دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد
گرایش معماری کامپیوتر

عنوان:

کشف تروا‌های سخت افزاری: یک رویکرد اندازه‌ـآگاه

نگارش:

سید بهنام حیدرشاهی

استاد راهنما:

دکتر شاهین حسابی

خرداد ۱۳۹۴


بسم الله الرحمن الرحیم

به نام خدا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد

عنوان: کشف تروا‌های سخت افزاری: یک رویکرد اندازه‌ـ آگاه

نگارش: سید بهنام حیدرشاهی

کمیته‌ی ممتحنین

استاد راهنما: دکتر شاهین حسابی امضاء:

استاد داور داخلی: دکتر سیاوش بیات‌ـ سرمدی امضاء:

استاد داور خارجی: دکتر سعید صفری امضاء:

تاریخ:

سپاس

از استاد بزرگوارم که با کمک‌ها و راهنمایی‌های بی‌دریغشان، بنده را در انجام این پروژه یاری داده‌اند، تشکر و قدردانی می‌کنم. هم‌چنین از خانواده‌ی عزیزم صمیمانه سپاس‌گزارم.

چکیده

با افزایش ساخت مدارات مجتمع نیمه‌هادی در کارخانه‌هایی جدای از محل طراحی آنها، نگرانی راجع به امکان جایگذاری مدارات مخرب یا بدافزار در مدار افزایش یافته است. یک مساله اصلی وجود دارد که امنیت و قابلیت اعتماد تراشه‌ها را تحت تاثیر قرار می‌دهد. تغییرات و یا اضافه کردن مدار با اهداف بداندیشانه که تروای سخت‌افزاری نام دارد، به راحتی توسط فرآیندهای غیرمطمئن در روند طراحی و ساخت تراشه‌ها قابل انجام است [۱]. جعل و تقلب در ساخت تراشه‌ها، در چند سال اخیر به شدت افزایش یافته است [۲].

با وجود تلاشهای بسیاری که برای تشخیص تروا و همچنین جلوگیری از درج تروا انجام شده است، همچنان فقدان یک روش جامع و کامل در این حیطه محسوس است. تمام روش‌های موجود، یا صرفاً برای تروا‌های کوچک دارای عملکرد مطلوب هستند، یا منحصراً برای تروا‌های بزرگ. این پژوهش در تلاش است تا جای خالیِ ذکر شده را با معرفی یک روش ترکیبی و اندازه‌ـآگاه پر کند. رویکرد اندازه‌ـآگاه، با محاسبه تاثیر اندازه تروا در نتیجه‌ی آزمون و تنظیم آزمون بسته به اندازه‌ی تروا، در ازای بالا بردن پیچیدگی آزمون، به طور میانگین دقت تشخیص را ۱۰ درصد از روش‌های پیشین بهبود می‌بخشد. همچنین روش پیشنهادی ما قادر خواهد بود از لحاظ سرعت آزمون، نسبت به کارهای پیشین، نتایج بین ۸۰ تا ۹۰ درصد بهتری را به ارمغان آورد.

کلیدواژه‌ها: تروای سخت‌افزاری، قابلیت اطمینان، آزمون VLSI، آزمون اندازه‌ـآگاه، معماری کامپیوتر

فهرست مطالب

فهرست شکل‌ها

فهرست جدول‌ها

فصل ۱

مقدمه

نخستین فصل این پایان‌نامه به معرفی مسئله، بیان اهمیت موضوع، ادبیات موضوع، اهداف تحقیق و معرفی ساختار پایان‌نامه می‌پردازد.

۱ـ۱ تعریف مسئله

مسئله‌ی **کشف تروا‌های سخت افزاری:** مدار مجتمع یک مدار الکترونیکی است که بر روی یک تراشه از مواد نیمه‌هادی ساخته می‌شود و میلیون‌ها ترانزیستور یا سایر عناصر الکترونیکی می‌تواند روی آن قرار گیرد. این مدارات در تمامی ابزارهای الکترونیکی از رایانه‌ها و تلفنهای همراه گرفته تا ابزارهای نظامی و فضایی کاربرد دارند. به طور معمول و به دلیل کاهش هزینه، تولید انبوه این مدارها اکثراً به کارخانه‌های غیر قابل اعتماد سپرده می‌شود. هرگونه اختلاف آگاهانه بین طراحی مدار و سخت افزار تولیدی، که هدف آن تغییر، اختلال در، یا سرقت اطلاعات از عملکرد سخت افزار باشد، به معنی اضافه شدن تروا در مدار است. برای حصول اطمینان درباره‌ی عاری بودن مدار از چنین سخت‌افزارهای بداندیشانه، روش‌های متعددی وجود دارد. پژوهش با ترکیب دو روشِ آزمون منطقی و اثرات جانبی، تلاش دارد به یک آزمون با دقت بهتر از کارهای پیشین برسد. هدف دوم این آزمون یافتن نقطه‌ی مرزی برای مقایسه کارایی دو روش مذکور، با در نظر گرفتن اندازه‌ی تروا است.

۱ـ۲ اهمیت موضوع

برطبق گزارشهای وزارت دفاع آمریکا[۱] و اسناد وزارت بازرگانی آمریکا [۲] مدارات مجتمع به شدت در برابر فعالیتهای بداندیشانه و مخرب، آسیب پذیر شده‌اند. تروا‌های سخت‌افزاری مهمترین مساله امنیت و قابلیت اطمینان مدارات مجتمع در سالهای اخیر بوده است.

۱ـ۳ ادبیات موضوع

تشخیص تروا‌های سخت‌افزاری بسیار سخت است. چرا که اثر آنها بر عملکرد مدار همیشه قابل رؤیت نیست. یک تروا که به صورت حرفه‌ای طراحی شده باشد، ممکن است شامل تعداد اندکی دروازه منطقی باشد که در مکان‌های مختلف مدار درج شده باشند. بنابراین تغییرات در پارامترهای مدار تقریبا قابل چشم‌پوشی است. از طرفی از آنجا که مدار فعال‌کننده تروا معمولاً به نحوی انتخاب می‌شود که در شرایط نادری فعال شود و خروجی مدار تروا تا حد امکان قابل مشاهده نباشد، تشخیص تروا با استفاده از آزمون‌های رایج، بسیار مشکل خواهد بود. از طرفی به علت طیف گسترده تروا‌ها، ارائه مدلی که تمامی تروا‌ها را بتواند مدل کند و جهت تشخیص تروا به کار رود، ناممکن است. اخیراً روش‌های مختلفی برای تشخیص تروا ارائه شده‌است. این روش‌ها می‌توانند به سه دسته روش‌های تحلیل اثرات جانبی، روش‌های فعالسازی تروا و معماری‌های نظارتی، دسته‌بندی شوند. در دسته اول پارامترهای جانبی مدار تحلیل می‌شود تا براساس تغییرات آن، حضور تروا تشخیص داده شود. در این روش‌ها ممکن است تحلیل‌های مبتنی بر توان [۳, ۴, ۵] ، مبتنی بر جریان [۶] و مبتنی بر تاخیر [۷, ۸] انجام شود. این روش‌ها وقتی موثر هستند که اثر تروا بر پارامترهای جانبی شدید و کاملاً قابل تشخیص باشد. اما مسائلی وجود دارد که اثر تروا بر این پارامترها را کم می‌کند. برای مثال نویز اندازه‌گیری، تغییرات فرآیند و تغییرات محیطی می‌تواند اثر تروا بر این پارامترها را کمرنگ کند. دسته دوم روش‌هایی هستند که قصد فعالسازی تروا به طور کامل را دارند [۹, ۱۰, ۱۱, ۱۲] . با فعال شدن تروا، به احتمال بالا با چک کردن خروجی‌های مدار، عملکرد مخرب آن قابل مشاهده خواهد بود. مساله اصلی در این روش‌ها این است که زمان لازم برای فعالسازی تروا به طور کامل چقدر باشد. گاهی اوقات زمان فعالسازی آنقدر زیاد است که استفاده از این روش به صرفه نخواهد بود. از طرفی یک طراح تروا حرفه‌ای ممکن است تروایی طراحی کند که در شرایط بسیار نادر فعال شود و استفاده از این روش را مشکل سازد. همچنین

ممکن است بعضی تروا‌ها اثری بر خروجی مدار نداشته باشند [۱۳]. ساختارهای ناظر برای جلوگیری از آسیبهای ناشی از تروا ارائه شده‌اند. برای نمونه ساختار DEFENSE در طراحی عملیاتی مدارات منطقی تعبیه می‌شود تا به صورت بی درنگ بر امنیت مدار نظارت کند [۱۴]. در [۱۵] یک معماری گذرگاه در هاSoC ارائه شده‌است که نسبت به درج تروا مقاوم سازی شده به نحوی که از دسترسی نامطمئن به داده‌های امن جلوگیری کند. مشکل این روش این است که نظارت بر تمام اجزای مداری با میلیون‌ها دروازه منطقی، ناممکن است. هدف ما در بخش نخست این پژوهش ارائه روش‌های نوینی برای تشخیص تروا است. این روش‌ها باید:

۱) برای تشخیص تروا‌هایی که حتی از تعداد کمی دروازه منطقی تشکیل شده‌اند، کارا باشند.

۲) اثر تغییرات فرآیند و تغییرات محیطی بر این روش‌ها باید حداقل باشد.

۳) اندازه‌گیری‌ها و فرآیند تشخیص در این روش‌ها باید تا حد امکان دارای حداقل سربار زمانی، سربار هزینه و سربار سخت‌افزاری باشد و استفاده از آن به سهولت ممکن باشد.

برای بهبود کارایی روش‌های تشخیص تروا و رفع محدودیت‌های آنها، روش‌های متعددی توسط جامعه محققان اطمینان و امنیت سخت‌افزاری ارائه شده که هدف آنها تغییر روال طراحی کنونی است. به این روش‌ها، روش‌های طراحی برای اطمینان سخت‌افزاری می گویند [۳]. هدف از این روش‌های بازدارنده تروا این است که مانع درج تروا شوند و تشخیص تروا‌ها را تسهیل کنند. برخلاف روش‌های تشخیص تروا که روش‌های منفعلانه هستند، روش‌های طراحی مطمئن، روش‌هایی فعال هستند. یعنی ساختار مدار را به نحوی تغییر می‌دهند تا مانع از درج تروا شوند. اکثر روش‌های بازدارنده از تروا، با هدف تسهیل در تشخیص تروا با استفاده از روش‌های تحلیل اثرات جانبی ارائه شده‌اند. از این پس به این روش‌ها، روش‌های مبتنی بر اثرانگشت اثرات جانبی گفته می‌شود. بعضی از این روش‌ها صرفا امکاناتی برای اندازه‌گیری پارامترهای جانبی فراهم می‌کنند ولی برخی دیگر مقادیر اندازه‌گیری شده را با مقادیر آستانه‌ای که از قبل تعریف شده‌اند، مقایسه می‌کنند. سربار طراحی عمده ترین چالش این روش‌هاست. در برخی از روش‌های طراحی مطمئن، با این فرض که طراح تروا، مدار تحریک تروا را از قسمت‌هایی انتخاب می‌کند که به ندرت فعال می‌شوند(کنترل‌پذیری پایینی دارند)، هدف، حذف رخدادهای نادر است. روش درج فلیپ فلاپ‌های پویش [۱۶] و روش ولتاژ معکوس [۱۷] با هدف متعادل کردن فرکانس گذار سیگنال‌های داخلی برای حذف رخدادهای نادر، ارائه شده‌اند. از طرفی برخی روش‌ها بدون حذف رخدادهای نادر، کاری می‌کنند که طراح تروا در تشخیص رخدادهای نادر دچار خطا شود. روش مبهم سازی [۱۸] به نوعی ساختار واقعی مدار را پنهان می‌کند تا حمله‌کننده نتواند

احتمال واقعی رخدادها را حساب کند و براساس آن در انتخاب محل درج تروا به خطا رود. مشکل اکثر این روش‌ها سربار طراحی است و اینکه در برخی موارد فرضیه استفاده از رخدادهای نادر چندان درست نیست. در مقابل روش‌های دیگری هستند که برای اینکه بر چنین فرضیاتی استوار نباشند، از روش طراحی برای آزمون تروا DFTT[۱] که در [۱۹] ارائه شده‌است استفاده می‌کنند. این روش‌ها سربار سخت‌افزاری بیشتری دارند. برخی دیگر از روش‌ها نیز در دسته روش‌های طراحی مطمئن جای میگیرند. روش‌هایی که هدفشان حفاظت از IP است. در [۲۰, ۲۱] مفهوم سخت‌افزار حامل اثبات PCH[۲] ارائه شده‌است که مبتنی بر روش حفاظت نرم‌افزاری کد حامل اثبات PCC[۳] است. این روش برای ممانعت از درج تروا در IP ارائه شده‌است. هدف ما در بخش دوم این پژوهش این است که روش‌های جدیدی برای تسهیل روش‌های تشخیص تروای ارائه شده و یا برای مقاوم سازی مدار در برابر درج تروا ارائه کنیم.. این روش‌ها باید:

۱) تا حد امکان سربار زمانی و سخت‌افزاری کمتری نسبت به روش‌های پیشین داشته باشند.

۲) طراحی مطمئن مدار را یا بسیار راحت کنند و یا ابزاری برای خودکار سازی این روند ارائه کنند.

۱ـ۴ اهداف تحقیق

در این پایان‌نامه سعی می‌شود که مسئله‌ی کشف تروای‌های سخت‌افزاری مورد مطالعه قرار گیرد.برای حصول اطمینان درباره‌ی عاری بودن مدار از چنین سخت‌افزارهای بداندیشانه، روش‌های متعددی وجود دارد. این پژوهش با ترکیب و تمرکز رویِ دو روشِ آزمون منطقی و اثرات جانبی، تلاش دارد در وهله نخست به یک آزمون کشف تروا با دقت بهتر از کارهای پیشین برسد. هدف دوم این آزمون یافتن نقطه‌ی مرزی برای مقایسه کارایی دو روش مذکور، با در نظر گرفتن اندازه‌ی تروا است. بعد از مطالعه‌ی کارهای انجام شده در این زمینه سعی می‌شود که مسئله به صورت دقیق‌تر مورد بررسی قرار گیرد.

[۱] Design for Trojan Testability

[۲] Proof Carrying Hardware

[۳] Proof Carrying Code

۱ـ۵ ساختار پایان‌نامه

این پایان‌نامه شامل پنج فصل است. فصل دوم دربرگیرنده‌ی تعاریف اولیه‌ی مرتبط با پایان‌نامه است. در فصل سوم مسئله‌ی کشف تروا و کارهای مرتبطی که در این زمینه انجام شده به تفصیل بیان می‌گردد. در فصل چهارم نتایج جدیدی که در این پایان‌نامه به دست آمده ارائه می‌گردد. در این فصل، به صورت دقیق و گام به گام الگوریتم‌ها، برنامه‌ها و شبیه سازهایی که در این پژوهش تولید و یا استفاده شده‌اند معرفی و بررسی می‌شوند. در نهایت خروجی شبیه‌سازی‌ها ارائه خواهد شد. فصل پنجم به نتیجه‌گیری و پیش‌نهادهایی برای کارهای آتی خواهد پرداخت.

فصل ۲

مفاهیم اولیه

دومین فصل این پایان‌نامه به معرفی مفاهیمی می‌پردازد که در پایان‌نامه مورد استفاده قرار می‌گیرند.

۲ـ۱ تروا‌های سخت‌افزاری

تروا‌های سخت‌افزاری مدارهایی با عملیات بداندیشانه هستند که ممکن است به مدار اصلی افزوده شوند. این تروا‌ها در مراحل مختلف از زمان طراحی در سطح انتقال ثبات RTL[۱] تا ساخت تراشه ممکن است توسط افراد یا شرکتها و کارخانجات غیرمطمئن در مدار جاسازی شوند [۳]. فرآیند طراحی و ساخت مدارات مجتمع شامل چهار مرحله عمده است: طراحی سطح انتقال ثبات(شامل مشخصه مدار، بلاکهای IP[۲] و افراد طراح)، طراحی فیزیکی(شامل ابزارهای CAD[۳] مدل‌ها و افراد طراح)، ساخت(شامل تولید ماسکها و لیتوگرافی)، و آزمون ساخت(شامل آزمودن ویفر و بسته بندی). در فرآیند طراحی در سطح RTL و فیزیکی فرض می‌شود که ابزارهای CAD مطمئن و قابل اعتماد هستند چرا که توسط شرکتهای معتبر طراحی می‌شوند. اما بلاکهای IP مدل‌ها و سلول‌های استانداردی که توسط طراحان استفاده می‌شوند می‌توانند نامطمئن باشند. همچنین فاز ساخت نیز می‌تواند نامطمئن باشد.

این تروا‌های سخت‌افزاری می‌توانند تراشه‌ها را تخریب کنند، باعث رفتار اشتباه شوند یا دسترسی

[۱] Register Transfer Level
[۲] Intellectual Property
[۳] Computer Aided Design

حمله‌کننده‌ها به کلیدهای سرّی را فراهم کنند. از سال ۲۰۰۷ محققین برای یافتن روش‌های تشخیص تروا برای جلوگیری از آسیب‌های آن، پژوهش‌هایی انجام داده‌اند [۴]. روش عمومی این است که با اعمال تعداد زیادی ورودی مختلف، تروا را فعال کنند و با مشاهده تفاوت در رفتار مدار بدون تروا با مدار دارای تروا، حضور تروا را تشخیص دهند. با این وجود، فعال‌سازی اغلب تروا‌ها کار بسیار مشکلی است. از طرفی ارزیابی کامل تمام بخش‌های مداری که شامل میلیون‌ها گیت است، زمان بسیار زیادی لازم دارد. یکی از راههای جایگزین استفاده از اطلاعات پارامترهای جانبی مدار از قبیل توان مصرفی، تاخیر مسیرها و جریان نشتی است. مدارات حاوی تروا پارامترهای جانبی متفاوتی با مدارات بدون تروا دارند. مشکل اصلی این روش‌ها این است که اثر تروا بر پارامترهای جانبی ممکن است توسط اثر تغییرات فرآیند پوشش داده شود. از طرفی تشخیص تروا‌های داخل IP در این روش بسیار مشکل است چرا که اکثر IP ها به صورت کد RTL ارائه می‌شوند.

به علت معایبی که روش‌های موجود دارند، لازم است روش‌های جدیدی ارائه شود تا امنیت و قابلیت اطمینان سیستم‌های الکترونیکی، بخصوص سیستم‌های با کاربرد بحرانی را بالا ببرد. یکی از اهداف ما در این پژوهش ارائه روش‌هایی برای تشخیص تروا‌های سخت‌افزاری به طور موثر است. هدف دیگر ما ارائه راهکارهایی برای مقاوم سازی مدار در برابر تروا است. در هر دو بخش ایده اصلی این است که به نحوی کنترل‌پذیری و مشاهده‌پذیری نقاط مختلف مدار را افزایش دهیم. چرا که اغلب طراحان تروا، تروا‌ها را در نقاطی درج می‌کنند که با اعمال بردارهای آزمون نتوان به راحتی به ورودی‌های فعالساز آنها و خروجی‌های آنها دسترسی داشت. بنابراین نمی‌توان آنها را به راحتی فعال نمود و اثرشان را در خروجی یا پارامترهای جانبی مشاهده کرد.

۲ـ۲ دسته‌بندی تروا‌های سخت‌افزاری

برای تسهیل فرآیند تشخیص تروا یا کاهش اثرات مخرب آن و ابداع روش‌های محافظت در برابر تروا، لازم است تا ابتدا تروا‌های سخت‌افزاری دسته‌بندی شوند و براساس این دسته‌بندی کارهای بعدی انجام شود. پژوهش‌های بسیاری در این زمینه انجام شده‌است [۲۳, ۲۲, ۱۳, ۱۱, ۳]. تروا‌های سخت‌افزاری را می‌توان براساس پنج ویژگی دسته‌بندی نمود:

- فازی از طراحی که تروا در آن به مدار افزوده می‌شود

- سطح انتزاع
- روش فعال شدن تروا
- عملکرد تروا
- محل قرارگیری

در ادامه به بررسی این ویژگی‌ها می‌پردازیم.

۲ـ۲ـ۱ فاز درج تروا

الف) فاز مشخصات

در این فاز مشخصات سیستم(هدف، محیط عملکرد، عملیات مورد انتظار، اندازه، توان، تاخیر و ...) تعریف می‌شود. در این فاز می‌توان مشخصه عملیاتی یا سایر قیود طراحی را تغییر داد. تروا در این فاز ممکن است نیازمندی‌های زمانی سخت‌افزار را دست کاری کند.

ب) فاز طراحی

در فاز طراحی قیود عملیاتی، منطقی، زمانبندی و فیزیکی برای نگاشت طرح روی تکنولوژی مقصد مدنظر قرار می گیرد. در این فاز طراحان ممکن است از بلاکهای IP دیگران یا سلولهای استاندارد آنها استفاده کنند که ممکن است شامل تروا باشند.

ج) فاز ساخت

در این فاز مجموعه ماسکها ساخته می‌شود و ویفرها براساس این ماسکها تولید می‌شوند. تغییر ماهرانه ماسکها می‌تواند منجر به اثرات مخربی شود یا تغییر ترکیبات شیمیایی در طی فرآیند تولید می‌تواند منجر به افزایش پدیده مهاجرت الکترونی در مدارات بحرانی شود و فرآیند خراب شدن مدار را تسریع کند.

د) فاز مونتاژ

در فاز مونتاژ تراشه و سایر عناصر سخت‌افزاری روی PCB مونتاژ می‌شوند. هر واسطی که دو عنصر را به هم مرتبط کند، پتانسیل درج تروا را دارد. برای مثال سیم بدون روکشی که روی PCB به یک عنصر وصل شده‌است، تزویج الکترومغناطیس بین سیگنال‌های روی بورد و سیگنال‌های محیط را باعث می‌شود. از این مساله می‌توان برای نشت اطلاعات یا تزریق اشکال استفاده کرد.

ه) فاز آزمون

در این فاز امکان درج تروا نیست ولی اهمیت آن به خاطر شانس تشخیص تروا است. اگر خود فرآیند آزمون قابل اعتماد باشد، می‌توان از آن برای تشخیص تروا استفاده کرد.

