مجله محاسبات نرم، جلد ۹، شماره ۱ (بهار و تابستان ۱۳۹۹)، صفحه ۱۲۴ـ۱۴۵
https://dx.doi.org/10.22052/scj.2021.111571

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir

دانشگاه کاشان
University of Kashan

یک الگوریتم بسیار سریع برای شبیه‌سازی اِشکال تأخیر مسیر مدارهای دیجیتال بر اساس پیمایش موازی مسیر بحرانی ✧

احمد احترام[۱]، کارشناسی ارشد، حسین صباغیان بیدگلی[۲*]، استادیار، حسین قسوری[۳]، استادیار، مجید دلشاد[۱]، دانشیار، شاهین حسابی[۴]، دانشیار

[۱] دانشکده فنی و مهندسی، گروه مهندسی برق، واحد اصفهان (خوراسگان)، دانشگاه آزاد اسلامی، اصفهان، ایران.

[۲] دانشکده مهندسی برق و کامپیوتر، گروه مهندسی کامپیوتر، دانشگاه کاشان، کاشان، ایران.

[۳] دانشکده مهندسی برق و کامپیوتر، گروه مهندسی برق، واحد کاشان، دانشگاه آزاد اسلامی، کاشان، ایران.

[۴] دانشکده مهندسی کامپیوتر، گروه مهندسی برق، دانشگاه صنعتی شریف، تهران، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۲۰ فروردین ماه ۱۴۰۰
پذیرش ۳۱ اردیبهشت ماه ۱۴۰۰

کلمات کلیدی:
آزمون
اِشکال تأخیر مسیر
شبیه‌سازی اِشکال
رهگیری مسیر بحرانی
مسیر نیرومند
مسیر غیرنیرومند

چکیده

شبیه‌سازی اِشکال تأخیر مسیر روشی برای ارزیابی کیفیت آزمون است که در آن تعداد مسیرهای شناسایی شده توسط یک مجموعه آزمون مشخص می‌شود. زمان اجرای شبیه‌سازی اشکال تاخیر مسیر به تعداد کل مسیرهای یک مدار وابسته است. افزایش روزافزون بزرگی و پیچیدگی مدارات دیجیتال و رابطه نمایی تعداد مسیرها بر حسب تعداد دروازه‌های یک مدار، شبیه‌سازی اِشکال تأخیر مسیر را برای مدارات امروزی به عملیاتی زمان‌بر تبدیل کرده است. از این رو نیاز به وجود الگوریتم‌های سرعت بالاتر به شدت احساس می‌شود. روش‌های موجود شبیه‌سازی اِشکال تأخیر مسیر بخاطر زمان اجرای طولانی، عدم دقت و یا نیاز به سخت‌افزار خاص دچار مشکل هستند. در کار حاضر یک الگوریتم بسیار سریع برای شبیه‌سازی اِشکال تأخیر مسیر ارائه می‌شود که ضمن افزایش سرعت، دقت خود را نیز حفظ می‌کند و از طرفی برای اجرا به سخت‌افزار خاصی نیاز ندارد. این روش بطور همزمان از چند تکنیک مختلف برای افزایش سرعت استفاده می‌کند. برخی از این تکنیک‌ها مانند رهگیری مسیر بحرانی (جهت کوچک کردن فضای جستجو)، ساده‌سازی شروط انتشار تاخیر مسیر (برای کاهش حجم محاسبات) و ایجاد چک لیست آرایه‌ای (به منظور حذف عملیات مقایسه و جستجو هنگام ادغام لیست مسیرهای شناسایی شده) نوآوری محسوب می‌شوند و بکارگیری آنها در کنار تکنیک‌های شناخته شده مانند اندیس‌گذاری مسیرها (برای جلوگیری از استخراج کامل مسیرها) و موازات ۳۲ بیتی (جهت اعمال ۳۲ بردار آزمون همزمان) باعث افزایش قابل توجه سرعت شده است. روش پیشنهادی بر روی تعدادی از مدارهای محک ISCAS85 و ITC99 آزمایش شده و نتایج ترکیب تکنیک های مختلف با یکدیگر و با تعدادی از کارهای گذشته مقایسه شده است. نتایج بدست آمده، تأثیر تکنیک‌های بکار رفته و بهبود حدود ۱۸۶ برابری نسبت به کارهای دیگران را نشان می‌دهد.

© ۱۴۰۰ ـ مجلهٔ محاسبات نرم، کلیه حقوق محفوظ است.

نحوه ارجاع به مقاله: احترام، احمد، صباغیان بیدگلی، حسین، قسوری، حسین، دلشاد، مجید، حسابی، شاهین، «یک الگوریتم بسیار سریع برای شبیه‌سازی اِشکال تأخیر مسیر مدارهای دیجیتال بر اساس پیمایش موازی مسیر بحرانی»، مجله محاسبات نرم، جلد ۹، شماره ۱، ص۱۲۴ـ۱۴۵، بهار و تابستان ۱۳۹۹.

۱. مقدمه

امروزه با پیشرفت فناوری ساخت مدارات دیجیتال همزمان با افزایش پیچیدگی محصولات درخواستی شاهد فشردگی زیاد اندازه تراشه‌ها و در نتیجه افزایش احتمال بروز اِشکال در ساخت این محصولات هستیم. به‌طورکلی انواع اِشکال در مدارات دیجیتال شامل اِشکال‌های دائمی[1] و غیردائمی (گذرا[2] و ادواری[3]) است. [۱]. در این میان اِشکال گذرا به طور معمول بر اثر نویزهای محیطی، نوسانات تغذیه و پرتوهای کیهانی ایجاد می‌شود. درحالی‌که اِشکال دائمی ناشی از نقص در تولید، سالخوردگی، تداخلات داخلی یا عوامل بیرونی است [۲][۳]. به این ترتیب محصولات تولید شده پس از ساخت و/یا در حین اجرا نیازمند اعمال آزمون‌هایی برای اطمینان از صحت عملکرد هستند. در آزمون اِشکال تأخیر که نوعی اِشکال دائمی محسوب می‌شود در هر مرحله از آزمون دو بردار متوالی استفاده می‌شود، تا میزان تأخیر انتشار تغییرات، از ورودی تا خروجی مدار بررسی شود. تولید بردارهای آزمون برای هر یک از انواع اِشکال بر اساس مدلی که برای آن اِشکال در نظرگرفته شده انجام می‌شود. در تولید آزمون، هدف یافتن یک مجموعه از بردارهای آزمون با بیشترین پوشش اِشکال است. برای ارزیابی کیفیت بردارهای آزمون تولید شده از شبیه‌سازی اِشکال استفاده می‌شود تا میزان پوشش اِشکال اندازه‌گیری شود. در شبیه‌سازی اِشکال لازم است به ازای هر بردار آزمون تک‌تک اِشکال‌های مدار بررسی شود. بنابراین شبیه‌سازی اِشکال یک عملیات زمان‌بر بوده و زمان آن وابسته به تعداد بردارهای آزمون و طول فهرست اِشکال‌ها است. از آنجا که مدارات امروزی، بدلیل بزرگتر بودن، دارای فهرست بلندتری از اِشکال‌ها و همچنین نیازمند مجموعه آزمون بزرگتری هستند و در نتیجه زمان شبیه‌سازی اِشکال نیز برای آنها به شدت افزایش یافته است. ازاین‌رو نیازمند الگوریتم‌های سریع‌تر برای شبیه‌سازی آزمون هستیم. اهمیت موضوع وقتی‌که بدانیم شبیه‌سازی اِشکال بخشی از عملیات تولید آزمون نیز هست بیشتر احساس می‌شود. یک الگوریتم سریع برای شبیه‌سازی آزمون زمان تولید آزمون را کاهش داده و می‌تواند در کاهش زمان عرضه به بازار نیز نقش مؤثری داشته باشد [۴].

اِشکال تأخیر یکی از انواع اِشکال است که به شکل‌های مختلفی مدل می‌شود. مدل‌های اِشکال تأخیر گذر و اِشکال تأخیر مسیر از مهم‌ترین مدل‌های اِشکال تأخیر محسوب می‌شود. در مدل اِشکال تأخیر گذر، اِشکال بصورت نقطه‌ای و در خروجی دروازه‌ها و بصورت کندی در گذر از صفر به یک یا گذر از یک به صفر در نظر گرفته می‌شود. بنابراین تعداد اِشکال‌ها محدود به تعداد سیم‌های اتصالی مدار است. در مدل اِشکال تأخیر مسیر، اِشکال‌ها روی مسیرها فرض می‌شود. یک مسیر شامل زنجیره‌ای از دروازه‌ها و سیم‌هاست که از یک سیم ورودی شروع شده و به یک سیم خروجی ختم می‌شود. در این مدل، هر مسیر می‌تواند شامل اِشکال تأخیر باشند. از آنجا که تعداد مسیر با افزایش تعداد دروازه‌های منطقی مدار بطور نمایی افزایش پیدا می‌کند، عملیات شبیه‌سازی آزمون و به تبع آن تولید آزمون برای مدارات بزرگ یک چالش جدی محسوب می‌شود [۵].

کارهای مختلفی با هدف کاهش زمان شبیه‌سازی اِشکال تأخیر مسیر صورت گرفته است که در بخش ۳ بطور مفصل‌تر به آن پرداخته خواهد شد. بطور کلی این کارها را می‌توان به سه دسته تقسیم کرد. دسته اول که روش‌های شمارشی نامیده می‌شوند، پوشش تک‌تک مسیرهای مدار را به ازای هر آزمون مورد بررسی قرار می‌دهند. مشکل اساسی این روش‌ها زمان طولانی اجرای آنهاست. دسته دوم روش‌هایی هستند که از شمارش تک‌تک مسیرها پرهیز می‌کنند. هر چند این الگوریتم‌ها از سرعت بالایی برخوردار هستند ولی در مقابل قادر نیستند مقدار دقیق پوشش اِشکال را بدست آورند. بعبارت دیگر میزان پوشش

✧ نوع مقاله: پژوهشی

* نویسنده مسئول
پست(های) الکترونیک: a.ehteram@iaukashan.ac.ir (احترام)
hsabaghianb@kashanu.ac.ir (صباغیان بیدگلی)
h.ghasvari@iaukashan.ac.ir (قسوری)
delshad@khuisf.ac.ir (دلشاد)
hessabi@sharif.edu (حسابی)

[1] Permanent Fault
[2] Transient Fault
[3] Intermittent Fault

اِشکال را بطور تقریبی بدست می‌آورند. دسته سوم که برای حفظ دقت تمام مسیرها را در نظر می‌گیرند، برای افزایش سرعت از پردازنده‌های گرافیکی به عنوان شتاب‌دهنده استفاده می‌کنند. در این روش‌ها نیز نیاز به وجود سخت‌افزار خاص مانند GPU یک نقطه ضعف محسوب می‌شود. به عبارت دیگر این الگوریتم‌ها روی هر سیستمی قابل اجرا نیستند.

در این مقاله یک الگوریتم بسیار سریع برای شبیه‌سازی اِشکال تأخیر مسیر ارائه می‌شود که ضمن افزایش سرعت دقت خود را هم حفظ می‌کند. الگوریتم پیشنهادی به سخت‌افزار خاصی نیاز ندارد و روی هر سیستم با پردازنده معمولی کار می‌کند. این روش بطور همزمان از چند تکنیک مختلف استفاده می‌کند که از جمله می‌توان به ساده‌سازی شروط انتشار مسیر، بکارگیری رهگیری مسیر بحرانی [۶] توسعه یافته، به منظور کاهش مسیر از طریق حذف مسیرهای غیر قابل شناسایی، موازات ۳۲ بیتی، اندیس‌گذاری مسیرها برای جلوگیری از استخراج کامل مسیرها و ایجاد چک لیست آرایه‌ای جهت حذف مرحله جستجو اشاره کرد. برخی از تکنیک‌های بکار رفته پیش از این در کارهای مرتبط استفاده شده ولی برخی دیگر نوآوری محسوب می‌شوند. بطور کلی موارد نوآوری را می‌توان بشکل زیر خلاصه کرد.

۱. ساده‌سازی شروط انتشار تأخیر مسیر با ترکیب شرایط مسیرهای نیرومند و غیرنیرومند و شمارش مجموع آنها و یافتن تعداد مسیرهای غیرنیرومند با کاهش تعداد مسیرهای نیرومند از آن مجموع
۲. توسعه روش رهگیری مسیر بحرانی برای شبیه‌سازی اشکال تأخیر و ارائه فرمول‌های مربوطه (توجه شود که روش رهگیری مسیر بحرانی روشی شناخته شده است که از ابتدا برای شبیه‌سازی اشکال چسبندگی ارائه شده و در اینجا توسعه داده شده است).
۳. ایجاد چک لیست آرایه‌ای بر اساس اندیس مسیرها برای حذف مرحله جستجو در ادغام لیست مسیرهای تازه شناسایی شده در لیست اصلی مسیرهای شناسایی شده.
۴. ترکیب تکنیک‌های پیشنهادی بندهای ۱ تا ۳ در کنار دو تکنیک مرسوم یعنی موازات ۳۲ بیتی و اندیس‌گذاری مسیرها بصورت ترکیبی که منجر به نتایج قابل توجه در افزایش سرعت شده است.

