



مدرس‌ان شریف

فصل اول

« نگاه کلی به سخت افزار کامپیوتر »

۱-۱ مقدمه

سیستم عامل هم به عنوان یک برنامه، سخت افزار کامپیوتر را مدیریت می‌کند و هم بستری را برای اجرای برنامه‌های کاربردی و برنامه‌های سودمند فراهم می‌سازد. از این رو به عنوان واسطی میان کاربر و سخت افزار عمل می‌کند. برای درک مناسب عملکرد سیستم عامل و نکات مربوط به آن، آشنایی مختصری با سخت‌افزار کامپیوتر و معماری آن ضروری است. لذا این فصل، مروری اجمالی از سخت‌افزار کامپیوتر، عملکردهای اساسی راه‌اندازی سیستم، ورودی / خروجی و ذخیره‌سازی را ارائه می‌کند.

۲-۱ عناصر اصلی

به طور کلی، یک کامپیوتر شامل یک یا چند نمونه از هر یک از اجزاء پردازنده، حافظه و ورودی/خروجی است. این اجزاء به گونه‌ای به یکدیگر متصل شده‌اند تا کار اصلی کامپیوتر یعنی اجرای برنامه‌ها را فراهم کنند. از این رو چهار جزء ساختاری و اصلی زیر وجود دارند:

۱- پردازنده: عملیات کامپیوتر را کنترل می‌کند و اعمال پردازش داده‌ها را انجام می‌دهد. وقتی فقط یک پردازنده وجود دارد، معمولاً به آن واحد پردازشگر مرکزی ((Central Processing Unit (CPU)) می‌گویند.

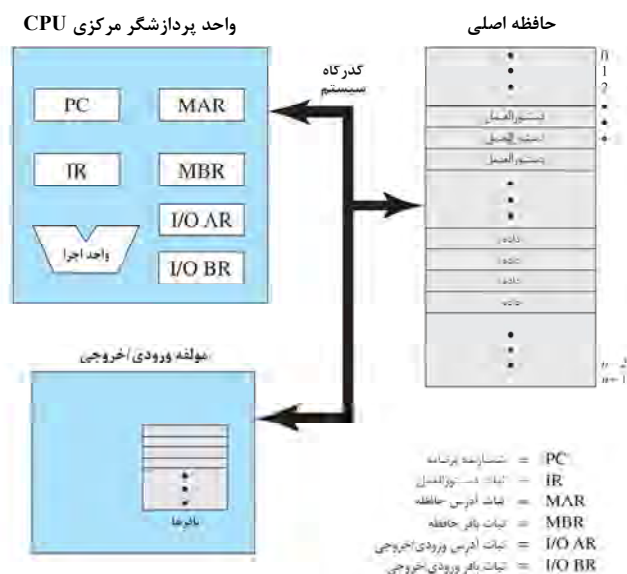
۲- حافظه اصلی: داده‌ها و برنامه‌ها را ذخیره می‌کند. این حافظه نوعاً ناپایدار است و به آن حافظه حقیقی یا اولیه نیز گفته می‌شود.

۳- مؤلفه‌های ورودی/خروجی: داده‌ها را بین کامپیوتر و محیط خارجی آن منتقل می‌کنند. محیط خارجی شامل انواع دستگاه‌ها از جمله حافظه ثانویه، تجهیزات ارتباطی و ترمینال‌هاست.

۴- گذرگاه سیستم: ساختارها و راهکارهایی که ارتباط بین پردازنده، حافظه اصلی و مؤلفه‌های ورودی/خروجی را فراهم می‌کنند.

شکل ۱-۱ این اجزاء را از دید کلی نشان می‌دهد. یکی از اعمال پردازنده مبادله با حافظه است. برای این منظور معمولاً از دو ثبت داخلی (برای پردازنده) استفاده می‌شود: ثبت آدرس حافظه ((Memory Address Register (MAR))، که محل خواندن یا نوشتن بعدی در حافظه را مشخص می‌کند و ثبت بافر حافظه ((Memory Buffer Register (MBR))، که در برگیرنده داده‌هایی است که قرار است در حافظه نوشته یا از آن خوانده شود. مشابه دو ثبت فوق، از یک ثبت آدرس ورودی/خروجی ((I/O Address register (I/OAR)) که یک دستگاه ورودی/خروجی خاص را مشخص می‌کند و از یک ثبت بافر ورودی/خروجی ((I/O Buffer Register (I/OBR)) برای تبادل داده‌ها بین پردازنده و یک مؤلفه ورودی/خروجی استفاده می‌شود.

یک مؤلفه ورودی/خروجی، داده‌ها را از دستگاه‌های خارجی به پردازنده و حافظه یا برعکس منتقل می‌کند. مؤلفه‌های ورودی/خروجی دارای بافرهای داخلی هستند تا بتوانند داده‌ها را تا زمان انتقال آنها نگهداری کنند.



شکل ۱-۱. نگاه کلی به اجزاء کامپیوتر

برای این که یک کامپیوتر اجرای خود را شروع کند؛ مثلاً موقع زدن دکمه power یا راهاندازی مجدد، جهت اجرا، به یک برنامه اولیه نیاز دارد. این برنامه اولیه یا برنامه بالا آورنده (Bootstrap program) باید ساده باشد که نوعاً در حافظه فقط خواندنی (Read Only Memory (ROM)) یا حافظه فقط خواندنی برنامه‌پذیر پاک شدنی (Electrically erasable programmable read only memory (EEPROM)) در سخت افزار کامپیوتر ذخیره می‌شود. این برنامه بالا آورنده، تمام وجوه سیستم را از ثبات‌های پردازنده گرفته تا کنترلر دستگاه‌ها و محتویات حافظه، مقداردهی اولیه می‌نماید. برنامه بالا آورنده باید چگونگی بارگذاری و شروع اجرای سیستم عامل را بداند. برای رسیدن به این هدف، برنامه بالا آورنده باید هسته سیستم عامل را یافته و داخل حافظه بارگذاری کند.

۱-۳ پردازنده

پردازنده به عنوان مهم‌ترین مؤلفه‌ی سخت‌افزار کامپیوتر شناخته می‌شود و مشتمل بر واحد محاسبات منطقی (ALU)، واحد کنترل (CU) و ثبات‌ها می‌باشد. وظیفه‌ی پردازنده، اجرای برنامه‌هایی است که درون حافظه اصلی هستند. هر برنامه‌ای که بخواهد اجرا شود شامل مجموعه‌ای از دستورالعمل‌ها است و پردازنده برای اجرای برنامه‌ها لازم است این دستورالعمل‌ها را طی فرآیندهای واگشی (خواندن دستورالعمل از حافظه)، رمزگشایی (تفسیر دستورالعمل و مشخص کردن عملی که باید انجام شود) و اجرا (کنترل روند اجرای دستورالعمل) به انجام برساند. این فرایند که برای پردازش یک دستورالعمل لازم است، چرخه دستورالعمل نامیده می‌شود.

