



ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES II

AULA 02: PROCESSAMENTO PARALELO: PROCESSADORES VETORIAIS

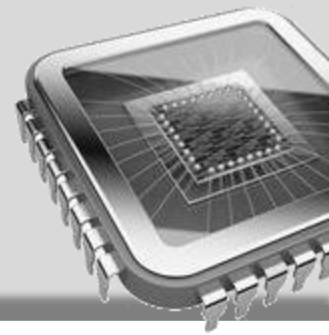
Prof. Max Santana Rolemberg Farias

max.santana@univasf.edu.br

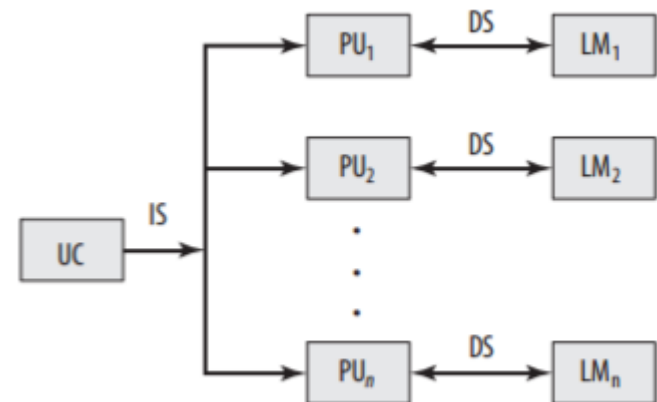
Colegiado de Engenharia de Computação



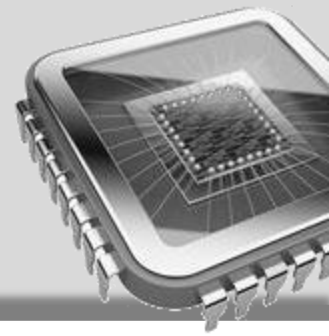
PROCESSADORES VETORIAIS



- Os processadores vetoriais são arquiteturas *pipelined* do tipo SIMD (*single instruction, multiple data*).
 - Uma única instrução de máquina controla a execução simultânea de uma série de elementos de processamento em operações básicas.

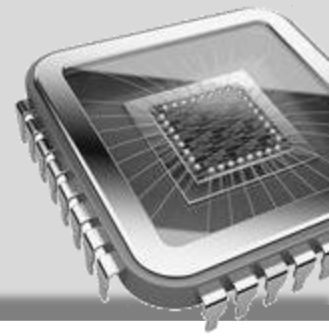


PROCESSADORES VETORIAIS



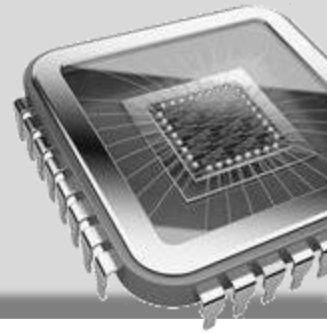
- Um processador vetorial possui instruções para operar em vetores.
 - Uma operação vetorial típica pode ser a multiplicação, subtração ou soma de dois vetores para obter um único vetor.
 - Em processadores escalares, essas operações são realizadas através de um *loop*.
 - Em processadores vetoriais, essas operações são realizadas em uma **única instrução vetorial**.

PROCESSADORES VETORIAIS: INSTRUÇÕES VETORIAIS



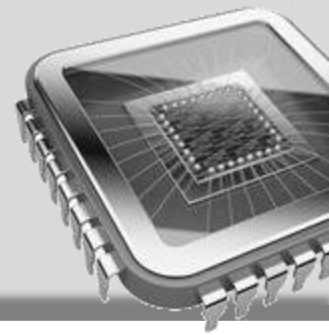
- As instruções vetoriais possuem as seguintes características:
 - Cada instrução equivale a um *loop*
 - O cálculo de cada resultado não depende de resultados anteriores
 - É possível haver pipelines profundos sem a ocorrência de dependências de dados.
 - O padrão de acesso à memória para a busca dos operandos é conhecido e regular.

