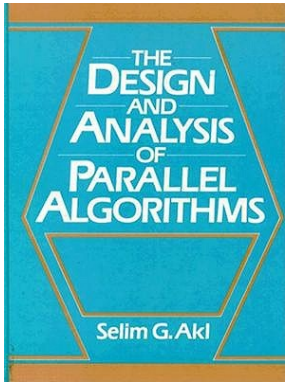
فصل ۱
معرفی

سید ناصر رضوی

rezavi@iust.ac.ir

۱۳۹۰

منابع و مراجع



- منبع اصلی:
  - «طراحی و تحلیل الگوریتم های موازی»؛ سلیم اکل
- سایر منابع:
  - «مقدمه ای بر محاسبات موازی»؛ ویراست دوم (۲۰۰۳)
    - مؤلفین: گراما، گوپتا، کاریپس، کومار
  - «الگوریتم های موازی»، (۲۰۰۸)
    - مؤلفین: کازانوا، لوگران، روبر
  - «هنر برنامه نویسی موازی»، (۱۹۹۳)
    - مؤلف: بروس لستر

فهرست مطالب

- بخش اول: **معرفی**

- بخش دوم: **الگوریتم های موازی**
  - انتخاب، ادغام، مرتب سازی، جستجو
  - عملیات ماتریسی
  - حل مسائل عددی
    - حل یک دستگاه معادلات خطی
    - ریشه یابی معادلات غیرخطی
    - حل معدلات دیفرانسیل جزئی

فهرست مطالب (ادامه)

- بخش دوم: **الگوریتم های موازی (ادامه)**
  - محاسبه تبدیلات فوریه
  - نظریه گراف
  - هندسه محاسباتی

- بخش سوم: **زبان های برنامه نویسی موازی**
  - آشنایی با زبان مالتی پاسکال

نحوه ارزیابی

- ارزیابی:
  - امتحان پایان ترم (۵۰ درصد)
  - تمرین ها (۲۵ درصد)
  - سمینار (۲۵ درصد)

- تحقیقات بیشتر و پیاده سازی (نمره تشویقی)

چرا کامپیوترهای موازی؟

- مثال:
  - پردازش داده های جمع آوری شده (10^{10} بیت در ثانیه) به وسیله ماهواره ها درباره وضعیت آب و هوا، آلودگی، کشاورزی، منابع طبیعی و ... نیاز به نرخ پردازش حداقل برابر با 10^{13} عمل در ثانیه دارد.

- مثال:
  - انجام یک عمل جراحی از راه دور نیاز به نرخ پردازش حداقل برابر با 10^{15} عمل در ثانیه دارد.

چرا کامپیوترهای موازی؟

- چند مثال دیگر:
  - اکتشاف نفت
  - برنامه ریزی اقتصادی
  - مدیریت پایگاه داده های بسیار بزرگ
  - ستاره شناسی
  - تشخیص گفتار به صورت بلادرنگ
  - روباتیک
  - آیرودینامیک
  - فیزیک هسته ای و فیزیک پلاسما
  - ...

کامپیوتر موازی

- کامپیوتر موازی:
  - یک کامپیوتر با تعداد زیادی از واحدهای پردازشگر

- حل مسأله با استفاده از یک کامپیوتر موازی:
  - تقسیم مسأله به تعدادی زیرمسأله
  - حل همزمان زیرمسائل با حل هر زیرمسأله به وسیله یک پردازشگر جداگانه
  - ترکیب نتایج برای به دست آوردن راه حل مسأله اصلی

- الگوریتم موازی:
  - روش حل یک مسأله به منظور اجرا بر روی یک کامپیوتر موازی

مدل های محاسباتی

- طراحی الگوریتم های موازی نیازمند داشتن درک درستی از مدل های محاسباتی است.

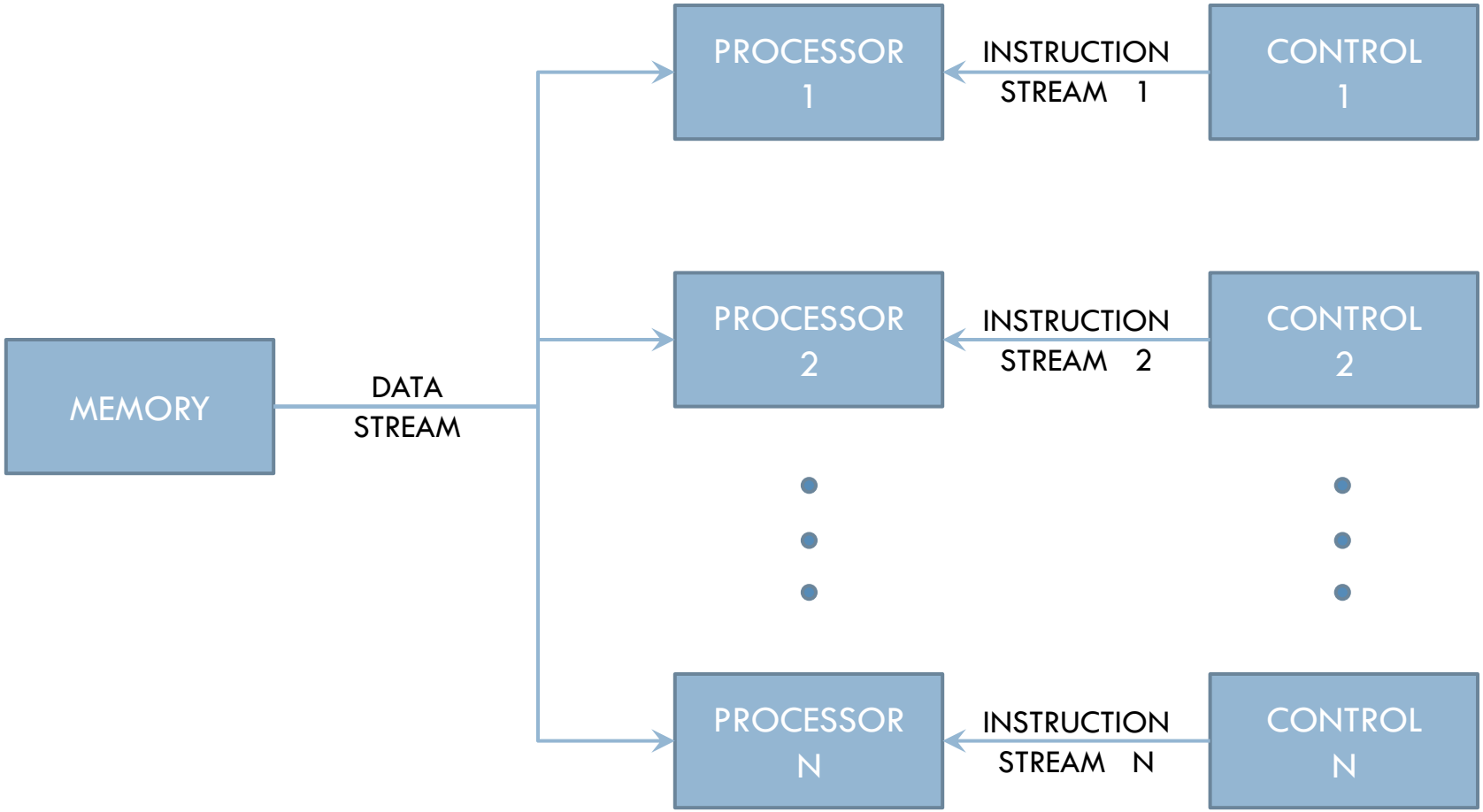
- چهار دسته از کامپیوترها:
  - یک دستورالعمل، یک داده (SISD)
  - چند دستورالعمل، یک داده (MISD)
  - یک دستورالعمل، چند داده (SIMD)
  - چند دستورالعمل، چند داده (MIMD)

