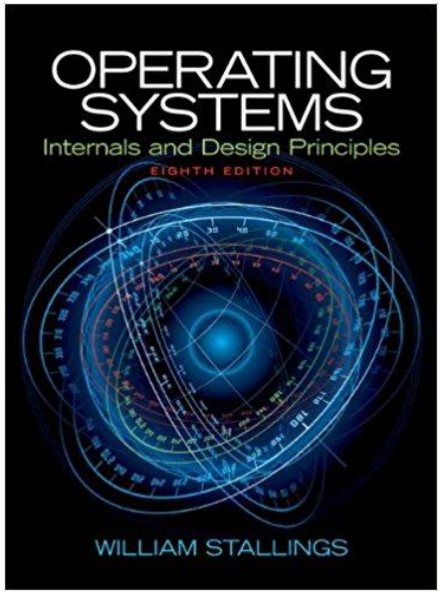
به نام خدا

درس سیستم های عامل

مرجع درس

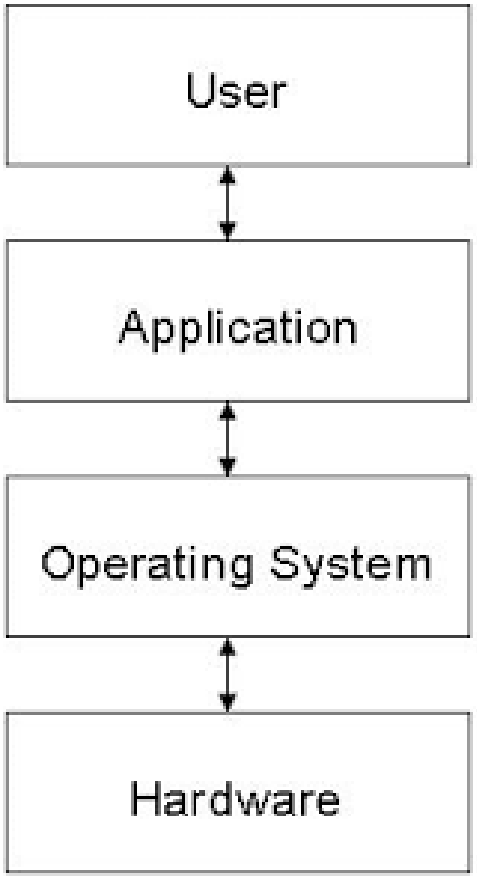
- **Operating Systems: Internals and Design Principles**
- *By: William Stallings*
- *8th edition*

- **سیستم های عامل** (ویرایش ششم)
- نویسنده: ویلیام استالینگز
- ترجمه: دکتر حسین پدرام، دکتر محسن صدیقی



سیستم عامل

- سیستم عامل، یک برنامه است که به صورت یک واسطه بین کاربر کامپیوتر و سخت افزار کامپیوتر عمل می کند.



طرح درس

پیش زمینه
- نگاه کلی به سخت افزار کامپیوتر
- نگاه کلی به سیستم عامل

فرآیندها
- شرح و کنترل فرآیندها
- نخ ها، چندپردازشی متقارن و ریزهسته ها
- همزمانی: انحصار متقابل و همگام سازی
- همزمانی: بن بست و گرسنگی

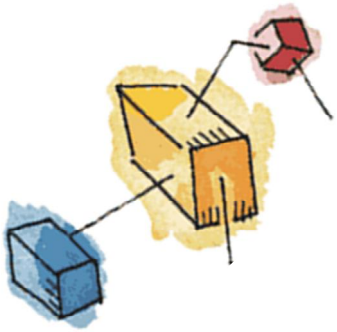
حافظه
- مدیریت حافظه
- حافظه مجازی

زمانبندی
- زمانبندی پردازنده

به نام خدا

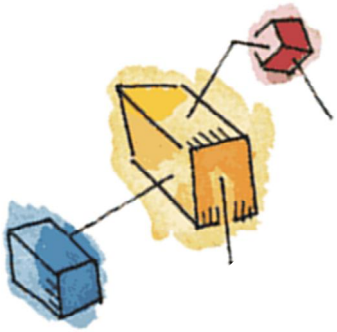
فصل اول:
نگاهی بر سیستم کامپیوتری

Computer System Overview



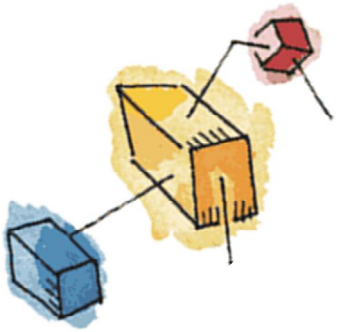
سیستم عامل

- سیستم عامل از منابع سخت افزاری یک یا چند پردازنده برای ارائه خدمات به کاربران استفاده می کند.
- مجموعه ای از سرویس ها را به کاربران سیستم ارائه می کند.
- حافظه ثانویه و I/O را از طرف کاربران مدیریت می کند.



سیستم عامل

- برای بررسی سیستم عامل، داشتن درک مناسبی از سخت افزار کامپیوتر ضروری است.
- در این فصل به معرفی اجمالی سخت افزار کامپیوتر پرداخته می شود.



عناصر اصلی سخت افزار

1. پردازنده

- کنترل و پردازش داده ها را به عهده دارد.

2. حافظه اصلی

- ذخیره و نگهداری داده ها و برنامه ها را به عهده دارد.
- فرار و ناپایدار است.
- به عنوان حافظه اولیه (primary memory) محسوب می شود.

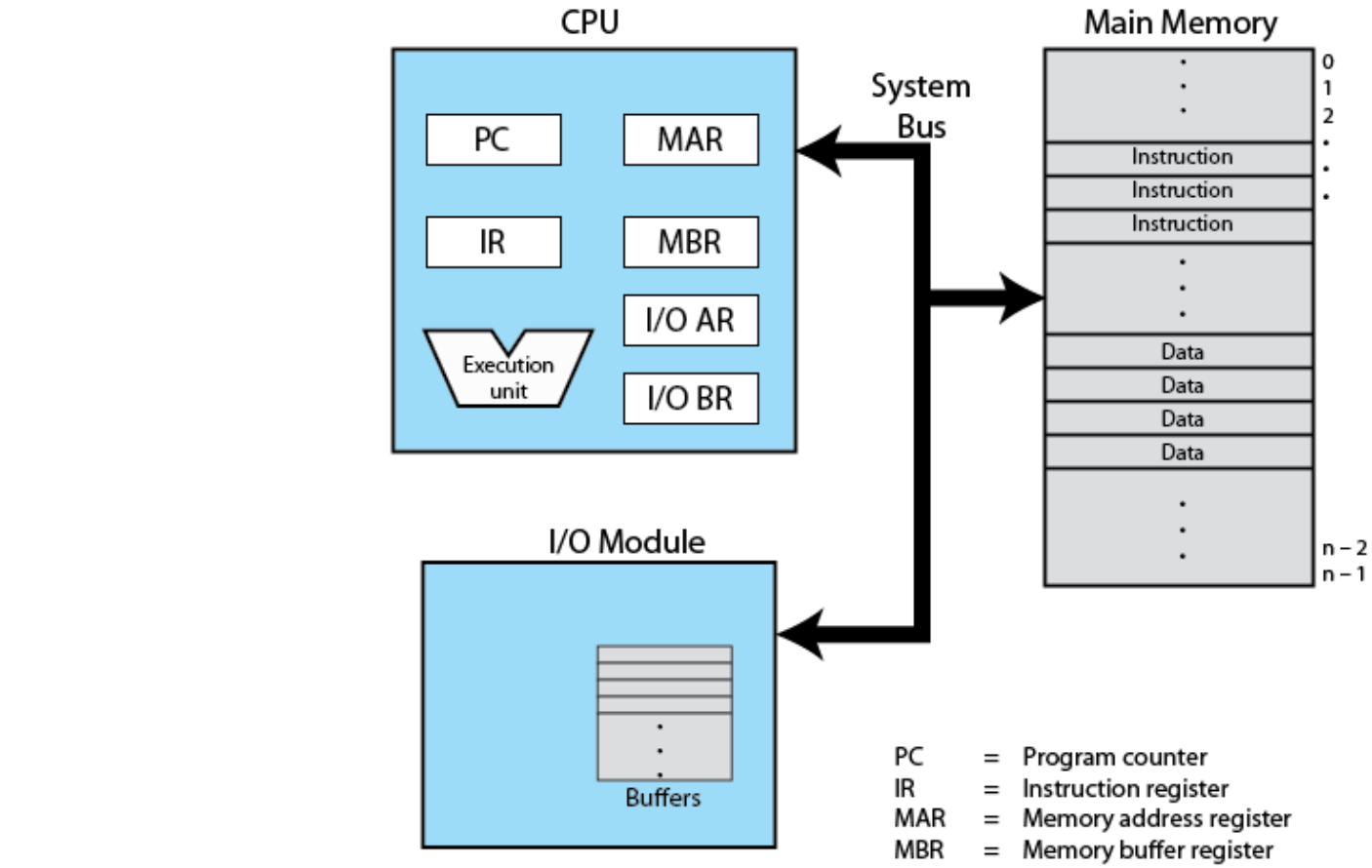
3. ماژول های ورودی/خروجی (I/O Modules)

