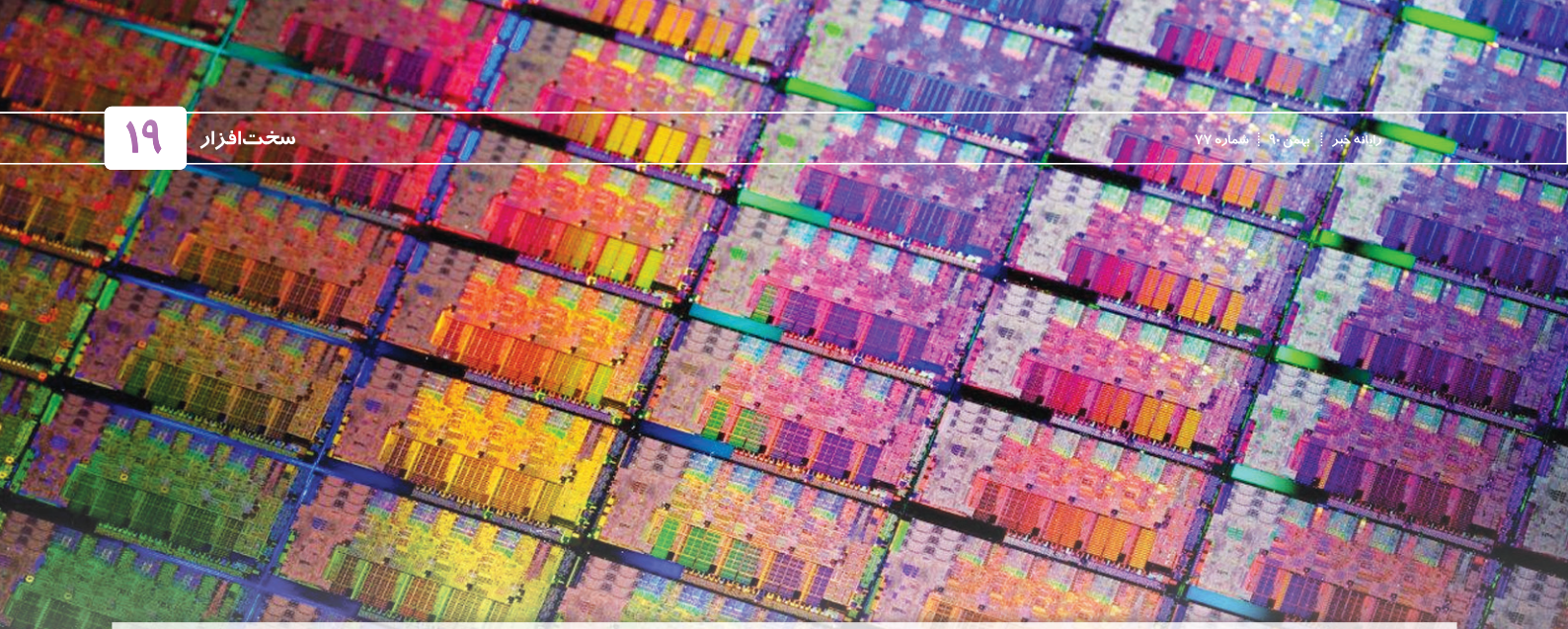


• این مقاله هم برای افراد حرفه‌ای و هم افراد مبتدی پیشنهاد می‌شود

کالبدشکافی CPU

هومن سیاری
Sayyari@ComputerNews.ir

پردازنده یک چیپ بسیار پیچیده است. حتی در پردازنده اتم شرکت اینتل که بسیار سبک و کوچک است هم بیش از ۴۰ میلیون ترانزیستور روی یک قطعه سیلیکونی در اندازه ناخن گنجانده شده است. جالب آنکه این تعداد بالای ترانزیستور همه چیز نیست، بلکه بخش مهم و پیچیده دیگر آن فرآیند تولید این چیپ‌ها است، جایی که طی یک پروسه مهندسی بسیار پیچیده‌تر این ترانزیستورها در کنار هم به عنوان یک هسته پردازشی قرار می‌گیرند تا پردازنده کامپیوترهای رومیزی، نوت‌بوک‌ها، تبلت‌ها و ... را بسازند.

هر چند ساخت فیزیکی پردازنده‌ها بسیار پیچیده است، اما فهم اصول کارکرد پردازنده‌های مدرن و اینکه آنها چگونه کار می‌کنند، چندان دشوار نیست. خیلی خوب است که دانشی هر چند مختصر از نحوه کارکرد پردازنده داشته باشیم. زمانی که قصد خرید یا ارتقای کامپیوتر خود را داشته باشید، این دانش به کار می‌آید تا بتوانید پردازنده مناسب کار خود را انتخاب نمایید.

در عین حال این دانش می‌تواند به درد عیب‌یابی سیستم بخورد. اینکه بدانیم دستورها چگونه درون پردازنده پردازش می‌شوند، می‌تواند به برنامه‌نویسان کمک کند تا بتوانند برنامه‌هایی بنویسند که با سرعت و راندمانی به مراتب بالاتر اجرا شوند، هر چند امروزه بهینه‌سازی پردازنده‌ها اغلب به صورت خودکار توسط کامپایلر انجام می‌شود و دیگر چندان نیازی به دخالت برنامه‌نویس نیست.

ساختار فیزیکی

وقتی به پردازنده می‌اندیشید، احتمالا یک چیپ مربع‌شکل کوچک با رویه فلزی به ذهن‌تان خطور می‌کند. این در واقع بسته‌بندی پردازنده است. این رویه قابل برداشتن نیست اما اگر دقت بیشتری داشته باشید، مربعی کوچک‌تر از جنس سیلیکون در زیر آن خواهید یافت. به این مربع کوچک‌تر die می‌گویند که ویژگی‌های عملیاتی یک پردازنده در آن جای داده شده است.

