Universidade Federal do Rio Grande do Sul
Instituto de Informática

Organização de Computadores

Aula 27

Máquinas paralelas

Máquinas paralelas

1. Introdução
2. Máquinas SIMD
3. Processadores vetoriais
4. Máquinas MIMD
5. Processadores dataflow

1. Introdução

- **classificação de máquinas paralelas – Flynn, 1966**
  - **SISD – single instruction, single data**
  - **SIMD – single instruction, multiple data**
  - **MISD – multiple instruction, single data**
  - **MIMD – multiple instruction, multiple data**
- **SISD**
  - **máquinas convencionais, apesar do paralelismo proporcionado por pipelines e superescalaridade**
- **SIMD**
  - **processadores vetoriais**
  - **processadores de array**
- **MIMD**
  - **multiprocessadores**

2. Máquinas SIMD

- **processador opera sobre vetores de dados**
- **controle único**
  - **1 contador de programa**
  - **1 bloco de controle**
  - **1 instrução sendo executada**
- **múltiplas unidades de execução (blocos operacionais) – cada um tem …**
  - **ALU**
  - **registradores de dados**
  - **registradores de endereço**
  - **memória de dados local**
  - **interconexões com unidades vizinhas**

Máquinas SIMD

- **máquinas SIMD reais têm uma mistura de instruções SISD e SIMD**
- **processador hospedeiro SISD**
  - **executa operações seqüenciais**
  - **calcula endereços**
  - **acessa memória de instruções**
- **máquinas SIMD são mais eficientes quando processam arrays em laços do tipo “for”**
  - **mais eficientes quando aplicação tem paralelismo de dados massivo**
- **máquinas SIMD são mais ineficientes em aplicações do tipo “case”**
  - **cada unidade de execução executa operação diferente, dependendo do dado**

Máquinas SIMD

- **exemplo: somar 128.000 números em máquina SIMD com 128 unidades de execução**
- **primeiro passo: dividir dados entre unidades de execução**
  - **processador hospedeiro armazena cada sub-conjunto de dados na memória local de cada unidade**
- **segundo passo: obter a soma de cada sub-conjunto**
  - **laço executado internamente a cada unidade**
  - **ler dado da memória local, somar, armazenar em variável local**

```
sum = 0 ;
for  i = 0 step 1 until 999
do  sum = sum + A1 [ i ] ;
```

Máquinas SIMD

- **terceiro passo**
  - **somar os resultados parciais**
  - **problema: cada resultado está numa unidade diferente**
- **estratégia**
  - **enviar resultados parciais para unidade vizinha e lá somar 2 parcelas**
  - **repetir processo recursivamente até restar um único resultado**

```
limit = 128 ; half = 128 ;
repeat
    half = half / 2 ;
    if      ( Pn >= half & Pn < limit )
    then send ( Pn / 2, sum );
    if      ( Pn < half )
    then sum = sum + receive ( ) ;
    limit = half ;
until  ( half = 1 ) ;
```

3. Processadores vetoriais

- **caso particular de máquinas SIMD**
- **“processador vetorial” com pipeline associado a um processador hospedeiro escalar convencional**
- **processador vetorial …**
  - **busca dados vetoriais de “registradores vetoriais” ou diretamente da memória**
  - **executa operações sobre os vetores através de 1 ou mais pipelines funcionais paralelos**
- **exemplo: Cray C-90**
  - **registrador vetorial tem 64 x 64 bits**
  - **2 pipelines funcionais vetoriais**
- **outras máquinas vetoriais**
  - **Convex C3, DEC VAX 9000, Fujitsu VP2000, Hitachi S-810, IBM 390/VF**