روش پیشنهادی بر روی تعدادی از مدارهای محک ISCAS85 و ITC99 آزمایش شده و نتایج ترکیب تکنیک‌های مختلف با یکدیگر و با تعدادی از کارهای گذشته مقایسه شده است. نتایج، تأثیر تکنیک‌های بکار رفته و بهبود قابل ملاحظه زمان اجرا را نسبت به کارهای دیگران نشان می‌دهد.

در بخش ۲ به مفاهیم اولیه و مطالب زمینه‌ای راجع به اِشکال تأخیر مسیر و شبیه‌سازی آن می‌پردازد. بخش ۳ شامل مروری بر کارهای مرتبط در زمینه شبیه‌سازی اِشکال تأخیر مسیر است. الگوریتم پیشنهادی در بخش ۴ شرح داده می‌شود. بخش ۵ شامل نتایج آزمایش‌ها و مقایسه آن‌ها با روش‌های دیگر است. در بخش ۶ نیز جمع‌بندی و کارهای آتی آورده شده است.

۲. مفاهیم و تعاریف اولیه

وجود تأخیر در مدارات دیجیتال، به این معنی که اثر تغییرات ورودی با مقداری تأخیر در خروجی ظاهر شود امری عادی است ولی مقدار این تأخیر نباید از حد معمول بیشتر باشد. اگر میزان تأخیر از حد معینی بیشتر شود به آن اِشکال تأخیر گفته می‌شود. برای اطمینان از عدم وجود اِشکال تأخیر در یک مدار دیجیتال از آزمون اِشکال تأخیر استفاده می‌شود.

۲,۱. آزمون اِشکال تأخیر

هر آزمون اِشکال تأخیر، شامل یک جفت بردار ورودی به‌صورت V_1V_2 است که به طور متوالی به ورودی‌های مدار اعمال می‌شود. بردار اول (V_1) جهت اعمال مقدار اولیه به مدار و بردار دوم (V_2) برای ایجاد تغییر در ورودی‌ها و بررسی تأخیر انتشار این تغییرات به خروجی است. درصورتی‌که تأخیر انتشار از حد معینی (که معمولاً دوره تناوب ساعت سیستم است) بیشتر شود اِشکال تأخیر تشخیص داده می‌شود.

۲,۲. مدل اِشکال

بطور کلی برای تولید و ارزیابی آزمون از مدل اِشکال استفاده

می‌شود. هر نوع اِشکال یک یا چند مدل مخصوص به خود دارد که بر اساس آن تولید و ارزیابی آزمون انجام می‌شود. برای اِشکال تأخیر از مدل‌های اِشکال متفاوتی استفاده می‌شود. در این میان مدل اِشکال تأخیر گذر[1] و مدل اِشکال تأخیر مسیر[2] کاربرد بیشتری دارند که در ادامه شرح داده می‌شوند.

۲,۳. مدل اِشکال تأخیر گذر

مدل اِشکال تأخیر گذر برای مدل کردن تأخیرهای طولانی استفاده می‌شود. در این مدل اشکال، هر یک از سیگنال (سیم)های مدار می‌تواند دچار اِشکال تأخیر گردد. اگر تغییر مقدار (یعنی لبه بالارونده یا لبه پایین رونده) آن سیگنال در محدوده زمانی معین به خروجی نرسد آن سیگنال دارای اِشکال تأخیر خواهد بود. به این ترتیب در این نوع مدل اشکال به تعداد دو برابر سیگنال (سیم)های مدار، اشکال وجود خواهد داشت. مزیت اصلی مدل اِشکال تأخیر گذر پیچیدگی پایین آن است بدلیل اینکه تعداد اِشکال‌های مدار به‌صورت خطی با تعداد دروازه‌های مدار رابطه دارد [۷].

۲,۴. مدل اِشکال تأخیر مسیر

در این مدل، یک اِشکال از تجمیع تأخیرهای کوچک روی مسیرها بوجود می‌آیند. بنابراین این مسیرها هستند که دچار اِشکال می‌شوند. منظور از مسیر یک توالی از دروازه‌ها و سیگنال (سیم)های متصل کننده آن‌ها است که از یک ورودی شروع و به یک خروجی ختم می‌شود. جهت تغییر (یعنی لبه بالارونده یا لبه پایین رونده بودن) سیگنال در ورودی نیز دو اشکال متفاوت ایجاد می‌کند. تعداد مسیرهای فیزیکی مدار و در نتیجه تعداد اِشکال‌های تأخیر مسیر می‌تواند به‌صورت نمایی برحسب تعداد دروازه‌های مدار افزایش یابد. بنابراین عملیات شبیه‌سازی اِشکال تأخیر مسیر از پیچیدگی بالایی برخوردار است [۸][۹].

۲,۵. شبیه‌سازی اِشکال تأخیر مسیر

هدف از شبیه‌سازی اِشکال تأخیر مسیر، مشخص کردن مسیرهایی است که در صورت داشتن اِشکال تأخیر با مجموعه آزمون داده شده قابل شناسایی هستند. گرچه در آزمون واقعی بردارهای آزمون با فاصله زمانی اعمال شده و شکل موج‌های خروجی نیز در طول زمان مشاهده می‌شوند ولی در اغلب روش‌های شبیه‌سازی اِشکال تأخیر، عملیات شبیه‌سازی بصورت منطقی انجام می‌شود و در آن عامل زمان در نظر گرفته نمی‌شود. برای این کار هر دو بردار V_1 و V_2 بطور همزمان به ورودی‌های مدار اعمال شده و مقدار منطقی تمام سیگنال‌های داخلی مدار به‌ازای دو بردار داده شده بطور همزمان محاسبه می‌گردد. در شبیه‌سازی اِشکال تأخیر مسیر، سعی می‌شود با توجه به مقادیر منطقی سیگنال‌های داخلی مدار که از انتشار مقادیر دو بردار آزمون بوجود آمده، تشخیص داده شود که کدام مسیرها، در صورت ابتلا به اِشکال تأخیر، قابل تشخیص خواهند بود. از آنجا که تأخیرهای واقعی دروازه‌ها در نظر گرفته نمی‌شود و مقدار این تأخیرها در نمونه‌های مختلف ساخته شده متفاوت است و از طرفی مقدار تأخیر مسیرهای مختلف می‌توانند بر تشخیص اِشکال یکدیگر تأثیر بگذارند، اظهار نظر قطعی در مورد آشکارسازی همه مسیرها امکان‌پذیر نخواهد بود و فقط در شرایط خاصی می‌توان با اطمینان در این زمینه اظهار نظر نمود. بر این اساس مسیرهای پوشش داده شده توسط یک آزمون به سه دسته‌ی مسیرهای حساس[3]، مسیرهای نیرومند[4] و مسیرهای غیرنیرومند[5] دسته‌بندی می‌شوند. دو دسته اول مستقل از تأخیر سایر مسیرها قابل تشخیص هستند. قابلیت تشخیص دسته سوم وابسته به تأخیر سایر مسیرها است. در ادامه به شرح بیشتر هر یک از انواع یاد شده می‌پردازیم. پیش از آن برای فهم بهتر مطلب، ذکر چند تعریف ضروری است.

[1] Transition delay fault (TDF)
[2] Path Delay Fault (PDF)
[3] Sensitive paths
[4] Robust paths
[5] Non-robust paths

۲،۵،۱. سیگنال‌های روی مسیر و خارج از مسیر

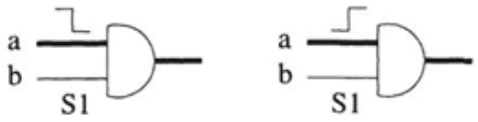
یک مسیر معین که از ورودی تا خروجی مدار امتداد دارد، در داخل مدار از میان تعدادی دروازه‌های منطقی عبور می‌کند. برای هر یک از این دروازه‌ها، یکی از ورودی‌ها روی آن مسیر و بقیه خارج از آن مسیر هستند.

۲،۵،۲. مقدار کنترلی و غیرکنترلی

مقدار صفر برای دروازه AND یک مقدار کنترلی است زیرا مقدار صفر در یکی از ورودهای آن (مستقل از مقدار سایر ورودی‌ها) خروجی را صفر می‌کند. بطور مشابه مقدار ۱ برای دروازه OR یک مقدار کنترلی است زیرا مقدار ۱ در یکی از ورودی‌های دروازه OR (مستقل از مقدار سایر ورودی‌ها) خروجی را یک می‌کند. بنابراین مقدار کنترلی برای دروازه AND برابر صفر و برای OR برابر ۱ است. در مقابل مقدار غیرکنترلی برای دروازه AND برابر ۱ و برای OR برابر صفر است. مقدار کنترلی و غیرکنترلی برای دروازه NOR مانند OR و برای NAND مانند AND است. بطور کلی بدون در نظر گرفتن نوع دروازه مقدار کنترلی را با نماد cv و مقدار غیرکنترلی را ncv نمایش می‌دهیم.

۲،۶. مسیر حساس

درصورتی‌که مقدار هر دو بردار V_1 و V_2 برای تمام سیگنال‌های خارج از مسیر، مقدار غیرکنترلی ایجاد کند، هر نوع تغییر ورودی از طریق مسیر مذکور به خروجی منتقل می‌شود و در صورت وجود اِشکال تأخیر در مسیر مذکور این اِشکال در خروجی مسیر تشخیص داده می‌شود. به یک چنین مسیری مسیر حساس گفته می‌شود. ویژگی مسیر حساس این است که در طول مسیر هیچ تغییری در مقدار سیگنال‌های خارج از مسیر اتفاق نمی‌افتد. اِشکال تأخیر یک مسیر حساس مستقل از تأخیر سایر مسیرهای مدار حتماً تشخیص داده می‌شود. اما تعداد کمی از مسیرهای مدار تحت چنین شرایطی تشخیص داده می‌شوند. شکل (۱) شرایط انتشار لبه‌ی بالارونده (شکل (۱)–الف) و پایین‌رونده (شکل (۱)–ب) را در مسیر حساس برای یک دروازه AND دو ورودی نشان می‌دهد. در این شکل سیگنال‌های روی مسیر بصورت ضخیم‌تر رسم شده و مقدار $S1$ به معنی مقدار ۱ برای هر دو بردار V_1 و V_2 است.



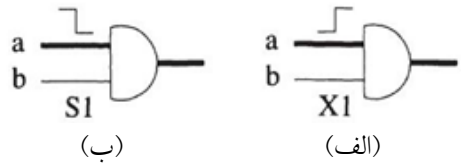
(ب) (الف)

شکل (۱): شرایط انتشار مسیر حساس برای دروازه AND دو ورودی (الف) برای لبه بالارونده (ب) برای لبه پایین رونده

۲،۷. مسیر نیرومند

مسیر نیرومند مانند مسیر حساس است با این تفاوت که در مسیر نیرومند برای انتشار تغییر از مقدار کنترلی به غیرکنترلی (cv → ncv) شرط ساده‌تری وجود دارد به این صورت که برای سیگنال‌های خارج از مسیر مقدار بردار اول (V_1) می‌تواند مقدار کنترلی هم باشد. به عبارت دیگر مقدار V_1 مهم نیست. شکل(۲) شرایط انتشار در یک دروازه AND برای دو لبه بالارونده (شکل (۲)–الف) و پایین‌رونده (شکل (۲)–ب) یعنی مقدار کنترلی به غیرکنترلی (cv → ncv) و مقدار غیرکنترلی به کنترلی (ncv → cv) را نشان می‌دهد. همان‌طور که در این شکل نشان داده شده است برای انتشار لبه بالارونده، مقدار بردار V_1 روی ورودی‌های خارج از مسیر، می‌تواند به‌صورت اختیاری ۰ یا ۱ باشد، به همین دلیل با X نشان داده شده است. ویژگی اصلی مسیر نیرومند این است که اِشکال تأخیر آن مستقل از تأخیر سایر مسیرها، قطعاً تشخیص داده می‌شود و تأخیر در سایر مسیرها روی آشکار شدن آن بوسیله آزمون جاری اثری ندارد [۱۰]. اگرچه به ازای هر جفت بردار آزمون تعداد مسیرهایی که بصورت نیرومند تشخیص داده می‌شوند نسبت به مسیرهای حساس بیشتر است ولی باز هم سهم کمی از مسیرهای مدار تحت شرایط نیرومند تشخیص داده می‌شوند. ضمناً با مقایسه شرایط تشخیص مسیر حساس و مسیر نیرومند می‌توان دریافت که مجموعه مسیرهای نیرومند آشکار شده به ازای یک آزمون (یا مجموعه آزمون) شامل مسیرهای حساس آشکار شده نیز

هست. به عبارت دیگر مجموعه مسیرهای حساس زیرمجموعه مسیرهای نیرومند است.