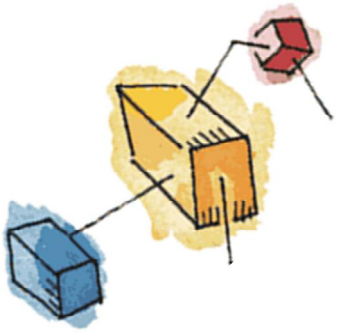
- انتقال داده ها میان کامپیوتر و دنیای خارج.
- دستگاه های حافظه جانبی، تجهیزات ارتباطی، ترمینال ها

4. گذرگاه سیستم (System bus)

- برقراری ارتباط بین پردازنده ها، حافظه اصلی و دستگاه های I/O

نمایی سطح بالا از اجزاء کامپیوتر

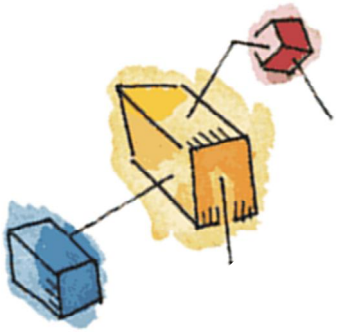




واحد پردازش مرکزی (CPU)

- برای تبادل اطلاعات بین پردازنده و حافظه اصلی
  - **Memory address register (MAR)**
    - آدرس محل بعدی برای عملیات خواندن یا نوشتن را مشخص می کند.
  - **Memory buffer register (MBR)**
    - حاوی داده ای است که قرار است در حافظه نوشته شود، و یا دریافت کنندهٔ داده ای است که از حافظه خوانده می شود.

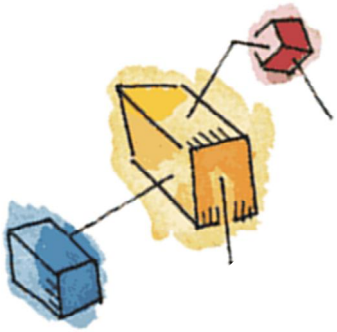
- برای تبادل داده ها بین **CPU** و **I/O**
  - **I/O Address Register (I/O AR)**
  - **I/O Buffer Register (I/O BR)**



انواع ثبات های پردازنده

- **ثبات های قابل رویت برای کاربر**
  - این امکان را به برنامه نویس زبان ماشین می دهند تا با استفاده بهینه از این ثبات ها میزان مراجعات به حافظه اصلی را به حداقل برساند.
  - می توانند توسط دستورات زبان ماشین مورد ارجاع و استفاده قرار گیرند.
  - برای تمامی برنامه (برنامه های سیستمی و برنامه های کاربردی) در دسترس اند.

- **ثبات های کنترل وضعیت** (غیرقابل رویت)
  - توسط پردازنده و برای کنترل عملیات پردازنده استفاده می شوند.
  - توسط روتین های ممتاز سیستم عامل و برای کنترل اجرای برنامه ها به کار گرفته می شوند.



ثبات های قابل رؤیت برای کاربر

1. ثبات های داده

- برای نگهداری داده ها و برای انتقال داده ها بین توابع (این ثبات ها همه منظوره اند)

2. ثبات های آدرس

- حاوی آدرس داده ها و دستور العمل ها در حافظه (همه منظوره یا تک منظوره)

– **ثبات شاخص** (Index register)

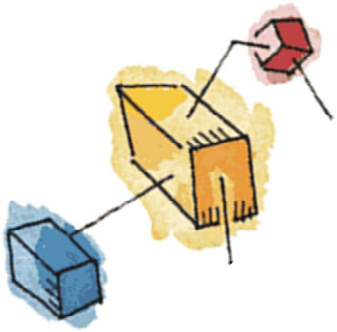
- در آدرس دهی شاخص استفاده می شود. با یک مقدار پایه جمع می شود تا آدرس موثر به دست آید.
- (آدرس شاخص + آدرس پایه = آدرس موثر)

– **اشاره گر قطعه** (Segment pointer)

- در آدرس دهی قطعه بندی، حافظه به قطعه هایی تقسیم شده و توسط یک قطعه (آدرس پایه) و یک offset مورد ارجاع قرار می گیرد.
- در این شرایط، از این ثبات برای نگهداری آدرس پایه (آدرس شروع) استفاده می شود.

– **اشاره گر پشته** (Stack pointer)

- برای اشاره به بالای پشته به کار می رود.



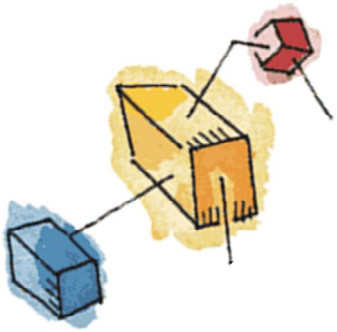
ثبات های کنترل وضعیت

- **شمارنده برنامه** (Program Counter) **PC**:
  - برای نگهداری آدرس دستور العمل بعدی که باید واکشی شود، به کار می رود.

- **ثبات دستورالعمل** (Instruction Register) **IR** :
  - برای ذخیره دستورالعمل فعلی که در حال اجرا است به کار می رود.

- **کلمه وضعیت برنامه**(Program Status Word)**PSW** :
  - یک یا چند ثبات که حاوی اطلاعات وضعیت می باشد.

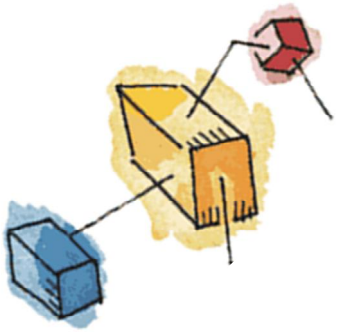
- **کدهای شرطی یا پرچم ها** (flags):
  - بیت هایی که در اثر نتیجه عملیات، توسط سخت افزار پردازنده مقداردهی می شوند.
  - مثلا: نتیجه مثبت، منفی، صفر یا overflow



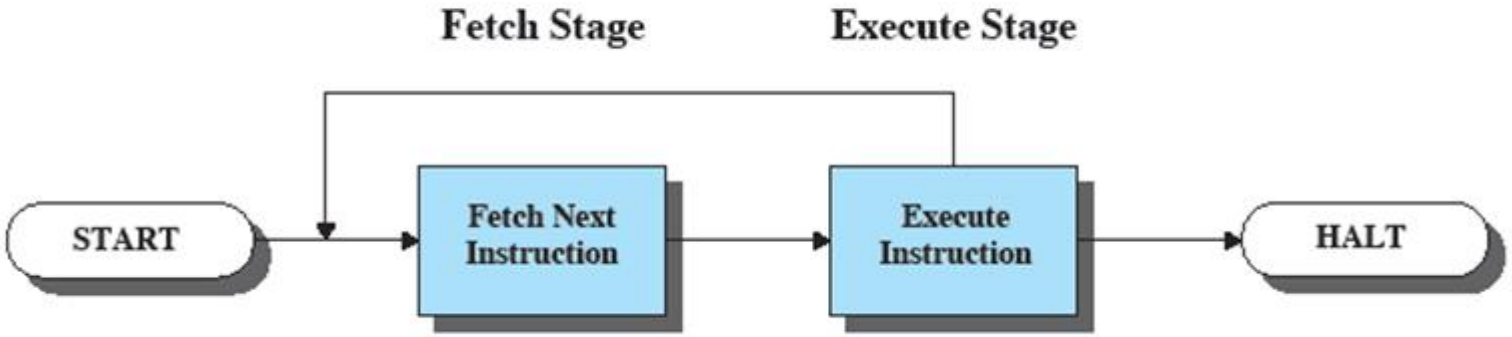
پردازش دستور العمل

- هر برنامه که اجرا می شود مجموعه ای از دستور العمل ها است که در حافظه ذخیره می شوند.

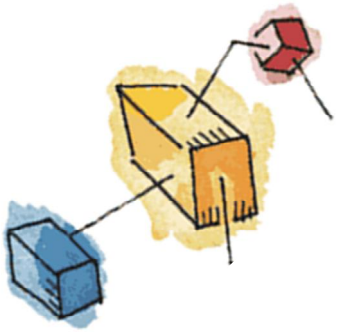
- پردازش دستور العمل دو گام دارد:
  - چرخه واکشی: پردازنده، دستورالعمل را از حافظه می خواند (واکشی می کند).
  - چرخه اجرا: پردازنده، دستور العمل واکشی شده را اجرا می کند.



