



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

درس فناوری های حافظه
دکتر حامد فربه

رضا آدینه پور
۴۰۲۱۳۱۰۵۵

تمرین شبیه سازی چهارم

تدریسار:

مرتضی عادلخانی (madelkhani@aut.ac.ir)
سارا زمانی (sara.zamani۷۳@aut.ac.ir)

۱. به سوالات زیر پاسخ دهید:

۱. PUM چیست و کدام نوع حافظه ها برای آن بیشتر استفاده می شوند؟ توضیح دهید چرا هر نوع حافظه استفاده می شود.

پاسخ: پردازش در حافظه (PUM) یک کانسپت محاسباتی است که در آن برخی از محاسبات ساده مانند جمع و ضرب به جای انتقال داده ها بین CPU و حافظه، مستقیماً در حافظه انجام می شوند. معمولاً از SRAM، DRAM و NVM ها در PUM استفاده می شود. که در ادامه به بررسی مزایا و معایب استفاده از هر کدام می پردازیم.

DRAM ها به دلیل اینکه رایج ترین نوع حافظه فرار با تراکم بالا و هزینه کم به ازای هر بیت هستند، به طور گسترده استفاده می شود. ویژگی های خازنی سلول های DRAM امکان انجام تکنیک های محاسباتی درون حافظه مانند عملیات منطقی و حسابی را فراهم می کند. مزایا: تراکم بالا، ارزان است.

معایب: فرار، نیاز به تازه سازی دوره ای و معمولاً تأخیر بیشتر نسبت به SRAM. اما در مقابل SRAM زمان های تأخیر کمتری و زمان دسترسی سریعتری نسبت به DRAM دارد و در مواردی که سرعت برای ما بسیار مهم است، (مانند Cache)، استفاده می شود. توانایی حفظ حالت بدون نیاز به تازه سازی، آن را برای عملیات های PUM مناسب می سازد.

مزایا: زمان های دسترسی سریع نسبت به DRAM، عدم نیاز به تازه سازی. معایب: تراکم کمتر و هزینه بیشتر به ازای هر بیت نسبت به DRAM.

در مقابل حافظه های فرار، انواع حافظه های غیر فرار مانند Flash، PCM و ReRAM به دلیل نگه داشتن داده بدون برق، برای ذخیره سازی پایدار و محاسبات مناسب هستند. این حافظه ها می توانند برخی عملیات منطقی را درون سلول های حافظه انجام دهند.

مزایا: غیر فرار بودن. معایب: عموماً سرعت نوشتن کندتر و دوام کمتر نسبت به DRAM و SRAM.

۲. نقاط ضعف UPMEM چیست؟

پاسخ: از نقاط ضعف UPMEM ها می توان به موارد زیر اشاره کرد:

(آ) انعطاف پذیری و قابلیت برنامه ریزی محدود: معماری PIM UPMEM برای انواع خاصی از عملیات (مانند وظایف داده محور مانند جست و جو در پایگاه داده و تحلیل) است. ممکن است به اندازه CPU یا های GPU سنتی عمومی و همه منظوره نباشد، که کاربرد آن را به بارهای کاری خاص محدود می کند.

(ب) یکپارچه سازی و سازگاری: یکپارچه سازی ماژول های PIM UPMEM با سیستم های موجود می تواند چالش برانگیز باشد. ممکن است مشکلات سازگاری با معماری های حافظه و پردازنده فعلی به وجود بیاید که نیاز به اصلاحات در کانفیگ نرم افزار و سخت افزار دارد.

(ج) مسائل مربوط به کارایی انرژی: در حالی که PIM هدفش کاهش مصرف انرژی با حداقل کردن حرکت داده ها بین حافظه و CPU است، صرفه جویی واقعی در انرژی می تواند وابسته به بار کاری باشد. برخی عملیات ممکن است همچنان مصرف انرژی قابل توجهی داشته باشند، به خصوص اگر منطق PIM به طور کامل برای آن کاربرد به خصوص بهینه سازی نشده باشد.

(د) توسعه و اشکال زدایی: همانطور که در کلاس هم بررسی شد، توسعه برنامه ها برای PIM نیاز به مدل های برنامه نویسی و ابزارهای جدید دارد. دیباگ و پروفایل کردن برنامه های PIM می تواند به دلیل طبیعت توزیع شده و درون حافظه ای محاسبات سخت تر از برنامه نویسی CPU/GPU سنتی باشد.

۳. ساختار Ambit را معرفی کرده و مزایا و معایب آن را توضیح دهید.

پاسخ: Ambit یک معماری PIM است که از بستر DRAM موجود برای انجام عملیات بیتی (مانند AND، OR، NOT) به طور مستقیم درون حافظه استفاده می‌کند. این معماری از ویژگی‌های آنالوگ سلول‌های DRAM و Sense Amplifier برای اجرای این عملیات استفاده می‌کند. از مزایای آن می‌توان به موارد زیر اشاره کرد:

(آ) کاهش حرکت داده‌ها:

با انجام محاسبات مستقیماً درون DRAM به طور قابل توجهی نیاز به حرکت داده بین CPU و حافظه را کاهش می‌دهد، که منجر به کاهش تاخیر و مصرف انرژی می‌شود.

(ب) توان محاسباتی بالا:

Ambit می‌تواند عملیات بیتی را بر روی حجم زیادی از داده‌ها به طور همزمان انجام دهد، که توان محاسباتی بالایی برای وظایف داده‌محور مانند جستجوهای پایگاه داده، رمزنگاری و شبکه‌های عصبی فراهم می‌کند.

(ج) تغییرات سخت‌افزاری حداقلی:

Ambit از زیرساخت DRAM موجود با تغییرات حداقلی استفاده می‌کند، که ادغام آن را با سیستم‌های فعلی نسبت به معماری‌های PIM آسان‌تر می‌کند.

همچنین از معایب Ambit می‌توان به موارد زیر اشاره کرد:

(آ) محدودیت در انواع عملیات:

Ambit به طور عمده برای عملیات بیتی طراحی شده است. نمی‌تواند به طور کارآمد محاسبات پیچیده‌تر ریاضی یا اعشاری را انجام دهد، که کاربرد آن را به انواع خاصی از عملیات محدود می‌کند.

(ب) پیچیدگی در برنامه‌نویسی:

برنامه‌نویسی برای Ambit نیاز به درک مدل عملیاتی خاص و محدودیت‌های آن دارد. برنامه‌نویس‌ها باید الگوریتم‌های خود را برای استفاده موثر از عملیات بیتی تطبیق دهند، که می‌تواند پیچیدگی نرم‌افزاری را افزایش دهد.

(ج) چالش‌های مقیاس‌پذیری:

در حالی که Ambit توان محاسباتی بالایی برای عملیات بیتی ارائه می‌دهد، مقیاس‌پذیری آن به سیستم‌های حافظه بزرگ‌تر یا ادغام آن با واحدهای پردازشی دیگر ممکن است چالش‌هایی از نظر هماهنگی و مدیریت ایجاد کند.