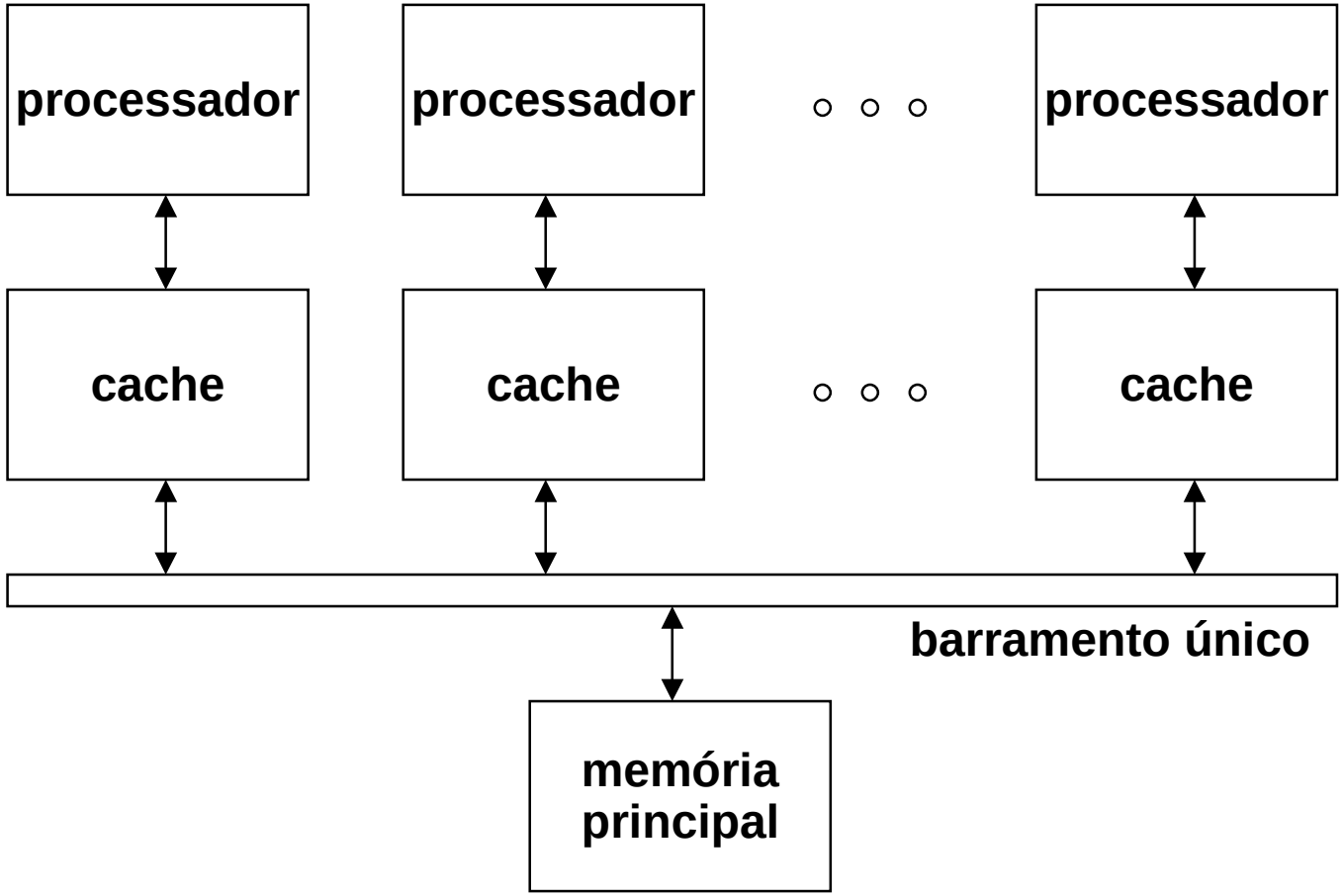
4. Máquinas MIMD

- **multiprocessadores**
  - **memória compartilhada**
  - **memória distribuída**
- **cada um destes modelos de multiprocessador tem um correspondente …**
  - **modelo de comunicação entre os processadores**
  - **modelo de sincronização entre os processadores**
- **máquinas escaláveis**
  - **número de processadores pode ser configurado**
- **eventual tolerância a falhas**
- **aplicação em servidores**

