گزارشی از مقاله

Design High-Confidence Computers using Trusted Instructional Set Architecture and Emulators

طراحی کامپیوترهایی با قابلیت اعتماد بالا با استفاده از معماری مجموعه دستورالعمل‌های ایمن شده و شبیه‌سازها [1]

شاونگ بائو

ارائه از : **مسعود باقری**

شماره دانشجویی: **۳۹۹۱۲۳۴۱۰۵۷۰۴۱**

ترم تحصیلی: **۲-۹۹**

درس: **معماری پیشرفته**

استاد: **دکتر جاسبی**

این مقاله در ژورنال پذیرفته شده ولی هنوز نهایی و چاپ نشده و ممکن است در طول مسیر در محاسبات و یافته های علمی آن تغییراتی حاصل شود.

[1] Emulation در لغت به معنای برتری‌جویی و انجام کاری دقیقا شبیه و یا بهتر از شخص دیگری می‌باشد. به عنوان مثال اگر من بگویم " I emulate my brother in piano" بدین معناست که من در پیانو زدن همانند برادرم و حتی بهتر از او می‌باشم. این لغت در انگلیسی روزمره دقیقا متضاد لغت Simulate که به معنای شبیه‌سازی و تظاهر در عمل است می‌باشد. مثلا اگر من بگویم " I simulate happiness" یعنی اینکه من تظاهر به شادمانی می‌کنم. اما هر گاه در اصطلاح کامپیوتری از لغت simulate استفاده شد، منظور شبیه‌سازی کامل یک سیستم است، یعنی یک سیستم کامل با هارد دیسکها و ram و cd-drive ها و غیره را درون یک سیستم‌عامل دیگر شبیه‌سازی کنیم. اما هرگاه از اصطلاح emulate استفاده شد، منظور تنها شبیه سازی قسمتی از یک کامپیوتر واقعی است تا یک برنامه خاص را بتوان بر روی سیستم عاملهای مختلف اجرا نمود. برای اجرای یک نرم‌افزار خاص همچون photoshop درون لینوکس لازم نیست که کل ویندوز را درون لینوکس شبیه سازی کرد (که اگر اینکار را بکنیم قطعا photoshop هم در آن اجرا می‌شود)، بلکه تنها لازم است آن تکه از سخت‌افزارها و دستورالعملهایی که photoshop برای اجرا به آنها نیاز دارد را شبیه‌سازی کنیم که به این عمل emulation گوییم.[ب۱]

❖ **تعریف مسأله و هدف اصلی مقاله:**

سوال اصلی مطرح شده در مقاله:

محاسبات با اطمینان بالا به سخت افزار و نرم افزار قابل اعتماد نیاز دارند. مجازی سازی سنگ بنا یا زیربنای رایانش ابری است و عملکرد، کشش و قابلیتی را فراهم می‌کند که نمی‌توان آنرا در دستگاه‌های محاسباتی معمول به‌دست آورد. عوامل بسیاری وجود دارد که می‌تواند امنیت را تحت تأثیر قرار دهد. در سطح نرم افزاری، بکدور یا درب‌های پشتی می‌توانند در برنامه‌های کاربردی وجود داشته باشند که آسیب پذیری سیستم‌ها را به همراه می‌آورند. در سطح کامپایلر، بهینه سازی می‌تواند عملکرد طبیعی را تغییر دهد بنابراین امنیت آن به خطر بیافتد و نقض شود. در سطح سخت افزاری، پیش‌بینی کننده انشعاب و خطوط لوله، می‌توانند فعالیت‌های شرطی یا اطلاعات انشعاب و پرش را از طریق روبرداری[۲] از کش سطح ۳ نشت دهند، جایی که همه میزبان‌های مجازی، حافظه کش پنهان یکسانی دارند. در سطح تراشه، سازندگان، با دانستن اینکه ۱۰۰ درصد از دروازه‌ها[۳] در هر پردازنده روی یک تراشه معمول استفاده نمی‌شوند، می‌توانند "بمب‌های منطقی" را در دروازه‌های باقیمانده قرار دهند.

توجه زیادی در طراحی، توسعه، شبیه سازی (سیمولاتور) و الگوبرداری یا تقلید (امولاتور) از پردازنده‌های جدید که در برابر سو استفاده‌ها و حملات ایمن هستند، لازم است. و سوال اصلی این است که؛ آیا می‌توان در طراحی پردازنده‌های نسل جدید، از روش‌هایی استفاده کرد که امنیت لازم تأمین شده و بتوان حملات کانال جانبی را دفع نمود؟

چه مشکلی باید برطرف شود؟

Spectre [۴] و Meltdown[۵] ، پردازنده‌های مدرنی را که از پیش‌بینی انشعاب[۶] و خطوط لوله[۷] استفاده می‌کنند، مورد سوء استفاده قرار داده و آسیب پذیر می‌کنند. محاسبات با اطمینان بالا، به معماری مجموعه دستورالعمل[۸] قابل اعتماد، هسته های نفوذناپذیر و غیرقابل تغییر و سیستم عامل های امن، متکی است. هدف

[2] Dump
[3] Gates
[۴] یک آسیب پذیری است که امکان خواندن مکان‌های دلخواه در حافظه اختصاص یافته به یک برنامه را فراهم می‌کند.

