Universidade Federal do Rio de Janeiro
Pós-Graduação em Informática
DCC/IM - NCE/UFRJ

Arquiteturas de Sistemas de Processamento Paralelo

Arquiteturas SIMD

Gabriel P. Silva

Arquiteturas SIMD

- Processadores Vetoriais
- Arquiteturas SIMD
- Arquiteturas Sistólicas

Processadores Vetoriais

- **Os processadores vetoriais são arquiteturas pipelined do tipo SIMD, ou seja, uma única instrução opera sobre vários dados, no caso, um vetor.**
- **Um processador vetorial possui instruções para operar em vetores: *um conjunto linear de valores.***
- **Uma operação vetorial típica pode adicionar dois vetores com 64 elementos em notação ponto flutuante para obter um único vetor de resultado com 64 elementos.**

Processadores Vetoriais

- A instrução vetorial é equivalente a um "loop" inteiro, onde cada iteração computa um dos 64 elementos do resultado, atualiza os índices e retorna para o início.
- As instruções vetoriais possuem as seguintes características:
  - Cada instrução equivale a um loop
  - O cálculo de cada resultado não depende de resultados anteriores → é possível haver pipelines profundos sem a ocorrência de dependências de dados
  - O padrão de acesso à memória para a busca dos operandos é conhecido e regular → benéfico utilizar memória com interleaving

Processadores Vetoriais

- Podem ser de dois tipos:
  - Memória-Memória
    - Arquiteturas mais antigas
    - Todas as operações vetoriais manipulam operandos na memória
  - Registrador-Registrador
    - Arquitetura usadas em todos os processadores vetoriais mais recentes
    - Todas as operações vetoriais, com exceção de load e store, trabalham com registradores
- Unidades funcionais são organizadas como pipelines profundos

Processadores Vetoriais

- **Exemplo:**

  Vetor C ← Vetor A * Vetor B

- **Estágios do Pipeline:**

  Lê elementos dos vetores A e B → LA, LB
  Soma expoentes → SE
  Multipica mantissas → MM
  Normaliza resultado → NR
  Armazena resultado em C → AC

Exemplo de Processamento Vetorial

LA/LB	1	2	3	4	5	6	7	8	9	10
SE		1	2	3	4	5	6	7	8	9
MM			1	2	3	4	5	6	7	8
NR				1	2	3	4	5	6	7
AC					1	2	3	4	5	6

Exemplo de Processamento Vetorial

Tempo de execução → T = [5 + (N-1)] * t

N → Número de elementos do vetor

t → Perído do Clock

- **Supondo que os vetores A e B possuam cada um 10.000 elementos e que a freqüência do clock dos pipelines é de 200 MHz, temos, que o tempo total gasto para executar a operação de cálculo de todos os elementos do Vetor C é dado por:**
- ***T = (5 + (10.000 - 1)) x 5 ns = 50 μs (aprox.)***
- **200 MFLOPS**

Problemas para a vetorização

- **Tamanho dos vetores:**
  - **Quando o tamanho dos vetores é maior do que o dos registradores vetoriais, o compilador deve quebrar a operação vetorial em uma seqüência de operações com vetores de tamanho igual ou menor do que o dos registradores vetoriais ➔ "overhead" introduzido pelo esvaziamento dos pipelines e operações adicionais de load/store.**

Problemas para a vetorização

- **Stride**
  - Em operações de multiplicação de matrizes, por exemplo, necessariamente o acesso ao vetor que armazena os elementos de uma das matrizes não será feito de forma seqüencial, já que as matrizes são armazenadas na memória de forma ordenada, ou por linha ou por coluna.
  - O salto entre os endereços de dois elementos consecutivos a serem utilizados é chamado ***stride.***
  - Em geral estão disponíveis instruções vetoriais do tipo *load/store* com a definição do valor do *stride*.

Problemas para a vetorização

- A existência de desvios condicionais em "loops":

```
for (i=1; i<256; i++)
    if (a[i] < 0)
        a[i] = a[i] + b[i];
```

Problemas para a vetorização

- Para que operações como a de soma do exemplo anterior possam ser vetorizadas, as arquiteturas de processamento vetorial devem suportar os chamados ***vetores de máscara*** associados às operações vetoriais, indicando sobre que elementos dos vetores devem ser efetuadas as operações, em função do resultado do teste relativo ao desvio condicional.