یک دستورالعمل، یک داده (SISD)

- مدل وان نیومن (ترتیبی)، اواخر دهه ۴۰

- **مثال ۱-۱:**
  - محاسبه مجموع n عدد با مرتبه $O(n)$

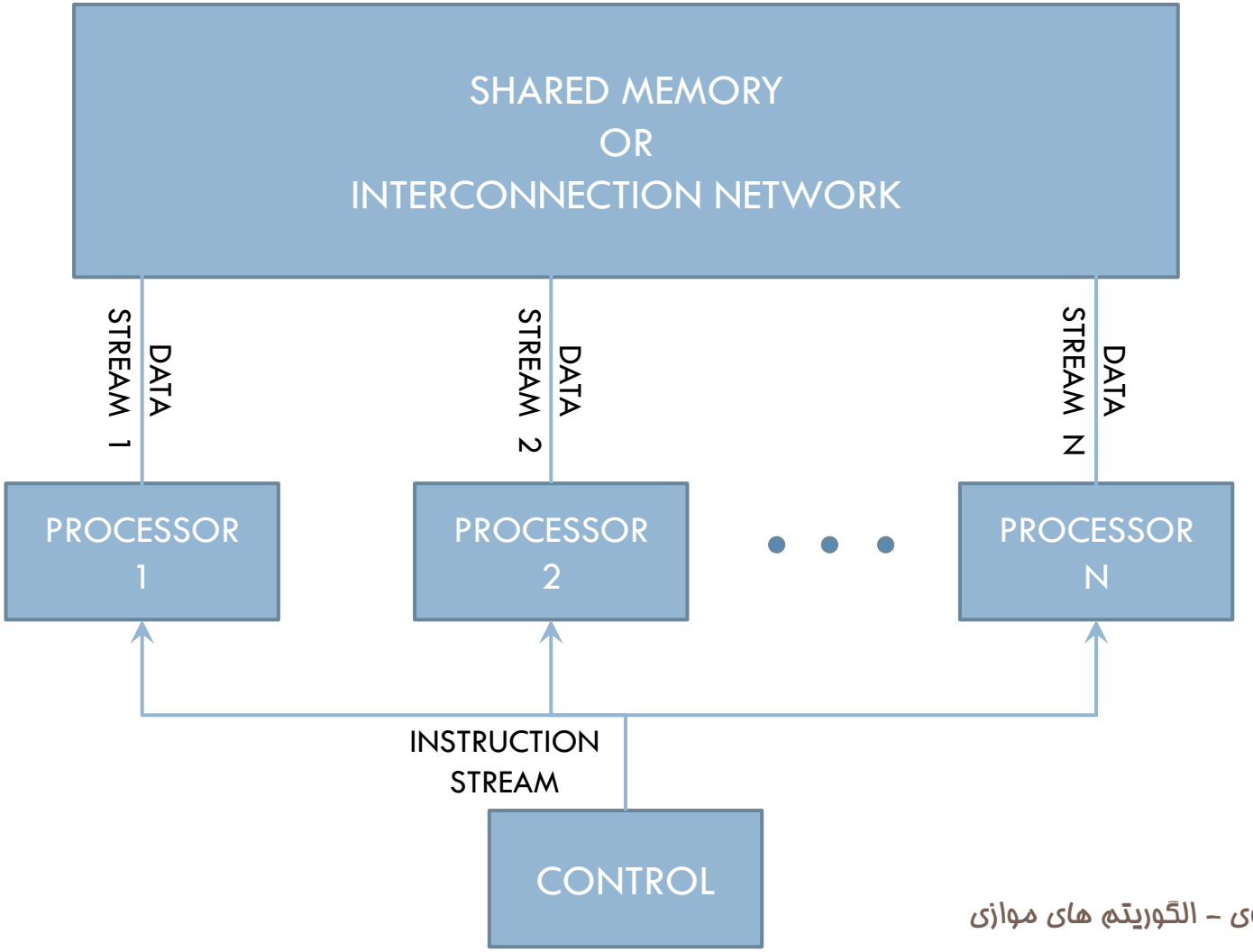
چند دستورالعمل، یک داده (MISD)



چند دستورالعمل، یک داده (MISD)

- **مثال ۱-۲:**
  - تشخیص اول بودن یک عدد
- **مثال ۱-۳:**
  - دسته بندی اشیا (عددی یا فیزیکی)

یک دستورالعمل، چند داده (SIMD)



SIMD با حافظه اشتراکی

- ماشین موازی با دستیابی تصادفی (مدل PRAM)
- ارتباط میان پردازنده ها از طریق حافظه مشترک انجام می شود

- تقسیم بندی بر اساس نحوه دسترسی همزمان به یک مکان یکسان از حافظه:
  - خواندن انحصاری، نوشتن انحصاری (EREW)
  - خواندن همزمان، نوشتن انحصاری (CREW)
  - خواندن انحصاری، نوشتن همزمان (ERCW)
  - خواندن همزمان، نوشتن همزمان (CRCW)

SIMD با حافظه اشتراکی

- نوشتن همزمان در یک مکان یکسان می تواند باعث بروز مشکل شود.

