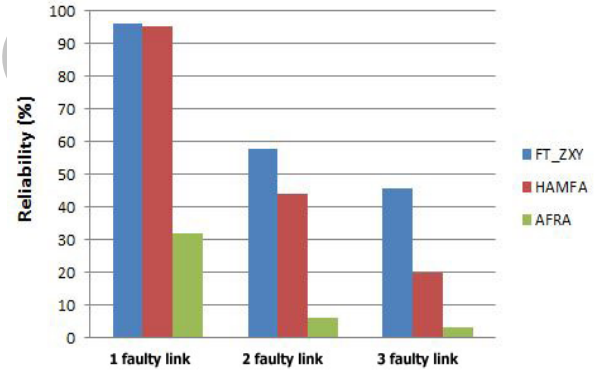

شکل ۱۲: میانگین تأخیر بسته در شبکه ۴×۶×۶ زیرترافیک یکنواخت.




شکل ۱۳: میانگین تأخیر بسته در شبکه ۴×۶×۶ زیرترافیک ترانهاده.




شکل ۱۴: میانگین تأخیر بسته در شبکه ۴×۶×۶ زیرترافیک نقطه‌داغ.




شکل ۱۵: مقایسه قابلیت اطمینان الگوریتم‌های مسیریابی FT_ZXY، HAMFA و AFRA.

الگوریتم‌های پیشنهادی، HAMFA و AFRA بر حسب میانگین تأخیر بسته، تحت الگوهای ترافیکی یکنواخت، ترانهاده و نقطه‌داغ با حجم ترافیکی نقطه داغ ۱۰٪، با تعداد اشکال صفر و یک برای شبکه‌های با اندازه ۴×۴×۴ و ۴×۶×۶ با یکدیگر مقایسه شده‌اند. نمودارهای به دست آمده با تزریق اشکال تصادفی به تعداد ۳۰۰ تا ۵۰۰ تکرار برای هر نقطه و میانگیری از آنها به دست آمده است و بنابراین این نمودارها اثر متوسط یک لینک خراب را بر روی کارایی شبکه نشان می‌دهند. همان طور که در نمودارها مشاهده می‌شود، اثر متوسط یک لینک خراب در شبکه ۴×۴×۴ بیشتر از شبکه ۴×۶×۶ است.

نمودارهای نشان داده شده، بهبودی الگوریتم مسیریابی پیشنهادی را نسبت به الگوریتم‌های HAMFA و AFRA از نظر میانگین تأخیر بسته نشان می‌دهد. الگوریتم FT_ZXY دارای منطق ساده‌ای است و سعی می‌کند کوتاه‌ترین مسیر به سمت مقصد را بیابد. در مقابل، الگوریتم‌های HAMFA و AFRA زمانی که اشکالی در مسیر مبدأ به مقصد ایجاد شود، مسیرهای طولانی‌تری را برای فرار از اشکال در نظر گرفته و به همین دلیل میانگین تأخیر در آنها افزایش می‌یابد.

## ۵-۲ ارزیابی قابلیت اطمینان

برای این که قابلیت اطمینان روش‌های FT_ZXY، HAMFA و AFRA اندازه‌گیری شود، تعداد لینک‌های خراب از یک تا سه به صورت تصادفی به شبکه اعمال شده‌اند اما این تزریق اشکال به گونه‌ای بوده که حداکثر یکی از اشکال‌ها در لینک‌های افقی رخ می‌دهد تا عملکرد روش پیشنهادی در برابر لینک‌های خراب عمودی که ضعیف‌تر هم هستند، بهتر نشان داده شود. یک شبکه غیر قابل اطمینان محسوب می‌شود اگر همه بسته‌های تولیدشده به مقصد نرسند. به عبارت دیگر اگر حتی یک بسته به مقصد نرسد شبکه را غیر قابل اطمینان به حساب می‌آوریم. نتیجه مقایسه‌ای که در شکل ۱۵ نشان داده شده است با استفاده از ۱۰۰۰۰ تکرار در شبکه توری ۴×۶×۶، زیرترافیک یکنواخت با نرخ تزریق ۰/۰۱ بسته در هر گره در هر پالس ساعت (معادل با ۰/۰۸ حداکثر بار ترافیکی) به دست آمده است.

همان طور که در شکل ۱۵ مشاهده می‌شود، الگوریتم پیشنهادی FT_ZXY قابلیت اطمینان بیشتری نسبت به HAMFA و AFRA دارد و هرچه تعداد اشکال بیشتر می‌شود این اختلاف افزایش می‌یابد.