چرخه اصلی دستورالعمل

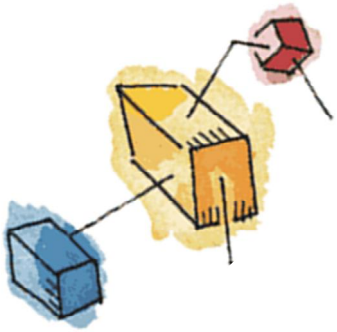


- پایان چرخه تنها در موارد زیر رخ می دهد:
  - خاموش شدن کامپیوتر
  - رخ دادن خطای غیر قابل جبران
  - رسیدن به فرمان توقف



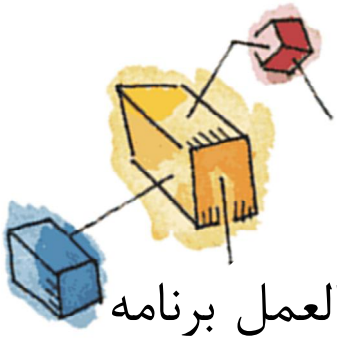
چگونگی واکشی و اجرای دستورالعمل

- در ابتدای هر چرخه، پردازنده دستورالعمل را از حافظه واکشی می کند.
- شمارنده برنامه (ثبات PC)، آدرس دستورالعمل بعدی را که می خواهد واکشی شود در خود نگه می دارد.
- پس از واکشی یک واحد به مقدار PC می افزاید تا دستور بعدی به ترتیب واکشی شود.



وقفه ها (Interrupt)

- وقفه علامتی است که از طرف یک منبع خارجی به پردازنده داده می شود و موجب توقف برنامه فعلی و توقف روال کار عادی پردازنده می شود.
- وقفه ها برای افزایش کارایی پردازنده استفاده می شوند.
- وقفه ها به پردازنده اجازه می دهند تا در حین اجرای عملیات I/O به اجرای دستورالعمل دیگری بپردازد.
- اکثر دستگاه های I/O کندتر از پردازنده هستند.
  - پردازنده باید متوقف شود و برای این دستگاه ها منتظر بماند



دسته های وقفه

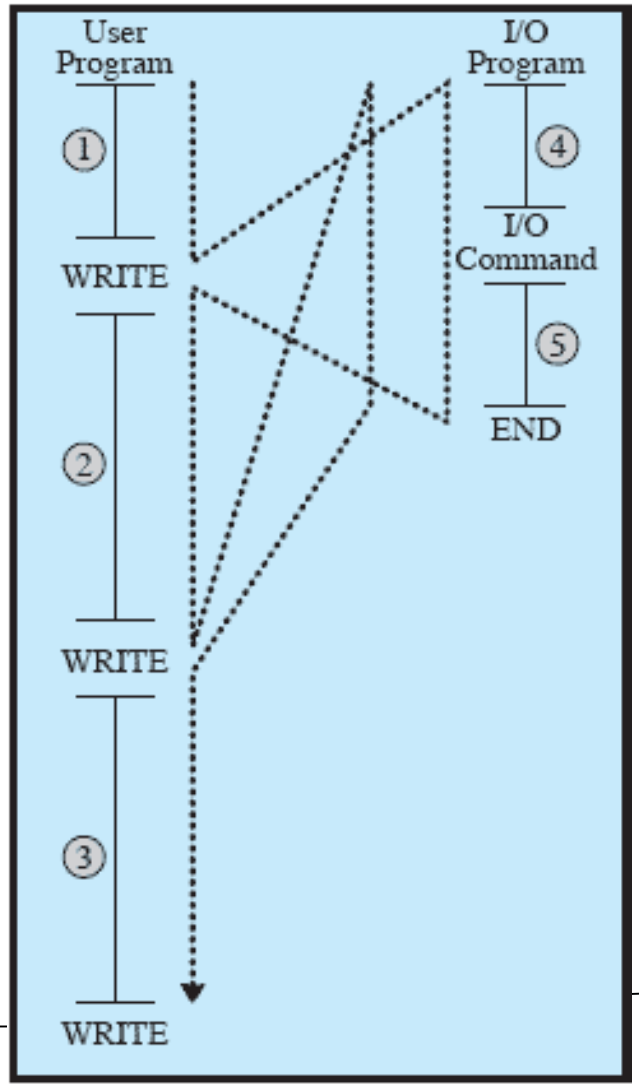
- **برنامه**: وقفه هایی که به دلیل بعضی شرایط حاصل از اجرای یک دستورالعمل برنامه بروز می کنند.
  - مانند: سرریز شدن محاسباتی، تقسیم بر صفر، تلاش برای اجرای یک عمل غیر مجاز، مراجعه به آدرس خارج از فضای مجاز

- **زمان سنج (Timer)**: وقفه هایی که توسط زمان سنج داخلی پردازنده تولید می شوند و به سیستم عامل اجازه می دهد بعضی عملیات را به صورت مرتب و دوره ای انجام دهد.

- **ورودی خروجی**: وقفه ای که توسط کنترل کننده I/O تولید می شوند.

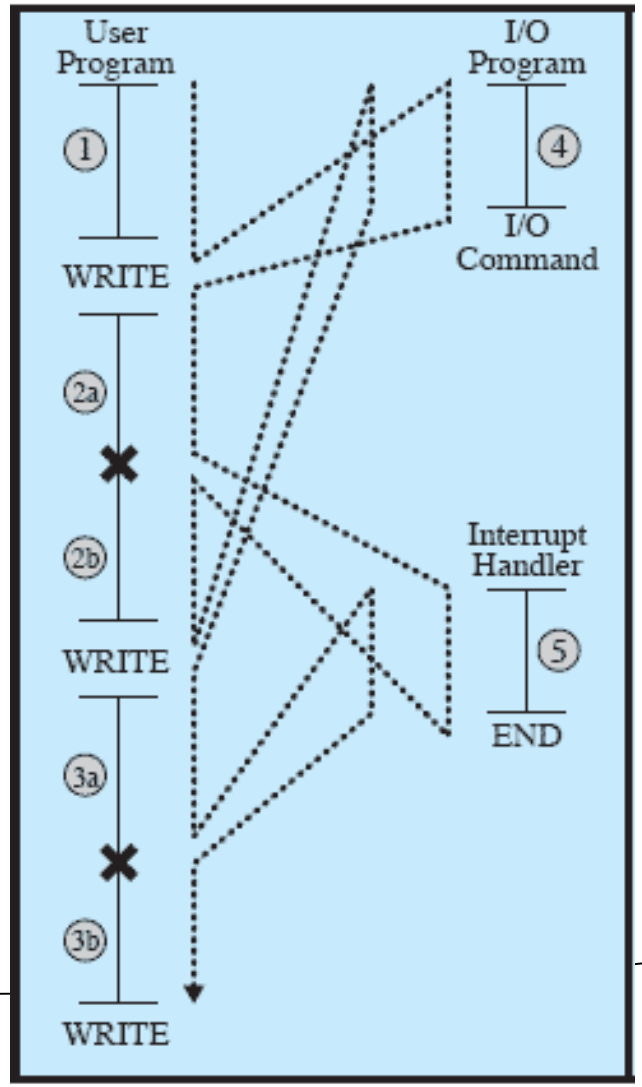
- **نقص سخت افزار**: با نقص سخت افزار تولید می شود. مانند خطای توازن حافظه

جریان کنترل برنامه بدون وقفه

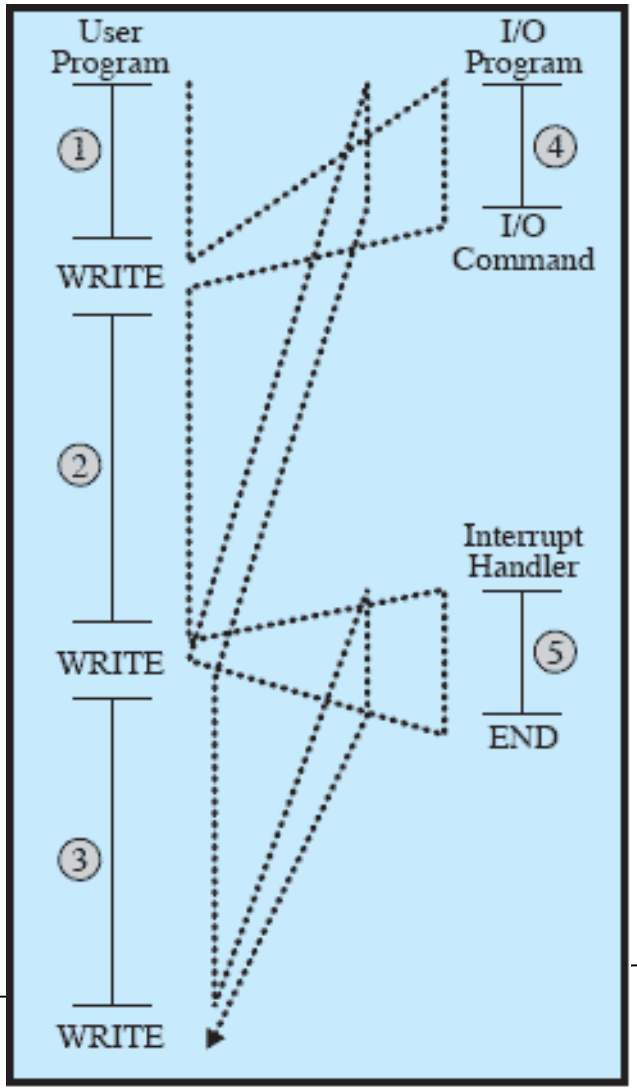


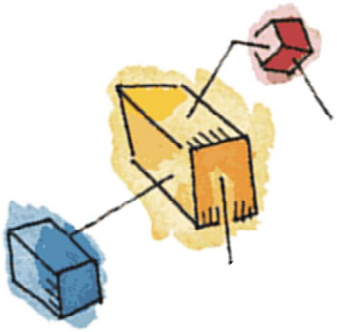
(a) No interrupts

جریان کنترل برنامه با وقفه (ورودی-خروجی کوتاه مدت)



جریان کنترل برنامه با وقفه (ورودی-خروجی بلند مدت)





گرداننده وقفه

- **چرخه دستورالعمل با وقفه:**
- در پایان چرخه اجرای دستورالعمل، پردازنده وجود وقفه را بررسی می کند.
- درصورتیکه وقفه مطرح باشد:
  - پردازنده اجرای برنامه را مسکوت و معلق می گذارد
  - روال خدماتی وقفه یا برنامه گرداننده وقفه مربوطه را اجرا می کند.