- بنابراین، به یک روش **قطعی** برای رفع تصادم نوشتن نیاز داریم:
  - روش اولویت: فقط پردازنده با کوچکترین شماره مجاز به نوشتن است.
  - روش سازگار: اگر کمیت هایی که قرار است نوشته شوند با یکدیگر برابر باشند، همه پردازنده ها مجاز به نوشتن هستند. در غیر این صورت هیچ یک از پردازنده ها اجازه نوشتن را ندارند.
  - روش تلفیقی: مجموع کمیت هایی که قرار است نوشته شوند، ذخیره گردد. (به جای عمل جمع از هر عمل جابجاپذیر و شرکت پذیر می توان استفاده کرد)

SIMD با حافظه اشتراکی

- **مثال ۱-٤:**
  - جستجوی x در یک فایل نامرتب بسیار بزرگ شامل n مدخل متمایز
- استفاده از مدل EREW با N پردازنده به طوری که $N \leq n$:
  - مرحله ۱) **انتشار** مقدار x به همه پردازنده ها در زمان $O(\log N)$
    - خواندن x توسط P_1 و ارسال آن به P_2،
    - ارسال همزمان x از P_1 و P_2 به ترتیب به P_3 و P_4،
    - ارسال همزمان x از P_1، P_2، P_3 و P_4 به ترتیب به P_5، P_6، P_7 و P_8،
  - مرحله ۲) جستجوی همزمان در زیرفایل ها به وسیله پردازنده ها در زمان $O(n/N)$

SIMD با حافظه اشتراکی

- اصلاح الگوریتم قبل با استفاده از متغیر بولی F
  - اگر یکی از پردازنده ها موفق به یافتن x شود، مقدار F را برابر با True قرار داده و در نتیجه عمل جستجو در همه پردازنده ها پایان می پذیرد.
  - زمان اجرا: Log N + (n/N) Log N
    - به دلیل انتشار مقدار F بین پردازنده ها

- یک روش بهتر به منظور استفاده مؤثر از ترفند خاتمه زودهنگام:
  - استفاده از مدل قدرتمندتر CREW
  - زمان اجرا: O(n/N)

SIMD با حافظه اشتراکی

- اگر در مثال ۱–٤ مدخل های فایل لزوماً متمایز نباشند، ممکن است بیش از یک پردازنده به طور همزمان بخواهند یافتن x را گزارش کنند.

- در این صورت نیاز به استفاده از مدل قدرتمندتر CRCW خواهیم داشت.

شبیه سازی دسترسی های چندگانه بر روی مدل EREW

- **شبیه سازی خواندن های همزمان:**
  - با استفاده از روش انتشار

- **شبیه سازی نوشتن های همزمان:**
  - تنها به شرطی که مقداری که قرار است نوشته شود در همه پردازنده ها یکسان باشد، نوشتن همزمان امکان پذیر است.
  - فرض می کنیم پردازنده P_i قرار است مقدار a_i را بنویسد.
  - روش کار شبیه به روش انتشار است:
    - به ازای $1 \le i \le N/2$، اگر a_i و $a_{i+N/2}$ برابر باشند، پردازنده P_i مقدار متغیر ثانویه b_i را برابر با true قرار می دهد و گرنه مقدار آن را برابر با false قرار می دهد.
    - به ازای $1 \le i \le N/4$، اگر b_i و $b_{i+N/4}$ هر دو برابر true باشند و $a_{i+N/4} = a_i$، پردازنده P_i مقدار b_i را برابر با true قرار می دهد و گرنه مقدار آن را برابر با false قرار می دهد.
    - و به همین ترتیب الی آخر
    - در نهایت اگر همه مقادیر مساوی باشند، پردازنده P_1 این مقدار را در آدرس مورد نظر می نویسد.

شبیه سازی دسترسی های چندگانه بر روی مدل EREW

- **شبیه سازی خواندن های همزمان (m از N):**
  - فرض کنید الگوریتم ما در کل نیاز به دستیابی به M مکان از حافظه مشترک داشته باشد.
  - ایده: برای هر یک از این M مکان، 2 – 2N مکان از حافظه را در نظر می گیریم.
    - به صورت یک درخت دودویی کامل با ریشه M و N برگ شماره دار
    - هر برگ متناظر با یک پردازنده با شماره یکسان است
    - پردازنده هایی که نیاز به دسترسی به مکان A دارند، درخواست خود را اعلام می کنند و این درخواست ها از درخت مربوطه به سمت ریشه حرکت می کنند
    - اولین پردازنده که درخواستش به ریشه می رسد مقدار A را خوانده و این مقدار در درخت به سمت پایین منتشر می شود.
  - فرآیند نوشتن های همزمان نیز مشابه فرآیند فوق است.