ثبات‌های زیر برای اجرای دستورالعمل‌ها حائز اهمیت هستند:

- شمارنده برنامه (Program Counter (PC)): حاوی آدرس دستورالعمل است که باید واگشی شود.
- ثبات دستورالعمل (Instruction Register (IR)): حاوی آخرین دستورالعملی است که واگشی شده است.

تمام پردازنده‌ها شامل یک یا مجموعه‌ای از ثبات‌ها هستند که معمولاً تحت عنوان کلمه وضعیت برنامه (Program Status Word (PSW)) حاوی اطلاعات وضعیت هستند. این ثبات، معمولاً علاوه بر کدهای وضعیت، شامل اطلاعات دیگری مانند بیت فعال / غیرفعال کردن وقفه و بیت حالت کاربر / هسته، نیز می‌باشد.

۱-۳-۱ سیستم‌های تک پردازنده

اکثر سیستم‌ها از یک پردازنده منفرد استفاده می‌کنند. تنوع سیستم‌های تک‌پردازنده، می‌تواند جالب توجه باشد، چرا که این سیستم‌ها از دستیار دیجیتال شخصی (Personal Digital Assistant (PDA)) تا کامپیوترهای بزرگ را شامل می‌شوند. در یک سیستم تک پردازنده، یک پردازنده اصلی وجود دارد که قابلیت اجرای مجموعه دستورالعمل‌های همه منظوره، شامل دستورالعمل‌های پردازنده‌های کاربر را دارد. تقریباً همه سیستم‌ها، پردازنده‌های خاص منظوره‌ی دیگری هم دارند. آنها ممکن است به شکل پردازنده‌های خاص دستگاه، نظیر دیسک، صفحه کلید و کنترلرهای گرافیکی باشند؛ یا در کامپیوترهای بزرگ، بیشتر به شکل پردازنده‌های همه منظوره، نظیر پردازنده‌های ورودی/خروجی که داده‌ها را به سرعت بین مؤلفه‌های سیستم انتقال می‌دهند، کار گذاشته شوند.

۱-۳-۲ سیستم‌های چند پردازنده

سیستم‌های چند پردازنده (Multiprocessor)، معروف به سیستم‌های موازی یا سیستم‌های با اتصال محکم (Tightly coupled systems) از اهمیت رو به رشدی برخوردار هستند. این قبیل سیستم‌ها، دو یا چند پردازنده با ارتباط نزدیک به هم دارند که به صورت مشترک از گذرگاه سیستم و گاهی ساعت، حافظه و دستگاه‌های جانبی استفاده می‌کنند.

سیستم‌های چندپردازنده سه مزیت اصلی دارند که عبارتند از:

- ۱- **بازده بالا:** با زیاد شدن تعداد پردازنده‌ها، انتظار داریم که کار انجام شده‌ی بیشتری در زمان کمتری بدست آوریم. با این وجود، ضریب تسریع با N پردازنده، N نیست و کمتر از N خواهد بود. زمانی که چندین پردازنده روی یک وظیفه با یکدیگر همکاری دارند، به منظور حفظ درستی کار تمام قسمت‌ها، مقداری سربار تولید می‌شود. این سربار به علاوه بر رقابت برای منابع مشترک، منافع مورد انتظار ناشی از افزایش تعداد پردازنده‌ها را کاهش می‌دهد.
- ۲- **صرفه اقتصادی:** سیستم‌های چندپردازنده هزینه کمتری نسبت به سیستم‌های تک‌پردازنده متعدد معادل خود دارند، زیرا می‌توانند دستگاه‌های جانبی، حافظه ثانویه و مولدهای برق را به صورت مشترک استفاده نمایند.
- ۳- **قابلیت اطمینان بالا:** اگر وظایف بین پردازنده‌های متعدد به درستی توزیع شوند، در این صورت، خرابی یک پردازنده باعث توقف کل سیستم نمی‌شود و تنها افت کارایی را در پی خواهد داشت. قابلیت اطمینان بالای یک سیستم کامپیوتری در بسیاری از برنامه‌های کاربردی امری حیاتی است.

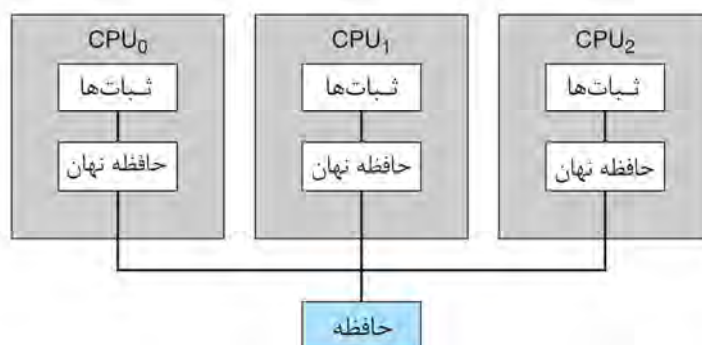
قابلیت تداوم ارائه سرویس، متناسب با سطح بقای سخت‌افزار، تنزل مطبوع (Graceful Degradation) نامیده می‌شود.

سطح بالاتری از تنزل مطبوع تحمل پذیر خطا (Fault Tolerant) نامیده می‌شود. سیستم‌هایی با این قابلیت می‌توانند به رغم از کار افتادن هر یک از تک مولفه‌ها همچنان به عملیات خود ادامه دهند. باید توجه داشته باشید که تحمل‌پذیری خطا مستلزم راهکاری است که خطا را تشخیص، شناسایی و حتی امکان اصلاح نماید.

سیستم‌های چندپردازنده‌ای که امروزه استفاده می‌شوند، دو نوع هستند:

- ۱- **چند پردازشی نامتقارن (Asymmetric Multiprocessing (AMP)):** هر پردازنده به وظیفه‌ای خاص گمارده می‌شود. یک پردازنده راهبر (Master)، سیستم را کنترل می‌کند؛ سایر پردازنده‌ها چشم به راهبر دارند و یا وظایف از پیش تعریف شده‌ای را انجام می‌دهند. این طرح یک رابطه راهبر - پیرو (Master - slave) تعریف می‌کند. پردازنده‌ی بالادست (راهبر)، کارها را زمان‌بندی کرده و در اختیار پردازنده‌های پایین‌دست (پیرو) قرار می‌دهد.
- ۲- **چند پردازشی متقارن (Symmetric Multiprocessing (SMP)):** هر پردازنده تمامی وظایف سیستم عامل را انجام می‌دهد. SMP به این معنی است که تمامی پردازنده‌ها نظیر هم هستند؛ هیچ‌گونه رابطه راهبر - پیرو میان پردازنده‌ها وجود ندارد. شکل ۱-۲ یک نمونه از معماری SMP را نشان می‌دهد. توجه داشته باشید که هر یک از پردازنده‌ها، مجموعه‌ی ثبات‌های خاص خود را در کنار یک حافظه نهان خصوصی (یا محلی) دارند؛ با وجود این، همه پردازنده‌ها در حافظه فیزیکی مشترک هستند.

