دانشگاه آزاد اسلامی

واحد علوم و تحقیقات سیرجان

سمینار کارشناسی ارشد مهندسی کامپیوتر

گرایش نرم افزار

موضوع :

پیاده سازی سیستم های چندعامله تحمل پذیر خطا برروی سخت افزارهای قابل پیکربندی

استاد راهنما:

جناب آقای دکتر خطیبی

نگارش:

حمیدرضا شول

تابستان93

چکیده

بسیاری از سیستم‌های چند عامله[1] تطبیق‌پذیر[2] کنونی توسط سخت افزارهای قابل‌پیکربندی مانند FPGA[3] پیاده‌سازی می شوند. زیرا که فرآیند پیاده‌سازی بسیار با سرعت انجام می‌شود و سخت افزار این سیستم‌ها می‌تواند با زمان تغییر کند، بنابراین استفاده از سخت افزارهای قابل-پیکربندی یک ایده بسیار عالی برای پیاده‌سازی سیستم‌های چند عامله تطبیق‌پذیراست. همچنین یک سخت افزارهای قابل‌پیکربندی قطعه‌ای قابل پیکربندی مجدد است که می‌تواند هر مدار دیجیتال را پیاده‌سازی کند. بنابراین با استفاده از یک سخت افزارهای قابل‌پیکربندی می‌توان در یک بستر سیلیکون انواع مدارات دیجیتال ترکیبی و ترتیبی را به صورت تطبیق‌پذیر ایجاد نمود. برای پیاده‌سازی یک کاربرد[4] به صورت چند عامله بر روی سخت افزارهای قابل-پیکربندی، جریان بیت[5] آن که توسط ابزارهای CAD تولید شده است در حافظه پیکربندی سخت افزارهای قابل‌پیکربندی قرار می‌گیرد. اما در هر نسل از تکنولوژی CMOS به منظور دستیابی به سرعت و تراکم بالا، ترانزیستورها در ابعاد کوچکتری ساخته می شوند. یکی از مشکلاتی را که این موضوع به دنبال دارد مسئله نرخ خطای نرم است که باعث تغییر یک بیت داده در حافظه پیکربندی می شود. مسئله اصلی رخ داد این نوع از خطاها پدیده Ionizing Radiation است. یکی از عوامل اصلی این پدیده پرتوهای کیهانی و ذرات باردار با انرژی بالا هستند،که این پرتوها و ذرات باردار دارای مقدار زیادی از انرژی هستند و می توانند باعث تغییر بار الکتریکی در نقاط مختلف یک مدار الکترونیکی شوند. از آنجایی که در هر نسل از تکنولوژیCMOS ابعاد نانو ولتاژ تغذیه مدار کاهش می یابد، بنابراین ظرفیت گره‌های خازنی نقاط مختلف در مدارات حافظه کاهش می یابد با کمترین مقدار انرژی وارد شده توسط پدیده Ionizing Radiation داده‌های سلول‌های حافظه عوض می شوند. در نتیجه اگر داده‌های پیکربندی یک سیستم چند عامله در سخت افزار قابل پیکربندی در اثر خطاهای نرم عوض شود سیستم چند عامل پیاده سازی شده خراب[6] خواهد شد. بنابراین نرخ خطای نرم چالش اساسی در پیاده‌سازی سیستم‌های چند عامله بر روی سخت افزارهای قابل‌پیکربندی مانند FPGA در تکنولوژی CMOS با ابعاد نانو هستند. در این طرح تحقیقاتی در راستاری بهبود عملکرد سخت افزارهای قابل‌پیکربندی در تکنولوژی CMOS با ابعاد نانو، از دو موضوع که در سخت افزارهای قابل‌پیکربندی مانند FPGA مشاهده می‌شوند استفاده شده است. ۱- در جریان بیت بیشتر کاربردهایی که بر روی سخت افزارهای قابل‌پیکربندی پیاده‌سازی می شوند، تعداد صفر ها از یک ها بسیار بیشتراست. ۲- سلول‌های حافظه پیکربندی در مسیر انتشار سیگنال‌ها در درون منابع منطقی و مسیر یابی قرار ندارند. به این ترتیب به منظور کاهش نرخ خطای نرم، سلول‌هایی برای حافظه پیکربندی سخت افزارهای قابل‌پیکربندی طراحی شده است که این سلول‌ها دارای نرخ خطای کمی هستند زیرا تمامی گره‌های حساس این سلول‌ها

[1] Multi Agent System
[2] Adaptive
[3] Field Programmable Gate Arrays
[4] Application
[5] Bit-Stream
[6] Failure

در مقابل برخورد ذرات سخت شده‌اند و برخورد ذرات نمی‌تواند داده این سلول ها را عوض کند. همچنین در این سلول‌ها جریان نشتی با استفاده از دو ولتاژ آستانه برای ترانزیستورها و پشته-ای از ترانزیستورهای خاموش سری[۷] کاهش یافته است.

کلمات کلیدی: سیستمهای چند عامله، سخت‌افزارهای قابل‌پیکربندی، سلول حافظه پیکربندی، بر خورد ذرات انرژی‌دار، جریان نشتی، مساحت سلول، بار بحرانی، نرخ خطای نرم.

[7] Stacked Effect

فهرست مطالب

عنوان صفحه

فهرست مطالب

فهرست مطالب

فصل اول

مقدمه و هدف

۱-۱-پیشگفتار

یک سخت افزار قابل پیکربندی عبارت است از یک مجموعه از منابع سخت افزاری قابل پیکربندی مجدد که با یک دیگر در ارتباط هستند و می‌توانند انواع کاربردها را روی آنها به صورت سخت افزار پیاده-سازی کرد[۱،۲]. یک FPGA یک تراشه سخت افزاری است که بر مبنای ایده سخت فزارهای قابل پیکربندی ساخته شده است[۱،۲]. بنابراین در حال حاضر یک FPGA یک نمونه پیشرفته و واقعه‌ای از یک سخت افزار قابل پیکربندی است[۱،۲]. در نتیجه در این طرح تحقیقاتی منظور از یک FPGA همان سخت افزار قابل پیکربندی است و تمامی ایده‌هایی که در این طرح تحقیقاتی برای سخت افزارهای قابل پیکربندی ارائه می‌شوند بر روی FPGA اعمال می شوند.

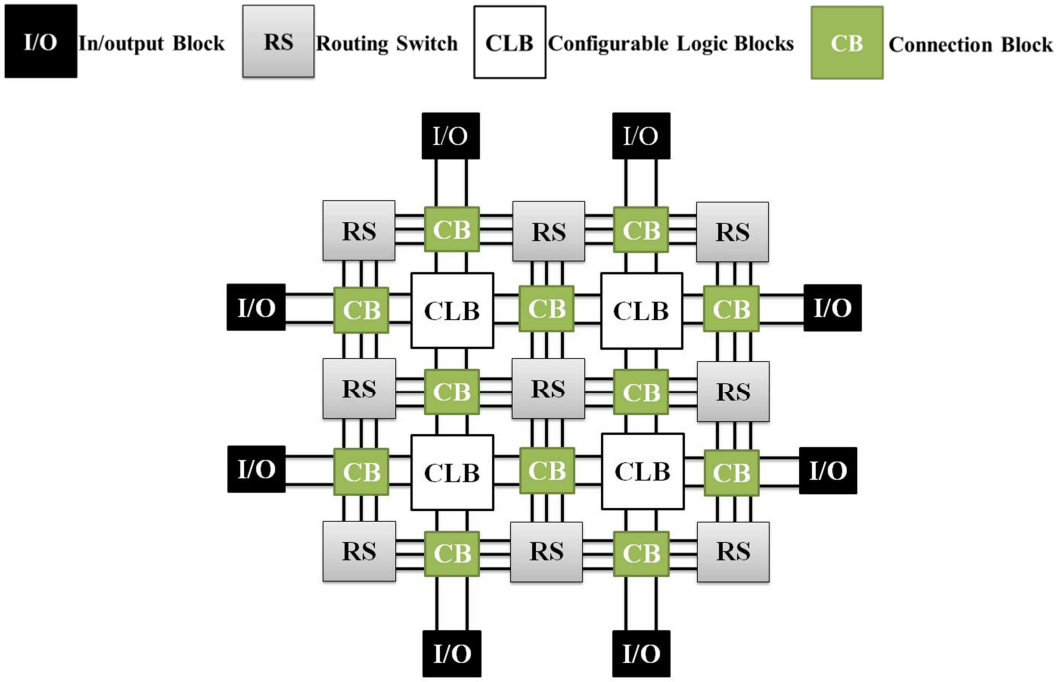
بسیاری از سیستم‌های دیجیتال کنونی توسط FPGA پیاده سازی می شوند[۱]. زیرا که فرآیند پیاده-سازی در FPGA می تواند بسیار با سرعت انجام می شود و سخت افزار این سیستم ها می تواند با زمان تغییر کند. بنابراین استفاده از FPGA یک ایده بسیار عالی برای پیاده سازی سیستم های دیجیتال تطبیق پذیراست[۱،۲]. همچنین یک FPGA قطعه‌ای قابل پیکربندی مجدد است که می‌تواند هر مدار دیجیتال را پیاده‌سازی کند. بنابراین با استفاده از یک FPGA می‌توان در یک بستر سیلیکون انواع مدارات دیجیتال ترکیبی و ترتیبی را ایجاد نمود[۱].

شکل ۱-۱ معماری جزیره‌ای یک FPGA را نمایش می‌دهد. همانطور که از این شکل مشخص است، هر FPGA از تعداد زیادی بلوک‌های‌منطقی‌قابل پیکربندی[1] تشکیل شده است. هر یک از این بلوک‌های‌منطقی-قابل پیکربندی می‌توانند قسمتی از یک مدار دیجیتال را پیاده‌سازی کند[۱]. به علاوه در یک FPGA تعداد زیادی سوئیچ‌های مسیریابی[2] قابل برنامه ریزی وجود دارد، که ارتباط بین قسمت های مختلف را پیاده سازی

[1] Configurable Logic Block یا CLB

[2] Routing Switch

می کنند[۱]. ارتباط یک FPGA با محیط بیرون از طریق بلوک های ورودی و خروجی[۱] انجام می شود. این بلوک های طوری هستند که می توان آنها را به صورت ورودی یا خروجی پیکربندی کرد [۱]. البته در FPGA های پیشرفته قطعاتی نیز به صورت جاسازی شده وجود دارد که این قطعات شامل حافظه ها یا واحد های محاسباتی هستند که در کاربرد های گوناگون می توان از آنها استفاده کرد.



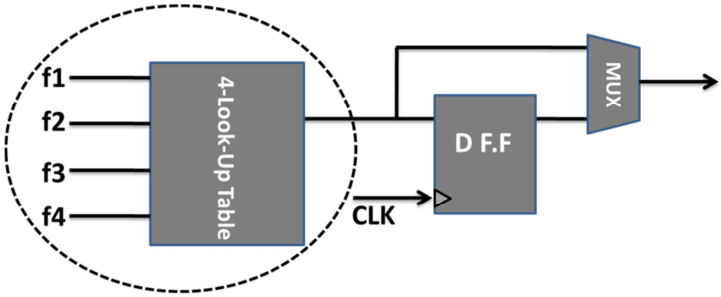
شکل۱-۱: معماری جزیره‌ای یک FPGA [۳]

۱-۲- ساختار یک بلوک منطقی قابل پیکربندی(CLB)

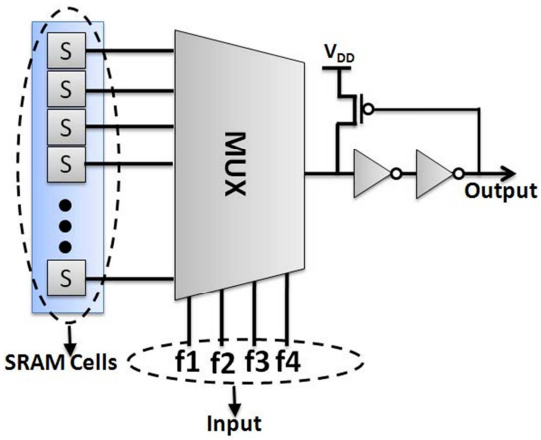
در شکل ۱-۲ ساختار کلی یک CLB نمایش داده شده است. بیشتر FPGA های تجاری از جدول های جستجوی[۲] با ۴ ورودی (LUT-۴) در CLB های خود استفاده می‌کنند[۴]. یک جدول جستجوی ۴ ورودی یک حافظه کوچک است که می تواند انواع توابع دیجیتال ترکیبی ۴ ورودی را پیاده سازی کند. فلیپ فلاپ موجود در یک CLB برای پیاده سازی مدارات دیجیتال ترتیبی استفاده می شود. همچنین در شکل ۲-۳

[۱] In/output Block
[۲] Lookup Table (LUT)

جزئیات پیاده سازی یک LUT-۴ نمایش داده شده است. سلول‌های SRAM[۱] در یک LUT، جدول درستی یک تابع دیجیتال را نگهداری می‌کنند و ورودی های تابع دیجیتال مورد نظر به ورودی های MUX داده می شوند، بنابراین براساس یک بردار ورودی که به ورودی تابع دیجیتال اعمال می‌شود، محتویات یکی از سلول های SRAM انتخاب می شود و به خروجی LUT انتقال داده می شود. همچنین به این نکته توجه شود که در یک CLB ممکن است چندین LUT و فلیپ فلاپ وجود داشته باشد به عنوان مثال در هر CLB موجود در FPGA Virtex™-II PRO شرکت Xilinx تعداد هشت LUT و هشت فلیپ فلاپ وجود دارد[۴].



شکل ۱-۲: ساختار کلی یک بلوک منطقی قابل پیکربندی (CLB) و یک جدول جستجو (LUT)



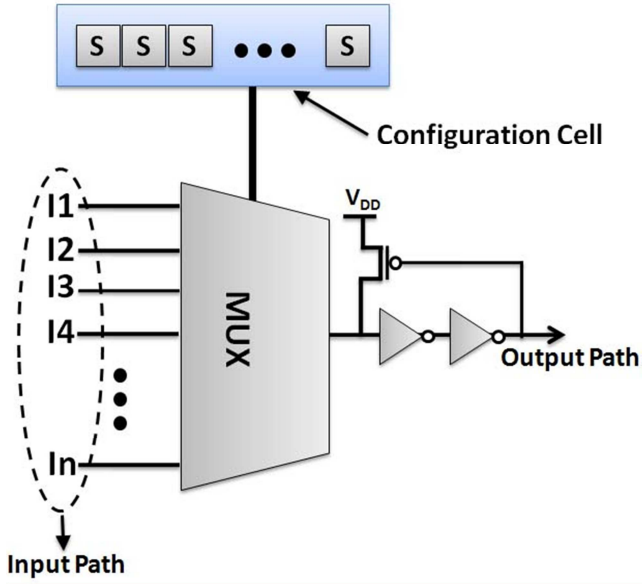
شکل ۱-۳: ساختار کلی یک جدول جستجوی (Look-Up Table) چهار ورودی [۳]

۱-۳- ساختار یک سوئیچ مسیریابی

ارتباط بین CLB های و سایر قطعات جاسازی شده در FPGA توسط یک شبکه بین ارتباطی قابل-پیکربندی ایجاد می شود. این شبکه بین ارتباطی قابل‌پیکربندی توسط یک سوئیچ قابل‌برنامه‌ریزی پیاده

[۱] Static Random Access Memory

سازی می شود. شکل ۱-۴ ساختار کلی یک سوئیچ مسیریابی قابل برنامه ریزی را نمایش می‌دهد. ورودی های MUX که با I1 تا In نام‌گذاری شده اند توسط MUX و بافر خروجی به ورودی یک CLB یا ورودی یک سوئیچ دیگر مسیر یابی می‌شوند. قابلیت پیکربندی در این سوئیچ توسط سلول های SRAM ایجاد می شود، این سلول ها مشخص می‌کنند که چه ورودی به خروجی سوئیچ مسیریابی شود.



شکل۱-۴: ساختار کلی یک سوئیچ مسیر یابی قابل‌پیکربندی [۳]

۱-۴- قطعات جاسازی شده

در بسیار از FPGA ها مدرن حافظه ها و واحد های محاسباتی به صورت جاسازی شده وجود دارند[۱]. وجود این قطعات جاسازی شده باعث افزایش کارایی کاربرد های پیاده سازی شده بر روی FPGA ها می شود. این قطعات جاسازی شده از طریق شبکه قابل پیکربندی به سایر قسمت‌های FPGA وصل می شوند و می توانند داده ها را از طریق شبکه قابل پیکر بندی دریافت کنند و بعد از انجام وظیفه درخواست شده داده های پردازش شده را به شبکه قابل پیکر بندی ارسال کنند[۱]. به عنوان مثال فرض کنید که می‌خواهید یک کاربرد پردازش تصور را بر روی FPGA پیاده سازی کنید. در این حالت وجود یک پردازشگر سیگنال‌های

دیجیتال[1] (DSP) جاسازی شده بسیار موثر است، زیرا داده های وارد شده به FPGA می توانند به صورت مستقیم به DSP ارسال شوند و مورد پردازش قرار گیرند. همچنین در بسیاری از FPGA ها حافظه های SRAM به صورت جاسازی شده وجود دارد که از این SRAM ها می توان در ذخیره سازی داده ها استفاده کرد. بنابراین وجود قطعات جاسازی شده در داخل یک FPGA باعث سهولت پیاده سازی کاربرد ها بر روی FPGA می شود.

۱-۵- پیکربندی FPGA

همانطور که در قسمت های قبلی گفته شد یک سخت افزار قابل پیکربندی[2] مانند FPGA شامل منابع منطقی و مسیریابی است، که عملکرد آنها توسط حافظه های قابل برنامه ریزی کنترل می شوند. مقادیر دودویی که در درون این حافظه ها نگهداری می شود تعیین کننده عملکرد منابع قابل برنامه ریز در درون یک FPGA هستند. فرآیند بار کردن مقادیر دودویی در درون حافظه های کنترلی قابل برنامه ریزی را بازپیکربندی[3] گویند[۵]، و یک دنباله مشخص از صفر ها و یک ها برای یک مکان مشخص از حافظه کنترلی به منظور پیاده سازی یک مدار خاص را پیکربندی[4] گویند[۵]. داده های پیکربندی (دنباله از صفر ها و یک ها) خود توسط نرم افزاری های CAD بر اساس مداری که باید یر روی سخت افزار قابل پیکربندی پیاده سازی شود تولید می شوند .

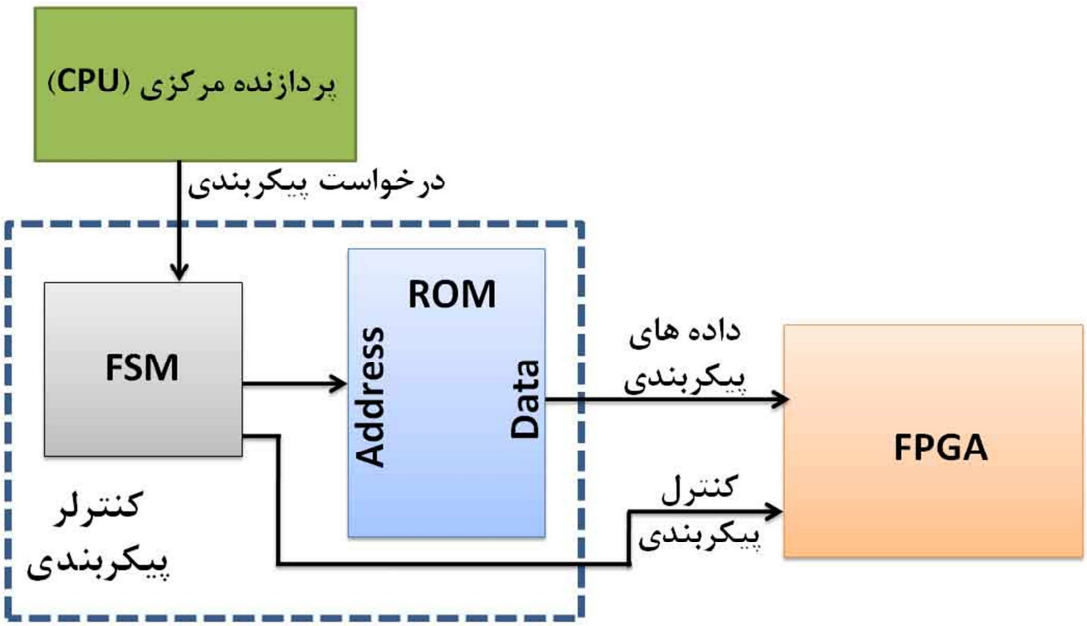
به طور کلی داده های پیکربندی در درون یک حافظه در خارج از FPGA نگهداری می شوند. در بسیاری از موارد داده های پیکربندی در درون حافظه اصلی ذخیره می شوند و CPU[5] داده‌های پیکربندی را از حافظه اصلی به FPGA انتقال می دهد[۵]. در مواردی دیگر داده های پیکربندی در درون یک ROM[6] ذخیره می شوند و توسط یک کنترل کننده پیکربندی داده‌های پیکربندی مستقیماً به FPGA منتقل می شود، در این حالت معمولاً درخواست بازپیکربندی توسط CPU صادر می شود و حافظه ROM و کنترل

[1] Digital Signal Proccessor
[2] Configurable Hardware
[3] Reconfiguration
[4] Configuration
[5] Central Processing Unit
[6] Read Only Memory

کننده پیکربندی در درون یک تراشه ساخته می شوند [۵]. همچنین کنترل کننده پیکربندی معمولاً توسط یک ماشین حالت متناهی[۱] پیاده سازی می شود شکل ۲-۵ ساختار این روش را نمایش می دهد.

۱-۵-۱- معماری های پیکربندیFPGA

معماری پیکربندی عبارت است از مدارات و سخت افزار هایی، که توسط این مدارات و سخت افزار ها می توان داده های مربوط به پیکربندی را در به داخل حافظه های پیکربندی انتقال داد. معماری های پیکربندی دارای پیچیدگی های متفاوتی هستند به شکلی که می توانند یک ثبات اتقالی ساده باشند یا یک ساختار قابل آدرس‌دهی باشند به طوری که بعد از انتقال داده ها بتوان داده‌های مورد نظر را دستکاری کرد. در ادامه چند معماری مرسوم پیکربندی معرفی خواهد شد.