- توضیحات رسمی انتشار چندگانه و ذخیره چندگانه در بخش ۳-٤

امکان سنجی مدل SIMD با حافظه اشتراکی

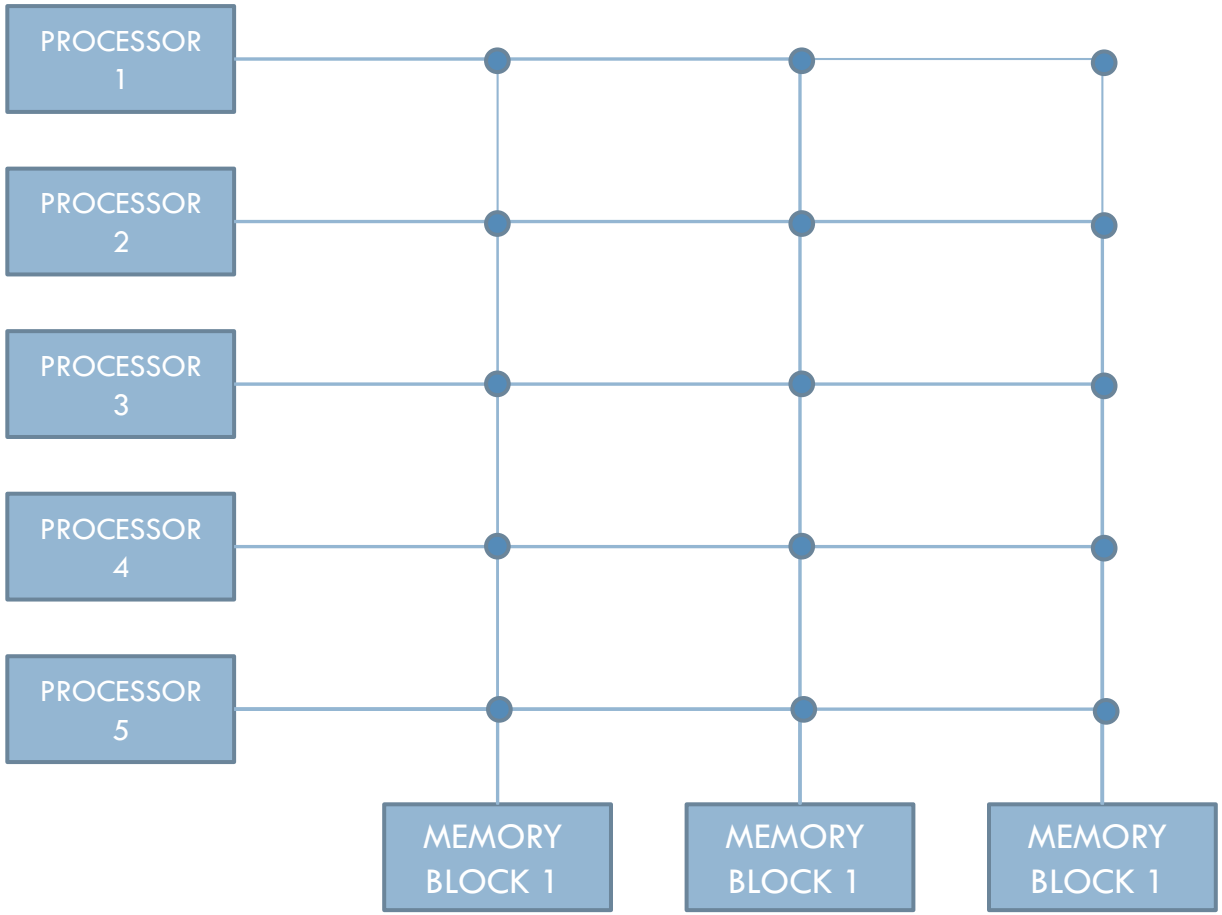
- آیا ساختن مدل SIMD با حافظه اشتراکی مقرون به صرفه است؟
  - هزینه بالای مدارهای دستیابی به حافظه به ویژه برای N و M های بزرگ
  - $cost = N \times f(M)$

- روش اول:
  - تقسیم بندی حافظه اشتراکی به R بلوک با اندازه های مساوی
  - در هر لحظه، فقط یک پردازنده می تواند به یک بلوک دسترسی داشته باشد.

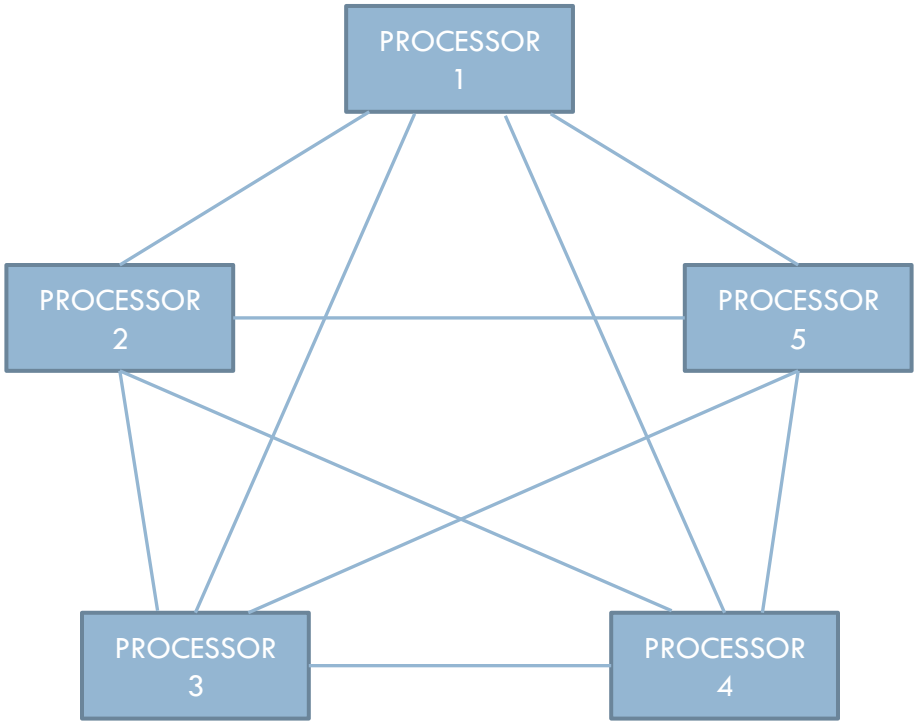
- روش دوم:
  - توزیع N مکان حافظه در میان N پردازنده به طور مساوی (M/N)
  - متصل نمودن هر زوجی از پردازنده ها از طریق یک خط ارتباطی مستقیم

روش اول) تقسیم بندی حافظه به R بلوک

$$cost = R \times f(M/R)$$



روش دوم) اتصال دو به دوی پردازنده ها



روش دوم) اتصال دو به دوی پردازنده ها

- نحوه آدرس دهی:
  - آدرس پردازنده مورد نظر شامل Log (N – 1) بیت آدرس، به علاوه
  - آدرس مکان حافظه مورد نظر در پردازنده پردازنده شامل Log (M/N) بیت

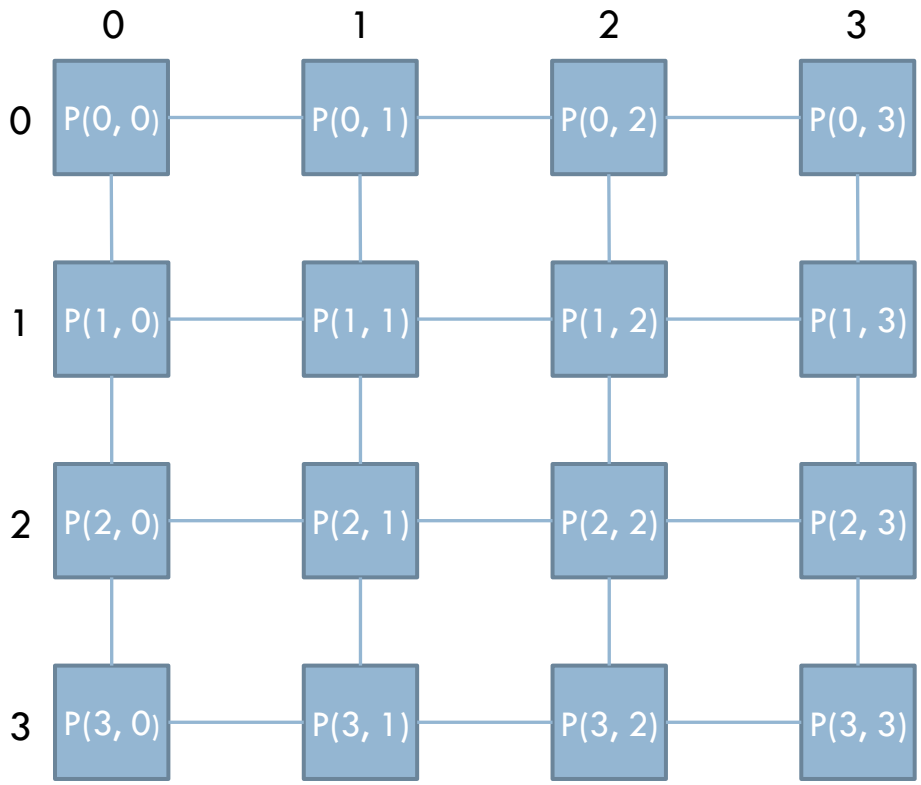
- هزینه: N(N-1)/2 خط ارتباطی دو طرفه
  - راه حل: استفاده از شبکه های اتصالی ساده تر