۲ـ۲ـ۲ سطح انتزاع

الف) سطح سیستم

در سطح سیستم عناصر سخت‌افزاری متفاوت، اتصالات، و پروتکل‌های ارتباطی که استفاده می‌شوند، توصیف می‌شوند. در این سطح تروا ممکن است توسط عناصر داخل سخت‌افزار هدف فعال شود. برای مثال ممکن است مقادیر کد ASCII که از صفحه کلید گرفته می‌شود، باعث فعال شدن تروا شود.

ب) سطح انتقال ثبات

در این سطح هر ماژول عملیاتی برحسب ثبات‌ها، سیگنال‌ها و عملیات بولی توصیف می‌شود. از آنجا که در این سطح حمله‌کننده کنترل کامل بر عملیات سخت‌افزاری دارد، تروا به راحتی قابل طراحی و درج است. برای مثال یک تروا ممکن است تعداد دفعات اجرای مرحله‌ای از الگوریتم رمز نگاری را با دو برابر کردن گام حلقه تکرار، نصف کند و منجر به اشکال در سیستم رمزنگاری شود.

ج) سطح دروازه‌های منطقی

در این سطح، طراحی به صورت اتصال بین دروازه‌های منطقی توصیف می‌شود. در این مرحله حمله‌کننده به سیستم به راحتی می‌تواند جنبه‌های مختلف تروای که می خواهد درج کند(مانند اندازه و محل درج) را کنترل کند. در این سطح تروا می‌تواند یک مقایسه‌گر ساده با دروازه‌های منطقی XOR باشد که بر سیگنال‌های داخلی تراشه نظارت می‌کند. تروا می‌تواند ترکیبی یا ترتیبی باشد. این تروا‌ها معمولا برای تغییر عملکرد طرح به کار می روند و از این رو به تروا‌های عملیاتی معروفند.

د) سطح ترانزیستور

در این سطح، مدار با استفاده از ترانزیستورهایی که برای ساختن دروازه‌های منطقی استفاده می‌شوند، توصیف می‌شود. در این سطح، طراح تروا کنترل کاملی بر مشخصه‌های مداری سیستم مانند توان و زمانبندی دارد. تروا ممکن است با افزودن یا کاستن ترانزیستورها یا تغییر اندازه آنها برای تغییر پارامترهای مدار، درج شده باشند. تروا‌های این سطح نیز از جمله تروا‌های عملیاتی هستند.

ه)سطح layout

در این سطح ابعاد و محل همه عناصر مدار توصیف می‌شود. در این سطح تروا ممکن است از طریق تغییر اندازه سیم‌ها، فواصل بین عناصر مدار و تخصیص دوباره لایه‌های فلز درج شود. برای مثال تغییر عرض سیم‌های فلزی شبکه ساعت در تراشه می‌تواند منجر به انحراف سیگنال ساعت شود. به این تروا‌ها تروا‌های پارامتری می گویند. بسته به تعداد دروازه‌های منطقی که توسط تروا در مدار درج می‌شود، تروا می‌تواند به دو دسته کوچک و بزرگ دسته‌بندی شود. همچنین براساس توزیع آن در طرح، به دو دسته متمرکز و توزیع شده دسته‌بندی شود.

۲ـ۲ـ۳ روش فعال شدن تروا

بعضی از تروا‌ها به نحوی طراحی می‌شوند که همیشه فعال هستند. بعضی دیگر، تا زمانی که توسط سیگنال یا الگوی خاصی تحریک نشوند، فعال نمی‌شوند. معمولاً تروا‌های پارامتری از دسته «همیشه فعال» هستند. تروا‌هایی که تحریک می‌شوند، نیازمند یک رخداد برای فعالسازی هستند. این رخداد

می‌تواند داخلی یا خارجی باشد. بعد از فعال شدن، این تروا‌ها ممکن است تا ابد فعال بمانند و یا بعد از مدت زمانی، به حالت غیرفعال بازگردند.

الف) تروا‌های با تحریک داخلی

این تروا‌ها توسط رخدادی که درون سیستم هدف رخ می‌دهد، فعال می‌شوند. این رخداد ممکن است مبتنی بر زمان یا مبتنی بر شرایط فیزیکی باشد. شرایط فیزیکی شامل گستره وسیعی از عوامل هستند. از جمله تداخلات الکترومغناطیس، رطوبت، ارتفاع، فشار جو، دما و غیره . همچنین ممکن است یک تروا هنگام ورود به یک حالت خاص از ماشین حالات طرح، فعال شود.

ب) تروا‌های با تحریک خارجی

این دسته تروا‌ها نیازمند یک ورودی از دنیای خارج به ماژول هدف هستند. این ورودی می‌تواند یک ورودی از کاربر مثل فشردن دکمه یا وارد کردن عبارت خاص، یا خروجی یک عنصر باشد. برای مثال تروا ممکن است توسط داده‌ای که از واسط RS-232 دریافت می‌کند، فعال شود. معمولا این دسته از تروا‌ها نیازمند یک مدار حسگر هستند تا بتوانند تحریک خارجی را دریافت کنند. در یک دسته‌بندی دیگر می‌توان تروا‌های با تحریک داخلی یا خارجی را به دو دسته تحریک شونده با حسگر یا تحریک شونده با شرایط منطقی، تقسیم کرد.

۲ـ۲ـ۴ عملکرد تروا

الف) تغییر عملیات

تروا‌ها ممکن است عملکرد مدار را تغییر دهند و یا خطای ظریفی ایجاد کنند که تشخیص آن سخت باشد. برای مثال ممکن است تروا کاری کند که یک ماژول تشخیص خطا، الگوی نادرست را به عنوان درست تشخیص دهد.

ب) کاهش قابلیت اطمینان

تروا‌ها می‌توانند با تغییر عمدی پارامترهای مدار، باعث کاهش کارایی شوند. آنها ممکن است مشخصات پارامتری، واسطه‌ای یا عملیاتی مثل توان و تاخیر را تغییر دهند. همچنین تروا می‌تواند اشکالی (مانند اشکال stuck-at و اشکال پل زنی) را در مدار ایجاد کند و باعث کاهش قابلیت اطمینان شود.

ج) نشت اطلاعات

تروا‌ها ممکن است اطلاعات حساس را فاش کنند. این کار با استفاده از کانال‌های آشکار یا پنهان انجام شود. این کانال‌ها می‌توانند شامل امواج فرکانس رادیویی، امواج نوری، گرما، توان و اثرات جانبی زمانبندی باشند.

د) رد کردن خدمات

این تروا‌ها می‌توانند مانع عملکرد یک تابع یا یک منبع شوند. تروا ممکن است از نظر فیزیکی باعث تخریب یا غیرفعالسازی یا تغییر پیکربندی مدار شود. این اثر می‌تواند موقت یا دایمی ظاهر شود.

۲ـ۲ـ۵ محل قرارگیری

تروا می‌تواند در یک عنصر واحد درج شود و یا در میان چندین عنصر مختلف توزیع شود. تروا‌ها می‌توانند در عناصر پردازشی، حافظه، ورودی/خروجی، شبکه تغذیه یا شبکه ساعت درج شوند. تروا‌هایی که بین عناصر توزیع می‌شوند، می‌توانند مستقل از هم کار کنند یا به عنوان یک گروه عمل کنند.

الف) تروا‌های واحد پردازشی

برای مثال این تروا ممکن است ترتیب اجرای دستورات را عوض کند.

ب) تروا‌های حافظه

شامل تروا‌های داخل بلاکهای حافظه و واسط حافظه می‌شود. این تروا‌ها ممکن است مقادیر داخل حافظه را تغییر دهند یا مانع از عملیات خواندن یا نوشتن به محل خاصی از حافظه شوند.

ج) تروا‌های ورودی/خروجی

این تروا‌ها می‌توانند در ابزارهای جانبی یا روی PCB قرار گیرند. چنین تروا‌هایی کنترل بر انتقال داده بین پردازنده و عناصر خارجی خواهند داشت.

د) تروا‌های منبع تغذیه

با تغییر ولتاژ یا جریان تغذیه باعث از کار افتادن تراشه می‌شوند.

ه) تروا‌های شبکه ساعت

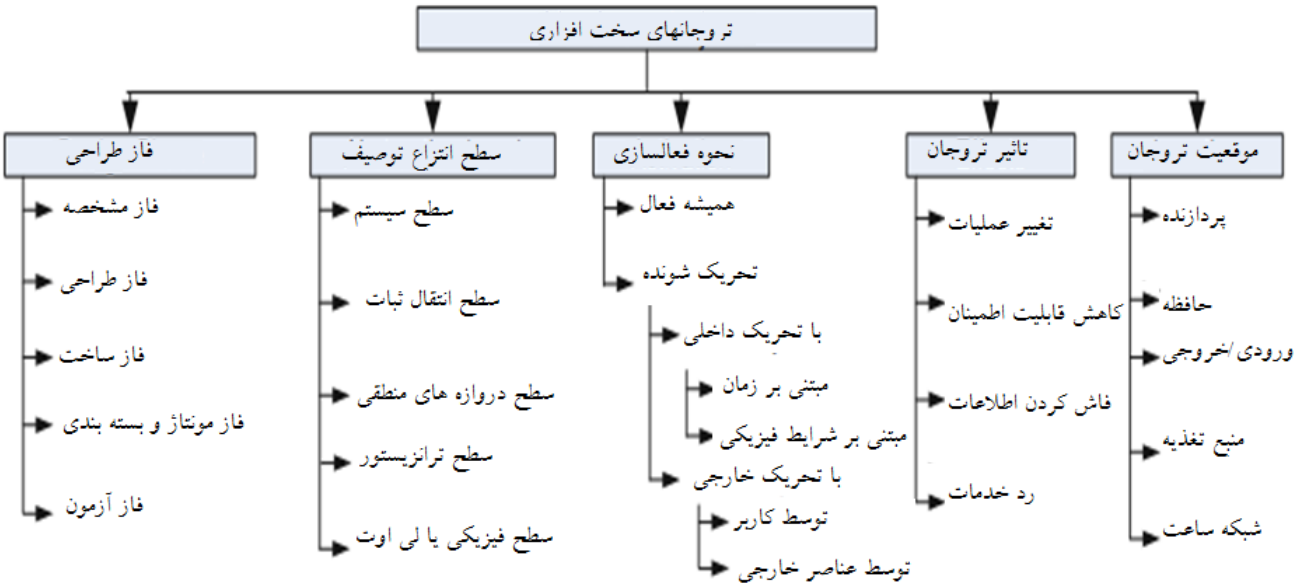
این تروا‌ها با تغییر فرکانس پالس ساعت یا افزودن تغییرات ناخواسته در سیگنال ساعت، منجر به اشکال در تراشه می‌شوند. همچنین این تروا‌ها می‌توانند پالس ساعت را متوقف کنند.

۲ـ۳ مدل کردن تروا‌های سخت‌افزاری

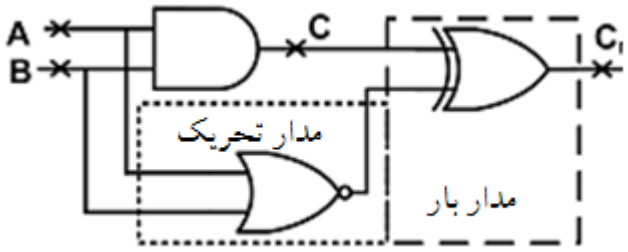
تروا‌های سخت‌افزاری از دو قسمت مدار تحریک و مدار بار تشکیل شده‌اند. مدار تحریک درواقع شرایط فعال شدن تروا را نشان می‌دهد و مدار بار کاری را که تروا بعد از فعال شدن انجام می‌دهد، مشخص می‌کند. در زیر مدل‌های ساده شده و استانداردی از تروا‌های سخت‌افزاری را با مدارهای تحریک و بار متفاوت معرفی می‌کنیم.

۲ـ۳ـ۱ مدار تحریک

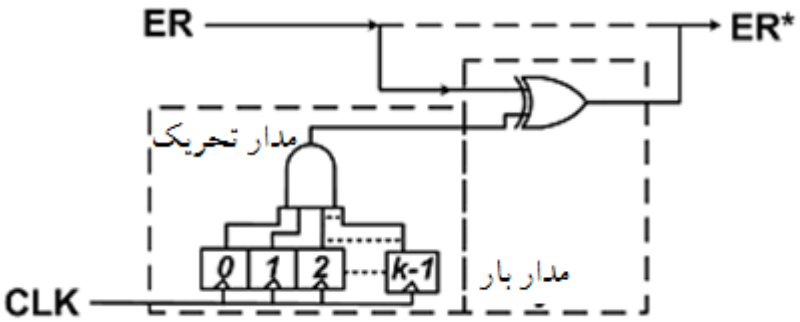
مدار تحریک در تروا‌ها می‌تواند آنالوگ یا دیجیتال باشد. تروا‌هایی با تحریک دیجیتال می‌توانند توسط یک مدار ترکیبی یا یک مدار ترتیبی تحریک شوند. در شکل مدار تروای با تحریک دیجیتال ترکیبی



شکل ۲ ـ ۱ : دسته‌بندی تروا‌ها



شکل ۲ـ۲: تروا با تحریک دیجیتال ترکیبی [۲۴]



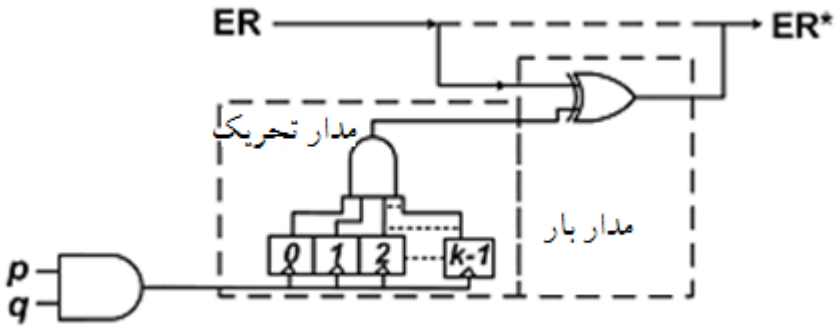
شکل ۲ـ۳: تروا با تحریک دیجیتال ترتیبی همگام [۲۴]

نشان داده شده‌است. در این مدار اگر هر دو ورودی A،B همزمان صفر شوند، تروا فعال می‌شود و مدار XOR که به عنوان مدار بار استفاده شده‌است، در صورت فعال شدن تروا، باعث تولید نتیجه اشتباه می‌شود.

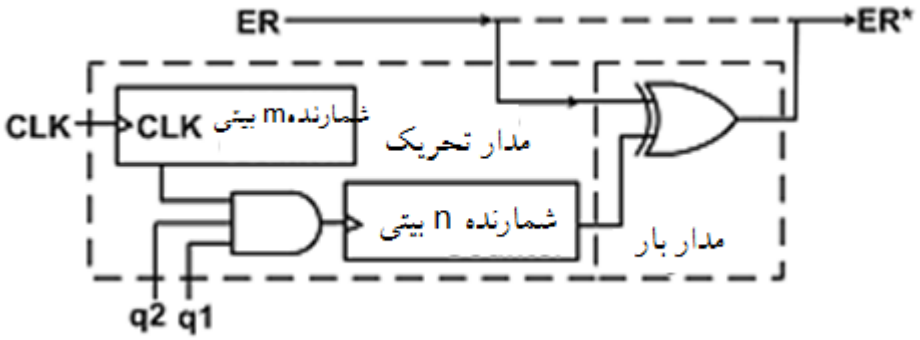
تروا‌های با تحریک ترتیبی که به آنها بمب ساعتی نیز می گویند، براثر رخداد رشته ای از وقایع یا براثر طولانی شدن یک رخداد خاص در مدت زمان از پیش تعریف شده، فعال می‌شوند. ساده ترین نوع مدار تحریک تروا‌های ترتیبی، شمارنده همگام است که بعد از رسیدن به یک تعداد شمارش، موجب تحریک تروا می‌شود. شکل ۲ـ۳ یک نمونه از این دسته تروا‌ها را نشان می‌دهد.

شمارنده‌های غیرهمگام نیز می‌توانند به عنوان مدار تحریک تروا‌ها استفاده شوند. نمونه ای از این تروا‌ها در شکل ۲ـ۴ آمده است.

در این شمارنده‌ها، تعداد رخداد وقایع خاص، شمرده شده و بعد از رسیدن به یک مقدار از پیش تعیین شده، تروا تحریک می‌شود. مدار تحریک می‌تواند مشابه شکل ۲ـ۵ ترکیبی از شمارنده‌های



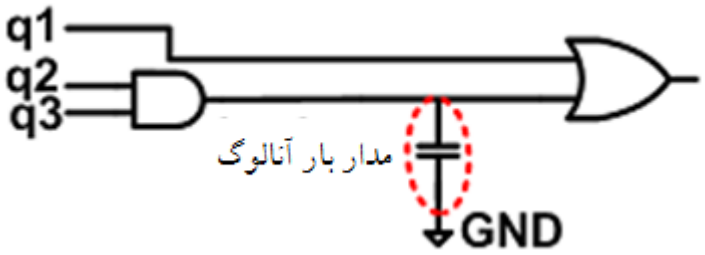
شکل ۲ـ۴: تروا با مدار تحریک دیجیتال ترتیبی از نوع شمارنده غیرهمگام [۲۴]



شکل ۲ـ۵: تروا با مدار تحریک دیجیتال ترتیبی با ترکیبی از شمارنده‌های همگام و ناهمگام [۲۴]

همگام و ناهمگام را شامل شود یا شامل ماشین‌های حالت پیچیده با انواع و ابعاد مختلف باشد.

تشخیص تروهایی با مدار تحریک ترتیبی با استفاده از تولید بردارهای آزمون، به مراتب سخت‌تر از آنهایی است که از مدار تحریک ترکیبی استفاده می‌کنند. چراکه لازم است رشته ای از رخدادهای نادر را ایجاد کنیم تا تروا فعال شود. دسته‌ای دیگر از تروها، آنهایی هستند که مکانیزم تحریکشان آنالوگ است. در این تروها، حسگرهای روی تراشه موجب تحریک تروا می‌شوند. برای مثال در بعضی پژوهشها افزایش فعالیت مدار و در پی آن بالارفتن دمای تراشه به عنوان عامل تحریک تروا در نظر گرفته شده‌است [۲۵].



شکل ۲ـ۶: تروا با مدار بار از نوع آنالوگ [۲۴]

۲ـ۳ـ۲ مدار بار

تروا‌ها را می‌توان براساس مدار بارشان تقسیم‌بندی نمود. مدار بار می‌تواند آنالوگ یا دیجیتال باشد. تروا‌های دیجیتال می‌توانند مقادیر دیجیتال گره‌هایی از مدار را تغییر دهند یا محتویات بخشی از حافظه را دستکاری کنند. در مقابل، تروا‌های با مدار بار آنالوگ ، پارامترهای مدار مانند کارایی، توان مصرفی و حاشیه نویز را دستکاری می‌کنند. برای مثال در شکل ۲ـ۶ با اضافه کردن خازن بار، تاخیر مسیر تغییر داده شده‌است. نوع دیگری از مدار بار آنالوگ، مداری است که موجب افزایش فعالیت سوئیچینگ شود تا بدین وسیله فرآیند کهولت مدار سرعت گیرد و زودتر از کاربیافتد. علاوه بر تروا‌هایی که گفته شد، بعضی تروا‌ها ممکن است حملات مبتنی بر نرم‌افزار را تسهیل کنند. از جمله حملات نرم‌افزاری می‌توان به تغییر سطح دسترسی، ایجاد درِ مخفی ورود به سیستم، و سرقت رمز عبور اشاره کرد.

فصل ۳

کارهای پیشین

در این فصل کارهای پیشین انجام‌شده روی مسئله به تفصیل توضیح داده می‌شود.