این ویژگی‌ها به شکل میلیون‌ها ترانزیستوری که روی سیلیکون حکاکی شده‌اند، عملیاتی می‌شوند. به فرایند حکاکی ترانزیستورها بر روی سیلیکون، اصطلاحا فوتولیتوگرافی می‌گویند. فوتولیتوگرافی در اصل تعمیم‌یافته عکاسی است. ابتدا چیزی شبیه نگاتیو عکاسی از شِمای مدار مجتمع تهیه می‌شود. این نگاتیو که در این جا «ماسک» نامیده می‌شود، برای تکثیر طرح بر روی هادی‌ها و نیمه‌هادی‌ها به کار گرفته می‌شود. تهیه نگاتیو به سادگی عکاسی نیست، اما با داشتن آن می‌توان به راحتی هزاران نسخه تکثیر کرد. بنابراین، روند کار به دو بخش اصلی تقسیم می‌شود: اول تهیه ماسک که می‌تواند زمان‌بر و پرهزینه باشد و دوم استفاده از ماسک برای تهیه نسخه‌های بعدی که باید سریع و ارزان باشد. برای تولید ماسک، ابتدا شِمای مدار در مقیاس به نسبت بزرگ طراحی می‌شود. سپس این طرح به صورت لایه نازکی از فلز (اغلب کروم) روی صفحه شفافی (اغلب شیشه یا سیلیکون) درمی‌آید که در مجموع به آن «ویفر» گفته می‌شود.

سپس فوتولیتوگرافی، در فرآیندی شبیه آنچه در تاریکخانه عکاسی اتفاق می‌افتد، ابعاد طرح را کوچک می‌کند. برای این کار یک دسته پرتو نور (اغلب نور فرابنفش لامپی جیوه‌ای) از ماسک عبور می‌کند و با استفاده از یک عدسی، تصویری روی سطح سیلیکون تشکیل می‌دهد. روی سیلیکون با لایه‌ای از جنس پلیمرهای آلی حساس به نور (فُتورِزیست) پوشانده شده است. قسمت‌هایی که نور دیده‌اند، در فرآیند تثبیت حذف شده و طرحی معادل طرح اولیه روی سطح سیلیکون پدیدار می‌شود. با قرار دادن چندین لایه فوتولیتوگرافی در کنار یکدیگر، امکان ساخت مدارات الکترونیکی سه‌بعدی در مقیاسی بسیار کوچک میسر می‌شود.

اندازه دقیق این ترانزیستورها معمولا در مقیاس نانومتر بیان می‌شود که کوچک‌ترین عنصری که توسط این فرآیند ساخت قابل تولید است را مشخص می‌کند. تولیدکنندگان چیپ دایم در حال کوچک کردن فرآیند ساخت هستند، به طور مثال اینتل پردازنده 80386 را با فرآیند ساخت ۱٫۵ میکرومتر تولید کرد، سپس پردازنده پنتیوم ۴ را با استفاده از فرآیند ساخت ۱۸۰ نانومتر تولید کرد و هم‌اکنون پردازنده Core i7 را با فرآیند ساخت ۳۲ نانومتری تولید می‌کند.

کوچک کردن فرآیند ساخت این امکان را برای تولیدکنندگان مهیا می‌سازد که بدون افزایش اندازه die چیپ‌هایی به مراتب پیچیده‌تر طراحی کنند. برای مثال پردازنده قدیمی 80386 از ۲۷۵۰۰۰ ترانزیستور بر روی یک die به اندازه ۱۰۴ میلی‌متر مربع ساخته شده بود، اما با پیشرفت فناوری فشرده‌سازی

فرایند ساخت، امروزه پردازنده Core i7-2600 با ۹۹۵ میلیون ترانزیستور بر روی یک die به اندازه ۲۱۶ میلی‌متر مربع ساخته می‌شود. یعنی در حالی که ابعاد die فقط ۴ برابر شده، تعداد ترانزیستورها بیش از ۳۶۰۰ برابر شده است!

کوچک شدن die و افزایش تعداد ترانزیستورها چند مزیت مهم دارد: اول آنکه تولید سیلیکون با درجه خلوص بالا بسیار گران‌قیمت است. یک die کوچک‌تر به معنای کاهش هزینه مواد اولیه است که این خود منجر به فروش ارزان‌تر چیپ می‌گردد. در عین حال سود تولیدکنندگان هم حفظ می‌شود.

از طرف دیگر وقتی ترانزیستور کوچک می‌شود، به طور طبیعی توان کمتری مصرف خواهد کرد و انرژی کمتری را به صورت حرارت تلف خواهد کرد. معنای این عبارت آن است که ترانزیستورهای کوچک‌تر می‌توانند در فرکانس بالاتری کار کنند بدون اینکه با مشکل افزایش بیش از حد حرارت مواجه شوند. از نگاه دیگر، چون انرژی کمتری مصرف می‌کنند، برای محیط زیست مناسب‌ترند و مزیت دیگر کم‌مصرف بودن، طول عمر بیشتر باتری در تجهیزات سیار مثل نوت‌بوک‌ها و تبلت‌ها است.

صرف‌نظر از فناوری ساخت، همه چیپ‌ها مقداری گرما تولید می‌کنند. این گرما توسط لایه فلزی روی چیپ‌ها به خارج منتقل می‌شود تا توسط یک فن یا کولر خارجی کاملا دفع گردد. حداکثر مقدار گرمایی که یک چیپ قادر به تحمل آن هست را با عنوان TDP نشان می‌دهند و با واحد وات بیان می‌شود. البته TDP میزان مصرف یک پردازنده را در شرایط نرمال مشخص نمی‌کند، اما به طور طبیعی می‌توان انتظار داشت که یک پردازنده با TDP پایین ۳۵ وات مصرف کمتری از یک پردازنده با TDP بالاتر ۶۰ وات داشته باشد.