Máquinas MIMD: problemas

- **encontrar aplicações que possam ser paralelizadas para um multiprocessador com ganho significativo de desempenho**
- **custo da comunicação entre os processadores diminui desempenho**
- **complexidade da programação paralela**

Máquinas MIMD com memória compartilhada



Máquinas MIMD com memória compartilhada

- **exemplo da soma de 128.000 números**
  - **supondo multiprocessador com 16 processadores**
- **todos os processadores têm acesso à mesma memória global**
  - **não é necessário dividir os dados entre memórias locais**
- **cada processador acessa uma parte dos dados a partir de um endereço diferente da memória global**
- **seja Pn o número do processador**
- **primeiro passo**
  - **cada processador soma os dados de seu subconjunto**

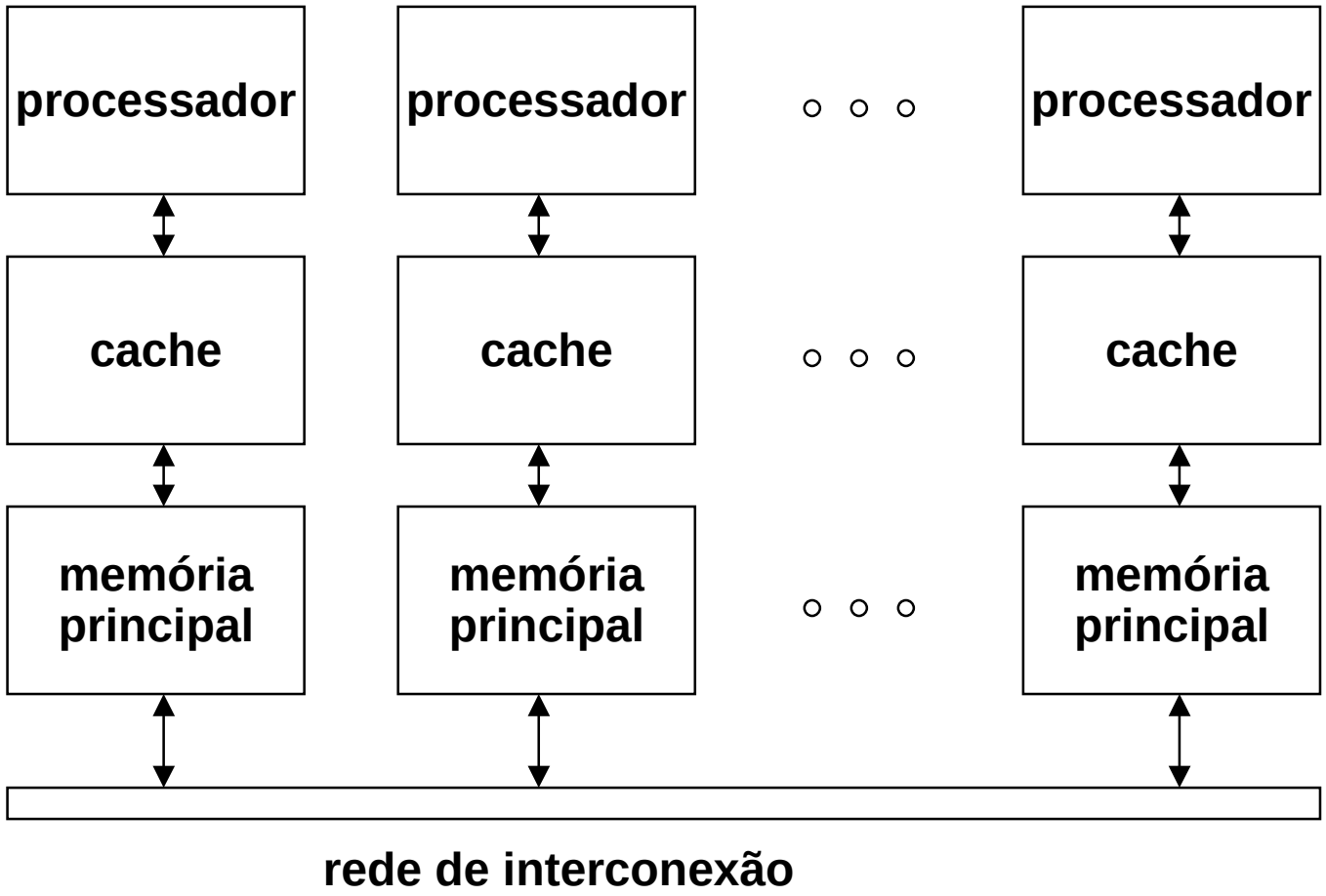
```
sum [ Pn ] = 0 ;
for  i = 8000 * Pn  step 1 until  8000 * ( Pn + 1 )
do  sum [ Pn ] = sum [ Pn ] + A [ i ] ;
```

