بنام خداوند جان و خرد

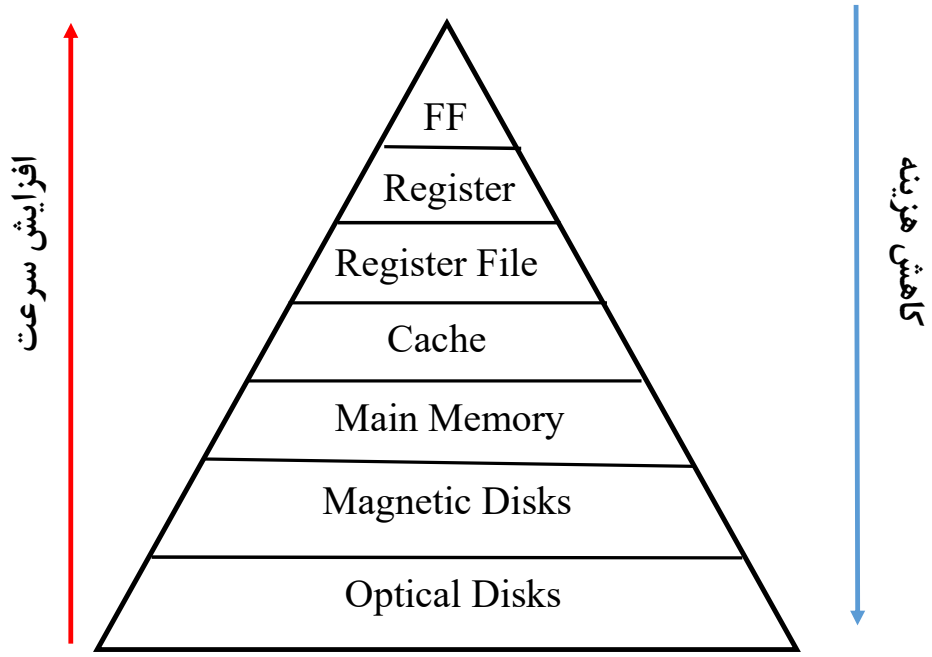
معماری کامپیوتر

جلسه ششم: سلسله مراتب حافظه

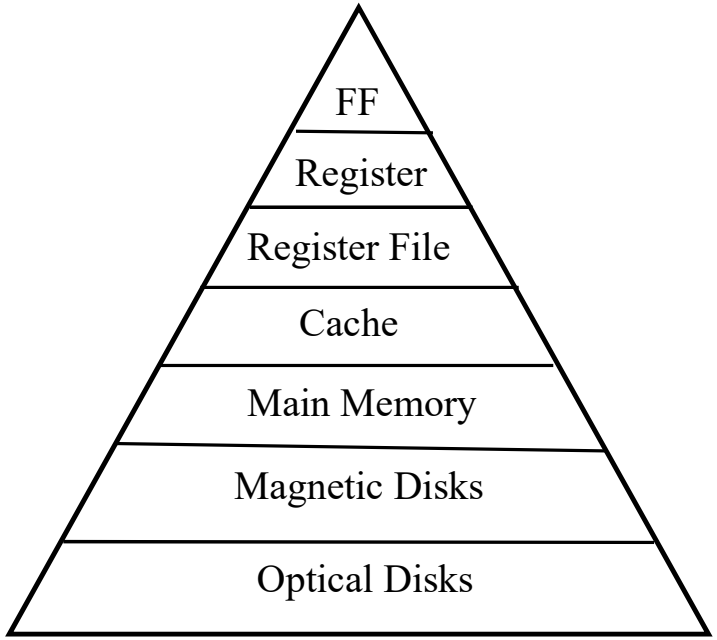
سلسله مراتب حافظه

- اهمیت پارامترهای کارایی و هزینه و در ساخت قطعات کامپیوتری از جمله حافظه‌ها
  - متضاد بودن این پارامترها و وجود trade-off مابین آن‌ها
  - هدف: حداقل هزینه همراه با بیشترین کارایی
- سلسله‌مراتب حافظه:
  - دسته‌بندی حافظه‌ها براساس این پارامترها و استفاده در کاربردهای مختلف برحسب نیاز

