2/8/2014

فصل اول

بسترهای پردازش موازی

مقدمه

- اجزاء یک کامپیوتر: پردازنده، حافظه و مسیرداده
  - گلوگاه هایی را در کل نرخ پردازشی یک سیستم ارائه می دهند.
- راه حل: ارائه چندگانگی (multiplicity) در اجزاء فوق.
  - موازی سازی ضمنی (implicity) از دید برنامه نویسان.
  - قابل وضوح برای برنامه نویسان به شکل های مختلف.

2/8/2014

موازی سازی ضمنی

- اجرای خط لوله و سوپراسکالر
  - اجرای خط لوله: تقسیم یک وظیفه بزرگ به وظائف کوچکتر و اجرای آنها در چندین مرحله ← خط لوله چند مرحله ایی
    - افزایش سرعت اجرای یک وظیفه
  - اجرای سوپر اسکالر: داشتن چندین خط لوله در یک پردازنده

3

موازی سازی ضمنی

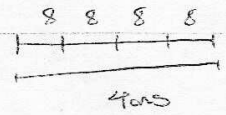
- خط لوله
  - قابلیت همپوشانی اجرای چندین دستورالعمل را در یک زمان دارد.
    - موازی سازی را در اجرای دستورات بکار می گیرد و از دید برنامه نویس مخفی است.
  - مثال: خط تولید ماشین: فرض شود ۲ ساعت زمان تولید یک ماشین است و ۱۰۰ ماشین بایستی تولید شود.
    - توان عملیاتی خط تولید: تعداد ماشینهای تولید شده در یک ساعت.
    - توان عملیاتی پردازنده: تعداد دستورات کامل شده در یک ثانیه است.
  - مرحله خط لوله: هر گام در خط لوله را مرحله (stage) خط لوله گویند. همه مراحل خط لوله بایستی دارای طول یکسانی باشند.
    - در خط تولید هر مرحله یک گروه کاری در خط تولید است.
  - Speedup خط لوله:

$$speedup = \frac{time\ per\ instruction\ on\ unpipelined\ machine}{time\ of\ pipe\ stage} x$$

4

2/8/2014

مثال:



$\frac{40}{8} = 5x$

$\frac{40}{10} = 4x$

این موازی سازی از دید برنامه نویس مخفی است

موازی سازی ضمنی

خط لوله

- فرض شود انجام یک عمل ۴۰ نانوثانیه طول می کشد و در ۵ مرحله قابل اجرا است. بنابراین زمان هر مرحله (درحالت ایده ال) برابر با ۸ نانو ثانیه است: speedup=40/8=5x حداکثر
- فرض شود همان عمل در ۵ مرحله انجام می شود که ۴ مرحله آن ۷.۵ نانو ثانیه و یک مرحله ۱۰ نانوثانیه باشد. لذا زمان یک مرحله ماکزیمم طول مراحل می باشد، یعنی ۱۰ نانوثانیه.

بنابراین:

Speedup=40/10=4x

حداکثر speed up = زمان اجرا سریال / زمان هر stage

این بالایی زمان دولت الت و طول هر stage برابر است در حالت ایده آل الت و آکومول stage ۵ برابر

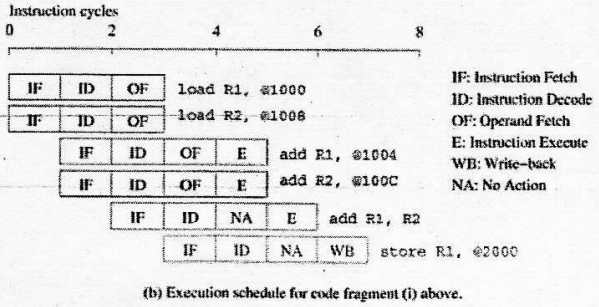
بنابراین باید در مخرج زمان stage ماکزیمم را در نظر بگیریم

خود کد نویسی خیلی تاثیر دارد در چگونه pipeline را مورد استفاده قرار گیرد

مثال: اجرای سوپراسکالر دو-مسیره

```
1. load R1, @1000      1. load R1, @1000      1. load R1, @1000
2. load R2, @1008      2. add R1, @1004       2. add R1, @1004
3. add R1, @1004       3. add R1, @1008       3. load R2, @1008
4. add R2, @100C       4. add R1, @100C       4. add R2, @100C
5. add R1, R2          5. store R1, @2000     5. add R1, R2
6. store R1, @2000                            6. store R1, @2000
        (i)                    (ii)                   (iii)
```

(a) Three different code fragments for adding a list of four numbers.



(b) Execution schedule for code fragment (i) above.

دستور R1 و R2 مستقل هستند از هم دستورس می توانند به صورت موازی اجرا شوند

دستور 3 و 4 مستقل هستند از هم

دستور 3 به 1 وابسته الت و دستور 3 به 2 و 4 وابسته الت

دستور پنجم به دستور سوم و چهارم وابسته الت و دستور 6 به 5 وابسته است پس می توانند به صورت موازی اجرا شوند

دستور ششم بعد از دستور پنجم اجرا شود

در کد اول از هر دو pipeline به خوبی استفاده نشده

در کد دوم به صورتی نوشته شده که از pipeline هم استفاده شده و pipeline به صورت موازی اجرا می شوند

2/8/2014

محدودیت های سیستم حافظه

- کارایی یک برنامه روی یک سیستم علاوه بر سرعت پردازنده، به قابلیت سیستم حافظه جهت انتقال داده ها به پردازنده نیز بستگی دارد.
- پهنای باند سیستم حافظه (Memory System Bandwidth):
  - نرخ انتقال داده ها از حافظه به پردازنده.
- تاخیر سیستم حافظه (Memory System Latency):
  - زمان انتقال یک بلاک داده بطول b از حافظه به پردازنده.

این دو تا رو در شکل بعدی داده اند (پهنای باند و تاخیر)

7

تاثیر تاخیر حافظه روی کارایی

- مثال: پردازنده ای با سرعت 1GHz (کلاک یک نانوثانیه) و یک حافظه متصل به آن با تاخیر ۱۰۰ نانوثانیه (بدون کاشه) در نظر بگیرید.
  - فرض شود پردازنده دو واحد جمع-ضرب دارد و چهار دستورالعمل را در یک سیکل کلاک (یک نانوثانیه) انجام می دهد. بنابراین پیک پردازنده 4GFLOPS است.
  - تاخیر حافظه ۱۰۰ سیکل است و برای هر درخواست حافظه از طرف پردازنده، بایستی پردازنده ۱۰۰ سیکل منتظر بماند.
  - فرض شود دو بردار ضرب داخلی می شوند. لذا برای هر دو عنصر نیاز به یک عمل ضرب-جمع است. لذا برای خواندن هر دو عنصر بایستی پردازنده ۱۰۰ نانوثانیه صبر کند. به عبارتی پیک پردازنده به 10MFLOPS کاهش پیدا می کند.

پیک پردازنده = سرعت پردازنده × (تعداد دستورالعمل ها)

8

Flops : Floating per second

$ab = a_1b_1 + a_2b_2 + \cdots + a_nb_n$ ضرب نقطه ای دو بردار

فقط a_1 و b_1 را می خواند که یک GFlops است که تقسیم بر ۱۰۰ شده است و می شود 10mFlops یعنی باید منتظر بماند

خواندن دو عدد است

$\frac{4G}{100}$ اگر پهنای باند ۴ گیگ باشد در ثانیه چند تا a_1b_1 و a_2b_2 و a_3b_3 و a_4b_4 می خواند که در نهایت پیک می شود

4

2/8/2014

بهبود کارایی با استفاده از کاشه

- مثال: پردازنده ای با سرعت 1GHz (کلاک یک نانوثانیه) و یک حافظه متصل به آن با تاخیر ۱۰۰ نانوثانیه (بدون کاشه) در نظر بگیرید. در این حالت یک کاشه با اندازه 32Kb و با تاخیر ۱ نانو ثانیه (یک سیکل کلاک) معرفی می شود.
  - فرض شود دو ماتریس A و B با ابعاد 32*32 در هم ضرب شوند و کاشه نیز آنقدر بزرگ است (عناصر را طوری در نظر می گیریم) که هر سه ماتریس A، B و C (ماتریس حاصلضرب) در آن جای گیرد. پس ابتدا دو ماتریس در کاشه بارگزاری می شوند: 2K داده بایستی واکشی شوند که معادل 2Kns=200us زمان می برد.
  - برای ضرب دو ماتریس n*n نیاز به 2*n^3 عمل است که در این مثال 64K عمل نیاز داریم که 64K/4=16K سیکل (۱ نانوثانیه) که معادل 16us است، زمان لازم است. که جمعا: 200+16us=216us که معادل نرخ 64K/216=303 MFLOPS است که ۳۰ برابر نسبت به مثال قبل بهتر شده است. بهرحال این مقدار 10% پیک پردازنده است.

9

تاثیر پهنای باند حافظه روی کارایی

- پهنای باند حافظه: نرخ انتقال داده ها بین پردازنده و حافظه
  - بستگی به پهنای باند گذرگاه بین پردازنده و حافظه دارد.
- مثال: ضرب دو بردار
  - فرض شود اندازه بلاک نسبت به مثال قبل به چهار کلمه در هر بلاک افزایش یابد. بنابراین در مثال قبل برای انجام هر عمل ضرب-جمع بایستی ۱۰۰ سیکل پردازنده منتظر بماند، درنتیجه پیک سرعت به 10MFLOS کاهش یافت.
  - حال در این مثال چون در هر ۱۰۰ سیکل چهار کلمه (بجای یک کلمه) بارگزاری می شود، لذا پیک سرعت به 40MFLOS نسبت به قبل افزایش می یابد.