سیستم‌های چند پردازنده متقارن (SMP)، امکان اشتراک پویای پردازنده‌ها و منابع (مانند حافظه) را میان پردازنده‌های مختلف فراهم ساخته و اختلاف کارایی بین پردازنده‌ها را کاهش می‌دهند.



شکل ۱-۲. معماری چند پردازشی متقارن



سولاریس نمونه‌ای از یک سیستم چند پردازشی متقارن (SMP) است که به صورت یک نسخه تجاری از یونیکس توسط شرکت سان میکروسیستمز طراحی شده است.

سیستم عامل‌های پیشرفته، اعم از ویندوز (xp، 7 و 8)، Mac OS X و لینوکس، از SMP پشتیبانی می‌کنند.

تفاوت چندپردازشی متقارن و نامتقارن می‌تواند ناشی از سخت‌افزار یا نرم‌افزار باشد. سخت‌افزار خاصی ممکن است تفاوت پردازنده‌ها شود یا این که نرم‌افزاری به گونه‌ای نوشته شود که یک پردازنده را راهبر و بقیه را پیرو قلمداد کند. برای نمونه، نسخه ۴ سیستم عامل sun OS شرکت سان میکروسیستمز، چند پردازشی نامتقارن را فراهم می‌کند در حالی که نسخه ۵ (سولاریس) در همان سخت‌افزار به صورت متقارن است.

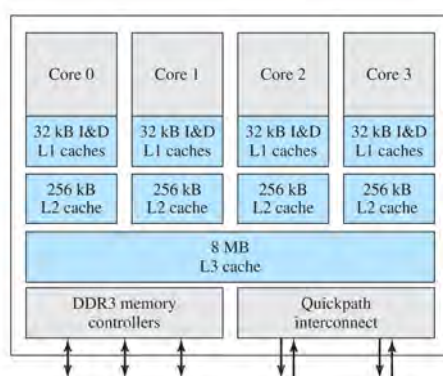
۳-۳-۱ پردازنده‌های چند هسته‌ای

نگرش اخیر طراحی پردازنده این است که بتوان در یک تراشه منفرد، هسته‌های (cores) محاسباتی متعدد کار گذاشت. در اصل این‌ها، تراشه‌های چندپردازنده هستند. یک کامپیوتر چند هسته‌ای که ما آن را به عنوان یک تراشه چند پردازنده می‌شناسیم، دو یا چند پردازنده (هسته) را در یک قطعه سلیکونی ترکیب می‌کند. به عنوان نمونه، هر هسته، متشکل از تمام مولفه‌های یک پردازنده مستقل مانند ثابت‌ها، واحد محاسباتی منطقی، سخت‌افزار خط لوله (Pipe Line)، واحد کنترل به علاوه دستورالعمل حافظه نهان سطح اول (L1) و حافظه نهان داده می‌باشد. تراشه‌های چند هسته‌ای جدید شامل حافظه نهان سطح دوم (L2) و سطح سوم (L3) نیز هستند.

هدف اصلی ایجاد کامپیوترهای چند هسته‌ای، افزایش کارایی با افزایش پیچیدگی طراحی پردازنده، جهت بهره‌برداری از موازی سازی در اجرای دستورالعمل‌ها و دسترسی به حافظه است. آن‌ها می‌توانند کارآمدتر از تراشه‌های متعدد با هسته‌های منفرد باشند، چرا که ارتباط روی تراشه سریع‌تر از ارتباط بین تراشه‌هاست. به علاوه، یک تراشه با هسته‌های متعدد نسبت به تراشه‌های متعدد تک هسته‌ای جریان برق کمتری مصرف می‌کند.

یک مثال از یک سیستم چند هسته‌ای، Intel Core i7 است که شامل چهار پردازنده x86 است. هر یک از این پردازنده‌ها دارای حافظه نهان (L2) اختصاصی به همراه یک حافظه نهان (L3) مشترک می‌باشند. (شکل ۳-۱)

تراشه Core i7، دو حالت از ارتباطات بیرونی با دیگر تراشه‌ها را پشتیبانی می‌کند. DDR3 Memory Controller که کنترلر حافظه برای حافظه اصلی (Double Data Rate (DDR)) است. این رابطه سه کانال ارتباطی با پهنای ۸ بایت را برای یک گذرگاه ۱۹۲ بیتی پشتیبانی می‌کند و قادر به ارسال و دریافت داده‌ها تا 32 GB در ثانیه خواهد بود. از طرف دیگر (Quick Path Interconnect (QPI)) یک اتصال نظیر به نظیر الکتریکی است که می‌تواند ارتباط میان تراشه‌های متصل به پردازنده را با سرعت بالا فراهم کند. اتصال QPI پهنای باندی تا 25.6GB/s را پشتیبانی می‌کند.



شکل ۳-۱. نمودار بلوک Intel Core i7

حافظه نهان (L1)، یک حافظه نهان سریع و کوچک تعبیه شده در درون پردازنده است و به دسترسی سریع پردازنده به داده‌های تکرارشونده (دستورات و داده‌ها) کمک می‌نماید. این نوع حافظه‌ی نهان، سریع ولی گران‌قیمت می‌باشد.

حافظه نهان (L2)، برای دسترسی سریع به داده‌های مهم و تکراری طراحی شده است.

حافظه نهان (L3)، جدیدترین نوع حافظه نهان است که در پردازش به حافظه‌های نهان L1 و L2 کمک می‌کند. این نوع حافظه نهان این قابلیت را به CPU می‌دهد که همزمان قدرت انجام چند عملیات را داشته باشد و در صورت تکمیل ظرفیت حافظه‌های نهان دیگر، در پردازش، آن‌ها را باری می‌کند.