PROCESSADORES VETORIAIS: USO



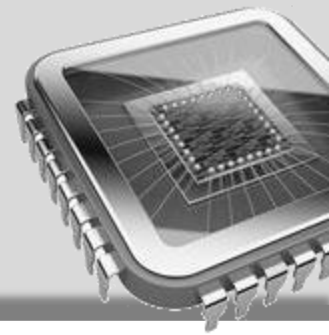
- Os processadores vetoriais são particularmente úteis para aplicações científicas e de engenharia
 - Simulações
 - Previsão do tempo
 - Aplicações que operam sobre grandes matrizes/vetores

PROCESSADORES VETORIAIS: VANTAGENS



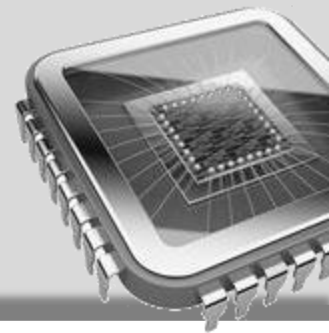
- Redução da quantidade de *fetch* e *decode* de instruções.
- Não há necessidade de verificação de conflitos de dados
 - As operações entre elementos dos vetores que estão na mesma operação são independentes.
- As operações entre elementos podem ser paralelizadas ou executadas em *pipeline*.
- Como há uma quantidade menor de instruções por programa, há uma quantidade menor de falha na cache de instruções.

PROCESSADORES VETORIAIS: TIPOS

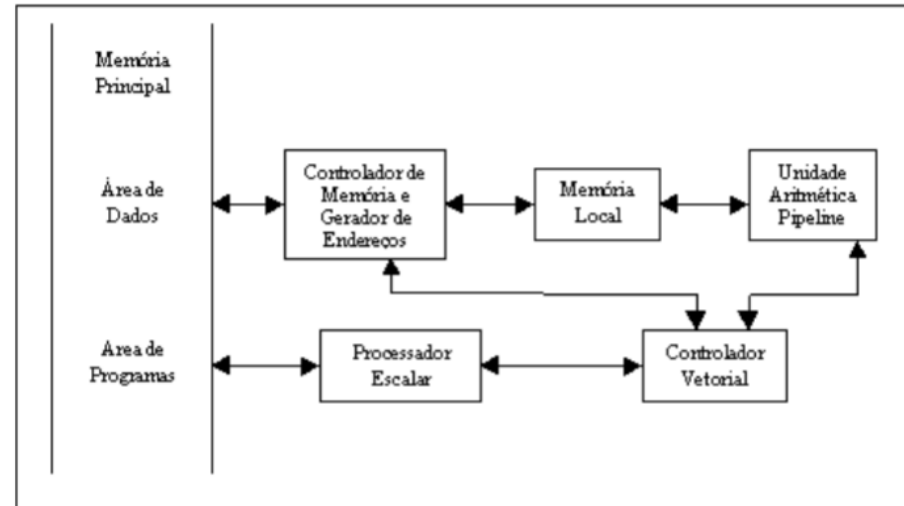


- Os processadores vetoriais podem ser de dois tipos:
 - Memória-Memória
 - Arquitetura mais antiga
 - Todas as operações vetoriais manipulam operandos na memória
 - Registrador-Registrador
 - Arquitetura usadas em todos os processadores vetoriais mais recentes.
 - Todas as operações vetoriais, com exceção de *load* e *store*, trabalham com registradores.