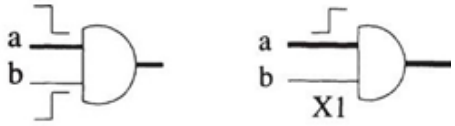
(ب) (الف)

شکل (۲): شرایط انتشار مسیر نیرومند برای دروازه AND دو ورودی (الف) برای لبه بالارونده (ب) برای لبه پایین رونده

۲،۸. مسیر غیرنیرومند

ویژگی اصلی مسیر غیرنیرومند این است که تأخیر در سایر مسیرها ممکن است باعث آشکار نشدن آن بوسیله آزمون جاری شود، درحالی‌که اگر تأخیر در سایر مسیرها وجود نداشته باشد آشکار می‌شود. یک آزمون غیرنیرومند وقتی بهتر است که گذر سیگنال‌های خارج از مسیر، شانس کمتری برای جلوگیری از عبور گذرهای روی مسیری داشته باشند و پیدا کردن چنین آزمون‌های غیرنیرومندی نیازمند دانستن اطلاعاتی در مورد تأخیرهای مدار است. با توجه به اینکه در اغلب روش‌های شبیه‌سازی اِشکال این اطلاعات در اختیار نیست و اغلب، شبیه‌سازی بصورت منطقی انجام می‌شود، در زمان شبیه‌سازی نمی‌توان بطور قطع در مورد آشکار شدن مسیرهای غیرنیرومند توسط آزمون جاری اظهار نظر نمود. در حقیقت علت نام‌گذاری مسیرهای غیرنیرومند هم به همین عدم قطعیت در آشکار شدن برمی‌گردد.

شکل (۳) شرایط انتشار مسیر غیرنیرومند برای یک دروازه AND را نشان می‌دهد. بطور کلی شرط انتشار مسیر غیرنیرومند گذر از مقدار کنترلی به غیرکنترلی ($cv \rightarrow ncv$) مانند مسیر نیرومند است (شکل (۳)-الف)، ولی برای انتشار گذر از مقدار غیرکنترلی به کنترلی ($ncv \rightarrow cv$) شرط متفاوتی وجود دارد (شکل (۳)-ب)، به این صورت که روی سیگنال‌های خارج از مسیر باید تغییر از مقدار کنترلی به مقدار غیرکنترلی (یعنی $cv \rightarrow ncv$) وجود داشته باشد. در شکل (۳)-ب اگر گذر خارج از مسیر بعد از گذر روی مسیر رخ دهد از انتشار تأخیر به خروجی جلوگیری می‌شود ولی اگر قبل از گذر روی مسیر رخ دهد اثر اِشکال تأخیر به خروجی منتقل شده و شناسایی می‌شود. بنابراین در این حالت تشخیص اِشکال تأخیر مسیر به تأخیر سیگنال‌های خارج از مسیر بستگی دارد.



(ب) (الف)

شکل (۳): شرایط انتشار مسیر غیر نیرومند برای دروازه AND دو ورودی (الف) برای لبه بالارونده (ب) برای لبه پایین رونده

تعداد مسیرهایی که با شرایط غیرنیرومند تشخیص داده می‌شوند نسبت به تعداد مسیرهای نیرومند خیلی بیشتر است ولی نسبت به تعداد کل مسیرهای یک مدار بخصوص برای مدارات خیلی بزرگ خیلی کم است. بالا بودن تعداد کل مسیرهای یک مدار و پایین بودن پوشش اِشکال تأخیر مسیر چالش‌هایی را برای عملیات مختلف آزمون بخصوص تولید آزمون و شبیه‌سازی آزمون ایجاد نموده است. در زمینه تولید آزمون تلاش‌های زیادی برای بالا بردن کیفیت آزمون با هدف افزایش پوشش اِشکال تأخیر مسیر انجام شده است. شبیه‌سازی آزمون چه به عنوان بخشی از عملیات تولید آزمون چه بطور مستقل نیازمند الگوریتم‌های سریع و دقیق است. در بخش بعدی کارهای مختلفی که در این زمینه صورت گرفته مرور می‌شود.

۳. مرور کارهای مرتبط

با توجه به اینکه تعداد مسیرهای یک مدار با افزایش تعداد دروازه‌ها بصورت نمایی افزایش می‌یابد شمارش دقیق تک‌تک مسیرها در شبیه‌سازی اِشکال برای مدارات بزرگ بسیار زمان‌بر می‌شود. به همین دلیل برخی روش‌ها با استفاده از تکنیک‌هایی از شمارش همه مسیرها پرهیز نموده‌اند. بنابراین روش‌های شبیه‌سازی اِشکال مسیر را می‌توان به دو دسته شمارشی و غیرشمارشی تقسیم کرد.

اولین بار در سال ۱۹۸۵روشی شمارشی برای شبیه‌سازی اِشکال تأخیر مسیر برای مدارهای ترکیبی ارائه شد که از یک جبر شش

مقداری برای شبیه‌سازی اِشکال تأخیر مسیر نیرومند استفاده می‌کرد [۱۱]. شبیه‌سازی بصورت منطقی و بدون در نظر گرفتن مقدار تأخیر دروازه‌ها در طول مسیر انجام می‌شد. ازاین‌رو قادر بود اِشکال تأخیر با هر اندازه‌ای را مدل نماید. این روش برای کاهش زمان شبیه‌سازی از پردازش موازی الگوهای آزمون استفاده می‌کرد. سپس در سال ۱۹۸۹ روشی مبتنی بر جبر چهار مقداری برای شبیه‌سازی اِشکال تأخیر مسیر نیرومند و غیرنیرومند ارائه شد. این روش در کنار جبر ۴ مقداری از اعمال موازی الگوهای آزمون و همچنین یک ساختمان داده مخصوص با نام درخت مسیر نیز استفاده می‌کرد. این ساختمان داده برای صرفه‌جویی در حافظه بخش‌های مشترک مسیرها را فقط یک‌بار ذخیره می‌کرد [۱۲]. در سال ۱۹۹۶ یک رویکرد مبتنی بر قانون برای تشخیص مسیرهای نیرومند و غیرنیرومند ارائه شد [۱۳]. این روش بجای جبر چند مقدار از منطق دو مقداری استفاده می‌کرد. در این روش همچنین قوانینی برای انتشار گلیچ[1] در نظر گرفته شده، که از آن برای تشخیص مسیرهای غیرنیرومند استفاده شده است. در سال ۲۰۰۰ الگوریتمی برای شبیه‌سازی اِشکال تفاضلی در مدارات ترتیبی ارائه شد. این الگوریتم برای صرفه‌جویی در حافظه، تفاضل بین حالت‌های مدار ترتیبی را در نظر می‌گرفت [۱۴]. این الگوریتم بصورت موازی پیاده‌سازی شده بود امکان اجرا بصورت بصورت بدبینانه و خوش‌بینانه را داشت. بنابراین نتایج شبیه‌سازی را بصورت حد بالا و پایین ارائه می‌کرد. در سال ۲۰۰۸ یک الگوریتم ژنتیک برای تولید آزمون ارائه شد [۱۵]. در این روش تعداد ورودی و خروجی و تعداد دروازه‌ها و تعداد سطوح مدار مشخص شده و برای بررسی مسیرهای پوشش داده شده از یک ساختمان داده درختی استفاده می‌شد. آزمون‌ها با استفاده از الگوریتم ژنتیک تولید و تعداد مسیرهای نیرومند و غیرنیرومند با استفاده از آن ساختمان داده مشخص می‌شد.

دسته‌ی دیگر از روش‌ها که روش‌های غیرشمارشی نامیده می‌شوند به نحوی از شمارش تمام مسیرها پرهیز می‌کنند. در سال ۱۹۹۲ روشی بر این اساس ارائه شد که در آن برای هر آزمون ورودی، مبنای تشخیص مسیر جدید، وجود یک سیم که قبلاً پوشش داده نشده بود [۱۶]. در این روش یک سیم با مقدار بالارونده از همان سیم با مقدار پایین‌رونده متفاوت در نظر گرفته می‌شد. تعداد مسیرهایی که اینگونه تشخیص داده می‌شدند نسبت به کل تعداد مسیر که با یک پیمایش از خروجی به ورودی شمارش می‌شدند یک تخمین بدبینانه از پوشش اِشکال ارائه می‌کرد که بطور معمول کمتر از مقدار واقعی پوشش اِشکال بود. مثال دیگر از روش‌های غیرشمارشی استفاده از تکنیک کاهش مسیر است. به عنوان مثال در روشی که در [۱۷] ارائه شده فقط مسیرهای طولانی در نظر گرفته شده است. این روش بدلیل پرهیز از شمارش تمام مسیرها تخمینی از پوشش اِشکال را ارائه داده ولی در عوض درجه پیچیدگی را از نمایی به چندجمله‌ای کاهش می‌دهد. این روش می‌تواند با افزایش درجه چندجمله‌ای تقریب بهتری از پوشش اِشکال ارائه دهد. روش آماری برای تخمین پوشش اِشکال روش دیگری است [۱۸] که در آن تعدادی آزمون تصادفی به مدار اعمال می‌شود و احتمال تشخیص هر یک از مسیرها بر اساس مقادیر قابلیت کنترل و قابلیت مشاهده‌ی سیم‌های مدار محاسبه می‌شود و مقدار پوشش اِشکال بر اساس احتمال تشخیص مسیرها بصورت تخمینی محاسبه می‌گردد.

در سال‌های اخیر استفاده از پردازنده‌های گرافیکی (GPU) برای افزایش سرعت مرسوم شده است. یک GPU با داشتن تعداد زیادی پردازنده که امکان پردازش موازی دارند می‌تواند به‌عنوان شتاب‌دهنده در کنار یک پردازنده میزبان استفاده شود. برخی کارهای انجام شده در زمینه شبیه‌سازی اِشکال مسیر از این روش برای غلبه بر پیچیدگی در مدارات بزرگ استفاده می‌کنند [۱۹] [۲۰]. نتایج تجربی کارآیی این روش‌ها را برای شناسایی تمام مسیرهای قابل آزمون نشان می‌دهد. در برخی از کارهایی که با این رویکرد انجام شده [۲۱]، شبیه‌سازی با در نظر گرفتن زمان و تأخیر دروازه‌ها انجام می‌شود که از این جهت نیز از شبیه‌سازهای منطقی پیشی می‌گیرد.

در میان روش‌هایی که با سه رویکرد یاد شده ارائه شده است دسته اول مشکل طولانی شدن زمان اجرا و دسته دوم مشکل

[1] Glitch

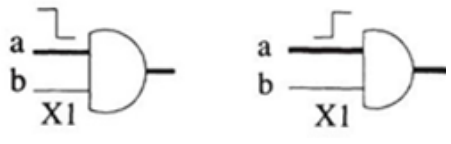
تقریبی بودن جواب و دسته سوم مشکل وابستگی به سخت‌افزار دارند. در کار حاضر روشی مبتنی بر رویکرد اول ارائه می‌شود که سعی کرده با بکارگیری چند تکنیک بطور همزمان به کاهش زمان اجرا بپردازد. این روش با توجه به اینکه تمامی مسیرها را در نظر می‌گیرد مشکل تقریبی بودن، که در روش‌های غیرشمارشی بود را ندارد. همچنین مشکل وابستگی به GPU را نیز که مربوط به دسته سوم روش‌ها بود ندارد. برخی از تکنیک‌های بکار رفته در این روش عبارتند از: ساده‌سازی شروط انتشار مسیر، اندیس‌گذاری ترتیبی مسیرها و استفاده از چک لیست آرایه‌ای محدود. این روش برای مدارات متوسط به پایین بخاطر اندیس‌گذاری یکتای مسیرها بطور دقیق عمل می‌کند. در مقابل برای مدارات بزرگ با بیش از ۲۳۲ مسیر بخاطر احتمال تکراری بودن اندیس مسیرها بصورت تقریبی عمل می‌کند. نتایج تجربی روی مدارات محک ISCAS85 و ITC99 افزایش چند صد برابری سرعت را نسبت به آخرین کارهای گزارش شده نشان می‌دهد.

۴. روش پیشنهادی

در این بخش روال پیشنهادی برای شبیه‌سازی اِشکال تأخیر مسیر (PDF) ارائه می‌شود. این روال یک روال کلی است که برای انواع مسیر (حساس، نیرومند و غیرنیرومند) از رویکرد رهگیری مسیر بحرانی استفاده می‌کند. رهگیری مسیر بحرانی یک روش شناخته شده در شبیه‌سازی اِشکال SAF و TDF است که با اعمال هر آزمون ابتدا تمام مسیرهای بحرانی مدار را مشخص نموده و سپس بدون تزریق اِشکال در مدار بطور همزمان تمام اِشکال‌های روی مسیرهای بحرانی مدار را شناسایی کند. تنها چالش این روش افزایش پیچیدگی آن در مقابله با پدیده‌ی خود بازدارندگی است که پیش‌تر راه‌حلی برای آن در [۶] ارائه گردید. در این مقاله روش رهگیری مسیر بحرانی برای شناسایی مسیرها توسعه داده می‌شود. به این ترتیب که فرمول‌های جدیدی برای پیمایش معکوس مسیر بحرانی ارائه می‌شود. این فرمول‌ها در سه نسخه ارائه می‌گردد که برای شناسایی مسیرهای حساس، نیرومند و غیرنیرومند استفاده می‌شوند. برای شناسایی هر یک از انواع مسیر یاد شده مراحل شبیه‌سازی اِشکال بر اساس روال کلی پیشنهادی و فرمول‌های پیشنهادی مخصوص خود انجام می‌شود. در کنار روال پیشنهادی از تکنیک‌هایی برای افزایش سرعت شبیه‌سازی اِشکال استفاده شده است که در ادامه این بخش شرح داده می‌شود. یکی از این تکنیک‌ها ساده‌سازی شروط انتشار مسیر است که در زیر بخش بعد به آن می‌پردازیم.