محیط پیرامونی CPU

CPU در یک سوکت اختصاصی روی مادربورد قرار می‌گیرد و از آن طریق با سایر قطعات سیستم ارتباط برقرار می‌کند. تا همین چند سال پیش ارتباط CPU با دیگر قطعات از طریق دو چیپ میانی بر روی مادربورد انجام می‌شد. این دو چیپ پل شمالی (north bridge) و پل جنوبی (south bridge) نام دارند. پل شمالی با بخش‌های پرسرعت مثل PCI-Express و گذرگاه‌های حافظه سر و کار دارد و پل جنوبی هم با بخش‌های کم سرعت‌تر مثل کارت صدا و هارددیسک ارتباط دارد. این دو چیپ را مجموعا «چیپست» می‌نامند.

در طراحی‌های مدرن، کنترلر حافظه و کنترلر PCI-Express به داخل CPU منتقل شده‌اند که این باعث کارایی بیشتر و سرعت بالاتر می‌گردد. مادربوردها هنوز هم با چیپست عرضه می‌شوند، اما اغلب آنها فقط پل جنوبی را دارند، چرا که بسیاری از کارکردهای پل شمالی داخل پردازنده جای داده شده است. این چیپست‌ها هم قابلیت‌های متفاوتی دارند، مثلا چیپست اینتل H67 از چیپ گرافیکی داخل پردازنده‌های Core i3 و Core i5 و Core i7 پشتیبانی می‌کند، اما چیپست اینتل P67 از این چیپ گرافیکی پشتیبانی نمی‌کند ولی در عوض دارای قابلیت اورکلاکینگ است. از چیپست‌های AMD هم می‌توان به چیپست 870 آن اشاره کرد که تنها از یک اسلات PCI-Express x16 پشتیبانی می‌کند، در حالی که چیپست 890FX آن می‌تواند تا ۴ اسلات PCI-Express x16 را پشتیبانی کند.

پردازنده چه کاری انجام می‌دهد؟

واقعیت آن است که پردازنده پیچیده‌ترین قطعه در مهندسی الکترونیک است. اما واقعا پردازنده چه کاری انجام می‌دهد؟ می‌توان برای شروع پردازنده را یک ماشین حساب ساده فرض کرد که عملیات ساده ریاضی را انجام می‌دهد و حاصل نتایج محاسبات را در رجیسترهایش ذخیره می‌کند. پردازنده می‌تواند محتویات رجیسترهایش را در حافظه سیستم ذخیره کند و یا مقداری را از حافظه سیستم به رجیستر منتقل کند. رجیسترها حافظه‌هایی کوچک در داخل پردازنده هستند که برای نگه‌داری نتایج محاسبات و یا انجام محاسبات میانی استفاده می‌شوند.

با توجه به ساختار ساده بالا می‌توان گفت که برنامه‌نویسی کامپیوتر عبارت است از فرایند آماده‌سازی دستورهای مناسب برای اجرا بر روی داده‌های متناسب با آن دستورها. در اصل این دستورها در زبان اسمبلی انجام می‌شوند. زبان اسمبلی زبانی است که دستورها را به نزدیک‌ترین صورت به زبان واقعی پردازنده نشان می‌دهد. امروزه بیشتر برنامه‌نویسی با زبان‌هایی مثل C++ انجام می‌شود و از کامپایلرها برای ترجمه زبان برنامه‌نویسی به زبان ماشین استفاده می‌شود.

وقتی پردازنده برنامه‌ای را اجرا می‌کند، در واقع دستورالعمل‌های آن برنامه را به ترتیب اجرا می‌نماید. لازم به ذکر است که منظور از برنامه تعدادی دستورالعمل است که پشت سر هم نوشته شده‌اند و از اجرای مرتب آن هدف برنامه تامین می‌شود. هر دستورالعمل برنامه برای اجرا در پردازنده از یک

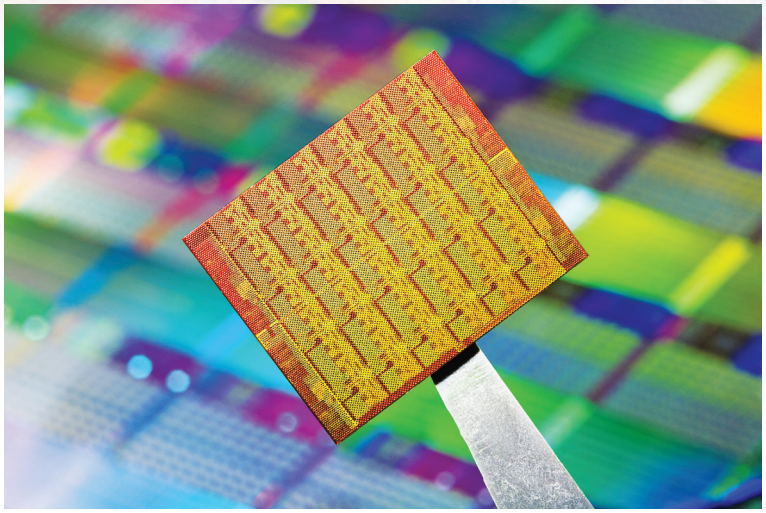
Stepping

انواع متفاوتی از پردازنده‌ها در بازار وجود دارند. بعضی برای سرورها و ایستگاه‌های کاری و بعضی برای نت‌بوک‌ها و تجهیزات کم‌مصرف طراحی شده‌اند. اما پردازنده‌های هم‌نوع بسیاری دیده می‌شود که از یک طراحی هسته یکسان در آنها استفاده شده است. برای مثال پردازنده AMD Athlon II X4 640 و Phenom II X4 940 و Phenom II X4 980 همگی ساختار هسته یکسان دارند، اما از مقدار متفاوت کش L3 و سرعت متفاوت فرکانس برخوردارند. در واقع ساختار چینش ترانزیستورها در همه آنها کاملا یکسان است.