[۵] یک آسیب پذیری سخت افزاری است که بر ریز پردازنده‌های Intel x86 و برخی پردازنده‌های IBM POWER از ریز پردازنده های مبتنی بر ARM تأثیر می گذارد و به فرایند تقلبی اجازه می‌دهد تا تمام حافظه را بخواند ، حتی اگر مجاز به انجام آن نباشد.

[6] branch prediction
[7] pipelines
[8] ISA

از این تحقیق پیشنهاد روش‌هایی در طراحی پردازنده‌های نسل جدید عنوان شده که قابل اعتماد بوده و در برابر این حملات مصونیت دارند.

چه ضرورتی برای مطرح شدن مسئله است؟

حملات کانال جانبی و بطور مشخص Spectre و Meltdown، پردازنده‌های جدید را آسیب پذیر می‌کند و آنها را به عنوان نقطه‌ای جهت سواستفاده هکرها قرار می دهد. غیرفعال کردن پیش بینی کننده انشعاب و خط لوله قطعاً راه حل خوبی نیست. شاید برداشت عمومی اینطور باشد یا اینطور به نظر برسد که این یک مسئله مجازی سازی است، اما اگر به مسائل ناشی از Meltdown بپردازیم، این اتفاق در حافظه نهان داخلی موجود در CPU رخ می‌دهد، که یک حافظه اساسی است و از خطوط لوله در بهبود عملکرد CPU / GPU استفاده می‌شود. غیرفعال کردن خطوط لوله، افزایش سرعت کلی حاصل از موازی سازی را محدود می‌کند، بنابراین پردازنده ها به طور قابل توجهی، کند می‌شوند. درعین حال، حملاتی که از نشتی حاصل از اشکال زدایی استفاده می‌کنند، می‌تواند مهاجمان را به عبور از ماشین‌های مجازی در یک محیط رایانش ابری هدایت کنند و از طرف دیگر، وصله‌های نرم افزاری فعلی فقط می‌توانند مسائل غیر ضروری اطراف Meltdown را برطرف کنند. این مقاله بر روی امنیت در سطح معماری مجموعه دستورالعمل (ISA) تمرکز دارد و برای حل این مشکل به دنبال ارائه روشی اصلاحی در طراحی پردازنده‌ها می‌باشد.

چه روشهایی قبلا برای این کار انجام شده؟

- پیاده سازی ماشین تورینگ[9] با استفاده از مفهوم برنامه انباشته یا برنامه ذخیره شده است. ماشین تورینگ ، مفاهیم نظری محاسبات را به صورت ریاضی و مسیرهای حل این محاسبات را تعریف می کند.[۱]
- مجموعه دستورالعمل‌های مبتنی بر دسترسی پی‌درپی یا متوالی و محاسبات خطی برای مدت طولانی از دهه ۱۹۴۵ تاکنون بر معماری کامپیوتر مسلط بوده‌اند.
- جان. ال هنسی، مجموعه دستورالعمل‌های کاهش یافته RISC[10] را در دهه ۱۹۸۰ پیشنهاد کرد. در معماری RISC کلیه دستورالعمل‌ها با تعداد چرخه مساوی اجرا می‌شوند و این معماری بر

[9] Turing Machine

[10] Reduced Instruction Set Computer

ریزپردازنده‌های ساده و کم هزینه متمرکز شده بود[۲]. از آن زمان، معماری RISC ، نقش اصلی را در طراحی و توسعه پردازنده مرکزی دارد. [۳]

- اپل به تازگی اعلام کرده است که جهت هدایت عملکرد و قدرت برنامه های هوش مصنوعی و یادگیری ماشین، Mac را به سیلیکون سفارشی خود منتقل می‌کند. طرح سیلیکون جدید اپل، از معماری RISC همراه با بهبود امنیت برای کاهش خطرات مرتبط با حمله روز صفر[۱۱] و محافظت در برابر اجرای کد از راه دور[۱۲] استفاده می‌کند.
- مولدر و همکاران، معماری اصلاح شده ای را ارائه می‌دهد که در آن برای محافظت از نشت کانال جانبی برای مدار، تولید ماسک[۱۳] می‌شود. در این روش برای تولید ماسک که یک نوع پوشش است، از رمزگذاری بلوک، رمزگذاری جریان[۱۴]، توابع هش یا سایر موارد اولیه رمزنگاری استفاده می‌کند[۸]. این روش در محافظت از حملات کانال‌های جانبی، موثر نشان داده شده است. مسئله این است که محاسبات رمزنگاری بر روی یک پردازنده محاسبه می‌شود. پس از به خطر افتادن پردازنده، محافظت از بین خواهد رفت.
- صدیقی و همکاران، طرحی را برای استفاده از FPGA در پیاده سازی RISC-V برای جلوگیری از نشتی در برابر حملات بدافزار ارائه داد. این کار، روابط بین BIOS[۱۵] و نرم افزار را در هنگام راه‌اندازی برقرار می‌کند. این سیستم، از بوت مبتنی بر TPM[۱۶] برای ایمن سازی جریان داده درون پردازنده استفاده می کند [۹]. بوت امن مبتنی بر TPM، برای جلوگیری از نشتی در طول فرآیند راه‌اندازی موثر است. با این حال، کندی داده‌ها بین TPM و پردازنده می‌تواند داده‌ها را در معرض مزاحمان قرار دهد.