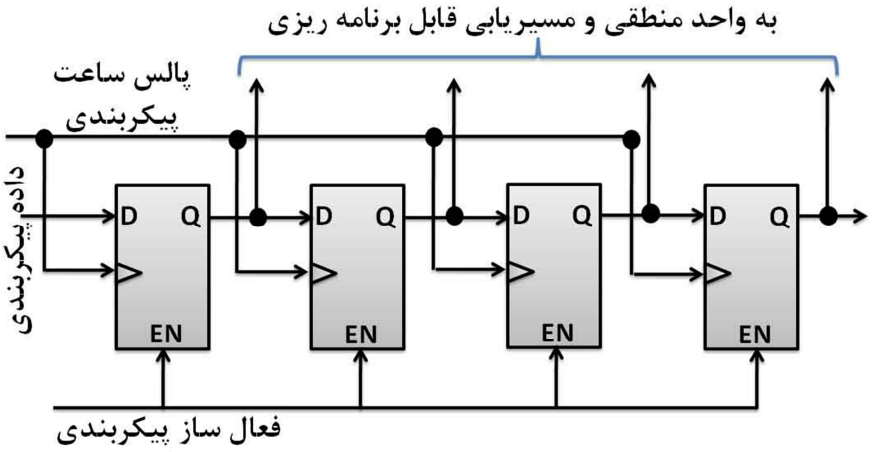


شکل ۱-۵: ساختار مرسوم برای ذخیره سازی داده های پیکربندی و انتقال آن به FPGA

[۱] Finite State Machine

۱-۵-۲- معماری Single-Context

این معماری یکی از معماری‌های مرسوم است که در بسیاری از FPGA های تجاری استفاده می شود[۵]، شکل ۱-۶ ساختار این معماری را نشان می‌دهد. در این معماری داده‌های پیکربندی به صورت سریال و به وسیله یک زنجیره از ثبات‌های انتقالی به داخل FPGA بار می شوند. مزیت این معماری این است که در فرآیند پیکربندی از تعداد پایه‌های کمی از FPGA استفاده می‌شود. اما عیب بزرگ این معماری این است هر تغییر در داده‌های پیکربندی مستلزم پیکربندی کل FPGA است، زیرا که در این معماری امکان تغییرداده های پیکربندی به صورت انتخابی وجود ندارد. بنابراین کوچکترین تغییر در داده های پیکربندی مستلزم پیکربندی کل FPGA و در نتیجه تأخیر ناشی از آن است.



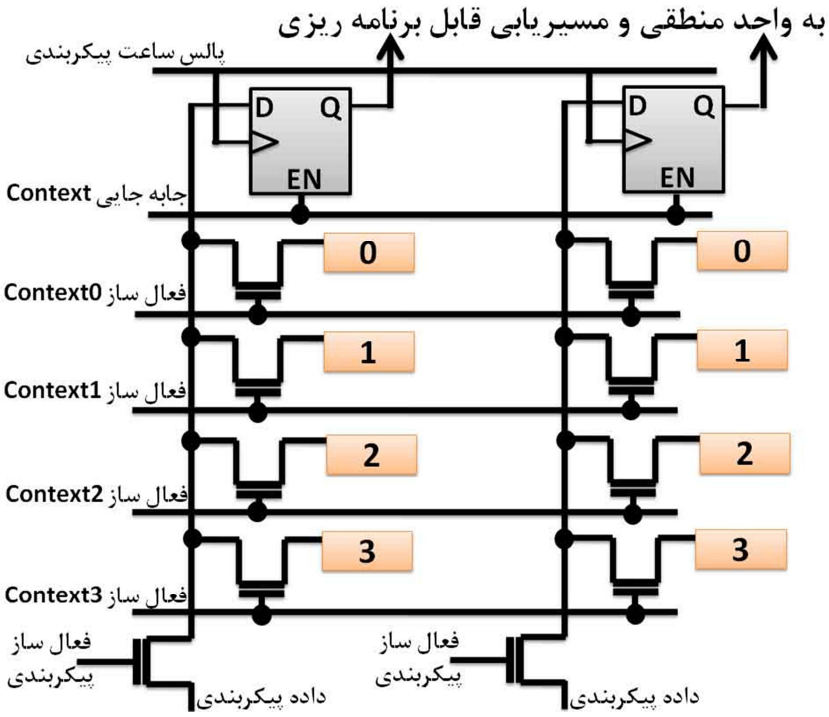
شکل۱-۶: معماری Single-Context

۱-۵-۳- معماریMulti-Context

همانطور که قبلاً ذکر شد مشکل معماری Single-Context تأخیر بسیار زیاد آن است. در معماری Multi-Context این امکان فراهم می‌شود که چندین پیکربندی برای منابع منطقی و مسیریابی قابل برنامه ریزی وجود داشته باشد. به این ترتیب در معماری Multi-Context در طول یک پالس ساعت می‌توان به پیکربندی دیگری جابه‌جا شد. ساختار این معماری در شکل ۱-۷ نمایش داده شده است. این شکل دو نقطه

پیکربندی با چهار Context را نمایش می دهد. همانطور که از شکل ۱-۷ مشخص است این معماری دارای دو مزیت عمده است. ۱- در این معماری امکان بار گذاری یک پیکربندی جدید در حین کارکرد FPGA وجود دارد به طوری که بارگذاری پیکربندی جدید هیچ وقفه‌ای در کارکرد FPGA به وجود نمی آورد[۵]. ۲- امکان جابه‌جا بین Context ها در یک پالس ساعت وجود دارد، البته این موضوع زمانی درست است که پیکربندی مورد نظر از قبل در یکی از Context ها ذخیره شده باشد. در غیر این صورت زمان جابه‌جایی از یک پیکربندی به دیگری همانند حال Single-Context خواهد بود[۵].

از عیب های این معماری سربار سخت افزاری زیاد آن است. بطوری که FPGA هایی که از این معماری پیکربندی استفاده می کنند دارای ظرفیت پایینی هستند. یکی دیکر از عیب‌های این معماری توان مصرفی پویا آن است. در حالت جابه‌جایی از یک Context به دیگری تعداد زیادی از نقاط پیکربندی به طور همزمان از صفر به یک یا برعکس تغییر مقدار می دهند که این باعث افزایش ضربه‌ای توان مصرفی پویا می شود و این افزایش ضربه‌ای ممکن است به خطوط تغذیه تراشه آسیب برساند[۵].



شکل ۱-۷: معماری Multi-Context

۱-۵-۴- معماری Partially Reconfiguration

در اکثر مواقع بازپیکربندی یک FPGA شامل یک قسمت کم از کل منابع FPGA می شود، بنابراین اگر فقط قسمتی از منابع که باید تغییر کنند باز پیکربندی شوند زمان بازپیکربندی به شدت کاهش می یابد. در حقیقت یک معماری پیکربندی Partially همانند یک حافظه RAM[۱] معمولی قابل آدرس دهی است[۵].

خانواده Virtex از FPGA های شرکت Xilinx دارای معماری Partially و معماری Single Context هستند و منابع قابل‌برنامه‌ریزی به شکل ستون هایی در FPGA سازماندهی شده اند[۶]. در این خانواده از FPGAها، حافظه پیکربندی به بلوک های قابل آدرس دهی با نام Frame تقسیم می شوند[۶]. هر یک از این Frame ها برای یک قسمت از منابع قابل‌برنامه‌ریزی در یک ستون از FPGA در نظر گرفته شده اند[۶]. در این خانواده از FPGA ها دو ثبات وجود دارد که عبارتند از FAR[۲] و FDIR[۳] در طول بازپیکربندی ثبات FDIRداده های پیکربندی را نگهداری می‌کند و ثبات FAR مشخص کننده آدرسی Frame است که داده های پیکربندی باید در آن Frame نوشته شوند[۶]. در این خانواده از FPGA ها در حالتی که پیکربندی به صورت Single-Context است FAR از مقدار صفر شروع می شود و به صورت اتوماتیک با هر بار نوشتن داده‌های ثبات FDIR در یک Frame افزایش می یابد[۶]. اگرچه معماری پیکربندی Partially از انعطاف-پذیری مناسبی برخوردار است اما به دلیل ساختار قابل آدرس‌دهی دارای سربار سخت افزاری نسبتاً زیادی است[۵]. همچنین در مواردی ارسال آدرس بلوک‌های قابل پیکربندی زمان زیادی از پیکربندی را به خود اختصاص می دهد به طوری که زمان پیکربندی در حالت Partially از حالت Single-Context زیاد تر شود[۵].

۱-۶- پیاده سازی کاربرد ها بر روی FPGA

به هر عملکرد مورد نظر کاربر[۴] یک کاربرد گفته می شود[۷]. هر کاربرد را می توان به چند وظیفه[۱] تقسیم کرد، بطوری که هر وظیفه یک قسمت از کاربرد را انجام می دهد و نتایج حاصل از یک وظیفه به

---

[۱] Random Access Memory
[۲] Frame Address Register
[۳] Frame Data Input Register
[۴] User

سایر وظایف دیگر کاربرد ارسال می شود[۷]. بنابراین می توان گفت که یک کاربرد عبارت است از مجموعه-ای از وظایف که برای اجرای کاربرد، مجموعه وظایف کاربرد با یک دیگر در حال تعامل هستند[۷].

یک FPGA مجموعه‌ای از منابع سخت افزار مستقل است، که این منابع سخت افزاری قابلیت محاسبات و ذخیره سازی را دارند و این منابع سخت افزاری از طریق یک شبکه ارتباطی با یک دیگر در حال ارتباط هستند[۷]. بنابراین می توان وظایف یک کاربرد را بر روی یک FPGA به صورت موازی با هم اجرا کرد که این باعث افزایش سرعت اجرای کاربردها روی FPGA می شود[۷]. یک روش برای پیاده سازی یک وظیفه بر روی سخت افزارهای قابل پیکربندی مانند FPGA استفاده از یک عامل سخت افزاری[۲] است[۲]. بنابراین برای پیاده سازی یک کاربرد روی سخت افزار های قابل پیکربندی مانند FPGA از سیستم های چند عامله[۳] استفاده می شود[۲].

۱-۶-۱- سیستم های چند عامله

یک سیستم چند عامله مجموعه ای از نهاد ها است که با یک دیگر در حال تعامل هستند به صورتی که این تعامل منجر به حل یک مسئله یا رسیدن به یک هدف خاص است[۲]. سیستمهای چند عامله زمانی مورد استفاده قرار می‌گیرند که هدف یا حل مسئله توسط یک نهاد قابل دست یافتن یا حل شدن نیست[۲]. یک عامل[۴] قابلیت جمع آوری داده را از محیط بیرون توسط ورودی های خود دارد و می تواند داده‌های دریافت شده را پردازش کند و بر روی محیط بیرون تأثیر بگذارد[۷]. سیستم‌های چند عامله دارای مزایای متعددی نسبت به سیستم‌های متمرکز هستند. به عنوان مثال سیستم های چند عامله بسیار انعطاف‌پذیرتر و مقیاس‌پذیرتر هستند و با پیچیده تر شدن مسئله می توان به آنها عامل‌های بیشتری اضافه کرد[۲]. همچنین در مقایسه با سیستم های متمرکز، سیستم های چند عامله دارای موازی سازی و در نتیجه سرعت بیشتری هستند زیرا که در یک سیستم چند عامله عناصر به صورت موازی با یکدیگر عمل می کنند[۲]. و سرانجام

---

[۱] Task
[۲] Hardware Agent
[۳] Multi Agent Systems
[۴] Agent

در مقایسه با سیستم های متمرکز، سیستم های چند عامله نسبت به خطا تحمل پذیرتر هستند. به این معنی که با خراب شدن یک عامل، سیستم با کیفیت کمتری می تواند به کار خود ادامه دهد[۲].

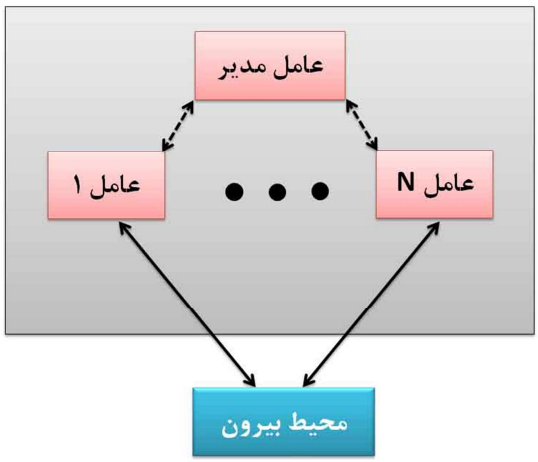
بنابراین سیستم های چند عامله یک رویکرد توزیع شده را برای حل یک مسئله دنبال می کنند. بسیاری از کاربرد امروزه دارای یک رویکرد چند عامله هستند، مانند کابرد های سیستم های کنترلی، کاربرد های پردازش صدا و تصویر، کاربرد های رمز نگاری و امنیت اطلاعات وSensor Fusion [۷]. معمولاً عامل‌ها توسط نهاد های نرم افزاری پیاده سازی می شوند. اما با پیشرفت تکنولوژی سخت‌افزارهای قابل‌پیکربندی این امکان فراهم شده است که سیستم های چند عامله در محیط متشکل از نهادهای سخت افزاری و نرم افزاری پیاده سازی شوند[۲،۷]. در چند سال اخیر تکنولوژی آرایه‌های گیت قابل برنامه ریزی میدانی (FPGA) که بر مبنای SRAM ساخته می شوند به عنوان یک چهارچوب سخت افزاری قابل پیکر بندی معرفی شده اند[۲،۵،۷]. به طوری که بسیاری از سیستم های دیجیتال کنونی توسط FPGA پیاده سازی می شوند زیرا فرآیند پیاده سازی بسیار با سرعت انجام می شود و سخت افزار این سیستم ها می توان با زمان تغییر کند[۱،۲]. بنابراین استفاده از FPGA یک ایده بسیار عالی برای پیاده سازی سیستم های دیجیتال چند عامله تطبیق‌پذیراست. همچنین یک FPGA قطعه ای قابل پیکر بندی مجدد است که می‌تواند هر مدار دیجیتال(یک عامل سخت افزاری) را پیاده سازی کند. بنابراین با استفاده از یک FPGA می توان در یک بستر سیلیکون انواع سیستم های دیجیتال چند عامله را ایجاد نمود.

به طور کلی تاکنون ۳ معماری کلی برای پیاده سازی سیستم های چند عامله بر روی سخت افزارهای قابل پیکربندی ارائه شده است که در زیر آورده شده اند.

Static Reconfiguration -۲-۶-۲

در این معماری تمام عامل های سیستم در ابتدای شروع عملکرد کاربرد بر روی FPGA پیکربندی می-شوند و یک عامل نیز به صورت عامل مدیر انتخاب می‌شود که این عامل مدیر عامل‌های فعال را در طول کاربرد مشخص می کند[۲]. شکل ۱-۸ این معماری را نشان می دهد. از مزیت های اصلی معماری ذکر شده

این است که هیچ زمانی صرف پیکربندی عامل‌ها نمی شود، زیرا که کل عامل‌های مورد نیاز کاربرد در زمان شروع کاربرد بر روی FPGA پیکربندی شده‌اند و تنها تأخیر موجود، تأخیر انتخاب عامل‌ها توسط عامل مدیر است[۲]. عیب عمده این معماری سربار سخت افزاری زیاد آن است، زیرا در این معماری کل عامل‌های کاربرد باید در ابتدا شروع کار کاربرد بر روی FPGA پیکربندی شوند حتی اگر به آنها نیازی نباشد[۲].

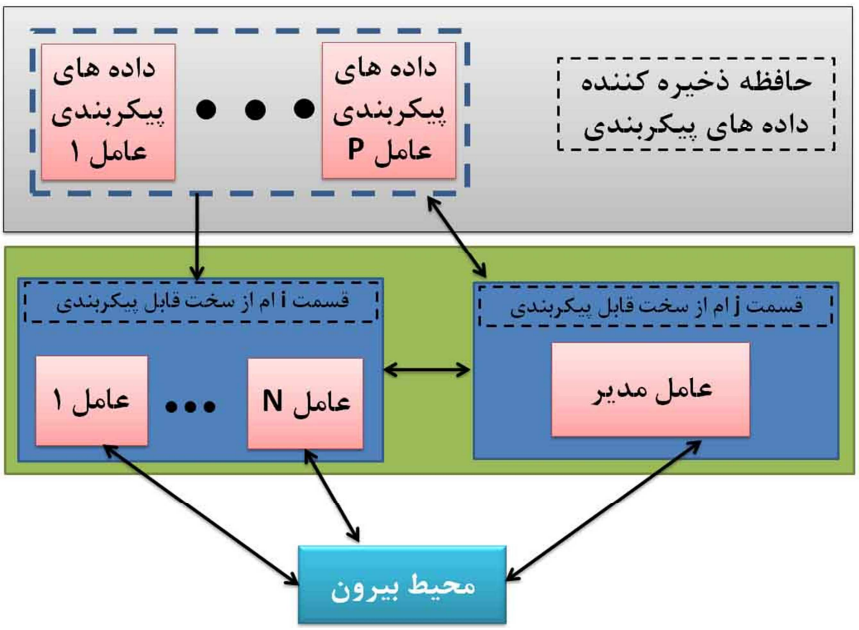


شکل ۱-۸ : معماری Static Reconfiguration

Non-Buffered Dynamic Reconfiguration-۳-۶-۲

در این معماری بر خلاف معماری قبلی در شروع کار یک زیر مجموعه از کل عامل‌های یک کاربرد که برای شروع کار کاربرد نیاز هستند بروی FPGA پیکربندی می‌شوند و سایر عامل ها که فعلاً به آنها نیازی نیست در حافظه نگداری می‌شوند و بر روی FPGA پیکربندی نمی‌شوند[۲]. شکل ۱-۹ ساختار این معماری را نمایش می‌دهد. در این معماری یک عامل مدیر وجود دارد که به طور پیوسته عامل هایی را که بر روی FPGAپیکربندی شده‌اند را مانیتور می‌کند و بر اساس شرایط محیط بیرون و وضعیت کاربرد، در صورت نیاز کاربرد به یک عامل جدید یک عامل که بر روی FPGA پیکربندی شده است و دیگر نیازی به آن نیست را با عامل جدید مورد نیاز جایگزین می کند[۲]. بنابراین FPGA یا سخت افزار قابل پیکربندی که این معماری بر روی آن پیاده سازی می شود باید قابلیت Partial Configuration را پشتیبانی کند[۲].

مزیت اصلی این معماری این است که در پیاده سازی کاربرد از منابع FPGA بصورت بهینه استفاده می شود زیرا فقط عامل هایی بر روی FPGA پیکربندی می شوند که در اجرای کاربرد به آنها نیاز است[۲،۶]. اما عیب عمده این معماری تأخیر ناشی از پیکربندی و جایکزین کردن یک عامل جدید با عامل قدیمی است و این موضوع کاربرد این معماری را در کاربردهای Real Time محدود می کند[۲،۷].

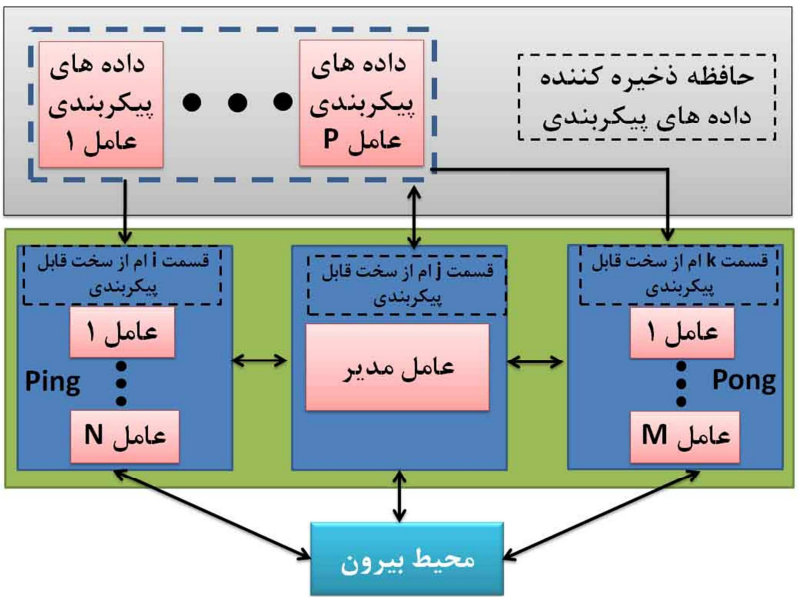


شکل ۱-۹ : معماری Non –Buffered Dynamic Reconfiguration

Buffered Dynamic Reconfiguration -۴-۶-۲

در این معماری دو قسمت از FPGA برای پیاده سازی کاربرد در نظر گرفته می‌شود یکی قسمت Ping و دیگری قسمت Pong است، و در هر لحظه یکی از این دو قسمت توسط عامل مدیر انتخاب می شود[۲]. شکل ۱-۱۰ ساختار این معماری را نمایش می دهد. در حقیقت این معماری یک Trade-off بین دو معماری قبلی است. نحوه عملکرد این معماری به این صورت است که در شروع کار یک زیر مجموعه از کل عامل های یک کاربرد که برای شروع کار کاربرد نیاز هستند بروی قسمت Ping پیکربندی می‌شوند و سایر عامل ها که فعلاً به آنها نیازی نیست در حافظه نگهداری می شوند[۲]. در این معماری یک عامل مدیر وجود دارد که به طور پیوسته عامل‌هایی را که بر روی قسمت فعال(Ping یا Pong) پیکربندی شده‌اند را مانیتور

می کند و بر اساس شرایط محیط بیرون و وضعیت کاربرد در صورت نیاز کاربرد به یک عامل جدید، عامل جدید و عامل های جاری را بر روی قسمت غیر فعال پیکربندی می کند و سپس قسمتی که با عامل جدید و عامل های قبلی پیکرندی شده است را فعال می کند[۲]. از مزیت اصلی این معماری این است که بدون وقفه به کار خود ادامه دهد زیرا در زمانی که قسمت غیر فعال در حال پیکربندی است قسمت فعال در حال انجام کار خود است[۲].