۴،۱. ساده‌سازی شروط انتشار مسیر

شرایط انتشار نیرومند و غیرنیرومند به‌ترتیب در شکل‌های (۲) و (۳) نشان داده شد. با ادغام این شرایط، شروط انتشار ساده‌تری حاصل می‌شود که در شکل (۴) نشان داده شده است. مجموعه‌ی مسیرهایی که این شروط را دارند اجتماع دو مجموعه مسیرهای نیرومند و مسیرهای غیرنیرومند خواهند بود، که ما آن‌ها را مسیرهای بحرانی می‌نامیم. علت این نام‌گذاری هم به استفاده از روش رهگیری مسیر بحرانی (CPT) برمی‌گردد. ویژگی این مجموعه از مسیرها سادگی شروط انتشار آن‌هاست. همان‌طور که در شکل (۴) دیده می‌شود شرط انتشار برای هر دو تغییر $(ncv \rightarrow cv)$ و $(cv \rightarrow ncv)$، غیرکنترلی بودن مقدار V_2 برای سیگنال‌های خارج از مسیر است که شرط بسیار ساده‌ای است و چک کردن آن در طول مسیر می‌تواند به راحتی و با سرعت انجام شود. این موضوع موجب افزایش سرعت شبیه‌سازی اِشکال می‌شود. اما لازم به یادآوری است که مسیرهای شمارش شده با این روش که ما آنها را مسیرهای بحرانی می‌نامیم، شامل اجتماع مسیرهای نیرومند و غیرنیرومند هستند.



(ب) (الف)

شکل (۴): شرایط انتشار مسیر بحرانی برای دروازه AND دو ورودی (الف) برای لبه بالارونده (ب) برای لبه پایین رونده

بطور کلی شروط انتشار برای مسیرهای حساس، نیرومند و

بحرانی ساده است ولی شروط انتشار برای مسیرهای غیرنیرومند از پیچیدگی بالایی برخوردار است. همان‌طور که در شرح انواع مسیر ذکر شد به ازای یک آزمون (با یک مجموعه آزمون) داده شده، مسیرهای حساس زیرمجموعه مسیرهای نیرومند هستند. از طرف دیگر مجموعه مسیرهای بحرانی از اجتماع مسیرهای نیرومند و غیرنیرومند تشکیل می‌شود. شکل (۵) روابط بین این مجموعه مسیرها را بصورت گرافیکی نشان می‌دهد. با توجه به این شکل می‌توان مجموعه مسیرهای غیرنیرومند را از تفاضل مجموعه مسیرهای بحرانی و مجموعه مسیرهای نیرومند بدست آورد. این موضوع باعث می‌شود بتوان بجای رهگیری مسیرهای غیرنیرومند که دارای شروط پیچیده‌ای است رهگیری مسیرهای بحرانی انجام شود که دارای شروط انتشار بسیار ساده‌تری است.

شکل (۵): یک توصیف نمایشی برای مجموعه مسیرهای حساس، نیرومند و غیرنیرومند و اجتماع آنها که مسیرهای بحرانی نامو اشتراک و اجتماع آن‌ها

۴،۲. روال کلی پیشنهادی

شکل (۶) روال کلی شبیه‌سازی اِشکال تـأخیر مسـیر را نشـان می‌دهد این روال پیشنهادی شبکه اتصالی مدار[1] تحت آزمون و مجموعه آزمون را بعنوان ورودی دریافت و عملیات شبیه‌سازی آزمون را برای هر آزمون در پنج مرحله شامل مراحل زیر انجام می‌دهد.

مرحله ۱: اعمال یک آزمون به ورودی‌های مدل مدار سالم

مرحله ۲: شبیه سازی منطقی روی به جلو بر روی مـدل مـدار سالم

مرحله ۳: پیمایش روبه عقب برای تعیین گره‌های بحرانی مدار

مرحله ۴: پیمایش روبه جلو به منظور اندیس‌گذاری و شمارش مسیرهای بحرانی

مرحله ۵: ادغام مسیرهای شناسـایی‌شـده‌ی آزمـون جـاری در لیست مسیرهای شناسایی شده تاکنون

در مرحلـه ۱ یـک جفـت بـردار آزمـون بصـورت V_1V_2 بـه ورودی‌های اصلی مـدار اعمـال مـی‌شـود. در مرحلـه ۲ یـک شبیه‌سازی منطقی روبه‌جلو انجام می‌شود تا مقادیر منطقی کلیه سیگنال‌های مدار بر اساس مقـادیر منطقـی بردارهـای آزمـون ورودی V_1V_2 و عملکرد دروازه‌های منطقی مشخص شـود. در جدول (۱) فرمول کلی خروجی تک‌تک دروازه‌های منطقی به صورت توابعی از ورودی‌های بیان شده است. برای هر دروازه تابع منطقی V_1 و V_2 بطور جداگانه بیان شده است.

شکل (۶): روال کلی پیشنهادی برای شبیه‌سازی اِشکال تأخیر مسیر

در مرحله ۳ با استفاده از توابعی که در جدول (۱) ارائـه شـده است محاسبات مسیر بحرانی رو به عقب بـا پیمـایش تمـامی مسیرها از خروجی به سمت ورودی انجام می‌شود. طـی ایـن مرحله یک مقدار منطقی (۰ یا ۱) برای بحرانی بودن کـه بـا C نشان داده شده است به هر گره مدار اختصاص می‌یابد. مقدار C

[1] Circuit netlist

بر اساس مقدار منطقی V_1V_2ی ورودی‌ها، مقدار منطقی V_1V_2ی خروجی‌ها و توابع عملکردی دروازه‌های منطقی محاسبه می‌شود. همان‌طور که در جدول مشاهده می‌شود مقدار C برای مسیرهای حساس، نیرومند و بحرانی بترتیب با نام‌های $C_{sp}(s)$، $C_{rp}(s)$و $C_{cp}(s)$ آورده شده است.

در مرحله ۴ بر اساس مقادیر بحرانی (C) که در مرحله‌ی قبل محاسبه شد ردیابی مسیر بحرانی به سمت جلو و همزمان شمارش تعداد مسیر از ورودی به خروجی انجام می‌شود. همزمان با شمارش مسیر از یک روش اندیس‌گذاری نیز برای تشخیص و جلوگیری از شمارش مجدد مسیرهای تکراری در بردارهای آزمون مختلف استفاده می‌شود. روش اندیس‌گذاری پیشنهادی قادر است یک اندیس مخصوص برای هر مسیر تولید کند. در بعضی روش‌های اندیس‌گذاری احتمال اختصاص اندیس مشترک به دو مسیر متفاوت وجود دارد ولی اگر این احتمال خیلی کم باشد، خطایی کوچکی را در شبیه‌سازی اِشکال ایجاد می‌کند که قابل چشم‌پوشی است. اندیس هر مسیر در طول حرکت از ورودی به خروجی ساخته می‌شود و در پایان این مرحله به ازای هر مسیر آشکار شده یک اندیس مسیر بدست خواهد آمد. توجه شود که در کارحاضر دو نوع اندیس‌گذاری استفاده شده است که در زیر بخش‌های بعدی شرح داده خواهد شد.

در مرحله ۵ تک‌تک مسیرهای آشکار شده با استفاده از اندیس مسیر مربوطه، در یک چک لیست آرایه‌ای، شامل مسیرهای مدار علامت زده می‌شوند.

۴،۳. توابع پیشنهادی برای رهگیری انواع مسیر

جدول (۱) توابع انتقالی پیمایش روبه‌جلو و رو به عقب که به ترتیب در مرحله ۲ و ۳ شبیه‌سازی PDF استفاده شده است را نشان می‌دهد. هر دو تابع برای دروازه‌های منطقی مختلف و انشعاب[1] ارائه شده است. سیگنال خروجی برای هر دروازه‌ی n ورودی Y_n و سیگنال‌های ورودی برای آن X_i نامیده می‌شود بطوری که $i = 1, 2, \dots, n$ ورودی‌های مختلف را نشان می‌دهد.

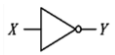
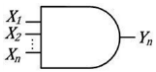
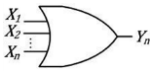
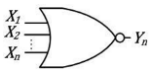
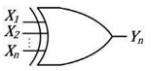
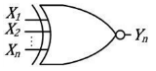
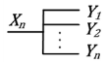
سیگنال‌های ورودی برای یک انشعاب با n خروجی با X_n و سیگنال‌های خروجی برای آن Y_i نامیده می‌شود بطوری که $i = 1, 2, \dots, n$ خروجی‌های مختلف را نشان می‌دهد. مقدار منطقی و حالت بحرانی سیگنال s به ترتیب با $V(s)$ و $C(s)$ نشان داده می‌شود. حالت بحرانی سیگنال‌ها برای انواع مسیر بحرانی، نیرومند و حساس به ترتیب با $C_{sp}(s)$، $C_{rp}(s)$ و $C_{cp}(s)$ نشان داده شده است. بعد از اجرای مراحل پیمایش روبه‌جلو و رو به عقب برای یک جفت بردار آزمون، در مرحله۴ یک حرکت روبه‌جلو روی مسیر بحرانی به همراه شمارش مسیر انجام شده و همزمان یک اندیس خاص برای هر مسیر بحرانی تولید می‌شود. قبل از ارائه جزئیات اندیس‌گذاری مسیر، تکنیک موازی‌سازی که در پیاده‌سازی الگوریتم به کار گرفته شده است در زیر بخش بعدی مورد بررسی قرار می‌گیرد. سپس به شرح روش اندیس‌گذاری مسیر و همچنین چک لیست آرایه‌ای پرداخته خواهد شد.

۴،۴. موازات ۳۲ بیتی در رهگیری انواع مسیر

هدف اصلی الگوریتم ارائه شده وقتی محقق می‌شود که به شکل موازی پیاده‌سازی شود. کاهش پیچیدگی شروط انتشار مسیر راه را برای موازی‌سازی بیتی رهگیری مسیر هموارتر می‌کند. در این بخش نحوه پیاده‌سازی نسخه‌ی ۳۲ بیتی الگوریتم پیشنهادی ارائه می‌شود. در این نسخه در هر مرحله بجای یک جفت، ۳۲ جفت بردار آزمون به‌صورت همزمان به ورودی مدار اعمال می‌شود و تمام مسیرهای آشکار شده توسط این ۳۲ آزمون همزمان پردازش می‌شوند. پیاده‌سازی باید طوری انجام شود که مسیرهایی که بوسیله‌ی بیش از یک بردار آزمون تشخیص داده می‌شود تنها یک‌بار شمرده شود. این کار برای مسیرهای مشترک بین ۳۲ جفت برداری که در یک مرحله اعمال می‌شوند در داخل هر مرحله انجام می‌شود. چالش بزرگتر اشتراک مسیر بین مراحل مختلف است که برای مقابله با آن از اندیس‌گذاری مسیرها استفاده می‌شود. که در ادامه شرح داده خواهد شد.

[1] Fanout

جدول (۱): توابع روبه جلو و روبه عقب دروازه‌های منطقی مختلف برای رهگیری مسیرهای بحرانی، حساس و نیرومند