روش پیشنهادی ارائه شده چیست؟

QEMU[۱۷] در RISC-V ، به دلیل ویژگی های معماری منبع باز، میان سکویی[۱۸] بودن(یعنی می‌تواند بر روی بیشتر سیستم عامل‌ها نصب شود)، تأخیر کم و ویژگی‌های مجازی سازی بهینه شده، توجه صنعت و دانشگاه را به خود جلب کرده است تا یکی از برترین پلتفرم های معماری و شبیه سازی باشد [۱۰ ، ۱۱]. این مزایا

[11] zero-day exploit
[12] remote code execution
[13] masks
[14] stream ciphers
[15] Basic Input/Output System
[16] Trusted Platform Module
[17] Quick Emulator: is a generic and open source machine emulator and virtualizer.
[18] cross platform

باعث شده تا استفاده از آن به خصوص در محیط رایانش ابری بسیار مفید واقع شود. معماری قابل اعتماد مبتنی بر محاسبات ماشین مجازی QEMU، امنیت دستگاه‌های دارای این معماری را بهبود می‌بخشد [۱۱].

این تحقیق یک رویکرد جامع را برای افزایش امنیت و نفوذناپذیری پردازنده‌های مرکزی نسبت به آسیب‌های امنیتی کانال جانبی، نشت حافظه، کپی و روبرداری از حافظه کش، و حملات نشتی حاصل از اشکال زدایی که می‌تواند مهاجمان را به عبور از ماشین‌های مجازی در یک محیط رایانش ابری هدایت کند، پیشنهاد می‌کند. از تراشه تولید کننده اعداد تصادفی کوانتومی برای ارائه بالاترین امنیت، تصادفی سازی غیرقابل شکستن و غیر قابل پیش بینی و آنتروپی کامل فوری برای محاسبات مربوط به رمزنگاری استفاده می‌شود..

❖ توضیح راه حل پیشنهادی مقاله برای حل مسئله:

برای توضیح نحوه عملکرد روش پیشنهادی لازم است مروری بر نحوه عملکرد حملات کانال جانبی و بطور مشخص، Spectre و Meltdown داشته باشیم.

آسیب Spectre:

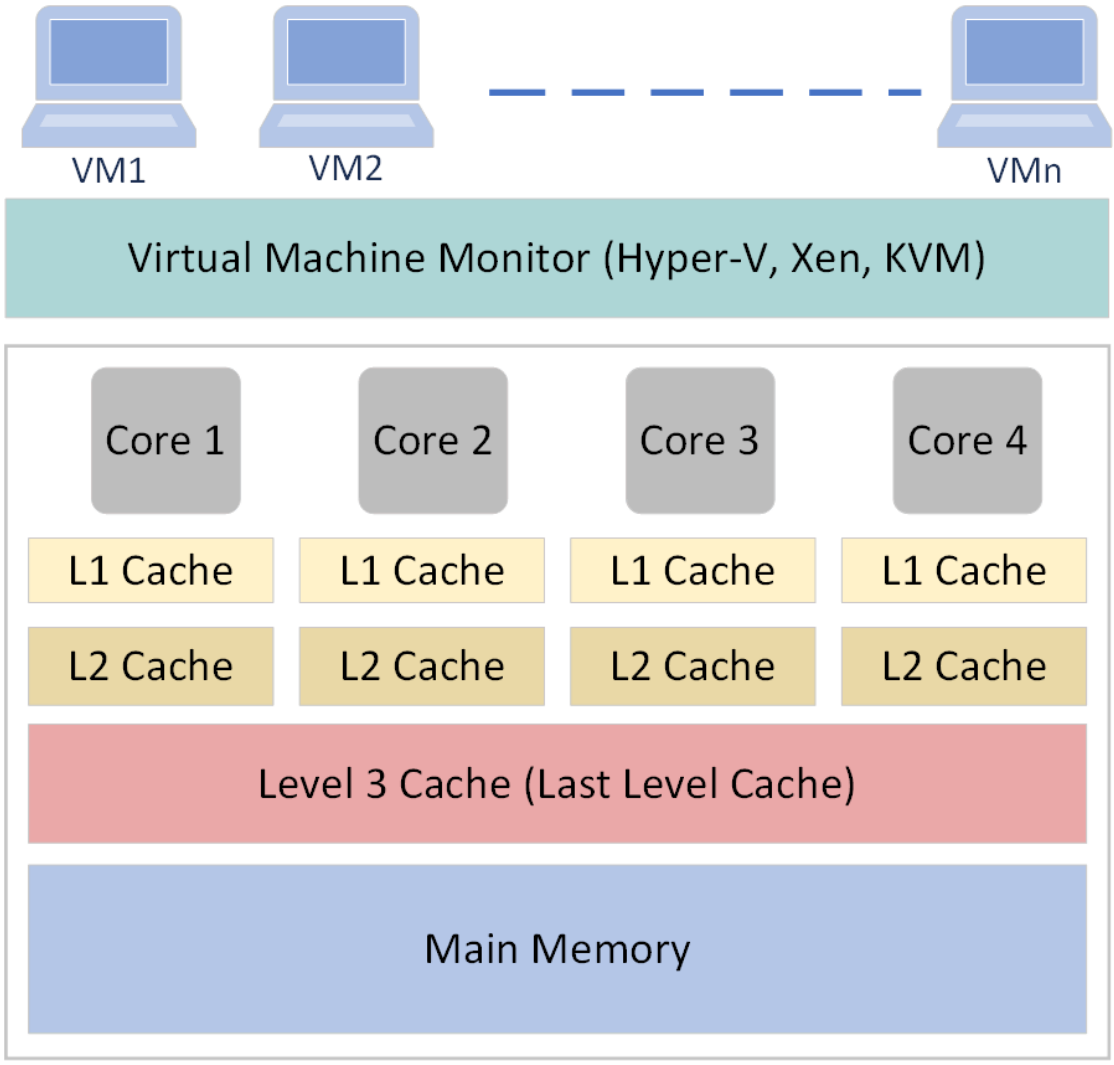
پردازنده‌های مدرن، از خطوط لوله و پیش بینی انشعاب برای به حداکثر رساندن عملکرد استفاده می‌کنند [۱۷]. پردازنده مرکزی (CPU)، مقصد را حدس می‌زند و سعی می‌کند دستورالعمل‌های پیشِ رو را برای اجرا و به حداکثر رساندن خط لوله یا پایپ لاین، واکشی کند یعنی بطور همزمان چند دستورالعمل یا داده را از حافظه مکان یابی و بارگذاری کند. هنگامی که یک پیش بینی"موفق" باشد، CPU آن را انجام می‌دهد، اگر "خطا" باشد، پردازنده آن را کنار می‌گذارد. حملات Spectre به دنبال همین "کنارگذاشته‌ها" هستند که اطلاعات قربانی را لو می‌دهد و از طریق یک کانال جانبی قربانی را به خطر می‌اندازد. این آسیب پذیری در بسیاری از پردازنده‌های رایج مورد استفاده در دستگاه‌های رایانش ابری و شخصی از جمله Intel، AMD و ARM مشاهده شده است.

روشهای مقابله با حمله Spectre ، شامل جلوگیری از اجراهای طولانی حدسی یا قیاسی است که منجر به کاهش قابل توجه عملکرد پردازنده می‌شود. اجراهای قیاسی اصلاح شده، امنیت را فراهم می‌کند اما عملکرد را کاهش داده؛ احراز هویت و مجوز دسترسی به هر وب سایت را در یک فرآیند جداگانه بهبود می‌بخشد[۱۸].

آسیب Meltdown:

خطوط لوله و اجرای خارج از دستور پویا در بهبود عملکرد پردازنده‌ها ضروری است. Meltdown از عوارض جانبی اجرای خارج از دستور در پردازنده‌های مدرن سو استفاده می‌کند. این حمله، مستقل از سیستم‌عامل یا سایر آسیب پذیری‌های نرم افزاری است. شکست در جداسازی فضای نشانی آدرس می‌تواند منجر به خواندن حافظه از سایر فرآیندها یا مجازی سازی توسط یک دشمن شود.

حافظه پنهان یا کش، حافظه‌ای سریعتر از حافظه اصلی است. پردازنده‌های مدرن دارای دو تا سه سطح حافظه پنهان هستند که هر سطح فضای بیشتری نسبت به سطح پایین‌تر از خود دارد ولی سرعت ارتباط با پردازنده به ترتیب کمتر می‌شود. حافظه نهان به صورت "فروش یکجا یا عمده" عمل می‌کند که در آن، داده‌ها نه بصورت قطعه‌ای بلکه بصورت بلوکی نوشته می‌شوند. حملات کانال جانبی حافظه پنهان، از تفاوت در زمان دسترسی حافظه استفاده می‌کنند. حملات Flush + Reload [۱۹] با خالی کردن مرتب یک مکان حافظه مورد هدف، از حافظه پنهان سطح آخر (سطح ۳) سو استفاده می‌کنند. اختلاف زمانی در بارگیری مجدد داده‌ها، نشانه‌ای از بارگیری داده‌ها توسط فرایندی دیگر است. پردازنده‌ها و حتی الگوریتم‌های رمزنگاری هر دو در برابر این حمله آسیب پذیر هستند. اگر یک کانال مخفی وجود داشته باشد، می‌تواند اطلاعات را از یک منطقه امنیتی به منطقه دیگر نشت دهد. در نتیجه، برای جلوگیری از چنین حملاتی، ضروری است ISA های قابل اعتمادی طراحی شود. شکل زیر یک معماری ابری معمول CPU را با سه سطح حافظه نهان نشان می‌دهد. از آنجا که حافظه پنهان سطح ۳ همیشه بین میزبان‌های مجازی به اشتراک گذاشته می‌شود، یک حمله کانال جانبی می‌تواند کلیدهای رمزنگاری را برباید.