این مسئله دلیل اقتصادی دارد و آن هزینه بالای فرایند فوتولیتوگرافی است. البته این کار ممکن گاهی هم مشکل‌ساز شود، مثلا اگر یک طراحی باگ داشته باشد، این باگ به همه پردازنده‌های آن خانواده منتقل خواهد شد. AMD این مسئله را به خوبی می‌داند، چرا که وقتی اولین پردازنده Phenom خود را قبل از سال ۲۰۰۷ عرضه کرد، همه مدل‌ها از یک باگ در طراحی رنج می‌بردند که موجب می‌شد راندمان پردازنده تا ۱۰ درصد کاهش یابد.

در این شرایط تنها کار برای اینکه تا حدی مشکل را برطرف کرد یا باگ را نشان داد Stepping است. در مورد مشکل فوق، چیپ‌های دارای باگ براساس یک طراحی که دارای Stepping با کد B2 بود ساخته شده بودند. چند ماه بعد وقتی AMD این طراحی را آپدیت کرد، با کد B3 به بازار ارایه شد. معمولا اولین تست طراحی پردازنده با کد A0 نشان داده می‌شود و اگر تغییر قابل توجهی در طراحی پردازنده رخ دهد، حرف انگلیسی اول تغییر می‌کند مثلا A به B و در صورتی که تغییر جزیی باشد شماره آن افزایش می‌یابد مثلا ۱ به ۲.

حتی وقتی مشکلی در ساختار یک چیپ وجود ندارد، بسیار متداول است که Stepping پردازنده در طول عمرش چندین بار تغییر کند. اینتل اولین پردازنده Core i7 را در سال ۲۰۰۸ ارایه کرد که کد C0 داشت، اما اگر یک سال بعد همان پردازنده را می‌خریدید، متوجه می‌شدید که کد آن D0 است. هیچ تغییر مهمی در طراحی آن رخ نداده بود، اما اورکلاکرها اعتقاد داشتند که مدل‌های D0 پایداری بیشتری در سرعت‌های بالاتر دارند. احتمالا اینتل راهی را برای ساده‌تر کردن طراحی آن یافته بود تا بتواند پردازنده را پایدارتر کند و روند کاری را متناسب با آن بهبود ببخشد.



فناوری تک‌چیپ اینتل برای محاسبات ابری

سیکل ۴ مرحله‌ای عبور می‌کند.

۱- Fetch : ابتدا کد دستورالعمل از حافظه خوانده می‌شود.
۲- Decode : سپس این دستورالعمل رمزگشایی می‌شود تا نیازمندی‌های آن برای اجرا تعیین شود. اگر دستورالعمل نیاز به داده‌ای داشته باشد که در حافظه قرار دارد، این کار انجام می‌شود (Read).
۳- Execute : سرانجام دستورالعمل‌ها اجرا می‌شوند.
۴- Store : در پایان نتایج در حافظه یا در رجیسترهای پردازنده ذخیره می‌شود.

سپس اجرا با کد دستورالعمل بعدی ادامه پیدا می‌کند.

اگر فرض کنیم پردازنده مثل یک ساعت عمل می‌کند، هر مرحله از مراحل چهارگانه بالا مثل یک «تیک» است که اصطلاحا به Clock Cycle یا پالس ساعت معروف است. تعداد تیک‌هایی که در هر ثانیه اجرا می‌شوند، نشان‌دهنده فرکانس کاری یا سرعت کلاک پردازنده است. پردازنده‌های متداول فرکانسی حدود ۲ گیگاهرتز برابر با ۲ میلیارد تیک در ثانیه دارند. البته پردازنده‌های مدرن قابلیت تنظیم خودکار فرکانس نسبت به افزایش یا کاهش فشار کاری را دارند. این قابلیت‌ها با نام Speed Step و Turbo boost شناخته می‌شوند. ما فرض می‌کنیم که فرکانس ثابت است و تغییر نمی‌کند.

در همین پردازنده فرضی که یک هسته ۲ گیگاهرتزی دارد، با توجه به اینکه هر دستورالعمل ۴ مرحله دارد، بنابراین ۵۰۰ میلیون دستورالعمل در هر ثانیه اجرا می‌شود (2.000.000.000 / 4 = 500.000.000). البته پردازنده‌های مدرن با توجه به فناوری‌های مدرن به کار رفته در آنها مثل Pipeline می‌توانند دستورهای به مراتب بیشتری را اجرا نمایند.

Pipelining

در پروسه ۴ مرحله‌ای توضیح داده شده در بالا هر مرحله از یک دستورالعمل توسط یک بخش مجزای هسته پردازنده انجام می‌شود. برای مثال وقتی بخش خواندن دستورالعمل (Fetch) اقدام به خواندن یک دستورالعمل کرد و آن را تحویل بخش بعدی داد، کاری برای انجام دادن ندارد تا کار این دستورالعمل تمام شود تا دوباره اقدام به خواندن دستورالعمل بعدی نماید. اما با Pipeline بعد از اینکه بخش خواندن دستورالعمل دستور اول را خواند و به بخش بعدی تحویل داد، بلافاصله اقدام به خواندن دستورالعمل بعدی می‌کند. در حالی که بخش رمزگشایی روی دستورالعمل اول کار می‌کند، بخش Fetch در حال خواندن دستورالعمل دوم است.