مثال ۱: کدام یک از موارد زیر، مزیت سیستم‌های چند پردازشی متقارن (SMP) نسبت به سیستم‌های چند پردازشی نامتقارن (ASMP) می‌باشد؟

- (۱) متعادل تر شدن بار سیستم
(۲) تحمل پذیری بیشتر خطا
(۳) قابلیت حمل بیشتر بر روی سیستم‌های سخت‌افزاری مختلف
(۴) هر سه گزینه

پاسخ: گزینه «۴» مزیت اصلی SMP، آن است که در آن پردازنده‌های زیادی می‌توانند همزمان بدون افت کارایی در پردازنده‌های متعدد اجرا شوند که هم باعث افزایش کارایی و هم متعادل تر شدن بار سیستم می‌شوند.

مثال ۲: کدام یک از سیستم‌های زیر برای سیستم‌های سرور مناسب می‌باشند؟

- (۱) چند هسته‌ای
(۲) چند پردازشی نامتقارن
(۳) تک هسته‌ای
(۴) تک پردازنده

پاسخ: گزینه «۱» یک تراشه با چند هسته، نسبت به چندین تراشه هر کدام با یک هسته، جریان برق کمتری مصرف می‌کند. در نتیجه، سیستم‌های چند هسته‌ای برای سیستم‌های سروری بسیار مناسب هستند.

۴-۱ وقفه

وقوع یک رویداد معمولاً به وسیله یک وقفه (Interrupt) از جانب سخت افزار یا نرم افزار علامت داده می‌شود. سخت افزار در هر لحظه ممکن است درخواست وقفه یا به معنای دقیق تر درخواست تعلیق (Interrupt Request) (که به اختصار IRQ) نامیده می‌شود) را از طریق گذرگاه سیستم با ارسال یک سیگنال به پردازنده صادر کند.

نرم‌افزار نیز ممکن است با اجرای عملیات خاصی به نام فراخوان سیستم (System Call) (یا فراخوان مانی‌تور) یک وقفه بدهد. جدول زیر، رایج‌ترین دسته‌بندی وقفه‌ها را نشان می‌دهد.

وقفه‌هایی که به وسیله کنترل کننده‌ی ورودی/خروجی تولید می‌شود تا کامل شدن طبیعی یک عمل یا شرایط خطا را اعلام نماید.	ورودی/خروجی	وقفه‌های سخت‌افزاری
وقفه‌ای که توسط زمان سنج داخلی پردازنده تولید می‌شود. این وقفه به سیستم عامل اجازه می‌دهد بعضی اعمال را به طور مرتب انجام دهد.	زمان سنج	
وقفه‌هایی که با نقص سخت‌افزاری تولید می‌شود، مانند نقص برق یا خطای توازن حافظه.	نقص سخت‌افزار	وقفه‌های نرم‌افزاری
وقفه‌هایی که توسط نرم‌افزار به واسطه‌ی یک خطا (مانند تقسیم بر صفر یا دسترسی غیرمجاز به حافظه) و یا با درخواست خاصی از یک برنامه کاربر جهت انجام یک سرویس سیستم عامل تولید می‌شود.	تله (یا استثنا) (Exception)	
این وقفه‌ها زمانی رخ می‌دهد که برنامه‌ی سطح کاربر، از سیستم عامل (سرویس‌های سطح هسته) بخواهد کارهایی را که برای سیستم عامل محفوظ هستند از طرف برنامه کاربر انجام بدهد.	فراخوان‌های سیستمی	

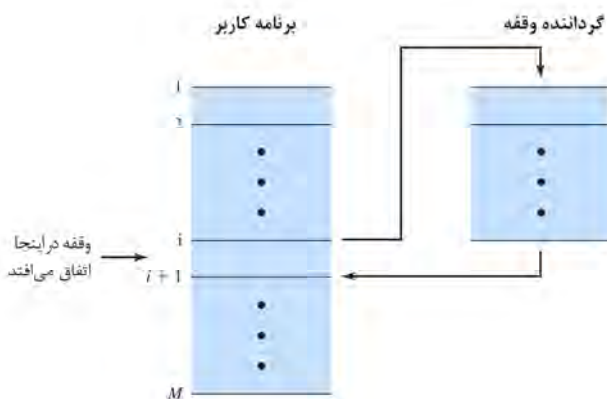
* تذکره ۱: در برخی منابع درسی سیستم عامل، همه وقفه‌های نرم‌افزاری از جمله فراخوان‌های سیستمی را، تله (Trap) در نظر می‌گیرند.

نکته: مهم‌ترین هدف استفاده از راهکار وقفه، افزایش کارایی پردازنده است.

زمانی که پردازنده وقفه‌ای را دریافت می‌کند، کار خود را متوقف کرده و بلافاصله اجرا را به یک مکان ثابت انتقال می‌دهد. مکان ثابت معمولاً آدرس شروع استقرار روال سرویس وقفه (Interrupt Service Routine (ISR)) را در بردارد.

* تذکره ۲: روال سرویس وقفه، گرداننده وقفه (Interrupt Handler) نیز نامیده می‌شود.

پاسخ به وقفه به معنی اجرای روال سرویس وقفه است. به محض تکمیل، پردازنده محاسبات وقفه دیده را از سر می‌گیرد. به عبارت دیگر می‌توان گفت، از نقطه نظر برنامه‌ی کاربر، وقفه تنها قطع موقتی اجرای عادی برنامه است. زمانی که پردازش وقفه کامل شد، اجرای برنامه ادامه می‌یابد. (شکل ۴-۱)



شکل ۱-۴. انتقال کنترل از طریق وقفهها

روال سرویس وقفه یا برنامه گرداننده وقفه، عموماً جزئی از هسته سیستم عامل است و معمولاً این برنامه با توجه به نوع وقفه، اعمال لازم را انجام می‌دهد.

وقفه‌ها یک بخش مهم در معماری کامپیوتر هستند. هر طراحی کامپیوتر، راهکار وقفه خاص خود را دارد، اما چندین وظیفه مشترک در میان آن‌ها وجود دارد. وقفه باید کنترل را به روال سرویس وقفه مناسبی انتقال دهد. روش سراسر انجام این انتقال، فراخوانی یک روال عمومی است تا اطلاعات وقفه را بررسی کند؛ این روال هم خود روال راه‌انداز مخصوص وقفه را فراخوان خواهد کرد. با وجود این، وقفه‌ها باید به سرعت راه‌اندازی شوند. از آنجایی که تنها تعداد از پیش تعریف شده‌ای از وقفه‌ها امکان‌پذیر است، به منظور تسریع این عملیات می‌توان، به جای مکانی ثابت، جدولی از اشاره‌گرها به روال وقفه‌ها را ذخیره ساخت. بدین ترتیب، سرعت سرویس‌دهی به وقفه افزایش می‌یابد. در این روش، روال وقفه به صورت غیرمستقیم از طریق جدول، بدون نیاز به روال عمومی فراخوانی می‌شود.