10

2/8/2014

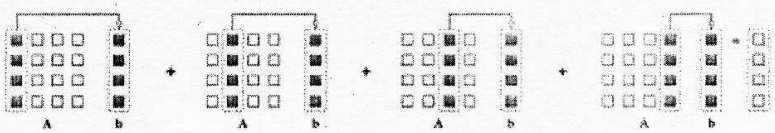
نتیجه گیری در خصوص پهنای باند

- مثالهای قبل نشان داد که با افزایش پهنای باند، نرخ پیک محاسباتی افزایش می یابد. این فرض زمانی درست است که ترتیب دسترسی به داده ها در حافظه طوری باشد که کلمات داده ایی پی در پی در حافظه توسط دستورات استفاده شوند. بعبارتی، توالی از دستورات داده های یک بلاک را بکار برند.
- Spatial locality (محلیت فضایی): از دید برنامه نویسی، دسترسی به داده ها ی پی در پی در حافظه با دستورات پی در پی که منجر به محاسبات موفق می شود.
  - اگر محاسبه ایی (یا الگوی دسترسی) طوری باشد که محلیت فضایی نداشته باشد، پهنای باند موثر خیلی کوچکتر از پیک پهنای باند خواهد شد.

تاثیر دسترسی با گامهای بلند (strided access)

- مثال: فرض شود که یک ماتریس بصورت row-major در حافظه ذخیره شده است و یک قطعه برنامه که قصد دارد مجموع ستونهای ماتریس را در یک بردار ذخیره کند.
  - روش اول: ماتریس 1000*1000

```
1  for (i = 0; i < 1000; i++)
2          column_sum[i] = 0.0;
3          for (j = 0; j < 1000; j++)
4                  column_sum[i] += b[j][i];
```



12

2/8/2014

دسترسی با گامهای بلند (strided access)

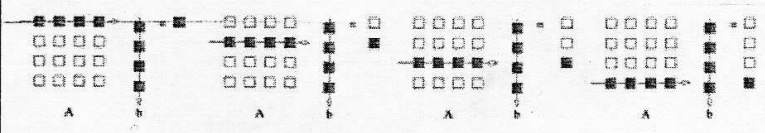
- روش اول: چون دسترسی به ماتریس بصورت ستون به ستون است و از طرفی ماتریس بصورت row-major ذخیره شده است، لذا دسترسی از یک عنصر به عنصر دیگر برابر 1000 گام است و این به این صورت است که برای هر عنصر بایستی یک بلاک (یک سطر از ماتریس) در کاشه بارگزاری شود. لذا اگر کاشه گنجایش فقط یک بلاک را داشته باشد، ۱۰۰۰۰۰۰ (بارگزاری) دسترسی به حافظه را داریم.

13

کاهش دسترسی با گامهای بلند (strided access)

- روش دوم: کد برنامه را بصورت زیر تغییر می دهیم:

```
1  for (i = 0; i < 1000; i++)
2          column_sum[i] = 0.0;
3  for (j = 0; j < 1000; j++)
4          for (i = 0; i < 1000; i++)
5                  column_sum[i] += b[j][i];
```



14

در روش دوم تغییر داده در این حالت ۱۰۰۰ تا سطر می خواند و همه شاخص ها در محاسبات شرکت می کنند در این حالت از کش کوچک باید در محاسبات استفاده کرد.

روشهای دیگر برای مرتفع کردن تاخیر حافظه

- کاهش عدم پاسخ مرورگر وب در پیک ترافیک شبکه:
  - پیش بینی صفحه های وب مورد نظر و مطالب مورد درخواست برای آنها
    - پیش واکشی (prefetching)
  - بازکردن چندی مرورگر وب و دیدن یک صفحه در هرکدام، بطوریکه اگر یکی از مرورگرها در حال بارگزاری یک صفحه است، صفحات دیگر در مرورگرهای دیگر می تواند خوانده شود.
    - چند نخی (multithreading)
  - در یک حرکت چندین صفحه را در یک مرورگر باز شود.
    - محلیت فضایی (spatial locality)

15

چند نخی برای مرتفع کردن تاخیر حافظه

- تعریف نخ: یک تک جریان کنترلی در جریان یک برنامه را نخ گویند.
- مثال: ضرب یک ماتریس a(n*n) در بردار b برای حصول بردار c:

```
1  for(i=0;i<n;i++)
2     c[i] = dot_product(get_row(a, i), b);
```

- چون هر عمل ضرب داخلی در کد فوق مستقل از یکدیگر هستند، بنابراین می توان کد فوق را بصورت زیر نوشت:

```
1 for(i=0;i<n;i++)
2    c[i] = create_thread(dot_product, get_row(a, i), b);
```

- در سیکل اول تابع dot_product به یک جفت بردار از عناصر دسترسی پیدا می کند و منتظر دریافت داده ها می ماند. در سیکل بعدی این تابع به جفت بردار دیگر از عناصر دسترسی پیدا می کند و الی آخر. اولین تابع بعد از x واحد زمانی (تاخیر حافظه) داده ها را بدست می آورد و الی آخر. بعبارتی در هر سیکل کلاک یک محاسبه انجام می شود.

8

2/8/2014

پیش واکشی برای مرتفع کردن تاخیر حافظه

- اگر کاشه وجود نداشته باشد، پردازنده باید برای ضرب a[i] در b[i] یک درخواست به حافظه برای دریافت جفت داده (عنصر) دهد.
  - فرض شود هر درخواست یک سیکل کلاک (۱ نانوثانیه) را لازم داشته باشد، بنابراین برای دو بردار ۱۰۰ عنصری ۱۰۰ نانوثانیه زمان صرف می شود.
- پیش واکشی: با وجود کاشه، پردازنده می تواند بجای بارگزاری یک جفت داده، یک جفت بردار را بارگزاری کند!

- تمرین: کدام روش کارایی بیشتری را در مرتفع کردن تاخیر حافظه دارد؟

17

سازماندهی بسترهای پردازش موازی

- سازمان فیزیکی: سازمان سخت افزاری واقعی بستر است.
- سازمان منطقی: دید برنامه نویس نسبت به بستر محاسباتی.
- دو جزء مهم محاسبات موازی از دید برنامه نویس:
  - تصریح وظائف موازی ⟸ ساختار کنترلی
  - مکانیزمی برای تعیین محاوره بین این وظائف ⟸ مدل ارتباطی

18

ساختار کنترلی بسترهای محاسبات موازی

- وظائف موازی در سطوح مختلفی تعیین می شوند:
  - هر برنامه در یک مجموعه از برنامه ها می تواند بعنوان یک وظیفه موازی دیده شود.
  - دستورالعملهای منحصری در یک برنامه می توانند بعنوان وظائف موازی دیده شوند.
- بین سطوح فوق محدوده ایی از مدلها برای تعیین ساختار کنترلی و هر کدام یک معماری که آنها را پشتیبانی کند، وجود دارد.
- مثال: موازی سازی یک دستورالعمل روی چندین پردازنده

```
1  for (i = 0; i < 1000; i++)
2          c[i] = a[i] + b[i];
```

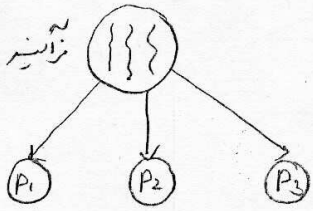
- چون هر جمع روی هر جفت عنصر کاملا از جمع های دیگر مستقل انجام می شود، لذا اگر مکانیزمی وجود داشته باشد که یک دستور با داده های مختلف روی همه پردازنده ها همزمان اجرا شود، حلقه فوق با سرعت خیلی بالایی انجام می شود. بنابراین یک معماری برای پیاده سازی این مکانیزم لازم است.

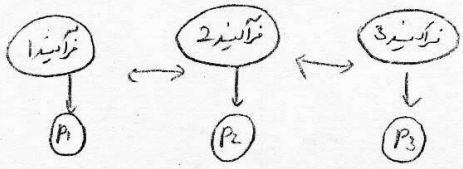
19

فرض می کنیم سه تا پردازنده داشته باشیم

3 تا مزانسیر داریم که در صورت موازی شدن این پردازنده ها اجرا شوند

می توانیم یک رشته را در صورت چندتا task در نظر بگیریم یعنی در صورت چندتایی



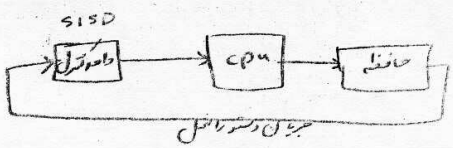


2/8/2014

فصل دوم
معماری های پردازش موازی

چهار دهه از محاسبات (four decades of computing)

- Batch era (1965)
- Time sharing era (1970)
- Desktop era (1977)
- Network era (1980-90)
- Current trends
  - Cluster of workstations
  - Grid computing
  - Cloud computing



2/8/2014

طبقه بندي flynn (flynn's taxonomy)

- جريانهاي اطلاعاتي در پردازنده ها را به ۲ دسته تقسیم کرد.

۱- جريان دستورالعمل (instruction stream) : دنباله ايي از دستورالعملها که توسط پردازنده اجرا مي شوند.

۲- جريان داده ها (data stream) : ترافيکي از داده ها که بين پردازنده و حافظه تبادل مي شوند.