Máquinas MIMD com memória compartilhada

- **segundo passo**
  - **somar resultados parciais**
- **diferenças em relação à solução SIMD**
  - **somas parciais não precisam ser enviadas e recebidas – cada processador simplesmente acessa o valor na memória global**
  - **processadores precisam se sincronizar explicitamente**
    - **processador só pode utilizar soma parcial de outro quando este tiver terminado de calcular a mesma**

```
half = 16 ;
repeat
   synch ( ) ;
   half = half / 2 ;
   if       Pn < half
   then  sum [ Pn ] = sum [ Pn ] + sum [ 2 * Pn ] ;
until  half = 1 ;
```

Máquinas MIMD com memória distribuída



Máquinas MIMD com memória distribuída

- **comunicação através da rede ocorre apenas para efeito de sincronização entre os processadores**
  - **processamento paralelo “fracamente acoplado”**
  - **pouca transferência de dados entre processadores**
- **barramento único: é utilizado em cada acesso à memória**
  - **gargalo na comunicação**
  - **processamento paralelo “fortemente acoplado”**
- **“memória compartilhada” x “memória distribuída” é uma falsa dicotomia**
  - **o oposto de “memória distribuída” é “memória centralizada” (localização física da memória)**
  - **o oposto de “espaço de memória compartilhado” é “espaços de memória múltiplos privativos”**
  - **estas duas questões são ortogonais**

Máquinas MIMD com memória distribuída

- **exemplo da soma de 128.000 números**
  - **supondo multiprocessador com 128 processadores**
- **cada processador tem acesso a parte dos dados, em sua memória local**
  - **código é então similar ao da solução SIMD**
- **primeiro passo: dividir dados entre processadores**
  - **processador que tem os dados faz a distribuição**
- **segundo passo: obter a soma de cada sub-conjunto**
  - **laço executado internamente a cada processador**
  - **idêntico à solução SIMD**

```
sum = 0 ;
for  i = 0 step 1 until 999
do  sum = sum + A1 [ i ] ;
```

Máquinas MIMD com memória distribuída

- **terceiro passo: somar os resultados parciais**
- **estratégia e código da solução SIMD funcionam**
  - **“send” e “receive” devem ser considerados como funções não apenas de comunicação, como no SIMD, mas também de sincronização**
  - **processador que executa “receive” irá trancar enquanto dado não vier**

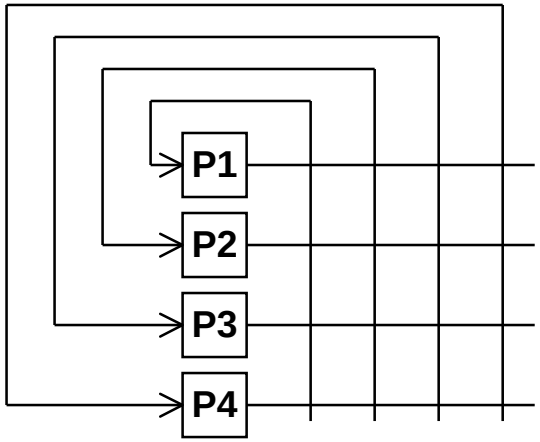
```
limit = 128 ; half = 128 ;
repeat
    half = half / 2 ;
    if      ( Pn >= half & Pn < limit )
    then send ( Pn / 2, sum );
    if      ( Pn < half )
    then sum = sum +  receive ( ) ;
    limit = half ;
until  ( half = 1 ) ;
```