Gate	Function	Logic Value	Criticality of Critical path (C_{cp}), Criticality of sensitive path (C_{sp}) and Criticality of robust path (C_{rp})
	$Y = NOT(X)$	$V_1(Y) = 1 - V_1(X)$ $V_2(Y) = 1 - V_2(X)$	$C_{cp}(X) = C_{cp}(Y)$, $C_{sp}(X) = C_{sp}(Y)$, $C_{rp}(X) = C_{rp}(Y)$
	$Y_n = AND_{i=1}^{n}(X_i)$	$V_1(Y_n) = \prod_{i=1}^{n} V_1(X_i)$ $V_2(Y_n) = \prod_{i=1}^{n} V_2(X_i)$	$C_{cp}(X_k) = C_{cp}(Y_n) \prod_{i\neq k}^{n=1} V_2(X_i)$ $C_{sp}(X_k) = C_{sp}(Y_n) \left[\prod_{i\neq k}^{n=1} V_1(X_i)\right] \left[\prod_{i\neq k}^{n=1} V_2(X_i)\right]$ $C_{rp}(X_k) = C_{rp}(Y_n) \left[\prod_{i\neq k}^{n=1} V_2(X_i)\right] \left(\left[\prod_{i\neq k}^{n=1} V_1(X_i)\right] V_1(X_k)[1 - V_2(X_k)] + [1 - V_1(X_k)]V_2(X_k)\right)$ $k = 1, 2, \dots, n$
	$Y_n = NAND_{i=1}^{n}(X_i)$	$V_1(Y_n) = 1 - \prod_{i=1}^{n} V_1(X_i)$ $V_2(Y_n) = 1 - \prod_{i=1}^{n} V_2(X_i)$	
	$Y_n = OR_{i=1}^{n}(X_i)$	$V_1(Y_n) = 1 - \prod_{i=1}^{n}[1 - V_1(X_i)]$ $V_2(Y_n) = 1 - \prod_{i=1}^{n}[1 - V_2(X_i)]$	$C_{cp}(X_k) = C_{cp}(Y_n) \prod_{i\neq k}^{n=1}[1 - V_2(X_i)]$ $C_{sp}(X_k) = C_{sp}(Y_n) \left[\prod_{i\neq k}^{n=1}[1 - V_1(X_i)]\right] \left[\prod_{i\neq k}^{n=1}[1 - V_2(X_i)]\right]$ $C_{rp}(X_k) = C_{rp}(Y_n) \left[\prod_{i\neq k}^{n=1}[1 - V_2(X_i)]\right] \left(\left[\prod_{i\neq k}^{n=1}[1 - V_1(X_i)]\right] [1 - V_1(X_k)]V_2(X_k) + V_1(X_k)[1 - V_2(X_k)]\right)$ $k = 1, 2, \dots, n$
	$Y_n = NOR_{i=1}^{n}(X_i)$	$V_1(Y_n) = \prod_{i=1}^{n}[1 - V_1(X_i)]$ $V_2(Y_n) = \prod_{i=1}^{n}[1 - V_2(X_i)]$	
	$Y_n = XOR_{i=1}^{n}(X_i)$	$XOR_{i=1}^{n}(X_i) = XOR_{i=1}^{n-1}(X_i) \oplus X_n = Y_{n-1} \oplus X_n$ $V_1(Y_n) = [V_1(Y_{n-1})[1 - V_1(X_n)] + [1 - V_1(Y_{n-1})]V_1(X_n)]$ $V_2(Y_n) = [V_2(Y_{n-1})[1 - V_2(X_n)] + [1 - V_2(Y_{n-1})]V_2(X_n)]$	$C_{cp}(X_k) = C_{cp}(Y_n)$ $C_{sp}(X_k) = C_{sp}(Y_n)$ $C_{rp}(X_k) = C_{rp}(Y_n)$ $k = 1, 2, \dots, n$
	$Y_n = XNOR_{i=1}^{n}(X_i)$	$XOR_{i=1}^{n}(X_i) = XOR_{i=1}^{n-1}(X_i) \oplus X_n = Y_{n-1} \oplus X_n$ $V_1(Y_n) = 1 - [V_1(Y_{n-1})[1 - V_1(X_n)] + [1 - V_1(Y_{n-1})]V_1(X_n)]$ $V_2(Y_n) = 1 - [V_2(Y_{n-1})[1 - V_2(X_n)] + [1 - V_2(Y_{n-1})]V_2(X_n)]$	
	n branch Fanout $Y_i = X_n$ $i = 1, 2, \dots, n$	$V_1(Y_i) = V_1(X_n)$ $V_2(Y_i) = V_2(X_n)$ $i = 1, 2, \dots, n$	$C_{cp}(X_n) = 1 - \prod_{i=1}^{n}[1 - C_{cp}(Y_i)]$ $C_{sp}(X_n) = 1 - \prod_{i=1}^{n}[1 - C_{sp}(Y_i)]$ $C_{rp}(X_n) = 1 - \prod_{i=1}^{n}[1 - C_{rp}(Y_i)]$

برای فرموله کردن نسخه موازی شبیه‌سازی تأخیر مسیر تمام پارامترهای تعریف شده در جدول (۱) از جمله V_1، V_2 و C به ۳۲ بیت گسترش داده می‌شوند به طوری که هر بیت از آن‌ها متناظر با یکی از ۳۲ آزمون ورودی است. یک مسیر بحرانی مسیری است که حداقل در یکی از بیت‌های C به صورت سرتاسری از ورودی تا خروجی مقدار ۱ داشته باشد. برای بررسی این موضوع برای هر مسیر از یک نقاب به نام C_{mask} استفاده می‌شود. مقدار C_{mask} حاصل AND پارامتر C تمام سیگنال‌های روی مسیر است. در صورتی که C_{mask} یک مسیر حداقل دارای یک بیت یک باشد آن مسیر بحرانی خواهد بود. به بیان ریاضی برای هر مسیر، که از ورودی اصلی w_1 شروع و به خروجی اصلی w_n خاتمه می‌یابد، C_{mask} توسط معادله (۱) و (۲) به ترتیب برای تغییر از صفر به یک به نام C^r_{mask} و برای تغییر از یک به صفر C^f_{mask} محاسبه می‌شود.

$$C^r_{mask}(wn) = [\overline{v1(w1)}v2(w1)]\ \mathrm{AND}_{i=1}^{n}\big(c(w_i)\big) \quad (۱)$$

$$C^f_{mask}(wn) = [v2(w1)\overline{v1(w1)}]\ \mathrm{AND}_{i=1}^{n}\big(c(w_i)\big) \quad (۲)$$

مقدار C_{mask} برای هر مسیر به طور مجزا مرحله به مرحله و موقع حرکت از ورودی به سمت خروجی (در مرحله ۴ الگوریتم) محاسبه می‌شود. مقدار C_{mask} برای تک‌تک سیگنال‌های روی مسیر به صورت مجزا ذخیره می‌شود. به این ترتیب با توجه به فرمول (۱) و (۲) مقدار C_{mask} خروجی یک دروازه روی یک مسیر با ضرب C_{mask} ورودی در Cی خروجی آن بدست می‌آید. این پیمایش روی مسیرها تا مادامی که نتیجه C_{mask} غیر صفر باشد ادامه پیدا می‌کند. توجه شود که اگر بیش از یک ورودی از یک دروازه روی مسیر (های متفاوت) قرار داشته باشند خروجی آن دروازه بین آن مسیرها مشترک است. در این صورت ممکن است مقدار C_{mask} تمام ورودی‌های یک دروازه به خروجی منتقل شود. به این ترتیب در خروجی هر دروازه شاهد افزایش تعداد C_{mask}های مسیرها خواهیم بود. در این‌صورت در ادامه مسیر عبور تک‌تک آن‌ها از دروازه‌های بعدی یکی‌یکی بررسی خواهد شد و برای هر یک از آن‌ها احتمال عبور و عدم عبور وجود دارد. این کار تا رسیدن به خروجی‌های اصلی ادامه پیدا می‌کند. تعداد C_{mask}های منتقل شده به هر خروجی بیانگر تعداد مسیرهای آشکار شده‌ای است که به آن خروجی منتهی می‌شوند. در نهایت حاصل جمع کل تعداد C_{mask}ها در خروجی به‌عنوان تعداد کل مسیرهای آشکار شده در نظر گرفته می‌شود.

به این ترتیب با اعمال هرجفت آزمون ۳۲ بیتی (یا ۳۲ جفت آزمون بطور همزمان)، یک مجموعه مسیر آشکار می‌شود. یک چالش مهمی که در اینجا مطرح می‌شود این است که ممکن است مجموعه مسیرهایی که بوسیله یک آزمون ۳۲ بیتی آشکار می‌شود با مجموعه مسیرهایی که بوسیله آزمون ۳۲ بیتی بعدی آشکار می‌شود هم‌پوشانی داشته باشند. در این صورت جمع کردن تعداد مسیرهای آشکار شده بوسیله آزمون ۳۲ بیتی مختلف بدون درنظر گرفتن همپوشانی بین آن‌ها صحیح نیست. راه‌حل اولیه استخراج کامل مجموعه مسیرهای تشخیص داده شده برای هر آزمون ۳۲ بیتی و ادغام این مجموعه‌ها و حذف مسیرهای تکراری و شمارش تعداد اعضای مجموعه نهایی است. این روش نیاز به زمان و حافظه زیادی دارد. راه‌حل دیگر اندیس‌گذاری مسیرها و استخراج مجموعه اندیس مسیرهای آشکار شده برای هر آزمون ۳۲ بیتی و مقایسه و ادغام آن‌ها است که در ادامه شرح داده خواهد شد. روش مقایسه اندیس‌ها بخاطر پرهیز از ذخیره کل سیگنال‌های روی مسیر نیاز به حافظه و زمان کمتری دارد. با این حال عملیات جستجو و مقایسه در این روش هم برای مدارات بزرگ به زمان و حافظه زیاد نیاز دارد. راه‌حل پیشنهادی ما به خدمت گرفتن یک چک لیست آرایه‌ای است که اندیس خانه‌های آن همان اندیس مسیرهاست. این تکنیک باعث حذف عملیات مقایسه و ادغام مجموعه‌های اندیس مسیر و افزایش سرعت الگوریتم پیشنهادی می‌شود. در ادامه دو روش اندیس‌گذاری و دو روش مقایسه و ادغام مجموعه مسیر را شرح می‌دهیم. از طرفی در آزمایش‌ها نیز سعی شده نتایج این روش‌ها و برخی ترکیبات مختلف آن‌ها باهم مقایسه شود.

۴،۵. استخراج کامل مسیرها

در این روش با اعمال هر آزمون ۳۲ بیتی تمام مسیرهای آشکار شده توسط آن استخراج می‌شود. هر مسیر شامل یک توالی را سیگنال‌ها از ورودی تا خروجی است. با اعمال آزمون ۳۲ بیتی بعدی مجموعه مسیرهای آشکار شده توسط آن نیز استخراج شده و با مجموعه اول ادغام می‌شود. برای جلوگیری از تکرار مسیرهایی که در بیش از یک آزمون ۳۲ بیت آشکار می‌شوند نیازمند مقایسه دو بدوی تک‌تک مسیرهای دو مجموعه هستیم. این کار نیازمند زمان و حافظه زیادی است. برای بهبود روش از روش اندیس‌گذاری مسیرها و استفاده از اندیس بجای کل مسیر استفاده می‌کنیم. این روش در ادامه شرح داده می‌شود.

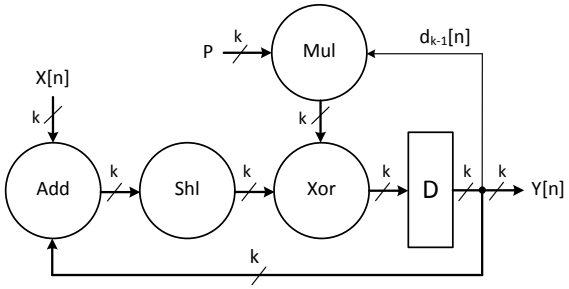
۴،۶. اندیس‌گذاری مسیر

ایده‌آل این است که در اندیس‌گذاری، به هر مسیر یک اندیس اختصاصی نسبت داده شود، ولی بسته به روش بگار گرفته شده برای اندیس‌گذاری ممکن است احتمال اختصاص اندیس مشترک وجود داشته باشند. در این صورت اگر این دو مسیر بوسیله دو آزمون ۳۲ بیتی متفاوت آشکار شوند به دلیل اندیس مشترک یک‌بار شمارش خواهند شد و این موضوع باعث تقریب در شبیه‌سازی اِشکال شده و میزان پوشش مسیر اعلام شده تقریبی خواهد بود. البته اگر خطای همپوشانی در اندیس‌گذاری ناچیز باشد می‌توان از خطای شبیه‌سازی صرفنظر کرد. در ادامه دو روش اندیس‌گذاری متفاوت معرفی می‌شود.

۴،۶،۱. اندیس‌گذاری مبتنی بر امضا

در شکل (۷) نمونه‌ای از اندیس‌گذاری مبتنی بر امضا با استفاده از یک گراف جریان داده[1] (DFG) نمایش داده شده است. این DFG در حقیقت عملکرد یک ثبات امضای چند ورودی[2] ($MISR$) را توصیف می‌کند که یک توالی از n عدد صحیح k بیتی که با $X[i]$ نشان داده می‌شود را (که در آن $i = 1, 2, \dots, n$ است) دریافت کرده و رشته دیگری از اعداد صحیح $Y[i]$ را (که در آن $i = 1, 2, \dots, n$ است) تولید می‌کند. ورودی P نیز یک عدد k بیتی ثابت است که نشان‌دهنده چندجمله‌ای ساخت $MISR$ است. DFG شامل ۴ تا عملگر Add، Shl، Xor و Mul می‌باشد که به ترتیب نشان‌دهنده یک جمع کننده k بیتی، یک ثبات انتقالی k بیتی، یک XOR با عرض k بیت و یک ضرب کننده ۱ بیت ضرب در k بیت است.

مستطیل D نشان‌دهنده یک حافظه k بیتی است. $MISR$ با یک مقدار اولیه $D[0]$ شروع به کار می‌کند. باارزش‌ترین بیت D که با $d_{k-1}[n]$ نشان داده می‌شود برای ضرب در P استفاده می‌شود. واضح است که بر اساس مقدار $d_{k-1}[n]$ که ممکن است ۰ یا ۱ باشد، نتیجه ۰ یا P خواهد بود. یک MISR ایده‌آل برای هر رشته ورودی $X[i]$ و مقادیر داده‌شده‌ی P، k و $D[0]$ یک امضا خاص (بدون همپوشانی) $Y[n]$ تولید می‌کند. در این مقاله از یک MISR با عرض ۳۲ بیت ($k = 32$) و از یک چندجمله‌ای $P = 0x08010400$ استفاده شده و به ترتیب دو مقدار مختلف (مکمل) $D_r[0] = 0xA3F75D19$ و $D_f[0] = 0x5C08A2E6$ برای هر مسیر فیزیکی با لبه بالارونده و لبه پائین در ورودی اصلی در نظر گرفته شده است. در این روش برای بدست آوردن یک اندیس مخصوص برای هر مسیر شماره‌ی، سیگنال‌های روی مسیر را بجای $X[i]$ به $MISR$ داده و از آخرین مقدار خروجی یعنی $Y[n]$ (که درواقع امضای مسیر است) به عنوان برچسب مسیر استفاده می‌شود.