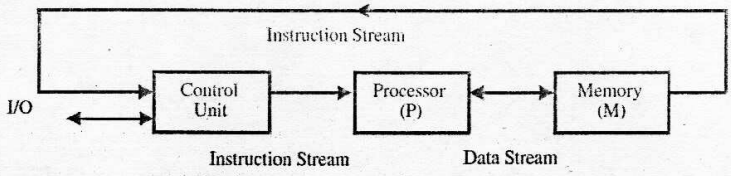
طبقه بندي flynn (flynn's taxonomy)

- طبقه بندي معماري کامپيوتر

– Single-Instruction Single-Data (SISD)
– Single-Instruction Multiple-Data (SIMD)
– Multiple-Instruction Single-Data (MISD)
– Multiple-Instruction Multiple-Data (MIMD)

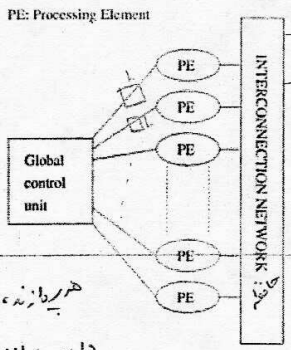
Single-Instruction Single-Data (SISD)

- کامپیوترها Van Neumann هستند.
- معماري SISD :



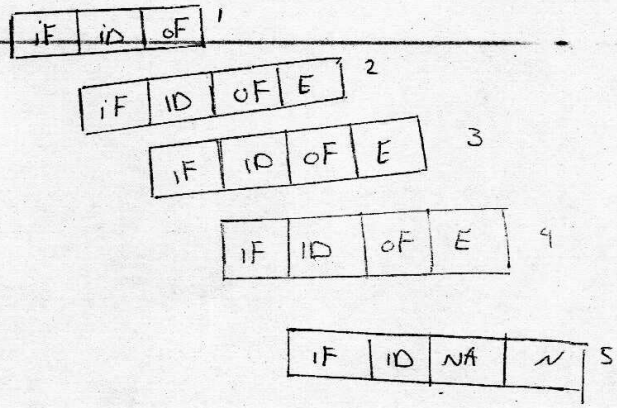
معماری SIMD

- Single Instruction-Multiple Data (SIMD)
  - یک واحد کنترل دستورالعملها را به هر واحد پردازشی توزیع می کند.
  - هر دستوالعمل بطورهمزمان توسط همه واحدهای پردازشی اجرا می شود.
  - مناسب برای محاسبات ساخت یافته روی ساختارهای داده ایی موازی نظیر آرایه ها است.
  - نیاز به "activity mask" دارد که مشخص شود کدام داده و عمل در عملیات شرکت داشته باشد یا خیر.
  - اجراهای شرطی می تواند کارای SIMD را کاهش دهند، لذا در استفاده از آنها باید دقت شود. (مثال اسلاید بعد را ببینید).



ادامه صفحه سوم 3

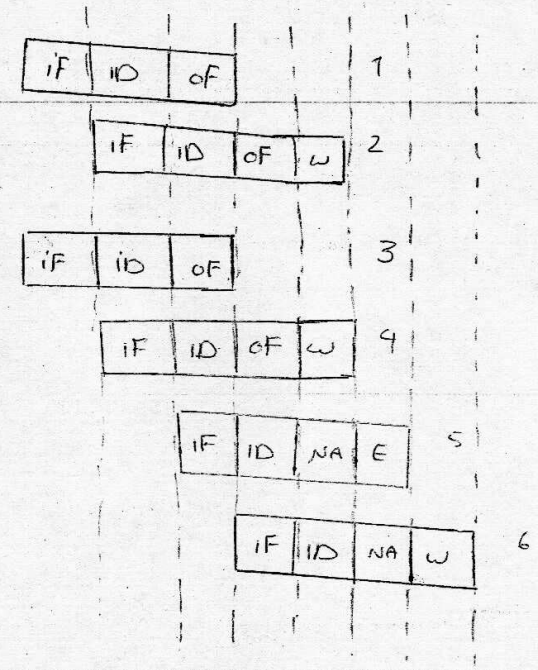
کد دوم به صورتی نوشته نشده که از pipeline هم استفاده نشده و pipeline به صورت موازی اجرا نمی‌شود.



زمان = 8T

کد سوم از هر دو pipeline استفاده کرده

pipeline اول

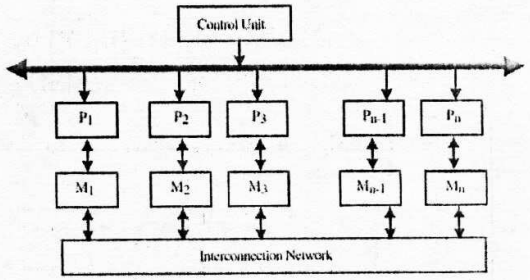


pipeline دوم

زمان = 7T

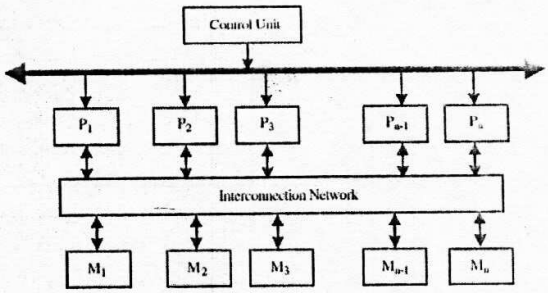
2/8/2014

انواع پیکربندي SIMD



- هر پردازنده حافظه محلي خود را دارد.
- پردازنده ها از طريق IN با هم در ارتباطند.
- اگر IN ارتباط بين دو پردازنده را مستقيماً برقرار نكند ، ارتباط از طريق پردازنده واسط انجام مي شود.

انواع پیکربندي SIMD



۱. • پردازنده ها و حافظه ها از طريق IN با هم در ارتباطند.

۲. • دو پردازنده مي توانند از طريق ماژول حافظه يا پردازنده واسط با هم ارتباط برقرار كنند.

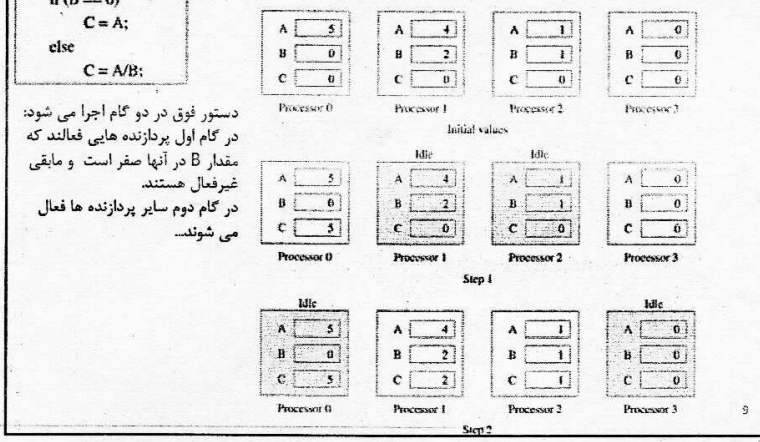
2/8/2014

کارایی در معماری SIMD

- مثال: کاهش کارایی معماری SIMD در اجرای دستورات شرطی:

```
if (B == 0)
    C = A;
else
    C = A/B;
```

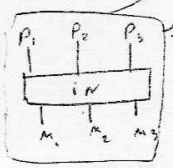
دستور فوق در دو گام اجرا می شود:
در گام اول پردازنده هایی فعالند که مقدار B در آنها صفر است و مابقی غیرفعال هستند.
در گام دوم سایر پردازنده ها فعال می شوند...



9

Multiple-Instruction Multiple-Data (MIMD)

- چندین پردازنده و چندین ماژول حافظه از طریق برخي IN بهم متصل مي شوند.
  1. حافظه مشترک (Shared Memory)
  2. انتقال پیام (Message Passing)

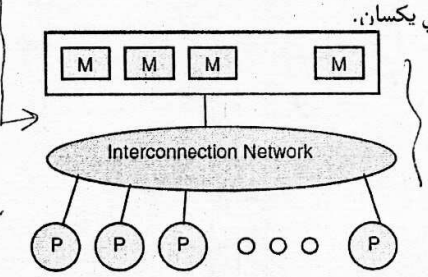


5

2/8/2014

معماري Shared Memory MIMD

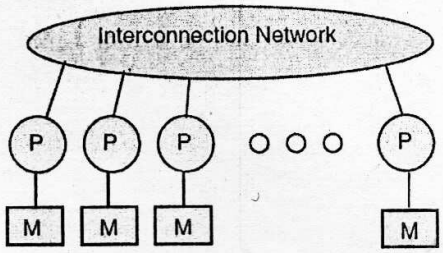
- ارتباط بين پردازنده ها از طريق حافظه مشترک صورت مي گيرد.
- هر پردازنده شانس دسترسي يکساني را براي خواندن/نوشتن حافظه دارد و همچنين سرعت دسترسي يکسان.



Shared Memory MIMD Architecture

معماري Message Passing MIMD

- در هر گره از IN ، هر پردازنده به حافظه محلي خود متصل است.
- داده ها از يک حافظه محلي به ديگري از طريق انتقال پيام صورت مي گيرد.
- انتقال پيام از طريق send/recieve صورت مي گيرد.
- گسترش پذير است بدليل توزيع شدن حافظه ها (ماژولها) و پردازنده ها



Message Passing MIMD Architecture

2/8/2014

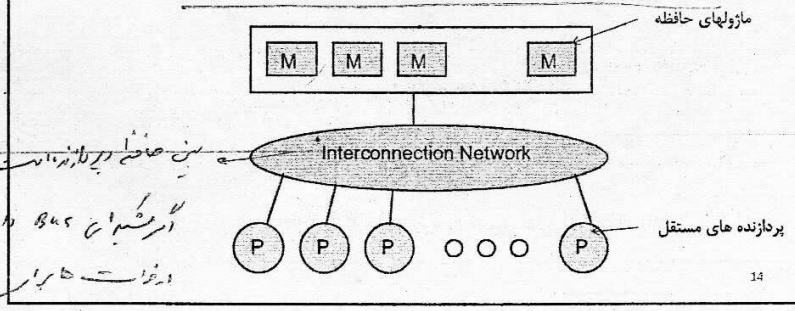
معماري Distributed-Shared Memory (DSM)

- تركيبي است از مزاياي حافظه مشترک و انتقال پيام.
- سخت افزار انتقال پيام را پشتيباني مي كند (حافظه ها توزيع شده اند)
- مدل برنامه نويسي تفكر حافظه مشترک است.

مدل ارتباطی بسترهای موازی

بسترهایی با فضای آدرس مشترک (حافظه مشترک)

- سیستم (معماری) حافظه مشترک از رده های مهم چند پردازنده ها است که همه پردازنده ها یک حافظه را به اشتراک می گذارند.
- تبادل داده ها (ارتباط) بین taskها روی پردازنده های مختلف از طریق خواندن/نوشتن روی حافظه اشتراکی صورت می گیرد.