بدین ترتیب تمام ۴ قسمت پردازنده به طور همزمان کار می‌کنند و در نتیجه در تئوری می‌توان ادعا کرد که در هر کلاک یک دستورالعمل اجرا می‌گردد، در حالیکه بدون Pipeline در هر ۴ کلاک یک دستورالعمل اجرا می‌شد.

البته در عمل راندمان سیستم به این مقدار نمی‌رسد. ما فرض کردیم که هر مرحله در Pipeline یک کلاک به طول می‌انجامد تا تکمیل شود، اما در عمل در بسیاری از مواقع مرحله Execute به دلیل پیچیدگی دستورالعمل بیش از یک کلاک طول می‌کشد. در پردازنده Sandy Bridge اینتل ضرب ۲ عدد ۳۲ بیتی در یکدیگر ۴ کلاک طول می‌کشد، در حالی که تقسیم آنها ۲۶ کلاک به طول می‌انجامد. وقتی این دستورها در حال انجام هستند Pipeline نمی‌تواند پیش برود و دچار وقفه می‌گردد. بنابراین تعداد دستورالعمل‌های اجراشده در هر کلاک در سیستمی که Pipeline دارد، کاملا وابسته به دستورهایی است که باید اجرا کند.

Pipeline در یک حالت دیگر هم دچار وقفه می‌شود. زمانی که با یک دستورالعمل شرطی (Conditional Branch) مواجه می‌شود، بر سر دوراهی قرار می‌گیرد. اگر شرط درست باشد، باید مثلا دستورالعمل A را اجرا کند و اگر شرط غلط باشد، باید دستورالعمل B را اجرا کند و بنابراین تا تعیین وضعیت شرط نمی‌تواند کاری انجام دهد. این شرط‌ها جزو اصول اولیه هر زبان برنامه‌نویسی محسوب می‌شوند، اما برای Pipeline مرگبارند! در این شرایط واحد خواندن دستورالعمل اقدام به پیش‌بینی می‌کند و یکی از دو دستورالعمل A یا B را برمی‌گزیند و آن را اجرا می‌کند. این پیش‌بینی براساس مداری به نام پیش‌بینی انشعاب یا Branch Predictor انجام می‌شود. این مدار با الگوریتم‌هایی که دارد، سعی می‌کند نتیجه شرط را پیش‌بینی کند و دستورالعمل مناسب را اجرا نماید. یکی از این الگوریتم‌ها استفاده از نتایج شرط‌های مشابه قبلی است. البته همیشه این پیش‌بینی درست از آب در نمی‌آید. اگر پیش‌بینی درست باشد که از وقفه ناشی از Pipeline جلوگیری شده اما اگر پیش‌بینی اشتباه باشد، باید دستورالعملی که به اشتباه انتخاب شده و اجرا شده است، حذف شود و دستورالعمل صحیح مجددا به جای آن خوانده شده و اجرا گردد که این فرآیند وقفه‌ای قابل توجه ایجاد می‌کند.

اگر چه مراحل اجرای یک دستورالعمل را به طور خلاصه در ۴ بخش بیان کردیم، اما بسیاری از پردازنده‌های مدرن اجرای یک دستورالعمل را به قسمت‌های بیشتر و کوچک‌تری تقسیم می‌کنند. به طور مثال پردازنده اینتل Core2Duo از ۱۴ مرحله در Pipeline استفاده می‌کند (به جای ۴ مرحله) در حالی که پردازنده اینتل Pentium D از ۳۱ مرحله در Pipeline استفاده می‌کرد. بسته به طول Pipeline در پردازنده، یک پیش‌بینی اشتباه می‌تواند تعداد قابل توجهی از مراحل Pipeline را خنثی کند. به طور کلی هر چه طول Pipeline بیشتر باشد، در شرایط ایده‌آل یعنی شرایطی که انشعاب یا شرط وجود ندارد راندمان Pipeline بالاتر می‌رود، اما با توجه به اینکه همیشه انشعاب یا شرط وجود دارد، هر چه طول Pipeline بیشتر باشد، در صورت بروز انشعاب یا شرط، وقفه آن بیشتر خواهد بود. بنابراین طول Pipeline استاندارد مشخصی ندارد و هر سازنده و از آن بدتر حتی یک سازنده در محصولات مختلفش از Pipelineهایی با طول‌های متفاوت استفاده می‌کند.

کش و اجرای خارج از نوبت

مشکل دیگری هم در سیستم‌هایی که از Pipeline استفاده می‌کنند وجود دارد. در سیستم مورد توضیح فرض کردیم خواندن دستورالعمل و داده از حافظه می‌تواند در یک کلاک انجام شود. در عمل بسته به سرعت ماژول‌های حافظه و پردازنده خواندن یک مقدار از حافظه و ذخیره در رجیسترهای پردازنده می‌تواند تا ۱۵ کلاک یا بیشتر طول بکشد.

همانطور که مشخص است این موضوع راندمان پردازنده را بسیار پایین می‌آورد، چرا که هر بار که نیاز به داده‌ای از RAM دارد، باید به اندازه ۱۵ کلاک منتظر شود. برای حل این مشکل پردازنده‌های مدرن از ۲ راه‌حل استفاده می‌کنند.

نخست استفاده از کش است. کش یک حافظه کوچک با سرعت بسیار بالاست که درون پردازنده قرار دارد. پردازنده به جای اینکه برای خواندن داده به حافظه رم مراجعه کند و ۱۵ کلاک معطل شود، به کش مراجعه می‌کند و تنها یک کلاک معطل می‌شود. البته بخشی از پردازنده وظیفه به‌روز نگه داشتن محتویات کش را دارد، چون اگر داده مورد نیاز در کش نباشد، باید به رم مراجعه کرد و باز همان ۱۵ کلاک معطلی! وقتی بخشی از پردازنده در حال به‌روزرسانی مداوم محتویات کش از روی رم است، بخش دیگر پردازنده یا واحد اجرایی در حال اجرای دستورالعمل‌هاست.