Máquinas MIMD com memória distribuída

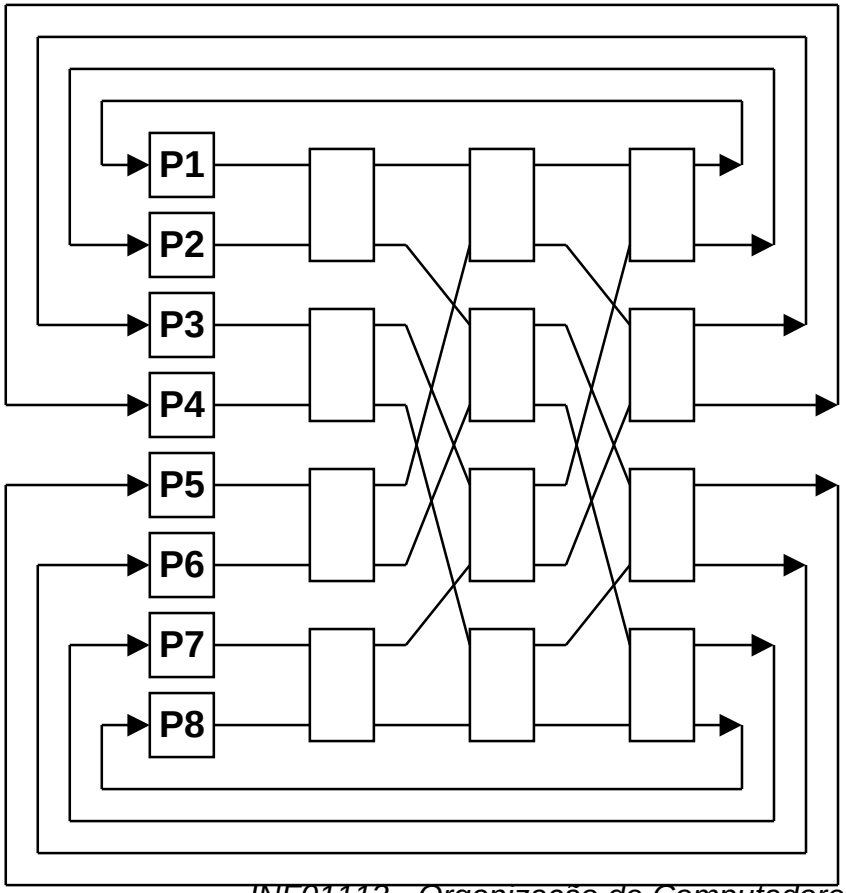
- **máquinas comerciais**
  - **Intel iPSC/2, 128 processadores, 1988**
  - **nCube, 1024 processadores, 1987**
  - **Intel Delta, 540 processadores, 1991**
  - **Thinking Machines CM-5, 1024 processadores, 1991**

Topologias da rede de interconexão

- **casos extremos de compromisso custo x desempenho**
  - **rede completamente conectada**
  - **barramento simples**
- **redes multi-estágio**