سلسله مراتب حافظه



سلسله مراتب حافظه



افزایش تاخیر در سطوح

D1<….<D6<D7

افزایش حجم در سطوح

C1<….<C6<C7

سلسله مراتب حافظه

- در بررسی حافظه‌ها چندین پارامتر حائز اهمیت تعریف می‌شوند:
  - Access Time
  - Cycle Time
  - Read Time
  - Write Time

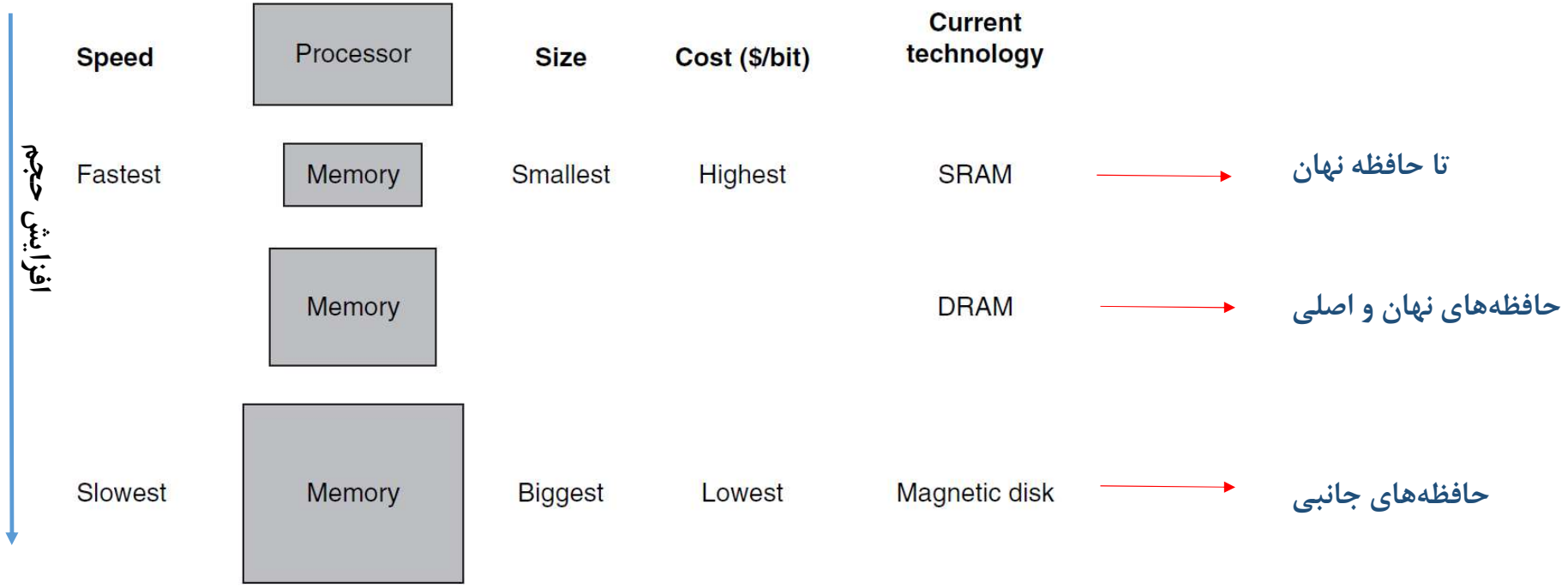
سلسله مراتب حافظه

- هدف اولیه: طراحی حافظه با حداقل قیمت و حداکثر سرعت
  - سرعت: دسترسی به اطلاعات
- ارضای قید سرعت: نگهداری اطلاعات در سطوح بالاتر مثلث سلسله‌مراتب
- ارضای قید هزینه: نگهداری اطلاعات در سطوح پایین‌تر مثلث سلسله‌مراتب

سلسله مراتب حافظه

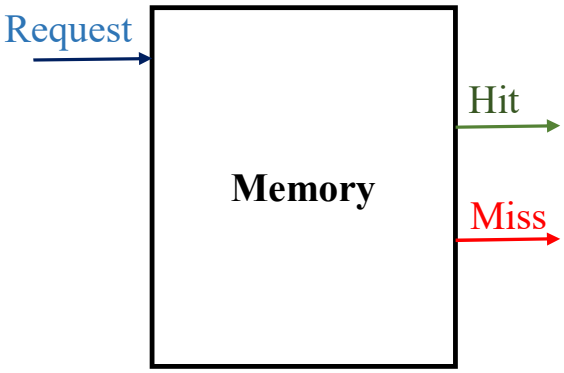
- راهکار: در طراحی حافظه سیستم کامپیوتری تمامی این سطوح وجود دارند ولی با اهداف گوناگون
- سعی بر آن است که داده‌ها تا جایی که بشوند در سطوح بالا نگهداری شوند (چرا؟)
- روش جستجو: از بالاترین سطح به پایین‌ترین سطح (چرا؟)