## ۵-۳ ارزیابی مساحت روی تراشه و توان مصرفی

برای ارزیابی توان مصرفی و مساحت روی تراشه، الگوریتم‌های مسیریابی FT_ZXY، HAMFA و AFRA را به همراه الگوریتم پایه ZXY برای مش ۴×۴×۴ با استفاده از زبان توصیف سخت‌افزار VHDL پیاده‌سازی کردیم. برای سنتز از ابزار Synopsys Design Compiler با کتابخانه استاندارد ۹۰ STMICRO نانومتر (با فرکانس کاری ۱ گیگاهرتز و ولتاژ منبع تغذیه ۱/۲۰ ولت در دمای ۲۵ درجه سانتی‌گراد) استفاده کردیم. نتایج به دست آمده در جدول‌های ۳ و ۴ نشان داده شده است.

در الگوریتم AFRA هر گره باید وضعیت اشکال لینک‌های عمودی

Archive of SID
www.SID.ir

جدول ۳: میزان مساحت روی تراشه.

| مسیریابی | مساحت $(\mu m)^2$ |
|---|---|
| ZXY | ۵۲۱۴ |
| FT_ZXY | ۵۷۴۷ |
| HAMFA | ۷۷۵۸ |
| AFRA | ۸۲۲۰ |

جدول ۴: توان مصرفی.

| مسیریابی | توان مصرفی (uw) |
|---|---|
| ZXY | ۳۳۸ |
| FT_ZXY | ۳۹۱ |
| HAMFA | ۴۵۹ |
| AFRA | ۴۸۶ |

متناظر خود در لایه‌های مختلف و همه گره‌هایی که در سطر متناظر با این گره قرار دارند را بداند. این مسئله سربار اطلاعاتی و در نتیجه سربار سخت‌افزاری زیادی را تحمیل می‌کند. الگوریتم HAMFA نیز دارای پیچیدگی ساختاری زیادی است. نتایج ارزیابی نشان می‌دهد که الگوریتم HAMFA نسبت به FT_ZXY به ۳۵٪ و الگوریتم AFRA نسبت به FT_ZXY به ۴۳٪ مساحت بیشتر احتیاج دارد. علاوه بر این، الگوریتم‌های HAMFA و AFRA نسبت به الگوریتم FT_ZXY به ترتیب به ۱۷٪ و ۲۴٪ توان مصرفی بیشتر احتیاج دارند.

## ۶- نتیجه‌گیری

طراحی الگوریتم‌های مسیریابی با قابلیت تحمل‌پذیری اشکال در شبکه روی تراشه، یکی از موضوعاتی است که در سال‌های اخیر توجه زیادی به آن شده است. با پیشرفت فناوری ساخت مدارهای مجتمع و ظهور مدارهای مجتمع سه‌بعدی، شبکه روی تراشه سه‌بعدی مطرح گردیده و همچنین الگوریتم‌های مسیریابی تحمل‌پذیر اشکال در مورد این نوع مدارها مورد بحث و بررسی قرار گرفته است. نکته قابل توجه این است که به علت ساخت ضعیف لینک‌های عمودی و بالابودن احتمال رخداد اشکال در آنها، تحمل‌پذیری اشکال در این نوع از شبکه‌ها اهمیت ویژه‌ای دارد. در این مقاله، یک روش جدید برای مسیریابی در شبکه روی تراشه سه‌بعدی به نام FT_ZXY معرفی گردید که قابلیت تحمل اشکال‌های منفرد در لینک‌های افقی و اشکال‌های چندگانه در لینک‌های عمودی را دارا است. این روش مبتنی بر چرخش است به این معنی که با ممنوع‌کردن بعضی از چرخش‌ها از ایجاد بن‌بست جلوگیری می‌کند و لذا این روش به کانال مجازی احتیاج ندارد. علاوه بر این محدوده اطلاعات مورد نیاز آن برای مدیریت اشکال به اندازه یک گام است. این دو ویژگی، سربار سخت‌افزاری الگوریتم مسیریابی پیشنهادی را کاهش داده است. نکته دیگر این که الگوریتم پیشنهادی بر پایه یک الگوریتم دوبعدی گسترش داده شده است و به عبارت دیگر دارای دو قسمت مجزای درون‌لایه‌ای و بین‌لایه‌ای است. این کار پیچیدگی الگوریتم و سربار آن را نسبت به روش‌های قبلی کاهش داده و مقیاس‌پذیری را افزایش داده است. در نتیجه این روش توانسته است نسبت به روش‌های قبلی، هم کارایی و هم قابلیت اطمینان را بهبود دهد.

## مراجع


[1] A. Agarwal, C. Iskander, and R. Shankar, "Survey of network on chip (NoC) architectures and contributions," *J. Eng. Comput. Arch.*, vol. 3, no. 1, pp. 1-15, 2009.

[2] D. Wentzlaff, *et al*., "On-chip interconnection architecture of the tile processor," *IEEE Micro*, vol. 27, no. 5, pp. 15-31, Sep. 2007.

[3] A. Hemani, *et al*., "Network on chip: an architecture for billion transistor era," in *Proc. of the 18th IEEE NORCHIP Conf.*, 8 pp., Nov. 2000.

[4] S. Murali, *et al*., "Analysis of error recovery schemes for networks on chips," *IEEE Design & Test of Computers*, vol. 22, no. 5, pp. 434-442, Sep./Oct.. 2005.