۳ـ۱ تشخیص تروا‌های سخت‌افزاری

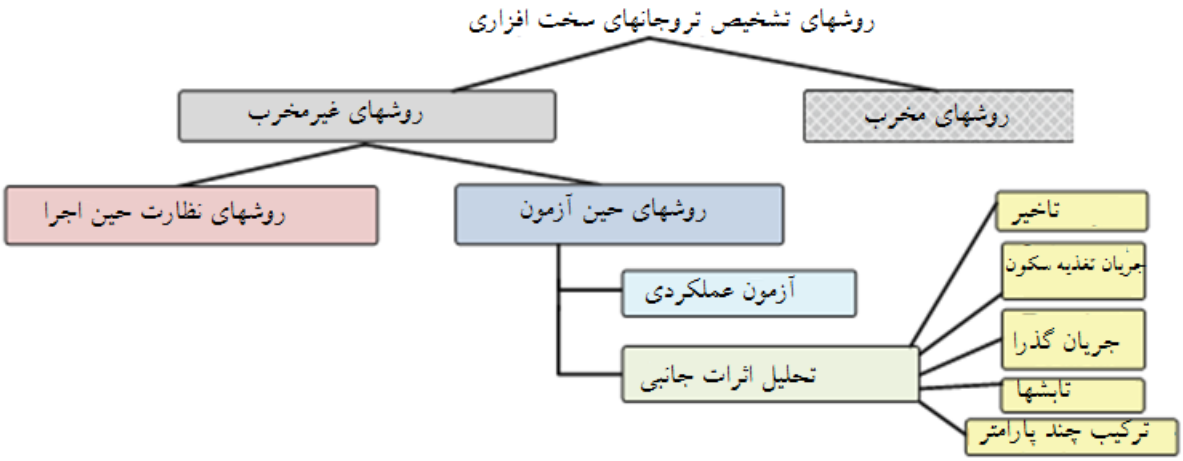
واگذار کردن طراحی و ساخت تراشه‌ها به شرکتها و کارخانجات خارجی در کنار افزایش استفاده از هسته‌های IP دیگر شرکتها و ابزارهای خودکار سازی طراحی که سایرین طراحی کرده‌اند، موجب شده‌است تا مدارات مجتمع نسبت به حملات تروا‌های سخت‌افزاری آسیب پذیرتر شوند. در بخش گذشته انواع تروا‌های سخت‌افزاری معرفی شدند. در این بخش انواع روش‌های تشخیص تروا را مرور می‌کنیم. روش‌های مختلف شرح داده می‌شوند و براساس قابلیت‌ها و محدودیت‌هایشان مقایسه می‌شوند. متاسفانه استفاده از روش‌های آزمون مدارات که در گذشته ارائه شده‌اند، برای تشخیص تروا مفید نیست. چرا که این روش‌ها در پی تشخیص عملکرد نادرست ناشی از اشکالات فرآیند ساخت هستند و عملکردهای اضافی ناشی از تروا را تشخیص نمی‌دهند. تروا‌هایی که به صورت هوشمندانه در مدار اضافه می‌شوند، معمولاً به ندرت تحریک می‌شوند و بنابراین تشخیص آنها سخت است. همچنین ممکن است برخی تروا‌ها هیچ اثری بر عملکرد عادی مدار نداشته باشند و برای مثال تنها اطلاعاتی را به خارج از مدار منتقل کنند [۲۶]. تولید بردارهای آزمونی که تمام نقاط مدار را به طور کامل بیازمایند، واقعاً شدنی نیست. مخصوصاً اگر تروا از نوع مدار ترتیبی باشد که با رخداد ترتیب خاصی از وقایع نادر فعال می‌شود. از سوی دیگر می‌توان تروا‌ها را براساس اثر جانبی که بر توان مصرفی یا تاخیر می

گذارند شناسایی کرد. با این روش دیگر نیازی به فعالسازی کامل تروا و مشاهده اثر آن بر خروجی مدار نیست. از طرفی دقت چنین روش‌هایی برای تشخیص تروهای کوچک، به علت اثر سوئی که تغییرات فرآیند و نویز بر تغییر توان یا تاخیر دارد، خیلی بالا نخواهد بود. استفاده از روش آزمون عملکردی و تحلیل اثرات جانبی برای تشخیص همه تروها کافی نیست. بنابراین می‌توان از روش نظارت حین اجرا برای بهبود سطح اطمینان از وجود تروا، استفاده کرد. برای مثال تروای که اطلاعاتی را از یک تراشه رمزنگاری از طریق کانال بی سیم نشت می‌دهد، ممکن است توان گذرای زیادی را در برهه‌ای از زمان که بنا نیست ارتباطاتی انجام شود، مصرف کند. در این حالت استفاده از روش‌های نظارت در حین اجرا، مناسبتر است. ارزیابی میزان قابلیت اعتماد در هسته‌های IP به علت عدم وجود مدل مرجعی که با آن مقایسه شوند، مشکل تر است. در این موارد می‌توان از آزمون‌های عملکردی استفاده کرد. این روش تنها زمانی موثر خواهد بود که اطلاعاتی درباره شرایط تحریک تروا و اثر تروا داشته باشیم. روش دیگر استفاده از اعتبار سنجی صوری است. روش‌های تشخیص تروای که تا کنون ارائه شده‌اند قابلیت‌ها و محدودیت‌های خاص خود را دارند. اما تا کنون روشی کامل برای تشخیص تمامی انواع تروها با درجه اطمینان بالا ارائه نشده‌است. یک راه حل برای بالا بردن درجه اطمینان، ترکیب روش‌های مختلف با یکدیگر است. در ادامه ابتدا به دسته‌بندی روش‌های تشخیص تروهای سخت‌افزاری می پردازیم و بعد از بررسی چالش‌های پیش رو در روش‌های تشخیص تروا، مروری کلی بر انواع روش‌های ارائه شده داریم.

۳ـ۲ دسته‌بندی روش‌های تشخیص تروا

در شکل ۳ـ۱ نمودار دسته‌بندی روش‌های تشخیص تروا ارائه شده‌است. این روش‌ها به دو دسته عمده مخرب و غیرمخرب دسته‌بندی می‌شوند. در روش‌های مخرب لازم است تراشه‌های ساخته شده با روش CMP[1] لایه برداری شوند تا تصاویر لایه به لایه با استفاده از میکروسکوپ الکترونی پویشی، گرفته شود. در این روش، رویکرد مهندسی معکوس بالا به پایین استفاده می‌شود. این روش‌ها به شدت گران و زمان بر(چندین ماه [۲۸]) هستند. این روش برای ارزیابی تک تک تراشه‌ها اصلا مقرون به صرفه نیست. تنها مورد استفاده از این روش بدست آوردن یک مدل مرجع برای حذف اثر تغییرات فرآیند در روش‌های دیگر تشخیص تروا است. روش‌های غیرمخرب را می‌توان به دو دسته رویکردهای

[1] Chemical-Mechanical Planarisation



شکل ۳ـ۱: دسته‌بندی روش‌های تشخیص تروا‌های سخت‌افزاری [۲۷]

نظارت حین اجرا و رویکردهای زمان آزمون تقسیم کرد.شایان ذکر است که روش‌های نظارت حین اجرا معمولاً روش‌های هجومی هستند که بعضی از روش‌های طراحی برای امنیت DfS[۲] از آنها استفاده می‌کنند. این روش‌ها می‌توانند از افزونگی موجود در مدار استفاده کنند. به این نحو که از یک هسته با قابلیت پیکربندی مجدد در سیستم‌های چندهسته‌ای برای ممانعت از اثرگذاری مدار دارای تروا بهره گرفته می‌شود تا با وجود تروا، قابلیت اطمینان سیستم تضمین شود. دسته‌ای دیگر از روش‌ها برای سیستم‌های با ماموریت بحرانی، روش‌های خود تخریبی هستند که به صورت خارجی توسط کاربر یا داخلی توسط ناظر تروا، فعال می‌شوند. روش‌های حین آزمون نیز می‌توانند از مدارات DfS کمک بگیرند. این مدارات می‌توانند حساسیت روش‌های تشخیص تروا را افزایش دهند و یا پوشش دهی تشخیص تروا‌ها را بیشتر کنند. اگر سیگنال فعال کننده حالت آزمون، به راحتی قابل تشخیص باشد، طراح تروا می‌تواند ترتیبی دهد که با فعال شدن این سیگنال تروا غیر فعال شود. روش‌های حین آزمون را می‌توان به دو دسته کوچکتر روش‌های مبتنی بر آزمون عملکردی و روش‌های مبتنی بر تحلیل اثرات جانبی، تقسیم کرد. روش‌های آزمون عملکردی [۸, ۲۷] بر تولید بردارهای آزمون و اعمال آنها برای فعالسازی تروا و مشاهده نتایج مخرب آن در خروجی‌های مدار متمرکز هستند. روش کار مشابه آزمون‌های لازم برای یافتن اشکالات [۳] است. اما مدل‌های تروا بسیار متفاوت با مدل‌های اشکال هستند. تروا‌ها هوشمندانه

[۲] Design for Security
[۳] stuck-at

در مدار درج می‌شوند و در مواقع نادری تحریک می‌شوند. تعداد کل تروا‌های ممکن از یک نوع و اندازه خاص، تابع نمایی از تعداد دروازه‌های منطقی مدار است. همچنین در مورد تروا‌های ترتیبی که باید رشته ای از رخدادها به ترتیب رخ دهد تا آن را فعال کند، ممکن است در طول زمان آزمون نتیجه تروا مشاهده نگردد. بنابراین روش‌های تشخیص اشکال را نمی‌توان برای تشخیص تروا به کار گرفت. از سویی دیگر روش‌های تحلیل اثرات جانبی مبتنی بر این حقیقت هستند که هر نوع درج مدارات مخرب در تراشه بایستی به صورت تغییر بعضی از پارامترها در سیگنال‌های جانبی مثل جریان نشتی، جریان تغذیه سکون [۲۹, ۳۰, ۳۱]، توان پویا [۴, ۱۰, ۳۲, ۳۳, ۳۴]، مشخصات تاخیری [۷, ۸]، تابشهای الکترومغناطیس ناشی از فعالیت سیگنال‌ها [۴] یا ترکیبی از موارد پیش گفته [۳۵, ۳۶] ، خودش را نشان دهد. روش‌های بسیاری مبتنی بر رویکرد تحلیل اثرات جانبی ارائه شده‌است. مشکل اصلی آنها این است که نسبت به نویز محیط و تغییرات فرآیند حساس هستند. بنابراین مساله تشخیص تروا به عنوان یک رویداد آماری با هدف بیشینه کردن احتمال تشخیص و کمینه کردن احتمال تشخیص غلط مدنظر قرار می‌گیرد. تولید بردارهای آزمون می‌تواند نقش مهمی در روش‌های تحلیل اثرات جانبی بازی کند. بدین صورت که حساسیت تشخیص تروا را خصوصاً در مورد تروا‌های کوچک در مدارات SoC بزرگ، افزایش دهد. معمولاً تروا در فضاهای خالی layout[۴] درج می‌شود و با اهداف خرابکارانه این مدارات به یکدیگر سیم بندی می‌شوند. با تمام این اوصاف، رویکرد تحلیل اثرات جانبی نسبت به رویکرد آزمون منطقی این مزیت را دارد که لازم نیست برای تشخیص تروا آن را فعال کنیم. بنابراین این دسته روش‌ها برای تشخیص تروا‌هایی موثر هستند که موجب تغییر عملکرد مدار نمی‌شوند و هدفشان افشای اطلاعات محرمانه از طریق سیگنال‌های جانبی است.

۳ـ۲ـ۱ رویکردهای مبتنی بر آزمون منطقی

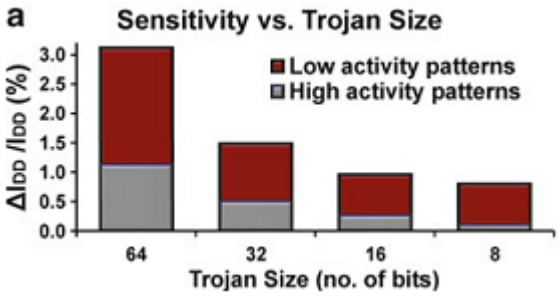
از آنجا که تشخیص تروا با آزمون عملکردی متفاوت با آزمون‌های رایج برای یافتن اشکالات مدار است، روش‌های آماری تولید بردار برای تشخیص تروا بسیار مناسبتر هستند. در [۹] روش تصادفی برای مقایسه احتمالاتی عملکرد مدار با مدل مرجع ارائه شده‌است. هر گونه تفاوت در عملکرد دو مدار برای تشخیص حضور تروا استفاده می‌شود و بردار ورودی که موجب این اختلاف شده‌است، اثر انگشت آن تروای خاص نامیده می‌شود. در این روش هیچ مدل خاصی برای تروا به منظور تولید بردارهای آزمون مدنظر قرار نگرفته‌است و صرفاً بر یافتن تساوی عملکرد دو مدار متمرکز است. این رویکرد برای تایید

[۴] چینش

اعتبار هسته‌های IP نیز کاربرد دارد. در [۳۷, ۳۸] یک روش تولید بردار آماری برای تشخیص تروا ارائه شده‌است که MERO نام دارد. در این روش مجموعه بهینه‌ای از بردارهای آزمون تولید می‌شود که هر گره با فعالیت کم در مدار را می‌تواند چندین بار به مقدار نادرش مقدار دهی کند. این روش شبیه روش آزمون N-Detect است [۳۹]. تعیین نادر بودن رخداد و تعداد گره‌های تحریک تروا و ماهیت تروا(ترکیبی یا ترتیبی) همگی متغیرهای ورودی به الگوریتم هستند. با فعالسازی تک به تک گره‌های نادر، احتمال تحریک تروای که با ترکیب نادری از این گره‌ها فعال می‌شود، بیشتر می‌شود. با این روش تحریک تروا از تشخیص آن ساده‌تر می‌شود. چراکه معمولاً مدار بار مربوط به تروا از مشاهده‌پذیری پایینی برخوردار است. با استفاده از دو معیار پوشش تحریک و پوشش تروا این الگوریتم ارزیابی شده‌است. هرچه تعداد دفعات مقداردهی به گره‌های نادر بیشتر شود، این دو معیار بهبود خواهد یافت. با این همه اینکار باعث افزایش زمان آزمون خواهد شد. همچنین در این روش برای مدارات ترتیبی، از فلیپ فلاپ‌های پویش برای کاهش مدت آزمون و افزایش پوشش‌دهی، استفاده شده‌است. علاوه بر روش‌های بالا، می‌توان از روش‌های DfS نیز برای افزایش قابلیت آزمون پذیری مدارات بزرگ با رویکرد بهبود کنترل‌پذیری و مشاهده‌پذیری گره‌هایی که محتمل است گره تحریک تروا یا گره بار باشند، استفاده کرد. با افزودن یک ماشین حالات کنترلی که تشخیص آن مشکل باشد، ماژولهای مختلف داخل مدار را می‌توان به صورت انتخابی آزمود. با اعمال رشته خاصی از ورودی‌ها، هر ماژول به وضعیت خاصی که وضعیتِ شفاف [۴۰] نام دارد، وارد می‌شود. در این وضعیت، سایر ماژولها به نحوی از چرخه عملکرد خارج می‌شوند و وجود یا عدم وجود تروا در این ماژول خاص بررسی می‌شود.

۳ـ۲ـ۲ روش‌های مبتنی بر تحلیل اثرات جانبی

تمامی روش‌های مبتنی بر تحلیل اثرات جانبی، بر مشاهده اثر تروا بر یک پارامتر فیزیکی مثل جریان تغذیه(گذرا یا دائمی)، توان مصرفی، یا تاخیر مسیرها، استوار هستند. مزیت استفاده از این روش‌ها در این است که حتی اگر مدار تروا در طول زمان آزمون، تاثیر قابل مشاهده‌ای در خروجی نگذارد، حضور مدار اضافه ناشی از تروا در پارامترهای جانبی قابل تشخیص است. مساله اصلی در این روش‌ها این است که در تکنولوژی‌های ابعاد نانو، اثرات جانبی با تغییر فرآیند به شدت تغییر می‌کنند و این امر در کنار نویز اندازه‌گیری، تشخیص تروا را بخصوص اگر مداری کوچک باشد، مشکل می‌کند. در ادامه انواع سیگنال‌هایی که در این روش‌ها به عنوان اثر جانبی از آنها استفاده می شود، مرور می‌شود و روش‌هایی که از این سیگنال‌ها استفاده کرده‌اند، شرح داده خواهد شد.



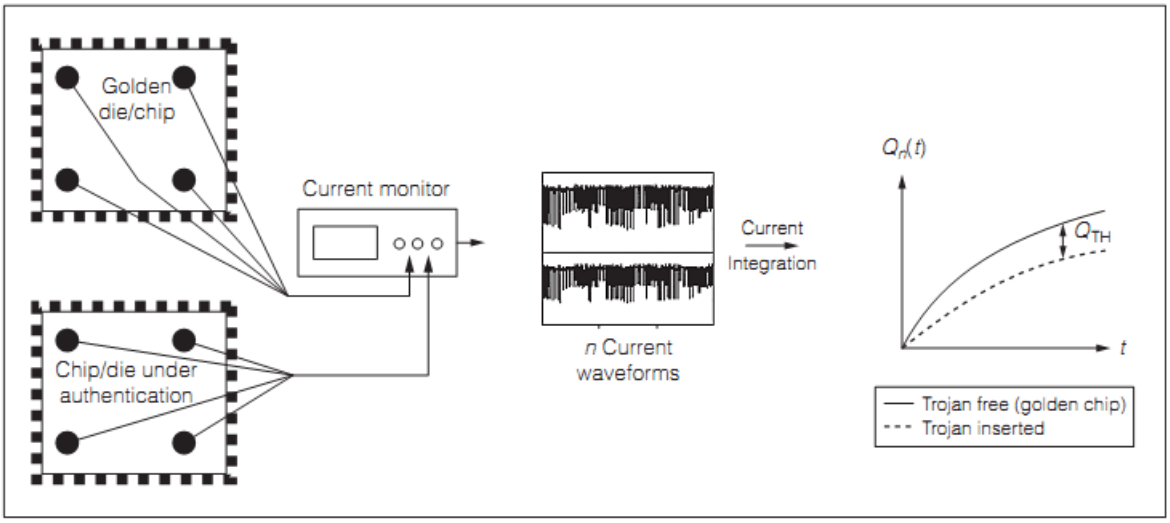
شکل ۳ـ۲: اثر اندازه تروا بر جریان نشتی و جریان گذرای تغذیه [۴۰]

IDDQ[۵] روشن است که جریانی که از منبع تغذیه کشیده می‌شود، با افزودن مدارات اضافه به مدار اصلی، تغییر خواهد کرد. هر دروازه منطقی اضافه باعث افزایش جریان نشتی می‌شود و اندازه‌گیری مجموع این جریان‌های اضافه، تشخیص تروا را تسهیل می‌کند. البته این افزایش ناشی از جریان نشتی دربرابر جریانی که یک مدار بزرگ چند میلیون گیتی از تغذیه می‌کشد، بسیار ناچیز است و تشخیص آن دشوار است. برای بالابردن احتمال تشخیص تروا، می‌توان جریان را از چندین پایه تغذیه اندازه‌گیری نمود. به این روش، روش مبتنی بر ناحیه گفته می‌شود.

IDDT[۶] جریان گذرای منبع تغذیه که نشانگر توان پویای ناشی از فعالیت سوئیچینگ مدار است، نیز از طریق پایه‌های تغذیه قابل اندازه‌گیری است. این جریان بیانگر تعداد دروازه‌های منطقی است که با اعمال بردار خاصی به ورودی، خروجی‌شان تغییر می‌کند. بنابراین اعمال بردارهای ورودی و اندازه‌گیری جریان گذرا می‌تواند حساسیت روش‌های تشخیص تروا را به وجود تروا افزایش دهد. همانطور که در شکل ۳ـ۲ نشان داده شده‌است، حساسیت در تشخیص تروا با کاهش اندازه تروا، کاهش می یابد. اما با انتخاب مناسب بردارهای ورودی، این حساسیت قابل افزایش است. این مساله مقیاس پذیری این روش را بهبود می بخشد. تاخیر: پارامتر سومی که می‌تواند در تشخیص تروا استفاده شود، تاخیر مسیرهاست. اگر تروا در مسیری باشد که تاخیرش اندازه‌گیری می‌شود، بسته به تعداد دروازه‌های منطقی که در مسیر اضافه شده‌است، تاخیر مسیر را افزایش خواهد داد. حتی اگر بردار ورودی اعمال شده، تروا را بطور کامل فعال نکند، مقداری خازن بار به گره اضافه می‌کند و بنابراین باعث افزایش تاخیر می‌شود. البته اگر تاخیر مسیر اصلی خیلی زیاد باشد، این تغییرات کوچک ممکن است به چشم نیاید و تغییرات ناشی از تغییر فرآیند، آن را بپوشاند. لازم به ذکر است که تنها تاخیر مسیرهایی که از

[۵] Integrated Dual Disorder Quiescent
[۶] Integrated Dual Disorder Treatment

ورودی‌ها شروع شده و به خروجی‌ها ختم می‌شوند، قابل اندازه‌گیری است. بنابراین برای مدارات ترتیبی اگر از پویش کامل استفاده نشود، زمان زیادی باید صرف اندازه‌گیری همه مسیرها شود. تشعشعات الکترومغناطیس: تابشهای الکترومغناطیس ناشی از فعالیت سوئیچینگ دروازه‌های منطقی مختلف، می‌تواند جهت تشخیص وجود مدار اضافی ناشی از تروا مورد مشاهده قرار گیرد. روش‌هایی که از این رویکرد استفاده می‌کنند در همان دسته‌ای قرار می‌گیرند که روش‌های استفاده کننده از جریان گذرا هستند. روش‌های موجود سعی می‌کنند با نرمال سازی یا تخمین گوشه‌های فرآیند، اثر تغییر فرآیند را بر تغییر پارامتر اندازه‌گیری شده، مدل کنند و بدین ترتیب اثر آنها را حذف نمایند. با استفاده از روش مبتنی بر ناحیه، که از چندین پایه تغذیه جداگانه اندازه‌گیری را انجام می‌دهد، نویزهایی که متناسب با مقدار اندازه‌گیری شده هستند (مثل نویز تغییر فرآیند) کاهش می یابد. می‌توان از پردازش سیگنال آماری برای محاسبه نویز فرآیند و کاهش اثر آن بر مقدار اندازه‌گیری شده استفاده کرد. روش اثرانگشت برداری از مدار [۴] برای کالیبراسیون نویز فرآیند استفاده می‌شود و برای تشخیص بخش‌هایی از گزارش توان مصرفی که حضور تروا را نشان می‌دهد، نیز استفاده شده‌است. با استفاده از این روش می‌توان تروهایی با ابعاد ۰,۰۱ درصد ابعاد مدار را شناسایی کرد. برای افزایش احتمال فعال شدن تروا با اعمال بردارهای آزمون به ورودی‌ها، روش تولید بردار آزمون مناسب باید انتخاب شود. برای مدارات ترتیبی بزرگ، روش فعالسازی مبتنی بر ناحیه مدار برای افزایش حساسیت روش‌های مبتنی بر اثرات جانبی موثر خواهد بود. مدار را می‌توان به بخش‌هایی که از نظر عملکردی از هم جدا هستند، تقسیم‌بندی کرد یا به صورت ساختاری به نحوی تقسیم‌بندی کرد که همپوشانی نواحی کمینه باشد. بعد از آن بردارهای آزمونی که به صورت هدایت شده تولید شده‌اند، برای افزایش فعالیت سوئیچینگ در ناحیه مورد نظر و با هدف کاهش فعالیت در سایر مدار، اعمال می‌شوند. بنابراین حساسیت روش تشخیص تروا در ناحیه فعال، افزایش می یابد. کارایی این روش در [۱۰] نشان داده شده‌است. روش بردار پایدار شده نیز برای افزایش بیشتر حساسیت در تشخیص تروا استفاده می‌شود. در این روش بردارهای اعمالی به ورودی‌ها برای چندین پالس ساعت ثابت نگهداشته می‌شوند تا تنها فعالیت ناشی از تغییر مقادیر عناصر حافظه ای وجود داشته باشد و بدین ترتیب جریان مدار اصلی تا حدامکان کاهش یابد. این روش در [۳۲] شرح داده شده‌است و برای بزرگ‌نمایی تفاوت بین توان مصرفی مدار اصلی و مدار دارای تروا استفاده شده‌است. روش دیگر مبتنی بر ناحیه برای کالیبراسیون نویز فرآیند در قالب شبکه تغذیه روی تراشه در [۵, ۲۹] ارائه شده‌است. نوعاً مدارات، یک شبکه تغذیه توزیع شده در لایه‌های فلزی بالا دارند که شامل برآمدگی‌هایی است که به پایه‌های مختلف تراشه متصل است. از بیرون تراشه و در



شکل ۳ـ۳: نحوه اعمال روش اندازه‌گیری جریان و بار به صورت محلی شده [۲۹]