جدول اشاره‌گرها معمولاً در محل‌های پایین حافظه ذخیره می‌شود (صد محل اول یا محل‌هایی از این دست). این محل‌ها آدرس روال سرویس وقفه‌ها را برای دستگاه‌های مختلف نگه می‌دارند. سپس این آرایه از آدرس‌ها، یا بردار وقفه (Interrupt vector)، بر اساس شماره‌ی منحصر به فرد دستگاه‌ها شاخص‌گذاری می‌شود تا با درخواست وقفه‌ای مشخص، آدرس روال سرویس وقفه را برای دستگاه وقفه دهنده فراهم نماید. وقفه‌های سیستم عامل‌های ویندوز و یونیکس به رغم تفاوت‌هایی که دارند، با این شیوه راه‌اندازی می‌شوند.

کدام گزینه در مورد وقفه‌ها (Interrupt) نادرست است؟

۱) وقفه‌های سخت‌افزاری همگام هستند.

۲) هدف اصلی استفاده از راهکار وقفه، افزایش کارایی پردازنده است.

۳) خطای بیت توازن در حافظه، جزء وقفه نقص سخت‌افزاری محسوب می‌شود.

۴) زمانی که یک فراخوانی سیستمی اجرا می‌شود، سخت‌افزار با آن به عنوان یک وقفه نرم‌افزاری رفتار می‌کند.

پاسخ: گزینه «۱» زمانی که وقفه سخت‌افزاری رخ می‌دهد، از سوی یک سخت‌افزار خارجی (مانند کنترلر دیسک) یک سیگنال هشدار به طور ناهمگام به پردازنده ارسال می‌شود. منظور از ناهمگام این است که وقفه سخت‌افزاری عمدتاً در یک لحظه تصادفی اتفاق می‌افتد که از قبل مشخص نیست در کجای برنامه جاری به وقوع می‌پیوندد و می‌تواند در اثنای اجرای یک دستورالعمل رخ دهد. این در حالی است که وقفه‌های نرم‌افزاری به صورت همگام و در اثر اجرای دستورالعمل خاصی از برنامه جاری (خواسته یا ناخواسته) اتفاق می‌افتد.

۵-۱ ساختار ذخیره‌سازی

پردازنده می‌تواند دستورالعمل‌ها را تنها از حافظه بردارد، لذا هر یک از برنامه‌ها باید برای اجرا در آن‌جا ذخیره شوند. کامپیوترهای همه منظوره، اکثر برنامه‌های خود را از حافظه با قابلیت نوشتن مجدد به نام حافظه اصلی (که حافظه با قابلیت دسترسی تصادفی (Random Access Memory) یا RAM نیز نامیده می‌شود) اجرا می‌کنند. حافظه اصلی عمدتاً با یک فناوری نیمه هادی به نام حافظه با دسترسی تصادفی پویا (Dynamic Random Access Memory (DRAM)) پیاده‌سازی می‌شود. کامپیوترها از شکل‌های دیگر حافظه هم استفاده می‌کنند. حافظه فقط خواندنی (ROM) به دلیل غیرقابل تغییر بودنش برای ذخیره برنامه‌های ایستا استفاده می‌شود. EEPROM نمی‌تواند دائماً تغییر کند و غالباً شامل برنامه‌های ایستا است. همه شکل‌های حافظه، آرایه‌ای از کلمات را فراهم می‌کنند. هر کلمه آدرس خاص خود را دارد. تعامل از طریق دنباله‌ای از دستورالعمل‌های load یا store

به آدرس‌های مشخصی از حافظه صورت می‌گیرد. دستورالعمل load، کلمه‌ای را از حافظه اصلی به یک ثبات داخل پردازنده انتقال می‌دهد و دستورالعمل store، محتویات یک ثبات را به حافظه اصلی منتقل می‌کند. جدا از دستورالعمل‌های مشخص load و store، پردازنده دستورالعمل‌هایی را به صورت خودکار از حافظه اصلی به‌منظور اجرا بارگذاری می‌کند.

نمونه‌ای از یک چرخه‌ی اجرای دستورالعمل که روی یک سیستم با معماری «وُن نیومن» (Von Neumann) به صورت زیر اجرا می‌شود، ابتدا دستورالعملی را از حافظه واکنشی نموده و آن دستورالعمل را در ثبات دستورالعمل (Instruction Register (IR)) ذخیره می‌کند. سپس دستورالعمل رمزگشایی (Decode) شده و ممکن است باعث واکنشی عملوندهایی نیز از حافظه شده و در برخی ثبات‌های داخلی ذخیره گردد. بعد از اجرای دستورالعمل روی عملوندها، ممکن است دو مرتبه نتیجه دستورالعمل در حافظه ذخیره شود. توجه داشته باشید که واحد حافظه، تنها جریانی از آدرس‌های حافظه را می‌بیند و نمی‌داند که آن‌ها چگونه و به چه منظوری (واکنشی داده یا دستورالعمل‌ها) تولید می‌شوند.

در حالت آرمانی می‌خواهیم که برنامه‌ها و داده‌ها به‌طور دائمی در حافظه اصلی ماندگار شوند. این امر به دو دلیل زیر ناممکن است:

۱- حافظه اصلی معمولاً آن‌قدر کوچک است که نمی‌تواند همه برنامه‌ها و داده‌های مورد نیاز را برای همیشه ذخیره کند.

۲- حافظه اصلی یک دستگاه ذخیره ناپایدار (فرآر) است که محتویات خود را با قطع برق یا به دلایل مشابه از دست می‌دهد.

بنابراین، اکثر سیستم‌های کامپیوتری برای گسترش حافظه اصلی، حافظه ثانویه را فراهم می‌کنند. حافظه ثانویه قادر است مقادیر بزرگی از داده‌ها را به صورت دائمی نگهداری کند.