انتقال کنترل از طریق وقفه

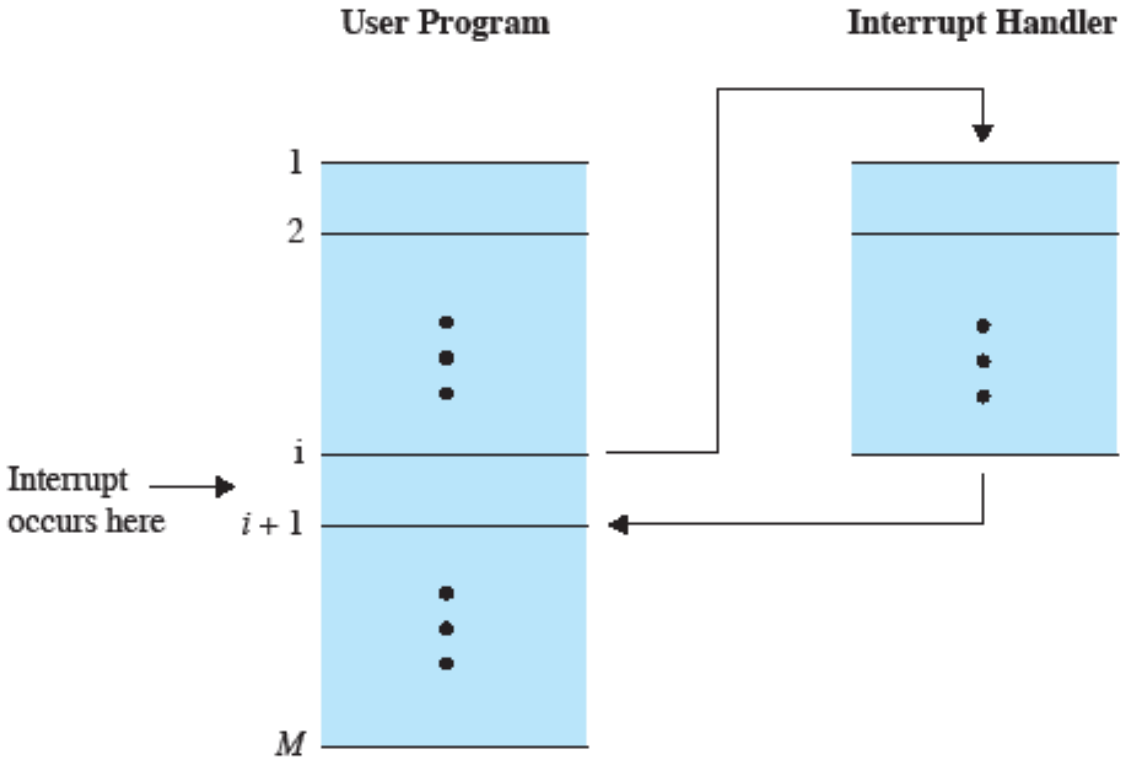
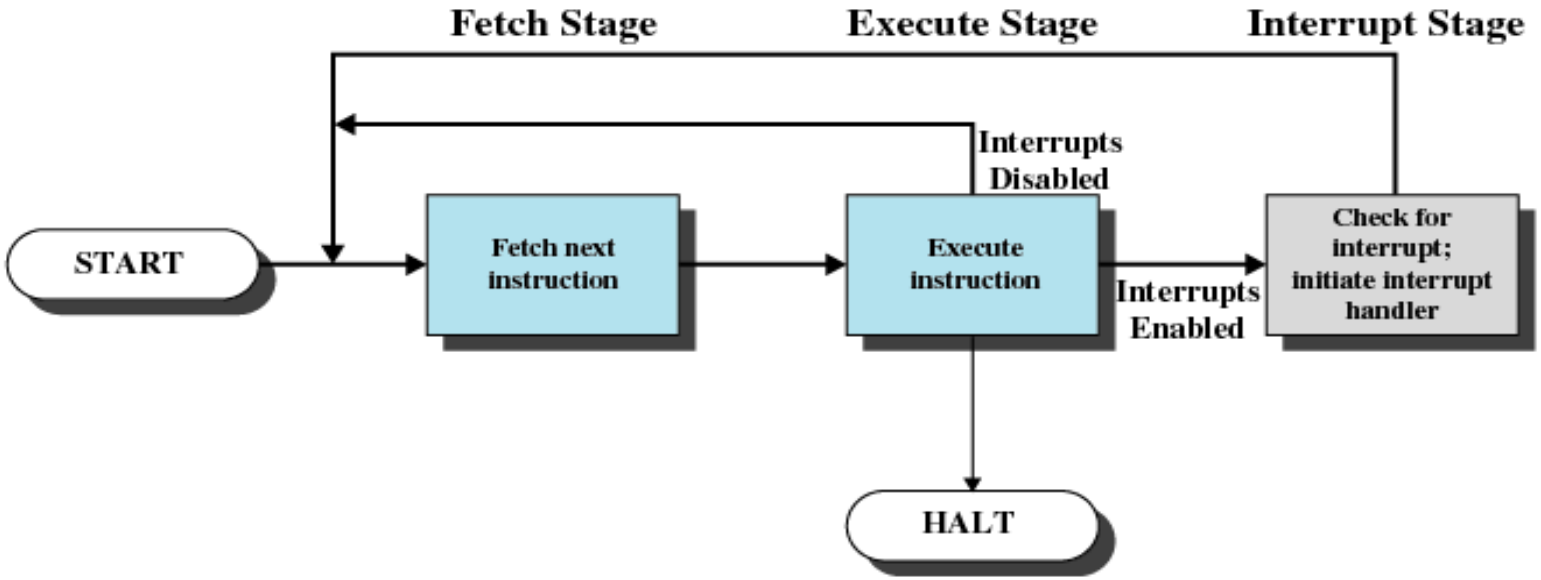
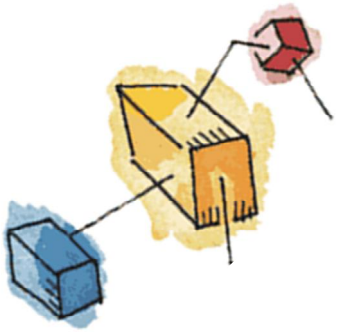


Figure 1.6 Transfer of Control via Interrupts

چرخه دستورالعمل با وقفه:





وقفه های چند گانه

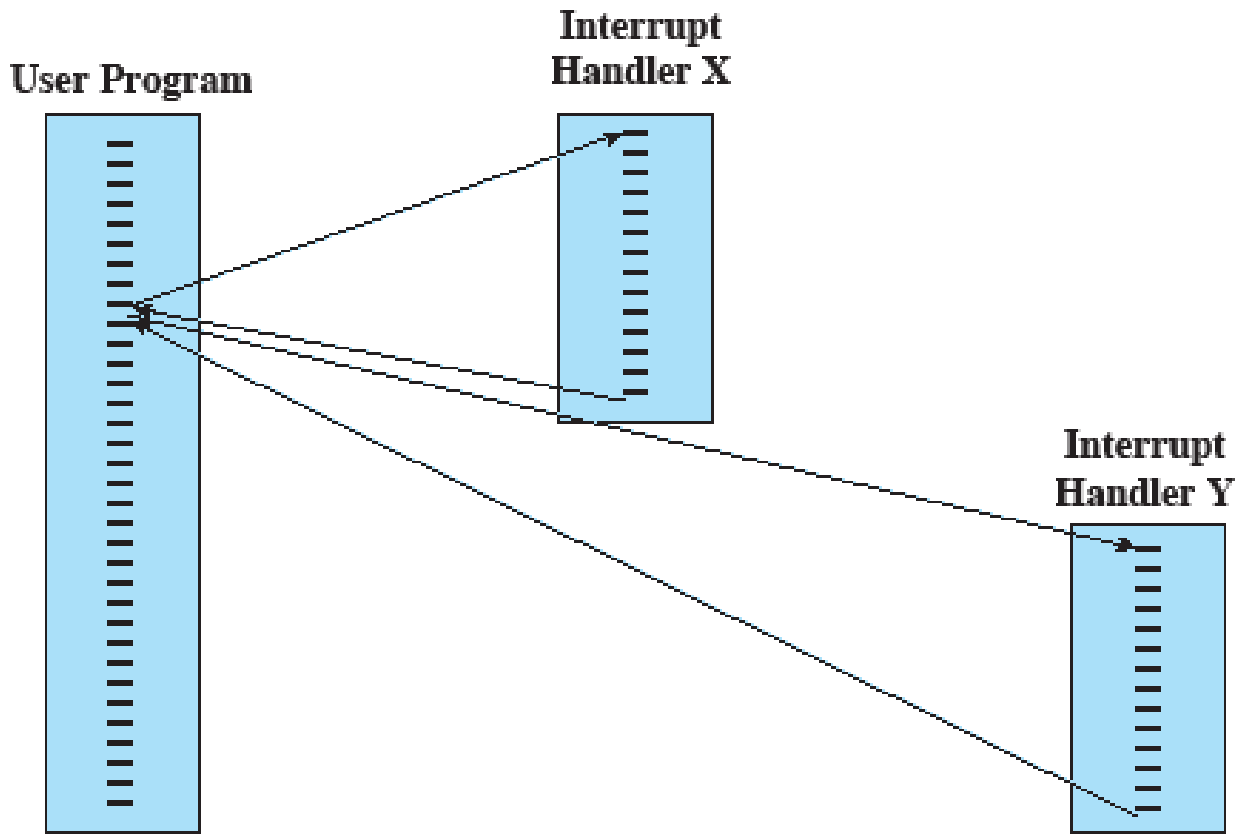
- بروز چندین وقفه با هم را وقفه چندگانه می گویند.
- در برخورد با وقفه های چند گانه دو رویکرد می تواند اتخاذ شود.

1. **غیر فعال کردن سایر وقفه ها در هنگام پردازش یک وقفه**
   - یعنی پردازنده می خواهد و می تواند سیگنال وقفه را نادیده بگیرد.
   - همه وقفه ها به طور ترتیبی قبل از اجرای برنامه کاربر توسط ریزپردازنده پردازش می شوند.
   - نکته منفی این رویکرد : اولویت نسبی یا محدودیت های زمانی برای وقفه ها نادیده گرفته می شوند.

2. **وقفه های تو در تو با تعریف اولویت نسبی برای وقفه ها**
   - پردازنده مجاز است برای پردازش وقفه با اولویت بالاتر وقفه با اولویت پایین تر را نادیده بگیرد.

انتقال کنترل با وقفه های چند گانه

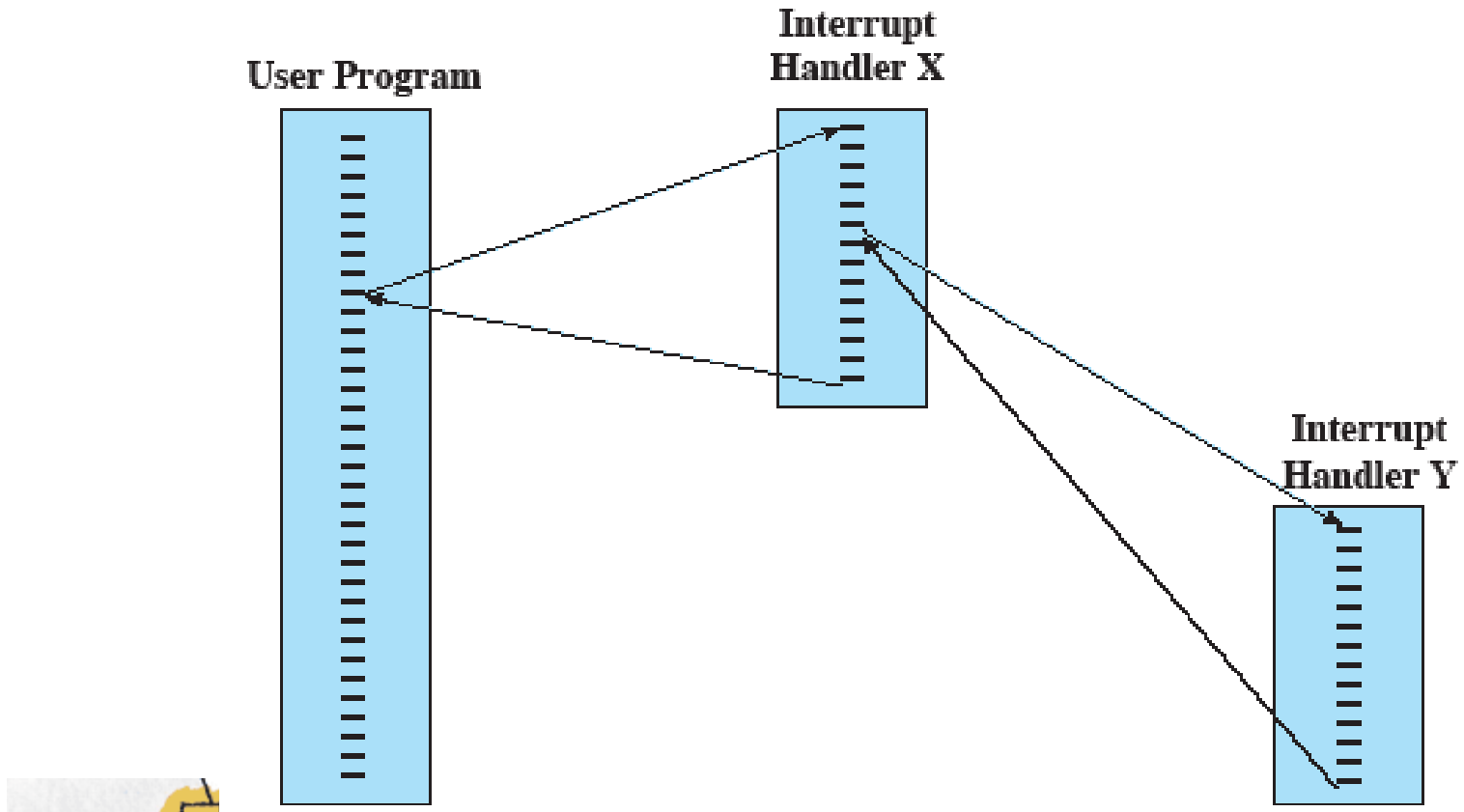
رویکرد اول : غیر فعال کردن سایر وقفه ها



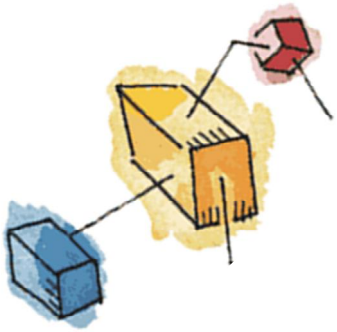
(a) Sequential interrupt processing

انتقال کنترل با وقفه های چند گانه

رویکرد دوم : اولویت بندی وقفه ها (وقفه های تو در تو)

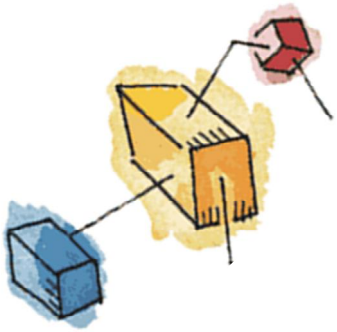


(b) Nested interrupt processing



چندبرنامگی (Multiprogramming)

- پردازنده بیش از یک برنامه برای اجرا دارد.
- ترتیب اجرای برنامه ها به اولویت نسبی آنها و میزان انتظارشان برای I/O وابسته است.
- هنگامی که برنامه با وقفه مواجه می شود، پردازنده کنترل را به برنامه گرداننده وقفه انتقال می دهد. پس از تکمیل برنامه گرداننده وقفه ممکن است کنترل بلافاصله به برنامه قبلی باز نگردد.