متأسفانه، وصله‌های نرم افزاری که برای مقابله با حمله Meltdown رایج هستند، سربار قابل توجهی برای پردازنده‌ها به ارمغان می‌آورند ، بنابراین سرعت را کاهش می‌دهند. تقسیم حافظه سخت روش دیگری برای حل این مشکل است. اینتل و بسیاری از پردازنده‌ها از فناوری تقسیم حافظه استفاده کرده‌اند. اما فضای هسته اصلی یا کرنال موجود و تقسیم فضای کاربر "امن" تلقی می‌شود که این خود یک حفره محسوب می‌شود.

روش پیاده سازی شده برای حل مسئله مقاله به چه صورت است؟

همانطور که نویسندگان مقاله اشاره کردند، حملات کانال جانبی بسیار گسترده بوده و تقریبا تمام تولیدکنندگان بر این موضوع اتفاق نظر دارند که راهکار مقابله با حملات Spectre و Meltdown در اصلاح معماری پردازنده‌ها می‌باشد.

اصلاح معماری از دو بخش اصلاح مجموعه دستورالعمل‌ها و پیاده‌سازی تشکیل می‌شود که با توجه به مسایل مالکیت معنوی و عدم افشای اطلاعات لازم توسط تولیدکنندگان پردازنده‌های معروف و از طرف دیگر گسترش دامنه استفاده از معماری RISC-V به عنوان یک معماری متن باز، این تحقیق بر روی معماری مذکور انجام شده است.

اصلی‌ترین اقدام امنیتی درنظر گرفتن ماژول مجزای تعیین کننده اعتبار dTPM[۱۹] بصورت یک تراشه یا میکروکنترلر جداگانه و ایزوله شده است که تمام منابع محاسباتی لازم در این بسته تراشه مجزا وجود دارد. یک TPM مجزا، کنترل کاملی بر منابع داخلی اختصاصی از جمله RAM ، حافظه غیر فرار و منطق رمزنگاری دارد.

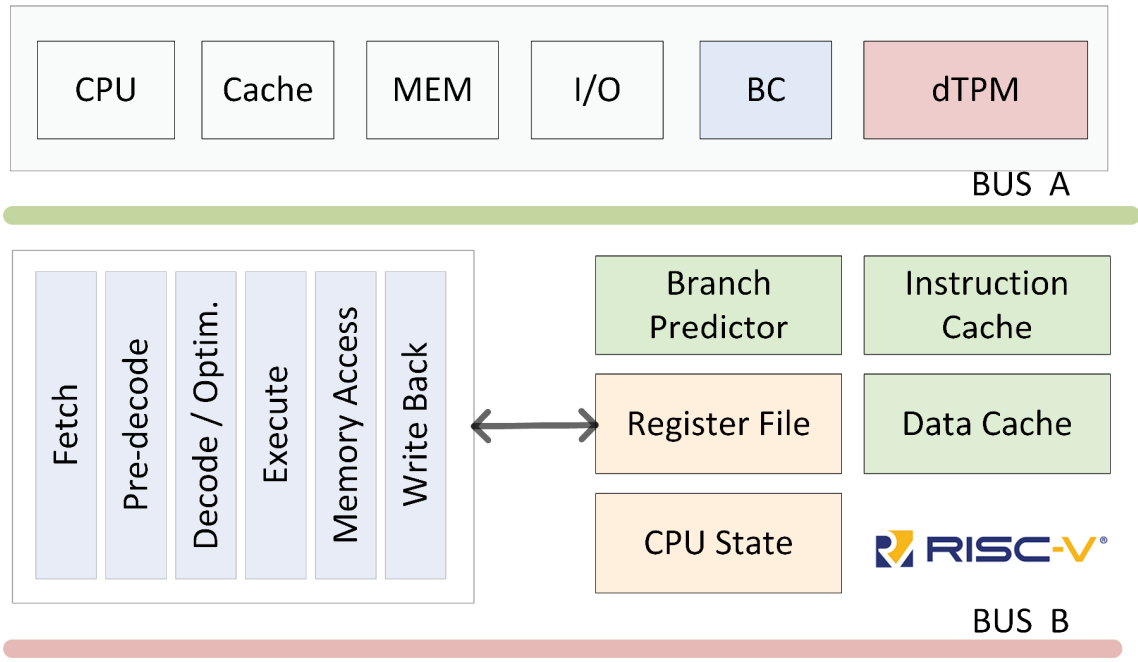
این تحقیق برای طراحی و تقلید از یک معماری ISA جامع قابل اعتماد TRISA[۲۰] که می‌تواند نسبت به حملات Spectre و Meltdown ایمن باشد، از یک مدل باس دوتایی [۲۲] ، ماژول مجزای تعیین کننده اعتبار[۲۳]، معماری RISC-V و شبیه ساز سخت افزار و نرم افزار QEMU-CHERI[۲۱] استفاده می‌کند. شکل زیر معماری ISA مورد اعتماد جدیدی را نشان می‌دهد. در ISA پیشنهادی دو باس یا گذرگاه وجود دارد. باس A (گذرگاه سبز)، باس داخلی بین منطقه سبز (بالا) و منطقه حفاظت‌شده DMZ[۲۲] (پایین) است. باس B، بین DMZ و اینترنت حرکت می‌کند. DMZ (منطقه قرمز) از معماری RISC-V استفاده می‌کند. اعتبار سنجی و مبادله کلید رمزنگاری، توسط یک dTPM انجام می‌شود که حاوی یک موتور رمزنگاری بر روی تراشه برای اهداف امنیت داده است.