پردازنده‌های مدرن از یک کش چندلایه استفاده می‌کنند. کوچک‌ترین و سریع‌ترین کش L1 یا سطح اول نام دارد که برای ذخیره دستورالعمل‌ها و داده‌های آنها مورد استفاده قرار می‌گیرد و معمولا ظرفیتی برابر با ۶۴ یا ۱۲۸ کیلوبایت دارد. بعد از آن کش‌های L2 و L3 قرار دارند. کش‌های L1 و L2 و L3 به ترتیب بزرگ‌تر و کندتر می‌شوند، یعنی کش L3 از همه بزرگ‌تر و البته کندتر است. واحد اندازه‌گیری L2 و L3 مگابایت است. کش حافظه‌ای بسیار گران‌قیمت است، چرا که هزینه تولید آن بسیار بالاست. یکی از مواردی که منجر به کاهش قیمت پردازنده می‌شود، کاهش حجم اندازه کش آن است. به همین دلیل تولیدکنندگان پردازنده معمولا پردازنده‌هایی با معماری یکسان را با مقادیر متفاوت کش و البته با قیمت‌هایی متفاوت عرضه می‌کنند.

پردازنده‌ها علاوه بر کش از فناوری اجرای خارج از نوبت دستورها یا OoOE هم استفاده می‌کنند. منظور از این فناوری آن است که لزوما دستورها به صورت پشت سر هم و ترتیبی اجرا نمی‌شوند، بلکه در شرایط خاص ترتیب اجرای آنها تغییر داده می‌شود. مثلا اگر دستورالعمل اول برای دریافت داده‌های مورد نیاز خود از حافظه معطل شود، پردازنده به سراغ دستورالعمل دوم می‌رود تا آن را در این فرصت اجرا کند، در حالی که اگر ترتیب اجرای دستورها اجرا می‌شد، دستورالعمل دوم تا زمان به پایان رسیدن دستورالعمل اول باید منتظر می‌ماند. بعد از اینکه دستورالعمل اول داده‌های مورد نیازش را از حافظه لود کرد، پردازنده به آن برمی‌گردد تا اجرای آن را ادامه دهد.

بدیهی است که اجرای خارج از نوبت دستورها با محدودیت‌هایی مواجه است، چرا که دستورها توسط برنامه‌نویس با ترتیب مشخصی نوشته شده‌اند و احتمالا نتیجه هر دستورالعمل به دستورالعمل بعدی داده می‌شود. اما در اجرای خارج از نوبت دستورات، این شرایط نقض می‌شود و برنامه به درستی کار نخواهد کرد. برای حل این مشکل از یک بافر استفاده می‌شود که دستورها بعد از مرحله Fetch به آن منتقل می‌شوند. دستورهای داخل این بافر می‌توانند در صورت آماده بودن داده‌های مورد نیازشان خارج از نوبت اجرا شوند، اما نتایج هر دستوری که آماده می‌شود به همان بافر برمی‌گردد. از این جا به بعد نوبت رعایت می‌شود و هر نتیجه به ترتیب و بعد از نتایج دستورهای قبلی و قبل از نتایج دستورهای بعدی به رجیسترهای پردازنده منتقل می‌شود.

اگر چه اجرای خارج از نوبت دستورها زمان زیادی از وقت پردازش پردازنده را کاهش می‌دهد، اما استفاده از سیستم بافر برای ذخیره دستورها و نتایج آنها هم کار پیچیده‌ای است. البته تمامی پردازنده‌های مدرن از این فناوری استفاده می‌کنند.

تنها پردازنده‌هایی که از اجرای خارج از نوبت دستورها استفاده نمی‌کنند، پردازنده‌های سری اتم اینتل هستند و شاید حالا متوجه شده باشید که چرا آنها نسبت به سایر پردازنده‌های اینتل تنبل‌تر هستند!

هسته‌ها و رشته‌ها

تا این جا نحوه کارکرد یک هسته مجزا را شرح دادیم، اما واقعیت این است که اغلب پردازنده‌های امروزی ۲ هسته یا بیشتر دارند. بنابراین آنها می‌توانند همزمان بیش از یک دستورالعمل را اجرا کنند. معنی این عبارت آن است که می‌توان برنامه‌های بیشتری را به صورت همزمان در کنار هم اجرا کرد. هر چند توسط یک هسته هم می‌توان چند برنامه را به صورت همزمان اجرا کرد، اما وقتی با استفاده از چند هسته این کار را انجام دهیم، راندمان به مراتب بالاتر خواهد رفت.

اجرای چند برنامه (نه چند دستورالعمل) به صورت همزمان را Multi-Tasking می‌نامند. اینکه در ویندوز به صورت همزمان در اینترنت باشید و موسیقی گوش دهید، نمونه‌ای از Multi-Tasking است. اگر پردازنده سیستم‌تان یک هسته داشته باشد،

SpeedStep و Turbo Boost

خیلی از کاربران به طور سنتی فکر می‌کنند پردازنده همیشه در سرعت اسمی خود کار می‌کند، در حالی که بیش از یک دهه است که پردازنده‌ها نسبت به لود کاری‌شان به صورت پویا تغییر فرکانس می‌دهند. فناوری Intel SpeedStep برای اولین بار با پردازنده پنتیوم۳ ارایه شد که در صورتی که پردازنده با تمام توان کار نمی‌کرد، به طور خودکار فرکانس آن کاهش می‌یافت و در صورتی که با یک پردازش سنگین روبه‌رو می‌شد، بلافاصله فرکانس را افزایش می‌داد. این قابلیت به کامپیوتر اجازه می‌دهد در زمانی که کاربر نیاز دارد با حداکثر توان کار کند و زمانی که کاربر پردازش سبکی مثل وبگردی دارد، با سرعت پایین‌تر و در نتیجه با مصرف توان کمتر و تولید حرارت پایین‌تر کار کند. پردازنده‌های AMD هم سال‌هاست که به این فناوری مجهزند. PowerNow! نام این فناوری AMD در پردازنده‌های موبایل و Cool'n'Quiet هم نام این فناوری AMD در پردازنده‌های دسکتاپ است.