سلسله مراتب حافظه

- برای طراحی حافظه به سه مورد باید توجه شود:
  - اندازه (ظرفیت)
  - سرعت (زمان دسترسی)
  - قیمت (هزینه)

سلسله مراتب حافظه

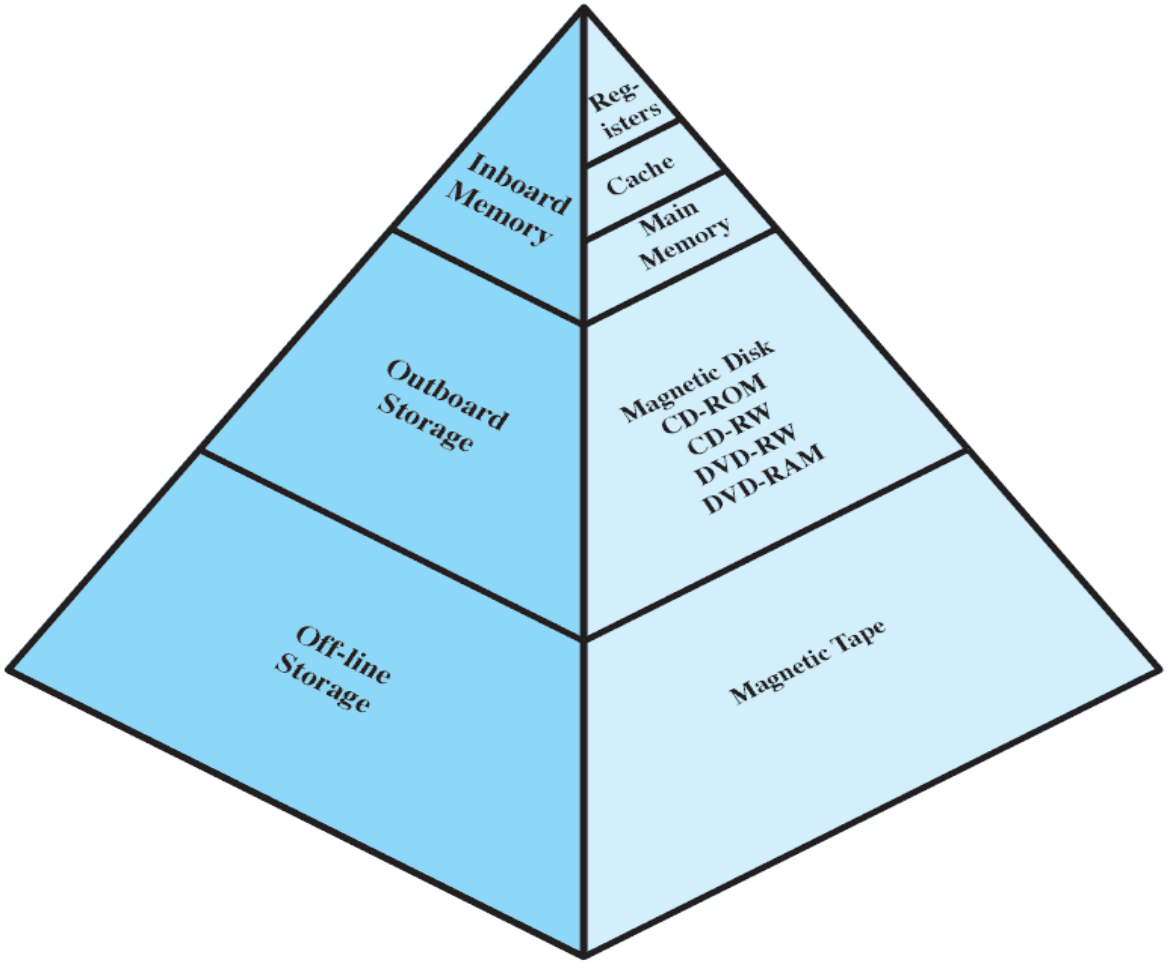
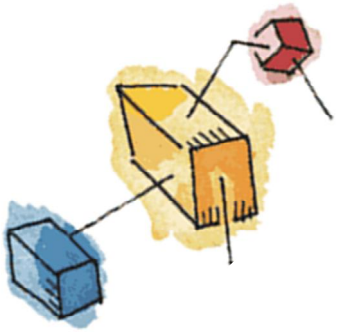


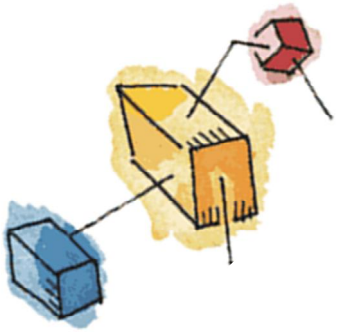
Figure 1.14 The Memory Hierarchy



پایین رفتن در سلسله مراتب حافظه

- کاهش هزینه
- افزایش ظرفیت
- افزایش زمان دسترسی (دستیابی کندتر به داده ها)
- کاهش فرکانس مراجعات پردازنده به حافظه توسط پردازنده

- هدف از این سلسله مراتب کاهش تعداد دفعات دسترسی پردازنده به حافظه های پایین تر هرم است.



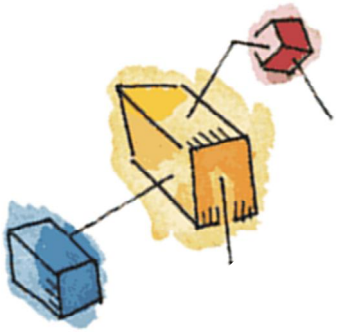
سلسله مراتب حافظه

- **ثباتها :** کوچکترین، گران ترین، و سریعترین نوع حافظه اند.

- **راه های توسعه حافظه اصلی:**
  - از نظر سرعت : استفاده از حافظه نهان
  - از نظر ظرفیت : استفاده از حافظه جانبی

- **حافظه نهان(cache):** تنها برای پردازنده قابل رؤیت بوده و برای تبادل داده ها بین حافظه اصلی و ثبات ها مورد استفاده قرار می گیرد.

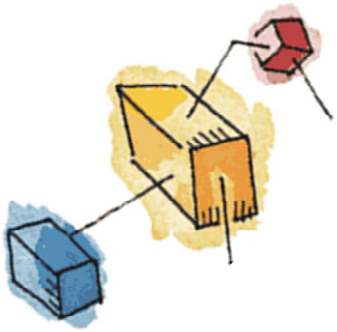
- **حافظه جانبی (ثانویه)**
  - حافظه ای خارجی
  - غیر فرار
  - مورد استفاده برای ذخیره فایل های داده و برنامه



حافظه نهان (Cache)

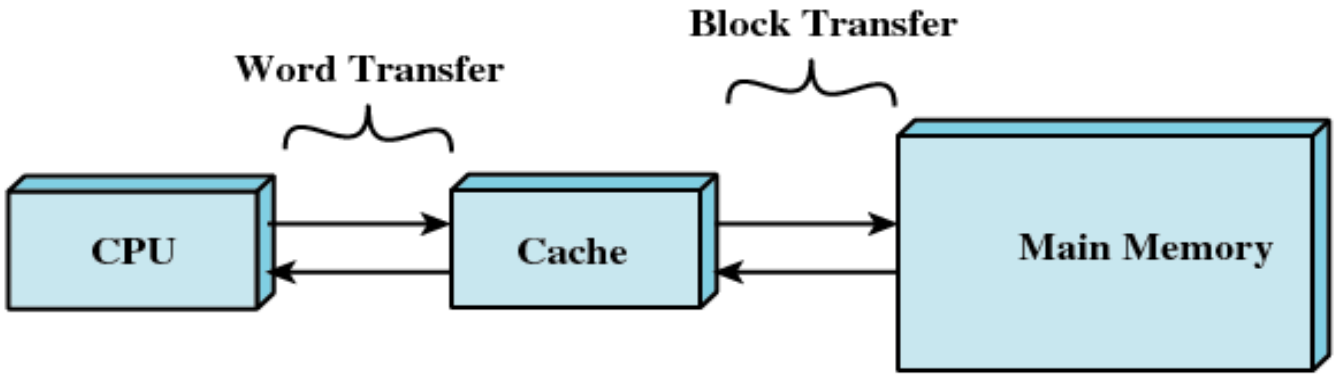
- سرعت پردازنده بیشتر از سرعت مراجعه به حافظه است.

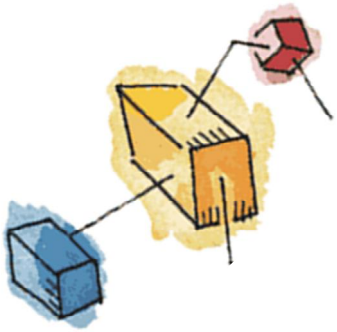
- برای مدیریت این موضوع، از اصل محلی بودن مراجعات و همچنین یک حافظه کوچک سریع استفاده می شود.



حافظه نهان (Cache)

- حافظه نهان برای سیستم عامل قابل مشاهده نیست.
- حافظه نهان به دلیل عدم تطابق سرعت حافظه اصلی با سرعت پردازنده طراحی شد و سرعت حافظه اصلی را افزایش می دهد.

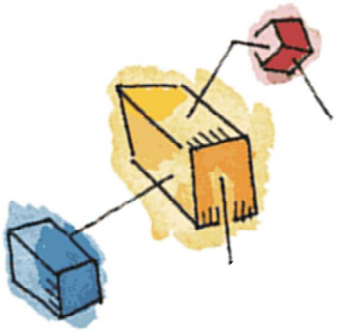




حافظه نهان (Cache)

- حافظه نهان حاوی کپی بخشی از محتوای حافظه اصلی است.
- پردازنده ابتدا حافظه نهان را بررسی می کند.
- اگر در حافظه نهان پیدا نشد بلاکی از حافظه اصلی که حاوی داده های مورد نیاز است به حافظه نهان آورده می شود.