سلسله مراتب حافظه



سلسله مراتب حافظه

- روش جستجو:
  - از بالاترین سطح شروع می‌کنیم به سمت پایین می‌رویم
    - **داده موجود بود:** Hit و پایان جستجو
    - **داده موجود نبود:** Miss و رفتن به لایه پایین



سلسله مراتب حافظه

- حالت ایده‌ال در جستجو برای هر بخش حافظه: hit شدن همه درخواست‌ها
- ارزیابی کارایی هر بخش حافظه: پارامتر Hit Ratio

$$\text{Hit Ratio} = \frac{\#\ \text{Hits}}{\#\ \text{Hits} + \#\ \text{Misses}}$$

- رابطه مستقیم بین نرخ hit و حجم حافظه
  - در نتیجه نرخ hit در سطوح بالا کمتر است
- ارجحیت جستجوی سلسله مراتبی نسبت به دسترسی مستقیم به حافظه اصلی (چرا؟)

سلسله مراتب حافظه

- هدف طراحان حافظه امروزی: بیشینه کردن نرخ hit در سطوح بالای حافظه
  - امروزه در cache این نرخ به بیش از ۹۴٪ رسیده است
- تعریف پارامتر میانگین زمان دسترسی در حافظه:

$$\text{Average Access Time} = h_1 d_1 + (1-h_1)\,(h_2 d_2 + (1-h_2)\,(\ldots\,(h_{n-1} dn\text{-}1 + (1-h_{n-1})\,d_n \ldots))$$

سلسله مراتب حافظه

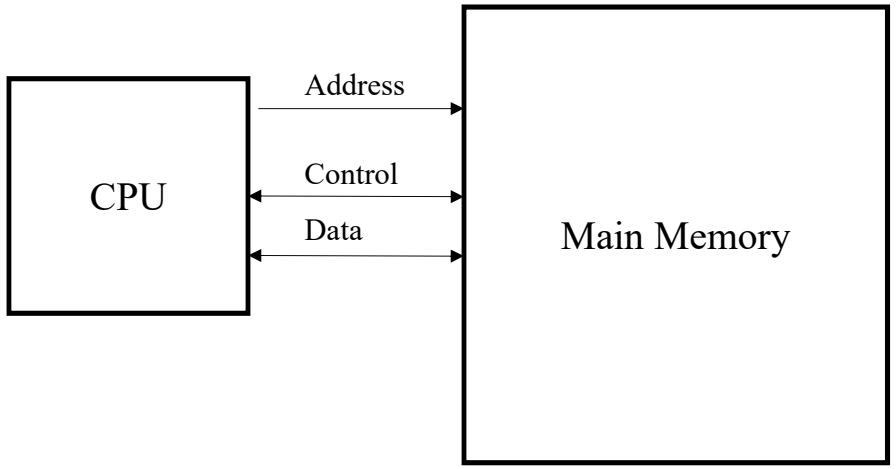
- مثال: سیستم کامپیوتری فرضی متشکل از دو سطح حافظه نهان با نرخ hit برابر ۹۹٪ و اصلی با نرخ hit برابر ۱۰۰٪ داریم. زمان دسترسی به آن‌ها به‌ترتیب 10 ns و 1µs می‌باشد. متوسط زمان پاسخ به درخواست یک داده در این سیستم چقدر است؟

Average access time = $h_1 d_1 + (1-h_1) * h_2 d_2$

$= 0.99 * 10 + 0.01 * 1 * 1000 = 9.9 + 10 = 19.9$ ns

حافظه نهان (Cache)

- ارتباط میان پردازنده و حافظه براساس مدل‌های اولیه سیستم‌های کامپیوتری



حافظه نهان (Cache)