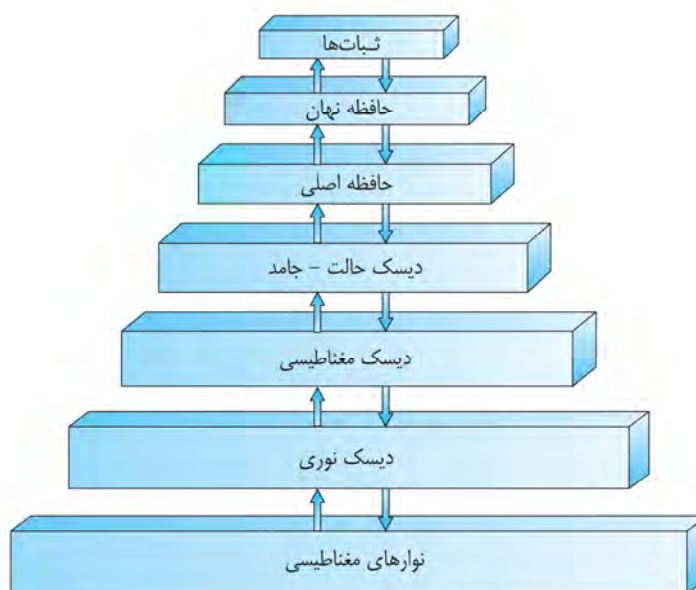
رایج‌ترین دستگاه حافظه ثانویه، دیسک مغناطیسی است که امکان ذخیره‌سازی برنامه‌ها و داده‌ها را فراهم می‌کند. اکثر برنامه‌ها (مرورگرهای وب، کامپایلرها، واژه‌پردازها و غیره) روی دیسک ذخیره می‌شوند تا این‌که در داخل حافظه اصلی بارگذاری شوند. بنابراین، بسیاری از برنامه‌ها، دیسک را همزمان به عنوان مبدأ و مقصد پردازش خود استفاده می‌کنند.

سیستم‌های ذخیره‌سازی دیگری نیز اعم از حافظه نهان، CD-ROM، نوارهای مغناطیسی و غیره وجود دارند. هر سیستم ذخیره‌سازی، امکاناتی برای ذخیره و نگهداری داده فراهم می‌کند.

تفاوت اصلی انواع سیستم‌های حافظه به

سرعت (زمان دسترسی)، هزینه، ظرفیت (حجم فیزیکی) و پایداری یا ناپایداری آن‌ها برمی‌گردد.

انواع سیستم‌های حافظه را می‌توان در یک سیستم کامپیوتری براساس سرعت و هزینه در یک سلسله مراتب سازماندهی کرد (شکل ۱-۵). سطوح بالاتر گران قیمت، اما سریع هستند.



شکل ۱-۵. سلسله مراتب حافظه

به‌طور کلی، در این ساختار سلسله مراتب از بالا به پایین که حرکت می‌کنیم، شرایط زیر اتفاق می‌افتد:

۱- کاهش هزینه به ازای هر بیت ۲- افزایش زمان دسترسی ۳- افزایش ظرفیت ۴- کاهش تعداد دفعات مراجعه پردازنده به حافظه

بنابراین حافظه‌های کوچک‌تر، گران‌تر و سریع‌تر با حافظه‌های بزرگ‌تر، ارزان‌تر و کندتر تلفیق شده‌اند.



پایه صحت شرط ۴ (مورد آخر)، اصلی است به نام اصل محلی بودن مراجعات (Locality Of Reference). در طول اجرای برنامه، مراجعات پردازنده به حافظه، برای داده‌ها و یا برای دستورالعمل‌ها، نزدیک به هم هستند. به عبارت دیگر، اگر به محلی از حافظه مراجعه شود، به احتمال زیاد در آینده نزدیک مجدداً به محدوده‌ی آن محل مراجعه خواهد شد. برنامه‌ها نوعاً تعدادی حلقه و زیربرنامه دارند. با ورود به یک حلقه یا زیربرنامه، مراجعات مکرری به مجموعه‌ی کوچکی از دستورالعمل‌ها وجود دارد. کار روی جدول‌ها و آرایه‌ها نیز متضمن دسترسی به مجموعه‌ای از کلمات داده‌هاست که دسته‌بندی شده‌اند. در درازمدت، دسته‌های مورد استفاده تغییر می‌کنند، ولی برای یک مدت کوتاه، پردازنده عمدتاً با دسته‌های ثابتی از آدرس‌های حافظه کار می‌کند. با توجه به مطلب فوق، سازماندهی داده‌ها در سطوح مختلف حافظه به گونه‌ای که درصد دسترسی به سطوح پایین‌تر، کم‌تر از درصد دسترسی به سطوح بالاتر باشد، امکان دارد.

اصل محلی بودن مراجعات:

مراجعات به حافظه دسته‌بندی شده هستند. در درازمدت دسته‌های مورد استفاده تغییر می‌کنند، ولی برای یک مدت کوتاه، پردازنده عمدتاً با دسته‌های ثابتی از مراجعه به حافظه کار می‌کند.

علاوه بر این که سیستم‌های حافظه از نظر سرعت و هزینه متفاوت هستند، از نقطه نظر پایداری و ناپایداری نیز با هم فرق دارند. چنانچه قبلاً متذکر شدیم، حافظه ناپایدار (فرار) (Volatile Storage)، محتویات خود را با قطع برق دستگاه از دست می‌دهد. در غیاب باتری و سیستم‌های پشتیبان مولد گران قیمت، داده‌ها باید برای حفاظت و ایمنی بیش‌تر در حافظه پایدار (غیر فرار) (Nonvolatile Storage) نوشته شوند. در سلسله مراتب نشان داده شده در شکل ۱-۵، سیستم‌های حافظه بالای دیسک حالت جامد، ناپایدار (فرار) هستند. دیسک‌های حالت جامد یا دیسک‌های حالت پایدار ((Solid - State Disk (SSD))، انواع مختلفی دارند و در واقع سریع‌تر از دیسک‌های مغناطیسی و البته پایدار (غیر فرار) هستند. یکی از انواع دیسک حالت جامد، طی عملیات عادی، داده‌ها را در یک آرایه ((Dynamic Random Access Memory) DRAM) بزرگی ذخیره می‌کند که ناپایدار است. اما شامل یک دیسک سخت مغناطیسی پنهان و یک باتری برای برق پشتیبان‌گیر هستند. اگر برق قطع شود، کنترلر دیسک حالت جامد، داده‌ها را از RAM به دیسک مغناطیسی کپی می‌کند. زمانی که برق دومرتبه وصل شود، کنترلر، داده‌ها را دوباره به RAM کپی می‌کند. شکل دیگر دیسک حالت جامد، حافظه فلش می‌باشد که در دوربین‌ها و دستیاران دیجیتال شخصی ((PDAها) (Personal Digital Assistants)، ربات‌ها و به صورت فزاینده در کامپیوترهای همه منظوره رواج پیدا کرده است.