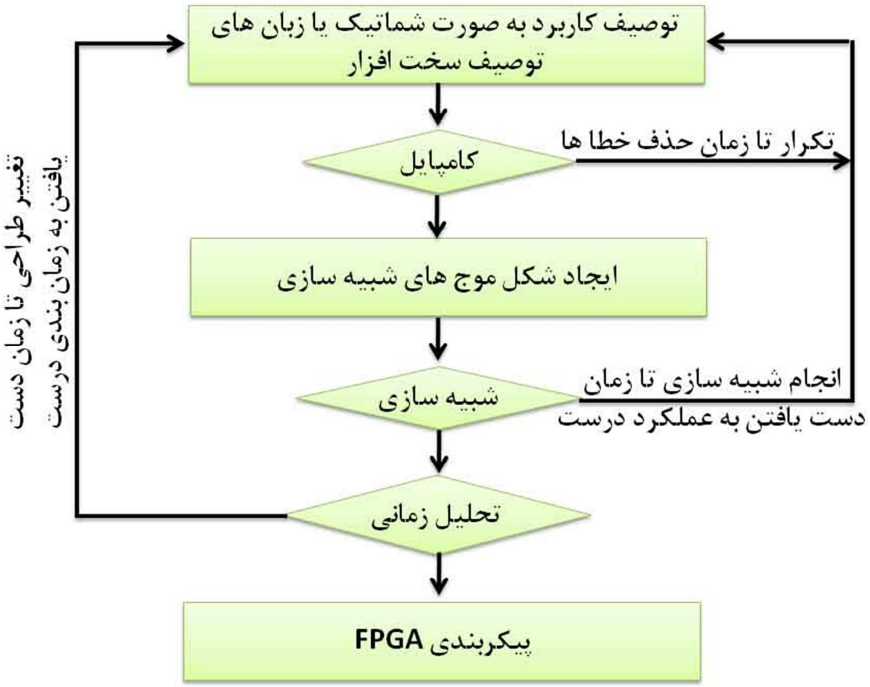


شکل ۱-۱۰: Buffered Dynamic Reconfiguration

۱-۷- فرآیند پیاده سازی کاربرد ها بر رویFPGA

شکل ۱-۱۱ فرآیند طراحی و پیاده سازی یک کاربرد بر روی یک FPGA را نمایش می دهد. در طراحی و پیاده سازی یک کاربرد بر روی یک FPGA ابزار های CAD نقش بسیار کلیدی و مهمی را دارند به طور که اکثر مراحل طراحی به صورت اتومانیک توسط این ابراز ها انجام می شود[۵]. همانطور که از شکل ۱-۱۱ مشخص است اولین مرحله در پیاده‌سازی یک کاربرد بر روی FPGA وارد کردن کابرد به ابزارهای CAD است. در این مرحله می توان کاربرد را با روش هایی مانند طراحی شماتیک یا استفاده از زبان های توصیف سخت افزاری مانند VHDL یا Verilog وارد ابزار های CAD کرد[۵]. در مرحله بعد طراحی وارد شده توسط ابزار CAD کامپایل می شود، این عمل تازمانی که خطایی در طراحی کاربرد وجود نداشته باشد تکرار

می شود. پس از این مرحله شکل موج های ورودی لازم مشخص می شوند و سپس در فرآیند شبیه سازی به کاربرد تحت بررسی اعمال می شوند[۵]. در صورتی که عملکرد مورد نظر از کاربرد، در حین فرآیند شبیه سازی مشاهد نشود، طراح باید کاربرد را اصلاح کند تا عملکرد مورد انتظارحاصل شود. سپس کاربرد از دیدگاه زمانبدی که باید بر روی FPGA داشته باشد مورد بررسی قرار می گیرد. همانند مرحله قبل اگر طراحی زمانبندی‌های مورد انتظار را برآورده نکند طراحی باید تغییر کند[۵]. پس از عبور موفق از این مرحله طراحی توسط ابزار CAD برروی FPGA مقصد سنتز می شود. در این مرحله یک فایل از رشته صفر ها و یک ها[۱] تولید می شود. این فایل در حقیقت داده های پیکربندی هستند که باید منابع موجود در FPGAتوسط آن پیکربندی شوند تا کاربرد مورد نظر بر روی FPGA پیاده سازی شود[۵]. پس از تولید این فایل توسط ابزار CAD با استفاده از یکی از معماری های پیکربندی معرفی شده داده های پیکربندی به داخل FPGA انتقال داده می شوند.



شکل ۱-۱۱: فرآیند طراحی و پیاده سازی یک کاربرد بر روی FPGA

[۱] Bit-stream File

۱-۸- نرخ خطاهای نرم

با پیشرفت تکنولوژی CMOS ابعاد ترانزیستورها در حال کوچکتر شدن است و ولتاژ تغذیه تراشه‌ها برای دست یافتن به توان مصرفی کمتر نیز در حال کاهش است[۸]. یکی از مشکلاتی را که این موضوع به دنبال دارد مسئله نرخ خطای نرم است. مسئله اصلی رخ داد این نوع از خطاها پدیده Ionizing Radiation است. از عوامل اصلی این پدیده ذرات انرژی‌دار مانند ذره α، پرتوهای کیهانی هستند، که این پرتوها و ذرات حاوی مقدار زیادی از انرژی هستند و می توانند باعث تغییر بار الکتریکی در نقاط مختلف یک مدار الکترونیکی شوند[۸،۹]. از آنجایی که در تکنولوژی‌های CMOS با ابعاد نانو ولتاژ تغذیه مدار کاهش می یابد و ظرفیت گره های خازنی نقاط مختلف در مدارات حافظه کاهش می یابد با کمترین مقدار انرژی وارد شده توسط پدیده Ionizing Radiation داده های سلول های حافظه عوض می شوند[۸]. بنابراین برخورد ذرات باردار انرژی دار با سلول های حافظه باعث عوض شدن داده های سلول های حافظه می شوند. در ادامه این موضوع به صورت کامل بررسی می شود.

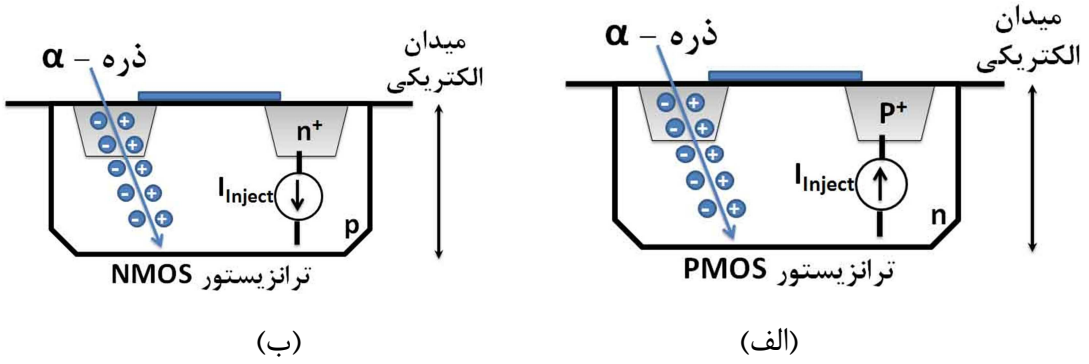
۱-۸-۱- برخورد ذرات باردار با ترانزیستورها

زمانی که یک ذره به ناحیه سورس یا درین یک ترانزیستور خاموش برخورد می کند، یک جریان گذرا بین سورس-بدنه یا درین بدنه ایجاد می شود. بنابراین برحسب نوع ترانزیستور دو حالت را می توان در نظر گرفت.

۱- زمانی که ذره به ناحیه سورس یا درین یک ترانزیستور خاموش PMOS برخورد می کند. در این حالت با نفوذ ذره انرژی دار در عماق بدنه، انرژی خود را در برخورد با بدنه از دست می دهد و باعث ایجاد یکسری حامل های اقلیت الکترون و حفره می شود. که این حامل های اقلیت تحت میدان الکتریکی موجود بین سورس-بدنه یا درین-بدنه باعث ایجاد یک جریان گذرا بین سورس-بدنه یا درین-بدنه می شود[۳،۱۰]. شکل ۱-۱۲-الف این مطلب را درمورد یک ترانزیستور PMOS نشان می دهد.

۲- در این حالت زمانی که ذره به ناحیه سورس یا درین یک ترانزیستور خاموش NMOS برخورد می کند. در این حالت با نفوذ ذره انرژی دار در عماق بدنه ترانزیستور، انرژی خود را در برخود با اتم های سازنده بدنه از دست می دهد و باعث ایجاد یکسری حامل های اقلیت الکترون و حفره می شود. که این حامل های اقلیت

تحت میدان الکتریکی موجود بین سورس-بدنه یا درین-بدنه باعث ایجاد یک جریان گذرا بین سورس-بدنه یا درین-بدنه می شود[۳،۱۰]. شکل ۱-۱۲-ب این مطلب را درمورد یک ترانزیستور NMOS نشان می دهد.



شکل ۱-۱۲ : برخورد یک ذره α با یک ترانزیستور (الف)- ترانزیستور PMOS (ب)- ترانزیستور NOMS

بنابراین تأثیر برخورد یک ذره به ناحیه سورس یا درین یک ترانزیستور را می توان توسط یک منبع جریان با رابطه زیر مدل کرد[۱۰،۱۱].

$$I(t) = I_0(e^{-\frac{t}{\tau_\alpha}} - e^{-\frac{t}{\tau_\beta}}) \quad (۱)$$

که در اینجا I_0 به میزان بار تزریق شده در برخورد ذره بستگی دارد و τ_α و τ_β دو ثابت زمانی در برخورد ذره با سطح سیلیکون هستند.

۱-۸-۲- نرخ خطای نرم

زمانی که یک ذره انرژی دار به یک گره در یک سلول حافظه برخورد می کند، باعث تزریق یک جریان ناگهانی در گره ای که ذره به آن برخورد کرده می شود. که این جریان ناخواسته ممکن است باعث عوض شدن داده سلول شود. بار بحرانی[۱] یکی از پارامترهای مهمی است که نشان دهنده میزان حساسیت یک سلول نسبت به برخورد ذرات انرژی دار با آن است[۸]. بار بحرانی به صورت کمترین مقدار بار تزریق شده توسط برخورد ذره به سلول که باعث عوض شدن داده سلول می شود تعریف می شود[۸]. اگر $I(t)$ جریان تزریق شده توسط برخورد ذره با گره سلول حافظه باشد بار بحرانی از رابطه زیر بدست می آید[۸].

$$Q_{Crit} = \int_0^{t_f} I(t)dt \quad (۲)$$

[۱] Critical Charge

که در رابطه بالا τ_f بازه زمانی است که ذره به گره حساس برخورد می کند تا زمانی که داده سلول عوض می شود.

نرخ خطای نرم در برخورد ذرات انرژی دار با گره حساس i یک سلول حافظه از رابطه زیر بدست می آید[۱۲].

$$SER_i = \frac{WOVi}{T_{clk}} k_i \int_{Q_{Crit(i)}}^{\infty} Prob(Q).dQ \quad (۳)$$

که WOV_i پنجره زمانی آسیب پذیری است و به صورت باز زمانی تعریف می شود که در این بازه اگر ذره با انرژی کافی به گره حساس i برخورد کند داده سلول حافظه عوض خواهد شد و $Prob(Q)$ تابع توزیع احتمال جمع آوری بار در برخورد ذره است. بنابراین انتگرال در رابطه ۳ برابر با احتمال جمع آوری باری بیشتر از بار بحرانی، در برخورد ذره با گره i است. در نتیجه اگر در یک سلول حافظه تعداد n گره حساس وجود داشته باشد نرخ خطای نرم برای این سلول حافظه از رابطه زیر بدست می آید[۱۲].

$$SER_i = \sum_{i=1}^{n} \frac{WOVi}{T_{clk}} k_i \int_{Q_{Crit(i)}}^{\infty} Prob(Q).dQ \quad (۴)$$

۱-۸-۳- خطاهای نرم در FPGA

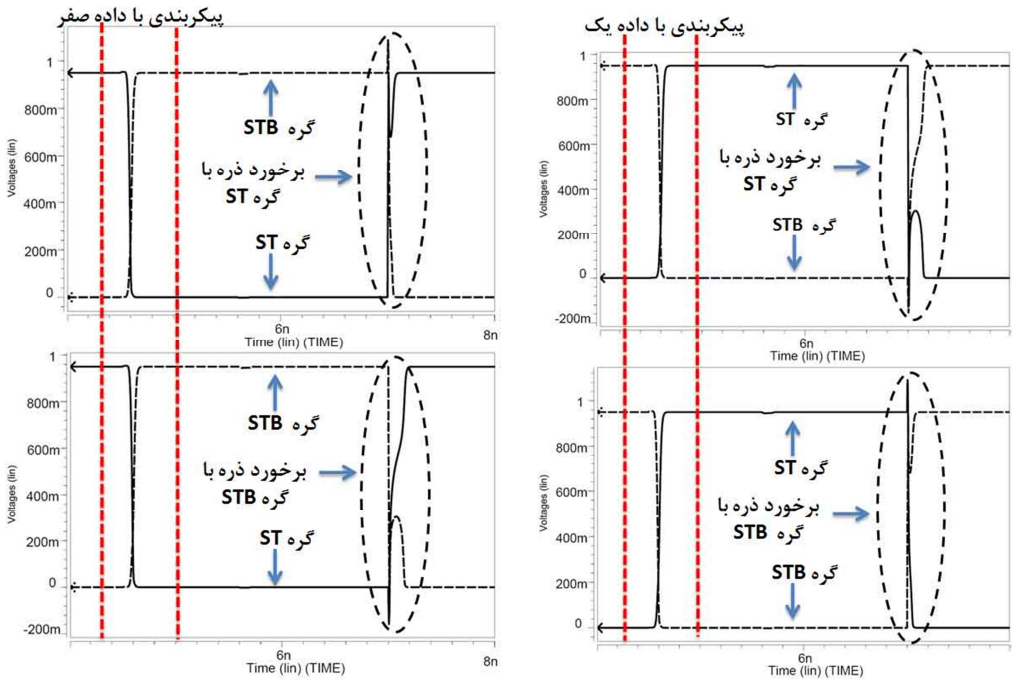
در مدارات CMOS با ابعاد نانو، از آنجایی که ابعاد کوچک هستند، گره ها در سلول‌های حافظه، دارای ظرفیت خازنی کمی هستند و ولتاژ تغذیه نیز کاهش یافته بنابراین مقدار بار بحرانی برای سلول‌های حافظه در تکنولوژی CMOS با ابعاد نانو کاهش یافته است و ذرات با انرژی کم نیز قادر به عوض کردن داده های سلول هستند[۸]. در نتیجه نرخ خطاهای نرم در تکنولوژی CMOS با ابعاد نانو در حال افزایش است[۱۲]. شکل ۱-۱۳ شکل موج برخورد یک ذره با گره های حساس یک سلول حافظه شش ترانزیستوری که توسط HSPICE را در تکنولوژی ۲۲ نانو متر شبیه سازی شده را نمایش می دهد. همانطور که از این شکل مشخص است در برخورد یک ذره با گره‌های حساس سلول داده سلول عوض شده است. این تغییرات ناخواسته در مقادیر حافظه را SEU[۱] می‌گویند. بطور کلی اثرات ناشی از خطاهای نرم بر روی یک FPGA می توان به دو دسته تقسیم کرد[۹،۱۳]:

۱- خطاهای گذرا

۲- خطاهای ماندگار.

[۱] Single Event Upset

در حقیقت SEU هایی که بر روی فلیپ فلاپ ها و حافظه های مدار پیاده سازی شده بر روی FPGA رخ می دهند از نوع خطاهای گذرا هستند. زیرا که این خطاها در پالس های ساعت بعدی توسط مقادیر جدید درست باز نوشته می شوند و اثر آنها از بین می رود[۹،۱۳]. اما SEU هایی که در حافظه های پیکربندی یک FPGA رخ می دهد از نوع خطا های ماندگار می باشد (توجه شود که این نوع خطاها با خطاهای فیزیکی موجود در FPGA متفاوت هستند). زیرا که این نوع از خطا ها در حافظه پیکربندی رخ می دهد و باعث عوض شدن عملکرد مدار پیاده سازی شده بر روی FPGA می شود، این نوع از خطا ها تا زمانی که یک FPGA دوباره پیکربندی نشود بر روی FPGA ماندگار هستند[۹،۱۳]. بنابراین این نوع از خطا ها را با یک پیکربندی مجدد می توان از بین برد[۹،۱۳]. بنابراین با توجه به کوچکتر شدن ابعاد و کاهش ولتاژ تغذیه در تکنولوژی های CMOS با ابعاد نانو و بیشتر شدن نرخ خطاهای نرم این موضوع به یک چالش بزرگ در FPGA تبدیل شده است.



شکل ۱-۱۳: شکل موج شبیه سازی شده در برخورد ذره با گره های حساس سلول شش ترانزیستوری پایه در حالتی که داده یک و صفر در سلول ذخیره شده است

فصل دوم

مواد و روش کار

۲-۱- مقدمه

همانطور که در قسمت ۱-۸ توضیح داده شد، در اثر برخورد ذرات انرژی‌دار با ترانزیستورها یک تراشه، یک سری پالس‌های ناخواسته در گره‌هایی که ذرات برخورد کرده‌اند ایجاد می شود. معمولاً در مدارات ترکیبی یک FPGA (Multiplexer های موجود در یک LUT یا یک سوئیچ مسیریابی) در بدترین حالت این پالس‌های ناخواسته همزمان با کلاک فلیپ فلاپ‌ها به ورودی داده فلیپ فلاپ ها می‌رسند و توسط فلیپ فلیپ‌ها دریافت و ذخیزه می‌شوند[۹،۱۳]. این نوع از خطاها را در یک FPGA همانطور که قبلاً گفته شد خطاهای گذرا می گویند، زیرا که این خطاها در پالس های ساعت بعدی توسط مقادیر جدید درست باز نوشته می شوند و اثر آنها از بین می رود[۹،۱۳]. اما خطاهای نرمی که در حافظه‌های پیکربندی یک FPGA رخ می دهد از نوع خطا های ماندگار می باشد. زیرا که این نوع از خطا ها در حافظه پیکربندی رخ می دهد و باعث عوض شدن عملکرد مدار پیاده سازی شده بر روی FPGA می شود، این نوع از خطا ها تا زمانی که یک FPGA دوباره پیکربندی نشود بر روی FPGA ماندگار هستند[۹،۱۳]. بنابراین خطاهای نرم در یک FPGA فقط در حافظه پیکربندی مسئله ساز هستند.

با پیشرفته تکنولوژی CMOS با ابعاد نانو و کوچکترن شدن ابعاد ترانزیستورها، نرخ خطاهای نرم در حال افزایش است. بنابراین با توجه به توضیحات بالا خطاهای نرمی که در حافظه پیکربندی FPGA رخ می-دهند مسئله ساز هستند. در نتیجه برای اینکه سیستم‌های چند عامله که بر روی سخت افزارهای قابل پیکربندی پیاده‌سازی می‌شوند تحمل پذیر خطا باشند، باید از سلول‌هایی در حافظه پیکربندی سخت افزارهای قابل پیکربندی استفاده کنیم که داده‌ی آنها در اثر برخورد ذرات انرژی دار داده عوض نشود.

در این طرح تحقیقاتی در راستاری بهبود عملکرد افزارهای قابل پیکربندی مانند FPGA، از دو موضوع که در FPGA مشاهده می‌شوند استفاده شده است. ۱- در جریان بیت[۱] بیشتر کاربردهایی که بر روی FPGA پیاده سازی می شوند، تعداد صفر ها از یک ها بسیار بیشتراست. به طوری که در جریان بیت بیشتر کاربردها ۸۷ درصد داده ها صفر هستند[۱۴،۱۵]. دلیل اصلی این تعداد زیاد صفر در جریان بیت کاربردها،

[۱] bit-stream

زیاد بودن تعداد بیت‌های بدون استفاده در حافظه پیکربندی منابع مسیریابی است[۱۴،۱۵]. ۲- واقعیت دیگری که در یک FPGA مشاهده می شود این است که تأخیر زمانی پیکربندی سلول‌های حافظه پیکربندی بر روی سرعت FPGA تأثیر گذار نیستند[۱۶]. زیرا که سلول‌های حافظه پیکربندی در مسیر انتشار سیگنال‌ها در درون منابع منطقی و مسیر یابی قرار ندارند[۱۶]. در ادامه سلول‌های جدیدی ارائه خواهند شد، که در طراحی این سلول های جدید از این دو موضوع که در FPGA مشاهده می شود استفاده خواهیم کرد.

در این طرح تحقیقاتی جهت حذف خطاهای نرم در سیستم‌های چند عامله‌ای که بر روی سخت افزارهای قابل پیکربندی پیاده‌سازی می‌شوند، دو سلول برای حافظه پیکربندی و حافظه‌های جاسازی شده طراحی خواهیم کرد. در ادامه نحوه عملکرد این دو سلول در حافظه پیکربندی و حافظه‌های جاسازی شده توضیح داده خواهند شد و برخورد ذرات انرژی دار، جریان نشتی، تغییرات فرآیند ساخت و چگونگی استفاده آنها در FPGA و حافظه های جاسازی شده مورد بررسی قرار خواهد گرفت.

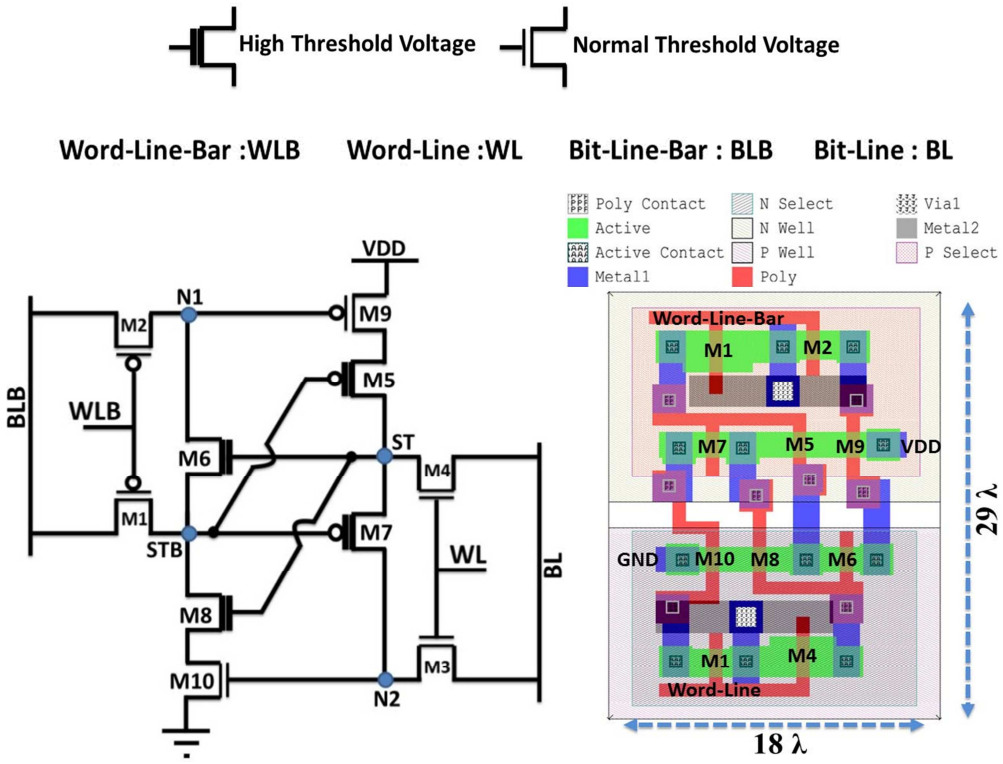
۲-۲- سلول سخت شده

شکل ۲-۱ شماتیک مداری و Layout یک سلول جدید سخت شده-صفر[۱] را در تکنولوژی ۲۲-nm با ولتاژ تغذیه ۰/۹۵V را نمایش می‌دهد. از این پس با نام 10T-H-NC[۲] به این سلول ارجاع خواهیم کرد. همچنین جدول ۲-۱ لیست اندازه ترانزیستورها و سطح ولتاژ آستانه را در این سلول نمایش می دهد.