14

7

2/8/2014

مدل ارتباطی بسترهای موازی
ویژگی های بسترهایی با فضای آدرس مشترک

1. برنامه نویسی در فضای آدرس مشترک، ساده است.
   - همه تبادلات فقط خواندنی (read-only) از دید برنامه نویس پنهانند و کد نویسی کاملا مشابه با کد نویسی سریال است.
   - در تبادلات خواندنی/نوشتنی (read/write) کد نویسی کمی مشکلتر خواهد بود که نیاز به انحصار متقابل برای دسترسی های همزمان است.
2. وجود کاشه در هر پردازنده مجادله برای دسترسی پردازنده ها به حافظه مشترک را کاهش داده است.
   - مساله همدوسی کاشه ها (cache coherency) ایجاد شده است : کلیه کپی ها در کاشه ها بایستی یکسان باشند.

15

مدل ارتباطی بسترهای موازی
طراحی بسترهایی با فضای آدرس مشترک (حافظه مشترک)

- مسائلی که باید در طراحی یک حافظه مشترک باید در نظر گرفته شود:
  1. – کاهش کارایی به دلیل مجادله (contention)
  2. – مسائل ارتباطی (همدوسی) (coherence)
- مجادله (contention): چندین پردازنده برای دسترسی به حافظه مشترک همزمان درخواست دهند.
  - وجود کاشه ممکن است مسئله مجادله را تا حدی حل کند، ولی داشتن چندین کپی از داده ها که بین کاشه ها منتشر شده اند، مسئله **همدوسی** را ایجاد می کند.
- همدوسی (coherency): کپی ها در کاشه ها همدوس هستند اگر مقادیر آنها با یکدیگر یکسان باشد. حال اگر یک پردازنده در کاشه خود تغییراتی را ایجاد کند، آن کاشه ناسازگار (inconsistent) است.

16

2/8/2014

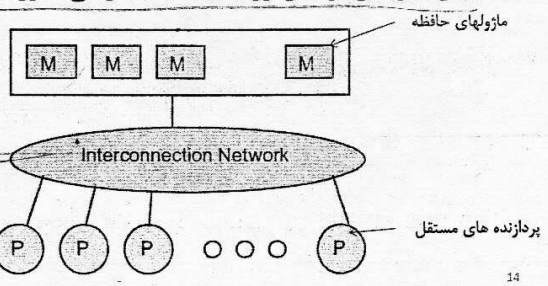
معماري Distributed-Shared Memory (DSM)

- ترکیبي است از مزاياي حافظه مشترک و انتقال پيام.
- سخت افزار انتقال پيام را پشتيباني مي کند (حافظه ها توزيع شده اند)
- مدل برنامه نويسي تفکر حافظه مشترک است.

مدل ارتباطی بسترهای موازی

بسترهایی با فضای آدرس مشترک (حافظه مشترک)

- سیستم (معماری) حافظه مشترک از رده های مهم چند پردازنده ها است که همه پردازنده ها یک حافظه را به اشتراک می گذارند.
- تبادل داده ها (ارتباط) بین taskها روی پردازنده های مختلف از طریق خواندن/نوشتن روی حافظه اشتراکی صورت می گیرد.

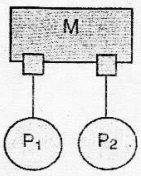


14

2/8/2014

مدل ارتباطی بسترهای موازی
طبقه بندی سیستم های حافظه مشترک

- ساده ترین سیستم حافظه مشترک شامل یک ماژول حافظه (M) و دو پردازنده P1 و P2 است:



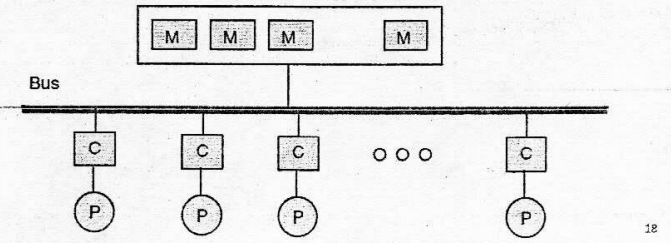
Shared memory via two ports.

- یک واحد داوری درون حافظه وجود دارد که سیگنالهای grant یا busy را از طریق کنترل کننده حافظه به پردازنده ها ارسال می کند.

17

مدل ارتباطی بسترهای موازی
طبقه بندی سیستم های حافظه مشترک بر اساس شبکه ارتباطی

- UMA (Uniform Memory Access)
  - دسترسی به حافظه مشترک توسط پردازنده ها از طریق IN به همان صورتی است که یک پردازنده به حافظه خودش دسترسی دارد.
  - همه پردازنده ها زمان یکسانی را برای دسترسی به هر موقعیت از حافظه دارند.
  - IN استفاده شده در UMA می تواند Single Bus، Multiple Bus و یا یک Crossbar Switch باشد.
  - چون دسترسی ها به حافظه مشترک بالانس است، این نوع سیستمها به Symmetric Multiprocessor-SMP نیز معروفند.



18

2/8/2014

مدل ارتباطی بسترهای موازی

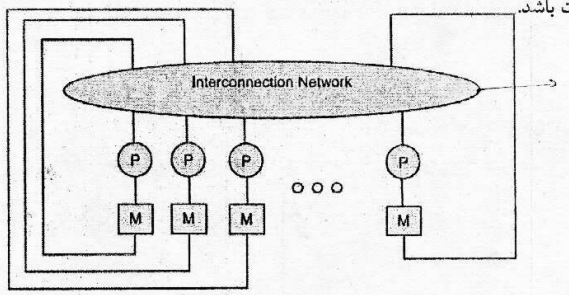
طبقه بندی سیستم های حافظه مشترک بر اساس شبکه ارتباطی

– NUMA (Nonuniform Memory Access)

– هر پردازنده به بخشی از حافظه مشترک دسترسی دارد.

– حافظه دارای یک فضای آدرس دهی است.هر پردازنده با آدرس واقعی به هر موقعیت از حافظه دسترسی دارد. بنابراین زمان دسترسی به هر ماژول حافظه بستگی به فاصله آن از پردازنده دارد.

– IN های استفاده شده در این رده می توانند گذرگاه های چند سطحی (سلسله مراتبی) و یا درخت باشد.



19

مدل ارتباطی بسترهای موازی

طبقه بندی سیستم های حافظه مشترک بر اساس شبکه ارتباطی

– COMA (Cache-Only Memory Architecture)

– همانند NUMA است ولی حافظه مشترک شامل حافظه کاشه است.

– داده ها از کاشه یک پردازنده به کاشه دیگر منتقل می شوند.

20

2/8/2014

مدل ارتباطی بسترهای موازی
موضوعاتی که باید در سیستمهای حافظه مشترک در نظر گرفت

۱. کنترل دسترسی (access control): بایستی درخواستهای پردازنده ها برای دسترسی به حافظه مشترک کنترل شود تا برخوردی (conflict) صورت نگیرد.
۲. همزمان سازی (synchronization): زمان دسترسی فرایندهای مشترک را به منابع مشترک محدود می کند.
۳. حفاظت (protection): قابلیتی از سیستم است که دسترسی غیرمجاز فرایندها را به منابعی که در اختیار سایر فرایندها است، مانع می شود.

مدل ارتباطی بسترهای موازی
بسترهای انتقال پیام (message passing platforms)

۱. هر پردازنده دارای حافظه خاص خودش است و ارتباط بین پردازنده ها فقط از طریق انتقال پیام صورت می گیرد.
۲. هیچ حافظه مشترکی بین پردازنده ها وجود ندارد.
۳. دو عمل اصلی send و receive در این نوع بسترها جهت تبادل داده ها نقش اصلی را دارند.

22

2/8/2014

مدل ارتباطی بسترهای موازی
بسترهای انتقال پیام (message passing platforms)

۱. • مدل: گره ها از طریق send/recieve با هم در ارتباطند (حافظه مشترک وجود ندارد).

۲. • گره: شامل پردازنده و حافظه محلی برای ذخیره کردن پیامها جهت ارسال یا دریافت.

۳. • فاکتورهای مهم در طراحی IN در سیستمهای انتقال پیام:

۱ – پهنای باند (link bondwidth): تعداد بیتهایی که می توانند در واحد زمان انتقال داده شوند.

۲ – تاخیر لینک (link latency): مدت زمانی که انتقال پیام کامل می شود.

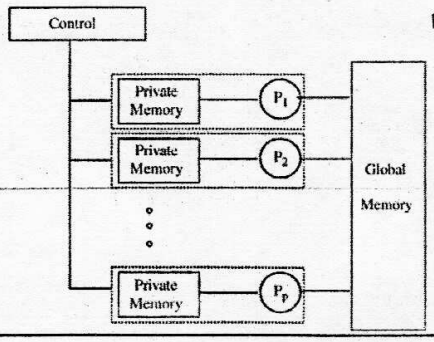
سازمان فیزیکی بسترهای موازی

• معماری ایده ال یک کامپیوتر موازی:

– Parallel Random Access Machine, PRAM

• P پردازنده فعال در یک سیکل همزمان عمل خواندن، محاسبه و نوشتن را انجام می دهند.

• دستورالعملها یکسان روی دا



24

2/8/2014

حالت های مختلف برای خواندن و نوشتن در PRAM

- خواندن انحصاری (ER): در یک زمان فقط یک پردازنده می تواند از هر مکان از حافظه بخواند.
- نوشتن انحصاری (EW): در یک زمان فقط یک پردازنده می تواند در هر مکان از حافظه بنویسد.
- خواندن همزمان (CR): چند پردازنده می توانند همزمان از یک مکان حافظه بخوانند.
- نوشتن همزمان (CW): چند پردازنده می توانند همزمان در یک مکان حافظه بنویسند.

25

حالت های مختلف برای خواندن و نوشتن در PRAM

- تداخل در نوشتن همزمان بایستی با استفاده از رویه زیر حل شود:
  - عمومی (Common): تمامی نوشته های همزمان مقدار یکسانی را ذخیره می کنند.
  - اختیاری (Arbitrary): تنها یک مقدار دلخواه انتخاب شده و ذخیره می شود. از بقیه مقادیر صرفه نظر می شود.
  - کمینه (Minimum): مقدار نوشته شده توسط پردازنده با کوچکترین شاخص ذخیره می شود و از بقیه مقادیر صرفه نظر می شود.
  - کاهش (Reduction): تمامی مقادیر با استفاده از توابع کاهش مثلا جمع، مینیمم، ماکزیمم و مانند آنها به یک مقدار کاهش می یابند.

