Pipeline.

Las Pipeline de las CPU modernas son básicamente canalizaciones de ejecución superescalar fuera de orden, lo que puede mejorar considerablemente el rendimiento de la canalización. Por ejemplo, una instrucción no relacionada podría ejecutarse fuera de orden porque la instrucción anterior podría necesitar esperar para leer la memoria, lo que requiere muchos ciclos de instrucción. [1]

Un motor de ejecución fuera de orden consta de varias partes: renombramiento de registros en orden, ejecución fuera de orden de instrucciones o microoperaciones, y retiro en orden. Para facilitar la ejecución fuera de orden y el envío secuencial de microoperaciones, el motor de ejecución fuera de orden incluye unidades de hardware como el búfer de reorden (ROB), el búfer de carga, el búfer de almacenamiento y la estación de reserva (RS). Cuando el motor de ejecución fuera de orden recibe microoperaciones del frontend, asigna recursos a estas microoperaciones. El orden de programación original se conserva en el búfer de reorden (ROB) para retirar las microoperaciones en orden.[1]

El predecodificador extrae bloques de 16 bytes de la caché de instrucciones y detecta el inicio de cada instrucción. Las instrucciones predecodificadas se insertan en la Instruction Queue (IQ). El decodificador puede decodificar hasta cuatro instrucciones por ciclo desde la IQ, usando un decodificador complejo y tres simples. Instrucciones con más de cuatro µops son manejadas por el Microcode Sequencer (MS). Los µops decodificados también se almacenan en el Decoded Stream Buffer (DSB) para aumentar el throughput en loops donde la decodificación es un cuello de botella. [2]

Predicción de saltos

Los procesadores modernos introducen técnicas de predicción de saltos para resolver los bloqueos en el pipeline debido a los peligros de control de flujo y mejorar el rendimiento de la ejecución de instrucciones. Sin embargo, debido a las características compartidas de las unidades predictoras de saltos y a los tiempos de ejecución retardados o estados microarquitectónicos que no se pueden revertir en caso de predicción incorrecta, los predictores de saltos están expuestos a amenazas significativas de ataques microarquitectónicos bien diseñados, que pueden ser lanzados desde software no privilegiado, resultando en consecuencias de seguridad más graves que los ataques de canal lateral tradicionales, como la filtración de claves criptográficas y la exfiltración de datos de enclaves SGX. [3]

Los productores a menudo necesitan comunicar cambios en el flujo de control o condiciones excepcionales a los consumidores. Hacer esto mediante operaciones normales de encolado y desencolado sería ineficiente. En su lugar, Pipette proporciona valores de control (CVs). Los valores de control son similares a otros valores que se pasan a través de colas, excepto que transmiten cambios en el flujo de control en lugar de datos de aplicación. Antes de iniciar la ejecución, cada hilo registra un manejador de control de desencolado, similar a un manejador de excepciones. Un hilo que desencola o consulta una cola con un valor de control en la cabeza salta al manejador de control de desencolado (este salto ocurre completamente a nivel

de usuario y no involucra al sistema operativo). El manejador recibe el valor de control y el ID de la cola que lo disparó, procesa el valor de control y luego vuelve al código principal de Pipette para continuar la computación. [4]

Ejecución especulativa

La tecnología de ejecución especulativa reduce el bloqueo de la canalización al predecir la dirección de destino del salto de rama. Si la ejecución es incorrecta, se realiza una reversión de la canalización y se cancela la ejecución especulativa. Sin embargo, el contenido de la caché no se revierte.[1]

Las predicciones de saltos precisas pueden ocultar una gran fracción de los stalls del frontend debido a la naturaleza desacoplada de los frontends de procesadores modernos. FDIP evita el acoplamiento entre la predicción de saltos y la recuperación de instrucciones, permitiendo que la prefetching guiada por predicción de ramas reduzca stalls del frontend. En las aplicaciones de centros de datos, eliminar mispredicciones de ramas no solo reduce los stalls del pipeline, sino que también disminuye los misses en la I-cache a través de FDIP, mostrando que la predicción de saltos es crítica para la eficiencia del pipeline y la ejecución especulativa. [5]

Los procesadores modernos incorporan técnicas de ejecución especulativa para prevenir bloqueos en el pipeline. Este proceso implica predecir la siguiente instrucción a ejecutar basándose en el historial de ejecuciones previas, lo que permite que las instrucciones potenciales se ejecuten con anticipación. Sin embargo, esta técnica de optimización tiene serias fallas de diseño. Las predicciones incorrectas resultan en que las instrucciones especulativas y sus implicaciones arquitectónicas sean descartadas. No obstante, rastros de estas acciones se retienen en el estado microarquitectónico, como en la caché y el TLB, lo que representa riesgos de seguridad. [6]

Bibliografías

[1] Q. Ke, C. Wang, H. Wang, Y. Lyu, Z. Xu, and D. Wang, “Model Checking for Microarchitectural Data Sampling Security,” in *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*, IEEE, Jul. 2022, pp. 56–63. doi: 10.1109/DSC55868.2022.00015.

[2] A. Abel and J. Reineke, “uiCA,” in *Proceedings of the 36th ACM International Conference on Supercomputing*, New York, NY, USA: ACM, Jun. 2022, pp. 1–14. doi: 10.1145/3524059.3532396.

[3] Q. Wang, M. Tang, K. Xu, and H. Wang, “Unveiling and Evaluating Vulnerabilities in Branch Predictors via a Three-Step Modeling Methodology,” *ACM Transactions on Architecture and Code Optimization*, vol. 22, no. 1, pp. 1–26, Mar. 2025, doi: 10.1145/3711923.

[4] Q. M. Nguyen and D. Sanchez, “Pipette: Improving Core Utilization on Irregular Applications through Intra-Core Pipeline Parallelism,” in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, IEEE, Oct. 2020, pp. 596–608. doi: 10.1109/MICRO50266.2020.00056.

[5] T. A. Khan *et al.*, “Whisper: Profile-Guided Branch Misprediction Elimination for Data Center Applications,” in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, IEEE, Oct. 2022, pp. 19–34. doi: 10.1109/MICRO56248.2022.00017.

[6] H. Jang, T. Kim, and Y. Shin, “SysBumps: Exploiting Speculative Execution in System Calls for Breaking KASLR in macOS for Apple Silicon,” in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA: ACM, Dec. 2024, pp. 64–78. doi: 10.1145/3658644.3690189.