[5] C. Grecu, L. Anghel, P. P. Pande, A. Ivanov, and R. Saleh, "Essential fault-tolerance metrics for NoC infrastructures," in *Proc. 13th IEEE Int. On-line Testing Symp.*, pp. 37-42, Crete, Greece, 8-11 Jul. 2007.

[6] C. Rusu, L. Anghel, and D. Avresky, "RILM: reconfigurable inter-layer routing mechanism for 3D multi-layer networks-on-chip," in *Proc. 16th. IEEE Int. On-Line Testing Symp.*, pp. 121-126, Corfu, Greece, 5-7 Jul. 2010.

[7] C. Feng, *et al*., "A low-overhead fault-aware deflection routing algorithm for 3D network-on-chip," in *Proc. IEEE Computer Society Annual Symp. on VLSI*, pp. 19-24, 4-6 Jul. 2011.

[8] A. M. Rahmani, *et al*., "Design and management of high-performance, reliable and thermal-aware 3D networks-on-chip," *IET Circuits, Devices & Systems*, vol. 6, no. 5, pp. 308-321, Sep. 2012.

[9] S. Akbari, et al, "AFRA: a low cost high performance reliable routing for 3D mesh NoCs," in *Proc. of 15th ACM/IEEE Design, Automation, and Test in Europe, DATE'12*, pp. 332-337, Dresden, Germany, 12-16 Mar. 2012.

[10] M. Ebrahimi, M. Daneshtalab, and J. Plosila, "Fault-tolerant routing algorithm for 3D NoC using hamiltonian path strategy," in *Proc. of 16th ACM/IEEE Design, Automation, and Test in Europe, DATE'13*, pp. 1601-1605, Grenoble, France, 18-22 Mar. 2013.

[11] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," in *Proc. of the 19th Annual Int. Symp. on Computer Architecture, ISCA'92*, pp. 278-287, 1992.

[12] C. Grecu, *et al*., "BIST for network-on-chip interconnect infrastructures," in *Proc. 24th IEEE VLSI Test Symp*, 6 pp., Berkeley, CA, USA, 30 Apr.- 4 May 2006.

[13] N. Dahir, T. Mak, R. Al-Dujaily, and A. Yakovlev, "Highly adaptive and deadlock-free routing for three-dimensional networks-on-chip," *IET Computers & Digital Techniques*, vol. 7, no. 6, pp. 255-263, Nov. 2013.

[14] Fazzino, F. Palesi, M. Patti, *2008 Noxim: Network-on-Chip Simulator*," URL:http://sourceforge.net/projects/noxim.



**مصطفی تقی‌زاده فیروزجائی** تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به ترتیب در سالهای ۱۳۸۹ و ۱۳۹۳ در دانشگاه صنعتی بابل به پایان رسانده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری، پردازش موازی و شبکه روی تراشه.

**مجتبی ولی‌نتاج** مدارک کارشناسی، کارشناسی ارشد و دکترای خود را در مهندسی کامپیوتر هر سه از دانشگاه تهران، به ترتیب در سال‌های ۱۳۷۹، ۱۳۸۱ و ۱۳۸۹ دریافت نمود. ایشان از سال ۱۳۸۹ به عنوان عضو هیات علمی در دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی نوشیروانی بابل در بابل مشغول به فعالیت است. زمینه‌های پژوهشی مورد علاقه ایشان شامل طراحی سیستم‌های قابل اطمینان، شبکه روی تراشه، حساب کامپیوتری، معماری کامپیوتر و سیستم‌های چندپردازنده‌ای است.

**مجتبی منصوری** در سال ۱۳۵۷ مدرک کارشناسی مهندسی سیستم‌های کامپیوتری خود را از دانشگاه میشیگان غربی در آمریکا و در سال ۱۳۶۱ مدرک کارشناسی ارشد علوم کامپیوتر خود را از دانشگاه ایالتی ویسکانسین-مدیسون در آمریکا دریافت نمود. نام‌برده از سال ۱۳۶۲ الی ۱۳۶۶ به عنوان کارشناس نرم‌افزار در شرکت Astronautics Corporation of America و از سال ۱۳۶۶ الی ۱۳۷۰ به عنوان کارشناس ارشد مهندسی نرم‌افزار در شرکت‌های General Electric Consulting Services و General Electric Medical Systems در آمریکا به کار مشغول بود. دکتر منصوری از سال ۱۳۷۱ در دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی بابل مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشکده می‌باشد. ایشان در سال ۱۳۹۱ موفق به اخذ درجه دکترا در مهندسی برق و کامپیوتر از دانشگاه صنعتی بابل گردید. زمینه‌های علمی مورد علاقه نام‌برده شامل موضوعاتی مانند پردازش زبان طبیعی، سیستم‌های پرس‌وجو، رایانش فوق‌سریع و سیستم‌های توزیعی می‌باشد.


Archive of SID

www.SID.ir