26

فصل سوم
شبکه اتصال درونی چند پردازنده ها
Multi Processor Interconnection Network

1

شبکه های اتصالی برای کامپیوترهای موازی

- یک شبکه اتصالی مکانیزمی را برای انتقال داده ها بین گره های پردازشی و یا بین پردازنده ها و حافظه ها فراهم می کند.
- هر شبکه اتصالی شامل n ورودی و m خروجی است.
- شبکه های اتصالی یا از لینک (link) و یا از سویچ (switch) ساخته می شوند.
  - لینک: یک رسانه فیزیکی نوعاً از سیم یا کابلهای فیبر نوری ساخته می شود.

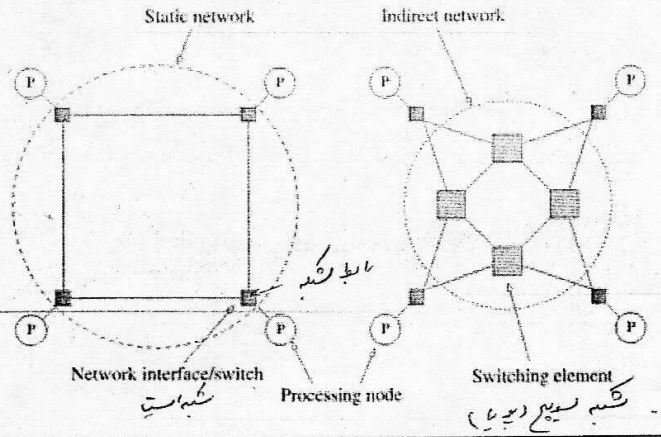
2

انواع شبکه های اتصالی

- شبکه ایستا (Static) یا مستقیم (Direct): شامل لینک های ارتباطی نقطه-به-نقطه بین گره های پردازشی است.
- شبکه پویا (Dynamic) یا غیر مستقیم (Indirect): شامل لینک های ارتباطی و سویچها است. یک لینک به لینک دیگر بصورت پویا از طریق سویچ متصل شده و یک مسیری بین گره های پردازشی و حافظه-ها بنا نهاده می شود.

3

انواع شبکه های اتصالی



2

سویچ در شبکه های اتصالی

- شامل پورت های ورودی و خروجی است.
- حداقل عملکرد سویچ: نگاشت پورت ورودی به پورت خروجی است.
- تعداد پورتهای یک سویچ، درجه (Degree) آن سویچ گفته می شود.
- سویچها ممکن است برخی عملیات داخلی انجام دهند، شامل : بافرینگ (زمانیکه پورت خروجی درخواست شده اشغال باشد)، مسیریابی (برای کاهش ازدحام در شبکه) و چندپخشی (خروجی یکسان روی چند پورت).

5

رابط شبکه در شبکه های اتصالی

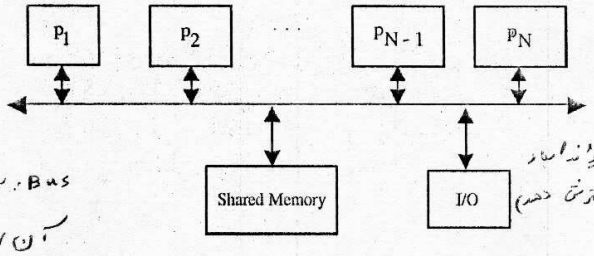
- ارتباط بین گره ها با شبکه را فراهم می کند.
- دارای پورت های ورودی و خروجی است که داده ها را به درون و بیرون شبکه هدایت می کند.
- شامل وظائفی نظیر:
  - بسته بندی داده ها (Packetizing data)
  - آنالیز اطلاعات مسیریابی
  - بافر کردن داده های ورودی و خروجی برای همسان سازی سرعت شبکه با گره های پردازشی.
  - خطایابی.

6

2/8/2014

شبکه های اتصالی مبتنی بر گذرگاه
Bus-based IN

- سیستمهای تک گذرگاه
  - هر پردازنده دارای کاشه خاص خود است.
  - اندازه این سیستم ها بین ۲ تا ۵۰ پردازنده است.
  - پیچیدگی شبکه ایی: برای سیستمهای مبتنی بر گذرگاه بر اساس تعداد گذرگاها است = O(1)
  - پیچیدگی زمانی: با میزان تاخیر ورودی به خروجی تعیین می شود = O(N)



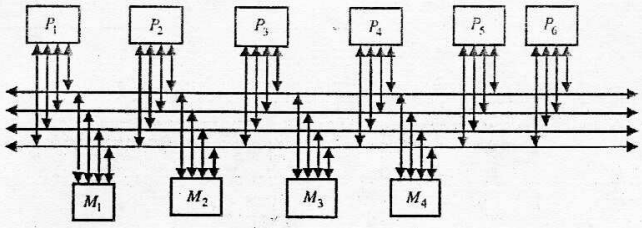
شبکه های اتصالی مبتنی بر گذرگاه
Bus-based IN

- سیستمهای چند گذرگاه
  - توسعه سیستمهای تک گذرگاه است.
  - از چندی گذرگاه موازی برای اتصال پردازنده ها و ماژولهای حافظه استفاده می شود.

شبکه های اتصالی مبتنی بر گذرگاه

انواع سیستمهای چند گذرگاه

- Multiple Bus with Full Bus-Memory Connection (MBFBMC)
  - N=6 (Processors)
  - M=4 (Memories)
  - B=4 (Buses)

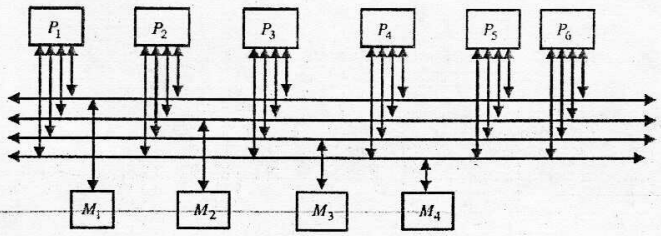


9

شبکه های اتصالی مبتنی بر گذرگاه

انواع سیستمهای چند گذرگاه

- Multiple Bus with Single Bus-Memory Connection (MBSBMC)
  - N=6 (Processors)
  - M=4 (Memories)
  - B=4 (Buses)

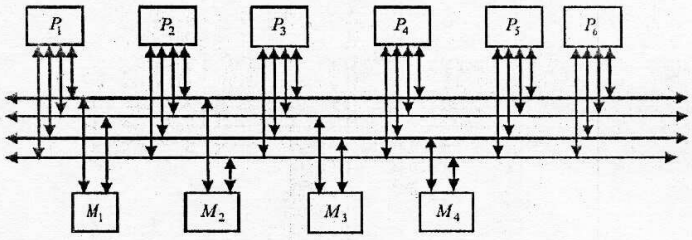


10

2/8/2014

شبکه های اتصالی مبتنی بر گذرگاه
انواع سیستمهای چند گذرگاه

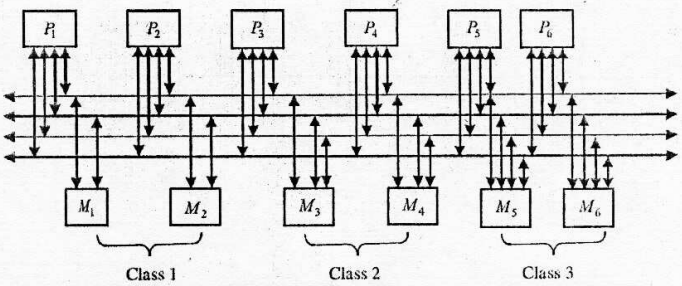
- Multiple Bus with Partial Bus-Memory Connection (MBPBMC)
  - N=6 (Processors)
  - M=4 (Memories)
  - B=4 (Buses)



11

شبکه های اتصالی مبتنی بر گذرگاه
انواع سیستمهای چند گذرگاه

- Multiple Bus with Class-based Memory Connection (MBCBMC)
  - هر کلاس از حافظه به زیر مجموعه ایی از گذرگاه ها متصل هستند
  - هر کلاس داوری خاص خود را دارد.



12

2/8/2014

شبکه های اتصالی مبتنی بر گذرگاه
مشخصه های معماریهای گذرگاه چندگانه

- K: تعداد کلاسها
- g: تعداد گذرگاه ها در هر کلاس
- Mj: تعداد ماژولهای حافظه در کلاس j

Connection Type	No. of Connections	Load on Bus i
MBFBMC	$B(N+M)$	$N+M$
MBSBMC	$BN+M$	$N+M_j$
MBPBMC	$B(N+M/g)$	$N+M/g$
MBCBMC	$BN+\sum_{j=1}^{k} M_j(j+B-k)$	$N+\sum_{j=\max(i+k-B,1)}^{k} M_j,\ 1 \le i \le B$

13

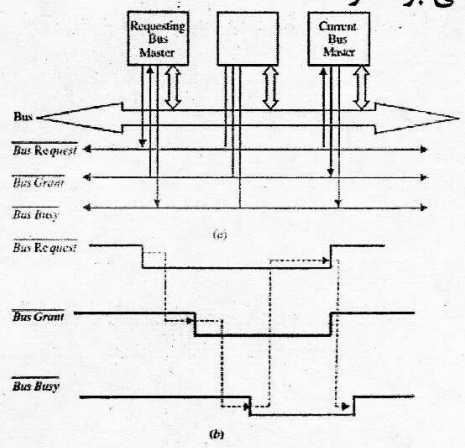
شبکه های اتصالی مبتنی بر گذرگاه
همزمان سازی گذرگاه

- انواع گذرگاه:
  1. همزمان (Synchronous): زمان برای هر انتقال مشخص است. هر دیوایس برای پذیرش یا تولید اطلاعات روی گذرگاه زمان مشخصی را دارد.
  2. غیرهمزمان (Asynchronous): مبتنی بر وجود داده ها و آمادگی دیوایسها برای انتقال داده است.
- در سیستمهای تک گذرگاه یک داوری گذرگاه (bus arbitration) نیاز است، بطوریکه هر پردازنده برای دسترسی به گذرگاه، درخواستی را به منطق داوری ارسال می کند. این عمل برای جلوگیری از مساله مجادله گذرگاه (bus contention) صورت می گیرد.
- درخواست پردازنده به منطق داوری و دریافت پاسخ، Handshaking نام دارد و توسط دو سیگنال انجام می شود:
  - Bus request: درخواست دسترسی به گذرگاه.
  - Bus grant: پذیرش درخواست. (عدم پذیرش درخواست: Bus busy)

14

2/8/2014

شبکه های اتصالی مبتنی بر گذرگاه
همزمان سازی گذرگاه



Bus handshaking mechanism (*a*) the scheme; and (*b*) the timing.