- به علت محلی بودن ارجاعات، احتمال دارد که مراجعات آتی به حافظه نیز در همان بلاک باشند.



- حافظه اصلی به M بلاک K کلمه ای تقسیم می شود و حافظه نهان به C شکاف K کلمه ای.
- از آنجا که M>>C بنابراین هیچ بلاکی نمی تواند برای مدت طولانی در حافظه نهان بماند.

- اصابت به معنای پیدا کردن اطلاعات در حافظه نهان است.

ساختار حافظه اصلی و حافظه نهان

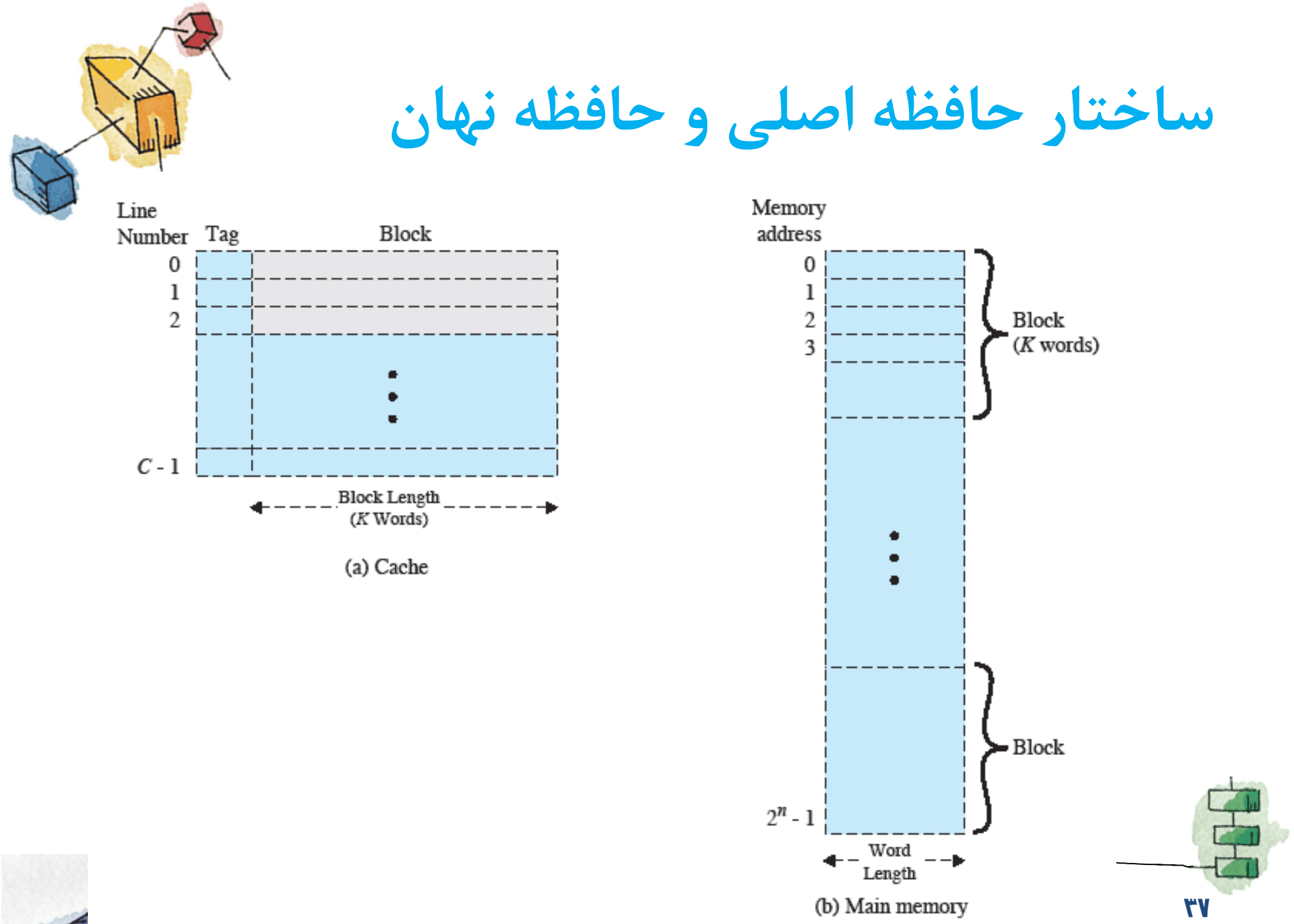
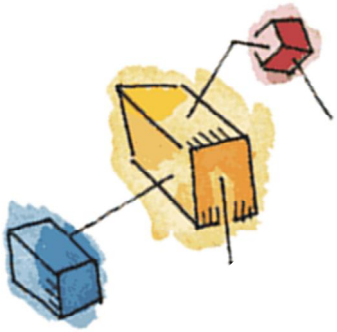


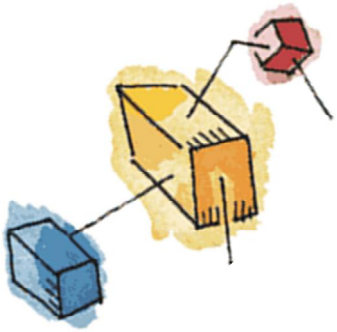
Figure 1.17 Cache/Main-Memory Structure



حافظه نهان (Cache)

- هر گاه پردازنده به داده ها نیاز دارد ابتدا آنها را در حافظه نهان جستجو نموده و اگر آنها را یافت بازیابی شان می کند.
  - در این حالت اصطلاحا گفته می شود اصابت یا Hit رخ داده است.
- در غیر این صورت آن ها را از حافظه اصلی می خواند.
  - در این حالت اصطلاحا گفته می شود عدم اصابت یا Miss رخ داده است.

- بنابراین در شرایط عدم اصابت، زمان صرف شده برای دسترسی به داده مورد نظر، برابر با مجموع زمان دسترسی به حافظه نهان و حافظه اصلی خواهد بود.

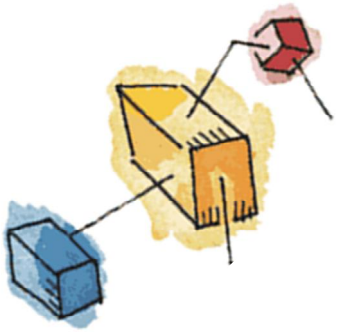


زمان دسترسی موثر

- زمان موثر دسترسی به یک کلمه از حافظه در سیستمی با حافظه دو سطحی که سطح اول حافظه نهان و سطح دوم حافظه اصلی باشد، با فرمول زیر قابل محاسبه خواهد بود:

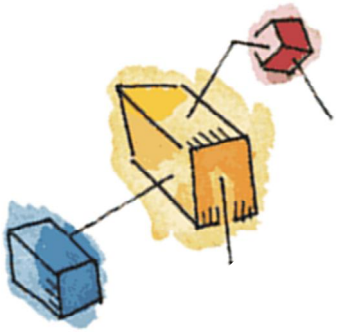
$$T_e = H \times T_c + [(1-H) \times (T_c + T_m)]$$

T_e: زمان دسترسی موثر
T_c: متوسط زمان دسترسی به حافظه نهان
T_m: متوسط زمان دسترسی به حافظه اصلی
H: نرخ اصابت



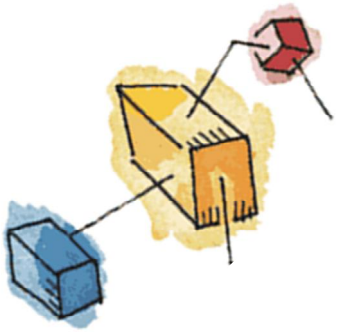
طراحی حافظه نهان

- **تابع نگاشت:**
  - محل بلاک جدید وارد شده به حافظه نهان را تعیین می کند.
- **الگوریتم تعویض:**
  - بلاکی که باید از حافظه نهان حذف شود تا بلاک جدید جایگزین شود را تعیین می کند.
  - معمولا از الگوریتم LRU(Least Recently Used) استفاده می شود.