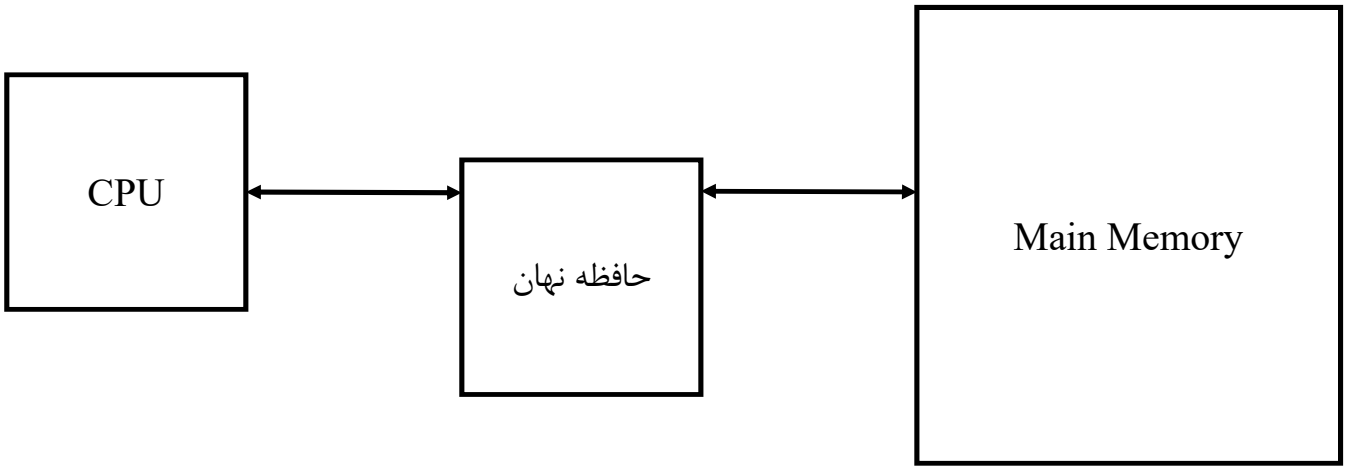
- مطرح شدن ایده پیش‌بینی درخواست‌های پردازنده به حافظه به‌دلیل مشاهده ترتیب و تکرار بودن آن‌ها
- استفاده از حافظه سریع میانی (cache) با هدف:
  - نگهداری درخواست‌های احتمالی
  - افزایش دادن سرعت دسترسی و رفع مشکل کاهش فرکانس پردازنده

حافظه نهان (Cache)

- مثال عملی: متصدی کتابخانه که جعبه کوچکی برای نگهداری کتاب‌هایی که اخیرا پس داده شده یا درخواست شده‌اند در نزدیکی خود دارد
  - جستجوی درخواست جدید از این جعبه شروع می‌شود
  - درصورتی که موجود نبود، به مخزن مراجعه می‌شود
- همین ایده دقیقا در پیاده‌سازی حافظه نهان بکارگرفته شده است

حافظه نهان (Cache)

- درنظر گرفتن فضایی بین پردازنده و حافظه اصلی (مشابه جعبه کوچک کتابدار)



حافظه نهان (Cache)

- **برای بالا بردن سرعت پردازنده:**
  - کاهش دادن فرکانس ساعت پردازنده: محدود است
  - بالا بردن حجم حافظه نهان
  - استفاده از چندین سطح حافظه نهان
    - On Chip Cache: داخل پردازنده
    - Off Chip Cache: خارج پردازنده
  - اضافه کردن تعداد پردازنده‌ها

حافظه نهان (Cache)

- **سوال اساسی:** در حافظه نهان چه اطلاعاتی ذخیره کنیم؟
  - **هدف اصلی:** افزایش سرعت دسترسی ⟵ نرخ hit بالا
  - **طراحی بد:** متوسط زمان دسترسی بیشتر نسبت به حالت دسترسی مستقیم به حافظه اصلی
- انتخاب داده‌های حافظه نهان، براساس پیشینه درخواست‌ها از حافظه اصلی
  - استخراج ویژگی‌های درخواست‌ها با نظارت بر انتقالات داده بین پردازنده و حافظه اصلی
  - خاصیت محلیت (locality) در داده‌ها

حافظه نهان (Cache)

- **خاصیت هم‌جواری (Locality):**
  - **هم‌جواری مکانی (Spatial Locality)**
  - **هم‌جواری زمانی (Temporal Locality)**
- **لحاظ کردن این خاصیت در طراحی حافظه نهان**