15

شبکه های اتصالی مبتنی بر گذرگاه
همزمان سازی گذرگاه

- انواع داوری:
  - Simple rotation priority: همه سطوح اولویت یک سطح کاهش پیدا می کنند تا اینکه پردازنده با اولویت پایین در اولویت بالا قرار گیرد. (مکانیزم ؟؟؟)
  - Equal priority: وقتی دو یا چند درخواست صورت پذیرد، همه درخواستها دارای شانس یکسانی دارند. برای هر پردازنده، شانس هر کدام 1/n است. (مکانیزم ؟؟؟)
  - Least recently used-LRU: اولویت بالاتر برای پردازنده ای است که در زمان طولانی تر از گذرگاه استفاده نکرده است.

16

2/8/2014

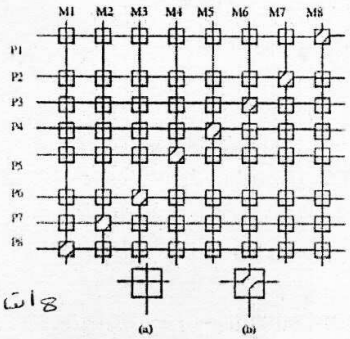
شبکه های اتصالی های مبتنی بر سویچ
Switch-based IN

- ارتباط بین پردازنده ها و حافظه ها از طریق سویچ صورت می گیرد.
- سه نوع توپولوژی وجود دارد:
  - Crossbar تقاطعی
  - Single-stage تک طبقه
  - Multi-stage چند طبقه

17

شبکه های اتصالی های مبتنی بر سویچ
شبکه های Crossbar

- نقطه مقابل گذرگاه، Crossbar است.
- گذرگاه فقط یک اتصال را برقرار می کند، درحالیکه Crossbar ارتباطات موازی را بین همه ورودیها و خروجی هایش برقرار می کند.



The figure illustrates the case of setting the SEs such that simultaneous connections between P_i and M_{8-i+1} for $1 \le i \le 8$ are made.

پیچیدگی شبکه از لحاظ تعداد سویچها: O(N^2)
پیچیدگی شبکه از لحاظ تاخیر ورودی به خروجی: O(1)
Crossbar یک شبکه غیربلوکه است.

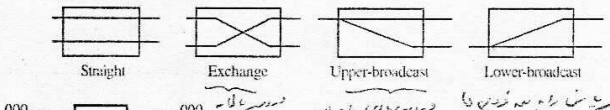
An 8 × 8 crossbar network (*a*) straight switch setting; and (*b*) diagonal switch setting.

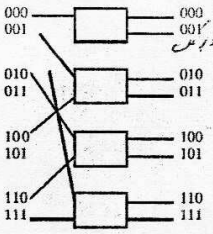
18

2/8/2014

شبکه های اتصالی های مبتنی بر سویچ
شبکه های Single-stage

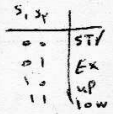
- یک طبقه از SEها بین ورودیها و خروجی ها قرار دارد.
- ساده ترین SEها که در این شبکه استفاده می شود، SEهای ۲*۲ است.

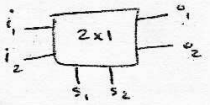




- برای ارتباط بین یک مبدا و یک مقصد داده ها بایستی چند بار شبکه را دور بزنند.

19





شبکه های اتصالی های مبتنی بر سویچ
شبکه های Single-stage

- دو عمل استفاده می شود:

Shuffle:

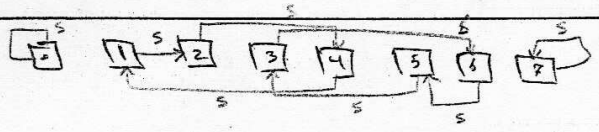
Exchan: $S(p_{m-1}p_{m-2}\cdots p_1p_0) = p_{m-2}p_{m-3}\cdots p_1p_0p_{m-1}$

$E(p_{m-1}p_{m-2}\cdots p_1p_0) = p_{m-1}p_{m-2}\cdots p_1\overline{p_0}$

- اگر تعداد ورودیها (پردازنده ها) برابر با N و تعداد خروجی ها (حافظه ها) برابر N باشد، تعداد SEها در یک طبقه برابر N/2 است.
- ماکزیمم طول یک مسیر از یک ورودی به یک خروجی در شبکه بر اساس تعداد SEها در طول مسیر محاسبه می شود و برابر است با LogN

- تمرین: طراحی شبکه Shuffle-Exchange برای N=8

20



2/8/2014

شبکه های اتصالی های مبتنی بر سویچ
شبکه های Single-stage

- مثال: در یک Shuffle-Exchange با ۸ ورودی، اگر مبدا (000) باشد، و مقصد (110) باشد، آنگاه دنباله Shuffle-Exchange و چرخش داده ها بصورت زیر

$E(000) \rightarrow 1(001) \rightarrow S(001) \rightarrow 2(010) \rightarrow E(010) \rightarrow 3(011) \rightarrow S(011) \rightarrow 6(110)$

پیچیدگی شبکه: O(N)
پیچیدگی زمانی: O(N)

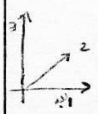
21

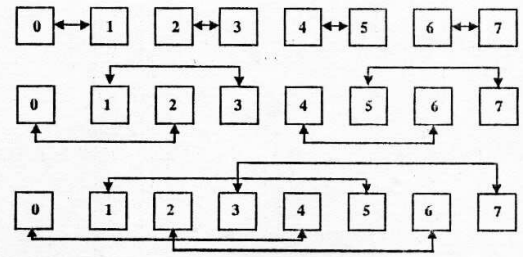
شبکه های اتصالی های مبتنی بر سویچ
شبکه های Single-stage

- شبکه مکعب (Cube network):
  - الگوی اتصالی برای شبکه مکعب بصورت زیر است:

$$C_i(p_{m-1}p_{m-2}\cdots p_{i+1}p_ip_{i-1}\cdots p_1p_0) = p_{m-1}p_{m-2}\cdots p_{i+1}\overline{p_i}p_{i-1}\cdots p_1p_0$$

- مثلا برای یک آدرس ۳ بیتی (N=8) داریم: $C_2(6) = 2, C_1(7) = 5 \text{ and } C_0(4) = 5$





22

2/8/2014

شبکه های اتصالی های مبتنی بر سویچ
شبکه های Single-stage

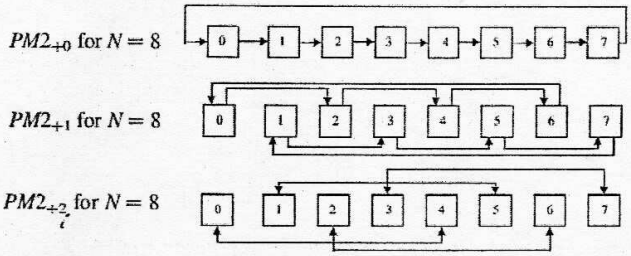
- شبکه *Plus-Minus* 2^i(*PM2I*)
- شامل 2K تابع ارتباطی است:
- K=LogN

$PM2_{+i}(P) = P + 2^i \bmod N (0 \le i < k)$

$PM2_{-i}(P) = P - 2^i \bmod N (0 \le i < k)$

- مثال: برای حالت N=8 داریم:

$PM2_{+1}(4) = 4 + 2^1 \bmod 8 = 6$



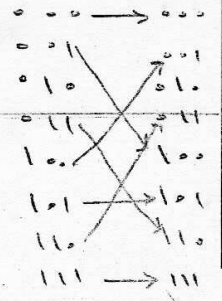
23

شبکه های اتصالی های مبتنی بر سویچ
شبکه های Single-stage

- شبکه Butterfly
- الگوی اتصالی برای شبکه Butterfly بصورت زیر است:

$B(p_{m-1}p_{m-2}\cdots p_1p_0) = p_0p_{m-2}\cdots p_1p_{m-1}$

B(000) = 000 B(010) = 010 B(100) = 001 B(110) = 011
B(001) = 100 B(011) = 110 B(101) = 101 B(111) = 111



24

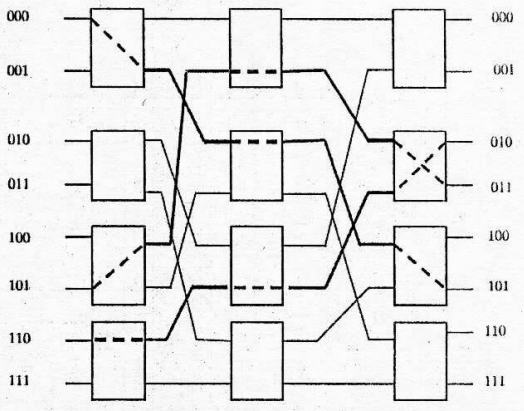
2/8/2014

شبکه های اتصالی های مبتنی بر سویچ
شبکه های Multi-Stage–(MN)

- فراهم کردن چندین مسیر همزمان بین پردازنده ها و حافظه ها
- در MNها مسیریابی بین مبدأ و مقصد بر اساس آدرس مقصد انجام می شود (self routing)
- در یک N*N MN، تعداد طبقات: Log(N)
- در یک N*N MN، تعداد بیتهای آدرس مقصد: Log(N)
- آدرس مقصد و تعداد طبقات از چپ به راست پیمایش می شود.
- اگر بیت iام برابر 0 باشد، آنگاه ورودی سویچ در طبقه iام به خروجی بالای آن وصل می شود.
- اگر بیت iام برابر 1باشد، آنگاه ورودی سویچ در طبقه iام به خروجی پایین آن وصل می شود.