[19] discrete Trusted Platform Module

[20] Trusted ISA

[21] CHERI-QEMU is an adaptation of the popular QEMU ISA emulator to implement the CHERI-MIPS and CHERI-RISC-V instruction sets.

[22] Demilitarized Zone



پس معماری TRISA، از EQMU-CHERI در بالای معماری RISC-V ISA، یک dTPM جداگانه ، جریان داده دو گذرگاهی و ویژگی اشکال زدایی برای جلوگیری از حملات کانال جانبی اعمال می‌کند. این مدل محافظتی شامل موارد زیر است:

- اصل حداقل بودن اختیارات : تنظیم مرزها، مجوزها و خط مشی‌های کنترل دسترسی.
- محافظت شدید از اشاره‌گرها: ساده‌ترین معماری‌ها برای حمایت از حفاظت از نشانگرهای زبان C و C ++.
- استفاده از مبانی محاسبه مطمئن (TCB[۲۳]): محافظت از سرریز بافر و نشت.
- محافظت از یکپارچگی: استفاده از هش و کنترل مجزا برای تضمین یکپارچگی
- نفوذناپذیر سازی در صلاحیت‌ها: کنترل استثنا بهتر برای موارد تغییر و پرش.
- آنتروپی کامل فوری: استفاده از RNG Quantum برای ارائه بالاترین امنیت در محاسبات رمزنگاری

[23] *Trusted Computing Bases*

کاهش حملات Spectre و Meltdown می‌تواند از طریق موارد زیر انجام شود:

• اجتناب از انشعاب یا پرش در حرکت‌های شرطی.

• استفاده از موانع حدسی یا قیاسی برای محدود کردن زمان انجام.

• خودداری از استفاده یا تخلیه‌های آشکار داده‌ها در سطح ریز معماری

• استفاده از فضاهای مختلف آدرس هسته اصلی (kernel) برای کاربران مختلف

• کنترل یکپارچگی در سطح معماری.

این تغییرات و کاهش‌های معماری باید در سطح سخت افزاری اتفاق بیافتد. در نظر گرفتن این فاکتورها در حین فرآیند طراحی و شبیه سازی تقلیدی ISA ضروری است، بنابراین پردازنده‌های جدید می‌توانند افزایش عملکرد سرعت خط لوله را حفظ کنند در حالی که اصل مجازی سازی – مجزا بودن نواحی کاری آن‌ها را تضمین می‌کند.

روش ارائه شده به چه صورت پیاده سازی شده؟(نرم افزاری یا بصورت اثبات ریاضی دقیقا توضیح داده شود)

ابتدا توسط امولاتور QEME شبیه سازی شده سپس با زبان chisel سخت افزار مورد نظر که بلوک fpga روی برد HiFive Unleashed توسعه دهنده پردازنده RICS-V پیاده‌سازی شده است.

QEMU بهترین انتخاب برای شبیه سازی پردازنده‌های مختلف است و بهترین عملکرد را در یک محیط مجازی فراهم می‌کند. دستورالعمل های RISC با توانمندی ارتقا یافته سخت افزاری (CHERI)، ISA های معمولی را با ویژگی جدیدی برای محافظت از حافظه ریز و تقسیم بندی نرم افزار گسترش و پیشرفت می‌دهد [۲۰]. استفاده از QEMU-CHERI در RISC-V، امنیت سطح سخت افزاری را فراهم می‌کند که می‌تواند برای پردازنده‌های واقعی تقلید شود [۲۱].

Chisel[۲۴] یک زبان طراحی سخت افزار با ویژگی‌های پیشرفته تولید مدار است. علاوه بر این، می‌تواند برای طراحی هر دو طرح منطق دیجیتالی ASIC و FPGA استفاده شود. این پیش‌نیازهای سخت افزاری، قدرت برنامه نویسی مدرن را در اختیار برنامه نویسان قرار می‌دهند تا بتوانند مدارهای پیچیده و قابل پارامتری شدن را برای سطح بهتر انتزاع و امنیت برنامه ریزی کنند.