طراحی حافظه نهان

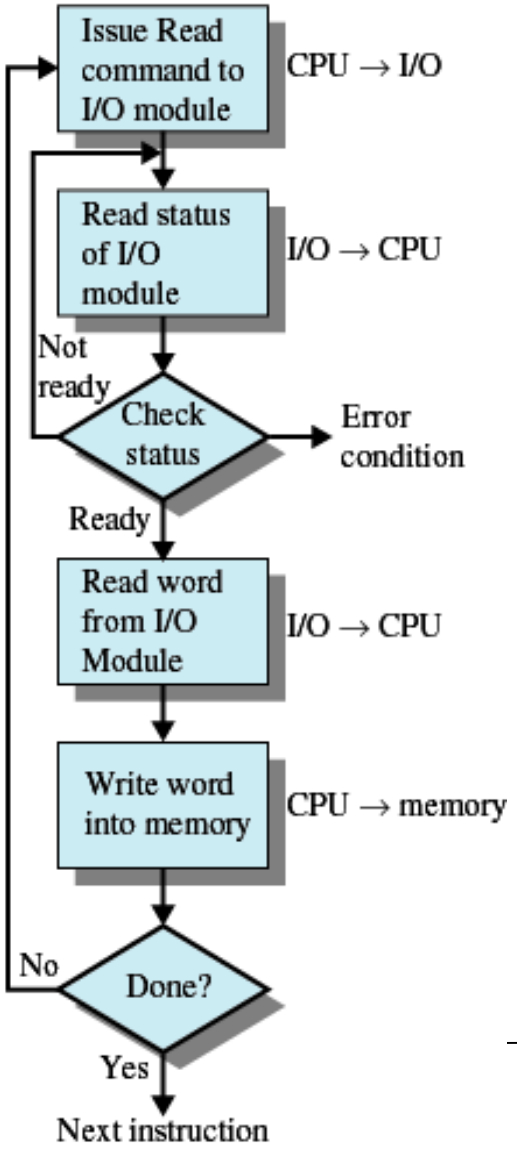
- **سیاست نوشتن:**
  - تعیین می کند چه زمانی بلاک موجود در حافظه نهان در حافظه اصلی نوشته شود.
    - زمانی که محتوای بلاک موجود در حافظه نهان تغییر کند.
    - هنگام جایگزینی آن روی حافظه اصلی
  - میزان نوشتن را حداقل می کند.
  - ولی باعث متروک ماندن حافظه می شود.



روش های انتقال ورودی/خروجی

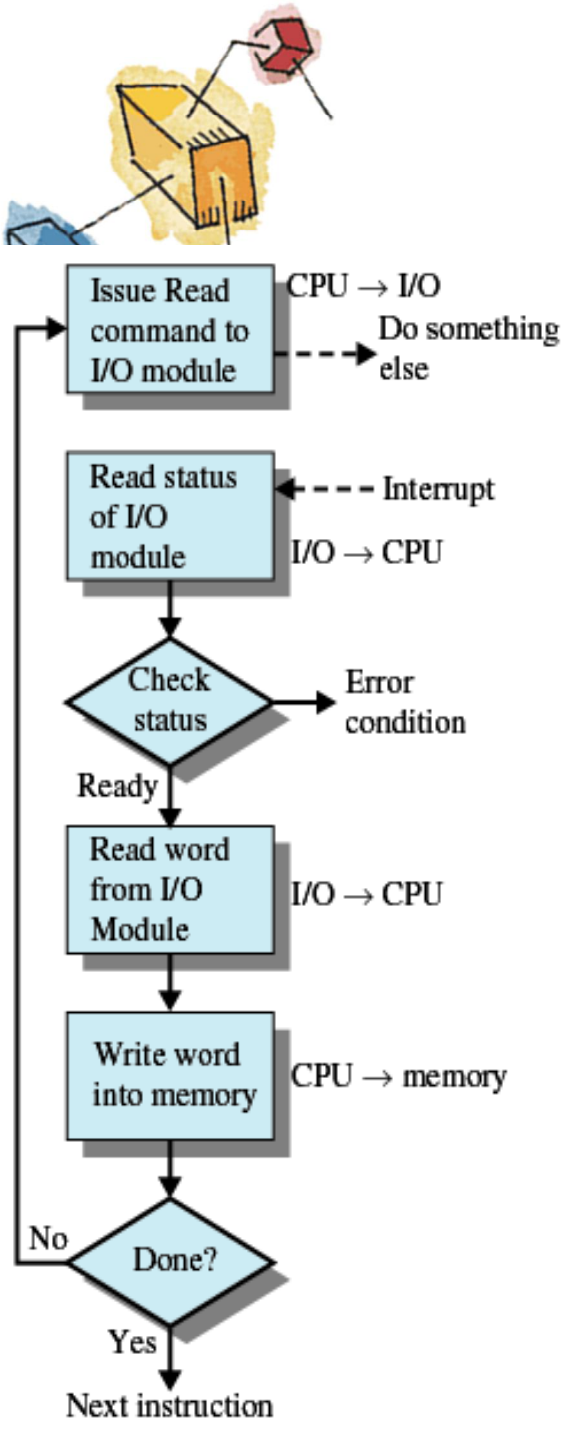
- ورودی/خروجی برنامه سازی شده (Programmed I/O)
- ورودی/خروجی مبتنی بر وقفه (Interrupt-Driven I/O)
- دسترسی مستقیم به حافظه (Direct Memory Access) یا DMA

ورودی / خروجی برنامه سازی شده

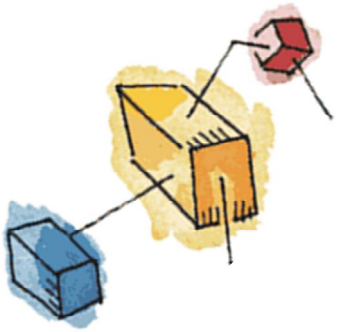


- هنگام مواجهه با یک عمل I/O در حین اجرای برنامه، CPU فرمان لازم را به مؤلفه I/O می فرستد.
- مؤلفه I/O فرمان را اجرا می کند.
- بیت های وضعیت را در ثبات وضعیت I/O تغییر می دهد.
- هیچ عمل دیگری انجام نمی دهد (بویژه وقفه نمی دهد)
- وظیفه CPU است که متناوباً وضعیت مؤلفه I/O را چک کند.
- در حین انجام عملیات توسط I/O، پردازنده مدام باید در چرخه تست وضعیت دستگاه باقی بماند. بنابراین کارایی سیستم بشدت پایین می آید.

ورودی/خروجی مبتنی بر وقفه

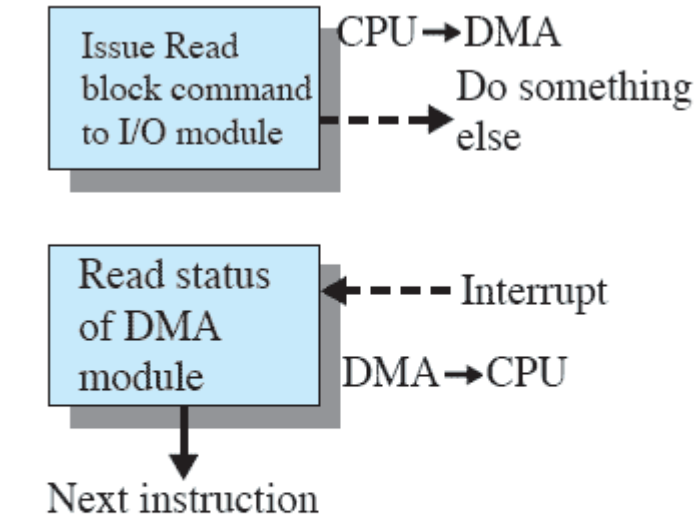


- در این روش CPU پس از صدور فرمان به I/O به کار دیگری مشغول می شود.
- وقتی I/O برای مبادله داده ها آماده شد وقفه ای به پردازنده صادر می کند.
- CPU اجرای دستور العمل فعلی را به پایان می رساند.
- CPU درخواست وقفه را شناسایی می کند.
- CPU مقدار PC و PSW را ذخیره می کند.
- CPU آدرس اولین دستور روال گرداننده وقفه را در PC کپی می کند.
- پس از اجرای روال گرداننده وقفه مقدار PC و PSW بازیابی می شود.
- CPU به اجرای برنامه قبلی خود باز می گردد.

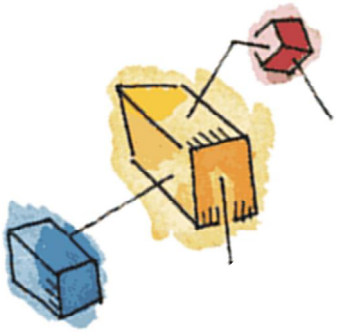


دسترسی مستقیم به حافظه (DMA)

- در این روش از مولفه ای به نام DMA استفاده می شود.
- هنگام خواندن یا نوشتن پردازنده دستورات زیر را به DMA صادر می کند و خودش کنار می رود.
  - عمل خواندن مورد نیاز است یا نوشتن
  - آدرس دستگاه I/O درگیر
  - محلی از حافظه که از آن محل خواندن یا نوشتن شروع می شود.
  - تعداد کلماتی که باید خوانده یا نوشته شوند.
- برای انتقال داده ها در این روش لازم است که گذرگاه در اختیار مولفه DMA قرار گیرد.
- پس از اتمام عملیات، مولفه DMA به پردازنده وقفه ای مبنی بر اتمام عملیات می دهد.
- در این روش پردازنده تنها در ابتدا و انتهای عملیات I/O درگیر این عملیات خواهد بود.
- DMA برای انتقال داده های بزرگ و چند کلمه ای، از دو روش دیگر کارآمد تر است.



(c) Direct memory access



پایان فصل اول