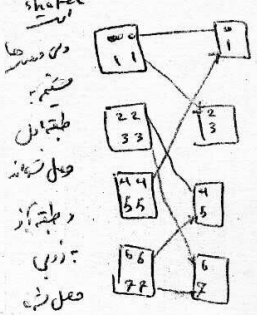
25

شبکه های اتصالی های مبتنی بر سویچ
شبکه Shuffle-Exchange



An example 8 × 8 Shuffle–Exchange network (SEN)

26



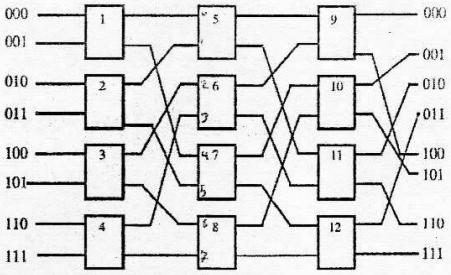
13

2/8/2014

شبکه های اتصالی های مبتنی بر سویچ

شبکه بانیان

- در N*N Banyan: تعداد SEها در هر طبقه: N/2
- در N*N Banyan: تعداد طبقه ها: Log(N)
- پیچیدگی شبکه ای به ازای تعداد کل SEها: O(Nlog(N))
- پیچیدگی زمانی به ازای تعداد SEها موجود در مسیر بین مبدأ و مقصد: O(log(N))
- مسیریابی؟

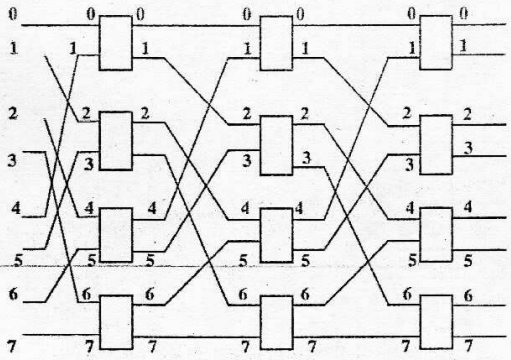


27

شبکه های اتصالی های مبتنی بر سویچ

شبکه امگا

- مسیریابی؟

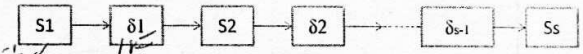


28

2/8/2014

شبکه های اتصالی های مبتنی بر سویچ
شبکه آرایش پذیر بنز (Benes)

- این امکان برای آرایش مجدد اتصالات از قبل ایجاد شده وجود دارد تا امکان ایجاد اتصالات جدید و همزمان وجود داشته باشد.
- شبکه آرایش پذیر Benes:
  - نحوه ایجاد اتصال:



- Si: سویچهای طبقه iام
- δi: اتصالات بین مرحله i و i+1
- S: تعداد طبقات و S≥3
- ni: تعداد ورودی/خروجی هر سویچ در طبقه iام به ازای i=1,2,…, (s+1)/2
- N: تعداد کل ورودیهای شبکه: ni≥2

$$\prod_{i=1}^{(\frac{s+1}{2})} n_i = N$$

29

شبکه های اتصالی های مبتنی بر سویچ
شبکه آرایش پذیربنز (Benes)

- مثال: اگر N=8 و s=5 باشد: i=1,2,3

$$\prod_{i=1}^{(3)} n_i = 8 \qquad n_i = 2 \rightarrow \begin{cases} n_1 = 2,\ n_2 = 2, n_3 = 2 \\ n_1 = 1,\ n_2 = 2, n_3 = 4 \end{cases}$$

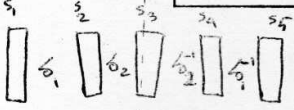
- ترکیب دوم درست نیست
- شرایط لازم برای شبکه آرایش پذیر:
  - شرط متقارن:

$$\delta_i = \delta_{s-i}^{-1} \qquad i = 1,2,3,\ldots,\frac{s-1}{2}$$

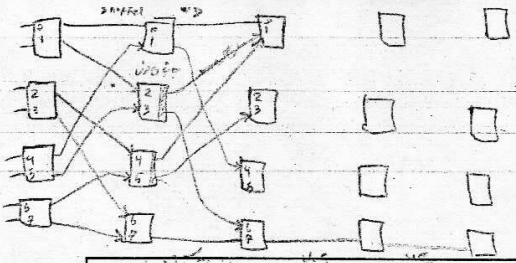
$$s_i = s_{s-i+1} \qquad i = 1,2,3,\ldots,\frac{s-1}{2}$$

δ_{s-i}^{-1} : معکوس اتصال δ_{s-i}^{-1} است.

30



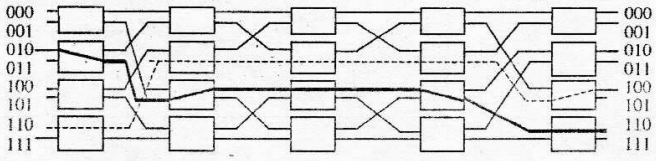
15



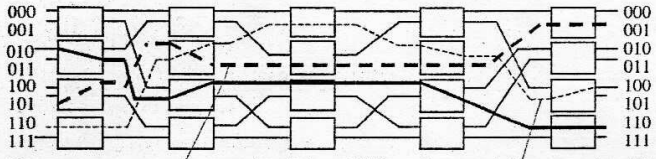
2/8/2014

شبکه های اتصالی های مبتنی بر سویچ
شبکه آرایش پذیر بنز (Benes)

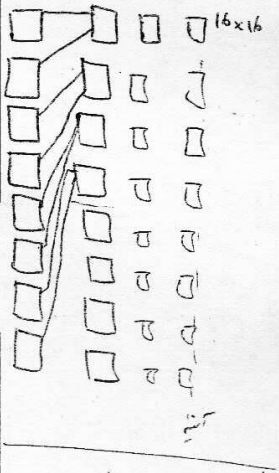
- نحوه اتصال نیمی از شبکه کافی است. $1 \le i \le (s-1)/2$
- مثال: برای s=5 و N=8 : $n_1 = n_2 = n_3 = 2$



Benes network with two simulatanceously established paths

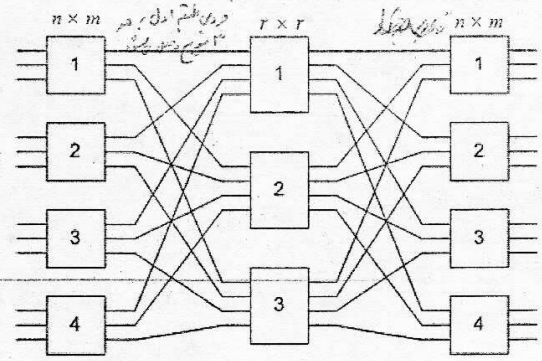


the rearrangement of connection 110 → 100 in order to satisfy connection 101 → 001



شبکه های اتصالی های مبتنی بر سویچ
شبکه غیر بلوکه کلاز (clos)

- شبکه Clos، (m , n , r)، آرایش پذیر است اگر m≥n باشد.
- مثال: r=4، n=3، m=3



32

شبکه های ایستا

- اتصالات ثابت یکطرفه یا دوطرفه بین پردازنده ها و حافظه ها.
- انواع شبکه های ایستا:
  - Completely Connected Networks-CCN
  - Limited Connected Networks-LCN

33

شبکه های ایستا

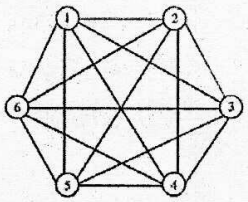
شبکه با اتصال کامل

Completely Connected Networks-CCN

- هر گره به کلیه گره های دیگر در شبکه متصل است.
- مزایا:
  - سرعت بالای انتقالات (فقط یک لینک باید پیمایش شود).
  - مسیریابی ساده است.
- معایب:
  - هزینه اتصالات بالا است: O(N^2)

    N: تعداد گره ها
  - پیچیدگی شبکه: O(N^2)
  - پیچیدگی زمانی: O(1)

تعداد اتصالات: $\frac{N(N-1)}{2}$



34

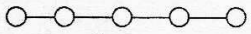
شبکه های ایستا
شبکه با اتصال محدود
Limited Connected Networks-LCN

- یک لینک مستقیم بین هر جفت گره وجود ندارد.
- مسیریابی بین دو گره از طریق گره های دیگر صورت می گیرد.
- طول یک مسیر بین دو گره برابر تعداد لینک هایی است که پیمایش می شوند.
- دو نیازمندی در LCNها وجود دارد:
  - نیاز به یک الگوی ارتباطی بین گره ها
  - نیاز به یک مکانیزم برای مسیریابی پیامها در شبکه.
- الگوهای ارتباطی:
  - Linear arrays
  - Ring (Loop) networks
  - Two Dimensional arrays (Mesh)
  - Tree networks
  - Cube networks

35

شبکه های ایستا
شبکه آرایه خطی (Linear Array)

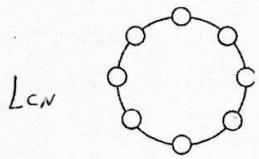
- هر گره به دو گره مجاورش متصل است (بجز گره های ابتدایی و انتهایی).
- برای ارتباط بین گره i وj ، (j>i)، پیام باید گره های $i+1, i+2, \ldots, j-i$ را پیمایش کند. برای (i>j)، داریم: $i-1, i-2, \ldots, i-j$
- بدترین حالت، ارتباط گره 1 با N است که N-1 گره باید پیمایش شود.
- مزایا: پیاده سازی و مسیریابی ساده است.
- معایب: شبکه کند است.
- پیچیدگی شبکه: O(N)
- پیچیدگی زمانی: O(N)



36

شبکه های ایستا
شبکه حلقوی (Ring)

- با اتصال گره ابتدایی و انتهایی شبکه Linear array، شبکه Ring شکل می گیرد.