سطح بورد، این پایه‌ها می‌توانند به یک منبع تغذیه یکپارچه متصل شده باشند. اندازه‌گیری جریان از پایه‌های مختلف تغذیه به ازای ورودی‌های مختلف، و در ادامه انتگرال‌گیری روی جریان، می‌تواند برای اندازه‌گیری انتقال بار الکتریکی در طول فعالیت سوئیچینگ به کار رود. هر مداری که دارای تروا باشد، بار بیشتری در واحد زمان جمع می‌کند. چراکه نسبت به مدار بدون تروا فعالیت سوئیچینگ بیشتری دارد. این اختلاف بار را با انتگرال‌گیری از جریان می‌توان تشخیص داد. همچنین از آنجا که اندازه‌گیری جریان از پایه‌های مختلف انجام می‌شود، موقعیت تروا نیز قابل تخمین است. شکل ۳ـ۳ نحوه اعمال این روش را نشان می‌دهد. روش‌های مختلف کالیبراسیون برای حذف تغییرات مقاومتی در پایه‌های تغذیه و حذف تغییرات فرآیند درون تراشه و بین تراشه قابل استفاده است. یک مدار کالیبراسیون شامل یک ترانزیستور است که بین پایه تغذیه و زمین وصل می‌شود و می‌تواند با استفاده از سیگنال کنترلی از یک فلیپ فلاپ، خاموش یا روشن شود. تکنیک ارائه شده در این مقاله می‌تواند ۵۰ درصد تروهای فعال شده و ۳۰ درصد تروهای غیرفعال را تشخیص دهد. برای افزایش احتمال فعال شدن سیگنال‌های نادر، از فلیپ فلاپ‌های پویش استفاده شده‌است. یک روش تغییر ترتیب سلولهای پویش آگاه به چینش [۳۳] می‌تواند کنترل‌پذیری سوئیچینگ در نواحی خاص را افزایش دهد. این روش احتمال سوئیچینگ مدارات تروا را افزایش می‌دهد و در نتیجه موجب افزایش احتمال تشخیص تروا می گردد. اندازه‌گیری تاخیر مسیرها در پایه‌های خروجی برای مجموعه‌ای از بردارهای آزمون، می‌تواند برای تشخیص حضور

تروا مورد استفاده قرار گیرد. در [۷] نشان داده شده‌است که حجم چنین اطلاعاتی راجع به تاخیرها که از مسیرهای مختلف مدار بدست می آید، بخصوص برای مدارات بزرگ، می‌تواند بسیار زیاد شود. روش‌های فشرده‌سازی اطلاعات مثل روش PCA[۷] را می‌توان برای کاهش ابعاد این اطلاعات بکار برد. این نقاط داده‌ای کاهش یافته را اثرانگشت تاخیر مسیر می‌نامند. روش سریع مشخصه‌سازی تاخیر [۸] ثبات‌های سایه را بهمراه مقایسه کننده درون تراشه قرار می‌دهد تا تاخیر مسیرهای داخلی ثبات تا ثبات را بیابد. این روش DfS از افزایش انحراف منفی کلاک ثبات سایه نسبت به کلاک عملیاتی سیستم استفاده می‌کند و نتیجه مقایسه را به ازای یک سری ورودی، ذخیره می‌کند تا توزیع تاخیر را بیابد. روش دیگر برای مشاهده اثر تروا بر تاخیر مسیرهای داخلی، این است که مسیرها را به صورت نوسانگرهای حلقوی [۴۱, ۴۲] پیکربندی کنیم. با درج تروا در این مسیرها، تاخیر مسیر، به علت تغییر مقاومت و یا بخاطر همشنوایی بین سیم‌ها، تغییر خواهد کرد. فرکانس این نوسانگرها با استفاده از شمارنده‌های روی تراشه محاسبه می‌شود. مساله اصلی در این روش حذف اثر تغییر فرآیند و تغییرات محیطی است. البته طراح تروا ممکن است به نحوی تروا را اضافه کند که اثری بر فرکانس نوسان نگذارد یا اینکه مدارِ شمارنده فرکانس را به نحوی دچار مشکل کند. یک نوسانگر حلقوی را می‌توان به عنوان ناظر حرارتی برای کالیبراسیون تاخیر ناشی از دما نیز بکار برد. این روش باید با روش تولید بردارهای آزمون ترکیب شود تا به پوشش دهی بالا و زمان آزمون پایین دست یابیم. این روش را می‌توان در زمان اجرا نیز به کار گرفت.

در [۳۱] از هر دو پارامتر تاخیر مسیر و جریان نشتی برای پیاده‌سازی یک روش تشخیص تروا در سطح دروازه‌های منطقی استفاده شده‌است. مساله تشخیص تروا را می‌توان به صورت یک مساله برنامه نویسی خطی فرمول بندی کرد به صورتی که تغییرات فرآیند هر دروازه منطقی به صورت یک ضریب ثابت برای جریان نشتی یا تاخیرش لحاظ شود. این روش توانایی تشخیص حتی یک دروازه منطقی اضافه را دارد

۳ـ۲ـ۳ رویکردهای نظارت زمان اجرا

تشخیص کامل تروهای با انواع و ابعاد مختلف در زمان آزمون تراشه از نظر عملی غیرممکن است. نظارتِ برخط محاسبات بحرانی می‌تواند سطح اعتماد به مدار را به شدت بالا ببرد. این روش‌ها می‌توانند

[۷] Principal Component Analysis

در هنگام تشخیص موارد اشکال در مدار، تراشه را غیرفعال کنند یا آن را دور بزنند و امکان عملیات قابل اطمینان را فراهم کنند. یکی از روش‌های نظارت زمان اجرا مبتنی بر اضافه کردن مدارات با قابلیت بازپیکربندی است که به آنها DEFENSE یا طراحی برای فعالسازی امنیت [۱۴] می‌گویند. چک کردن عملیاتی می‌تواند به صورت همروند با عملیات عادی سیستم انجام شود و در صورت بروز اختلاف با عملیات عادی، شمارنده‌های متناسب را تحریک می‌کند. معمولا هسته با قابلیت بازپیکربندی، عملیات مداری را که به درستی عمل نمی‌کند، پیاده‌سازی می نماید و مدار دارای مشکل غیرفعال می‌شود یا دورزده می‌شود. یک رویکرد ترکیبی سخت‌افزاری و نرم‌افزاری برای نظارت زمان اجرا در [۴۳] ارائه شده‌است. یک ماژول ساده به نام حصار سخت‌افزاری که خارج از CPU است مدنظر قرار می‌گیرد. تروا‌هایی که در اینجا مدنظر هستند، از نوع رد خدمات می باشند. با استفاده از چک کردن دوره‌ای توسط سیستم عامل وجود یا عدم وجود تروا‌ها بررسی می‌شود. این روش تنها ۲/۲ درصد سربار کارایی به سیستم تحمیل می‌کند. در [۴۴] یک روش ترکیبی سخت‌افزاری/نرم‌افزاری به نام BlueChip ارائه شده‌است که شامل مولفه‌های زمان طراحی و مولفه‌های زمان اجراست. در این روش تشخیص مدارات بلااستفاده، با استفاده از آزمون‌های اعتبارسنجی انجام می‌شود و به عنوان مشکوک برچسب می خورند. در طول زمان اجرا مدارات مشکوک حذف می‌شوند و با یک مدار تشخیص استثنا جایگزین می‌شوند. با این کار مدار می‌تواند عملیات خود را بدون مشکل انجام دهد. این روش برای از بین بردن اثر تروا‌های سخت‌افزاری است که اهدافی شبیه تروا‌های نرم‌افزاری دارند. این تروا‌ها هدفهای مختلفی دارند. از جمله افزایش امتیاز یک برنامه از حالت کاربر معمولی به حالت فوق کاربر، تخصیص دسترسی به حافظه محدود شده یا شروع حملات DoS که شبیه به فراهم کردن امکان اجرای کد مخرب است. در مورد پردازنده‌های چندهسته‌ای می‌توان یک روش زمان اجرای خودزمانبند پیاده‌سازی نمود [۴۵] که به وسیله آن، نرم‌افزاری‌هایی با عملیات مشابه روی چندین هسته اجرا می‌شوند. خروجی‌های هسته‌های مختلف با یکدیگر مقایسه می‌شود تا به صورت پویا سطح اطمینان تک تک هسته‌ها چک شود.

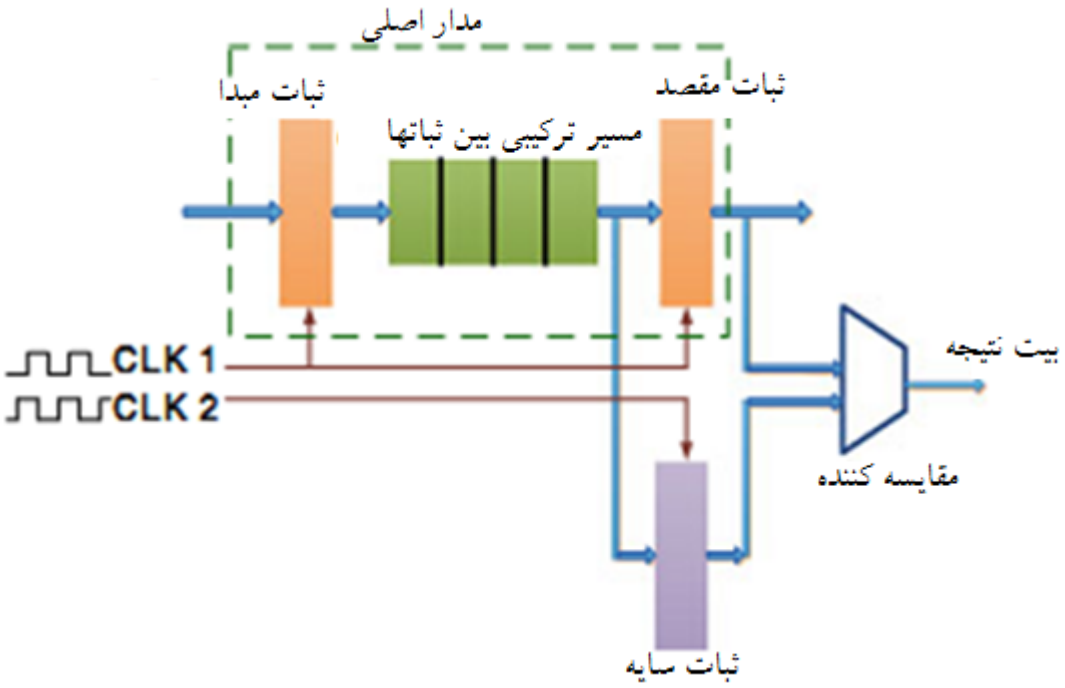
۳ـ۲ـ۴ روش‌های طراحی مطمئن

برای بهبود کارایی روش‌های تشخیص تروا و رفع محدودیت‌های آنها، روش‌های متعددی توسط محققان امنیت سخت‌افزاری ارائه شده‌است که هدف آنها تغییر روال طراحی کنونی است. به این روش‌ها، روش‌های طراحی برای اطمینان سخت‌افزاری می گویند [۳]. هدف از این روش‌های بازدارنده تروا، این است که مانع درج تروا شوند و یا تشخیص تروا‌ها را تسهیل کنند. برخلاف روش‌های تشخیص تروا

که روش‌های منفعلانه هستند، روش‌های طراحی مطمئن، روش‌هایی فعال هستند. یعنی ساختار مدار را به نحوی تغییر می‌دهند تا مانع از درج تروا شوند. برای رسیدن به این هدف، چرخه طراحی مدارات مجتمع باید دستخوش تغییر شود. اکثر روش‌های بازدارنده از تروا، با هدف تسهیل در تشخیص تروا با استفاده از روش‌های تحلیل اثرات جانبی ارائه شده‌اند. از این پس به این روش‌ها، روش‌های مبتنی بر اثرانگشت اثرات جانبی گفته می‌شود. بعضی از این روش‌ها صرفاً امکاناتی برای اندازه‌گیری پارامترهای جانبی فراهم می‌کنند ولی برخی دیگر مقادیر اندازه‌گیری شده را با مقادیر آستانه‌ای که از قبل تعریف شده‌اند، مقایسه می‌کنند. سربار طراحی، عمده ترین چالش این روش‌هاست. چراکه مدارات اندازه‌گیری و مقایسه، به نوبه خود می‌توانند پیچیده باشند و بخش زیادی از مساحت تراشه را اشغال کنند. بر همین اساس اکثر روش‌های بازدارنده از تروا برای اندازه‌گیری و مقایسه تاخیرها استفاده می‌شوند. چرا که مدارات اندازه‌گیری تاخیر نسبتاً سربار مساحت کمتری دارند. سایر روش‌ها بر این فرض استوار هستند که طراحان تروا تنها از رخدادهای نادر برای تحریک تروا استفاده می‌کنند. این روش‌ها سعی دارند با افزایش احتمال فعال شدن کامل تروا‌ها در طول فاز آزمون، روش‌های آزمون ساختاری/عملکردی را بهبود دهند. از میان این روش‌ها، روش مبهم سازی [۱۸] به نوعی مدار را پنهان می‌کند تا حمله‌کننده نتواند احتمال واقعی رخدادها را حساب کند و براساس آن در انتخاب محل درج تروا به خطا رود. در مقابل روش درج فلیپ فلاپ‌های پویش [۱۶] و روش ولتاژ معکوس [۱۷] با هدف متعادل کردن فرکانس گذار سیگنال‌های داخلی برای حذف رخدادهای نادر، ارائه شده‌اند. روش‌های دیگری هستند که برای اینکه بر چنین فرضیاتی استوار نباشند، از روش طراحی برای آزمون تروا (DFTT) که در [۱۹] ارائه شده‌است استفاده می‌کنند. در نهایت روش‌هایی که هدفشان حفاظت از هاIP است، نیز در این بخش مورد بررسی قرار خواهند گرفت. در [۲۰, ۲۱] مفهوم سخت‌افزار حامل اثبات (PCH) ارائه شده‌است که مبتنی بر روش حفاظت نرم‌افزاری کد حامل اثبات (PCC) است.

روش ثبات‌های سایه

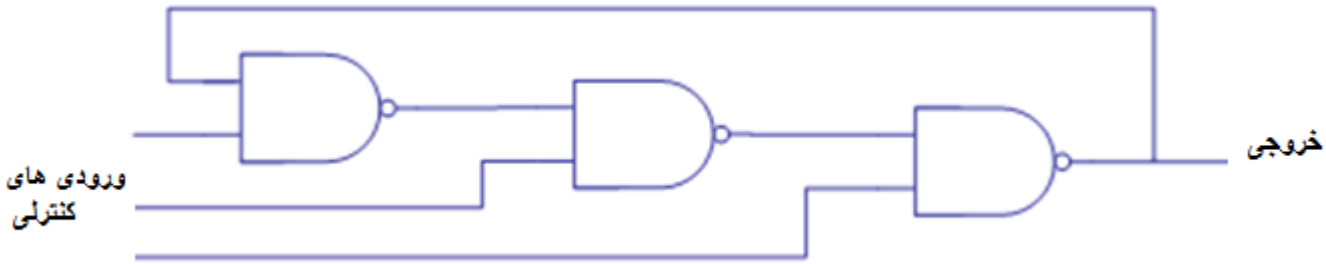
ایده استفاده از ثبات‌های سایه اولین بار در [۸] ارائه شد و بعد از آن در [۲۷] مورد ارزیابی قرار گرفت. روش تشخیص تروا براساس اثرانگشت تاخیر مسیر، اولین بار در [۷] ارائه شد. در این مقاله نویسندگان نشان داده اند که با کمک تحلیل داده آماری این روش می‌تواند تروا‌های سخت‌افزاری با اندازه ۰,۲ درصد مساحت تراشه را تشخیص دهد. مساله پیش روی این روش مشکل بودن اندازه‌گیری تاخیر مسیرهای داخلی است. ایده ثبات‌های سایه راه حل این مشکل را ارائه داده است. شکل ۳ـ۴ معماری پایه این



شکل ۳ـ۴: معماری پایه محاسبه تاخیر مسیرهای داخلی با استفاده از ثبات سایه [۷]

روش را نشان می‌دهد.

این معماری شامل یک ثبات سایه، یک مقایسه‌گر و یک ثبات نتیجه است. ثبات سایه با پالس ساعتی متفاوت با ساعت سیستم کار می‌کند. ساعت این ثبات، یک سیگنال است که از انحراف منفی در پالس ساعت سیستم حاصل می‌شود. در واقع فاز پالس ساعت ثبات سایه قابل تنظیم است. برای اندازه‌گیری تاخیر مسیرهای میانی، در ابتدا فاز ساعت سایه همان فاز ساعت سیستم است. بنابراین مقادیر درون ثبات سایه و ثبات مقصد مشابه هستند و خروجی مقایسه‌گر صفر است. سپس فاز ساعت سایه کاهش می یابد تا جایی که خروجی مقایسه‌گر یک شود و مقدار ۱ در ثبات نتیجه ذخیره می‌شود. این مقادیر از طریق زنجیره پویش، خوانده می‌شوند. معایب این روش: ۱ـ گام تغییر فاز پالس ساعت سایه در دقت اندازه‌گیری تاخیر نقش اساسی دارد و تعیین کننده کارایی این روش است. استفاده از تولیدکننده سیگنال با دقت بالا نیز مساحت و توان مصرفی زیادی را تحمیل می‌کند. ۲ـ وجود تغییرات فرآیند و نویز اندازه‌گیری می‌تواند دقت نتایج را کم کند. راه حل رایج حل این مشکل، استفاده از تحلیل داده آماری برای حذف اثر این دو است. ۳ـ با افزایش ابعاد مدار اصلی، تعداد مسیرهای داخلی نیز افزایش یافته و بردارهای آزمون بیشتری برای پوشش این مسیرها و بهبود پوشش آزمون، باید اعمال شود. این

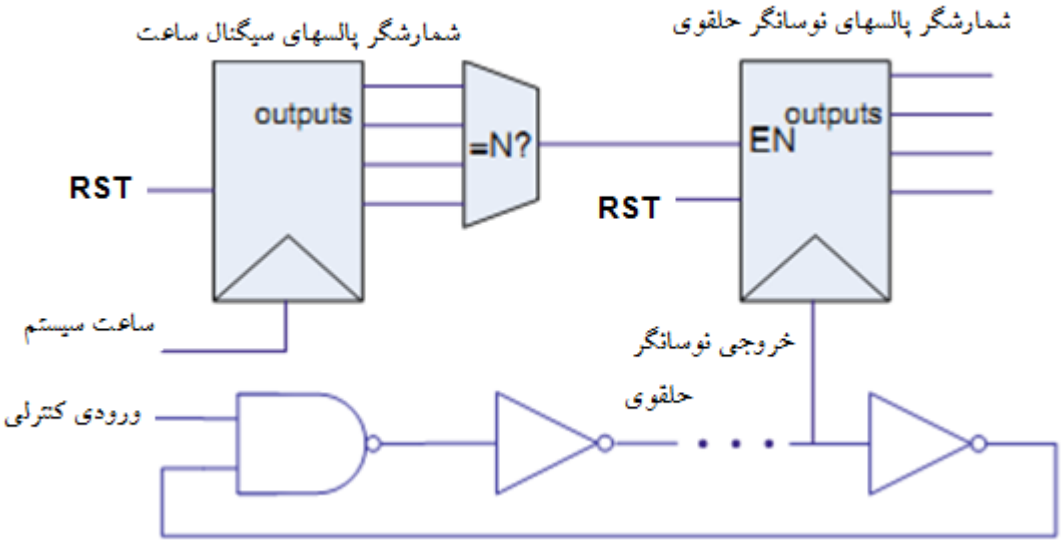


شکل ۳ـ۵: مدار نوسانگر حلقوی ساده [۴۱]

امر هزینه را بالا می برد. ۴ ـ هرچه تعداد مسیرهای داخلی بیشتر شود، ثبات‌های سایه بیشتری نیاز است و شبکه سیگنال ساعت ثبات‌های سایه باید گسترده تر شود. بنابراین هم سربار مساحت خواهیم داشت و هم کارایی ممکن است کاهش یابد.