جدول ۲-۱: اندازه ترانزیستورها و سطح ولتاژ آستانه در10T-H-NC

Transistor	Size (W/L)	Threshold Voltage
M1	84nm/22nm	Normal
M2	44nm/22nm	Normal
M3	44nm/22nm	Normal
M4	84nm/22nm	Normal
M5	44nm/22nm	High
M6	44nm/22nm	High
M7	44nm/22nm	High
M8	44nm/22nm	High
M9	44nm/22nm	Normal
M10	44nm/22nm	Normal

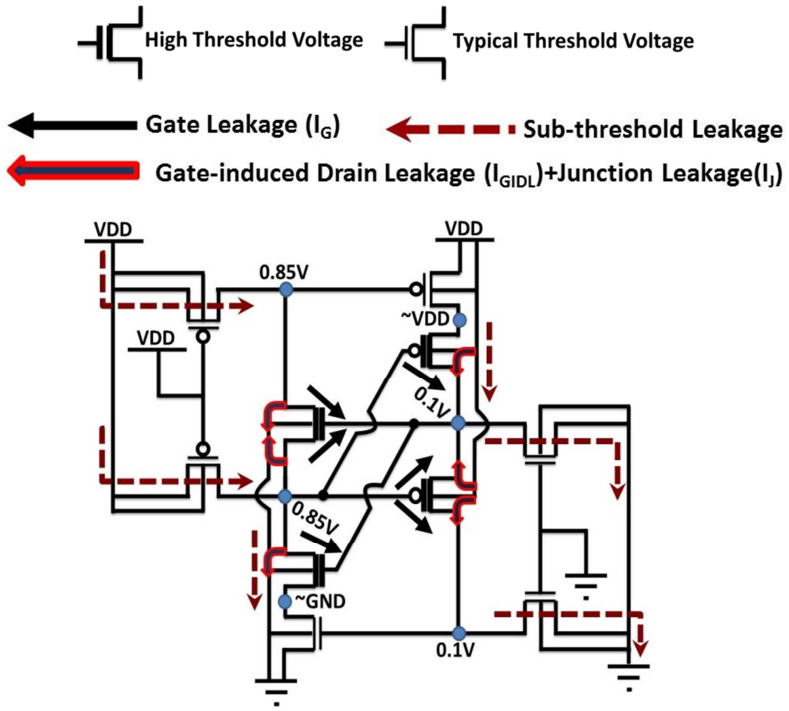
[۱] Zero-Hardened
[۲] 10T Hardened New SRAM



شکل ۲-۱: شماتیک مداری 10T Hardened New SRAM Cell (یا به صورت اختصار 10T-H-NC) و Layout آن در قوانین طراحی مقیاس‌پذیر

در طول سیکل بیکاری که داده یک در سلول ذخیره شده است، ترانزیستورهای M5، M6، M7، M8، M9 و M10 روشن هستند. بنابراین یک فیدبک مثبت بین گره هایST و STB وجود دارد. گره‌های ST و N2 توسط ترانزیستورهای M5، M7 و M9 به سطح ولتاژ V_{DD} کشیده می‌شود، و گره‌های STB و N1 توسط ترانزیستورهای M6، M8 و M10 به سطح ولتاژ GND کشیده می‌شود. اما در طول سیکل بیکاری زمانی که داده صفر در سلول ذخیره شده است، ولتاژ گره STB در سطح بالا است و ولتاژ گره ST در سطح ولتاژ پایین است، در این حالت تمامی ترانزیستورهای M5، M6، M7، M8، M9 و M10 خاموش هستند. شکل ۲-۲ موئلفه های جریان نشتی را در حالت بیکاری زمانی که داده یک در سلول جدید ذخیره شده است را نمایش می‌دهد. برای ذخیره سازی داده یک بدون انجام سیکل تازه سازی، در سمت چپ جریان نشتی ترانزیستورهای دستیابی (M1 و M2) که گره های STB وN1 را شارژ می کنند باید از جمع جریان های نشتی که گره های STB وN1 را دشارژ می‌کنند بزرگتر باشند. همچنین در سمت راست جریان نشتی

ترانزیستورهای دستیابی (M3 و M4) که گره های ST وN2 را دشارژ می کنند باید از جمع جریان های نشتی که گره های STB وN2 را شارژ می‌کنند بزرگتر باشند. در نتیجه نامساوی های زیر باید برقرار باشند تا داده یک بتواند بدون سیکل تازه سازی در سلول دخیره شود.



شکل ۲-۲: موئلفه های جریان نشتی در سیکل بیکاری زمانی که داده صفر در10T-H-NC ذخیره شده است

Left Side: *Node STB:* $I_{SD\text{-}M1}$ *(at* V_{SD}*=0.1V) ≫ {*$I_{Gate\text{-}M5}$*+*$I_{Gate\text{-}M6}$*+2*$I_{Gate\text{-}M7}$*+*$I_{Gate\text{-}M8}$ *} (at* $|V_{GS}|$*=0.75V) +*$I_{DS\text{-}M8}$ *(at* V_{DS}*=~0.85V) +*$I_{GIDL\text{-}M8}$ *(at* $|V_{DG}|$*=0.75V) +*$I_{J\text{-}M8}$ *(at* $|V_{DB}|$*=0.85V) +* $I_{GIDL\text{-}M6}$ *(at* $|V_{DG}|$*=0.75V) +*$I_{J\text{-}M6}$ *(at* $|V_{DB}|$*=0.85V)* (1)

Node N1: $I_{SD\text{-}M2}$ *(at* V_{SD}*=0.1V) ≫* $I_{Gate\text{-}M6}$ *(at* $|V_{GS}|$*=0.75V) +* $I_{GIDL\text{-}M6}$ *(at* $|V_{DG}|$*=0.75V) +*$I_{J\text{-}M6}$ *(at* $|V_{DB}|$*=0.85V)* (2)

Right Side: *Node ST:* $I_{DS\text{-}M4}$ *(at* V_{DS}*=0.1V) ≫ {*$I_{Gate\text{-}M5}$*+2*$I_{Gate\text{-}M6}$*+*$I_{Gate\text{-}M7}$*+*$I_{Gate\text{-}M8}$*} (at* $|V_{GS}|$*=0.75V) +*$I_{SD\text{-}M5}$ *(at* V_{SD}*=~0.85V) +* $I_{GIDL\text{-}M7}$ *(at* $|V_{DG}|$*=0.75V) +*$I_{J\text{-}M7}$ *(at* $|V_{DB}|$*=0.85V) +* $I_{GIDL\text{-}M5}$ *(at* $|V_{DG}|$*=0.75V) +*$I_{J\text{-}M5}$ *(at* $|V_{DB}|$*=0.85V)* (3)

Node N2: $I_{DS\text{-}M3}$ *(at* V_{DS}*=0.1V) ≫* $I_{Gate\text{-}M7}$ *(at* $|V_{GS}|$*=0.75V) +* $I_{GIDL\text{-}M7}$ *(at* $|V_{DG}|$*=0.75V) +*$I_{J\text{-}M7}$ *(at* $|V_{DB}|$*=0.85V)* (4)

برای برقرار کردن نامساوی های ۱ تا ۴ در زمانی که داده یک در سلول ذخیره شده است، از جریان نشتی زیرآستانه ترانزیستور های دستیابی شامل I_{SD-M1}، I_{SD-M2}، I_{SD-M3} و I_{SD-M4} استفاده می کنیم. بنابراین برای برقرار کردن نامساوی‌های ۱ تا ۴ Bit-Line در سطح GND نگهداشته می شود و Bit-Line-Bar در سطح V_{DD} نگهداری می شود، همانند آنچه که در شکل ۲-۲ نمایش داده شده است. از آنجایی که داده ها روی گره‌های ST و STB ذخیره می‌شوند، همانطور که در شکل ۲-۲ نمایش داده شده است، بیشتر جریان-های نشتی روی این گره‌ها وجود دارد. برای کاهش جریان نشتی که گره ST را دشارژ می‌کنند و جریان نشتی که گره STB را شارژ می کند، ما در ترانزیستورهای M5، M6، M7 و M8 از ولتاژ آستانه بالا استفاده کرده‌ایم. از آنجایی که جریان نشتی زیرآستانه با کاهش ولتاژ آستانه به صورت نمایی زیاد می شود، جریان نشتی زیرآستانه ترانزیستورهای M1 و M4 بسیار بزرگتر از جریان نشتی زیر آستانه ترانزیستورهای M5، M6، M7 و M8 است. نتایج شبیه سازی با HSPICE در تکنولوژی ۲۲ –nm BPTMs[۱] نشان می دهد که، با این انتساب ولتاژهای آستانه به ترانزیستورها داده یک بدون نیاز به سیکل تازه سازی در سلول ذخیره می شود.

فرآیند پیکربندی در 10T-H-NC به صورت زیر است.

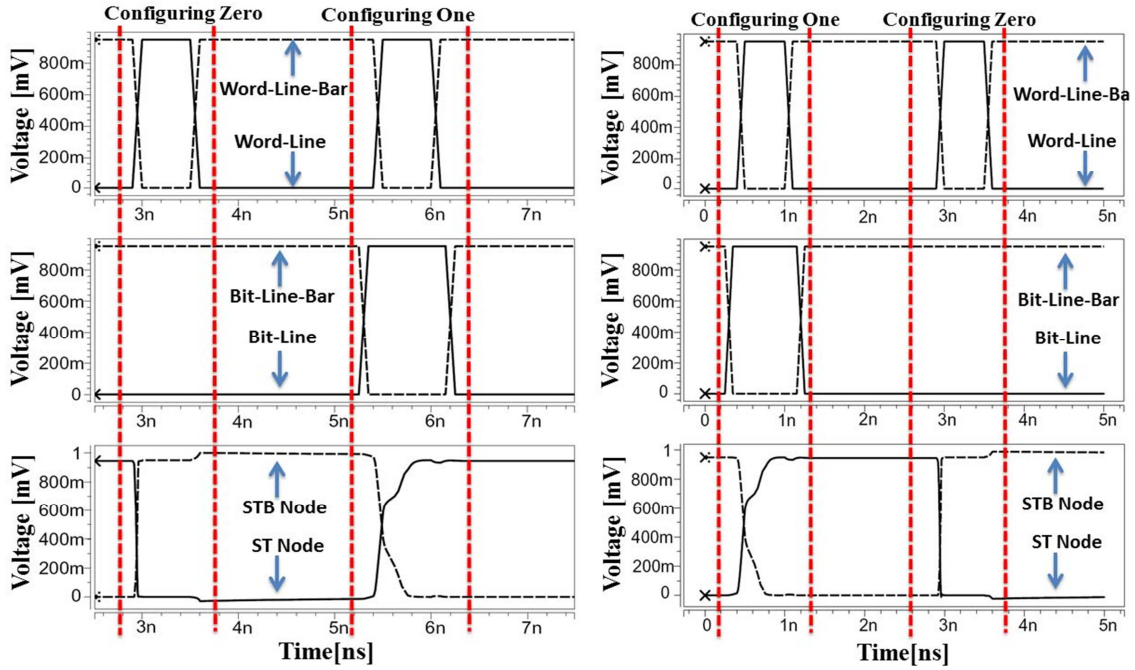
۱- قرار دادن داده روی خطوط Bit-Line: در این مرحله داده و مکمل آن روی Bit-Line و Bit-Line-Bar قرار می گیرد و سپس Word-Line1 و Word-Line2 به ترتیب به ولتاژ های GND و V_{DD} وصل می شود.

۲- در این مرحله دو حالت را می توان در نظر گرفت. الف)داده یک است: در این حالت گره‌های STB وN1 توسط ترانزیستور های دستیابی M1 و M2 به ولتاژ V_{DD} کشانده می‌شوند. و گره‌های ST و N2 توسط ترانزیستورهای M3 وM4 به GND کشیده می شوند. ب)- داده صفر است: در این حالت گره‌های STB وN1 توسط ترانزیستورهای دستیابی M1 و M2 به ولتاژ V_{TP} کشانده می‌شوند. و گره‌های ST و N2 توسط ترانزیستورهای M3 وM4 به V_{DD}-V_{TN} کشیده می‌شوند. بنابراین ترانزیستورهای M5، M6، M7، M8، M9 و M10 روشن می‌شوند و یک فیدبک مثبت بین گره های ST و STB ایجاد می شود. در اینجا V_{TP} ولتاژ آستانه ترانزیستور PMOS است و V_{TN} ولتاژ آستانه ترانزیستور NMOS است.

[۱] Berkeley Predictive Technology Models

۳- در اتمام سیکل پیکربندی سلول به حالت بیکاری می رود و Word-Line1 و Word-Line2 به ترتیب به ولتاژ V_{DD} و GND برمی‌گردند. و Bit-Line و Bit-Line-Bar بترتیب به GND و V_{DD} بر می‌گردند.

برای بررسی درستی عملکرد 10T-H-NC در طول سیکل‌های پیکربندی و بیکاری از دو سناریو که در ادامه آمده است استفاده شد است. ۱-پیکربندی سلول با داده صفر و سپس پیکربندی سلول با داده یک ۲- پیکربندی سلول با داده یک و سپس پیکربندی سلول با داده صفر. شکل ۲-۳ شکل موج شبیه‌سازی شده توسط HSPICE را برای این دو سناریو در تکنولوژی ۲۲–nm BPTM نمایش می‌دهد.



شکل ۲-۳: شکل موج شبیه سازی شده توسط HSPICE جهت بررسی درستی عملکرد 10T-H-NC در طول سیکل های بیکاری و پیکربندی

۲-۳- جریان نشتی در سلول سخت شده

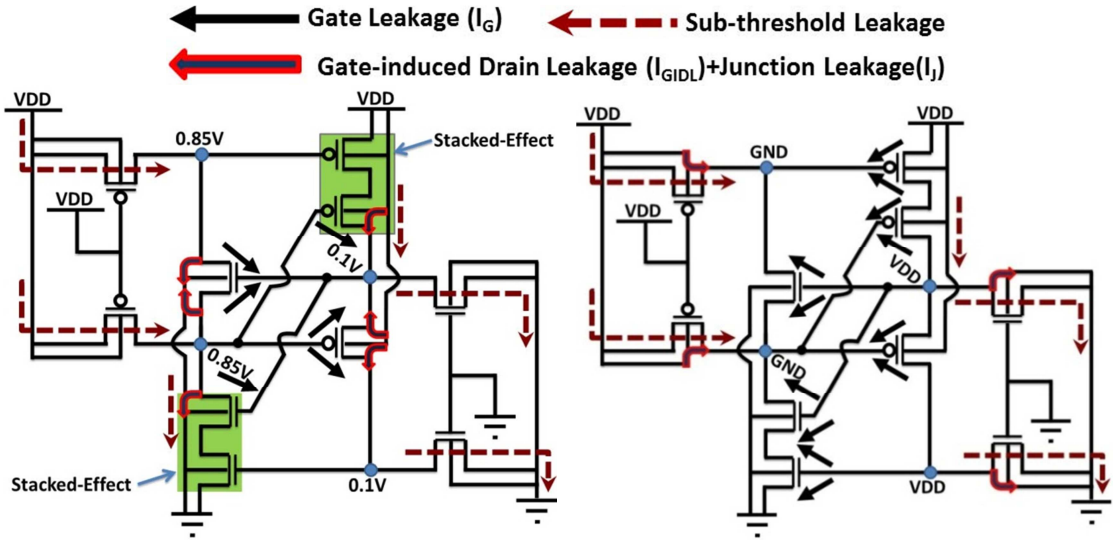
اندازه ترانزیستورها و ولتاژ تغذیه در هر نسل از در تکنولوژی CMOS به شدت در حال کاهش است. بنابراین برای نگهداری کارایی در ترانزیستورها ولتاژ آستانه ترانزیستورها در حال کاهش است[۱۷]. با کاهش ولتاژ آستانه ترانزیستورها جریان نشتی زیرآستانه آنها افزایش می‌یابد. بنابراین جریان نشتی زیر آستانه در در تکنولوژی‌های پیشرفته در حال افرایش است، زیرا که با پیشرفت تکنولوژی، ولتاژ آستانه ترانزیستورهای در رون یک تراشه در حال کاهش است[۱۷]. به طور کلی جریان نشتی یک تراشه به تعداد ترانزیستورهای

یک تراشه بستگی دارد. در درون یک FPGA پیشرفته امروزی حافظه پیکربندی بسیار زیادی برای منابع منطقی و مسیریابی وجود دارد، در کنار این حافظه‌های پیکربندی در FPGA های پیشرفته امروزی حافظه های جاسازی شده بسیار زیادی وجود دارد. بنابراین تعداد بسیار زیادی از ترانزیستورهای یک تراشه FPGA پیشرفته به حافظه‌ها تخصیص داده شده است، در نتیجه توان مصرفی ایستای بسیار زیادی توسط حافظه‌ها در درون یک FPGA مصرف می‌شود.

به دلیل کاهش جریان نشتی زیرآستانه با افزایش ولتاژ آستانه، یک روش ساده و بسیار معمول برای کاهش جریان نشتی زیرآستانه استفاده از ترانزیستورها با ولتاژ آستانه بالا است [۱۷، ۱۶]. اما این روش باعث افزایش تأخیر بسیار زیادی در مدار می‌شود و فقط می‌توان از آن در جاهایی از مدار استفاده کرد که تأخیر در آن نقاط بحرانی نیست [۱۶]. در مورد FPGA این نکته باید یادآوری شود که سلول‌های حافظه پیکربندی در مسیر انتشار سیگنال‌ها در درون منابع منطقی و مسیر یابی قرار ندارند[۱۶]. در نتیجه تأخیر زمانی پیکربندی سلول‌های حافظه پیکربندی بر روی سرعت FPGA تأثیر گذار نیستند[۱۶]. بنابراین می-توان در حافظه‌های پیکربندی FPGA ها از ترانزیستورها با ولتاژ آستانه بالا استفاده کرد. یکی دیگر از روش های مرسوم برای کاهش جریان نشتی زیر آستانه استفاده از پشته‌ای از چندین ترانزیستور خاموش سری به جای یک ترانزیستور خاموش است[۴]. جریان نشتی که از یک پشته از ترانزیستورهای خاموش سری عبور می‌کند کمتر است از جریان نشتی که از یک ترانزیستور خاموش عبور می‌کند [۴]. همچنین همانطور که در قسمت ۲-۱ گفته شد، در جریان بیت بیشتر کاربردهایی که در بروی FPGA پیاده‌سازی می شوند ۸۷ درصد داده‌ها صفر هستند [۱۴،۱۵]. دلیل اصلی این تعداد زیاد صفر در جریان بیت کاربردها زیاد بود تعداد بیت‌های بدون استفاده در حافظه پیکربندی منابع مسیریابی است [۱۴،۱۵]. بنابراین برای کاهش متوسط جریان نشتی حافظه پیکربندی باید جریان نشتی سلول در حالتی که داده صفر در سلول ذخیره شده است از حالتی که یک در سلول ذخیره می شود کمتر باشد. در نتیجه در سلول‌های جدیدی که برای کاهش جریان نشتی در این طرح تحقیقاتی طراحی شده‌اند، به صورتی هستند که جریان نشتی سلول در حالتی که داده صفر در سلول ذخیره شده است از حالتی که یک در سلول ذخیره می‌شود کمتر باشد. در این طرح تحقیقاتی

از این پس سلول‌هایی که برای کاهش جریان نشتی طراحی می‌شوند از آنها به عنوان سلول‌های IL[1] یاد خواهد شد.

شکل ۲-۴ جریان نشتی زیر آستانه را در حالتی که داده یک و داده صفر در 10T-H-NC ذخیره شده است را نمایش می‌دهد. همانطور که در این شکل نمایش داده شده است موئلفه‌های جریان نشتی زیر آستانه در حالت صفر توسط پشته‌هایی از ترانزیستورهای خاموش سری کاهش یافته است. همچنین اگر به شکل۲-۴ توجه شود تعداد موئلفه‌های جریان نشتی گیت در حالتی که صفر ذخیره شده است کمتر است از تعداد موئلفه‌های جریان نشتی گیت در حالتی که یک ذخیره شده است. بنابراین جریان نشتی سلول در حالتی که صفر ذخیره شده است کمتر است از جریان نشتی حالتی که یک در سلول ذخیره شده است، در نتیجه متوسط جریان نشتی در 10T-H-NC کاهش یافته است. با توجه به توضیحاتی که در بالا ذکر شد در این سلول برای کاهش بیشتر جریان نشتی زیرآستانه از ترانزیستورها بیشتری با ولتاژ آستانه بالا استفاده می کنیم. شکل ۲-۵ شماتیک مداری این سلول نمایش می‌دهد. ما این سلول را 10T-IL-H-NC[2] می نامیم.



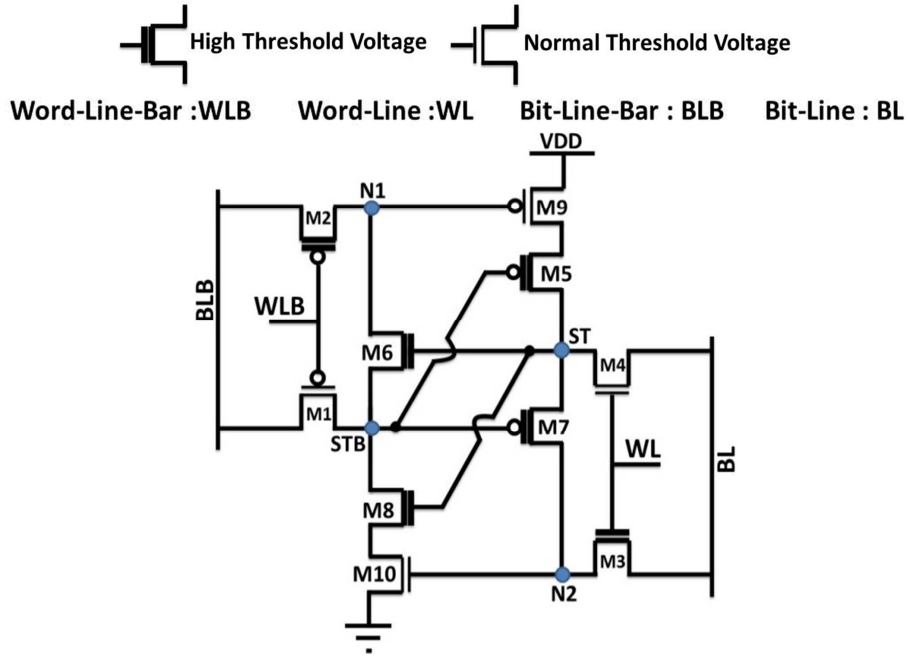
شکل ۲-۴: موئلفه‌های جریان نشتی زیر آستانه در 10T-H-BNC در حالت‌هایی که صفر و یک در سلول ذخیره شده

[1] Improved-Leakage
[2] 10T Improved-Leakage Hardened New SRAM Cell

۲-۴- برخورد ذرات انرژی دار با سلول های سخت شده- صفر

در این قسمت برخورد ذرات انرژی‌دار با سلول 10T-H-NC بررسی می‌شود. ما برای مدل کردن برخورد ذرات انرژی‌دار با گره‌های حساس یک سلول از یک منبع جریان مستقل با معادله (۲) استفاده می کنیم.