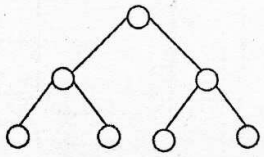
37

شبکه های ایستا
شبکه درختی (Tree)

- برای ارتباط بین گره iام با گره سطح jام، (i>j)، پیام باید گره های سطوح $i-1, i-2, \ldots, j+1$ را پیمایش کند.
- تعداد پردازنده ها در یک شبکه با K سطح:

$$N(k) = 2^0 + 2^1 + 2^2 + \cdots + 2^k = \frac{(2^k - 1)}{2 - 1} = 2^k - 1$$

- عمق درخت: $\lceil \log_2^N \rceil$
- پیچیدگی شبکه: $O(2^K)$
- پیچیدگی زمانی: $O(\log_2^N)$

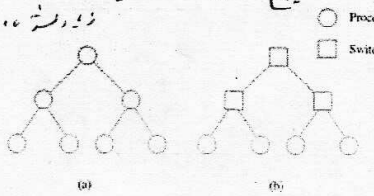


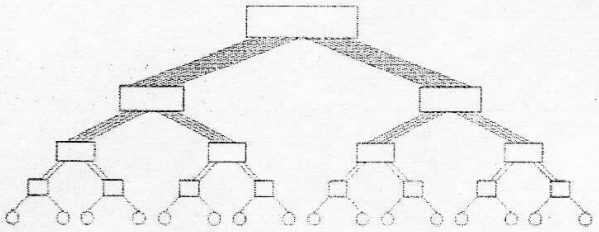
38

2/8/2014

شبکه های ایستا
شبکه درختی (Tree)

- شبکه هاي مبتني بر درخت
  - شبکه درخت ایستا (a)
  - شبکه درخت پویا (b)
  - شبکه هاي fat tree ©

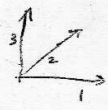


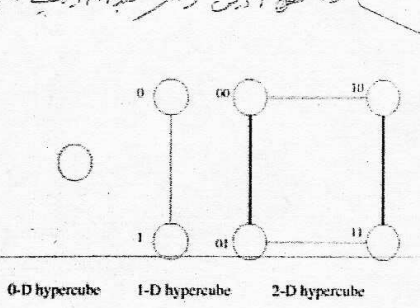


39

شبکه های ایستا
شبکه مکعب (Cube)

- شبکه هاي مکعب چند بعدي

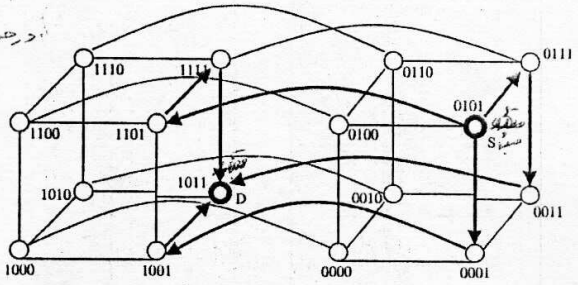




40

شبکه های ایستا
شبکه مکعب (Cube)

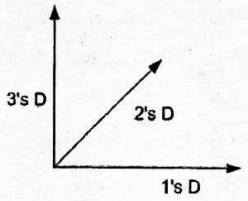
- یک گراف بدون جهت با 2^n رأس که از 0 تا 1-(2^n) برچسب خورده اند.
- آدرس هر گره با گره مجاورش فقط در یک بیت تفاوت دارند.



41

شبکه های ایستا
شبکه مکعب (Cube)

- در یک سیستم چند پردازنده ای مبتنی بر مکعب، عناصر پردازنده در رأس گراف قرار دارند. یالها اتصال بین پردازنده ها را نشان می دهد.
- هر پردازنده در یک n-Cube به n پردازنده دیگر متصل است.
- مسیریابی در n-Cube:
  - با XOR کردن آدرس مبدأ با مقصد، مقدار 1 در نتیجه XOR نشان دهنده اینست که پیام باید لینک قرار گرفته در بعد متناظر با موقعیت 1 را پیمایش کند.



n ... 4 3 2 1	D
X_{n-1} ... X_3 X_2 X_1 X_0	XOR

42

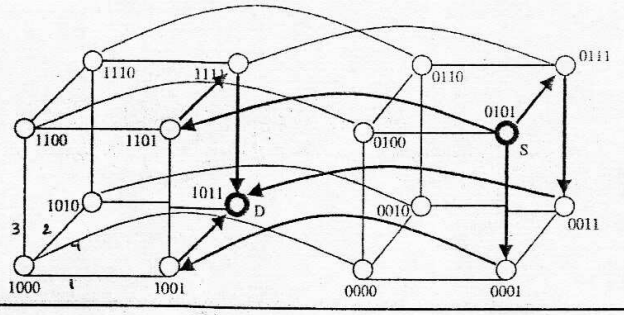
2/8/2014

شبکه های ایستا
شبکه مکعب (Cube)

- مثال: مسیریابی بین مبدأ (0101) و مقصد (1011) در یک 4-Cube:

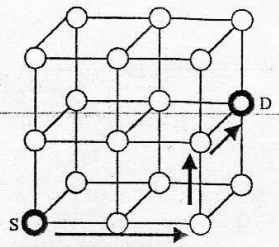
```
1011
0101
----
1110
```

- پس باید ابعاد 2، 3 و 4 در 4-Cube پیمایش شوند. ترتیب پیمایش مهم نیست. بنابراین ۳ مسیر وجود دارد:



شبکه های ایستا
شبکه مش (Mesh)

- جاییکه n: تعداد ابعاد و Ki: بعد اُام $nD\ Mesh = K_0 \times K_1 \times \ldots \times K_{n-1}$
- گره (i, j, k) به گره های (i±1, j±1, k±1) مجاورش متصل است.
- معماری Mesh با اتصالات wrap around نتیجه در شبکه Torus می شود.
- مکانیزمهای مسیریابی در Mesh:
  - Dimension-ordering routing: پیام در یک مرتبه یک بعد را پیمایش می کند تا به مختصات خاصی برسد قبل از اینکه بعد بعدی را پیمایش کند.
- مثال: مسیریابی از مبدأ (000) به مقصد (211):



۶

2/8/2014

فصل چهارم

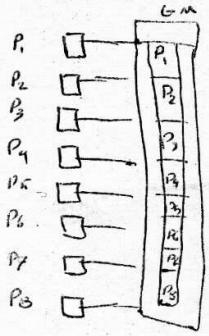
الگوریتم هایِ بردارها و ماتریس ها

1

شبیه سازی دسترسی چندگانه روی یک EREW PRAM

- الگوریتمی که برای EREW طراحی شود، روی سایر مدلها قابل اجرا است، ولی عکس آن درست نیست.
- فرض شود مکان حافظه X در یک زمان مورد نیاز تمامی پردازنده ها در PRAM باشد.
  - در مدل CRCW/CREW PRAM تمامی پردازنده ها می توانند در زمان ثابتی خواندن همروند انجام دهند.
  - در مدل EREW PRAM باید مکانیزم پخش-همگانی (Broadcasting) دنبال شود.

2



1

2/8/2014

پخش-همگانی روی مدل EREW PRAM

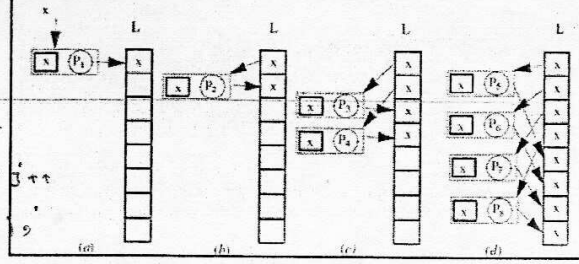
- P1 مقدار X را می خواند و آنرا به P2 می شناساند.
- P1 و P2، X را به ترتیب به P3 و P4 بصورت موازی نشان می دهند.
- P1، P2، P3 و P4 بصورت موازی X را بترتیب به P5، P6، P7 و P8 نشان می دهند.
- این هشت پردازنده X را برای هشت پردازنده دیگر مشخص می کنند و الی آخر...

3

مثالی از پخش-همگانی روی مدل EREW PRAM

- برای P=8
- آرایه L به طول ۸ بعنوان فضای کاری در حافظه مشترک برای توزیع مقدار X به تمامی پردازنده ها استفاده می شود.

```
Algorithm Broadcast_EREW
Processor P1
  y (in P1's private memory) ← x
  L[1] ← y
for i=0 to log p - 1 do
  forall Pj, where 2^i + 1 ≤ j ≤ 2^(i+1) do in parallel
    y (in Pj's private memory) ← L[j - 2^i]
    L[j] ← y
  endfor
endfor
```



4

2

2/8/2014

تحلیل الگوریتم های موازی

- پیچیدگی الگوریتم: تابعی است که زمان اجرا الگوریتم را بر حسب n (تعداد ورودی های الگوریتم) می دهد.
- معیارهای اندازه گیری پیچیدگی الگوریتم های موازی:
  - زمان اجرا، زمان سپری شده طی اجرای الگوریتم است (زمان اجرای اولین پردازنده تا پایان اجرای آخرین پردازنده).
  - تعداد پردازنده ها که الگوریتم برای حل مساله بکار می گیرد.
  - هزینه الگوریتم موازی که حاصلضرب زمان اجرا در تعداد پردازنده ها است.

5

قوانین طراحی الگوریتمهای موازی

- تعداد پردازنده ها باید با اندازه مساله محدود شود.
- زمان اجرا باید به اندازه قابل توجهی از کمترین زمان اجرای بهترین الگوریتم متوالی کمتر باشد.

6