روش استفاده از نوسانگرهای حلقوی

به منظور کاهش هزینه آزمون روش ثبات‌های سایه، در عین استفاده از تاخیر مسیرها، بعضی محققان به این فکر افتاده‌اند که بجای اندازه‌گیری مسیرهای موجود، مسیرهای جدیدی ایجاد کنند و تاخیر آنها را اندازه‌گیری کنند. روش استفاده از نوسانگرهای حلقوی از رایج ترین این روش‌هاست [۴۱, ۴۲]. چرا که مساحت آنها بسیار کمتر از سایر معماری‌های بازدارنده از تروا است و از طرفی از آنجا که معماری نوسانگر حلقوی بسیار ساده است، درج آنها اثر بسیار کمی بر طراحی اولیه می‌گذارد. شکل ۳ـ۵ یک مدار نوسانگر حلقوی ساده را نشان می‌دهد. استفاده از دروازه‌های NAND به جای NOT کنترل‌پذیری نوسانگر را بهبود می‌دهد. تنها وقتی هر سه سیگنال کنترلی فعال باشد، نوسان انجام می‌شود و در حالت عادی این مدار سربار توان مصرفی تحمیل نمی‌کند. علاوه بر این وجود این سیگنال‌های کنترلی باعث می‌شود طراحان تروا متوجه جزئیات مدار نشوند. ایده اصلی پشت این روش، این است که هر تغییری در مدار اولیه پارامترهای نوسانگر حلقوی را نیز تغییر خواهد داد. مساله مطرح در این روش این است که چند نوسانگر باید در مدار درج شود و کجا باید اینکار انجام شود؟ روش دیگری که از ایده نوسانگرهای حلقوی استفاده می‌کند، به جای درج این نوسانگرها در مکان‌های مختلف، با استفاده

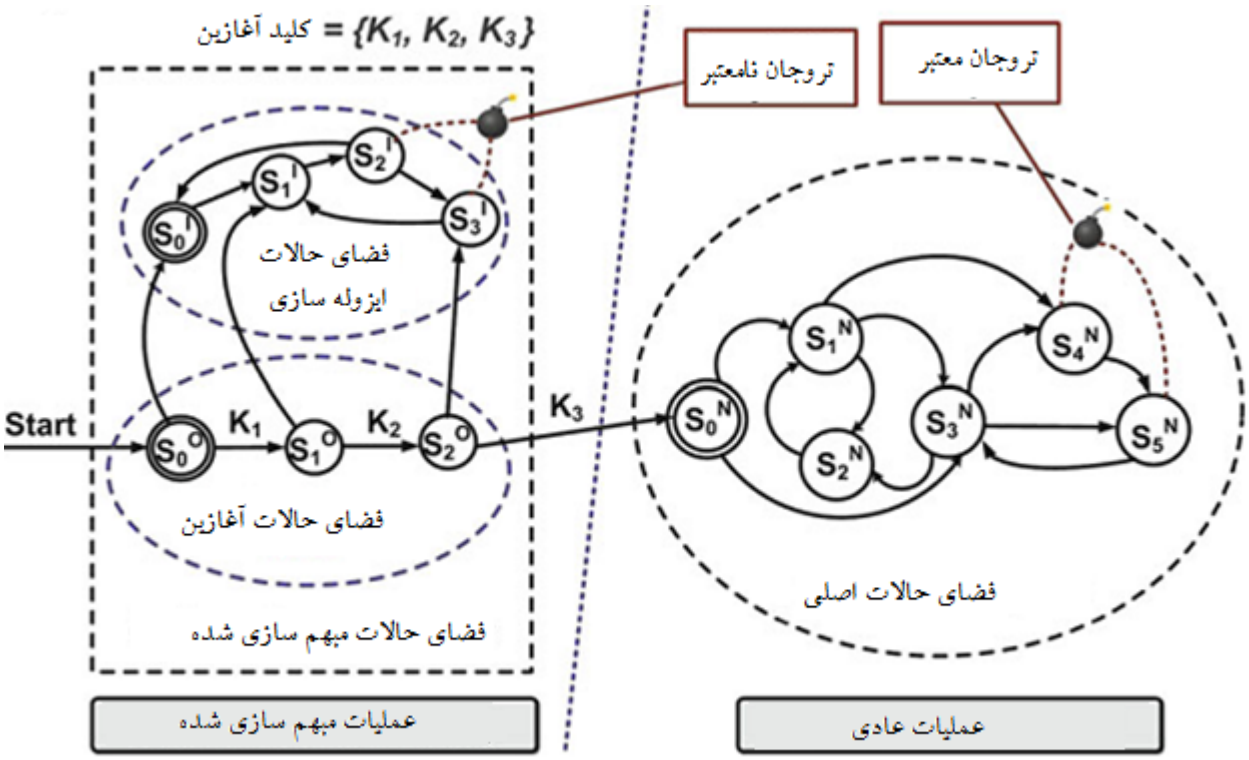


شکل ۳ـ۶: پیمانه اندازه‌گیری فرکانس [۴۲]

از دروازه‌های منطقیِ داخل مدار اصلی و با افزودن مالتی پلکسرها، دروازه‌های NAND و معکوس کننده‌ها، نوسانگرهای حلقوی را می سازد. این روش حساسیت روش تشخیص تروا را بیشتر می‌کند. در بدترین حالت اضافه کردن مدار تروا ممکن است نوسانگر را خاموش کند. وقتی مدار کوچک باشد، ساختن نوسانگرهای حلقوی ساده است. اما برای مدارهای پیچیده‌تر طراحان باید از الگوریتم‌هایی استفاده کنند که فرآیند درج نوسانگرها را خودکار انجام دهند. مساله دیگر نحوه محاسبه و اندازه‌گیری فرکانس نوسانگرهاست که به عنوان نشانه‌ای از تغییر تاخیر مسیرها مدنظر قرار می گیرد. اغلب برای اینکار از ماژول‌های اندازه‌گیری فرکانس روی خود تراشه استفاده می‌شود. شکل ۳ـ۶ یک نمونه از این ماژول‌ها را نشان می‌دهد. با شروع به کار مدار هر دو شمارنده شروع به شمارش می‌کنند. اولی با فرکانس مدار و دومی با فرکانس نوسانگر حلقوی. وقتی خروجی شمارنده اول برابر N شود، شمارنده دوم از کار می افتد و نسبت خروجی این شمارنده‌ها نشان دهنده فرکانس نوسانگر است. این روش سربار مساحت و توان مصرفی را افزایش می‌دهد.

۳ـ۲ـ۵ روش‌های مبتنی بر حذف رخدادهای نادر

در [۱۱] با فرض اینکه طراح تروا صرفاً از رخدادهای نادر برای تحریک تروا استفاده می‌کند، ایده بردارهای تروا را ارائه کرده‌اند. این بردارها رخدادهای با فرکانس پایین را تحریک می‌کنند تا بدین

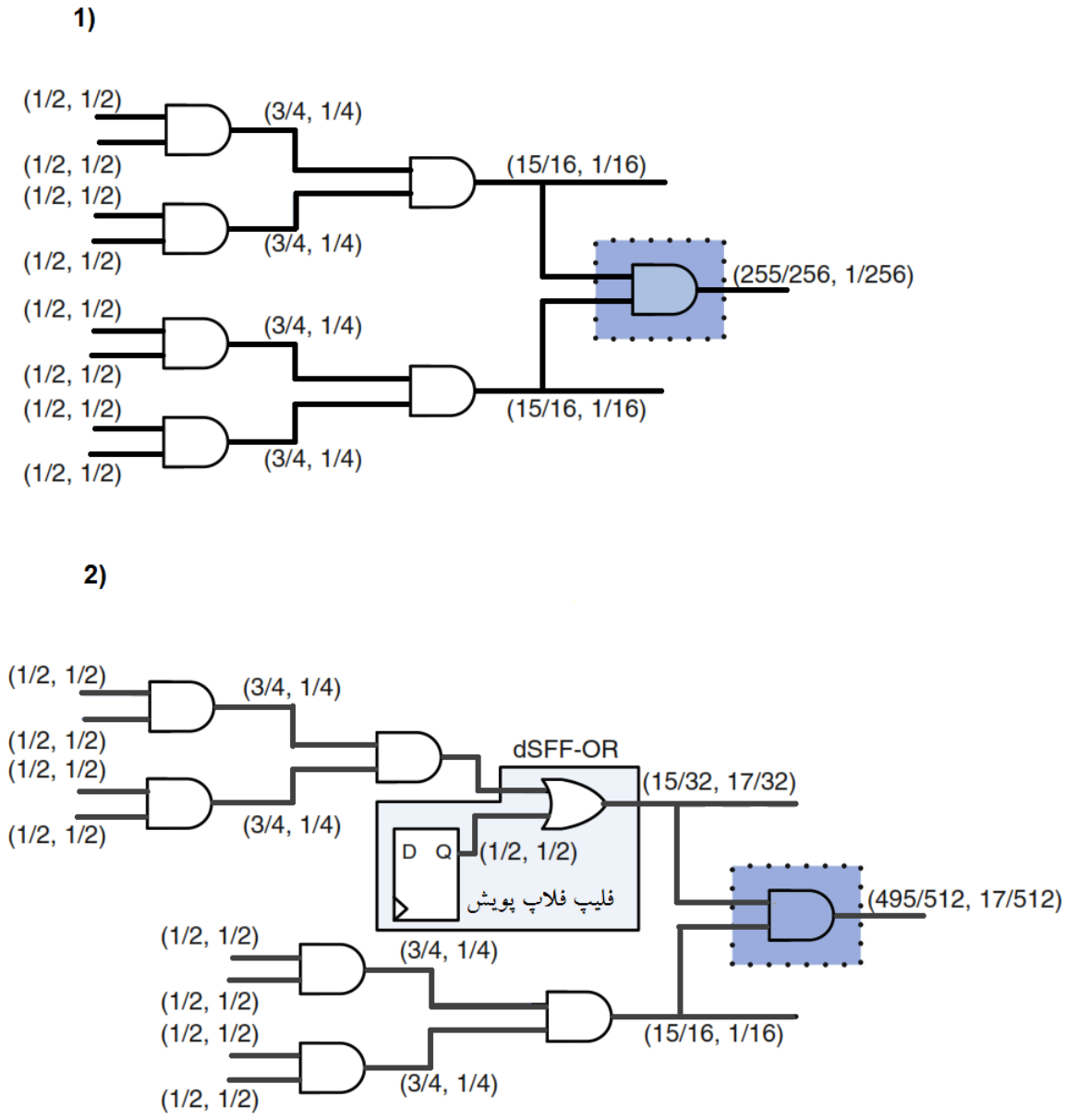


شکل ۳ـ۷: مدار مبهم‌سازی‌شده که شامل مدار اصلی می باشد [۱۸]

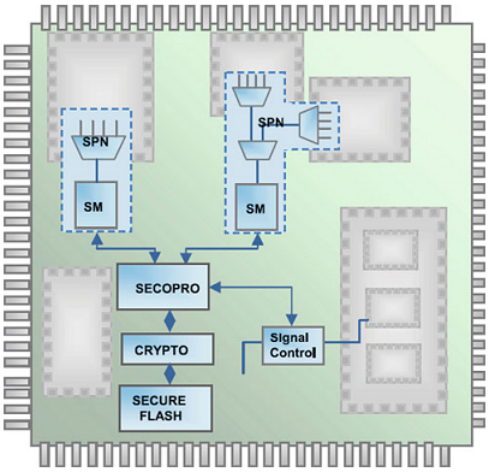
وسیله قابلیت تشخیص در روش‌های آزمون ساختاری قبلی، بهبود یابد. روش مبهم سازی طراحی [۱۸]، فلیپ فلاپ‌های پویش [۱۶] و روش معکوس سازی ولتاژ [۱۷]، همگی بر این فرضیه استوار هستند و روش‌های بازدارنده ساختاری/عملکردی محسوب می‌شوند. مبهم سازی طراحی به معنی این است که طراحی به نحوی تغییر داده شود که از نظر عملیاتی مشابه طرح اولیه باشد ولی فهم منطق درون آن برای طراح تروا سخت‌تر باشد. بطوری که مهندسی معکوس طرح بسیار مشکل‌تر شود. در [۱۸] روشی برای بازدارندگی از تروا ارائه شده‌است و ماشین حالات مدار و گذار بین حالات به نحوی تغییر داده شده‌است که یک حالت مبهم‌سازی‌شده در سطحی بالاتر از عملیات اصلی (حالت عادی) مدار تعریف شود. شکل ۳ـ۷ عملیات مبهم‌سازی‌شده و عملیات عادی را نشان می‌دهد. تنها راه رفتن از حالات مبهم‌سازی‌شده به حالات عادی، کلید K۳ است.

الگوی ورودی که باعث گذار از حالات مبهم‌سازی‌شده به حالات عادی می‌شود را رشته کلید آغازین

می‌گویند. بدون دانستن این کلید، احتمال نفوذ حمله‌کننده به حالت عمکردی عادی، بسیار کم می‌شود. بنابراین تحلیل احتمال بروز رخدادها توسط طراح تروا، اطلاعات غلطی را به وی خواهد داد. برای آنکه احتمال یافتن کلید را کاهش دهیم، باید فضای حالات مبهم‌سازی‌شده بسیار بزرگ شود. تروا‌هایی که بعد از مبهم‌سازی طراحی به آن اضافه می‌شوند، به دو دسته تقسیم می‌شوند. دسته اول تروا‌هایی هستند که همه یا بخشی از مدار تحریکشان شامل حالات داخل بخش مبهم‌سازی‌شده است و دسته دوم آنهایی هستند که تمام مدار تحریکشان شامل حالات بخش عادی مدار است. تروا‌های دسته اول در حالت عادی، به هیچ وجه تحریک نمی‌شوند و نگرانی درباره آنها نداریم. اما تروا‌های دسته دوم ممکن است تحریک شوند. اما به علت اینکه احتمالاتی که طراح تروا از شبیه‌سازی‌ها بدست آورده، ارقام اشتباهی بوده است، لزوماً مدار تحریک این تروا‌ها شامل رخدادهای نادر نخواهد بود. برای پیاده‌سازی مبهم‌سازی طراحی، می‌توان از ابزارهای طراحی خودکار استفاده کرد. برای خودکارسازی این کار، الگوریتمی در [۱۸] ارائه شده‌است. معایب این روش: در بسیاری از موارد، فرضیات این روش صحیح نخواهد بود. اولین فرض این است که طراح تروا تنها از رخدادهای نادر برای تحریک تروا استفاده می‌کند. این در حالیست که اولاً برخی از تروا‌ها همیشه فعال هستند. ثانیاً تعریف نادر بودن یک رخداد ممکن است از دید طراح سیستم و طراح تروا متفاوت باشد. فرض دوم این روش این است که طراح تروا هیچ دیدی نسبت به نحوه مبهم‌سازی طراحی ندارد. اگر هریک از این فرضیات درست نباشد، کارایی این روش کاهش خواهد یافت. در [۱۶] احتمال گذار سیم‌های داخلی با یک توزیع هندسی مدل شده‌است و روش بازدارنده از تروای ارائه شده‌است که می‌تواند احتمال گذار مدارات تروا عملیاتی را افزایش دهد. برای این کار، فلیپ فلاپ‌های اضافی با نام فلیپ فلاپ‌های ساختگی، به مدار اضافه می‌شوند به نحوی که عملکرد مدار را تغییر ندهند. شکل ۳ـ۸ مدار اصلی و مدار شامل فلیپ فلاپ‌های ساختگی را نشان می‌دهد. همانطور که در شکل نشان داده شده‌است، احتمال یک شدن خروجی مدار با این روش ۸,۵ برابر شده‌است. این روش از دو طریق می‌تواند به تشخیص تروا و بازدارندگی از تروا کمک کند: ۱ ـ تحلیل اثرات جانبی مبتنی بر توان مصرفی: به علت اضافه کردن فلیپ فلاپ‌ها و دروازه‌های منطقی مربوط به آنها به مدار ، فعالیت تروا در حالت آزمون بیشتر خواهد شد و توان بیشتری مصرف خواهد کرد. ۲ ـ آزمون عملیاتی: فلیپ فلاپ‌های اضافه می‌توانند احتمال گذار سیم‌های داخلی را به نحوی تعدیل کنند که احتمال فعال شدن تروا افزایش یابد. بنابراین فرضیه رخداد نادر که طراح تروا طرح خود را بر آن استوار کرده‌است، صحیح نخواهد بود و در طول فاز آزمون نتایج غلط در خروجی‌ها مشاهده خواهد شد. در [۱۷] یک روش وارونه‌سازی ولتاژ برای افزایش فعالیت تروا بدون افزودن دروازه‌های منطقی



شکل ۳ـ۸: مدار اصلی و مدار شامل فلیپ فلاپ‌های ساختگی [۱۶]



شکل ۳ـ۹: معماری SoC شامل پیمانه‌های DEFENSE [۱۴]

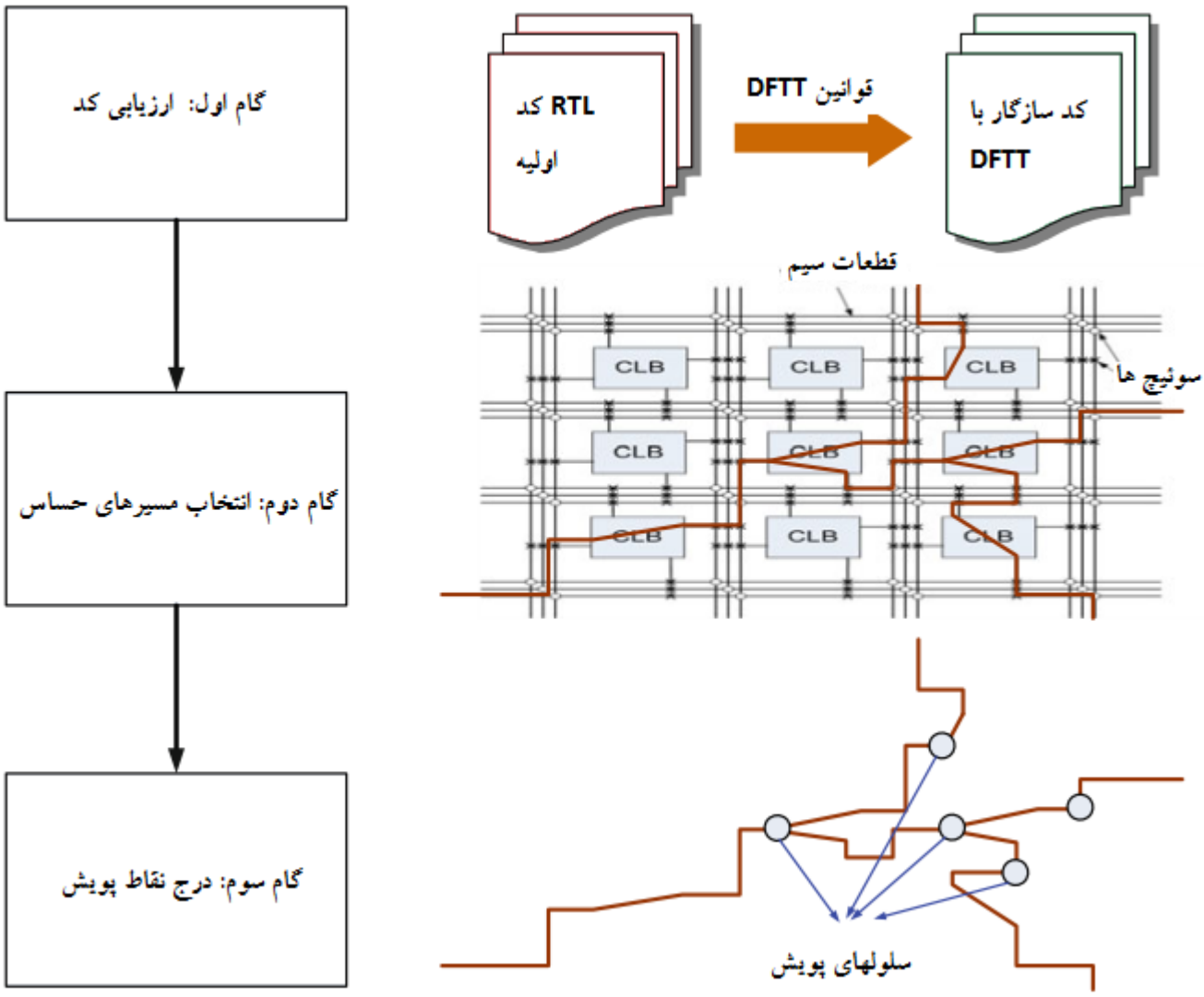
اضافه، ارائه شده‌است. ایده اصلی این است که با جابجا کردن تغذیه و زمین در دروازه‌های منطقی، عملکرد آن به نحوی تغییر می‌کند که خروجیِ با احتمال کم، بیشتر رخ خواهد داد. در منطق CMOS، وارونه کردن ولتاژ تغذیه باعث کاهش پتانسیل دروازه منطقی می‌شود و این امر باعث می‌شود پتانسیل طبقات بعدی نیز کاهش یابد. این امر باعث می‌شود بعد از چند طبقه، دیگر سیگنال در مدار منتشر نشود. برای جلوگیری از این امر نیز در این مقاله راهکاری ارائه شده‌است. در [۱۴] منطق زیرساختی برای انجام بررسی‌های امنیتیِ برخط در طول عملکرد عادی مدار، ارائه شده‌است که متمرکز بر حوزه SoC است. مدار با قابلیت بازپیکربندی موسوم به DEFENSE که مخفف «طراحی برای امنیت» است، به SoC افزوده می‌شود تا نظارت بر ناهنجاری‌ها را در زمان اجرا انجام دهد. شکل ۳ـ۹ معماری چنین SoC ای را نشان می‌دهد. پیمانه اصلی این زیرساخت، زوج مدار SPN یا شبکه پویش سیگنال و SM یا ناظر امنیت است. SPN که شبکه ای از مالتی پلکسرهای توزیع شده‌است، به نحوی پیکربندی می‌شود که زیرمجموعه‌ای از سیگنال‌های مهم تعریف شده توسط کاربر را انتخاب کند و به واحد SM منتقل کند. در این واحد رفتار مورد انتظار کاربر بررسی می‌شود. پردازشگر امنیت و کنترل SECOPRO به نحوی SPN را پیکربندی می‌کند که به طور پویا سیگنال‌های لازم را انتخاب کند. این پیکربندی‌ها رمز شده و در یک حافظه امن نگهداری می‌شود. وقتی ناهنجاری رفتاری تشخیص داده شود، پیمانه کنترل سیگنال SECOPRO را فعال می‌کند تا مقادیر سیگنال‌های مشکوک را بازنویسی کند و سیستم را به حالت عادی برگرداند. همچنین، هسته‌ای که رفتار نادرست داشته است، کنار گذاشته می‌شود. معایب این روش: سربار سخت‌افزاری ناشی از پیمانه‌های این زیرساخت مساله قابل تاملی است.

۳ـ۲ـ۶ طراحی برای آزمون تروا

در [۱۹] یک روش مقاوم سازی سخت‌افزار در برابر تروا ارائه شده‌است که مشابه روش مرسوم طراحی برای آزمون (DFT) در آزمون اشکال است. از آنجا که هدف این روش، مقاوم سازی دربرابر تروا است، طراحی برای آزمون تروا نام گرفته‌است. البته علیرغم شباهت در نام، تفاوت‌های بسیاری بین DFT و DFTT وجود دارد. هدف DFT یافتن اشکالات داخل مدار بدون تروا با ایجاد بردارهای آزمون است. در حالیکه هدف DFTT تشخیص حضور یا عدم حضور تروا با استفاده از این بردارهاست. در شکل ۳ـ۱۰ سه گام اصلی این روش که در ادامه توضیح داده می‌شود، نشان داده شده‌است. گام اول: ارزیابی کد: کد HDL مربوط به مدار اصلی با استفاده از قوانین کد نویسی DFTT به یک کد سازگار با DFTT تبدیل می‌شود. گام دوم: انتخاب مسیرهای حساس: فرض بر این است که طراحان تروا قصد دارند مداری به مدار اصلی اضافه کنند که از طریق آن اطلاعات حساس داخلی را سرقت کنند. بدین منظور قبل از درج تروا، تلاش می‌کنند تا اهمیت نسبی سیگنال‌های داخلی(مانند کلید رمز در مدارات رمز نگاری) را ارزیابی کنند. بنابراین ابزار DFTT که برای خودکار سازی عملیات DFTT طراحی شده‌است، مسیرهایی را که سیگنال‌های حساس از آنها عبور می‌کنند یا به عملکرد سیگنال‌های حساس مدد می‌رسانند، به نحوی از مسیر بین ورودی‌های مدار تا خروجی‌های مدار برکنار می دارد. گام سوم: درج نقاط پویش: براساس مسیرهای حساسی که در گام دوم انتخاب شد، سلولهای پویش در کد سازگار با DFTT درج می‌شود. این گام مشابه درج فلیپ فلاپ‌های پویش در هنگام استفاده از DFT است. اما سلولهای پویش کمی متفاوت هستند. بعد از اینکه طراحی ما با این روش مقاوم سازی شد، بقیه مراحل آزمون مشابه DFT است.