Supercomputadores Vetoriais Comerciais

- **NEC-SX5**
  - Até 16 processadores
  - Período de Clock: 4 ns
  - Desempenho de Pico: 128 GFLOPS (123)
- **Cray T90 ***
  - Até 32 processadores
  - Período de Clock: 2.2 ns
  - Desempenho de Pico: 56 GFLOPS (36.6)
- **Fujitsu VPP-5500U**
  - Um processador
  - Período de clock: 3,3 ns
  - Desempenho de Pico: 9,6 GFLOPS (8,7)

Supercomputadores Vetoriais Comerciais

- Cray T90

Perspectivas para os Processadores Vetoriais

- Em 2000 o setor dos computadores paralelos correspondeu a aproximadamente dois terços do mercado de computação técnica de alto desempenho.
- Os supercomputadores vetoriais ainda são os equipamentos preferidos em determinadas atividades, inclusive nas aplicações automobilísticas e aeroespaciais.
- Sua participação no mercado continuará a cair ao mesmo tempo que os supercomputadores paralelos deverão manter a tendência de se tornarem cada vez mais poderosos.

Arquiteturas SIMD

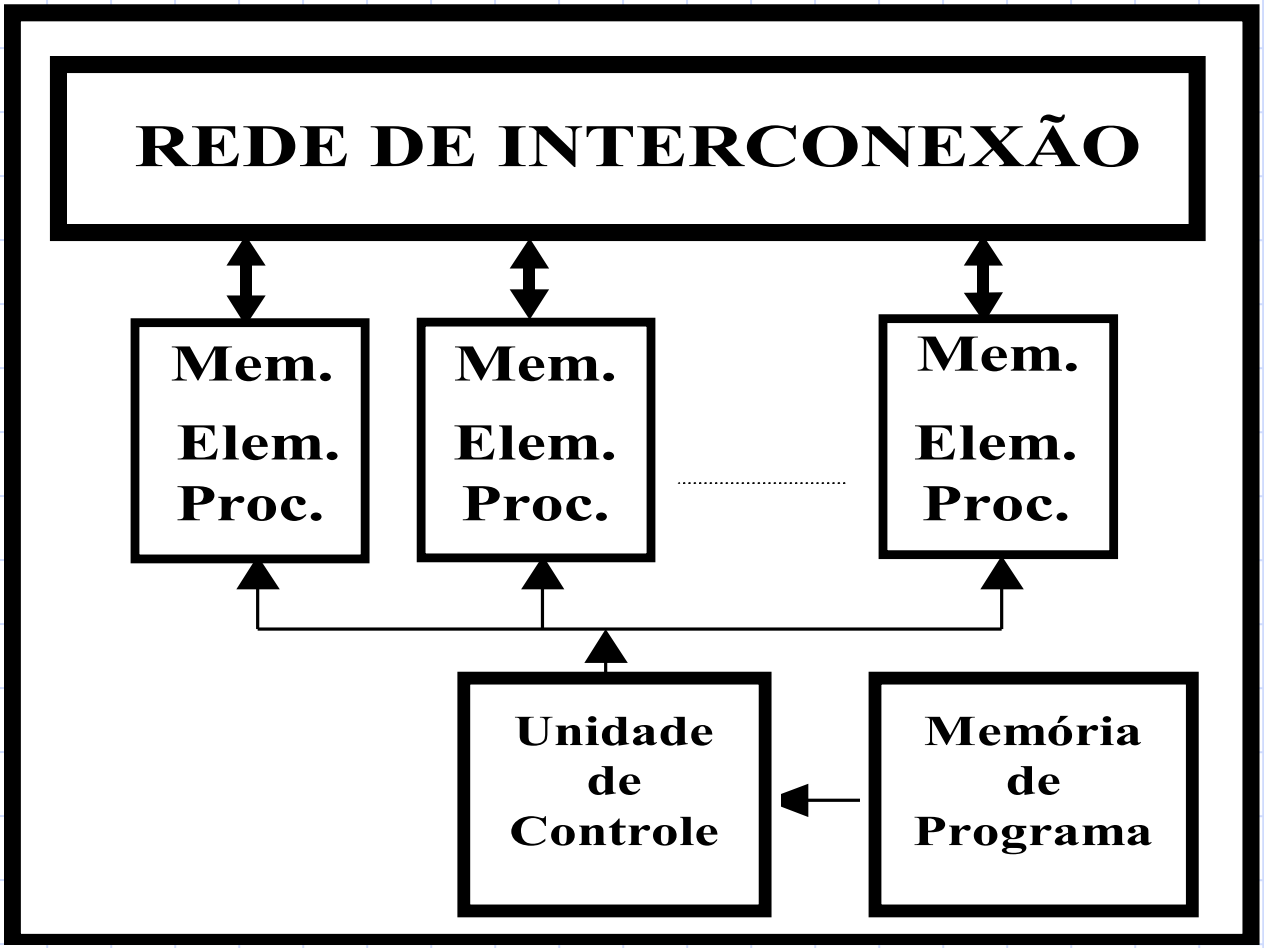
Arquiteturas SIMD

- É uma classe importante de processadores, devido a fatores como:
  - Simplicidade de conceitos e programação
  - Regularidade da estrutura
  - Facilidade de escalabilidade em tamanho e desempenho
  - Aplicação direta em uma série de aplicações que requerem paralelismo para obter o desempenho necessário.

Arquiteturas SIMD

- Processadores executam sincronizadamente a mesma instrução sobre dados diferentes.