استفاده از شبکه های ساده تر در SIMD

- آرایه خطی

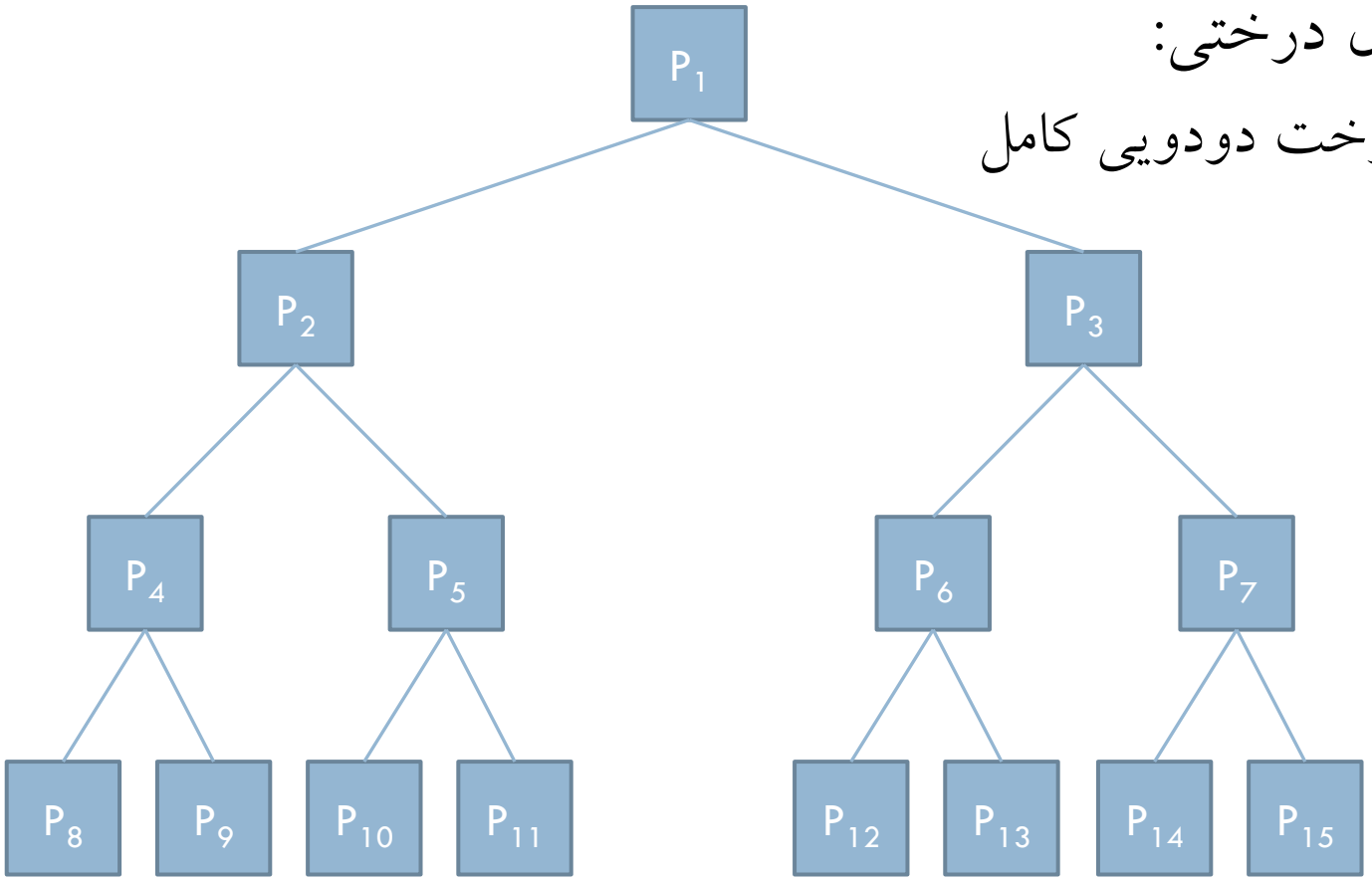
استفاده از شبکه های ساده تر در SIMD

- آرایه دو بعدی (mesh)