بطورکلی زمانی که یک ذره با 10T-H-NC برخورد می کند ۸ حالت متفاوت را می توان در نظر گرفت.



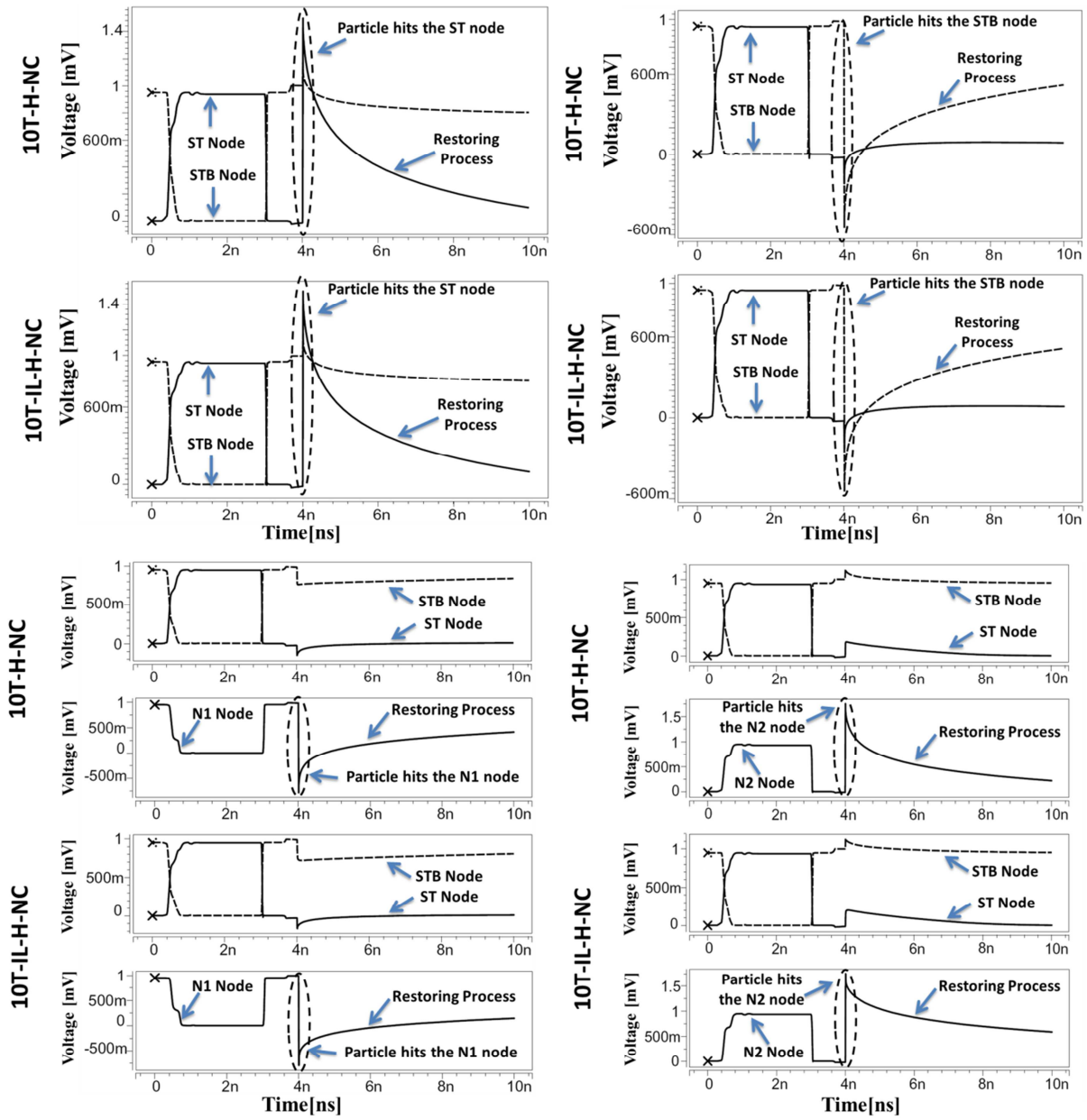
شکل ۲-۵: شماتیک مداری 10T Improved-Leakage Hardened New SRAM Cell (یا به صورت اختصار 10T-IL-H-NC)

۱- صفر در سلول ذخیره شده و یک ذره با گره ST برخورد می‌کند. در این حالت چون ترانزیستورهای M5 و M7 خاموش هستند، برخورد یک ذره انرژی دار با یک ترانزیستور خاموش PMOS یک پالس مثبت روی گره ST تولید می‌کند. اما این پالس مثبت هیچ تأثیری روی گره STB ندارد، زیرا ترانزیستورM7 خاموش است و هرگزنه پالس مثبتی روی گره ST نمی‌تواند به گره N2 انتشار پیدا کند. بنابراین ترانزیستور M10 خاموش است و هیچ مسیری وجود ندارد که گره STB را دشارژ کند. در نتیجه ولتاژ گره STB پیدار می‌ماند و بار تزریق شده توسط برخورد ذره به وسیله ترانزیستور M4 تخلیه می‌شود.

۲- صفر در سلول ذخیره شده و یک ذره با گره STB برخورد می‌کند. در این حالت چون ترانزیستورهای M6 و M8 خاموش هستند، برخورد یک ذره انرژی‌دار با یک ترانزیستور خاموش NMOS یک پالس منفی روی گره STB تولید می‌کند. اما این پالس منفی هیچ تأثیری روی گره ST ندارد، زیرا ترانزیستورM6 خاموش است و هرگرنه پالس منفی روی گره STB نمی‌تواند به گره N1 انتشار پیدا کند. بنابراین ترانزیستور M9 خاموش است و هیچ مسیری وجود ندارد که گره ST را شارژ کند. در نتیجه ولتاژ گره ST پایدار می‌ماند و بار کاهش یافته توسط برخورد ذره به وسیله ترانزیستور M1 جبران می‌شود.

۳- صفر در سلول ذخیره شده و یک ذره با گره N1 برخورد می کند. در این حالت چون ترانزیستور M6 خاموش است، برخورد یک ذره انرژی دار با یک ترانزیستور خاموش NMOS یک پالس منفی روی گره N1 تولید می‌کند. اما این پالس منفی هیچ تأثیری روی گره ST ندارد، زیرا ترانزیستور M6 خاموش است و هرگرنه پالس منفی روی گره N1 نمی‌تواند به گره STB انتشار پیدا کند. بنابراین ترانزیستور M5 خاموش است و هیچ مسیری وجود ندارد که گره ST را شارژ کند. در نتیجه ولتاژ گره ST پیدار می‌ماند و بار کاهش یافته توسط برخورد ذره به وسیله ترانزیستور M2 جبران می‌شود.

۴- صفر در سلول ذخیره شده و یک ذره با گره N2 برخورد می‌کند. در این حالت چون ترانزیستور M7 خاموش است، برخورد یک ذره انرژی دار با یک ترانزیستور خاموش PMOS یک پالس مثبت روی گره N2 تولید می کند. اما این پالس مثبت هیچ تأثیری روی گره STB ندارد، زیرا ترانزیستورM7 خاموش است و هرگرنه پالس مثبتی روی گره N2 نمی‌تواند به گره ST انتشار پیدا کند. بنابراین ترانزیستور M8 خاموش است و هیچ مسیری وجود ندارد که گره STB را دشارژ کند. در نتیجه ولتاژ گره STB پایدار می‌ماند و بار تزریق شده توسط برخورد ذره به وسیله ترانزیستور M3 تخلیه می‌شود. شکل ۲-۶ شکل‌موج شبیه‌سازی شده توسط HSPICE را برای حالت‌های ۱ تا ۴ را در 10T-ZH-NC نمایش می‌دهد.



شکل ۲-۶: شکل موج شبیه سازی شده توسط HSPICE را برای حالت های ۱ تا ۴ در 12T-ZH-NC

۵-یک در سلول ذخیره شده و یک ذره با گره ST برخورد می‌کند. در این حالت چون ترانزیستور M4 خاموش است، برخورد یک ذره انرژی دار با یک ترانزیستور خاموش NMOS یک پالس منفی روی گره ST تولید می‌کند. این پالس منفی به گره N2 انتشار پیدا می‌کند زیرا که ترانزیستور M7 در این حالت روش است. بنابراین این پالس منفی ترانزیستورهای M6 وM8 را خاموش می‌کند. این باعث می شود که گره‌های

N1 و STB به حالت امپدانس بالا می‌روند. در نتیجه هیچ تغییری روی ولتاژ این گره‌ها رخ نمی‌دهد و بار کاهش یافته روی گره‌های ST وN2 توسط ترانزیستورهای M5 و M7 ترمیم می‌شود.

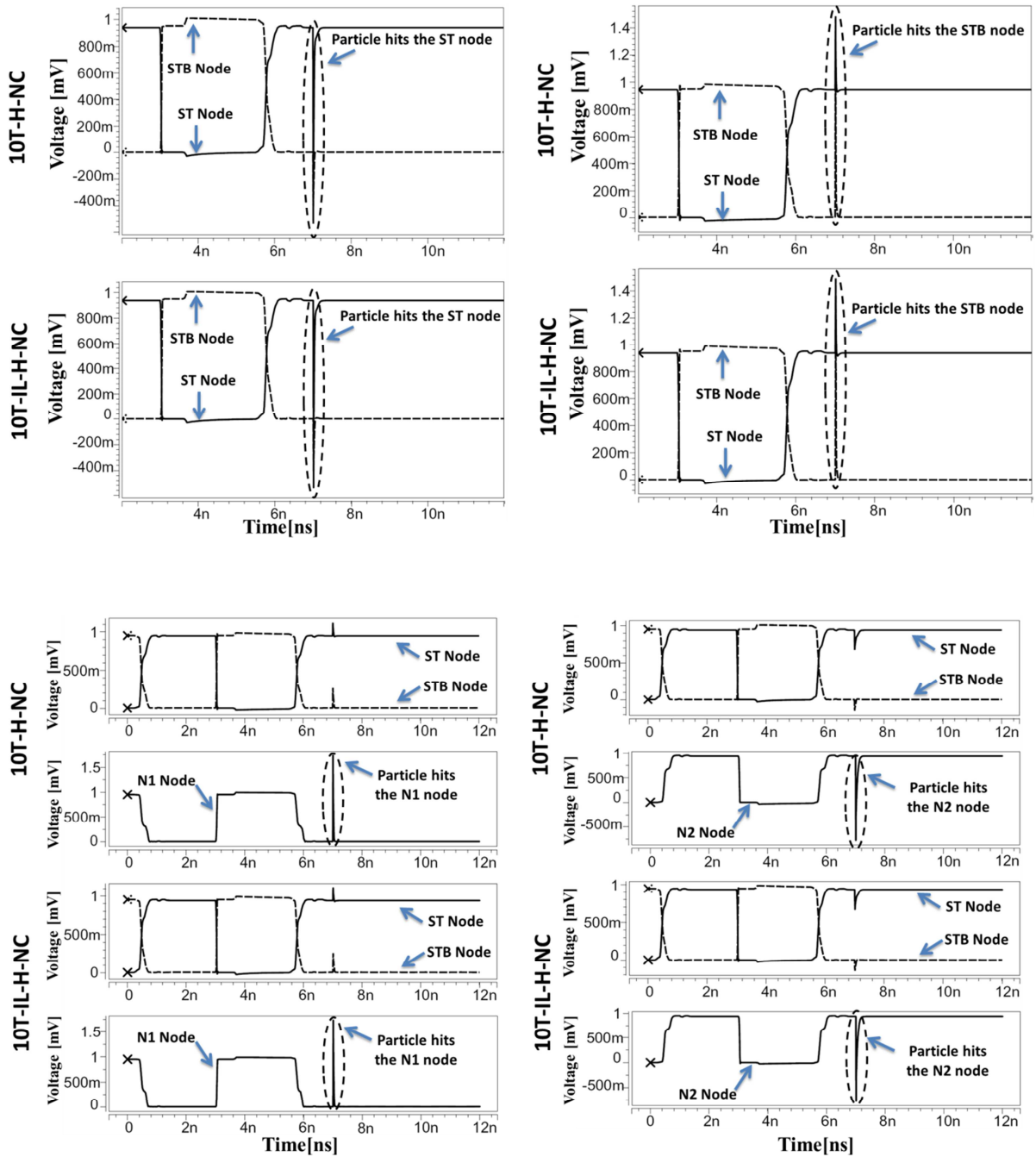
۶-یک در سلول ذخیره شده و یک ذره با گره STB برخورد می‌کند. در این حالت چون ترانزیستور M1 خاموش است، برخورد یک ذره انرژی‌دار با یک ترانزیستور خاموش PMOS یک پالس مثبت روی گره STB تولید می‌کند. این پالس مثبت ترانزیستورهای M5 و M7 را خاموش می‌کند و بنابراین گره‌های ST و N2 در حالت امپدانس بالا قرار می‌گیرند، در این نتیجه هیچگونه تغییری روی این گره‌های رخ نمی‌دهد و بار تزریق شده توسط برخورد ذره به وسیله ترانزیستورهای M8 و M10 تخلیه می‌شود.

۷-یک در سلول ذخیره شده و یک ذره با گره N1 برخورد می‌کند. در این حالت چون ترانزیستور M2 خاموش است، برخورد یک ذره انرژی‌دار با یک ترانزیستور خاموش PMOS یک پالس مثبت روی گره N1 تولید می‌کند. این پالس مثبت به گره STB انتشار پیدا می‌کند زیرا ترانزیستور M6 روشن است. در نتیجه این پالس مثبت ترانزیستورهای M5، M7 و M9 را خاموش می‌کند و بنابراین گره‌های ST و N2 در حالت امپدانس بالا قرار می‌گیرند، در این نتیجه هیچگونه تغییری روی این گره‌های رخ نمی‌دهد و بار تزریق شده توسط برخورد ذره به وسیله ترانزیستورهای M6، M8 و M10 تخلیه می‌شود.

۸-یک در سلول ذخیره شده و یک ذره با گره N2 برخورد می‌کند. در این حالت چون ترانزیستور M3 خاموش است، برخورد یک ذره انرژی‌دار با یک ترانزیستور خاموش NMOS یک پالس منفی روی گره N2 تولید می‌کند. این پالس منفی به گره ST انتشار پیدا می‌کند زیرا که ترانزیستور M7 در این حالت روشن است. بنابراین این پالس منفی ترانزیستورهای M6، M8 و M10 را خاموش می‌کند. این باعث می شود که گره‌های N1 و STB به حالت امپدانس بالا می‌روند. در نتیجه هیچ تغییری روی ولتاژ این گره‌ها رخ نمی‌دهد

و بار کاهش یافته روی گره‌های ST وN2 توسط ترانزیستورهای M5 و M7 ترمیم می‌شود. شکل ۲-۷ شکل-موج شبیه‌سازی شده توسط HSPICE را برای حالت‌های ۵ تا ۸ را در 10T-H-NC نمایش می‌دهد.

در نتیجه در زمانی که داده صفر یا یک در سلول ذخیره شده است، برخورد ذرات با گره‌های حساس 10T-H-NC نمی‌تواند داده سلول را عوض کند.



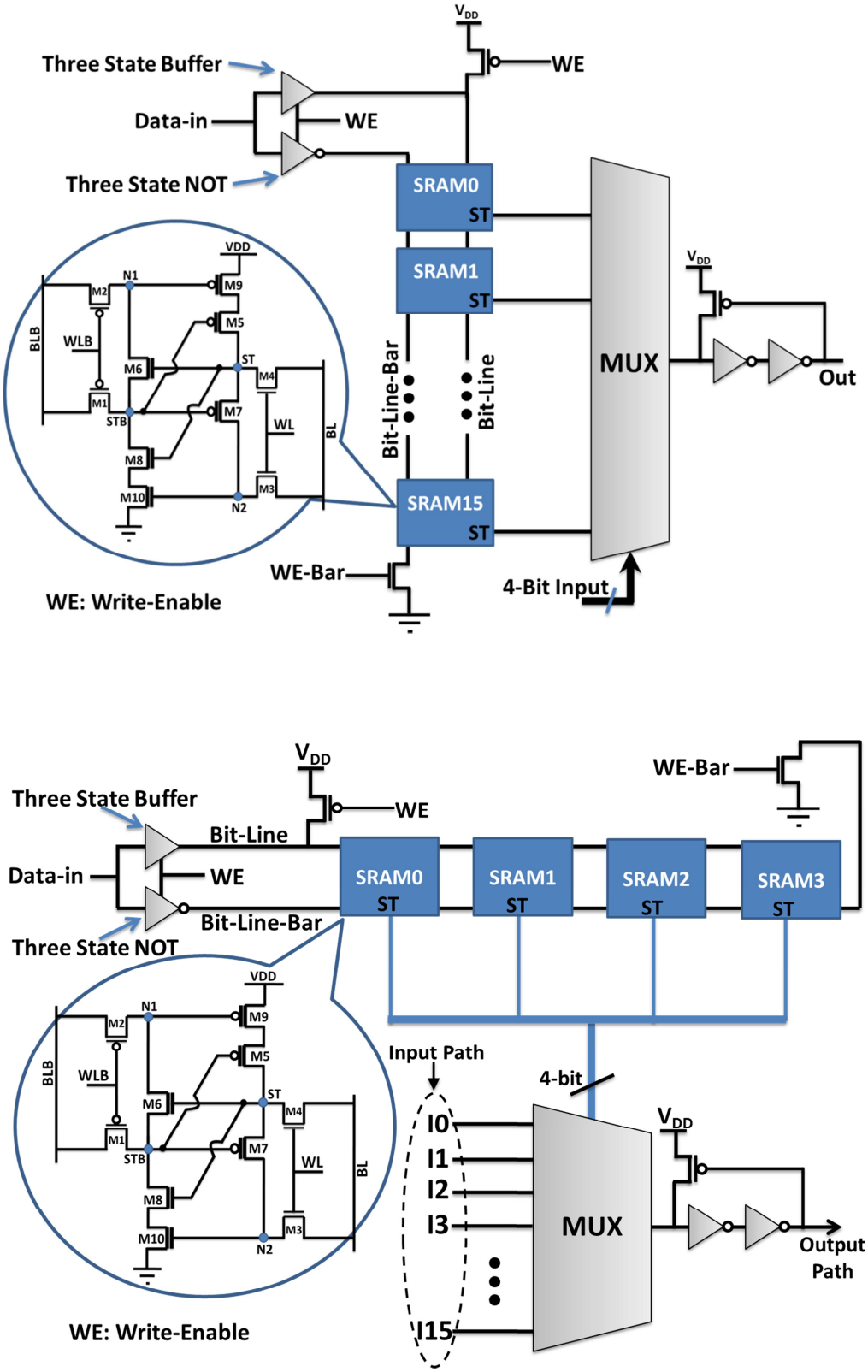
شکل ۲-۷: شکل موج شبیه سازی شده توسط HSPICE را برای حالت های ۵ تا ۸ در 12T-ZH-NC

۲-۵- معماری LUT و سوئیچ مسیریابی بر مبنای 10T-ZH-BNC

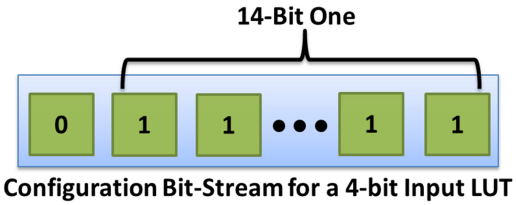
همانطور که گفته شد LUT و سوئیچ مسیریابی موئلفه‌های اصلی یک سخت‌افزار قابل پیکربندی است. این موئلفه‌ها شامل حافظه‌های پیکربندی هستند، بنابراین اگر در این قطعات یک سخت افزار قابل پیکربندی از سلول‌های سخت شده استفاده شود خطاهای نرم روی سیستم‌های چند عامله که بر روی سخت‌افزارهای قابل پیکربندی پیاده‌سازی می‌شوند تأثیری ندارد.

شکل ۲-۸ معماری یک LUT و سوئیچ مسیریابی را بر مبنای 10T-H-NC نمایش می‌دهد. همانطور که از شکل ۲-۸ مشخص است در LUT و سوئیچ مسیریابی خطوط Bit-Line و Bit-Line-Bar بین تمامی سلول‌ها مشترک است. در زمان انجام عمل پیکربندی، Write-Enable فعال است و داده و مکمل آن روی خطوط Bit-Line و Bit-Line-Bar قرار می‌گیرد. در طول سیکل بیکاری Write-Enable غیر فعال است و خطوط Bit-Line و Bit-Line-Bar به ترتیب روی GND و V_{DD} نگهداری می‌شوند.