- Utiliza vários processadores especiais muito mais simples, organizados em geral de forma matricial.
- Muito eficiente em aplicações onde cada processador pode ser associado a uma sub-matriz independente de dados (processamento de imagens, algoritmos matemáticos, etc.).

Arquitetura Básica



Granulosidade

- É a relação entre o número de elementos processadores e o paralelismo do conjunto de dados sobre os quais opera.
- Um sistema com poucos elementos de dados por elemento processador é qualificado como granulosidade fina.
- Não é prático termos apenas um elemento processador por elemento de dados nas arquiteturas SIMD.

Conectividade

- Vizinhança (Malha ou Toro)
  - Diâmetro Grande
  - Largura de Banda Ampla
- Árvore
  - Diâmetro Pequeno
  - Largura de Banda Estreita
- Pirâmide
  - Diâmetro Pequeno
  - Programação Complexa
- Hipercubo
  - Diâmetro Pequeno
  - Largura de Banda Ampla

Conectividade

Rede	Diâmetro	Largura de Banda
Barramento	1	1
Crossbar	1	N
Vizinhança	$2 (N)^{1/2}$	N
Árvore	2 log N	2 log N
Pirâmide	$2 \log(N)^{1/2}$	$2 \log(N)^{1/2}$
Hipercubo	log N	N
Cubo	1	N

Complexidade do Processador

Single-Bit	**Granulosidade fina. Usualmente utilizados para processamento de imagem**
Inteiro	**Uma solução de compromisso. Utilizados para visão ou computação genérica**
Ponto Flutuante	**Granulosidade grossa. Usualmente utilizados para computação científica.**