دیسک حالت جامد (SSD)، به دیسکی گفته می‌شود که در آن هیچ قطعه متحرکی وجود ندارد. هیچ قسمتی از اجزاء این نوع دیسک در حال چرخش نیست. همان‌طور که می‌دانیم دیسک‌های سخت معمولی (HDD)، چندین گرداننده دیسک و هد متحرک دارند. دیسک حالت جامد از نظر ابعاد و شکل ظاهری، مشابه دیسک‌های سخت معمولی هستند، اما قطعات متحرک ندارند و به جای آن‌ها از حافظه‌های فلش و حافظه‌های SDRAM (Synchronous DRAM) استفاده می‌شود. دیسک‌های حالت جامد به دلیل نداشتن قطعه متحرک زمان تأخیر بسیار کم‌تری نسبت به دیسک‌های معمولی دارند. از لحاظ مصرف انرژی و تولید حرارت در کامپیوتر، دیسک‌های حالت جامد به مراتب انرژی کم‌تری مصرف می‌کنند و حرارت کم‌تری نیز تولید می‌کنند. از این‌رو، استفاده از آن‌ها در کامپیوترهای رومیزی و لپ‌تاپ‌ها بسیار ایده‌آل و مقرون به صرفه است.

در اوایل سال ۲۰۱۴ شرکت HP پروژه‌ای به نام "The Machine" را معرفی کرد که با تولید انبوه آن در سال‌های آینده مفهوم سلسله مراتب حافظه با رویکرد کلاسیک فعلی احتمالاً به فراموشی سپرده خواهد شد. این سخت‌افزار توانایی آدرس‌دهی ۱۶۰ پتابایت اطلاعات در کمتر از ۲۵۰ نانوثانیه را داراست و بدین ترتیب می‌توان انتظار تولد حافظه‌های ماندگاری با ظرفیت‌ها و سرعت‌های بسیار بالا را در چند سال آینده داشت.

۱-۶ حافظه نهان

در تمام چرخه‌های دستورالعمل، پردازنده حداقل یک‌بار برای واکنشی دستورالعمل و غالباً بیش از یک‌بار برای واکنشی عملوندها و / یا ذخیره‌سازی نتایج به حافظه مراجعه می‌کند. واضح است که نرخ اجرای دستورالعمل‌ها توسط پردازنده محدود به زمان چرخه‌ی حافظه است. در حقیقت این محدودیت به خاطر مسأله ریشه‌دار عدم تطابق سرعت پردازنده و سرعت حافظه بوده است. در طی سالیان متمادی، افزایش سرعت پردازنده همواره از افزایش سرعت دسترسی به حافظه بیش‌تر بوده است. چیزی که با آن مواجه هستیم، تعادل بین سرعت، هزینه و اندازه است.

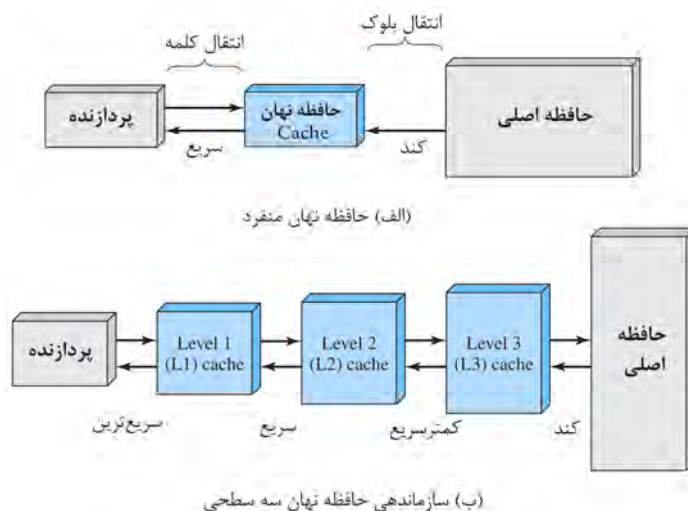
حالت ایده آل این است که حافظه اصلی با همان فناوری ثبات‌های حافظه ساخته شود تا زمان چرخه حافظه قابل مقایسه با زمان چرخه پردازنده باشد. اما این راهبرد همواره بسیار گران بوده است. بهره‌گیری از اصل محلی بودن مراجعات به حافظه راه حل مسأله است. این بهره‌گیری با استفاده از یک حافظه کوچک و سریع بین پردازنده و حافظه اصلی، به نام **حافظه نهان** (cache) فراهم می‌گردد.

حافظه‌ای که برای جبران اختلاف سرعت پردازنده و حافظه اصلی استفاده می‌شود، **حافظه نهان** نام دارد.

نقش حافظه نهان ارائه سریع‌ترین حافظه‌ی موجود و در عین حال ارثه‌ی حافظه‌ای بزرگ از انواع ارزان‌تر حافظه‌های نیمه هادی است. این مفهوم در شکل ۶-۱ (الف) تبیین شده است. یک حافظه نسبتاً بزرگ و کند از حافظه اصلی به همراه یک حافظه کوچک‌تر و سریع‌تر حافظه نهان وجود دارد. حافظه نهان حاوی بخشی از حافظه اصلی است. وقتی پردازنده می‌خواهد کلمه‌ای از حافظه بخواند، وجود آن را در حافظه نهان بررسی می‌کند. اگر وجود داشته باشد، به پردازنده تحویل می‌شود. در غیر این صورت یک بلوک از حافظه اصلی، شامل تعداد ثابتی از خانه‌های حافظه، به حافظه نهان منتقل شده و سپس کلمه موردنظر به پردازنده تحویل داده می‌شود. هنگامی که یک بلوک از داده‌ها به حافظه نهان آورده می‌شود تا یک مراجعه به حافظه انجام شود، به دلیل اصل محلی بودن مراجعات، احتمالاً به زودی به دیگر کلمات آن بلوک نیز مراجعه خواهد شد.

شکل ۶-۱ (ب)، استفاده از حافظه نهان سه سطحی را نشان می‌دهد. حافظه نهان سطح دوم ((L2 Level 2)، کندتر و بزرگ‌تر از حافظه نهان سطح اول (L1) و حافظه نهان سطح سوم (L3)، کندتر و بزرگ‌تر از حافظه نهان سطح دوم (L2) است.

به عبارت دیگر می‌توان گفت، حافظه سطح بالاتر (L1)، کوچک‌تر، سریع‌تر و گران‌تر (به ازای هر بیت) از حافظه سطح پایین‌تر (L2 و L3) است.