.

نحوه مقایسه ایده مطرح شده با دیگر ایده‌های مطرح شده در مقاله:

با توجه به دامنه خسارات وارده و گستردگی حملات Spectre و Meltdown اقدامات زیادی نیز بخصوص از جانب تولید کنندگان پردازنده ها برای مقابله با آن صورت گرفته. لذا اقدامات شاخص معرفی شده و درباب مقایسه به تفاوت تکنیک‌ها ابزار و زیرساخت این روش پرداخته شده است. مثلا قابلیت‌هایی که دستورالعمل‌های کاهش یافته و پردازنده‌ها و شبیه‌سازهای متن باز و ... ایجاد می‌کنند.

بسیاری از محققان موافق هستند که RISC بهترین ISA با هدف کاربرد عمومی است. به عنوان یک ISAای منبع باز، خطر انحصار را کاهش می‌دهد، زیرا تحت مالکیت یک شرکت نیست، که تحت آسیب پذیری‌های درهای پشتی یا بکدورها قرار گیرد. این همچنین یک مدل خوب برای کشورهای در حال توسعه است که از ابزارهای توسعه معماری خود برای محاسبات با اطمینان بالا استفاده می کنند.

در زیر برخی از مشخصات اصلی RISC-V آورده شده است:

• سادگی: کتابچه راهنمای RISC-V ISA دارای ۷۶۷۰۲ کلمه (۲۳۶ صفحه) است در حالی که کتابچه x86-32 شامل بیش از دو میلیون کلمه (۲۱۹۸ صفحه) است.

• مدولار بودن: هسته RISC-V، مجموعه کاملی از نرم افزار را از کامپایلرهای سیستم عامل تا اشکال زداها را اجرا می‌کند. این امر، دستورالعمل‌های فلاپس (عملیات ممیّز شناور در ثانیه) ضرب و تقسیم را با دقت دو برابر امکان پذیر می‌کند.

[24] The constructing hardware in a Scala Embedded Language

• بازدهی: RISC-V، امکان پیاده سازی‌های کم مصرف را فراهم می‌کند، که برای برنامه‌های اینترنت اشیا ایده‌آل است و همچنین از برنامه یا کاربردهای پیشرفته پشتیبانی می‌کند.

• رزرو فضای کددستور یا آپکد: رزرو فضای آپکد، RISC-V را قادر می‌سازد تا بین هسته‌های با هدف کاربرد عمومی و هسته‌های خاص منظوره DSA جفت یا کوپل شود. این کار همچنین فضای آپکد را برای شتاب دهنده‌های سفارشی ایجاد می‌کند.

• امنیت بهتر: پیاده سازی‌های مبتنی بر منبع باز را می‌توان به صورت منبع باز (برای مسائل مالی)، یا برای برخی از نیازهای امنیتی، منبع بسته کرد. این امر پتانسیل کد یا مدارهای مخرب در پردازنده‌های موجود را از بین می‌برد و اجازه می‌دهد تا در سیستم‌ها و برنامه‌های با اطمینان بالا مورد استفاده قرار گیرد.

QEMU می‌تواند از طریق ترجمه باینری پویا یا دینامیکی، از پردازنده‌ها تقلید کند. این یک سیستم شبیه سازی یا تقلیدی سیستم کامل است که می‌تواند سیستم‌های عامل را برای هر ماشین و روی هر نرم افزار پشتیبانی شده‌ای اجرا کند. این امر به ویژه در طراحی و توسعه پردازنده‌های جدید که دارای مکانیسم‌های امنیتی بهتر برای جلوگیری از جاسازی کد و بمب‌های منطقی (logic bombs) بالقوه همراه پردازنده‌های موجود هستند، بسیار مفید است.

در زیر برخی از ویژگی‌های QEMU آورده شده است:

• تقلید کامل از ISA

• شبیه سازی مدل کاربر

• اجرای ماشین های مجازی همراه با عملکرد تقریباً طبیعی

• شبیه سازی شبکه

• جداسازی مهمان [کاربر فاقد مجوز] و امنیت TLS

نقاط قوت و ضعف مقاله:

مقاله حاضر، ایده بسیار خوب و عالی با دلایل محکم و قوی ارائه نموده ولی عملا هیچگونه پیاده سازی انجام نشده. نه کدی در اختیار قرار داده شده و شکل و نمودار و نتایج آزمایش در برابر حمله و ...

تمام تمرکز مقاله و ایده مطرح شده بر روی استفاده از تکنیک‌های نرم افزاری و سخت افزاری متن باز برای حذف نقش زنجیره تأمین کنندگان و مالکین است درحالیکه خودش هیچ جزییاتی در اختیار قرار نداده. مثلا اصلاحات در چگونگی آدرس دهی، واکشی و رمزگشایی بخش آپکد و ...