Autonomia Local

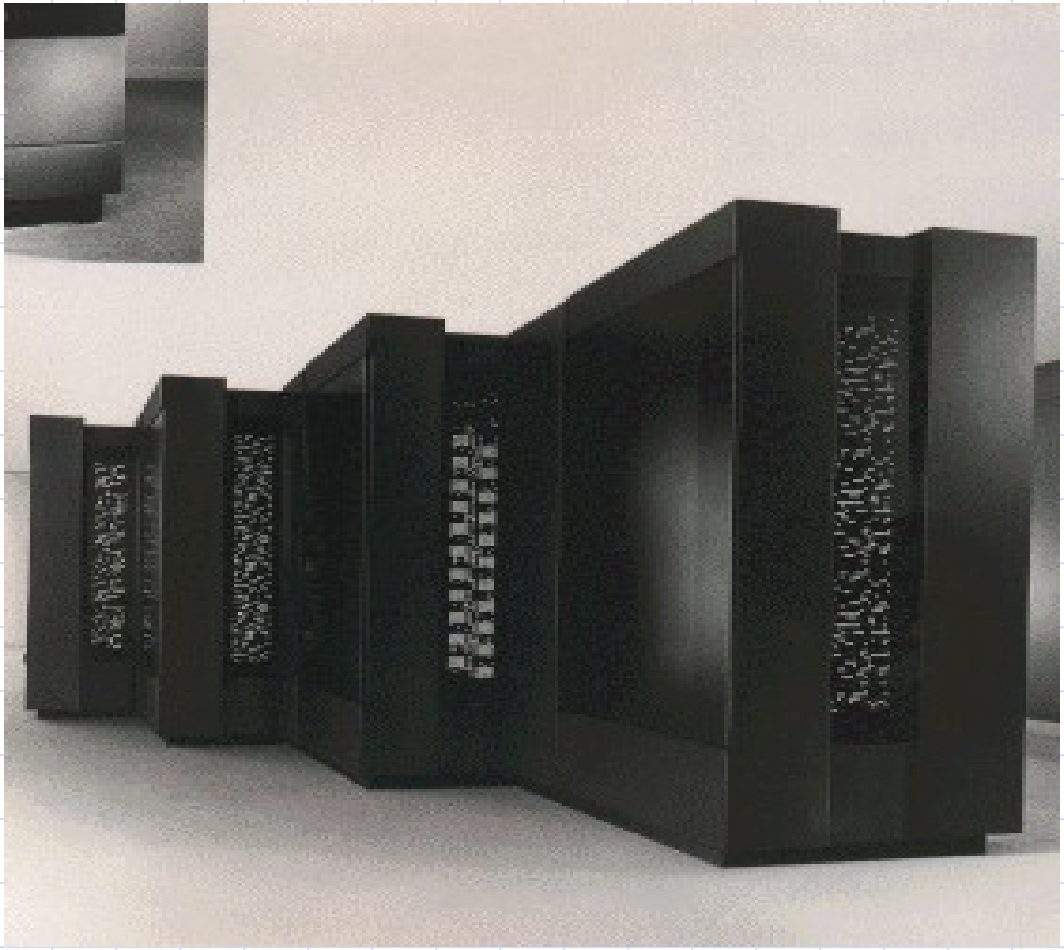
Nenhuma	**Todos os elementos processadores (EP) executam a mesma operação**
Atividade	**Alguns EPs podem estar inativos**
Dados	**Alguns EPs escolhem diferentes fontes de dados**
Conectividade	**Os EPs escolhem sua própria conectividade**
Função	**Os EPs escolhem os seus cálculos**
Algoritmo	**Os EPs escolhem os seus próprios algoritmos**

Exemplos

- Illiac IV
  - Matriz 8 x8 (1968)
- MPP *
  - Matriz 128 x128 (1983)
- Connection Machine *
  - Hipercubo (16K x 4) processadores (1985)
- MasPar
  - Multi-level crossbar switch (1990)
  - 16K processadores