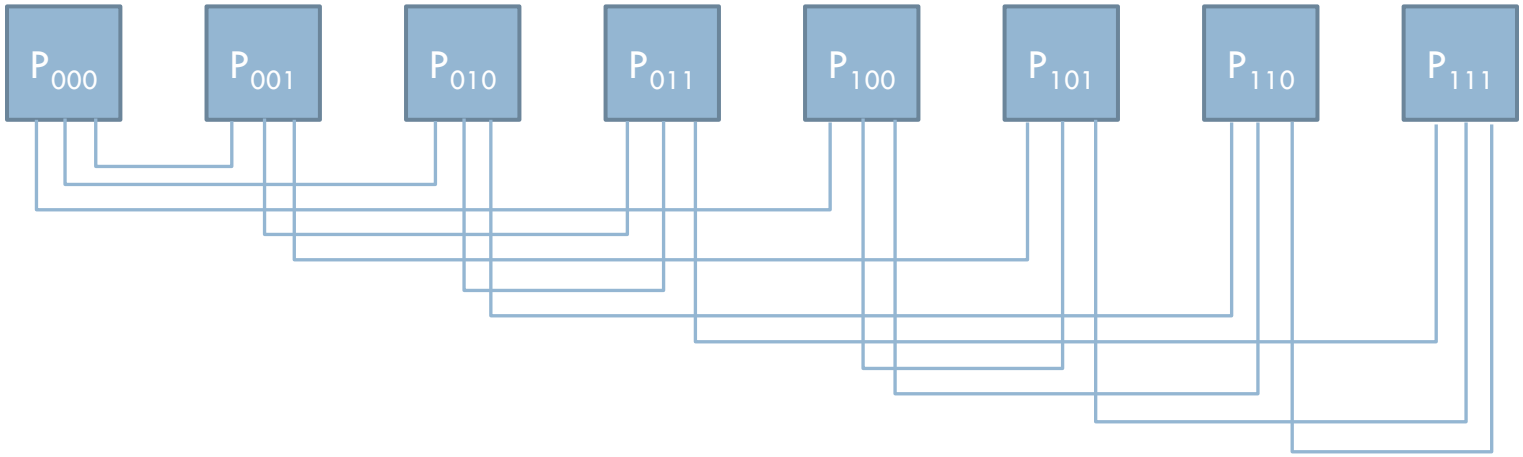
استفاده از شبکه های ساده تر در SIMD

- اتصال درختی:
  - درخت دودویی کامل



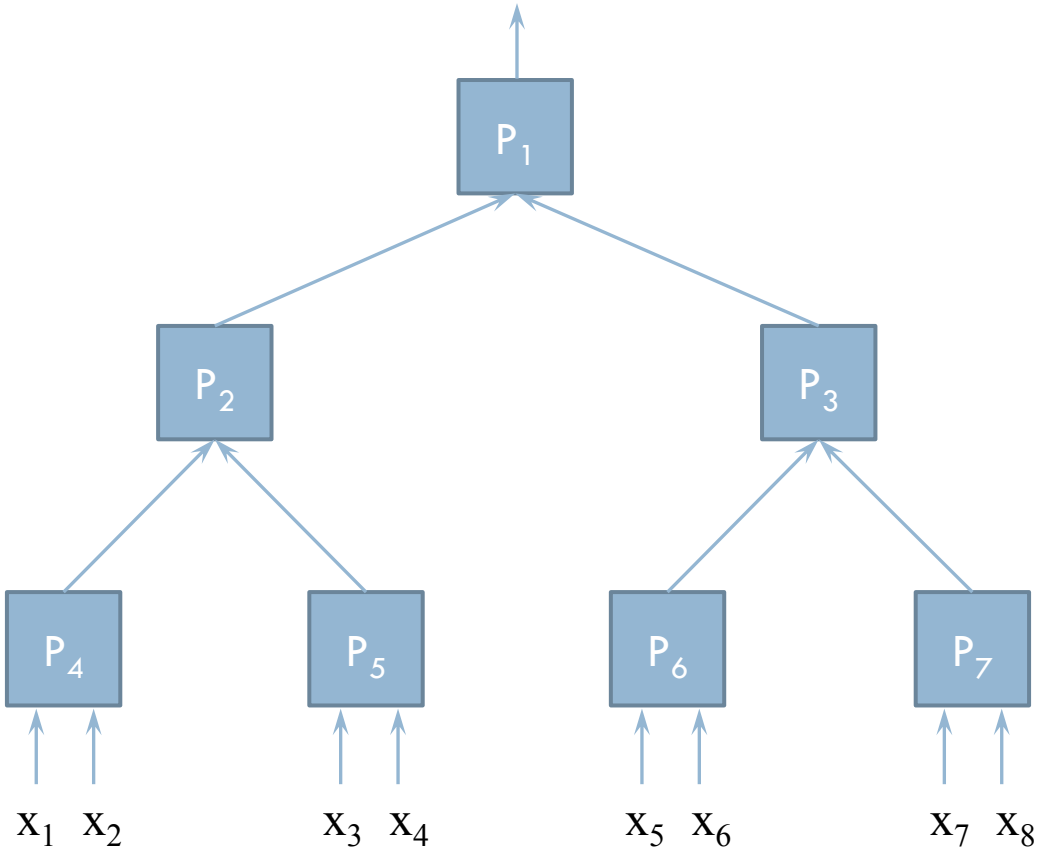
استفاده از شبکه های ساده تر در SIMD

- اتصال مکعبی



مثال) استفاده از اتصال درختی

- **مثال ۱-۵)** محاسبه مجموع n عنصر در زمان O(Log N)



مثال) استفاده از اتصال درختی

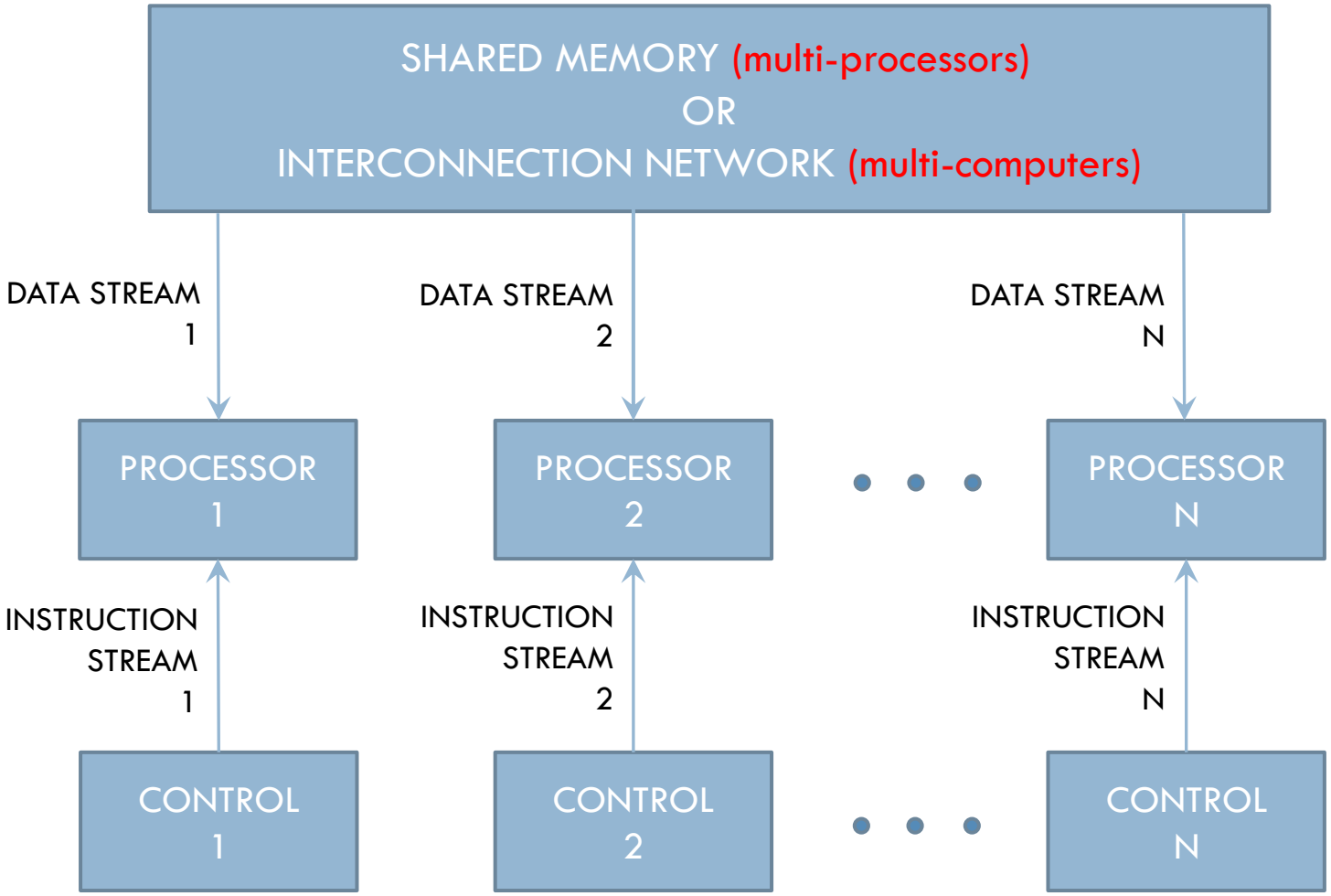
- محاسبه مجموع m مجموعه از عناصر هر یک شامل n عنصر
  - روش ترتیبی: mn
  - روش موازی با استفاده از اتصال درختی: m(log n)
  - با استفاده از **فرآیند لوله کشی**: Log n + (m – 1)