جمع بندی و پیشنهادات برای کارهای آتی:

نویسندگان مقاله عقیده دارند که شبیه سازی ۶۴ بیتی RISC-V TRISA نشان می‌دهد که از طریق بازنگری اساسی در سطح ISA با ملاحظاتی برای افزایش سرعت سخت افزار، امنیت با کمک حفاظت از حافظه، کنترل بهتر پیش بینی انشعاب یا پیش بینی کننده پرش، جلوگیری از خالی کردن حافظه ریز معماری ایمن، افزایش کنترل یکپارچگی، موتور رمزنگاری خارجی و استفاده از RISC-V TCB، بهبود می‌یابد.

رویکرد جامع برای افزایش امنیت در سطح ISA یک گام اساسی برای مقابله با حملات Spectre و Meltdown است که در غیر این صورت با استفاده از وصله‌های نرم افزاری در سطح کنونی امکان پذیر نخواهد بود. این مطالعه، سیستم‌های رایانه ای مورد اعتماد را برای محاسبات با اطمینان بالا برای نسل بعدی آن به ارمغان می‌آورد که نه تنها پیامدهایی در علم محاسبات و رایانش دارد بلکه منافع زیادی نیز در امنیت ملی دارد.

شبیه سازی:

ارائه و شبیه سازی آنچه در مقاله شبیه سازی یا پیاده سازی شده است:

شبیه سازی تقلیدی TRISA روی QEMU :

TRISA در سیستم ۶۴ بیتی لینوکس (اوبونتو ۱۸,۱۰)، آزمایش می‌شود. پس از ساخت QEMU با هدف RISC-V، سیستم آماده اجرای RISC-V Linux در یک برد توسعه یافته است یعنی SiFive HiFive Unleashed یا یک برد Avalanche همراه با Microsemi PolarFire FPGA [۲۵].

برد HiFive دارای ویژگی‌های است که باعث شده در تمام دنیا برای اهداف توسعه‌ای بکار برده شود. این برد شامل:

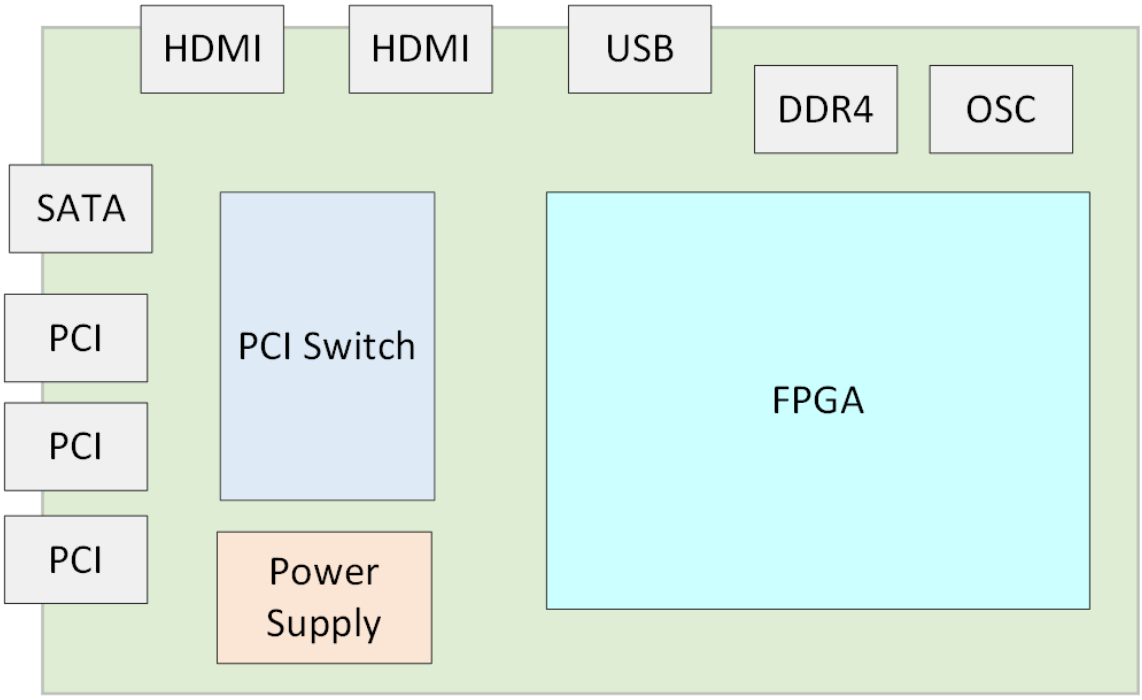
- یک PolarFire FPGA کم مصرف
- سوئیچ ۲۴ اسلات PCIe و اتصال دهنده کارت Express PCI

• اتصالات SS و SATA

• ۱۶ حافظه ۴ گیگابایتی DDR4 و حافظه فلش

• نرم افزار طراحی FPGA

شکل زیر بلوکی مربوط به برد Microsemi HiFive unleashed expansion را نشان می‌دهد [۲۶].
این برنامه برای ISA مورد آزمایش برنامه ریزی شده است



[ب۱] هومند، بیژن، آشنایی با ماشین‌های مجازی، ماهنامه الکترونیکی ایران تاکس، مرداد ۱۳۸۵؛ ۲۲:۵