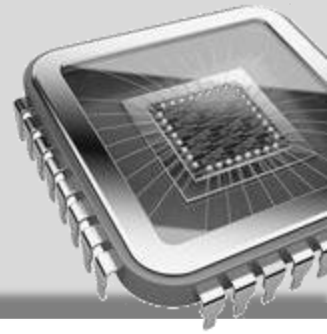
PROCESSADORES VETORIAIS: ARQUITETURA



- Processador escalar
 - Realiza operações sequenciais
 - Gerenciamento de E/S e memória
- Controlador de memória e Gerador de endereços
 - Recebem os endereços, onde os operandos são buscados
- Unidade Aritmética Pipeline
 - Executa as instruções vetoriais

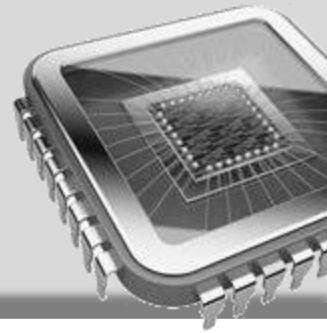


PROCESSADORES VETORIAIS: ENDEREÇAMENTO DE VETORES



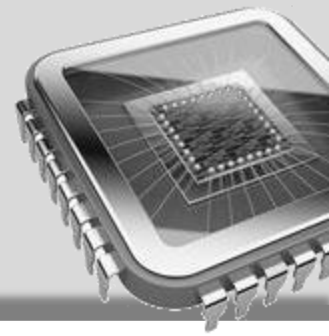
- O endereçamento de vetores pode ser de três maneiras:
 - Acesso contínuo
 - Os elementos estão contínuo na memória
 - Acesso não contínuo, mas regular
 - Os elementos estão separados por intervalos de tamanho k
 - Acesso indexado
 - Os elementos estão em posições arbitrárias na memória e o endereço de cada posição é armazenada na posição correspondente de um vetor auxiliar.

PROCESSADORES VETORIAIS: PIPELINE VETORIAL

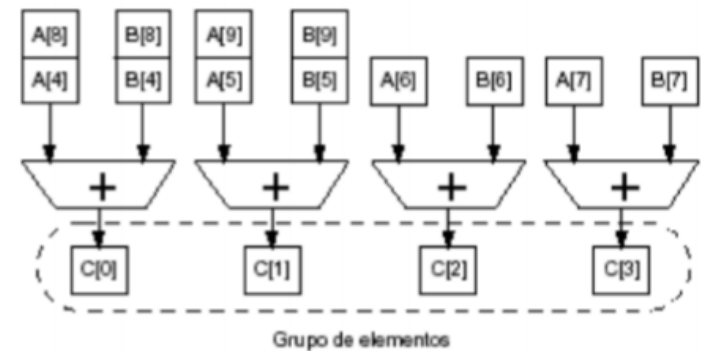
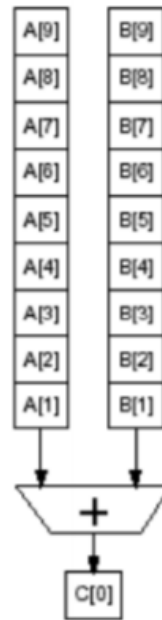


- O tempo de execução de uma instrução vetorial não é o mesmo tempo de uma instrução escalar.
 - ADDV não leva a mesma quantidade de ciclos que uma instrução ADD
 - Se isso acontecesse, seriam necessárias n ALU, uma para cada elemento do vetor, o **que é impraticável**.

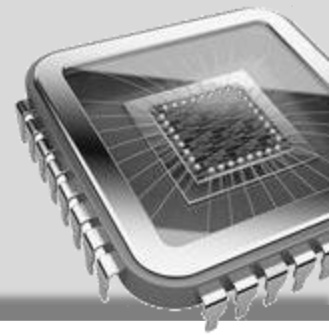
PROCESSADORES VETORIAIS: DESEMPENHO



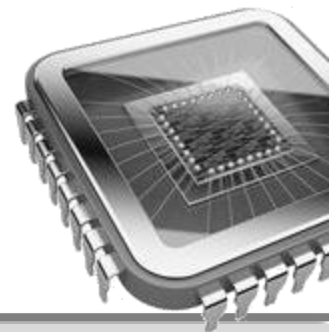
- Para melhorar o desempenho, pode-se aumentar o número de ALUs.