جمع بندی مدل SIMD

- مدل های SIMD:
  - نسبت به مدل های MISD بسیار رایج تر هستند.
  - تحلیل، طراحی و پیاده سازی الگوریتم ها برای این مدل ها نسبتاً ساده است.
  - تنها زمانی قابل استفاده هستند که مسأله مورد نظر قابل تقسیم به تعدادی **زیرمسأله یکسان** باشد.

- برای مسائلی که زیرمسائل آنها لزوماً یکسان نیستند، از مدل MIMD استفاده می کنیم.

مدل چند دستورالعمل، چند داده (MIMD)



مدل چند دستورالعمل، چند داده (MIMD)

- چندپردازنده ها (ماشین های پیوند قوی)
  - کامپیوترهای MIMD با حافظه اشتراکی
  - انواع: EREW، CREW، ERCR و CRCW

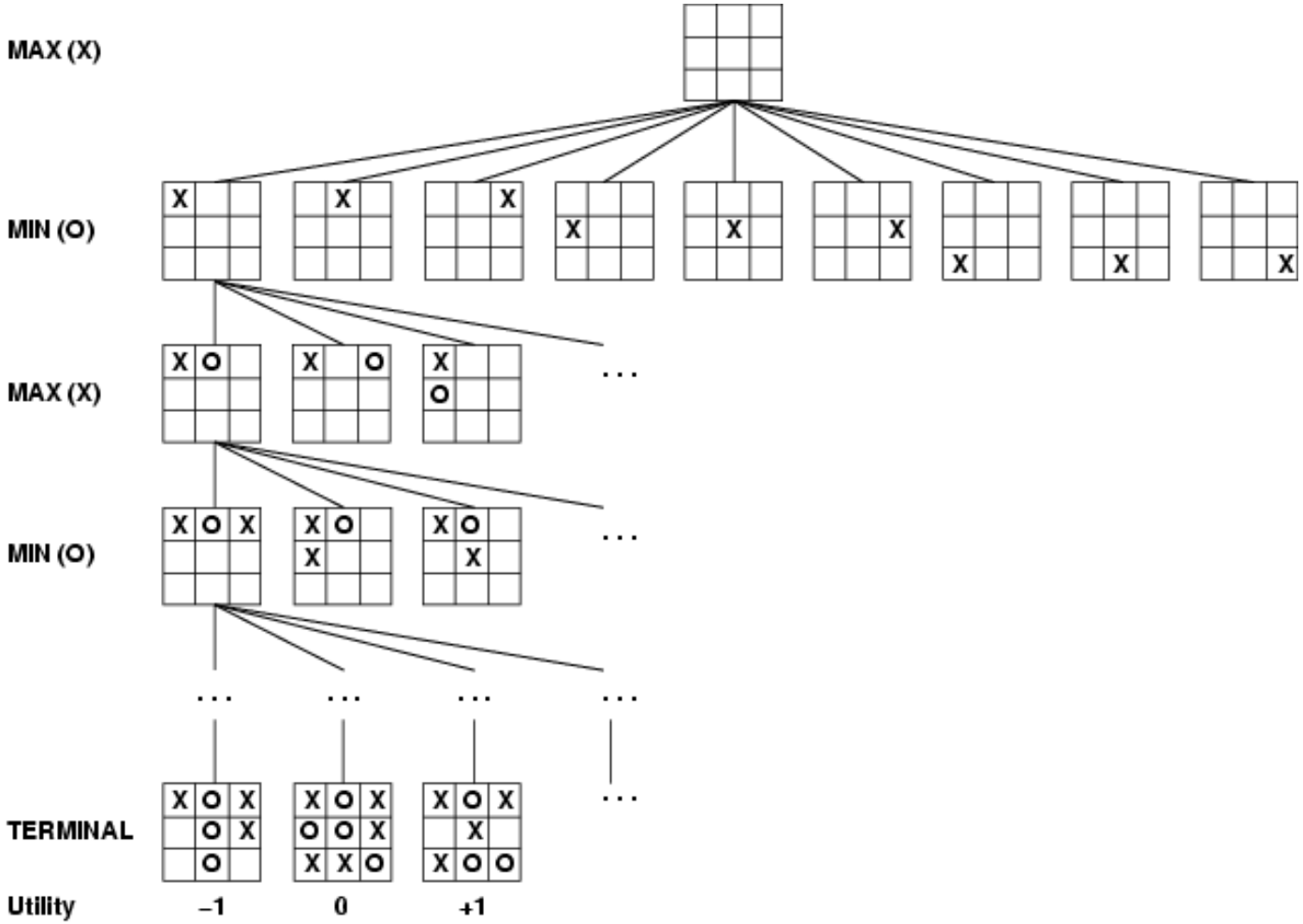
- چندکامپیوترها یا سیستم های توزیع شده (ماشین های پیوند ضعیف)
  - کامپیوترهای MIMD با شبکه اتصالی
  - انواع بر اساس فاصله فیزیکی میان انواع پردازنده ها
    - فاصله نزدیک: چندکامپیوترها (مثلاً در یک اتاق)
    - فاصله دور: سیستم های توزیع شده (مثلاً در شهرهای مختلف)
      - تعداد انتقالات داده ای بسیار مهم تر تعداد عملیات انجام شده