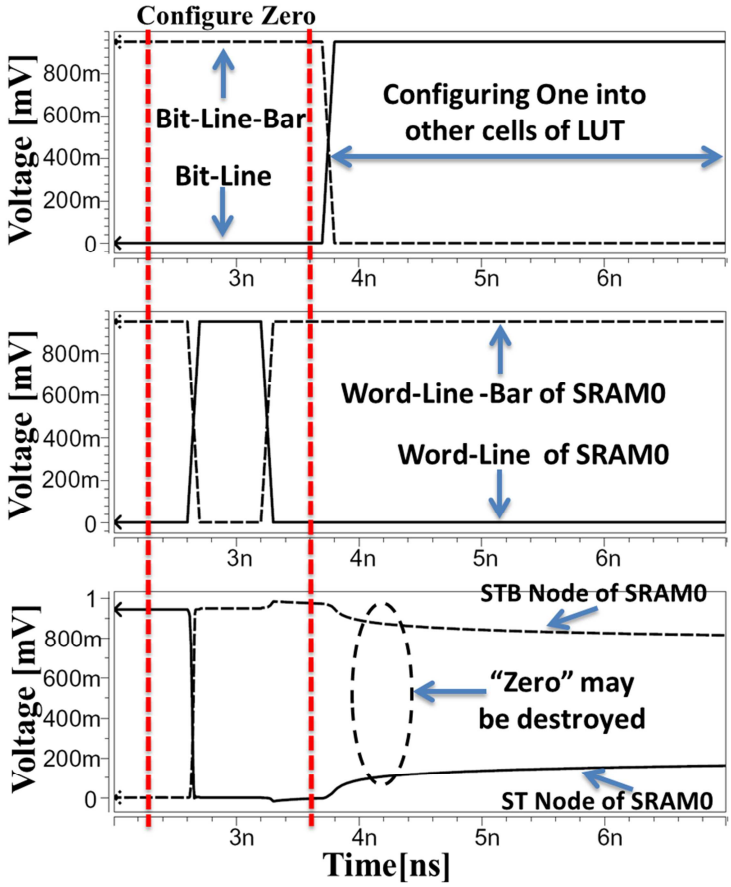
همانطور که در قسمت ۲-۲ توضیح ذکر شد، 10T-H-NC برا ذخیره سازی داده یک از جریان نشتی ترانزیستورهای دستیابی استفاده می‌کند، برای این منظور در طول سیکل بیکاری خطوط Bit-Line و Bit-Line-Bar به ترتیب روی GND و V_{DD} نگهداری می‌شوند. اکنون حالتی را در نظر بگیرید که جریان بیت شکل ۲-۹-الف باید در LUT شکل ۲-۸ پیکربندی شود. بیت اول (صفر) در SRAM0 از LUT پیکربندی می‌شود و بیت‌های بعدی (یک‌ها) باید در سلول های ۱ تا ۱۵ پیکربندی شوند. از آنجایی داده‌های سلول‌های ۱ تا ۱۵ یک هستند خطوط Bit-Line و Bit-Line-Bar برای مدت زمان زیادی به ترتیب در سطح V_{DD} و GND نگهداری می‌شوند. درنتیجه ممکن است که داده صفر که در SRAM0 ذخیره شده است خراب شود. شکل ۲-۹-ب شکل‌موج شبیه‌سازی شده توسط HSPICE را برای حالت بالا نمایش می‌دهد. این وضعیت ممکن است در سوئچ مسیریابی نیز رخ دهد.



شکل ۲-۸: معماری LUT و سوئیچ مسیریابی بر مبنای 10T-H-NC



(الف)



(ب)

(پ)

شکل ۲-۹: (الف) یک جریان بیت برای پیکربندی (ب) شکل موج شبیه سازی شده برای SRAM0 در یک LUT با ۴ ورودی

(پ) اضافه کردن یک سیکل بیکاری بعد از هر دو یک متوالی در یک جریان بیت

برای اجتناب از وضعیت بالا در یک LUT و یک سوئیچ مسیریابی بعد از هر دو یک متوالی در جریان بیت یک سیکل بیکاری در نظر می‌گیریم. این روش در شکل ۲-۹-پ نمایش داده شده است. به این ترتیب با این سیکل بیکاری که بعد از هر دو یک قرار می گیرد، یک زمان به سلول‌هایی که دارای داده صفر هستند داده می‌شود که توسط جریان‌های نشتی ترانزیستورهای دستیابی خود داده صفر خود را ترمیم کنند. اما اضافه کردن این سیکل‌های بیکاری بعد از هر دو صفر متوالی باعث افزایش تأخیر پیکربندی می‌شود. ولی افزایش تأخیر زمان پیکربندی تأثیری روی کارایی FPGA ندارد[۱۳،۱۶].

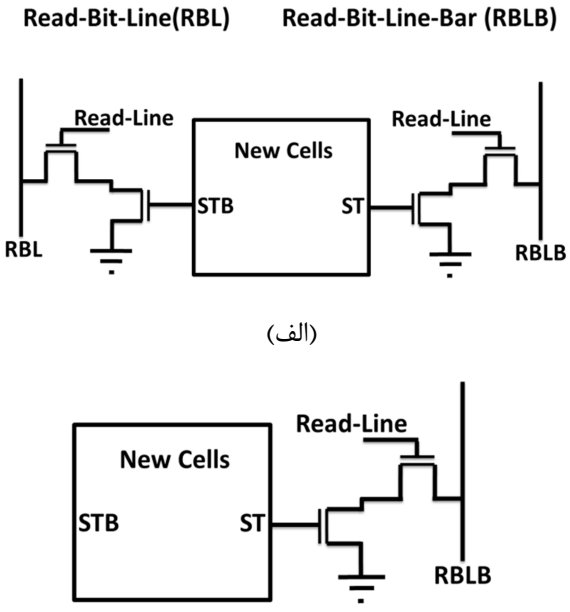
۲-۶- استفاده از سلول‌های جدید برای حافظه‌های جاسازی شده در سخت افزارهای قابل پیکربندی

همانطور که درقسمت‌های ۱-۴ گفته شد در بسیاری از FPGAهای پیشرفته حافظه‌ها و واحدهای محاسباتی به صورت جاسازی شده وجود دارند[۱]. این قطعات جاسازی شده از طریق شبکه قابل پیکربندی به سایر قسمت‌های FPGA وصل می‌شوند و می‌توانند داده‌ها را از طریق شبکه قابل پیکر بندی دریافت کنند و بعد از انجام وظیفه درخواست شده داده‌های پردازش شده را به شبکه قابل پیکر بندی ارسال کنند[۱]. یکی از این قطعات جاسازی شده حافظه‌های نهان هستند که برای ذخیره‌سازی داده‌های در حال پردازش در FPGA استفاده می‌شوند. به طور کلی تمامی سلول‌هایی جدید با پورت و بدون پورت که در این فصل طراحی شده‌اند می توان از آنها در حافظه‌های جاسازی شده استفاده کرد. در قسمت‌های بعدی نحوه استفاده سلول‌های جدید در حافظه های نهان جاسازی شده توضیح داده خواهد شد.

سلول‌های 10T-H-NC و 10T-IL-H-NC همگی دارای یک پورت برای نوشتن داده هستند و در سلول‌های بدون پورت برای نوشتن داده از خطوط تغذیه استفاده می‌شود. بنابراین همگی سلول‌های جدید دارای یک پورت برای نوشتن داده هستند. در نتیجه با اضافه کردن یک پورت خواندن به این سلول‌ها می-توان آنها را در یک حافظه جاسازی شده استفاده کرد. به دو صورت می توان یک پورت خواندن به یک سلول اضافه کرد که عبارتند از: ۱-پورت خواندن تک خطی[۱] ۲-پورت خواندن تفاضلی[۱]. سلول‌هایی که از پورت‌های

[۱] Single-Ended Read Port

خواندن تک خطی استفاده می‌کنند دارای مساحت کمی هستند. همچنین سلول هایی که از پورت خواندن تک خطی استفاده می‌کنند دارای جریان نشتی کمتری هستند، زیرا همیشه یک حالت در پورت وجود دارد (در حالت صفر یا یک) که در آن حالت بدلیل وجود پشته‌ای از ترانزیستورهای خاموش جریان نشتی کمی مصرف می شود. اما پورت‌های خواندن تک خطی دارای سرعت کمتری در طول سیکل خواندن هستند. سلول‌هایی که از پورت‌های خواندن تفاضلی استفاده می کنند دارای مساحت و جریان نشتی بیشتری هستند، ولی به دلیل انجام تغییرات ولتاژ تفاضلی در سیکل خواندن دارای سرعت بیشتری در طول سیکل خواندن هستند. بطورکلی در طراحی حافظه‌های جاسازی شده سرعت نوشتن داده‌ها در سلول مهم نیست، زیرا بیشتر تأخیر نوشتن ناشی از ارسال اطلاعات روی خطوط Bit-Line است، زیرا این خطوط که دارای بار خازنی زیادی هستند[۱۷]. از طرف دیگر از آنجایی که Write-Driverها قسمت بسیار کمی از کل یک حافظه جاسازی شده هستند، می توان با طراحی Write-Driverهای قوی و بزرگ تأخیر را روی خطوط Bit-Line به شدت کاهش داد[۱۷]. بنابراین در طراحی یک حافظه جاسازی شده تأخیر خواندن بسیار مهمتر از تأخیر نوشتن است.



شکل ۲-۱۰: (الف) پورت خواندن تفاضلی (ب) پورت خواندن تک خطی

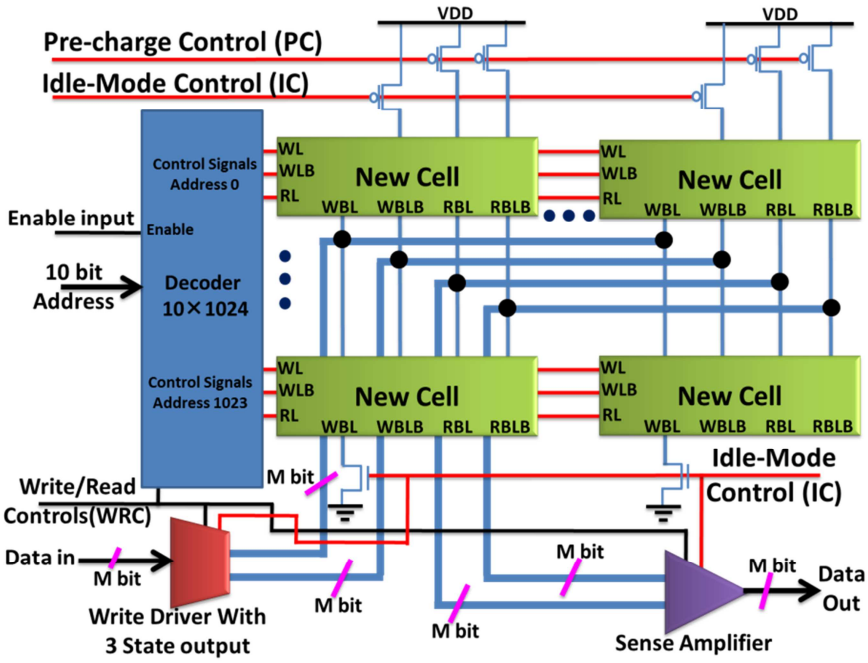
[۱] Differential Read Port

از آنجایی که سلول‌های 10T-H-NC و 10T-IL-H-NC همگی دارای یک پورت برای نوشتن داده هستند، در نتیجه با اضافه کردن یک پورت خواندن به این سلول‌ها می‌توان آنها را در یک حافظه جاسازی شده استفاده کرد. به دو طریق می‌توان به سلول‌های جدید یک پورت اضافه کرد. شکل۲-۱۰ نحوه اضافه کردن پورت خواندن را به سلول‌های ذکر شده را نمایش می‌دهد.

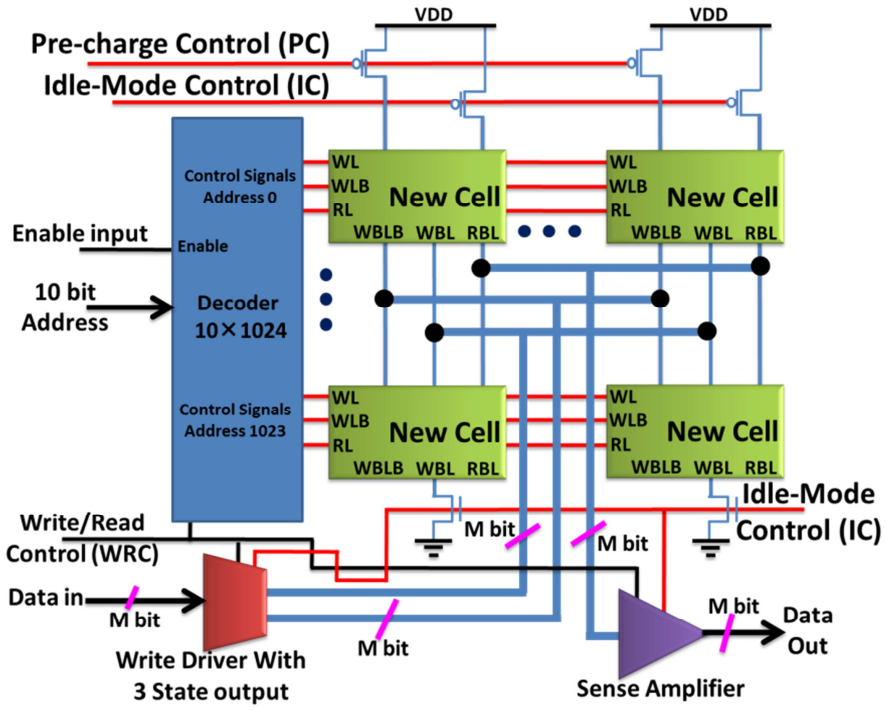
شکل ۲-۱۱ معماری یک بلاک حافظه جاسازی شده را با اندازه 1K×M-bit بر مبنای سلول 10T-H-NC با پورت خواندن تک خطی و پورت خواندن تفاضلی نمایش می‌دهد. در هر سیکل نوشتن فعال‌ساز دیکودر سطری فعال می‌شود و آدرس سطر کلمه مورد نظر رمزگشایی می‌شود، درنتیجه Word-Line کلمه مورد نظر فعال می‌شود و داده M بیتی در کلمه انتخاب شده نوشته می‌شود. داده‌ها توسط Write-Driver را روی خطوط Write-Bit-Line و Write-Bit-Line-Bar قرار می‌گیرند. در انتهای عملیات نوشتن خطوط Write-Bit-Line و Write-Bit-Line-Bar به ترتیب به ولتاژ GND و V_{DD} کشیده می‌شوند. در هر سیکل خواندن خطوط Bit-Line به V_{DD} پیش‌شارژ می‌شوند، سپس فعارساز رمزگشای سطری فعال و آدرس سطر مورد نظر رمزگشایی می‌گردد و با فعال شدن خط Read-Line برای کلمه انتخاب شده داده کلمه، روی خطوط Bit-Line قرار می‌گیرد و داده مورد نظر توسط تقویت کننده حسی[۱] خوانده می‌شود.

همانطور که در قسمت ۲-۲ توضیح داده شد، 10T-H-NC برا ذخیره سازی داده صفر از جریان نشتی ترانزیستورهای دستیابی استفاده می‌کند، برای این منظور در طول سیکل بیکاری خطوط Write-Bit-Line و Write-Bit-Line-Bar به ترتیب روی V_{DD} و GND نگهداری می‌شوند. اکنون حالتی را در نظر بگیرید که بلاک داده شکل ۲-۱۲-الف با طول h کلمه باید در یکی از بلاک‌های حافظه شکل ۲-۱۱ نوشته شود. زمانی که کلمه اول در آدرس X نوشته شد، کلمات بعدی بلاک داده باید در آدرس‌های X+1، X+2،...،X+h-1 نوشته شوند. و از آنجایی که داده این کلمات یک است تمای خطوط Write-Bit-Line و Write-Bit-Line-Bar در تمامی سطرها برای مدت طولانی در سطح V_{DD} و GND نگهداشته می‌شود. به این ترتیب امکان دارد که داده‌های صفر که در آدرس X و سایر آدرس‌ها ذخیره شده‌اند خراب شوند.

[۱] Sense Amplifier



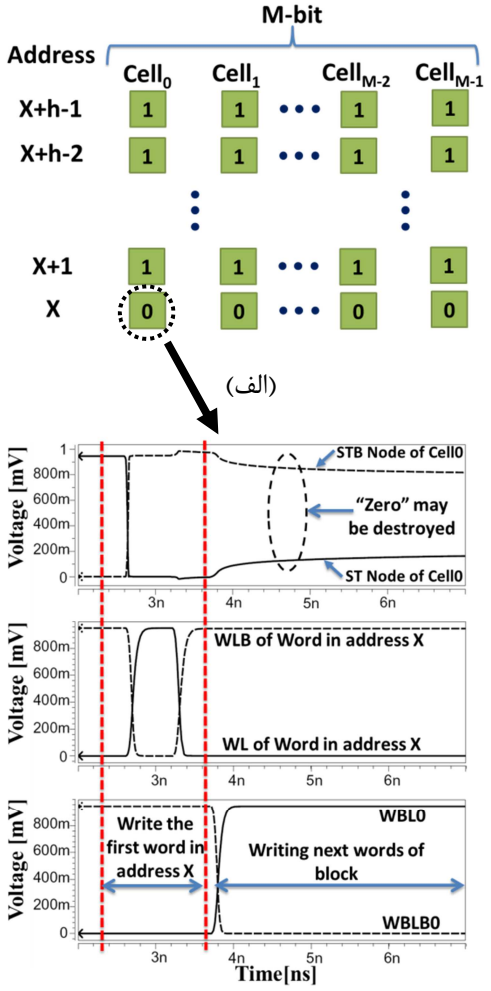
(الف)



(ب)

شکل ۲-۱۱: (الف) معماری یک بلاک حافظه جاسازی شده بر مبنای سلول 10T-H-NC با پورت خواندن تفاضلی (ب) معماری یک بلاک حافظه جاسازی شده بر مبنای سلول 10T-H-NC با پورت خواندن تک خطی

شکل ۲-۱۲-ب شکل‌موج شبیه‌سازی شده برای رخ داده حالت بالا را نمایش می‌دهد. همانطور که در این شکل‌موج نمایش داده شده است، پس از مدت کوتاهی که Write-Bit-Line و Write-Bit-Line-Bar به ترتیب روی V_{DD} و GND نگهداشته شده‌اند، صفرهایی که در آدرس X ذخیره شده‌اند ممکن است در زمان نوشتن داده یک در سایر آدرس‌ها خراب شوند.

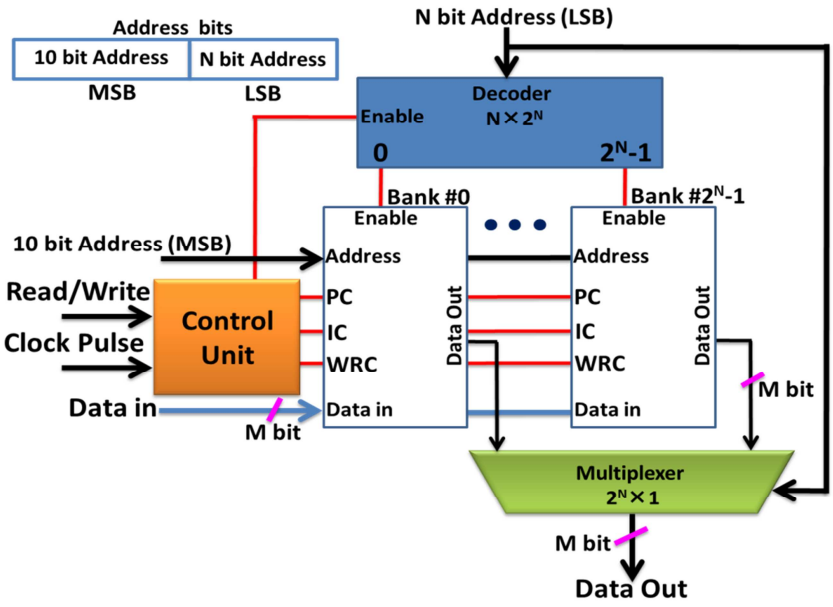


(ب)

شکل ۲-۱۲: (الف) یک بلاک داده با طول h کلمه (ب) شکل‌موج شبیه‌سازی شده برای $Cell_0$ در آدرس X زمانی که بلاک بالا در یک بانک حافظه شکل ۲-۱۱ نوشته می‌شود

برای اجتناب از رخ داده حالت بالا در بلاک حافظه‌های که بر مبنای 10T-H-NC طراحی شده است، ما از تکنیک Interleaving استفاده می‌کنم تا آدرس‌های متوالی را در بلاک‌های حافظه متفاوت قرار دهیم

[۱۸]. بر مبنای این تکنیک شکل ۲-۱۳ معماری کلی یک حافظه جاسازی شده با اندازه $2^{N}K\times Mbit$ را برمبنای بلاک‌های شکل ۲-۱۱ نمایش می‌دهد.



شکل ۲-۱۳: معماری حافظه جاسازی شده بر مبنای سلول 10T-H-NC

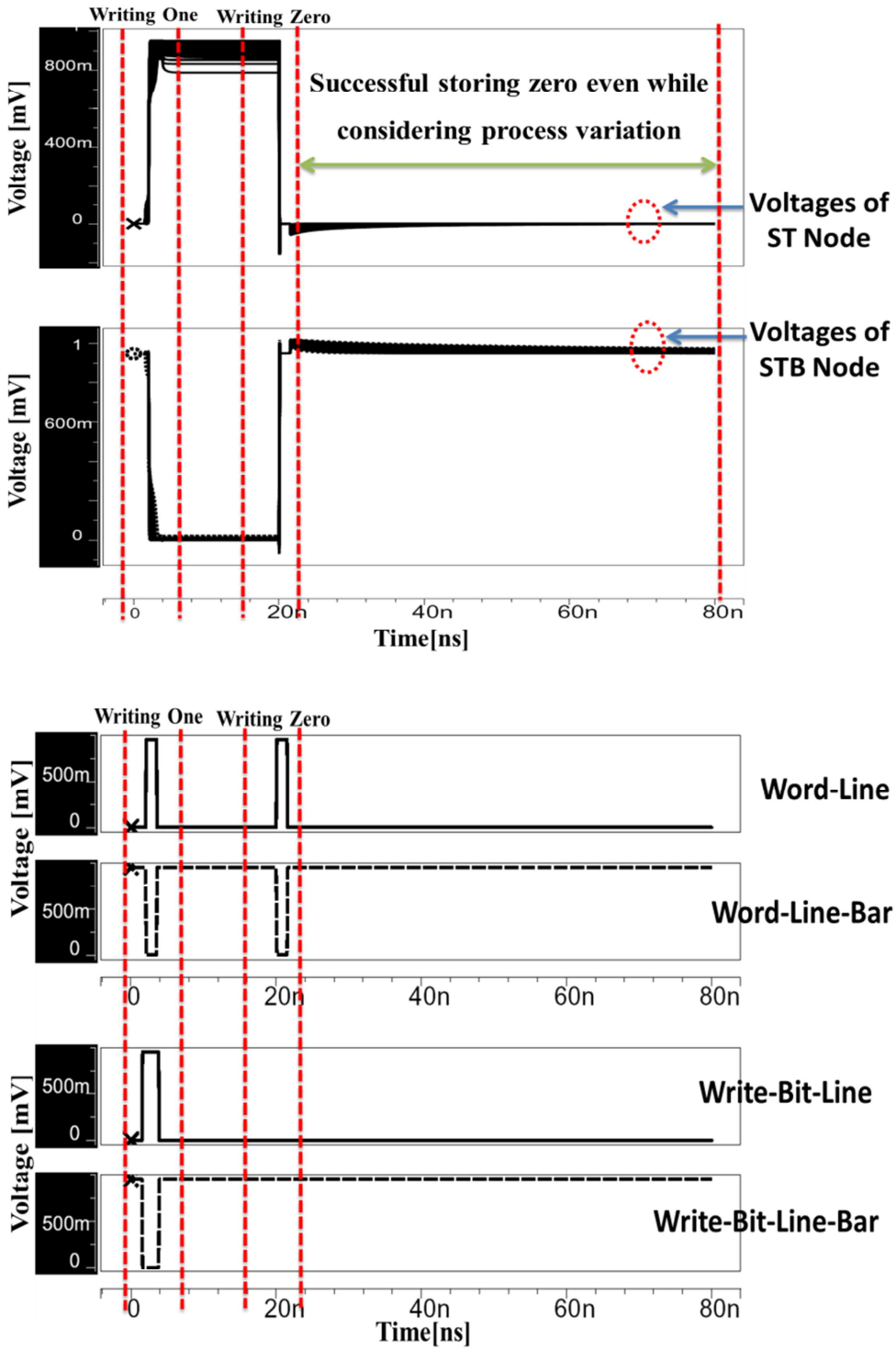
در این معماری واحد کنترل سیگنال‌های کنترلی را برای بانک‌های حافظه تولید می‌کند. بنابراین زمانی که یک بلاک باید در حافظه نوشته شود، اولین کلمه از بلاک داده در یک بانک حافظه نوشته می‌شود و کلمه بعدی در بلاک بعدی نوشته می‌شود. بنابراین بعد از هر نوشتن کلمه یک سیکل بیکاری به بانک حافظه‌ای که عمل نوشتن روی آن انجام شده است داده می‌شود تا داده‌های صفر خود را ترمیم کند. همچنین به دلیل استفاده از آدرس‌دهی Interleaving در این معماری تمامی مزیت‌هایی این روش که در مرجع [۱۸] توضیح داده شده است، در این معماری وجود دارد.