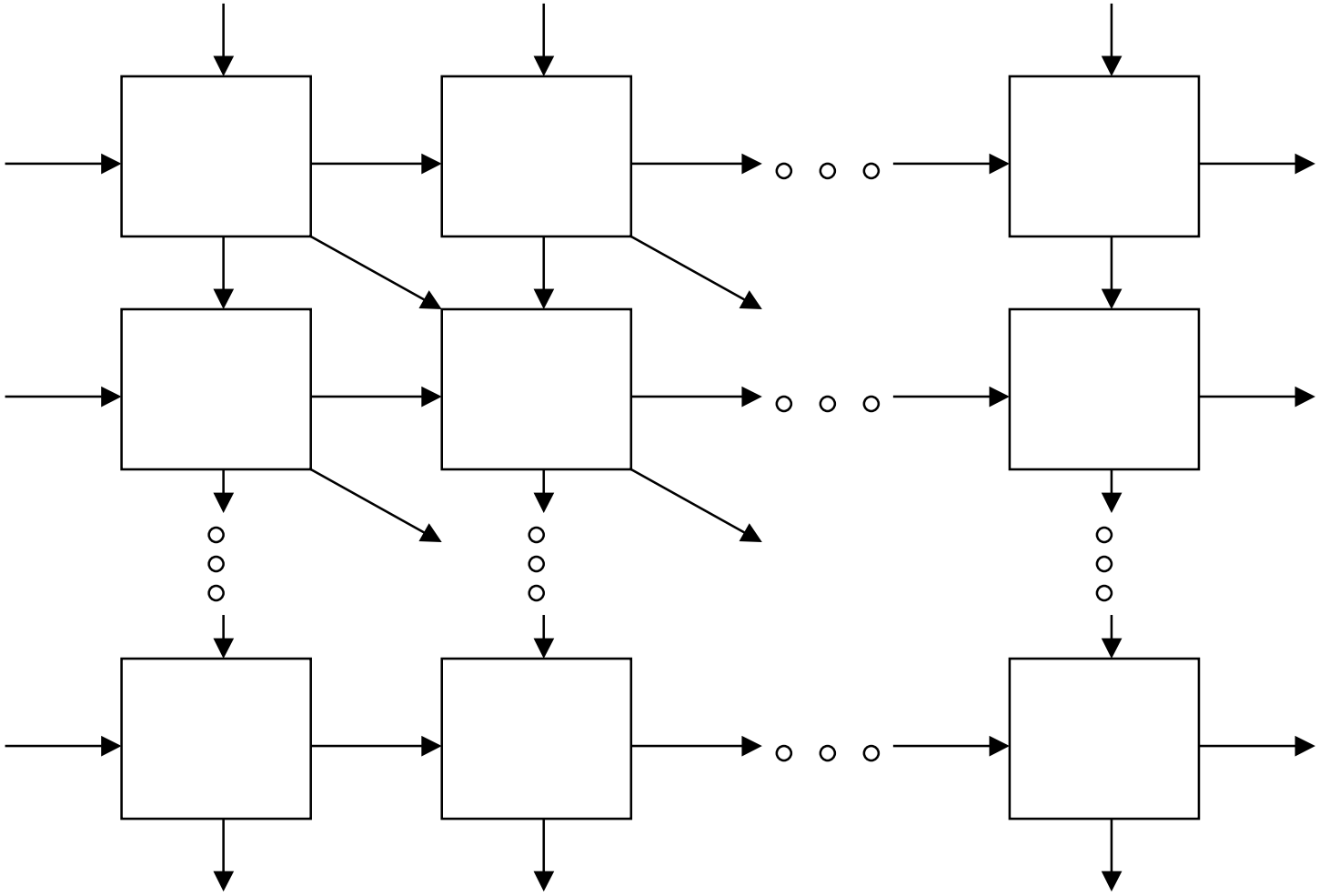


chave crossbar



rede ômega

5. Processadores dataflow



Processadores sistólicos

- **processamento “data-flow”**
  - **cada processador executa operação quando dados de entrada estão disponíveis**
- **processadores elementares**
  - **executam operação única, não programáveis**
- **aplicações em processamento digital de sinais**
  - **processamento de imagens, voz, ...**
- **integração num único chip**
- **utilizados …**
  - **em sistemas eletrônicos dedicados**
  - **como unidade funcional especializada de um processador hospedeiro convencional**