مدل چند دستورالعمل، چند داده (MIMD)

- مثال ۱-٦) برنامه های کامپیوتری برای انجام بازی های راهبردی مانند شطرنج

  - راه حل ترتیبی:
    - جستجوی مسیرها در درخت بازی به صورت اول-عمق
  - راه حل موازی:
    - توزیع زیردرخت های ریشه در بین پردازنده ها

مدل چند دستورالعمل، چند داده (MIMD)



تحلیل الگوریتم ها

- معیارهای ارزیابی الگوریتم های موازی:
  - زمان اجرا
    - شمارش تعداد **مراحل محاسباتی**:
      - محاسبه تعداد دفعات اجرای عمل اصلی برحسب اندازه ورودی (در بدترین حالت)
    - شمارش تعداد **مراحل مسیریابی**:
      - تعداد انتقالات داده ای بین پردازنده ها
    - کران های بالا و پایین
      - مثال: کران پایین ضرب دو ماتریس $n \times n$ برابر است با n^2 .
      - کران پایین الگوریتم های مرتب سازی مبتنی بر مقایسه برابر است با $n \log n$.
  - تعداد پردازنده ها
  - هزینه

تحلیل الگوریتم ها: نمادهای مجانبی

- فرض کنید $f(n)$ و $g(n)$ دو تابع از اعداد صحیح مثبت به اعداد حقیقی مثبت باشند.

- تعریف نماد O: $g(n) \in O(f(n))$
  - می گوییم $g(n)$ حداکثر از مرتبه $f(n)$ است، اگر ثابت های مثبت c و n_0 وجود داشته باشند به گونه ای که به ازای هر $n \geq n_0$ داشته باشیم:

$$g(n) \leq c \cdot f(n)$$

- تعریف نماد Ω: $g(n) \in \Omega(f(n))$
  - می گوییم $g(n)$ حداقل از مرتبه $f(n)$ است، اگر ثابت های مثبت c و n_0 وجود داشته باشند به گونه ای که به ازای هر $n \geq n_0$ داشته باشیم:

$$g(n) \geq c \cdot f(n)$$

تحلیل الگوریتم ها: نمادهای مجانبی

- **مثال ۱-۱۲)** استفاده از نمادهای مجانبی

  - مسأله ضرب ماتریس ها
    - حد پایین: $\Omega(n^2)$
    - حد بالا: $O(n^{2.5})$

  - مسأله مرتب سازی
    - حد پایین: $\Omega(n \log n)$
    - حد بالا: $O(n \log n)$

تحلیل الگوریتم های موازی

- در تحلیل الگوریتم های موازی، دو عامل دیگر نیز باید در نظر گرفته شوند:
  - مدل محاسبات موازی
  - تعداد پردازنده ها

- **مثال ۱-۱۳)** محاسبه مجموع n^2 عدد با استفاده از یک مش $n \times n$
  - تعداد مراحل مسیریابی: $\Omega(n)$

تحلیل الگوریتم های موازی

- **نرخ تسریع (speedup)**
  - معیاری به منظور ارزیابی کیفیت الگوریتم موازی

$$\text{نرخ تسریع} = \frac{\text{زمان اجرای بهترین الگوریتم ترتیبی شناخته شده در بدترین حالت}}{\text{زمان اجرای الگوریتم موازی در بدترین حالت}}$$

- **مثال ۱-۱۴)** محاسبه نرخ تسریع در مثال جستجوی یک فایل شامل n عنصر بر روی یک مدل SIMD (از نوع CREW) با N پردازنده

$$O(n) / O(n/N) = O(N)$$

تحلیل الگوریتم های موازی

- مثال ۱-۱۵) محاسبه نرخ تسریع الگوریتم محاسبه مجموع n عنصر با استفاده از ساختار درختی شامل $n - 1$ پردازنده
- $O(n/\log n)$

- به طور کلی:

$$1 \leq \text{speed up} \leq N$$

تحلیل الگوریتم های موازی

- **هزینه**

$$c(n) = t(n) \times p(n)$$

هزینه = زمان اجرا × تعداد پردازنده ها

- می گوییم یک الگوریتم موازی دارای **هزینه بهینه** است، اگر هزینه با حد پایین شناخته شده برای مسأله مورد نظر هم مرتبه باشد.

تحلیل الگوریتم های موازی

- مثال ۱-۱۷)
  - محاسبه هزینه الگوریتم موازی برای مثال جستجوی یک فایل شامل n عنصر بر روی یک مدل SIMD (از نوع CREW) با N پردازنده:

$$c(n) = O(n/N) * O(N) = O(n)$$

  - در نتیجه این الگوریتم موازای دارای هزینه بهینه است.

تحلیل الگوریتم های موازی

معیار کارایی

- اگر بهترین الگوریتم ترتیبی برای مسأله مورد نظر ناشناخته باشد، در این صورت کارایی الگوریتم موازی به صورت زیر تعریف می شود.

$$\text{کارایی} = \frac{\text{زمان اجرای بهترین الگوریتم ترتیبی شناخته شده در بدترین حالت}}{\text{هزینه الگوریتم موازی}}$$

شبه کدهای کتاب

- انواع عملیات موجود در یک الگوریتم موازی
  - عملیات ترتیبی
  - عملیات موازی

- بیان عملیات ترتیبی با استفاده از:
  - ساختاری شرطی مانند if … then … else …
  - حلقه های for و while
  - دستورات انتساب
  - عملیات ورودی و خروجی
  - عملیات منطقی (مانند AND، OR، NOT)

شبه کدهای کتاب

- بیان عملیات موازی:

```
do steps i to j in parallel
    step i
    step i + 1
    …
    step j
```

```
for i = j to k do in parallel
    {the operation to be performed by processor  i}
end for
```

```
for i = r, s, …, t do in parallel
    {the operation to be performed by processor  i}
end for
```

```
for all i in S do in parallel
    {the operation to be performed by processor  i}
end for
```