۳ـ۲ـ۷ سخت‌افزار حامل اثبات

در [۴۶] نویسندگان تلاش کرده‌اند محدودیت روش‌های تشخیص تروا رایج را بیان کنند. اینکه همه این روش‌ها سعی در بررسی حضور تروا در تراشه‌های ساخته شده دارند. اما درباره بررسی حضور تروا در طرح، قبل از ساخت آن، تلاش اندکی انجام شده‌است. روش‌هایی که تا کنون برای طراحی سخت‌افزار مطمئن معرفی شدند، هیچکدام تلاشی برای حفاظت از هایIP سخت‌افزاری در برابر تروا نکرده‌اند. درمورد هایIP نرم‌افزاری در سال ۱۹۹۶ روش PCC یا کد حامل اثبات، ارائه شده‌است. در این روش یک اثبات سوری که به صورت خودکار قابل صحت سنجی است، تعیین می‌کند که آیا کد مورد نظر از



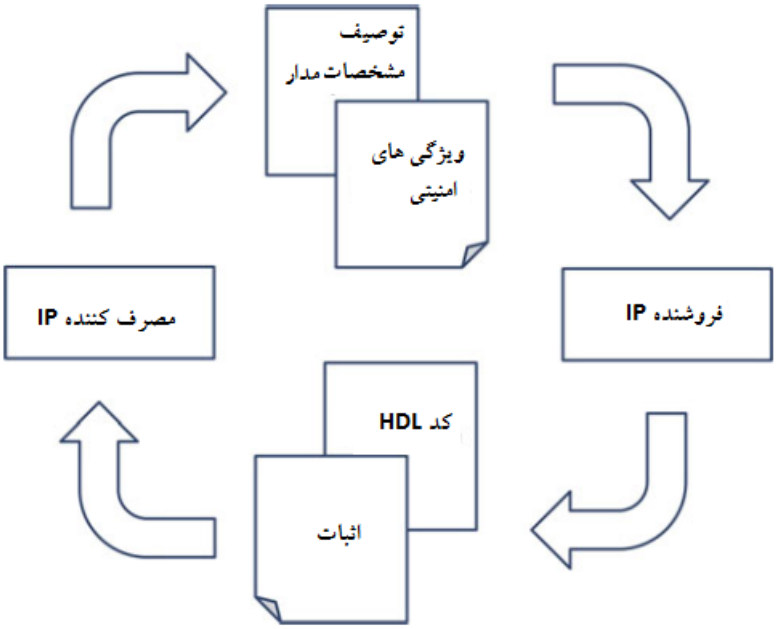
شکل ۳ـ۱۰: گامهای اصلی در روشDFTT [۱۹]

یک سری ویژگی‌های سوری پیروی می‌کند یا نه. بعد از آن، این اثبات با کد ترکیب می‌شود تا گیرنده بتواند به طور خودکار کد را نسبت به اثبات چک کند و تصمیم بگیرد که کد را اجرا کند یا نه. ایده گسترش این روش به حوزه سخت‌افزار مطمئن اولین بار در [۲۰] مطرح شد. نویسندگان این مقاله با اشاره به گسترش استفاده از هاFPGA و ابزارهای قابل بازپیکربندی، بر لزوم ارائه روش PCH تاکید کرده‌اند. به عنوان اولین گام در امنیت سخت‌افزار قابل اثبات، نویسندگان این مقاله روشی ارائه کرده‌اند که اثبات‌های مربوط به یکسانی مدارهای ترکیبی پیاده‌سازی شده در FPGA را فراهم می‌کند. این روش نیازمند یک تابع مشخصه S(x) برای هر عملیات منطقی و یک پیاده‌سازی I(x) استخراج شده از netlist مربوط به FPGA است. با این دو ورودی، اثبات به طور خودکار برای نشان دادن یکسانی S(x) با I(x) تولید می‌شود. نویسندگان این مقاله ساختاری به نام miter ایجاد کرده‌اند که از اعمال تابع XOR روی S(x) و I(x) حاصل شده‌است. هنگامی خروجی miter درست می‌شود که به ازای یک x این دو با یکدیگر نامساوی باشند. بنابراین اگر miter به هیچ وجه درست نباشد، یعنی مشخصه با پیاده‌سازی برابر است. محدودیت این روش این است که باید تابع بولی معادل مدار را داشته باشیم. در [۲۱] برای رفع محدودیت روش PCH ، PCHIP را برای ضمانت اثبات‌های مربوط به کد HDL مدار به جای رشته بیتی FPGA ارائه کرده‌اند. در این روش یک پروتکل جدید برای طراحی هسته‌های IP سخت‌افزاری ارائه شده‌است که در ۳ـ۱۱ نشان داده شده‌است.

۳ـ۲ـ۸ مقایسه روش‌های تشخیص تروا

مقایسه رویکردهای تشخیص تروا بر اساس اندازه نسبی تروا

جدول زیر خلاصه ای از مزایا و معایب نسبی روش‌های مبتنی بر آزمون منطقی را در مقایسه با روش‌های مبتنی بر تحلیل اثرات جانبی برای تشخیص تروا نشان می‌دهد. واضح است که دو روش مکمل یکدیگر هستند. بنابراین رویکردهایی که نقاط قوت هر دو را ترکیب کنند، می‌توانند مورد توجه بیشتری قرار گیرند. مزیت اصلی روش‌های زمان آزمون، عدم وجود سربار سخت‌افزاری است. درحالیکه عیب اصلی آنها نیازمندی به یک مدار مرجع یا مدار بدون تروا برای انجام مقایسه‌هاست. روش‌های زمان اجرا معمولاً سربار کارایی و توان مصرفی بالایی دارند ولی در مقابل امکان ایجاد اطمینان ۱۰۰% را فراهم می‌کنند.



شکل ۳ـ۱۱: روند طراحی هسته‌های IP و تولید و بررسی اثبات در روش PCHIP [۲۱]

بررسی و مقایسه روش‌های تشخیص تروا

در این بخش برخی از روش‌های تشخیص تروا حین آزمون شرح داده می‌شود و مزایا و معایب نسبی آنها مقایسه می‌شود. اکثر روش‌های تشخیص تروا که تا کنون ارائه شده‌است، در این دسته قرار می‌گیرند. برخی از این روش‌ها از روش آزمون عملکردی استفاده می‌کنند و برخی دیگر اثرات جانبی را تحلیل می‌کنند.

جدول ۳ـ۱: مقایسه آزمون‌های منطقی و اثرات جانبی

رویکرد تحلیل اثرات جانبی	رویکرد آزمون منطقی	
برای تروا‌های بزرگ موثر است تولید بردارهای آزمون ساده است	برای تروا‌های کوچک موثر است درمقابل نویز فرآیند تاثیرپذیر نیست	**مزایا**
نسبت به نویز فرآیند حساس است تشخیص تروا‌های کوچک مساله ساز است	تولید بردارهای آزمون پیچیده است. تشخیص تروا‌های بزرگ مساله ساز است	**معایب**

نوع روش	نام روش	تروهای قابل تشخیص	سربار	معایب دیگر
روش‌های آزمون منطقی	روش تولید بردارهای آزمون	اغلب تروهای با تحریک ترکیبی، تعداد معدودی از تروهای با تحریک ترتیبی(فقط تروهایی که با شرایط خاص فعال می‌شوند)	این روش در حین آزمون انجام می‌شود و سربار زمانی بسیار زیادی دارد.(۴۴,۵ ساعت برای مدار c۷۵۵۲)	طبق نتایج مقاله، پوشش دهی تروها برای مدارهای ترتیبی نسبت به روش بردارهای تصادفی تفاوت چندانی نداشته و در برخی موارد کمتر شده‌است. ضمن اینکه این روش نیاز به تحلیل قبلی مدار برای یافتن رخدادهای نادر دارد. MERO [۲۸]
	روش شفاف کردن پیمانه‌ها (افزایش کنترل پذیری و مشاهده پذیری) [۲۹]	فقط تروهای خیلی بزرگ (فقط تروهایی که در شرایط خاص فعال می‌شوند)	طبق گزارش مقاله ٪۵ سربار مساحت، ٪۱۳ سربار توان مصرفی، ٪۵ سربار تاخیر و ۹ پایه اضافه لازم است.	نحوه اعمال کلیدها به پیمانه‌ها و ایجاد امضای خروجی، در نتیجه این روش موثر است که در این مقاله توضیح چندانی داده نشده‌است.

نوع روش	نام روش	تروا‌های قابل ئشخیص	سربار	معایب دیگر
روش‌های تحلیل اثرات جانبی با رویکرد اندازه گیری توان مصرفی	روش ناحیه بندی [۱۰]	فقط تروا‌های با تحریک ترتیبی که در شرایط خاص فعال می‌شوند	سربار زمانی حین آزمون زیاد است و هرچه مدار بزرگتر شود، این سربار بیشتر می‌شود.	میزان نتیجه گیری کاملا وابسته به شخص است. چرا که هیچ روند خودکارسازی برای ناحیه‌بندی و ایجاد بردارهای آزمون ارائه نشده‌است.
	روش اعمال بردارهای پایدار شده [۲۴]	اغلب تروا‌های با تحریک ترکیبی، تعداد معدودی از تروا‌های با تحریک ترتیبی (فقط تروا‌هایی که با شرایط خاص فعال می‌شوند)	سربار زمانی زیاد در حین آزمون به علت ثابت نگهداشتن بردار ورودی برای مدت زمان تعیین شده لازم است.	افزایش حرارت و در پی آن کاهش عمر مدار به علت افزایش خودخواسته توان مصرفی
	روش محاسبه جریان از طریق پایه‌های تغذیه مختلف [۶]	این روش فقط برای یک مدار ترکیبی آزموده شده‌است. برای مدارات ترتیبی خیلی مناسب نخواهد بود.	سربار ابزارهای جانبی اندازه‌گیری دقیق جریان ـ سربار زمانی حین آزمون	پوشش تروا پایین در حد ٪۵۰ برای تروا‌های فعال و ٪۳۰ برای تروا‌های غیرفعال

نوع روش	نام روش	تروا‌های قابل تشخیص	سربار	معایب دیگر
	استفاده از ترتیب دهی مجدد فلیپ فلاپ‌های پویش [۱۶]	انواع تروا‌های کوچک و بزرگ (فقط تروا‌هایی که با شرایط خاص فعال می‌شوند)	سربار زمانی حین آزمون	عدم تشخیص برخی تروا‌های با تحریک ترتیبی- نیاز به پایه‌های اضافه برای تراشه، بسته به کوچکترین تروا قابل تشخیص
روش‌های تحلیل اثرات جانبی با رویکرد اندازه گیری تاخیر	محاسبه تاخیر مسیرها [۷]	فقط تروا‌های با تحریک ترکیبی- تشخیص تروا‌های بزرگ راحت تر است.	سربار زمانی حین آزمون	عدم تشخیص تروا‌هایی که در مسیر بین ورودی و خروجی مدار واقع نشده و در مسیرهای داخلی اند.
	محاسبه تاخیر مسیرهای داخلی با استفاده از ثبات‌های سایه [۸]	انواع تروا‌ها با اولویت تروا‌های بزرگ (هرچه اندازه تروا‌ها کوچکتر باشد، نیاز به سربار بیشتر زمانی و پیچیدگی بیشتر طراحی است)	سربار زمانی زیاد حین آزمون- سربار مساحت و توان ناشی از تولید کننده سیگنال کلاک — سربار مساحت و توان ثبات‌های سایه	دقت تشخیص تروا وابسته به گام تغییر فاز کلاک است که پیچیدگی را زیاد می‌کند. با افزایش ابعاد مدار شبکه توزیع کلاک دردسرساز می‌شود.

نوع روش	نام روش	تروا‌های قابل تشخیص	سربار	معایب دیگر
	تشخیص تغییرات تاخیر با استفاده از نوسانگرهای حلقوی [۳۰, ۳۱]	فقط تروا‌های با تحریک ترکیبی- تشخیص تروا‌های بزرگ راحت تر است.	سربار مساحت و توان نوسانگرها یا مدارات اضافه جهت ساختن آنها در مدار- سربار زمانی حین آزمون برای خواندن نتایج- سربار مساحت و توان اندازه گیرهای فرکانس	تعداد نوسانگرها و موقعیت درج آنها کاملا وابسته به دانش فرد طراح است.

جدول ۳ـ۲: مقایسه روش‌های مقابله با تروا

فصل ۴

روش پیشنهادی

در این فصل روش پیشنهادی و نتایج جدید به‌دست‌آمده در پایان‌نامه توضیح داده می‌شود.

۴ـ۱ راهکار ما

تلاش این پروژه برای رسیدن به راهکاری ترکیبی و نوین است. در روش پیشنهاد شده، هم از آزمون منطقی و هم از تحلیل اثرات جانبی استفاده می‌شود. پس این رویکرد را رویکرد ترکیبی می‌نامیم. مزایا و اهداف استفاده از این روش، متشکل از دو موردِ اصلی است:
(۱) رسیدن به راهکاری که بر طبق استانداردهایی مانند درصد پوشش تروا‌ها(دقت آزمون) و یا تعداد بردار لازم(سرعت آزمون)، نسبتاً بهتر از یا قابل مقایسه با کارهای مرتبط پیشین باشد.
(۲) به دست آوردن قاعده‌ای برای انتخاب رویکرد بهینه، بین دو رویکرد ذکر شده، با توجه به اندازه‌ی نسبی تروا.

همان طور که گفته شد، راهکار ترکیبی معرفی شده در این پروژه، از هر دو رویکرد استفاده می‌کند. آزمون منطقی یک مدار الکترونیکی، به طور ساده شده، عبارت است از مشاهده خروجی‌های یک مدار در پاسخ به ورودی‌هایی که خروجی آن‌ها در مدار سالم، از قبل معلوم است. هرگونه مغایرت در خروجی‌ها، به منزله وجود تروا در مدار تفسیر خواهد شد. در این پروژه، آزمون، نه تنها در مرحله آزمون منطقی از بسیاری از روش‌های پیشین کاراتر است، بلکه مهم تر از آن، احتمال موفقیت در مرحله دوم، توسط اندازه‌گیری اثرات جانبی را افزایش می‌دهد. به طور دقیق تر، درمرحله اول، درهنگام تولید بردارهای

آزمون منطقی، به جای استفاده از از بردارهای تصادفی، با کمک‌گیری از "ابزار کمکی تروا"، سعی می‌کنیم حدس بزنیم که تروا‌ها در چه گره‌ای از مدار ظاهر خواهند شد. در هنگام آزمون نیز تروا را در نقاطی از مدار قرار می‌دهیم که احتمال حمله به آن‌ها بیشتر است. نرم افزار مذکور برای این پروژه طراحی و پیاده‌سازی شده است. توضیح وعملکرد این نرم افزار به در بخش ۴ـ۳ انجام شده است. همچنین از آنجا که نقاط حساس مدار را بدست آورده‌ایم، با الگوریتمی شبیه MERO سعی خواهد شد که فعالیت مدار در نقاط محتمل برای حمله، افزایش داده شود. طراحی و پیاده‌سازی چنین الگوریتمی، یکی از چالش‌های اصلی این پروژه بود. به طور ساده، خروجی این الگوریتم تعداد محدودی بردار تست، با درصد پوشش بسیار مطلوب و بالا است. توضیحاً اضافه می‌شود که کم بودن تعدادِ بردارهای آزمون، یعنی پائین آمدن زمان آزمون. این امر هدف ۱ را ارضا خواهد کرد.

در ادامه، درباره‌ی چگونگی پیاده‌سازی روش اندازه‌گیری اثرات جانبی در این پروژه توضیح داده می‌شود. پارامتر مورد تعقیب ما، توان مصرفی خواهد بود. چگونگی این اندازه گیری برای مدارهای سنتز شده، در بخش بعد توضیح داده خواهد شد. اما در این بخش، ارائه یک توضیح منتزع از فرایند اندازه‌گیری توان مصرفی، ضروری به نظر می‌رسد: در مرحله قبل، مدار را به گونه‌ای تحریک کردیم که فعالیت تروا‌ها بالا رود. حالا که تغییرات همه گره‌ها را محاسبه کرده و در اختیار داریم، اقدام به محاسبه توان مصرفی می‌کنیم. اگر توان مصرفی مدار زیر آزمون، بالاتر از حداکثر توان مصرفی مدار سالم باشد، مدار زیرآزمون، حاوی تروا تشخیص داده می‌شود. برای رسیدن به هدف ۲، باید دو مرحله ذکر شده را برای تروا‌های کوچک و بزرگ، روی مدارهای مرجع، انجام داد. پس از تکرارهای متعدد، به این جمع‌بندی خواهیم رسید که در مدارهای ترکیبی مشخصی—که این مدارها در بخش ۴ـ۳ معرفی شده اند— اندازه تروا‌های مورد بررسی چگونه می‌تواند نوع رویکرد را مشخص کند. به عبارت دیگر، با دسترسی به هدف ۲ قادر خواهیم بود که گزارش کنیم: "برای یافتن هر تروای مشخص، در هر مدار مشخص، بهینه است از چه رویکردی استفاده شود." بدین ترتیب، مفهوم "رویکرد اندازه‌ـآگاه تشخیص تروا"، پیادهسازی خواهد شد.

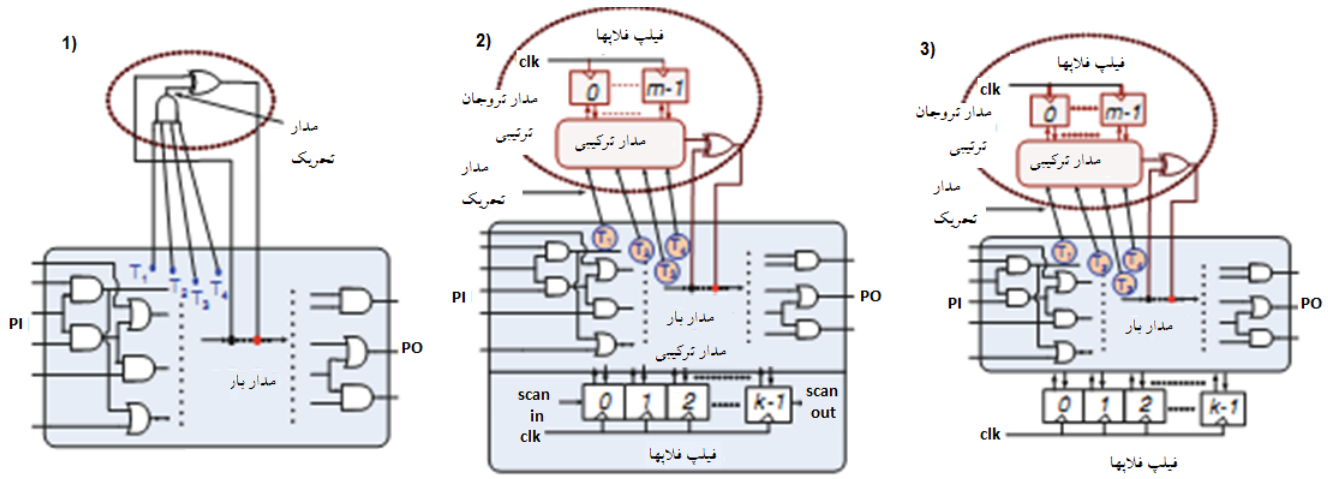
۴ـ۲ چالش‌های پیش رو در روش‌های تشخیص تروا

چالش‌های اصلی روش‌های تشخیص تروا را می‌توان سه چالش عمده دانست:

- انتخاب مدل مناسب برای تروا
- تولید بردارهای آزمون برای فعالسازی تروا یا افزایش حساسیت به تروا در روش‌های مبتنی بر اثرات جانبی
- حذف یا کالیبراسیون نویز اندازه‌گیری، محیط و فرآیند

مدل‌سازی تروا

پژوهشگران تا کنون مدل‌های متفاوتی برای تروا انتخاب کرده‌اند. برای اعمال یکدستی در فرآیند و ایجاد امکان مقایسه بین روش‌های مختلف، طبقه بندی سازمان یافته‌ای برای تروها براساس پارامترهایی که توصیفگر اندازه، میزان مخفی بودن، احتمال فعال شدن، اثر ناشی از تروا و غیره است، ارائه شده‌است. برای مقاصد تولید بردار آزمون، استفاده از مدل تروا با تحریک دیجیتال و اثر دیجیتال، از سایر مدل‌ها مفیدتر است. مدل عمومی برای تروهای ترکیبی در فصل قبل نشان داده شد. شرط تحریک تروا ترکیبی، رخداد یک مقدار n بیتی در گره‌های داخلی است که فرض شده‌است به اندازه کافی نادر است. خروجی مدار بار گرهی است که وقتی تروا فعال می‌شود، مقدار منطقی‌اش معکوس می‌شود. برای مشکل کردن تشخیص تروا می‌توان از مدل ترتیبی استفاده کرد که برای فعالسازی تروا لازم است این واقعه نادر، چندین بار تکرار شود. در مدل‌های تروا ترتیبی از یک ماشین حالت استفاده می‌شود که در ساده ترین حالت یک شمارنده است. همچنین به جای مدار بار که در شکل ۴ـ۱ با دروازه XOR مدل شده‌است، هر مدار ترکیبی دیگری می‌تواند قرار گیرد. مدار تحریک نیز که در شکل ۴ـ۱ به صورت دروازه منطقی AND نمایش داده شده، می‌تواند در حالت کلی هر مدار ترکیبی دیگری باشد. برای تشخیص تروا با استفاده از روش‌های تحلیل اثرات جانبی، مدل تروا می‌تواند بسیار ساده و در حد یک دروازه منطقی غیرفعال باشد تا اثرش را بر جریان تغذیه سکون مدل کند یا یک خازن باشد تا اثرش را بر تاخیر مسیر مدل کند. با این وجود برای تشخیص تروا با استفاده از نظارت بر جریان گذرا یا تابشهای الکترومغناطیس و غیره، لازم است در مدارات تروا فعالیت سوئیچینگ اعمال شود. در این موارد می‌توان از مدل‌های ترواییِ که در روش‌های آزمون منطقی استفاده می‌شوند، استفاده کرد. هرچه اندازه تروا بزرگتر شود، اثرش بر سیگنال‌های جانبی بیشتر می‌شود. از طرفی با افزایش اندازه مدار اصلی، اثر تروا بر این سیگنال‌ها قابل اغماض خواهد بود. بنابراین در مقایسه مدل‌های تروا، اندازه نسبی تروا به مدار اصلی و اثر تروا بر سیگنال‌های جانبی مقایسه می‌شود.