شکل (۷): نمونه‌ای از اندیس‌گذاری مبتنی بر امضا با استفاده از یک گراف جریان داده

برای مثال فرض کنید مسیری شامل سیگنال‌های $[w_2, w_7, w_8, w_9, w_{13}, w_{15}, w_{17}]$ است که بترتیب از چپ به

[1] Data Flow Graph
[2] Multiple-Input Signature Register

راست قرار گرفته باشند را در نظر بگیرید این مسیر برای اعمال به $MISR$ بصورت $X[1:7] = [\,2,7,8,9,13,15,17\,]$ نمایش داده می‌شود. پس از اعمال آن به ورودی $MISR$ برچسب مسیر $Y[7] = 0x8796$ حاصل خواهد شد.

۴،۶،۲. اندیس‌گذاری ترتیبی

در این روش مسیرها بصورت ترتیبی اندیس‌گذاری می‌شود. این کار در دو مرحله انجام می‌شود. ابتدا برای هر گره (سیم) مدار دو متغیر بنام‌های $PathCount$ و PathIndex اختصاص می‌یابد. در مرحله اول با حرکت از خروجی به سمت ورودی عملیات شمارش مسیر انجام می‌شود و تمام $PathCount$ ها مقدار می‌گیرند و در مرحله دوم در حرکت از ورودی به سمت خروجی عملیات مقداردهی به $PathIndex$ تمام سیم‌های مدار انجام می‌شود. پس از این مرحله اندیس هر مسیر، برابر حاصل جمع $PathIndex$ تمام سیم‌های روی آن مسیر خواهد بود. این دو مرحله در ادامه به طور دقیق‌تر شرح داده می‌شود.

الف) شمارش مسیر با پیمایش معکوس: به $PathCount$ تمام خروجی‌ها مقدار یک داده و پیمایش معکوس به سمت ورودی‌ها آغاز می‌شود. با عبور از هر دروازه، مقدار $PathCount$ خروجی در $PathCount$ تمام ورودی‌ها کپی می‌شود ولی با عبور از هر انشعاب (FanOut) مقدار $PathCount$ تمام خروجی‌های آن باهم جمع شده و به $PathCount$ ورودی انتساب داده می‌شود. به این ترتیب در پایان پیمایش مقداری $PathCount$ هر یک از ورودی‌های مدار تعداد کل مسیرهای فیزیکی که از آن ورودی نشأت گرفته و به خروجی می‌رسد را مشخص می‌کند.

ب) اندیس‌گذاری مسیر با پیمایش مستقیم: در این مرحله مقدار $PathIndex$ تمام سیگنال‌های مدار تعیین می‌شود. پیمایش از ورودی‌های مدار شروع می‌شود و تا خروجی‌های مدار ادامه پیدا می‌کند. عملیات پیمایش برای دروازه‌های منطقی ساده است و مقدار PathIndex برای خروجی تمام دروازه‌ها صفر تعیین می‌شود. محاسبه اصلی برای ورودی‌های مدار و خروجی‌های انشعاب(FanOut)ها انجام می‌شود. ثابت بودن ترتیب در پیمایش ورودی‌های مدار و خروجی‌های انشعاب اهمیت دارد. هرچند انتخاب نوع ترتیب مهم نیست و می‌توان بعنوان مثال ورودی‌های مدار یا خروجی‌های هر انشعاب را به همان ترتیبی که در ساختمان داده‌های مربوطه ذخیره شده است در نظر گرفت. مقدار $PathIndex$ هر یک از ورودی‌های اصلی مدار برابر حاصل جمع $PathCount$ تمام ورودی‌های ماقبل آن ورودی بر اساس ترتیب ثابتی است که از قبل برای آن‌ها در نظر گرفته شده است. به طور مشابه مقدار $PathIndex$ هر یک از خروجی‌های انشعاب برابر حاصل جمع $PathCount$ تمام خروجی‌های ماقبل آن خروجی بر اساس ترتیب ثابتی است که برای آن‌ها در نظر گرفته شده است.

توجه شود که روش توضیح داده شده برای اندیس‌گذاری صرفاً مسیرهای فیزیکی در نظر می‌گیرد. درصورتی‌که با در نظر گرفتن نوع تغییر مقدار لاجیک در ابتدای مسیر تعداد مسیرهای منطقی دو برابر تعداد مسیرهای فیزیکی خواهد بود. بنابراین برای در نظر گرفتن مسیرهای منطقی (که شامل مسیر فیزیکی و تغییر در ورودی است) می‌توان مرحله اول یعنی پیمایش معکوس را با گذاشتن عدد ۲ در $PathCount$ تمام خروجی‌های اصلی مدار اصلاح کرد. این موضوع در شکل (۸) رعایت شده است. این شکل یک نمونه کامل از اندیس‌گذاری ترتیبی را نشان می‌دهد که در آن $PathCount$ ها با رنگ سبز و $PathIndex$ ها با رنگ آبی و اندیس دو مسیر به عنوان نمونه با رنگ نارنجی نشان داده شده است. برای هر مسیر فیزیکی دو اندیس در نظر گرفته شده که مربوط به دو حالت لبه بالارونده و پایین‌رونده در ورودی است.

۴،۷. روش‌های تجمیع مجموعه مسیرها

اندیس‌گذاری مسیرها با هدف کاهش هزینه زمان و حافظه در استخراج مجموعه مسیرهای آشکار شده برای هر آزمون ۳۲ بیتی و مقایسه و تجمیع مجموعه مسیرهای مربوط به آزمون‌های ۳۲ بیتی مختلف انجام می‌شود. برای تجمیع مجموعه مسیرها بسته به نوع اندیس‌گذاری می‌توان از دو روش متفاوت استفاده نمود. روش اول مقایسه و ادغام مجموعه اندیس‌ها و روش دوم

استفاده از یک چک لیست آرایه‌ای است که هر یک از آن‌ها در ادامه شرح داده می‌شود.

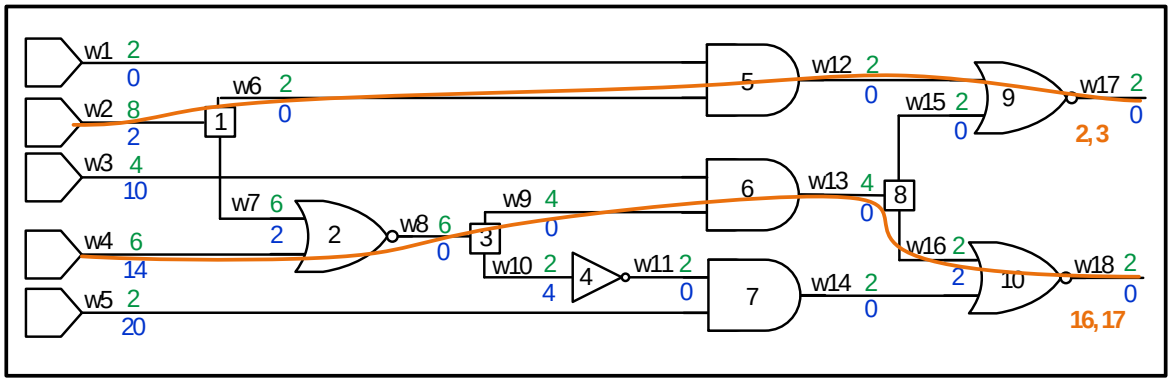
۴،۷،۱. روش مقایسه و ادغام مجموعه اندیس

وقتی از اندیس‌گذاری مبتنی بر امضا یا اندیس‌گذاری ترتیبی استفاده شود برای تجمیع مجموعه مسیرها نیازمند مقایسه مجموعه اندیس‌ها و ادغام آن‌ها هستیم. در این روش برای هر خروجی مدار یک آرایه با طول پویا برای نگهداری اندیس‌های تمام مسیرهای آشکارشده‌ی منتهی به آن خروجی در نظر گرفته شده و با اعمال هر آزمون ۳۲ بیتی اندیس‌های مسیرهایی که جدیداً آشکار شده به آن اضافه می‌شود. در این روش برای جلوگیری از ذخیره اندیس‌های تکرار لازم است تک‌تک اندیس‌های جدید با کل اندیس‌های موجود در آرایه مقایسه شود که این کار زمان‌بر است. از طرفی به ازای هر مسیر باید یک اندیس که معمولاً از نوع int و ۳۲ بیتی است ذخیره شود که این امر برای مدارهای خیلی بزرگ به حافظه زیادی نیاز دارد. لذا برای کاهش حافظه و زمان جستجو روش چک لیست آرایه‌ای پیشنهاد شده است که در ادامه شرح داده می‌شود.

۴،۷،۲. روش چک لیست آرایه‌ای

در این روش یک آرایه بیتی در نظر گرفته می‌شود که طول آن به تعداد کل مسیرهای مدار است. برای پیاده‌سازی آرایه بیتی می‌توان از یک آرایه از int با طول ۳۲ برابر کوتاه‌تر استفاده کرد. این روش به اندیس‌گذاری ترتیبی مسیرها نیاز دارد. از اندیس مسیر بعنوان اندیس آرایه استفاده می‌شود. در حقیقت آرایه بیتی یک چک لیست است که مقدار اولیه تمام بیت‌های آن صفر است عملیات تجمیع با یک دادن به تمام مکان‌هایی که اندیس آن در لیست مسیرهای تجمیعی قرار دارند انجام می‌شود. در این روش نیازی به مقایسه و کشف اندیس تکراری نیست و اندیس‌های تکراری صرفاً باعث دوباره یک دادن به یک مکان خواهد شد. این روش به علت عدم نیاز به جستجو و مقایسه سرعت تجمیع را بالا می‌برد ولی با توجه به اینکه تعداد بیت‌های آرایه برابر تعداد کل مسیرهای مدار است و برای مدارات بزرگ تعداد مسیرهای به‌شدت افزایش می‌یابد از نظر ظرفیت حافظه باز هم با محدودیت روبرو خواهد بود. ولی از نظر سرعت بهبود خوبی نسبت به روش مقایسه و ادغام خواهد داشت. یک راه حل برای مدارات بزرگ محدود کردن ظرفیت حافظه با یک اندیس (مثل) ۳۲ بیتی است. در این صورت ظرفیت آرایه و در نتیجه تعداد اندیس‌های ممکنه برابر ۱- ۲۳۲ خواهد بود و برای مداراتی که تعداد مسیرهای بیشتر دارند باعث همپوشانی اندیس و در نتیجه تقریب در پوشش مسیر بدست‌آمده خواهد شد.

یک مثال کامل از مراحل شبیه‌سازی اِشکال تاخیر مسیر به روش پیشنهادی در پیوست (الف) آورده شده است که در آن مراحل مختلف شبیه‌سازی اِشکال پیشنهادی برای سه آزمون مختلف شرح داده شده و با شکل نشان داده شده است. در بخش بعد نتایج شبیه‌سازی اِشکال تأخیر مسیر برای روش پیشنهادی ارائه می‌شود.



شکل (۸): مثال اندیس‌گذاری ترتیبی

۵. نتایج شبیه‌سازی

همان‌طور که پیش از این توضیح داده شده رویکرد اصلی ارائه شده در این مقاله برای شبیه‌سازی اِشکال تأخیر مسیر، رویکرد پیمایش مسیر بحرانی با موازات ۳۲ بیتی است که برای استخراج مسیر از روش اندیس‌گذاری ترتیبی و برای تجمیع مجموعه مسیرهای آشکار شده از چک لیست آرایه استفاده کرده است. در این بخش ابتدا نتایج شبیه‌سازی اِشکال تأخیر مسیر برای روش پیشنهادی با چهار نسخه دیگر آن که از نظر برخی ویژگی‌ها متفاوت هستند مقایسه می‌شود. سپس نتایج روش پیشنهادی را با آخرین نتایج گزارش شده در مقالات مرتبط مقایسه می‌شود. پیاده‌سازی روش پیشنهادی به زبان C++ انجام شده و همه آزمایش‌ها روی سیستمی با پردازنده Core i7 با فرکانس کاری 3.6 GHz و حافظه اصلی 16GB اجرا شده است.

۵,۱. بخش اول نتایج

در این بخش نتایج شبیه‌سازی اِشکال تأخیر مسیر برای روش پیشنهادی با چهار نسخه دیگر که در برخی ویژگی‌ها مانند موازات، استخراج مسیر و تجمیع مسیرهای آشکار شده متفاوت است مقایسه می‌شود. بنابراین نتایج برای ۵ نوع شبیه اِشکال گزارش شده است. جدول (۲) ویژگی‌های هر یک از این شبیه‌سازی‌ها را نشان می‌دهد. هر یک از جداول (۳) تا (۵) نتایج پنج نوع شبیه‌سازی اِشکال را در ۵ ستون با هم مقایسه می‌کند. جدول (۳) مربوط مسیرهای بحرانی، جدول (۴) مربوط به مسیرهای نیرومند و جدول (۵) مربوط به مسیرهای حساس است. مقایسه این نتایج برتری روش پیشنهادی را نسبت به سایر روش‌ها از نظر سرعت در هر سه جدول نشان می‌دهد.