Thinking Machines

- CM-5



Thinking Machines

- CM5 fat tree network

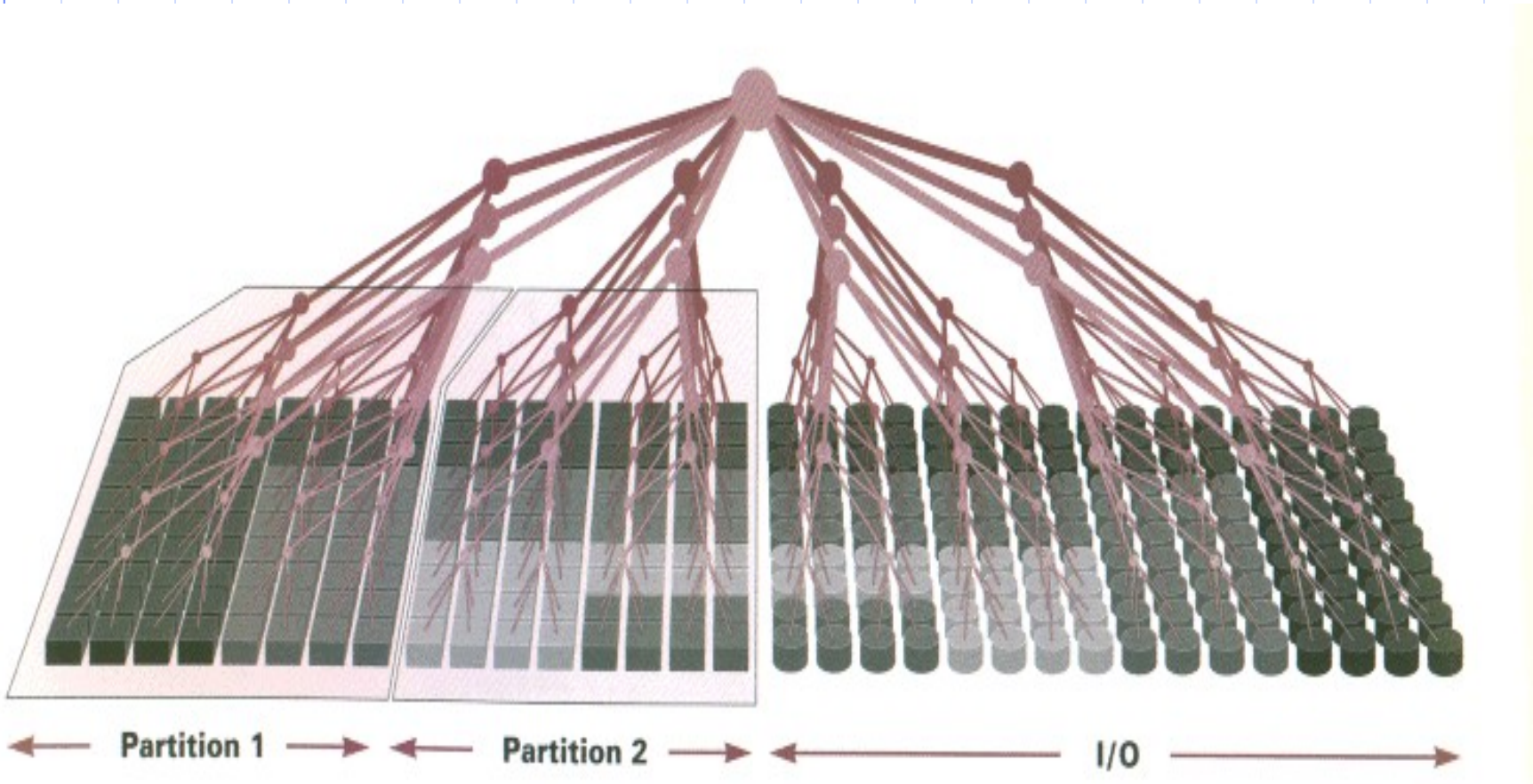


Figure 2

Arquiteturas Sistólicas

Arquiteturas Sistólicas

- Termo utilizado por H.T. Kung, em 1978, em analogia com o funcionamento do coração.
- Uma aplicação típica é a multiplicação de matrizes:

$$C_{mn} = \sum A_{im} \, X \, B_{nj}$$

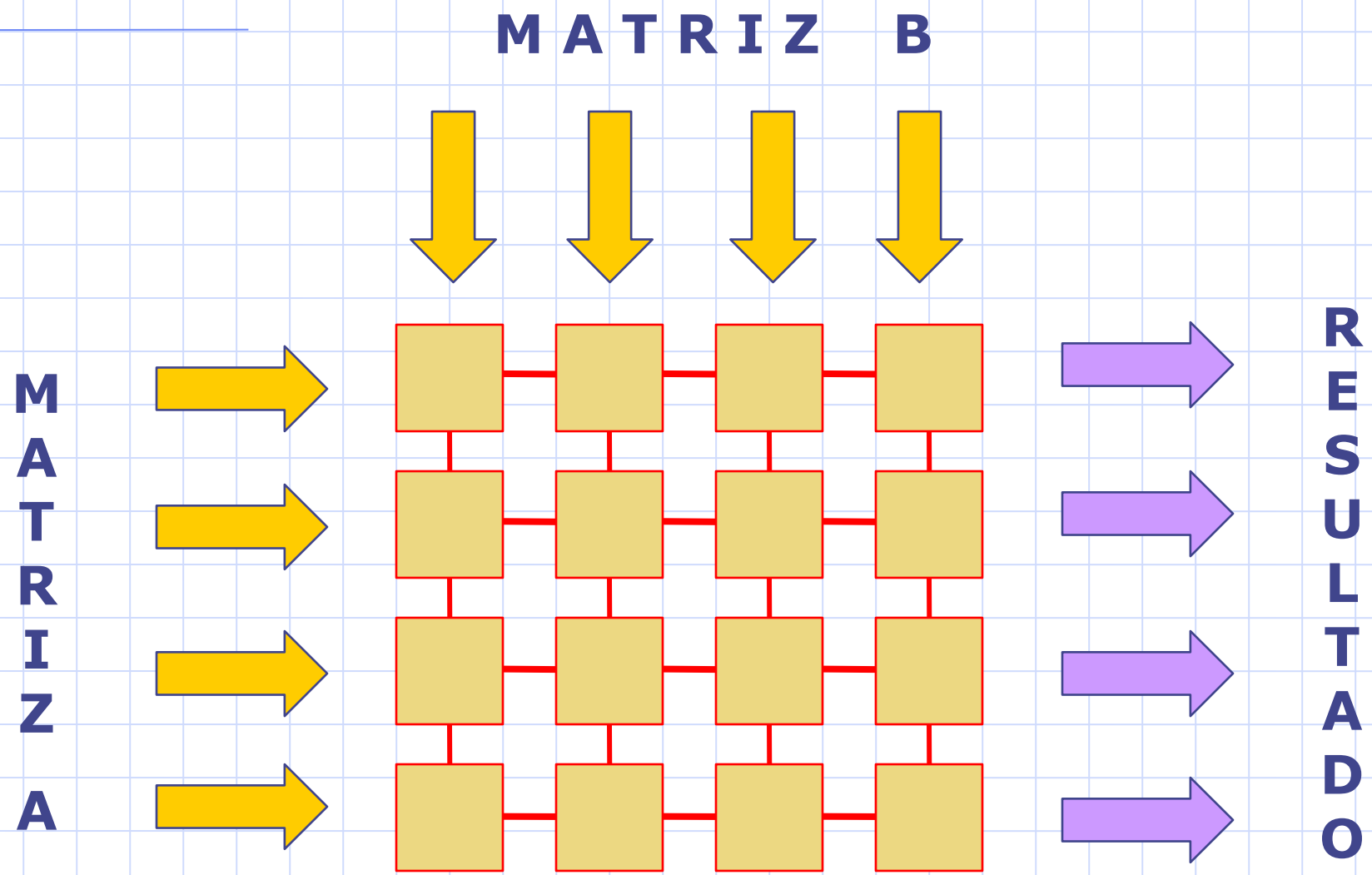
Características Básicas

- **Possui as seguintes características:**
  - **Cada elemento do arranjo computa um função simples e única**
  - **Cada elemento está conectado apenas aos seus vizinhos mais próximos através apenas de um caminho de dados uni-direcional**
  - **O único sinal de controle enviado através do arranjo é um pulso de relógio, utilizado para sincronizar as operações**
  - **Os dados de entrada são inseridos nos dois lados da matriz e passam através do arranjo em duas direções ortogonais**
  - **Os dados de saída são extraídos como uma série de elementos de um dos dois lados restantes da matriz**

Características Básicas

- **Dimensão:**
  - 1, 2 ou mais dimensões
- **Precisão:**
  - 1 bit ou multi-bit
  - Inteira ou ponto-flutuante
- **Conectividade**
  - Linear, matriz, hexagonal, etc.
  - Uni-direcional ou bi-direcional
- **Sincronicidade**
  - Fixa ou programável

Organização da Matriz



Elemento Processador