۲-۷- اثر تغییرات فرآیند ساخت روی عملکرد سلول‌های جدید

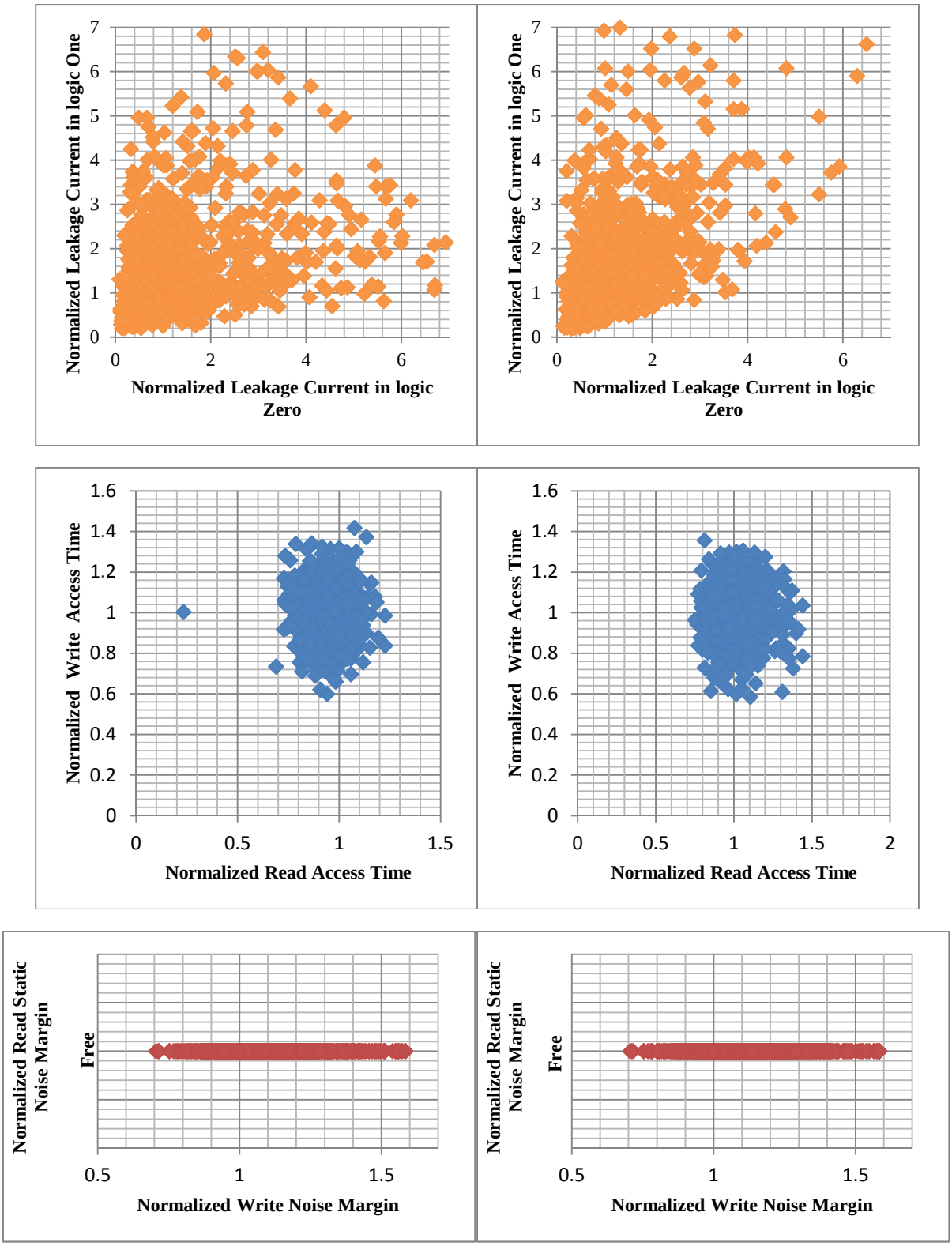
همانطور که در قسمت‌های قبل گفته شد برای اینکه 10T-H-NC و 10T-IL-H-NC بتوانند داده خود را ذخیره کنند، نامساوی‌هایی ۱ تا ۴ در سمت راست و چپ باید برقرار باشد. اکنون این سوأل مطرح می‌شود

که در اثر تغییرات فرآیند ساخت ممکن است که این نامساوی‌ها نتوانند برقرار باشند و سلول نتواند داده خود را ذخیره کند. برای پاسخ به این سوأل، ما تغییرات فرآیند ساخت را با استفاده از شبیه‌سازی Monte Carlo روی این دو سلول اعمال کرده‌ایم. شکل ۲-۱۴ ولتاژهای گره‌های ST و STB سلول 10T-H-NC را تحت تغییرات فرآیند ساخت نمایش می‌دهد. شکل موج مشابحی نیز برای سلول 10T-IL-H-NC بدست آمد است. همانطور که از این شکل مشخص است با وجود تغییرات فرآیند ساخت این دو سلول می‌توانند به درستی کار کنند. برای شبیه‌سازی تغییرات فرآیند ساخت از شبیه‌سازی Monte Carlo با تعداد دفعات اجرا ۱۰۰۰ استفاده کرده‌ایم. پارامترهای L، W، V_T و T_{OX} دارای توزیع احتمال نرمال هستند و هر کدام به صورت مستقل تغییر می کنند. همچنین هر پارامتر در این بررسی دارای واریانس 3δ به میزان ۳۰ درصد است. شبیه‌سازی‌های این قسمت در تکنولوژی BPTM 22–nm انجام شده‌اند.

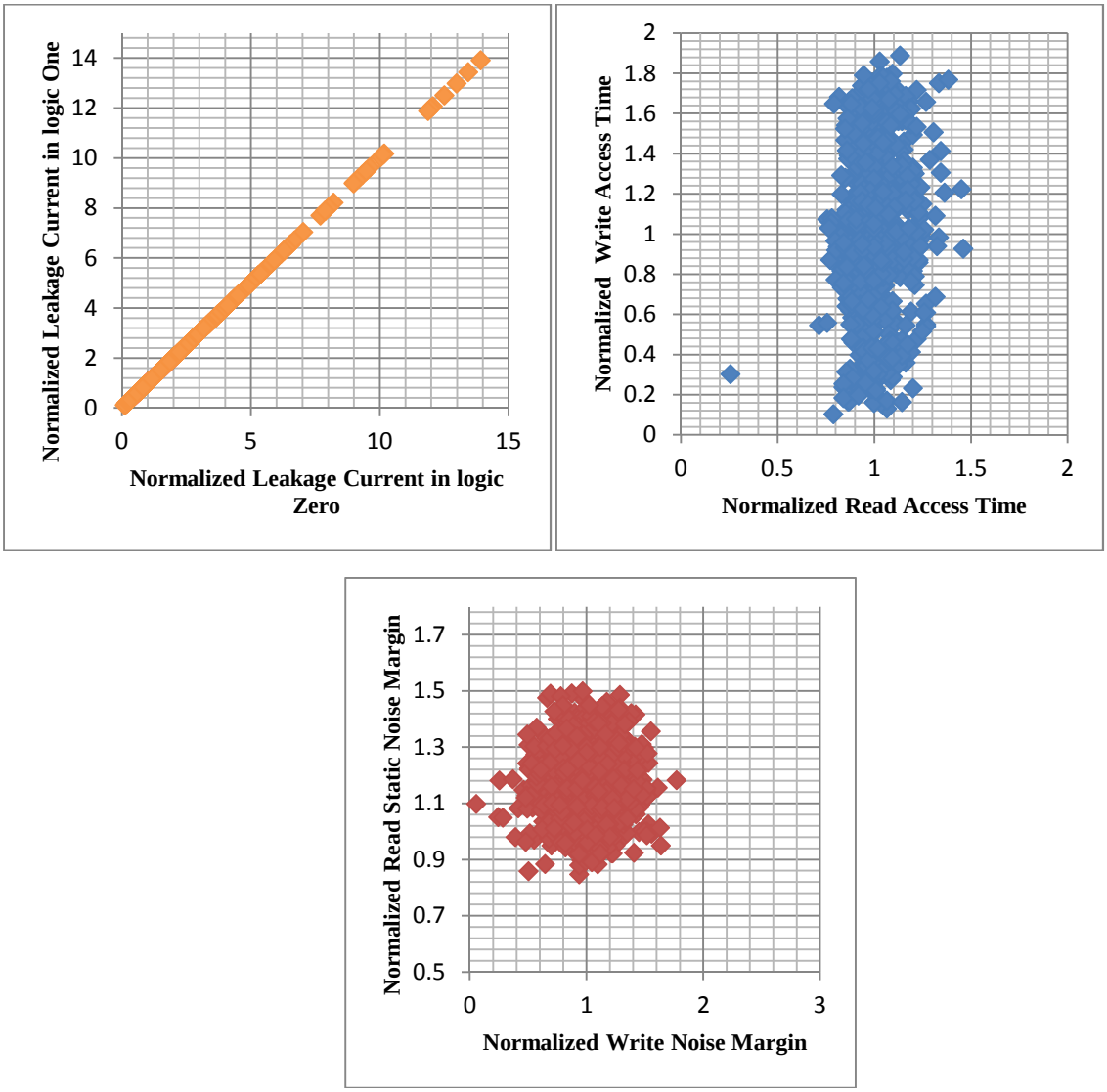
شکل ۲-۱۵ انواع پارامترهای سلول‌های جدید را تحت تأثیر تغییرآت فرآیند ساخت را نمایش می‌دهد. همچنین شکل ۲-۱۶ انواع پارامترهای سلول شش ترانزیستوری پایه را تحت تأثیر تغییرات فرآیند ساخت به منظور مقایسه نمایش می‌دهد. در این شکل پارامترهای سلول‌ها نسبت به حالت معمولی که تغییرات فرآیند وجود ندارد نرمالایز شده‌اند.



شکل ۲-۱۴: شکل‌موج شبیه‌سازی شده برای گره‌های ST و STB سلول 10T-H-NC در اثر تغییرات فرآیند ساخت



شکل ۲-۱۵: پارامترهای متفاوت سلول‌های جدید تحت تأثیر تغییرات فرآیند ساخت (نمودارهای سمت راست مربوط به سلول 10T-IL-H-NC است و نمودارهای سمت چپ مربوط به سلول 10T-H-NC است)



شکل ۲-۱۶: پارامترهای متفاوت سلول شش ترانزیستوری پایه تحت تأثیر تغییرات فرآیند ساخت

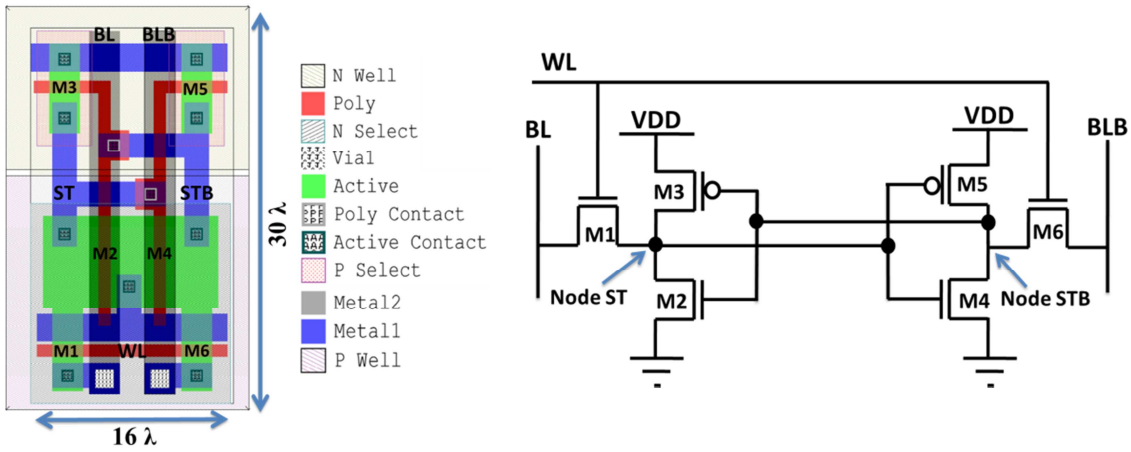
فصل سوم

نتایج و بحث

۳-۱- مقدمه

در این فصل به بیان کارهای مرتبط و مقایسه آنها و با کارهایی جدیدی که در فصل ۲ طراحی شداند می‌پردازیم. به طورکلی کارهای مرتبط قبلی را می‌توان به ۴ دست تقسیم کرد که عبارتند از: ۱- سلول شش ترانزیستوری پایه[1] ۲- سلول‌های آگاه از صفر با جریان نشتی کم ۳- سلول‌های سخت شده ۴-سلول‌های با حاشیه نویز ایستای خواندن آزاد. در ادامه به معرفی و تحلیل سلول‌های هر دسته می‌پردازیم و آنها را با سلول‌های جدید که در این طرح تحقیقاتی طراحی شده‌اند مقایسه می‌کنیم.

Word-Line : WL Bit-Line : BL Bit-Line-Bar : BLB ST : Storage STB : Storage-Bar



(ب) (الف)

شکل ۳-۱: (الف) شماتیک مداری این سلول شش ترانزیستوری پایه SRAM (یا به صورت اختصار 6T Cell) (ب) Layout آن در قوانین طراحی مقیاس‌پذیر (β=۳ و γ=۳)

۳-۲- سلول شش ترانزیستوری پایه

شکل ۳-۱ شماتیک مداری و Layout یک سلول شش ترانزیستوری پایه را نمایش می‌دهد. مبنای ذخیره سازی داده در این سلول استفاده از فیدبک‌های مثبتی است که توسط دو معکوس کننده ضربدری

[1] Conventinal Six Transistor SRAM Cell (6T Cell)

ایجاد می‌شوند. و توسط ترانزیستورهای دستیابی داده در این سلول نوشته و یا از آن خوانده می‌شود. در این سلول دو پارامتر وجود دارد روی مشخصات سلول تأثیر گذار هستند که به صورت زیر تعریف می شوند.

β=Drivability of pull-Down Transistors (M2 and M4)/ Drivability of Acess Transistors (M1 and M6)
γ=Drivability of Access Transistors (M1 and M6)/ Drivability of pull-up Transistors (M3 and M5)

۳-۳- سلول‌های آگاه از صفر با جریان نشتی کم

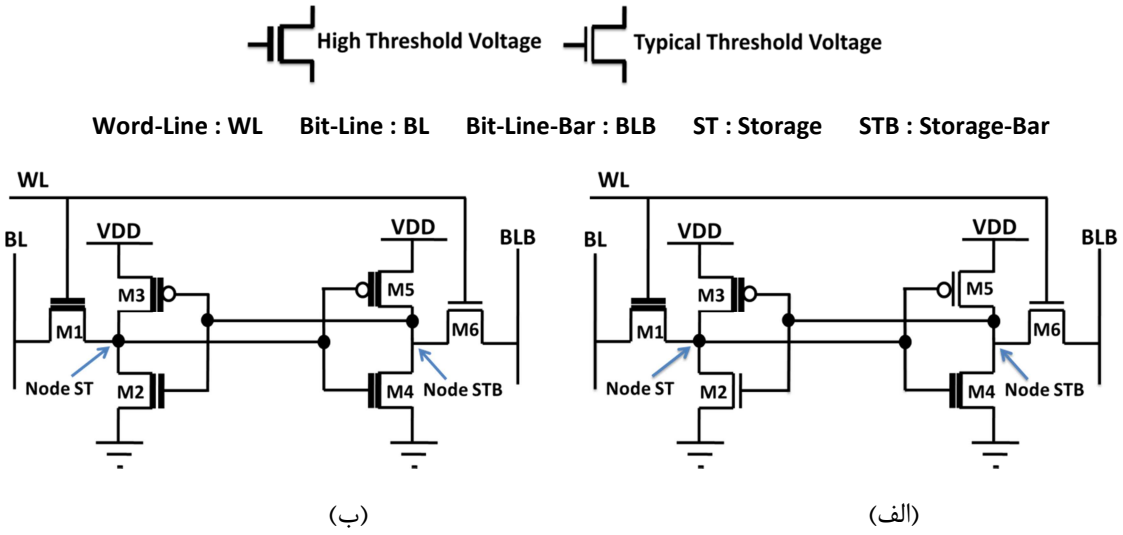
بر مبنای این مشاهده که بیشتر داده‌های موجود در جریان بیت اکثر کاربردها صفر هستند، دو سلول نا متقارن با جریان نشتی کم در مراجع [۱۷] ارائه شده‌اند. در این سلول‌ها بار بحرانی توسط استفاده از ترانزیستورها با دو ولتاژ آستانه در زمانی که صفر در سلول ذخیره شده است افزایش یافته است. شکل ۳-۲ شماتیک مداری این دو سلول را نمایش می‌دهد. همچنین این دو سلول نامتقارن جریان نشتی سلول را در حالتی که داده صفر در سلول ذخیره شده است را کاهش می‌دهند، به این ترتیب باعث کاهش متوسط جریان نشتی می‌شوند. این سلول‌ها فقط باربحرانی را زمانی که داده صفر در سلول ذخیره می‌شود را افزایش می‌دهند، بنابراین یک ذره با انرژی کافی می تواند داده این سلول‌ها را عوض کند. حال آنکه در تمامی سلول‌های جدیدی که در فصل ۲ طراحی شده‌اند، زمانی که داده صفر یا یک در سلول ذخیره شده است برخورد ذرات نمی‌تواند داده سلول را عوض کند. بنابراین سلول‌های جدیدی که طراحی شده‌اند دارای نرخ خطای نرم کمتری نسبت به سلولهای BAC و ILAC هستند. جدول ۳-۱ این سلول‌ها را با سایر سلول‌ها مقایسه می‌کند.

۳-۴- سلول‌های سخت شده

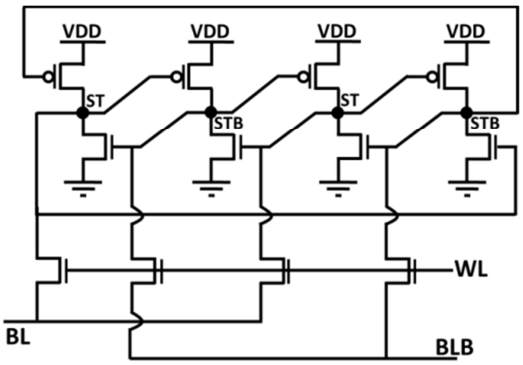
یکی از شناخته‌ترین سلول‌های سخت شده در مرجع [۱۹] با نام DICE[۱] ارائه شده است. شماتیک مداری این سلول در شکل ۳-۳ نمایش داده شده است. در قوانین طراحی Layout یکسان این سلول ۸۱ درصد از سلول شش ترانزیستوری پایه بزرگتر است [۱۹]. این سلول به صورت کامل سخت است و داده آن در مقابل پالس‌های ناخواسته ناشی از برخورد ذرات عوض نمی‌شود. به طور کلی سلول‌های سخت جدید

[۱] Dual Interlocked Cell

طراحی شده دارای جریان نشتی کمتری نسبت به DICE هستند. جدول ۳-۱ این سلول‌ها را با سایر سلول-ها مقایسه می‌کند.