با مقایسه میانگین زمان‌های گزارش شده در ستون‌های جداول (۳) تا (۵) می‌توان میزان تأثیر هرکدام را دریافت. به عنوان مثال برای بررسی تأثیر موازات ۳۲ بیتی بر افزایش سرعت است. می‌توان میانگین زمان اجرای روش سوم را بر روش پنجم تقسیم نمود. این کار کاهش ۵ برابری در زمان اجرا را نشان می‌دهد. برای بررسی استفاده از اندیس‌گذاری مسیرها و ایجاد لیست اندیس‌ها بجای لیست کامل زنجیره اتصالات مسیرها می‌توان پاسخ روش اول را بر روش سوم تقسیم نمود. این عمل کاهش ۷/۶ برابری زمان اجرا را نشان می‌دهد. همچنین می‌توان دریافت که تکنیک چک لیست آرایه‌ای برای تجمیع لیست اندیس مسیرها بطور میانگین در حدود ۵/۳ برابر زمان اجرا را کاهش می‌دهد. این عدد از تقسیم میانگین زمان اجرای روش چهارم بر روش پنجم (روش پیشنهادی) بدست می‌آید.

۵,۲. بخش دوم نتایج

جدول (۶) نتایج روش پیشنهادی را با آخرین نتایج گزارش شده در مقالات مرتبط مقایسه می‌کند. ستون سمت راست میزان افزایش سرعت را نسبت به بهترین نتیجه گزارش شده نشان می‌دهد. میزان افزایش سرعت برای هر مدار از تقسیم حداقل زمان گزارش شده از کارهای دیگران بر زمان اجرای الگوریتم پیشنهادی که بترتیب در ستون‌های سوم و دوم از راست جدول آورده شده‌اند، بدست آمده است. در سطر پایانی جدول، میزان افزایش سرعت روش پیشنهادی به صورت میانگین نشان داده است. برای محاسبه این مقدار میانگین حداقل زمان گزارش شده برای هر مدار از کارهای دیگران محاسبه شده و بر میانگین زمان اجرای الگوریتم پیشنهادی برای مدارهای مختلف تقسیم شده است. نتیجه بهبود ۱۸۶ برابری را نشان می‌دهد.

جدول (۲): معرفی ویژگی روش‌های شبیه‌سازی اِشکال آزمایش شده

روش‌های شبیه‌سازی اِشکال تأخیر مسیر	پیمایش مسیر		موازات		استخراج مجموعه مسیرها			تجمیع مجموعه مسیرها		
	مستقیم	بحرانی	۱ بیتی	۳۲ بیتی	مسیر کامل	اندیس مبتنی بر امضا	اندیس ترتیبی	ادغام مسیر کامل	ادغام اندیس‌ها	چک لیست آرایه‌ای
1bit_CPT_PathExt		✓	✓		✓			✓		
1bit_offpath	✓		✓			✓			✓	
1bit_CPT		✓	✓			✓			✓	
32bit_offpath	✓			✓		✓			✓	
32bit_CPT		✓		✓			✓			✓

جدول (۳): مقایسه نتایج روش‌های شبیه‌سازی اِشکال تأخیر مسیر برای مسیر بحرانی، CT:CPU_TIME و DP:#DETPATHS

Critical		1bit_CPT_PathExt		1bit_offpath		1bit_CPT		32bit_offpath		32bit_CPT	
Benchs	#TotalPaths	DP	CT	DP	CT	DP	CT	DP	CT	DP	CT
b11_C	21144	5856	3.296	5848	1.984	5856	1	5848	0.188	5856	0.094
b12_C	25788	9409	7.094	9409	2.532	9409	1.797	9409	0.25	9409	0.14
b13_C	1398	1130	1.187	1130	0.813	1130	0.61	1130	0.046	1130	0.047
b14_C	186784982	NA	NA	452008	38.985	452156	14.329	452009	10.484	452156	0.875
b15_C	$2^{36}<\#<2^{37}$	NA	NA	258575	25	259739	14.078	258579	4.719	259739	1.094
b17_C	$2^{40}<\#<2^{41}$	NA	NA	747532	104.203	749766	61.813	747539	15.64	749766	4.11
c1355	8346432	327454	258.375	325807	5.906	327454	0.922	325818	4.375	327454	0.156
c1908	1458114	37896	7.109	37768	2.782	37896	1.469	37768	0.61	37896	0.125
c2670	1359920	52228	30.969	51703	6.516	52228	2.422	51703	2.562	52228	0.235
c3540	57353342	NA	NA	345895	74.5	364710	2.75	345983	69.016	364710	0.421
c432	167852	9707	1.389	9707	0.593	9707	0.313	9707	0.125	9707	0.031
c499	18880	10812	5.126	10812	0.875	10812	0.406	10812	0.125	10812	0.047
c5315	2682610	152265	124.172	148606	12.469	152265	4.031	148617	4.954	152265	0.437
c7552	1452988	159199	330.281	156594	28.516	159199	6.969	156622	18.062	159199	0.782
c880	17284	6784	1.453	6784	1.062	6784	1.078	6784	0.172	6784	0.109
s15850	329476092	NA	NA	128496	28.141	283018	15.531	128888	9.203	283018	1.437
s38417	2783158	202382	873.328	199564	49.672	202382	35.25	199575	4.906	202382	2.641
s38584	2161446	87486	594.375	86084	52.046	87486	37.359	86095	3.813	87486	2.469

جدول (۴): مقایسه نتایج روش‌های شبیه‌سازی اِشکال تأخیر مسیر برای مسیر نیرومند، CT:CPU_TIME و DP:#DETPATHS

Robust		1bit_CPT_PathExt		1bit_offpath		1bit_CPT		32bit_offpath		32bit_CPT	
Benchs	#TotalPaths	DP	CT	DP	CT	DP	CT	DP	CT	DP	CT
b11_C	21144	2449	1.938	2449	1.547	2449	0.859	2449	0.125	2449	0.047
b12_C	25788	5502	3	5502	2.172	5502	2.125	5502	0.172	5502	0.094
b13_C	1398	941	0.891	941	0.734	941	0.531	941	0.047	941	0.031
b14_C	186784982	NA	NA	58319	20.39	58319	13.578	58319	2.312	58319	0.531
b15_C	$2^{36}<\#<2^{37}$	NA	NA	29915	17.218	29917	12.828	29915	1.485	29917	0.844
b17_C	$2^{40}<\#<2^{41}$	NA	NA	95434	75.828	95435	59.031	95434	5.781	95435	2.688
c1355	8346432	2595	1.249	2595	1.156	2595	0.782	2595	0.14	2595	0.047
c1908	1458114	3721	2.61	3711	2.25	3721	1	3711	0.14	3721	0.078
c2670	1359920	7875	3.984	7861	2.828	7875	1.781	7861	0.235	7875	0.109
c3540	57353342	0	0	20208	3.031	20227	1.813	20208	0.359	20227	0.125
c432	167852	1860	0.484	1860	0.421	1860	0.25	1860	0.031	1860	0.016
c499	18880	2246	0.563	2246	0.5	2246	0.312	2246	0.047	2246	0.016
c5315	2682610	15213	9.453	15213	5.594	15213	2.969	15213	0.468	15213	0.203
c6288	$2^{66}<\#<2^{67}$	NA	NA	49995	5.25	50210	3.25	49997	0.75	50210	0.547
c7552	1452988	24687	13.375	24679	7.453	24687	4.047	24679	0.75	24687	0.265
s15850	329476092	NA	NA	14141	17.469	14170	13.234	14141	1.125	14170	0.812
s38417	2783158	63664	204.001	63580	38.485	63664	28.578	63580	2.968	63664	1.985
s38584	2161446	40981	238.312	40969	46.328	40981	31.75	40969	2.953	40981	1.938

جدول (۵): مقایسه نتایج روش‌های شبیه‌سازی اِشکال تأخیر مسیر برای مسیر حساس، CT:CPU_TIME و DP:#DETPATHS

Sensitive		1bit_CPT_PathExt		1bit_offpath		1bit_CPT		32bit_offpath		32bit_CPT	
Benchs	#TotalPaths	DP	CT	DP	CT	DP	CT	DP	CT	DP	CT
b11_C	21144	1475	1.72	1475	1.422	1475	0.812	1475	0.11	1475	0.046
b12_C	25788	2532	2.593	2532	1.984	2532	1.516	2532	0.125	2532	0.078
b13_C	1398	834	0.829	834	0.703	834	0.484	834	0.047	834	0.031
b14_C	186784982	NA	NA	9046	17.656	9046	12.796	9046	1.594	9046	0.484
b15_C	2^{36}<#<2^{37}	NA	NA	8378	15.266	8378	12.593	8378	1.625	8378	0.781
b17_C	2^{40}<#<2^{41}	NA	NA	25956	71	25956	58.953	25956	4.344	25956	2.406
c1355	8346432	121	1.28	121	1.109	121	0.688	121	0.125	121	0.031
c1908	1458114	2223	2.562	2216	2.219	2223	0.969	2216	0.125	2223	0.063
c2670	1359920	4101	3.376	4099	2.547	4101	1.734	4099	0.188	4101	0.093
c3540	57353342	NA	NA	387	2.547	387	1.688	387	0.219	387	0.062
c432	167852	406	0.422	406	0.36	406	0.219	406	0.031	406	0.016
c499	18880	101	0.546	101	0.5	101	0.313	101	0.047	101	0.016
c5315	2682610	7995	6.672	7995	4.906	7995	3.031	7995	0.406	7995	0.157
c6288	2^{66}<#<2^{67}	NA	NA	14646	4.297	14629	3.063	14646	0.5	14629	0.469
c7552	1452988	11122	9.782	11122	7.282	11122	3.875	11122	0.625	11122	0.218
s15850	329476092	0	0	8121	16.609	8132	12.765	8121	1.032	8132	0.734
s38417	2783158	32172	133.406	32169	35.703	32172	26.922	32169	2.625	32172	1.703
s38584	2161446	25989	171.405	25984	43.437	25989	31.11	25984	2.828	25989	1.797

جدول (۶): مقایسه زمان اجرای روش پیشنهادی برای شبیه‌سازی اِشکال تأخیر مسیر با تعدادی از کارهای مرتبط به ازای ۱۰۰۰۰ جفت آزمون تصادفی

Circuit	Total Path	[۱۲]	[۱۳]	[۲۲]	[۲۳]	[۲۴]	Minimum	Our Method	Speedup
c432	83926	291.4	**17.53**	-	-	42	17.53	0.047	372.98
c499	9440	624.5	**19.22**	-	-	-	19.22	0.063	305.08
c880	8642	413.7	29.81	36	34	**26**	26	0.140	185.71
c1355	4173216	1676.5	215.04	312	108	**79**	79	0.203	389.16
c1908	729057	886.2	**92.70**	105	138	94	92.70	0.203	456.65
c2670	679960	2200.7	429.05	427	39	**30**	30	0.344	87.21
c3540	28676671	5897.8	3161.8	514	211	**166**	166	0.546	304.03
c5315	1341305	4913	487.76	731	82	**74**	74	0.640	115.63
c7552	726494	7743.6	628.36	1135	105	**98**	98	1.047	93.60
						Average:	66.94	0.359	186.46

۶. جمع‌بندی

یک روش بسیار سریع برای شبیه‌سازی اِشکال تأخیر مسیر ارائه شد. این روش برای افزایش سرعت از تلفیق شروط انتشار تاخیر مسیرهای نیرومند و غیرنیرومند که شروط ساده‌تری ایجاد می‌کند برای پیدا کردن پوشش اجتماع آنها استفاده می‌کند. این روش با استفاده از پیمایش مسیر بحرانی تمام گره‌های بحرانی مدار را علامت زده و کار جستجوی و استخراج مسیر را به همان گره‌ها محدود می‌سازد تا با کاهش فضای جستجو نیز باعث افزایش سرعت شود. استفاده از چک لیست آرایه‌ای در کنار اندیس‌گذاری مسیرها در هنگام شمارش مسیرها نیز باعث حذف عملیات مقایسه و جستجو هنگام ادغام مجموعه مسیرهای تازه شناسایی شده در لیست اصلی و در نتیجه افزایش سرعت می‌شود. استفاده از تکنیک‌های یاد شده در کنار موازات ۳۲ بیتی باعث افزایش سرعت چشمگیر روش پیشنهادی شده است، به طوری که نتایج روش پیشنهادی نسبت به آخرین نتایج

گزارش شده در مقالات مرتبط بطور میانگین ۱۸۶ برابر کاهش زمان اجرا را نشان می‌دهد. روش پیشنهادی برای شمارش مسیرهای غیرنیرومند از شمارش مسیرهای بحرانی که اجتماع مسیرهای نیرومند و غیرنیرومند است استفاده می‌کند. سپس با تفریق تعداد مسیرهای نیرومند از تعداد مسیرهای بحرانی تعداد مسیرهای غیرنیرومند شناسایی شده را نیز بدست می‌آورد.

از آنجا که شبیه‌سازی اِشکال خود بخشی از عملیات تولید آزمون است پیش‌بینی می‌شود بکارگیری روش پیشنهادی در عملیات تولید آزمون نیز بتواند منجر به کاهش زمان اجرا گردد. بررسی این موضوع به عنوان یکی از کارهای آتی در نظر گرفته شده است.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

مراجع

[1] Aguiar Y.Q., Zimpeck A.L., Meinhardt C., and Reis R., "Permanent and single event transient faults reliability evaluation EDA tool", Microelectronics Reliability, 64: 63-67, 2016.