شکل ۴ـ۱: انواع تروا در مدارهای ترکیبی و ترتیبی [۴۱]

تولید بردارهای آزمون

تولید بردارهای آزمون یکی از مهمترین بخش‌های هر روش تشخیص تروا است. لازم است در طی فرآیند آزمون مدار، هیچ سیگنال فعالساز آزمونی[۱] به طور آشکار موجود نباشد. چرا که مدار تحریک تروا می‌تواند از این سیگنال استفاده کند و با فعال شدن آن، مدار تحریک غیرفعال شود و مانع از تشخیص تروا شود. اعمال بردارهایی که به صورت تصادفی نقاط مختلف مدار را بیازمایند مفید نیست. چرا که تروها به صورت هوشمندانه درج می‌شوند. بطوریکه سیگنال‌های ورودی تحریک دارای کنترل‌پذیری پایین و سیگنال‌های خروجی بار، دارای مشاهده‌پذیری پایینی باشند. در شکل ۴ـ۱ انواع مختلف تروا در مدارات ترکیبی و ترتیبی نشان داده شده‌است. چالش بعدی پیش روی تولید بردارهای آزمون، تعداد بیش از اندازه زیاد تروهای ممکن است که با استفاده از مجموعه محدودی از گره‌های داخلی قابل ساخت هستند. با توجه به این موضوع، تولید بردارهای آزمونی که به طور کامل تمام تروهای ممکن را تشخیص دهند، عملاً ناممکن است. حتی وقتی گره‌های تحریک را به چهار گره و گره‌های بار را به یک گره محدود کنیم، برای مداری به کوچکی C۸۸۰ از مجموعه مدارات ISCAS-۸۵ با ۴۵۱ دروازه منطقی، می‌توان $T \approx ۴/۱ \times ۱۰۱۰$ مدار تحریک و $T \times (۴۵۱ - ۴) \approx ۱/۸ \times ۱۰۱۳$ تروا مختلف داشت. با توجه به مسائل مطرح شده استفاده از روش‌های تولید بردار آزمون آماری [۳۷] رایج شده‌است.

[۱] Test Enable

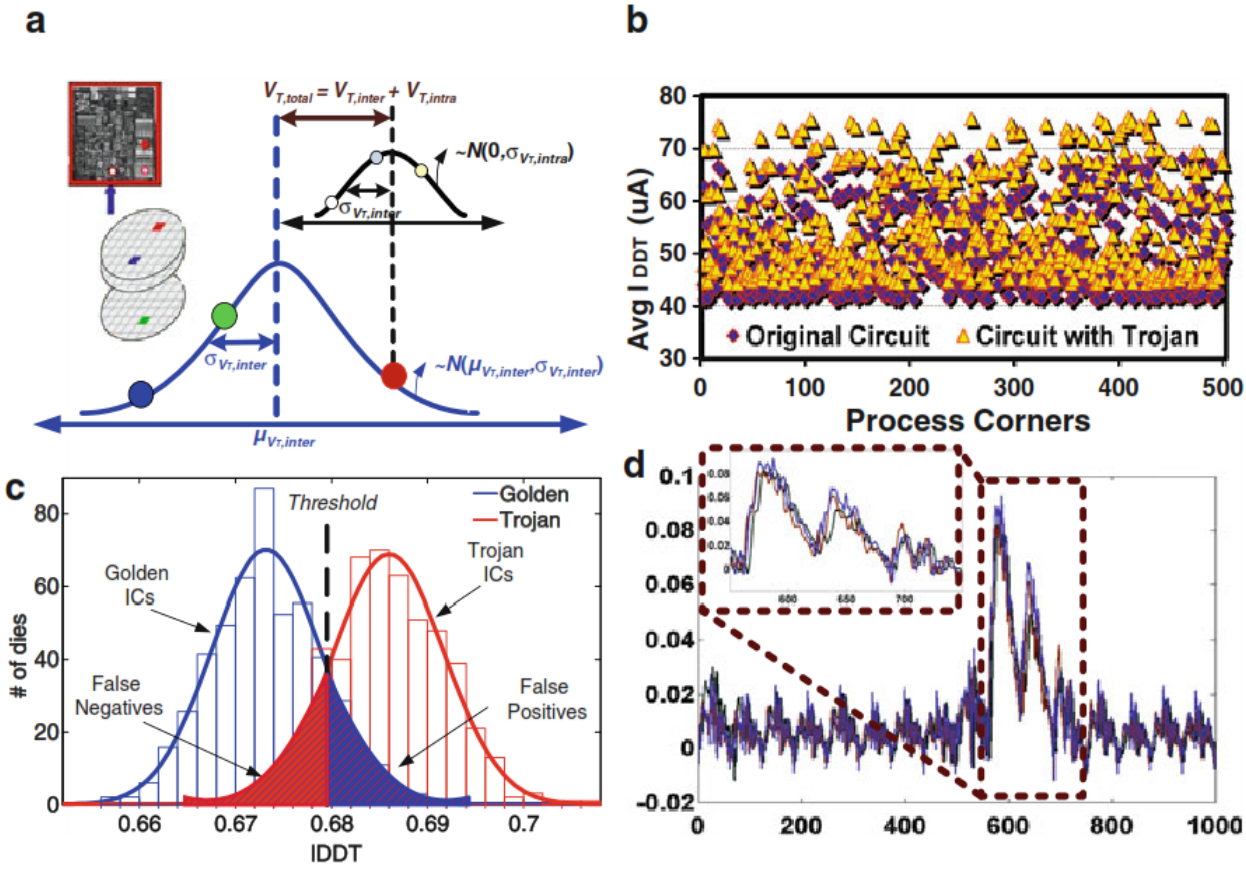
نویز اندازه‌گیری، محیط و فرآیند

در روش‌های تحلیل اثرات جانبی، تولید بردارهای آزمون به مراتب ساده‌تر است چرا که نیازی به فعالسازی تروا برای مشاهده نتیجه آن بر پارامتری مثل جریان تغذیه نیست. با این وجود تکنولوژیهای جدید در ابعاد نانو از مشکل تغییرات وسیع در پارامترهای فرآیند رنج می برند. نشان داده شده‌است [۳۸] که تغییرات فرآیند در تکنولوژی ۱۸۰ نانومتر تا ۳۰ درصد اختلاف در تاخیر و ۲۰ برابر تغییر در جریان نشتی را موجب می‌شود. شکل ۴ـ۲ اثر تغییرات فرآیند را بر ولتاژ آستانه ترانزیستورها نشان می‌دهد که با توزیع گوسی مدل شده‌است. جریان شبیه‌سازی شده همپوشانی زیادی در جریان را نشان می‌دهد که نشانگر اثر مدار تروا است که توسط نویز فرآیند پوشش داده شده‌است. در بخش c شکل، مشاهده می‌شود که تغییرات ناشی از نویز فرآیند تابعی از مقدار پارامتر اندازه‌گیری شده‌است. بنابراین برای مدارات بزرگ با فعالیت سوئیچینگ بالا، جریان تغذیه گذرا می‌تواند تغییرات زیادی داشته باشد که ممکن است اثر تروهای کوچک بر جریان را پوشش دهد. بنابراین ممکن است تراشه‌های دارای تروا به عنوان سالم تلقی شوند.

یک راه حل، افزایش حساسیت به تروا است. انتخاب بردار ورودی که پارامترهای جانبی در شرایط اعمال آن اندازه‌گیری می‌شوند، نقش مهمی در مقدار اندازه‌گیری شده دارد. بردارها باید به نحوی انتخاب شوند که نقش مدار اصلی کمینه و نقش تروا بیشینه شود. بخش‌بندی به نواحی مختلف و تولید بردار هدایت شده برای القای فعالیت بیشتر در محل‌های محتمل وجود تروا، می‌تواند این امر را میسر کند. از طرف دیگر با کاهش تاثیر نویز فرآیند می‌توان حساسیت روش‌های تشخیص تروا را بالا برد. میانگین گیری آماری از نویز فرآیند و حذف آن از سیگنال اندازه‌گیری شده در پژوهش‌های اخیر [۳۸] برای افزایش حساسیت انجام شده‌است.

۴ـ۳ شبیه سازی

دز این بخش ضمن بررسی روند شبیه‌سازی، نرم افزارها و ابزار مورد استفاده در این پروژه را مرور می‌کنیم.



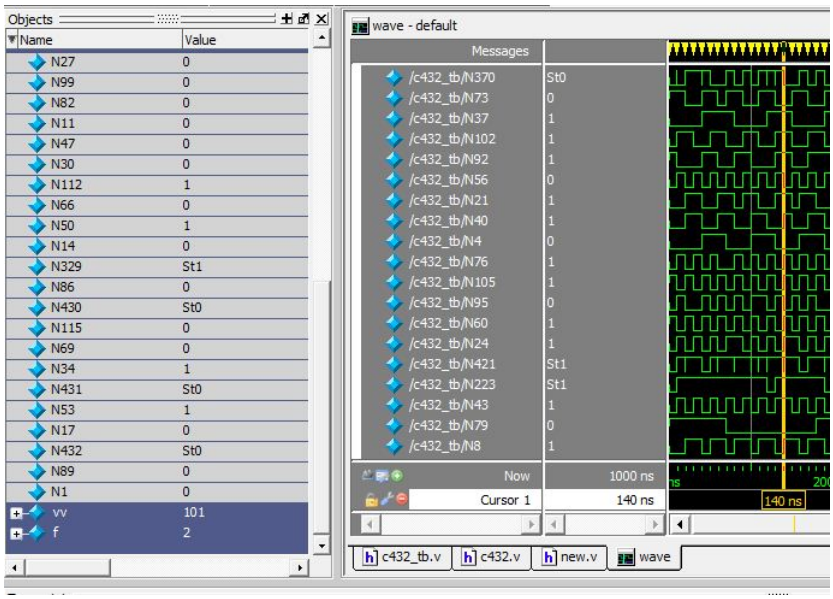
شکل ۴ـ۲: اثر تغییر فرآیند بر ولتاژ آستانه و جریان نشتی [۳۸]

۴ـ۳ـ۱ محیط شبیه‌سازی

در این بخش انواع محیط، ابزار و نرم‌افزارهایی که برای شبیه‌سازی در این پروژه مورد استفاده قرار گرفته می‌شوند را بررسی خواهیم کرد.

- **ModelSim**

برای آزمون منطقی مدار، و همچنین به دست آوردن میزان فعالیت مدار بر اثر یک دسته بردار آزمون در قالب یک فایل VCD از این محیط شبیه‌سازی بهره می‌بریم. این عمل روی ۱۰ مدار از سری ISCAS-۸۵ C انجام شد و خروجی آن، ورودی مرحله بعد است. در شکل مقادیر گره‌های مدار C۴۳۵ قابل مشاهده است.



شکل ۴ـ۳: نمای شبیه‌ساز ModelSim

برای آزمون منطقی، از این شبیه‌ساز استفاده می‌کنیم.

- **vcd2saif commandline tool**

برای محاسبه توان مصرفی، و همچنین پیدا کردن محل قراردادن تروا در مدار، بهتر است فایل فعالیت مدار را به صورت saif. در آوریم. این ابزار عموماً روی سیستم‌های لینوکس که نرم‌افزار design vision را نصب داشته باشند، پیدا می‌شود. خروجی این مرحله، به عنوان ورودی در هر سه مرحله آتی لازم است. این مرحله، در شبیه سازی ویژه این پروژه انجام شد، و خروجی آن به مرحله Trojan Insert Helper داده شد.

- **Synopsis Design Compiler**

برای آزمون اثرات جانبی، ابتدا نیاز به مدار سنتز شده داریم. این نرم افزار مدار را برای ما سنتز می‌کند.

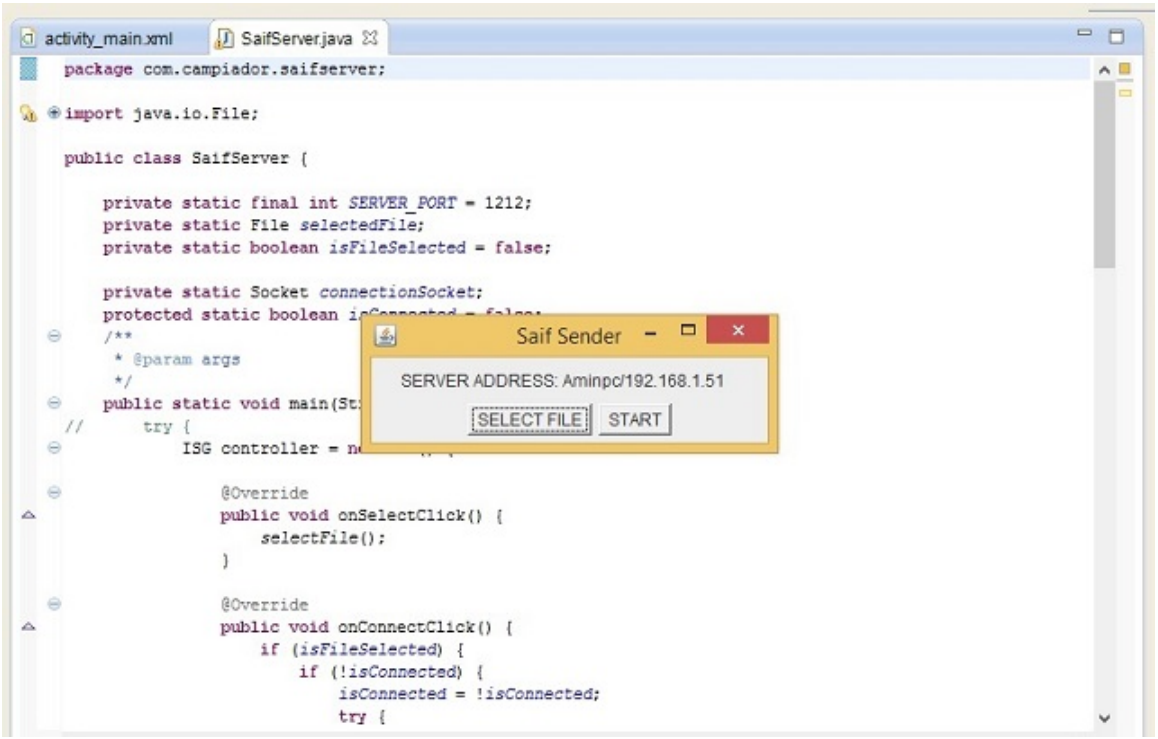
- **Synopsis Power Compiler**

محاسبه توان مصرفی، از روی یک فایل saif و یک مدار سنتز شده، توسط این ابزار میسر است.

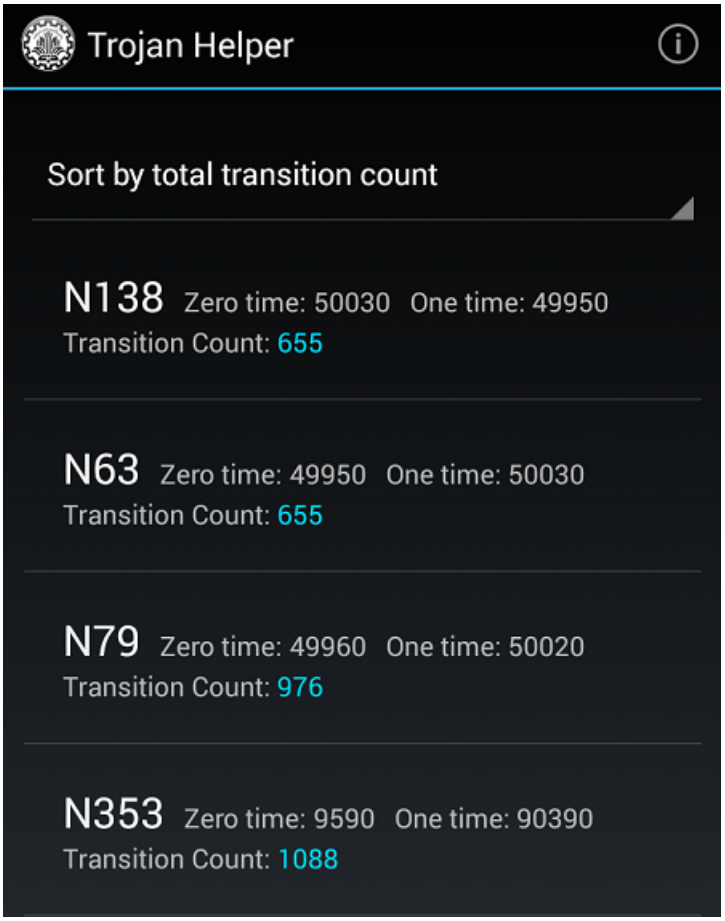
- **Trojan Helper**

این نرم افزار، که برای همین پروژه طراحی و پیاده سازی شده است، امکان شناسایی و انتخاب بهترین محل برای قرار دادن تروا در یک مدار را می‌دهد. روش کار بدین صورت است که اطلاعات فایل saif به صورتی تحلیل میشود که گرههای مدار به ترتیب زمان صفر بودن، مدت زمان یک بودن، یا نرخ تغییرات برای کاربر دسته بندی می‌شوند. فرض اصلی این نرم افزار این است که تروا‌ها معمولا در گره‌های راکد و کم تغییر جایگذاری می‌شوند و در غیر این صورت به راحتی قابل کشف هستند. این نرم افزار از دو ابزار تشکیل شده است:

(۱) بخش سرور ـ به زبان جاوا ـ این ابزار یک فایل saif. می‌گیرد، یک Socket برای انتقال اطلاعات به صورت بیسیم، تحت استاندارد IEEE 802.11 می‌سازد. بعد از برقراری ارتباط، این ابزار اقدام به فرستادن اطلاعات فعالیت تک تک گره‌های مدار، تحت پروتکل TCP از روی فایل saif. می‌نماید. نمایی از رابط کاربری این ابزار در ۴ـ۴ آمده است.



شکل ۴ـ۴: ابزار saif



شکل ۴ـ۵: ابزار saif ـ مشاهده اطلاعات

(۲) بخش کلاینت ـ این بخش هم برای رایانه و هم برای گوشی هوشمند و تبلت پیاده سازی شده است. وظیفه این ابزار، پارس کردن اطلاعات فرستاده شده از طرف سرور، آنالیز و نمایش آن‌ها به گونه‌ایست که کاربر بتواند به راحتی تصمیم بگیرد کدام نقطه برای جایگذاری تروا مناسب‌تر است. برای مثال، در شکل ۴ـ۵، کاربر درخواست کرده که فهرست گره‌های مدار به ترتیب صعودیِ تعداد سوئیچ‌ها از صفر به یک و برعکس، نمایش داده شود. پس طبق محاسبات انجام شده در مثال بالا، احتمال حضور تروا در گرههای N۱۳۸ و N۶۳، از باقی گره‌ها بیشتر است. در نتیجه، بهتر است برای شبیه‌سازی یک تروا ۲ بیتی برای آزمون، پایه‌های ورودی تروا را از این دو گره بگیریم.

- **Insert Trojan**

این برنامه command-line tool به زبان جاوا نوشته شده است. هدف این برنامه قرار دادن یک تروا با اندازه مشخص در یک مدار verilog است. طرز استفاده از این دستور بصورت زیر است:
رو

```
java InsertTrojan "../input/c400.v" "4"
```

در این صورت برنامه یک تروا با اندازه ۴، در مدار c۴۰۰ قرار می‌دهد و خروجی را در مدار c۴۰۰trojan.v ذخیره میکند.

- **Tetramax**

از این نرم‌افزار برای حذف بردارهایی که اثر آن‌ها در خروجی قابل مشاهده نیست، استفاده می شود.

۴ـ۳ـ۲ مجموعه داده

برای مدارهای میزبان تروا در آزمون‌ها، از مدارهای ISCAS-۸۵ benchmark استفاده شد. برای بردارهای ورودی، از حدود ۱۰۰۰ بردار تصادفی با الگوریتمی شبیه الگوریتم MERO طراحی و پیاده‌سازی شد. همان طور که در بخش ۳ بیان شد، ویژگی این الگوریتم در آن است که می‌تواند فعالیت مدار را در گره‌های محتمل به حضور تروا، بیشتر کند. به طور ساده، این اگوریتم می‌تواند مجموعه بردارهایی تولید کند که گره‌های تروا‌خیز مدار را بیشتر از حالت عادی تحریک کند.

۴ـ۳ـ۳ نتایج شبیه‌سازی

در شبیه‌سازی‌های این پروژه، بستری فرآهم آوردیم تا بتوانیم یک مدار را تحریک و شبیه‌سازی کنیم. در ادامه این شبیه‌سازها را مرور می‌کنیم.

۴ـ۴ الگوریتم تولید بردار آزمون

این الگوریتم سعی می‌کند به جای استفاده از الگوهای‌های تصادفی، به دنبال مجموعی برداری بگردد که گره‌های تروا‌خیز مدار را تا حد امکان تحریک کنند. این الگوریتم در ۴ـ۴ بیان شده است. با داشتن لیت گره‌های حساسی که در بخش قبل بدست آوردیم، سعی می‌کنیم بردارهای ورودی را تا جای ممکن کم کنیم. این بردارها باید هر گره‌ی حساس را حداقل N بار به مقدار کمیابش تحریک کند.