PROCESSADORES VETORIAIS: PROBLEMAS

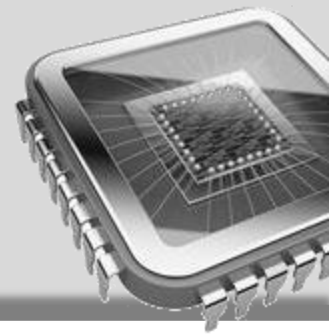


- Tamanho dos vetores
 - Quando o tamanho do vetor é maior do que o dos registradores vetoriais, o compilador deve quebrar em várias operações vetoriais.
- *Stride*
 - Quantidade de dados que devem ser “saltadas” na leitura/escrita
 - Utilizado principalmente quando se trabalha com matrizes, pois estas não são armazenadas sequencialmente na memória.
 - Normalmente as instruções de *load* e *store* possuem um valor para se especificar o *stride* .
- Existência de desvios condicionais nos *loops*.



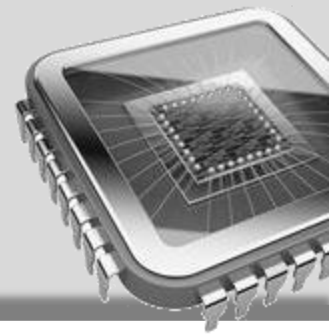
**QUAIS MICROPROCESSADORES POSSUEM
INSTRUÇÕES VETORIAS?**

INSTRUÇÕES VETORIAIS EM MICROPROCESSADORES

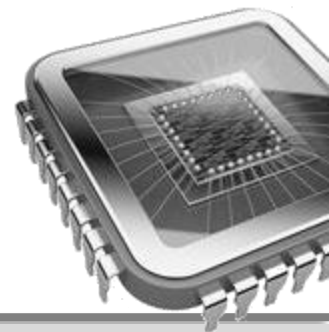


- MMX (Pentium II)
 - Registradores vetoriais de 64 bits
- 3DNow (AMD)
- SSE (Streaming SIMD Extension) (Pentium III)
 - 8 Registradores vetoriais de 128 bits e novo conjunto de instruções (+70)
- SSE2 (Pentium IV)
 - Registradores vetoriais de 512 bits com elementos de 64 bits mais novas instruções (+144)
- SSE3 (Pentium IV)
- SS4 (Intel Core)

INSTRUÇÕES VETORIAIS EM MICROPROCESSADORES: PROBLEMAS

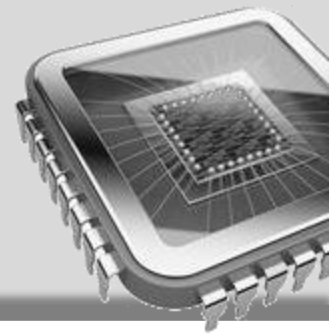


- Vetores curtos e poucos registradores
- Modos de endereçamento de memória simples (espaçamento único)
- Falta de suporte em linguagem de programação
- Conseqüentemente, instruções SIMD geralmente são encontradas apenas em bibliotecas de rotinas gráficas de baixo nível



QUAIS SÃO OS SUPERCOMPUTADORES VETORIAIS COMERCIAIS?

SUPERCOMPUTADORES VETORIAIS

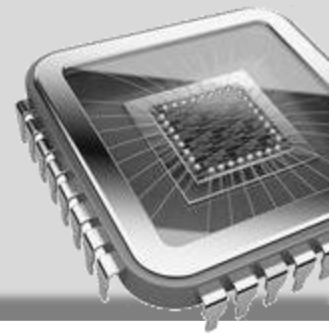


- **NEC-SX5**
 - Até 16 processadores
 - Período de clock de 4 ns
 - Desempenho de 128 GFLOPS

- **Cray T90**
 - Até 32 processadores
 - Período de clock de 2,2 ns
 - Desempenho de 56 GFLOPS

- **Fujitsu VPP-5500U**
 - 1 processador
 - Período de clock de 3,3 ns
 - Desempenho de 9,6 GFLOPS

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.