با معرفی پردازنده Core i7 در سال ۲۰۰۸ اینتل این اصل را توسعه بخشید و Turbo Boost نامید. Turbo Boost براساس این اصل استوار است که پردازنده ۴ هسته‌ای Core i7 یک TDP مشخص دارد. این TDP بر مبنای کارکرد با تمام توان هر ۴ هسته محاسبه شده است. حال اگر فقط یک یا ۲ هسته کار کند، امکان افزایش فرکانس اسمی آنها وجود دارد، چرا که در این صورت توان مصرفی آن دو هسته برابر توان مصرفی ۴ هسته در حالت استاندارد می‌شود و مشکل افزایش توان مصرفی و افزایش غیرمجاز حرارت پردازنده رخ نمی‌دهد.

در پردازنده‌های مدرن Core i5 و Core i7 این قابلیت بسیار بارز است و حتی در صورتی که هر ۴ هسته فعال باشند نیز عمل می‌کند. AMD نیز همین فناوری را با نام Turbo Core در پردازنده‌های جدیدش مثل Phenom II X6 و پردازنده‌های سری A گنجانده است. بنابراین فراموش نکنید که وقتی می‌گویید فلان پردازنده ۲,۴ گیگاهرتز است، در واقع شاید چیزی در حدود ۸۰درصد مواقع آن پردازنده در فرکانسی پایین‌تر از ۲,۴ گیگاهرتز کار کند و البته در مواردی خاص فرکانسی بالاتر از ۲,۴ گیگاهرتز هم خواهد داشت.

زمـان پردازش خود را بین ایـن دو برنامه یعنی IE و Media Player تقسـیم می‌کند، مثلا یک نانوثانیه بـه IE و یک نانوثانیه بـه Media Player می‌دهد و دوبـاره بـه IE باز می‌گردد. تقسـیم زمان پردازش پردازنـده بیـن چنـد برنامـه را Time-Sharing می‌نامند. سـوییچ پردازنده بین برنامه‌ها آن قدر سریع است که کاربر به هیچ عنوان متوجه آن نمی‌شود.

اما اگر پردازنده دو هسـته داشته باشد، یک هسته به IE و هسته دیگر به Media Player اختصاص داده می‌شـود و در نتیجـه راندمان کار بالاتـر خواهد بود. اگـر در یک پردازنده دوهسـته‌ای به صورت همزمان ۴ برنامه اجرا شـود، دوباره هر هسـته به نوبه خود از Time-Sharing استفاده می‌کند.

اسـتفاده از یک پردازنده چندهسته‌ای به خودی خود منجر به افزایش سـرعت یک برنامه خاص نمی‌شود. دلیـل امر آن اسـت که اغلـب برنامه‌ها بـه صورتی برنامه‌نویسـی شـده‌اند که هـر دسـتورالعمل بعد از دسـتورالعمل بعدی اجرا گردد، نه اینکه همزمان اجرا شـوند. این پردازش ترتیبی که رشـته یا thread هم نامیده می‌شـود، مناسب استفاده از چند هسته نیست. دلیل اسـتفاده از نام رشـته برای این گونه پردازش‌ها آن اسـت که دستورات پشت سر هم و به ترتیب باید اجرا شوند و نمی‌توان ترتیب آنها را به هم زد و یا چند دستورالعمل را همزمان اجرا کرد.

البته کاربردهایی هم هستند که قابلیت تفکیک شدن به چند رشـته را دارند. برای مثـال برنامه‌ای را تصور کنید که قرار است یک فولدر پر از فایل‌های صوتی را به mp3 تبدیل کند. اگر از یک پردازنده ۴ هسـته‌ای اسـتفاده شود، هر هسته وظیفه تبدیل یک فایل را بر عهـده می‌گیرد و در نتیجه بـه صورت همزمان روی ۴ فایـل عمل تبدیـل انجام می‌شـود. بنابراین زمان تبدیل کل فایل‌ها نسـبت به پردازنده تک‌هسته‌ای به یک‌چهارم کاهش می‌یابد. اغلب برنامه‌های رندرینگ سه‌بعدی می‌توانند از چند هسته استفاده کنند.

پردازنده‌هـای اینتـل از یـک ویژگـی بـه نـام Hyper-Threading بـرای رسـیدن بـه راندمانی بالاتـر اسـتفاده می‌کنند. بـا این فناوری هر هسـته پردازنـده خـودش را به صورت ۲ هسـته مجازی به سیستم‌عامل نشان می‌دهد. در حقیقت هر هسته فقط می‌توانـد یک دسـتورالعمل را در هر لحظه اجرا کند، اما از آنجایی که این گونه پردازنده‌ها دو سری رجیستر مجزا دارند، می‌توانند بین دو thread سـوییچ کرده و مانند Time-sharing عمل نمایند، با این تفاوت که در Time-Sharing فقط یک سـری رجیستر وجود دارد، اما در Hyper-Threading دو سـری رجیستر وجـود دارد و در نتیجه Time-Sharing با راندمانی به مراتب بالاتر انجام خواهد شد.

البتـه واضـح اسـت کـه یـک هسـته بـا Hyper-Threading راندمانی پایین‌تر از دو هسـته فیزیکی جداگانه دارد، اما خیلی بهتر از یک هسته بدون Hyper-Threading است. تست‌های مختلف نشان می‌دهـد کـه Hyper-Threading موجب افزایش راندمان تا ۳۰ درصد بالاتر نسـبت بـه همان پردازنده بدون Hyper-Threading می‌گردد.