مدار	تعداد بردار با MERO	٪ کاهش تعداد بردار
c۲۶۷۰	۸۴۲۲	۹۶
c۳۵۴۰	۱۲۶۴۸	۸۷
c۵۳۱۵	۱۶۶۵۲	۸۴
c۶۲۸۸	۱۰۱۹۴	۹۰
c۷۵۵۲	۷۶۵۹	۹۳
s۱۳۲۰۷	۱۸۱۵۷	۸۲
s۱۵۸۵۰	۲۳۴۲۶	۷۷
s۳۵۹۳۲	۲۱۸۱۱	۷۸
میانگین	**۱۲۹۹۶**	**۸۶**

جدول ۴ـ۱: کاهش تعداد بردارها با MERO در بردارهای تصادفی

Algorithm 1 Procedure MERO - Generate reduced test pattern set for Trojan detection

inputs:

	▷ Circuit netlist
L	▷ List of rare nodes with associated rare values
V	▷ List of random patterns
N	▷ Number of times a rare condition should be satisfied

output:

R_V	▷ Reduced pattern set

1: Read circuit and generate *hypergraph*
2: **for all** nodes in L **do**
3: number of times node satisfies rare value (A_R)$\leftarrow 0$
4: $R_V \leftarrow \varnothing$
5: **for all** random pattern in V **do**
6: Propagate values
7: Count the # of nodes (C_R) in L with their rare value satisfied
8: Sort vectors in V in decreasing order of C_R
9: **for all** vector v_i in decreasing order of C_R **do**
10: **for all** bit in v_i **do**
11: Perturb the bit and re-compute # of satisfied rare values (C'_R)
12: **if** ($C'_R > C_R$) **then**
13: Accept the perturbation and form v'_i from v_i
14: Update A_R for all nodes in L due to vector v_i
15: **if** v'_i increases A_R for at least one rare node **then**
16: Add the modified vector v_i to R_V
17: **if** ($A_R \geqslant N$) for all nodes in L **then**
18: break

آزمون اثرات جانبی	آزمون منطقی	اندازه تروجان
۷۸	۹۷	۲
۸۲	۹۵	۴
۸۶	۹۴	۸
۸۷	۹۳	۱۲
۹۲	۷۶	۱۶

جدول ۴ـ۲: اثر اندازه تروا بر نتیجه آزمون (نرخ کشف تروا)

۴ـ۵ شبیه‌ساز تروا یاب

این شبیه‌ساز از الگوریتم بخش قبل استفاده می‌کند، مدار سالم را با بردارهای تولید شده شبیه‌سازی می‌کند و خروجی‌های مدار را برای استفاده ذخیره می‌کند. سپس با استفاده از ابزار درج تروا، مدارهای حاوی تروا تولید می‌کند و سعی می‌کند خروجی‌های مدار را با مدار سالم مقایسه کند. مشاهده اختلاف در خروجی‌ها به معنی کشف تروا است و در صورت وقوع، برنامه به سراغ تروای بعدی می‌رود. در فصل بعد بهبودهایی که بر اثر استفاده از این بردارهای هوشمند به دست آمد، در برابر بردارهای تصادفی مقایسه خواهند شد.

۴ـ۶ بررسی اثر اندازه تروا بر نتیجه آزمون

در جدول ۴ـ۲ مشاهده می‌شود که با بالا رفتن اندازه مدار، به دلیل گم شدن اثرات جانبی در لابه لای تغییرات فرایند، پیدا کردن مدار تروا سخت تر شد. ولی هرگاه مقدار نسبی مساحت تروا زیاد شد، آزمون اثرات جانبی بهتر جواب داد. این آزمون میانگین نتایج به دست آمده از مدارات ISCAS-۸۵ را نمایش می‌دهد. از سوی دیگر همان طور که با منطق و احتمالات پیش‌بینی می‌شد، آزمون منطقی در پیدا کردن تروهای کوچک اکیداً موفق‌تر بوده است.

۴ـ۷ مقایسه نرخ کشف آزمون اثرات جانبی با استفاده از بردارهای هوشمند و تصادفی

Ckt.	Trojan Cov. for 100K Random Vectors (%)					Trojan Cov. for Reduced Smart Vectors (%)				
	Trojan Size					Trojan Size				
	2	4	8	16	32	2	4	8	16	32
c432	100	99	99	98	98	100	99	99	98	98
c499	100	99	98	98	98	100	99	98	98	98
c880	99	99	98	97	95	99	99	98	97	94
c1355	97	95	94	92	90	99	99	98	97	95
c1908	98	96	95	93	92	99	99	98	97	96
c2670	96	91	88	83	74	98	98	97	95	92
c3540	96	92	88	83	71	97	95	94	92	88
c5315	90	88	86	79	70	96	95	93	90	87
c6288	88	86	82	76	68	90	90	87	85	80
c7552	86	86	80	70	59	91	87	81	74	68

Table 4-3: مقایسه آزمون اثرات جانبی با بردارهای هوشمند و تصادفی (نرخ کشف تروا)

همان طور که در جدول ۵ـ۱ مشاهده می‌شود بردارهای هوشمند ما بسیار بهتر از بردارهای تصادفی[۴۷] تروا‌ها را کشف می‌کنند. دلیل آن است که بیشتر توان مصرفی در سلول‌های TSMC 130 nm متشکل از توان پویا می‌باشد. این توان پویا وابسته به میزان سوئیچینگ گره‌های مدار است. حال که ما مدار را ثابت نگه داشته و بیشتر سعی بر فعال کردن گره‌های تروا‌خیز داشتیم، این اختلاف به خوبی در توان مصرفی خود را نشان داد. نکته دیگری که به نظر می‌رسد این است که با بزرگ شدن هرچه بیشتر مدار میزبان، روش هوشمند ما ارزش خود را بیشتر نشان می‌دهد.

فصل ۵

نتیجه‌گیری

در این فصل، ضمن جمع‌بندی نتایج جدید ارائه‌شده در پایان‌نامه، مسائل باز باقی‌مانده و همچنین پیش‌نهادهایی برای ادامه‌ی کار ارائه می‌شوند.

۵ـ۱ جمع‌بندی نتایج بدست آمده

تروا‌های سخت‌افزاری مدارهایی با عملیات بداندیشانه هستند که ممکن است به مدار اصلی افزوده شوند. رویکردهای بسیاری برای جلوگیری از اثرات مخرب تروا‌ها وجود دارد. ما در این پژوهش بر دو رویکرد کشف تروا از طریق آزمون منطقی و اثرات جانبی تمرکز کردیم. در بخش آزمون منطقی، روش آزمونی ارائه و پیاده‌سازی کردیم که در مقابل روش‌های آزمون تصادفی مرسوم، با حفظ نرخ کشف، زمان بسیار کمتر و بهتری ارائه می‌دهد. به عبارت دیگر، تعداد بردارهای آزمون به میزان ۸۰ تا ۹۰ درصد کاهش یافت. سپس با استفاده از همین بردارهای هوشمند، آزمون اثرات جانبی را انجام دادیم. در بخش آینده تاثیر استفاده از آزمون بردارهای هوشمند را بر آزمون اثرات جانبی، به عنوان دستاورد اوّل این پروژه مرور نهایی خواهیم کرد. در نهایت، در فصل پیش اندازه‌ی تروجان را به عنوان یک پارامتر تعیین کننده مورد مطالعه قرار دادیم. ما از آزمایش‌هایی که روی ۱۰ مدار از ISCAS-۸۵ صورت گرفت، به حد آستانه‌ای برای انتخاب بهترین روش آزمون بین دو روش منطقی و اثرات جانبی رسیدیم. در ادامه این حد را مرور می‌کنیم.

مدل مدار	بردار هوشمند	بردار تصادفی
c۴۳۲	۹۹	۹۹
c۴۹۹	۹۸	۹۸
c۸۸۰	۹۸	۹۸
c۱۳۵۵	۹۷	۹۴
c۱۹۰۸	۹۸	۹۵
c۲۶۷۰	۹۶	۸۸
c۳۵۴۰	۹۳	۸۶
c۵۳۱۵	۹۲	۸۳
c۶۲۸۸	۹۲	۸۰
c۷۵۵۲	۸۶	۷۷

جدول ۵ـ۱: مقایسه آزمون اثرات جانبی با بردارهای هوشمند و تصادفی (میانگین نرخ کشف تروا)

۵ـ۱ـ۱ یک آزمون منطقی و اثرات جانبی بهتر

با توجه به جدول ۵ـ۱، مقایسه شد که آزمون اثرات جانبی با بردارهای هوشمند، به طور میانگین ۱۰ درصد و تا ۳۵ درصد نتایج بهتری از آزمون اثرات جانبی با بردارهای تصادفی به دست می‌دهد. منظور از نتایج بهتر، نرخ کشف تروا بالاتر، در زمان برابر است. همچنین در فصل قبل مشاهده شد،الگوریتم نولید بردار آزمون منطقی ما، تعداد بردارها را حدوداً به یک دهم بردارهای تصادفی کاهش می‌دهد.

۵ـ۱ـ۲ مشاهده تاثیر اندازه تروا در نتیجه آزمون

بررسی‌ها و به طور خلاصه ۴ـ۲ نشان داد هر چه تروا‌ها کوچکتر باشند، آزمون منطقی کاراتر و هرچه بزرگتر باشند، بهتر است از آزمون اثرات جانبی استفاده کنیم. در مدارهای ISCAS-۸۵ مرز اندازه نسبی ۰/۱ درصد برای انتخاب روش برتر به دست آمد.

۵ـ۲ مسائل باز و کارهای آتی

۵ـ۲ـ۱ آزمون خودکار اندازه‌آگاه

ما در این پژوهش مرزی برای انتخاب نوع آزمون برای اندازه تروا بدست آوردیم. حال اگر شبیه‌ساز خود را بگونه‌ای برنامه‌ریزی کنیم که بعد از محاسبه اندازه نسبی تروا، فقط یک رویکرد آزمونِ بهتر را انجام دهد، آزمون خودکاری داریم که به احتمال زیاد سرعت میانگین، دقت و پیچیدگی آن برای هر تعداد از تروا‌ها، به ترتیب بیشتر، بیشتر و کمتر خواهد بود.(به نسبت هر کدام از آزمون‌های اثرات جانبی و منطقی)

۵ـ۲ـ۲ محل فرضی تروا

ما در این پروژه بنا به تحقیقاتی که در فصل ۳ بررسی شد، فرض کردیم گره‌هایی تروا‌خیز هستند که کمترین فعالیت را در ازای ورودی‌های تصادفی دارند. سپس، توانستیم بردارهای هوشمند، هدفمند و کوتاه‌تری را به دست آوریم. در واقع هدف هر الگوریتم هوشمندی، کاهش فضای نمونه برای سریع‌تر به جواب رسیدن است. اما سوالی که به ذهن متبادر می‌شود این است که آیا فرضی که از سال ۲۰۱۲ درباره‌ی محل احتمالی تروا صورت گرفته است، در سال‌های آتی هم درست خواهد ماند؟ به نظر می‌رسد بررسی فاکتورهای آماری دیگری درباره‌ی محل قرارگیری تروا، بررسی روند طی شده، و برون‌یابی موقعیت تروا در آینده، می‌توان زمینه‌ی یک تحقیق آماری و با ارزش باشد.

۵ـ۲ـ۳ مدل مدار و تروا

ما مجموعه آزمون‌هایمان را روی تروا مدل XOR-payload AND-trigger که ترکیبی می‌باشد انجام دادیم؛ محدودیت دیگر این پژوهش در استفاده از مدارهای میزبان ترکیبی بود. یک موضوع داغ برای ادامه این پژوهش، مدل‌سازی تروا‌های ترتیبی، و آزمایش روی مدارهای میربان ترتیبی مانند ISCAS-۸۹ و مشاهده شباهت‌ها و تفاوت‌ها در رفتار آن مدارها می‌باشد.

۵ـ۲ـ۴ ایجاد فعالیت نسبی بیشتر برای تروا‌ها

در تولید بردارهای هوشمند، ما بر این نکته تمرکز کردیم که تا جای ممکن، گره‌های تروا‌خیز را تحریک کنیم. یک راه حل جایگزین این است که گره‌های بی تروا را (هم) ثابت و بی‌فعالیت نگاه داریم. این روش جالب به نظر می‌رسد، زیرا فاکتوری که مهم است، فعال‌تر بودن نسبیِ گره‌های تروا‌خیز است. به نظر می‌رسد این افزایش فعالیت تروا‌ها، بهبود قابل قبولی در نتایج آزمون به ارمغان خواهد آورد.

۵ـ۲ـ۵ افزایش دقت شبیه‌سازی

در آزمون‌های جانبی، هرچه پارامترهای شبیه‌سازی جامع‌تر باشند، نتایج آن به واقعیت نزدیک‌تر خواهد بود. ما در این پروژه تنها نویز تغییرات فرایند را در نظر گرفتیم. به عبارت دیگر، فرض کردیم نویز اندازه‌گیری و محیط صفر باشند. اضافه کردن هریک از این دو، ارزیابی دقیق‌تری را به همراه خواهد داشت.

۵ـ۲ـ۶ آزمون واقعی

این پروژه به دلیل در اختیار نداشتن تجهیزات آزمون، محدود به شبیه سازی بود. بدیهی است که نتایج بدست آمده از آزمون واقعی به مراتب موثق‌تر خواهند بود.

مراجع

[1] Defense Science Board Task Force. High performance microchip supply. Technical report, 2005.

[2] United States Department of Commerce Bureau of Industry and S. O. of Technology Evaluation. Defense industrial base assessment : Counterfeit electronics. Technical report, 2010.

[3] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. volume 27, pages 10–25. IEEE, 2010.

[4] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using ic fingerprinting. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 296–310. IEEE, 2007.

[5] R. Rad, J. Plusquellic, and M. Tehranipoor. A sensitivity analysis of power signal methods for detecting hardware trojans under real process and environmental conditions. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(12):1735–1744, 2010.

[6] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic. Hardware trojan detection and isolation using current integration and localized current analysis. In *Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS'08. IEEE International Symposium on*, pages 87–95. IEEE, 2008.

[7] Y. Jin and Y. Makris. Hardware trojan detection using path delay fingerprint. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 51–57. IEEE, 2008.

[8] J. Li and J. Lach. At-speed delay characterization for ic authentication and trojan horse detection. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 8–14. IEEE, 2008.

[9] S. Jha. Randomization based probabilistic approach to detect trojan circuits. In *High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE*, pages 117–124. IEEE, 2008.

[10] M. Banga and M. S. Hsiao. A region based approach for the identification of hardware trojans. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 40–47. IEEE, 2008.

[11] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty. Towards trojan-free trusted ics: Problem analysis and detection scheme. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1362–1365. ACM, 2008.

[12] H. Salmani, M. Tehranipoor, and J. Plusquellic. A novel technique for improving hardware trojan detection and reducing trojan activation time. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 20(1):112–125, 2012.

[13] X. Wang, M. Tehranipoor, and J. Plusquellic. Detecting malicious inclusions in secure hardware: Challenges and solutions. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 15–19. IEEE, 2008.

[14] M. Abramovici and P. Bradley. Integrated circuit security: new threats and solutions. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, page 55. ACM, 2009.

[15] L.-W. Kim, J. D. Villasenor, and C. K. Koç. A trojan-resistant system-on-chip bus architecture. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*, pages 1–6. IEEE, 2009.

[16] H. Salmani, M. Tehranipoor, and J. Plusquellic. New design strategy for improving hardware trojan detection and reducing trojan activation time. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*, pages 66–73. IEEE, 2009.

[17] M. Banga and M. S. Hsiao. Vitamin: Voltage inversion technique to ascertain malicious insertions in ics. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*, pages 104–107. IEEE, 2009.

[18] R. S. Chakraborty and S. Bhunia. Security against hardware trojan through a novel application of design obfuscation. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, pages 113–116. ACM, 2009.

[19] Y. Jin, N. Kupp, and Y. Makris. Dftt: design for trojan test. In *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on*, pages 1168–1171. IEEE, 2010.

[20] S. Drzevitzky, U. Kastens, and M. Platzner. Proof-carrying hardware: Towards runtime verification of reconfigurable modules. In *Reconfigurable Computing and FPGAs, 2009. ReConFig'09. International Conference on*, pages 189–194. IEEE, 2009.

[21] E. Love, Y. Jin, and Y. Makris. Enhancing security via provably trustworthy hardware intellectual property. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 12–17. IEEE, 2011.

[22] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor. Trustworthy hardware: Identifying and classifying hardware trojans. *Computer*, 43(10):39–46, 2010.

[23] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic. Power supply signal calibration techniques for improving detection resolution to hardware trojans. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 632–639. IEEE Press, 2008.

[24] R. S. Chakraborty, S. Narasimhan, and S. Bhunia. Hardware trojan: Threats and emerging solutions. In *High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International*, pages 166–171. IEEE, 2009.

[25] Z. Chen, X. Guo, R. Nagesh, A. Reddy, M. Gora, and A. Maiti. Hardware trojan designs on basys fpga board. *Embedded System Challenge Contest in Cyber Security Awareness Week-CSAW*, 2008, 2008.

[26] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson. Trojan side-channels: Lightweight hardware trojans through side-channel engineering. In *Cryptographic Hardware and Embedded Systems-CHES 2009*, pages 382–395. Springer, 2009.

[27] D. Rai and J. Lach. Performance of delay-based trojan detection techniques under parameter variations. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*, pages 58–65. IEEE, 2009.

[28] R. Torrance and D. James. Reverse engineering in the semiconductor industry. In *Custom Integrated Circuits Conference, 2007. CICC'07. IEEE*, pages 429–436. IEEE, 2007.

[29] J. Aarestad, D. Acharyya, R. Rad, and J. Plusquellic. Detecting trojans through leakage current analysis using multiple supply pad s. *Information Forensics and Security, IEEE Transactions on*, 5(4):893–904, 2010.

[30] Y. Alkabani and F. Koushanfar. Consistency-based characterization for ic trojan detection. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, pages 123–127. ACM, 2009.

[31] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. Hardware trojan horse detection using gate-level characterization. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, pages 688–693. IEEE, 2009.

[32] M. Banga and M. S. Hsiao. A novel sustained vector technique for the detection of hardware trojans. In *VLSI Design, 2009 22nd International Conference on*, pages 327–332. IEEE, 2009.

[33] H. Salmani, M. Tehranipoor, and J. Plusquellic. A layout-aware approach for improving localized switching to detect hardware trojans in integrated circuits. In *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on*, pages 1–6. IEEE, 2010.

[34] D. Du, S. Narasimhan, R. S. Chakraborty, and S. Bhunia. Self-referencing: a scalable side-channel approach for hardware trojan detection. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 173–187. Springer, 2010.

[35] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia. Multiple-parameter side-channel analysis: a non-invasive hardware trojan detection approach. In *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, pages 13–18. IEEE, 2010.

[36] M. Tehranipoor and C. Wang. *Introduction to hardware security and trust*. Springer, 2012.

[37] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. Mero: A statistical approach for hardware trojan detection. In *Cryptographic Hardware and Embedded Systems-CHES 2009*, pages 396–410. Springer, 2009.

[38] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. In *Proceedings of the 40th annual Design Automation Conference*, pages 338–342. ACM, 2003.

[39] I. Pomeranz and S. M. Reddy. A measure of quality for n-detection test sets. *Computers, IEEE Transactions on*, 53(11):1497–1503, 2004.

[40] R. S. Chakraborty, S. Paul, and S. Bhunia. On-demand transparency for improving hardware trojan detectability. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pages 48–50. IEEE, 2008.

[41] J. Rajendran, V. Jyothi, O. Sinanoglu, and R. Karri. Design and analysis of ring oscillator based design-for-trust technique. In *VLSI Test Symposium (VTS), 2011 IEEE 29th*, pages 105–110. IEEE, 2011.

[42] X. Zhang, A. Ferraiuolo, and M. Tehranipoor. Detection of trojans using a combined ring oscillator network and off-chip transient power analysis. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 9(3):25, 2013.

[43] G. Bloom, B. Narahari, and R. Simha. Os support for detecting trojan circuit attacks. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*, pages 100–103. IEEE, 2009.

[44] M. Hicks, M. Finnicum, S. T. King, M. Martin, and J. M. Smith. Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 159–172. IEEE, 2010.

[45] D. McIntyre, F. Wolff, C. Papachristou, S. Bhunia, and D. Weyer. Dynamic evaluation of hardware trust. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*, pages 108–111. IEEE, 2009.

[46] M. S. Hsiao and M. Tehranipoor. On trust in third-party hardware ips.

[47] B. Mathew and D. G. Saab. Combining multiple dft schemes with test generation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 18(6):685–696, 1999.

واژه‌نامه

الف

آزمون اثرات جانبی ..... side-channel test

آزمون منطقی ..... logic test

ابتکاری ..... heuristic

استراتژی ..... strategy

ب

بارگذاری ..... loading

برچسب ..... label

بسته‌بندی ..... packing

بهترین پاسخ ..... best response

بیشینه ..... maximum

پ

پایداری ..... robustness

پشتیبان ..... support

نرخ کشف ..... coverage rate

پوششی ..... covering

ت

تروای سخت افزاری ..... hardware trojan

تقاطع ..... intersection

تقسیم‌بندی ..... partition

توزیع‌شده ..... distributed

ح

حرکت ..... action

د

دودویی ..... binary

ر

رفتار ..... behaviour

ز

زمان‌بندی ..... scheduling

س

ساختی ..... constructive

سخت‌افزار حامل اثبات .. proof-carrying hardware

سود ..... pay off, utility

ع

عمل ..... action

ق

قوی ..... strong

ک

کمینه ..... minimum

م

مجموعه ..... set

منطقی ..... logical

موازی ..... parallel

ن

نتیجه‌ی نهایی ..... outcome

Abstract

With constant increase in the rate of VLSI circuits manufactured in sites separate from the designers and computer architects, global concern regarding the possibility of integration of malware by the manufacturing foundries has arisen. Particularly, one main issue that affects reliability of the chips is modifications or additions with malicious intention, known as Hardware Trojans, which are easily applicable during design and manufacturing phase of chips. There has been an increasing fraud in chip-set manufacturing. Hardware Trojans may leak confidential information outside the chip, to the attacker, may alter the function of circuit, or completely fail a system.

Hence search for new Trojan Detection methods is absolutely essential. Almost all the present methods are restricted, in that they are suitable only for small Trojans or the gigantic ones. This project strives to fill the gap, by introducing a combined size-aware approach, which is well-suited to striking a balance between tiny and very large Systems-on-Chip. Comparable in speed, our approach is able to offer higher accuracy than its predecessors at the expense of a more complex test design.

Keywords: Hardware Trojan Detection, Reliability, System-on-Chip, VLSI

Sharif University of Technology

Department of Computer Engineering

M.Sc. Thesis

Hardware Trojan Detection: A Size-aware Approach

By:

Seyed Behnam Heydarshahi

Supervisor:

Dr. Shaahin Hessabi

May 2015