شکل ۳-۲: (الف) شماتیک مداری BAC[۱] (ب) ILAC[۲]

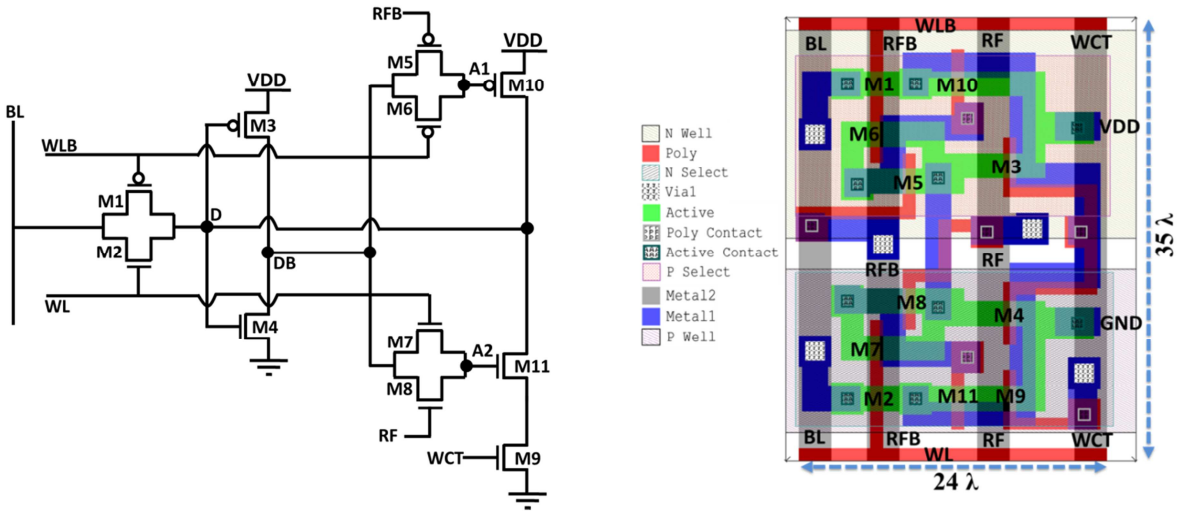


شکل ۳-۳: شماتیک مداری Dual Interlocked Cell (یا به صورت اختصار DICE)

سلول سخت شده دیگری که از سیگنال Refresh استفاده میکند با نام 11TRHC[۳] در مرجع [۲۰] ارائه شده است. این سلول از ۱۱ ترانزیستور استفاده می‌کند. شکل ۳-۴ شماتیک مداری و Layout این سلول را نمایش می‌دهد. زمانی که Word-Line در سطح ولتاژ بالا است Word-Line-Bar در سطح ولتاژ پایین است، ترانزیستورهای M6 و M7 روشن هستند و سلول به صورت نرمال کار می‌کند. اما زمانی که Word-Line در

[۱] Asymmetric memory SRAM Cell
[۲] Improved-Leakage Asymmetric SRAM Cell
[۳] 11T Refresh Hardend Cell

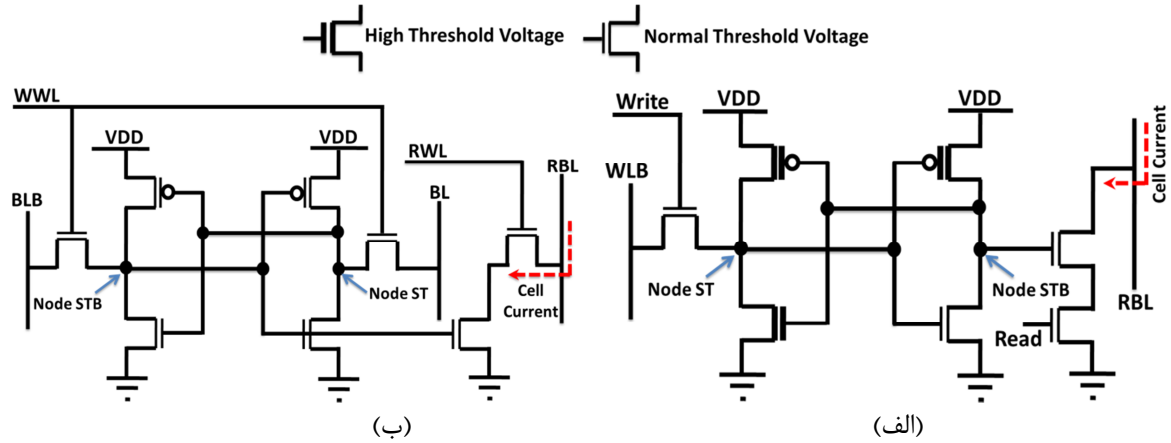
سطح ولتاژ پایین است Word-Line-Bar در سطح ولتاژ بالا است، ترانزیستورهای M6 و M7 خاموش هستند و مسیر فیدبک قطع است در نتیجه پالس‌های ناخواسته نمی‌توانند در طول مسیر فیدبک انتشار پیدا کنند و به نقطه شروع خود برسند و داده سلول را عوض کنند. به علاوه در این سلول گیت ترانزیستورهای M10 و M11 از یکدیگر جدا شده‌اند تا گره A1 در مقابل پالس های ناخواسته منفی و گره A2 در مقابل پالس‌های ناخواسته مثبت سخت شوند. . زمانی که ترانزیستورهای عبوری مسیر فیدبک را قطع می‌کند، گره-های A1 و A2 توسط بار الکتریکی که روی خازن‌های پارازیتری این گره‌ها وجود دارد وضعیت خود را حفظ می‌کنند. اما با گذشت زمان بار این گره‌ها کاهش می‌یابد. برای حل این مشکل سیگنال‌های RF و RFB به گیت ترانزیستورهای M8 و M5 وصل می شوند تا این دو ترانزیستور را به صورت لحظه‌ای روشن کنند و گره‌های A1 و A2 بتوانند بار خود را ترمیم کنند. سلول 11TRHC از سیگنال تازه ساز استفاده می‌کند و این سیگنال تازه ساز باید به تمامی سلول‌های حافظه پیکربندی FPGA مسیریابی و اعمال شود[۲۰]. درنتیجه بار خازنی خط سیگنال تازه ساز بسیار بالا است و توان مصرفی پویای قابل توجه‌ای در سیکل‌های تازه سازی مصرف می‌شود. به طور کلی سلول‌های سخت جدید طراحی شده دارای جریان نشتی کمتری نسبت به 11TRHC هستند. جدول ۳-۱ این سلول را با سایر سلول‌ها مقایسه می‌کند.



شکل ۳-۴: شماتیک مداری 11T Refresh Hardened Cell (یا به صورت اختصار 11TRHC) و Layout آن در قوانین طراحی مقیاس‌پذیر

۳-۵- سلول‌ها با حاشیه نویز ایستای خواندن آزاد

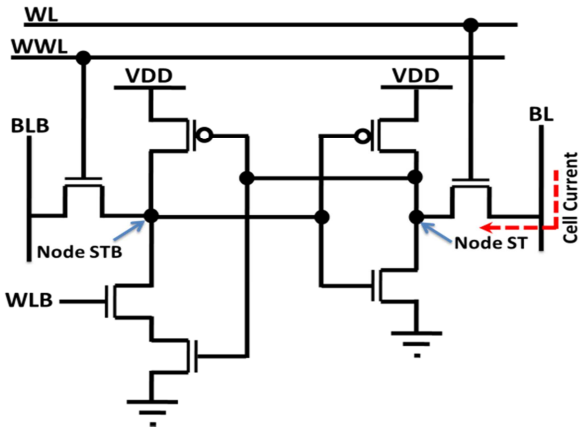
به طور کلی در سلول شش ترانزیستوری پایه[۱] جریان سلول و حاشیه نویز ایستای خواندن به شدت به قدرت جریان دهی ترانزیستورهای دستیابی سلول بستگی دارند. با افزایش قدرت جریان دهی ترانزیستورهای دستیابی، جریان سلول افزایش پیدا می کند ولی حاشیه نویز ایستای خواندن کاهش پیدا می کند. بنابراین در سلول شش ترانزیستوری پایه حاشیه نویز ایستای خواندن، یک رابطه معکوس با جریان سلول دارد و دستیابی به یک سلول شش ترانزیستوری پایه با حاشیه نویز ایستای خواندن بالا و جریان سلول زیاد بسیار مشکل است. دو استراتژی کلی که برای رفع این مشکل سلول شش ترانزیستوری پایه وجود دارد. استراتژی اول این است که، عناصری که داده را نگهداری می کند از عناصری که داده را می خوانند جدا کنیم. دو سلول بر مبنای این استراتژی گزارش شده است[۲۱،۲۲]. شکل ۳-۵ شماتیک مداری این دو سلول را که بر مبنای این استرانژی طراحی شده اند را نمایش می‌دهد.



شکل ۳-۵: (الف) شماتیک مداری Dual-Port Dual VT Cell طراحی شده در مرجع [۲۱] (ب) شماتیک مداری Dual-Port Cell طراحی شده در مرجع [۲۲]

[۱] Conventinal Six Transistor SRAM Cell (6T Cell)

استراتژی دوم این است که، در زمان انجام عمل خواندن برای جلوگیری از عوض شدن داده در زمان عمل خواندن، فیدبک (LOOP) موجود بین دو معکوس کننده توسط یک ترانزیستور قطع شود. نام این روش LOOP-Cutting است. شکل ۳-۶ شماتیک مداری این سلول را که بر مبنای این استرانژی طراحی شده است را نمایش می‌دهد.



شکل ۳-۶: شماتیک مداری Loop-Cutting Cell طراحی شده در مرجع [۲۳]

۳-۶- مقایسه کارایی سلول‌های متفاوت

در این قسمت بر مبنای نتایج شبیه‌سازی‌هایی که ما در این طرح تحقیقاتی انجام داده‌ایم، جدول ۳-۱ بار بحرانی و نرخ خطاهای نرم را در سلول‌های جدید طراحی شده و تمامی سلول‌هایی که در بالا ذکر شده‌اند نمایش می‌دهد. نرخ خرابی داده‌های سلول‌ها در اثر برخورد ذرات انرژی‌دار را نرخ خطای نرم گویند و با افزایش بار بحرانی به صورت نمایی کاهش می‌یابد[۲۴]. واحد نرخ خطای نرم خرابی در زمان[۱] است. یک FIT برابر با یک خرابی در 10^9 ساعت است [۱۵]. به طورکلی رابطه زیر برای محاسبه نرخ خطای نرم در یک سلول حافظه استفاده می شود.

$$SER \propto N_{flux} \times A_{node-i} \times e^{-\frac{Q_{Crit-node-i}}{Q_S}} \quad (۵)$$

در رابطه بالا N_{flux} شدت نوترون ها، A_{node-i} مساحت گره حساس iام سلول، Q_S کارایی یا میزان تجمع بار و $Q_{Crit-node-i}$ بار بحرانی گره iام است.

[۱] Failure in Time (FIT)

جدول ۳-۱- مقایسه کارایی سلول‌های متفاوت

SRAM Cells / Metrics	Critical Charge of Different nodes				Soft Error Rate		Leakage Current (V_{DD}=0.95V)		Normalized Area cell	Read mode	Write Mode	Read Static Noise Margin	Using Refresh Signal
	ST 1 to 0	ST 0 to 1	STB 1 to 0	STB 0 to 1	Storing 0	Storing 1	Storing 0 (Common Case)	Storing 1					
6T Cell (β=3, γ=3)	59fC	91fC	59fC	91fC	6934 FIT	6934 FIT	316nA	316nA	1	DIFF	DIFF	Limit	No
BAC	18fC	136fC	245fC	32fC	2900 FIT	10443 FIT	1nA	309nA	1	DIFF	DIFF	Limit	No
ILAC	50fC	80fC	60fC	80fC	7190 FIT	7528 FIT	0.07nA	116nA	1	DIFF	DIFF	Limit	No
Loop-Cutti-ng cell	72fC	131fC	77fC	88fC	5535 FIT	6607 FIT	314nA	315nA	1.13	SED	DIFF	Free	No
Dual-Port cell	63fC	107fC	72fC	109fC	6155 FIT	6386 FIT	506nA	216nA	1.3	SED	DIFF	Free	No
Dual-Port Dual V_T cell	40fC	23fC	30fC	89fC	10300 FIT	7658 FIT	150nA	81nA	1.05	SED	SED	Free	No
DICE	∞	∞	∞	∞	0	0	355nA	355nA	1.81	DIFF	DIFF	Limit	No
RC	∞	∞	∞	∞	0	0	480nA	285nA	1.75	SED	SED	Limit	Yes
BC (δ=4)	∞	∞	∞	∞	0	0	75nA	296nA	1.83	DIFF	DIFF	Free	No
ILC (δ=5)	∞	∞	∞	∞	0	0	3.5nA	293nA	1.66	SED	DIFF	Free	No

DIFF: Differential **SED: Single-Ended**

FIT: one failure in one billion hours.

Free: there is no correlation between read SNM and SRAM cell current

Limit: there is an inverse correlation between read SNM and SRAM cell current.

فصل چهارم

نتیجه‌گیری و پیشنهادات

۴-۱- نتیجه‌گیری

با پیشرفت تکنولوژی CMOS و کوچکتر شدن ابعاد ترانزیستورهای نرخ خطای نرم در حال افزایش است. بنابراین نرخ خطای نرم چالش اساسی در پیاده‌سازی سیستم‌های چند عامله بر روی سخت افزارهای قابل پیکربندی در تکنولوژی CMOS با ابعاد نانو است. در یک سخت افزارهای قابل پیکربندی مانند FPGA خطاهای نرمی که در حافظه پیکربندی رخ می‌دهد باعث خرابی سیستم چند عامله‌ی پیاده‌سازی شده بر روی سخت افزارهای قابل پیکربندی می‌شود. بنابراین در این طرح تحقیقاتی، برای حل این چالش اساسی در پیاده‌سازی سیستم‌های چند عامله بر روی سخت‌افزارهای قابل‌پیکربندی، سلول‌هایی حافظه پیکربندی جدید طراحی شده است. در این سلول‌های زمانی که صفر یا یک در سلول ذخیره شده است برخورد ذرات انرژی‌دار با گره‌های حساس نمی‌تواند داده سلول را عوض کند، در نتیجه نرخ خطای نرم در این سلول‌ها صفر است. در تمامی این سلول‌های جدید جریان نشتی با استفاده از دو ولتاژ آستانه برای ترانزیستورها و پشته‌ای از ترانزیستورهای خاموش سری کاهش یافته است. تمامی سلول‌هایی که در این طرح تحقیقاتی ارائه شده‌اند دارای سربار مساحتی هستند که مقدار این سربار مساحتی بسته به نوع سلول متفاوت است، اما در تکنولوژی CMOS با ابعاد نانو به دلیل تجمع بالای ناشی از کاهش ابعاد ترانزیستورها سربار مساحتی سلول‌های طراحی شده در این طرح تحقیقاتی پارامتر محدود کنند مهمی نیست.

۴-۲- پیشنهادات

زمانی که یک ذره با انرژی بالا به درین یا سورس یک ترانزیستور MOS در سلول حافظه برخورد می-کند به علت تقسم بار[۱] چندین گره را به صورت همزمان تحت تأثیر قرار می‌دهد[۲۵]. زمانی چندید گره یک سلول به صورت همزمان تحت تأثیر برخورد یک ذره قرار می‌گیرند داده آن عوض می شود. تمامی سلول-

[۱] Charge Sharing

هایی که تاکنون در کارهای گذشته ارائه شده‌اند و سلول‌های جدیدی که در فصل ۲ طراحی شده‌اند، زمانی که یک ذره با انرژی بالا به یکی از گره‌های حساس برخورد می‌کند و در اثر تقسیم بار چندین گره تحت تأثیر قرار می‌گیرد، داده آنها عوض می‌شود. تقسیم بار بشدت تابع‌ای از Layout سلول و فاصل بین گره ضربه خورده و گره‌های مجاور است [۲۶].

از طرف دیگر در تکنولوژی‌های CMOS آینده که ابعاد ترانزیستورها بسیار کوچک (مانند تکنولوژی ۱۶-nm) خواهد بود مسئله تقسیم بار بسیار بیشتر خواهد شد. بنابراین در تکنولوژی‌های CMOS آینده مسئله تقسیم بار یکی از چالش‌های اساسی در طراحی سلول حافظه پیکربندی است. بنابراین طراحی سلول-های حافظه پیکربندی سخت شده با توان مصرفی ایستای پایین که پدیده تقسیم بار نتواند داده سلول را عوض کند می‌تواند به عنوان هدف کارهای تحقیقاتی بعدی مطرح گردد.

مراجع

[1] J. Lamoureux and W. Luk, An overview of low-power techniques for field-programmable gate arrays, NASA/ESA Conference on Adaptive Hardware and Systems, 2008, pp. 338-345.

[2] Hamid Reza Naji, B. E. Wells, L. Etzkorn," Creating an adaptive embedded system by applying multi-agent techniques to reconfigurable hardware", **Elsevier** Journal of Future Generation Computer Systems, 2004, pp. 1055-1081.

[3] C. L. Hsu, and T. H. Chen, "Built-in self-test design for fault detection and fault diagnosis in SRAM-based FPGA", **IEEE** Transactions on Instrumentation and Measurement Vol. 58 NO.7, 2009, pp. 2300-2315.

[4] J. H. Anderson, and F. N. Najm, "Active leakage power optimization for FPGAs", **IEEE** Transactions on Computer-Aided Design of Integrated Circuits and Systems Vol. 25 No. 3, 2006, pp. 423-437.

[5] S. Hauck, and A. Dehon, Reconfigurable Computing, **Morgan Kaufman**, 2008.

[6] Xilinx Virtex-5 FPGA User Guide, "www.xilinx.com/support/documentation/user_guide"

[7] Hamid Reza Naji, "Solving Complex Computational Problems Using Multiagents Implemented in Hardware", **IEEE** Journal of Computing in Science and Engineering, pp. 54-63, 2008.

[8] V. Chandra and R. Aitken, Impact of technology and voltage scaling on the soft error susceptibility in nanoscale CMOS, **IEEE** International Symposium on Defect and Fault Tolerance of VLSI Systems, 2008, pp. 114-122.

[9] H. Ebrahimi, M. SahebZamani, and H. R. Zarandi, Mitigating soft errors in SRAM-based FPGAs by decoding configuration bits in switch boxes, **Elsevier** Microelectronics Journal Vol. 42 , 2011, pp.12–20.

[10] M. Omana, D. Rossi and C. Metra, Latch susceptibility to transient faults and new hardening approach, **IEEE** Transactions on Computers, Vol. 56 No. 9, 2007, pp. 1255-1268.

[11] A. Ejlali, B.M. Al-Hashimi, M.T. Schmitz, P. Rosinger, S.G. Miremadi, Combined time and information redundancy for SEU-tolerance in energy-efficient real-time systems, **IEEE** Transactions on Very Large Scale Integration Systems Vol. 14 No. 4, 2006, pp. 323- 335.

[12] M. Omana, D. Rossi and C. Metra, Latch susceptibility to transient faults and new hardening approach, **IEEE** Transactions on Computers Vol. 56 No. 9, 2007, pp. 1255-1268.

[13] Gh. Asadi, and M. B. Tahoori, Soft error rate estimation and mitigation for SRAM based FPGAs, in: Proceeding of 13th international symposium on Field-programmable gate arrays, 2005, pp. 149-160.

[14] S. Srinivasan, A. Gayasen, N. Vijaykrishnan, M. Kandemir, Y. Xie, and M. J. Irwin, Improving soft-error tolerance of FPGA configuration bits, International Conference on Computer Aided Design, 2004, pp. 107-110.

[15] B. S. Gill, G. Papachristou, and F. G. Wolff, A new asymmetric SRAM cell to reduce soft errors and leakage power in FPGA, Design Automation & Test in Europe Conference & Exhibition, 2007, pp.1-6.

[16] A. Lodi, Luca Ciccarelli, and R.Guerrieri, Low leakage techniques for FPGAs, **IEEE** Journal of Solid-State Circuits Vol. 41 No. 7, 2006, pp. 1662-1672.

[17] N. Azizi, F. Najm, and A. Moshovos, Low-leakage asymmetric-cell SRAM, **IEEE** Transactions on Very Large Scale Integration Systems Vol. 11 No. 4, 2003, pp. 701–715.

[18] D. A. Patterson, and J. L. Hennssy, Computer Organization & Design: The Hardware / Software Interface, 2edtion, **Morgan Kaufmann**, 1998, pp. 562-564.

[19] T. Calin, M. Nicolaidis, and R. Velazco, Upset Hardened Memory Design for Submicron CMOS Technology, **IEEE** Transactions on Nuclear Science, Vol. 43 No. 6, 1996, pp. 2874 - 2878.

[20] Sh. Lin, Y. B. Kim, and F. Lombardi, A 11-Transistor Nanoscale CMOS Memory Cell for Hardening to Soft Errors, **IEEE** Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 19 No. 5, 2011, pp. 900-904.

[21] Tawfik, S.A., and Kursun, V., "Low Power and Robust 7T Dual-Vt SRAM Circuit", IEEE International Symposium on Circuits and Systems, 2008, p.1452 - 1455.

[22] L. Chang et al., "Stable SRAM cell design for the 32 nm node and beyond," in Symp. VLSI Technology Dig., 2005, pp. 128–129.

[23] K. Takeda et al., "A read-static-noise-margin-free SRAM cell for low-VDD and high-speed applications," **IEEE** Journal of Solid-State Circuits, Vol. 41, No. 1, 2006.

[24] B. Amelifard, F. Fallah, and M. Pedram, Leakage minimization of SRAM cells in a dual-Vt and dual-Tox Technology, **IEEE** Transactions on Very Large Scale Integration Systems Vol. 16 No. 7, 2008, pp. 851-860.

[25] Sh. M. Jahinuzzaman, D. J. Rennie, and M. Sachdev, A Soft Error Tolerant 10T SRAM Bit-Cell with Differential Read Capability, **IEEE** Transactions Nuclear Science Vol. 56 No. 6, 2009, pp. 3768-3773.

[26] O. A. Amusan, L. W. Massengill, M. P. Baze, A. L. Sternberg, A. F. Witulski, B. L. Bhuva, and J. D. Black, Single event upsets in deep-submicrometer technologies due to charge sharing, **IEEE** Transactions on Device and Materials Reliability Vol. 8 No. 3, 2008, pp. 582–589.

Abstract:

Reconfigurable hardware is ideal for adaptive multi-agent systems, since it is reconfigurable and can be programmed to implement any digital logic. Therefore, we can implement any combinational and sequential digital logic on silicon infrastructure by using reconfigurable hardware. Devices in CMOS Technology have been scaled down aggressively with each technology generation to achieve a higher integration density and performance. An application in multi-agent system form can be implemented on reconfigurable hardware by configuring bit-stream which produced by CAD tools. However, the aggressive scaling of CMOS technology must also consider the soft error rate due to Ionizing Radiation phenomena. In nanao-scaled CMOS technology due to the lower supply voltage and the smaller capacitance, the amount of charge stored on a circuit node is increasingly smaller, thus energy particles travelling through the silicon bulk can create minority carriers. These carriers may be collected by the source/drain diffusion and alter the voltage value of the nodes. Therefore, an application in multi-agent system form fails on reconfigurable hardware, if data of configuration memory of reconfigurable hardware changes due to soft error rate. Thus, in nano-scaled CMOS technology, the reduction of soft error rate is the most important challenges in implementing multi-agent systems on reconfigurable hardware. To overcome these difficulties, based on the observations that most configuration bit-streams of reconfigurable hardware such as FPGA are zeros across different designs and that configuration memory cells are not directly involved with signal propagation delays in reconfigurable hardware such as FPGA, this research work presents hardened cell. These new hardened cells are completely hardened and cannot flip from particle strikes at sensitive cell

nodes. Also, in these new configuration memory cells, leakage current of cells is reduced by using dual-threshold voltage and stacked effect.

Keywords: ***Multi-Agent Systems, Reconfigurable Hardware, Configuration Memory Cell, Particle Strike, Leakage current, Cell Area, Critical Charge, Soft Error Rate.***