دستورهای توسعه‌یافته

در ابتدای مقاله ما پردازنده را به صورت یک ماشـین حسـاب فرض کردیم. این یک قیاس اشـتباه نیست اما از پردازنده‌های مدرن چنان ویژگی‌های پیشـرفته‌ای دارند که ماشـین حسـاب فقط می‌تواند رویای آنها را در سر بپروراند!

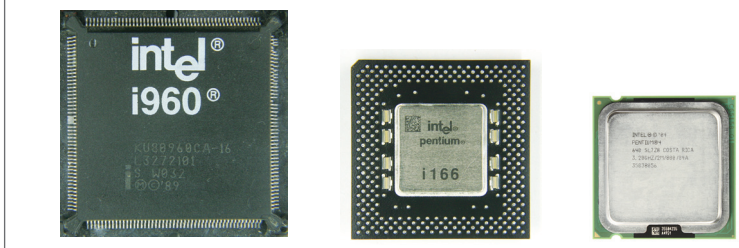
برای مثال اغلب ماشـین حساب‌های جیبی می‌توانند تا ۸ رقم را نشـان دهند کـه بنابراین بزرگ‌ترین عدد قابل نمایش با آنها ۹۹,۹۹۹,۹۹۹ اسـت. این در حالی اسـت که تمامی پردازنده‌های مـدرن می‌توانند روی اعداد ۳۲ بیتی باینری کار کنند که به معنای اسـتفاده از اعـداد تـا ۴,۲۹۴,۹۶۷,۲۹۶ رقم اسـت. همچنین با توجه به اینکـه تمامی پردازنده‌های مدرن از پردازش ۶۴ بیتی هم پشـتیبانی می‌کننـد، در نتیجه می‌توانند از اعـداد تـا ۱۸,۴۴۶,۷۴۴,۰۷۳,۷۰۹,۵۵۱,۶۵۶ رقم نیـز اسـتفاده نماینـد. در واقع در سیسـتم‌عامل‌های ۶۴ بیتی امکان اسـتفاده از میلیون‌ها ترابایت حافظه وجود دارد، در حالی که در سیستم‌عامل‌های ۳۲ بیتی تنها ۴ گیگابایت حافظه قابل استفاده است.

پردازنـده از یک سـری دسـتورهای توسـعه‌یافته یا Extension پشتیبانی می‌کند که این دستورها منجر به افزایش سـرعت آن در بعضی از کاربردهای خاص می‌گـردد. مثلا دسـتورهای SSE نمونه خوبی از این سری دسـتورهای توسـعه‌یافته اسـت که در تمامی پردازنده‌های مدرن دیده می‌شـود. دستورهای SSE زمانی کاربرد دارند که قرار اسـت یک کار مشـخص مثلا جمع یا تفریق بر روی محدوده‌ای از داده‌ها صورت گیرد. SSE این داده‌ها را به قسـمت‌های مجزا تبدیل کرده و سپس همه قسمت‌ها را به سرعت انجام می‌دهد. در یک کلام باید گفت که SSE زمانی کاربرد دارد که قرار اسـت یک دسـتور بر روی تعداد زیادی داده اجرا شود. اسـتفاده از این گونه دسـتورها در پردازنده، کار مهندسی پیچیده و دقیقی را در مرحله طراحی پردازنده می‌طلبـد، اما وقتی از این ویژگی اسـتفاده می‌کنید به ارزش آن پی می‌برید و افزایش راندمان ناشـی از آن شـما را متعجب می‌کند. SSE در پردازش ویدیویی و فشرده‌سازی داده‌ها بسیار موثر است.

نـوع دیگـری از دسـتورهای توسـعه‌یافته کـه در پردازنده‌هـای مـدرن دیده می‌شـود، مجازی‌سـازی سـخت‌افزاری یـا Virtualization اسـت. معمولا نرم‌افزارهای مجازی‌سـازی به عنوان یک واسطه بین محیط مجازی و سـخت‌افزار واقعی عمل می‌کنند که می‌تواند موجب کندی قابل توجه شـود. دسـتورهای توسعه‌یافته مجازی‌سازی به کدهای محیط مجازی این اجازه را می‌دهند که مستقیما روی پردازنده اجرا شوند، بدون اینکه از واسطه‌ای استفاده شود و در نتیجه باعث افزایش قابل توجه سرعت محیط مجازی می‌گردند.

دسـتورهای توسـعه‌یافته بـرای ایجـاد امنیـت هم کاربـرد دارنـد. اینتـل در سـال ۲۰۱۰ پردازنده‌های Core i3 و Core i5 و Core i7 خـود را بـا یک سـری دسـتورهای توسـعه‌یافته جدیـد به نام AES-NI عرضه کرد که پردازنده را با سـرعت بالا قادر بـه رمزنگاری و رمزگشـایی داده‌ها با اسـتفاده از الگوریتـم AES می‌کـرد. اینها تنهـا نمونه‌ای از دستورهای توسـعه‌یافته بودند و اشاره به همه آنها از حوصله این مقاله خارج است.

همه این دستورهای توسـعه‌یافته و ویژگی‌ها موجب می‌شـوند تا فاصله بین پردازنده‌های مدرن و ماشین حسـاب فرسنگ‌ها شـود. در حقیقت پردازنده معجزه مهندسـی اسـت. وقتی به این موضوع فکر کنید که از عمـر پردازنده فقـط نیم قرن می‌گـذرد، این همه پیشرفت غیرقابل تصور می‌شود، اما از آن هیجان‌آورتر اندیشیدن به آینده پردازنده‌هاست! ■



پردازنده‌های گوناگون شرکت اینتل