شکل ۶-۱. حافظه اصلی و حافظه نهان

۱-۷ عملکرد حافظه‌های دو سطحی

همان‌طور که قبلاً اشاره کردیم، حافظه نهان به عنوان بافری بین حافظه اصلی و پردازنده برای ایجاد یک حافظه داخلی دو سطحی مطرح شد. این معماری دو سطحی با استفاده از خاصیت محلی بودن مراجعات، موجب ارتقای کارایی در مقایسه با حافظه تک سطحی شده است. راهکار حافظه اصلی و حافظه نهان بخشی از معماری کامپیوتر است که در سخت افزار پیاده‌سازی شده و نوعاً برای سیستم عامل قابل رؤیت نمی‌باشد. با این حال دو رویکرد دیگر از حافظه دو سطحی وجود دارد که حداقل بخشی از آن‌ها در سیستم عامل پیاده‌سازی شده است: یکی حافظه مجازی و دیگری حافظه نهان دیسک است که در فصل‌های پایانی به بررسی آن‌ها خواهیم پرداخت. در این قسمت به بعضی ویژگی‌های عملکرد حافظه‌های دو سطحی نگاه مختصری می‌کنیم. عامل برتری عملکرد حافظه دو سطحی، اصلی به نام محلی بودن مراجعات است که پیش‌تر آن را معرفی کردیم. اصولاً اصل محلی بودن مراجعات به دو دسته زیر تقسیم می‌شود:

۱- **محلی بودن فضایی (مکانی) (Spatial locality):** بیانگر این موضوع است؛ زمانی که یک مکان از حافظه مورد دسترسی قرار گرفت، احتمال زیادی وجود دارد که به زودی، مکان‌های نزدیک به آن مکان نیز مورد دسترسی قرار گیرند. مانند دسترسی پی‌درپی به داده‌های یک آرایه.

۲- **محلی بودن زمانی (Temporal locality):** بیانگر این موضوع است؛ زمانی که یک مکان از حافظه اخیراً مورد دسترسی قرار گرفته است، احتمال زیادی وجود دارد که به زودی، مجدداً مورد استفاده و دسترسی قرار گیرد. مانند اجرای مجموعه‌ای از دستورات عمل‌ها در یک حلقه تکرار. از خاصیت محلی بودن مراجعات، می‌توان در شکل‌دهی حافظه دو سطحی استفاده کرد.



حافظه سطح بالاتر (M1) کوچک‌تر، سریع‌تر و گران‌تر (به‌ازای هر بیت) از حافظه سطح دوم (M2) است. فرض کنید M1 حافظه نهان و M2 حافظه اصلی است. اگر بخواهیم یک کلمه را از حافظه اصلی (M2) بخوانیم؛ در مواقع مراجعه به حافظه، ابتدا در حافظه نهان (M1) برای دسترسی تلاش می‌شود. اگر این کلمه در حافظه نهان (M1) باشد، یعنی اصابت (Hit) صورت می‌گیرد و کلمه مورد نظر از حافظه نهان خوانده می‌شود. در غیر این صورت می‌گوییم، عدم اصابت (Miss) رخ داده است و یک بلوک (حاوی کلمه) از حافظه اصلی (M2) به حافظه نهان (M1) آورده شده و سپس دسترسی در حافظه نهان (M1) انجام می‌شود.

وقتی بلوکی به حافظه‌ی نهان (M1) آورده می‌شود، به دلیل محلی بودن مراجعات، تعدادی از دسترسی‌های بعدی در همان بلوک خواهد بود، که در مجموع موجب ارائه سرویس‌های سریع‌تر می‌گردد.

متوسط زمان دسترسی به حافظه از رابطه زیر به دست می‌آید:

$$T_a = H \times T_1 + (1 - H) \times (T_1 + T_2)$$

و یا به صورت ساده‌تر:

$$T_a = T_1 + (1 - H) \times T_2$$

که در آن:

T_a = متوسط زمان دسترسی به حافظه دو سطحی، T_1 = زمان دسترسی به M1 (حافظه نهان)، T_2 = زمان دسترسی به M2 (حافظه اصلی)،
 H = نسبت (احتمال) اصابت (کسری از دفعات که عنصر مورد مراجعه در M1 بوده است)

مثال ۴: یک سیستم حافظه دو سطحی را در نظر بگیرید که شامل یک پردازنده، یک حافظه اصلی و یک حافظه نهان است. اگر اطلاعات مورد درخواست در ۸۰٪ اوقات در حافظه نهان باشد و زمان دستیابی به حافظه نهان $2ns$ و زمان دستیابی به حافظه اصلی $100ns$ باشد، متوسط زمان دسترسی کدام گزینه است؟

- (۱) $23/9ns$ (۲) $22ns$ (۳) $21/9ns$ (۴) $24ns$

$$2ns = T_1 \quad 100ns = T_2$$

پاسخ: گزینه «۲»

$$0/8 = H$$

$$T_a = T_1 + (1 - H) \times T_2 = 2 + (1 - 0/8) \times 100 = 22ns$$

مثال ۵: در یک حافظه دو سطحی، فرض کنید $T_1 = 2ns$ و $T_2 = 100ns$ باشد. اگر T_s را متوسط زمان دسترسی به حافظه در نظر بگیریم و

پارامتر کارایی دسترسی را برابر $\frac{T_1}{T_2}$ تعریف کنیم، نرخ اصابت چقدر باشد که حداقل کارایی دسترسی برابر ۸۰٪ داشته باشیم؟

- (۱) $0/9825$ (۲) $0/9975$ (۳) $0/9875$ (۴) $0/9950$

$$T_s = T_1 + (1 - H) \times T_2$$

پاسخ: گزینه «۲»

در اینجا، می‌خواهیم H را به گونه‌ای به دست آوریم که $\frac{T_1}{T_2} \geq 0/8$ باشد:

$$\frac{T_1}{T_2} \geq 0/8 \Rightarrow T_1 \geq 0/8 \times (T_1 + (1 - H) \times T_2) \Rightarrow 0/2T_1 \geq 0/8 \times (1 - H) \times T_2 \Rightarrow H \geq 1 - \frac{T_1}{4 \times T_2} \Rightarrow H \geq 1 - \frac{1}{4 \times 100} \Rightarrow \boxed{H \geq 0/9975}$$

۸-۱ ساختار ورودی/خروجی

حافظه تنها یکی از انواع دستگاه‌های متعدد ورودی/خروجی (I/O) در یک کامپیوتر است. بخش اعظمی از کُد سیستم عامل به خاطر اهمیت قابل اطمینان و کارایی یک سیستم و به خاطر تنوع ذاتی دستگاه‌ها به مدیریت ورودی/خروجی اختصاص داده می‌شود.

یک سیستم کامپیوتر همه منظوره شامل پردازنده‌ها و کنترلر دستگاه‌های متعدد است که از طریق یک گذرگاه مشترک به یکدیگر متصل می‌شوند. هر یک از کنترلر دستگاه‌ها، مسئول یک نوع دستگاه خاص است. بسته به کنترلر امکان دارد که بیش از یک دستگاه به آن متصل شود.

برای نمونه، تعداد هفت دستگاه یا بیش‌تر می‌توانند به کنترلر اسکازی (رابط کوچک سیستم‌های کامپیوتری) (Small Computer Systems Interface (SCSI)) متصل شوند.