[2] Shojiro A., VLSI Design and Test for Systems Dependability, Japan, Springer, 2019

[3] Parhami B., Dependable Computing: A Multilevel Approach, forthcoming book whose draft is available:https://web.ece.ucsb.edu/~parhami/text_dep_comp.htm.

[4] Bushnell M.L. and Agrawal V.D., Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits, Kluwer Academic Publishers, 2002.

[5] Krstic A. and Cheng K.-T., Delay Fault Testing for VLSI Circuits, US, Springer, 1998.

[6] Ehteram A., Sabaghian-Bidgoli H., Ghasvari H., and Hessabi S., "A Simple and Fast Solution for Fault Simulation Using Approximate Parallel Critical Path Tracing," in Canadian Journal of Electrical and Computer Engineering, 43(2): 100-110, 2020.

[7] Liu X., Hsiao M.S., Chakravarty S., and Thadikaran P.J., "Efficient transition fault ATPG algorithms based on stuck-at test vectors," J. Electronic Testing: Theory and Applic., 19(4):437–445, 2003.

[8] Liu T., Zhou Y., Liu Y., and Cai S., "Harzard-based ATPG for improving delay test quality," J. Electronic Testing: Theory and Applic., 31(1): 27–34, 2015.

[9] Jayanthy S. and Bhuvaneswari M.C., "Delay Fault Testing of VLSI Circuits", Test Generation of Crosstalk Delay Faults, pp. 15-35, 2019.

[10] Pomeranz I., "Non-Masking Non-Robust Tests for Path Delay Faults," 2020 IEEE 38th VLSI Test Symposium (VTS), San Diego, CA, USA, pp. 1-6, 2020.

[11] Smith G.L., "Model for delay faults based upon paths," Proc. Int. Test Conf., pp. 342-349, 1985.

[12] Fink F., Fuchs K., and Schulz M.H., "Robust and nonrobust path delay fault simulation by parallel processing of patterns," in IEEE Transactions on Computers, 41(12): 1527-1536, 1992.

[13] Majhi A.K., Jacob J., and Patnaik L.M., "A Novel Path Delay Fault Simulator using Binary Logic," VLSI Design, 4: 167-179, 1996.

[14] Chakraborty T.J., Agrawal V.D., and Bushnell M.L., "Path delay fault simulation of sequential circuits," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 8(2): 223-228, 2000.

[15] Jayanthy S., Bhuvaneswari M.C., and Kavitha T., "Simulation based ATPG for path delay faults in digital circuits using genetic algorithm," in Proceedings of the National Conference on Adaptive Sensors and Intelligent Systems, pp. 80–84, 2008.

[16] Pomeranz I. and Reddy S.M., "An efficient non-enumerative method to estimate path delay fault coverage," IEEE/ACM International Conference on Computer-Aided Design, Santa Clara, CA, USA, pp. 560-567, 1992.

[17] Pomeranz I. and Reddy S.M., "An efficient nonenumerative method to estimate the path delay fault coverage in combinational circuits," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 13(2): 240-250, 1994.

[18] Heragu K., Agrawal V.D., and Bushnell M.L., "Statistical methods for delay fault coverage analysis," Proceedings of the 8th International Conference on VLSI Design, New Delhi, India, pp. 166-170, 1995.

[19] Ali Y., Yamato Y., Yoneda T., Hatayama K., and Inoue M., "Parallel Path Delay Fault Simulation for Multi/Many-Core Processors with SIMD Units," IEEE 23rd Asian Test Symposium, Hangzhou, China, pp. 292-297, 2014.

[20] Schneider E., Holst S., Kochte M.A., Wen X., and Wunderlich H., "GPU-accelerated small delay fault simulation," Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, pp. 1174-1179, 2015.

[21] Schneider E., Kochte M.A., Holst S., Wen X., and Wunderlich H.-J., "GPU-accelerated simulation of small delay faults," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36(5): 829–841, 2017.

[22] Gharaybeh M.A., Agrawal V.D., and Bushnell M.L., "False-path removal using delay fault simulation," Proceedings Seventh Asian Test Symposium (ATS'98), Singapore, pp. 82-87, 1998.

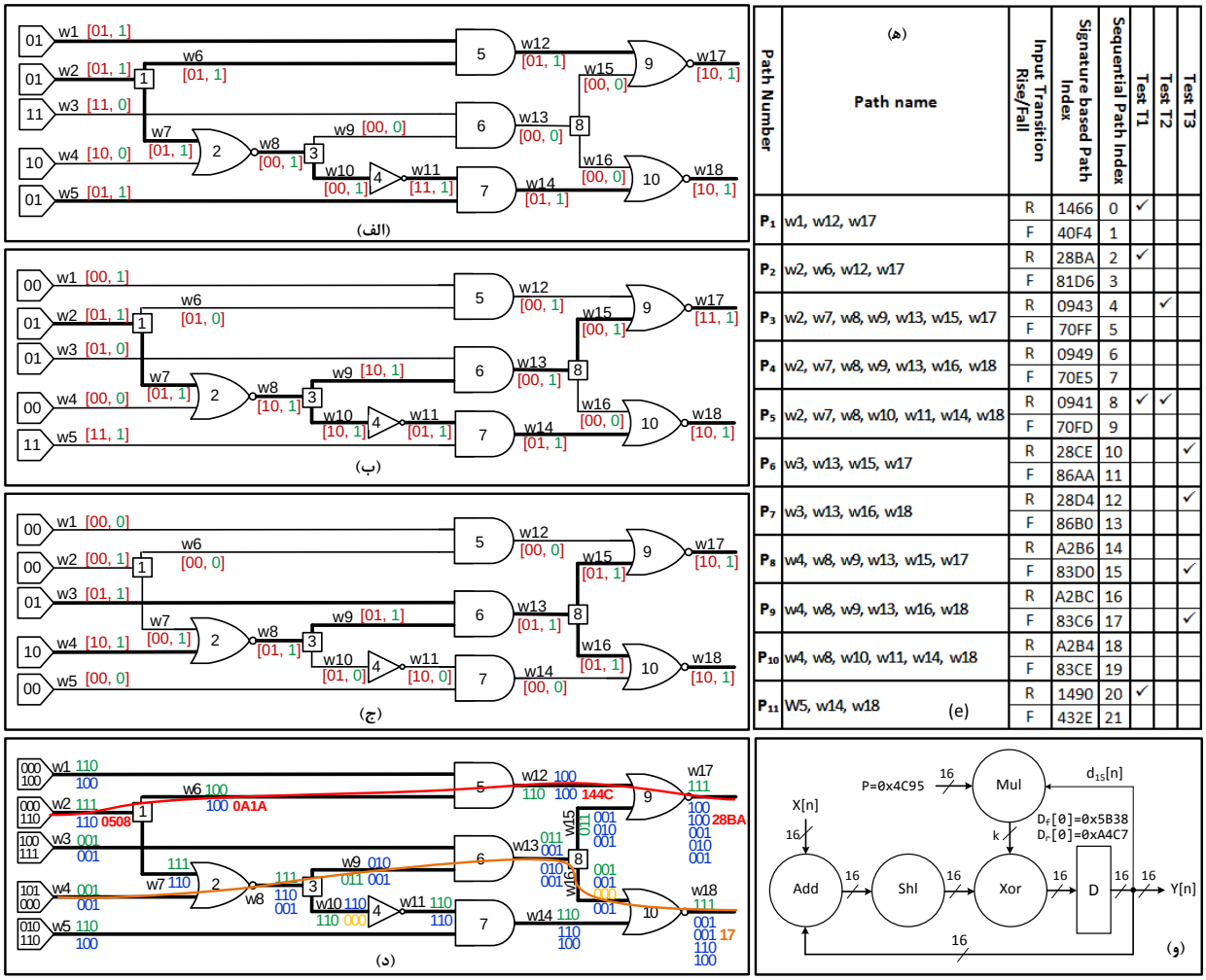
[23] Gjermandez O., "Exploiting arithmetic built-in self-test techniques for path delay fault testing," Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway, 2006.

[24] Manikandan P., Larsen B.B., and Aas E.J., "An Enhanced Path Delay Fault Simulator for Combinational Circuits," 14th Euromicro Conference on Digital System Design, Oulu, Finland, pp. 375-381, 2011.

پیوست

پیوست الف: یک مثال کامل از مراحل شبیه‌سازی اشکال تاخیر مسیر به روش پیشنهادی

شکل (۹) تمام مراحل شبیه‌سازی اِشکال پیشنهادی بـرای سـه آزمون مختلف را نشان می‌دهد. در این مثال هدف پیدا کـردن مسیرهای بحرانی است که توسط این سه آزمون تشخیص داده می‌شوند. شبیه‌سازی اِشکال ابتدا برای هر یک از آزمون‌های $T1$، $T2$و $T3$ بطور مجزا انجام شده است که به ترتیب در شکل‌های (الف)، (ب) و (ج) نشان داده شده است. سپس همان عملیات برای هر سه آزمون بطور موازی تکرار شده کـه در شـکل (د) نشان داده شده است. این چهار شکل نتایج را پـس از اجـرای مراحل ۱ تا ۳ الگوریتم پیشنهادی نشان می‌دهد. برای هر یک از سیم‌های مدار مقادیر V_1V_2 و C در قالب $[V_1V_2, C]$ در کنار آن سیم نوشته شده است.



(ه)

Path Number	Path name	Input Transition Rise/Fall	Signature based Path Index	Sequential Path Index	Test T1	Test T2	Test T3
P_1	w1, w12, w17	R	1466	0	✓		
		F	40F4	1			
P_2	w2, w6, w12, w17	R	28BA	2	✓		
		F	81D6	3			
P_3	w2, w7, w8, w9, w13, w15, w17	R	0943	4		✓	
		F	70FF	5			
P_4	w2, w7, w8, w9, w13, w16, w18	R	0949	6			
		F	70E5	7			
P_5	w2, w7, w8, w10, w11, w14, w18	R	0941	8	✓	✓	
		F	70FD	9			
P_6	w3, w13, w15, w17	R	28CE	10			✓
		F	86AA	11			
P_7	w3, w13, w16, w18	R	28D4	12			✓
		F	86B0	13			
P_8	w4, w8, w9, w13, w15, w17	R	A2B6	14			
		F	83D0	15			✓
P_9	w4, w8, w9, w13, w16, w18	R	A2BC	16			
		F	83C6	17			✓
P_{10}	w4, w8, w10, w11, w14, w18	R	A2B4	18			
		F	83CE	19			
P_{11}	W5, w14, w18	R	1490	20	✓		
	(e)	F	432E	21			

شکل (۹): مثالی از شبیه‌سازی اِشکال تأخیر مسیر برای سه آزمون *T1* تا *T3* (الف) نتایج *T1* پس از مرحله ۳ (ب) نتایج *T2* پس از مرحله ۳ (ج) نتایج *T3* پس از مرحله ۳ (د) نتایج آزمون سه بیتی موازی بعد از مرحله ۴ (ه) چک لیست تشخیص مسیر (و) گراف جریان داده‌ی MISR اندیس‌گذاری بر اساس امضا

در جدول شکل (ه) نام مسیر و لیست سیگنال(سیم)های تشکیل دهنده مسیر به ترتیب در ستون‌های ۱ و ۲ نشان داده شده‌اند. در ستون ۳ هر سلول جدول به دو ردیف تقسیم شده تا برای هـر مسیر فیزیکی دو مسیر منطقی یکی برای لبه بالارونده و دیگری

برای پایین‌رونده در ورودی مشخص نماید. لبه بالارونده با R و لبه پایین‌رونده با F نشان داده شده است. ستون ۴ و ۵ به ترتیب شامل اندیس مبتنی بر امضا و اندیس ترتیبی مسیرها است. سه ستون آخر شامل نتایج آشکارسازی آزمون‌های $T1$، $T2$ و $T3$ است. وضعیت بحرانی بودن (مقدار C) و نقاب (مقدار C_{mask}) نیز هر دو به شکل سه بیتی کنار سیگنال‌ها به ترتیب با رنگ‌های سبز و آبی نشان داده شده است. در شکل (د) دو مسیر نمونه را که یکی بر اساس اندیس مبتنی بر امضا و دیگری بر اساس اندیس ترتیبی مشخص شده نشان می‌دهد. این دو مسیر به ترتیب با دو رنگ قرمز و نارنجی نشان داده شده است.

شکل (و) نیز گراف جریان داده را برای $MISR$ ای که در این مثال جهت اندیس‌گذاری مبتنی بر امضای مسیر استفاده شده نشان می‌دهد. در این $MISR$ عرض بیت (یعنی k) برابر ۱۶ و شاخص چندجمله‌ای (یعنی P) برابر $0x4C95$ و مقدار اولیه (یعنی D[0]) برای مسیر با لبه بالارونده در ورودی 0xA4C7 و برای مسیر با لبه پایین‌رونده در ورودی $0x5B38$ در نظر گرفته شده است. همان‌طور که کمی بالاتر اشاره شد لیست اندیس‌های اختصاص یافته بوسیله این $MISR$ در ستون چهارم جدول شکل (ه) آمده است.