Computação de Alto Desempenho

Módulo 1

Aula – 01

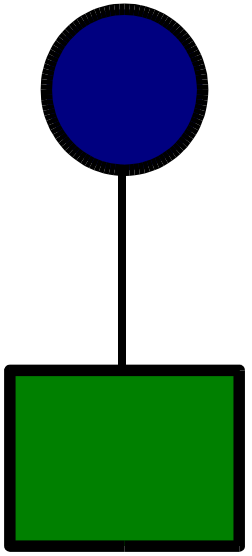
Março 2002

Conteúdo

- Arquitetura de Computadores – Conceitos Básiocs
  - Processadores
  - Hierarquias de Memórias
- Avaliação de Desempenho
- Otimizações
- BLAS
- ATLAS

Arquitetura de Computadores

- Máquina de von Neunnam
- Composta por :
  - CPU
    - Registradores
    - Unidade aritimetica e lógica (ALU)
  - Memória
  - Barramento (Bus)



- Tudo em um computador acontece em um "tick" do processador. A frequência do proc. Determina a velocidade do processador

- Desenho Básico – Single Instruction Stream
- Instruções são processadas sequencialmente
- Ex.: a = c+b
  - Recebe e decodifica a inst.
  - Calcula os endereços
  - Recebe os valores
  - Operação é executada
  - Resultado é escrito na memória
- Obs. Existe um movimento de dados e instruções entre CPU e memória

Paralelismo em um Processador

- Múltiplas ALU – podendo operar em paralelo
  - Soma : Multiplicação
  - O compilador precisa reordenar as operações
  - Entretanto aumenta o movimeto de dados entre a CPU e Mem.

- Pipelining
  - Cada uma responsável por decodificar e executar uma instr.
  - Linha de montagem, cada segmento é responsável por uma pequena tarefa
  - A cada ciclo se produz um resultado quando o Pipeline está cheio....

Tipos de Pipelines

- Pipelines de instruções
- Referência de memória
- Operações de ponto flutuante
  - Segmentação da ALU

Pipeline de Instrução

1 – Recebe e decodifica a inst.

2 – Calcula os endereços

3 – Recebe os valores

4 – Operação é executada

5 – Resultado é escrito na memória

- É necessário manter o pipeline cheio e com um fluxo constante.
- Qualquer operação mais complicada (raiz quadrada) ou um acesso muito lento a memória pode causar uma retenção.
- **If** pode causar problemas.
  - As operações são seriais
  - O conteúdo pode ser descartado

```
If (A .lt. B) then
    C=B -  A
else
    C=A -  B
end if
```

Tipos de Processadores

- CISC
  - Complex Instruction Set Computer
  - Tamanho da instrução não é constante
  - Ex.: Pentium Pro (hibrido)
- RISC
  - Reduced Instruction Set Computer
  - Tamanho da instrução é constante
  - O número de instruções é menor
  - Ex.: RS-6000Power2, UltraSparc,Alpha

CISC

```
R1 <- [addr(A) = 4 * R2] + 1
return
```

RISC

```
R3 <-  4 * R2
R3 <-  addr(A) + R3
R5 <-  [R3]
R1 <- R5 + 1
return
```

Classificação dos Proc.

- Super Escalar
  - As instruções (independentes) são feitas em paralelo, em pipelines separados
  - Isso é feito
  - Desempenho :
    - Processador
    - Compilador

- SuperPipelined
  - Os Pipelines têm os seus estágios mais complicados subdivididos
  - Ex.: 8 estágios pipeline de inst. MIPS R4000

    14 do PPro
- Long Instruction Word (LIW)
  - Super escalar
  - Quem identifica a independência é o compilador
  - For a de uso

Memória

- O fluxo de dados da memória para o proc. é a parte mais crítica do computador.
- As memórias são mais lentas e cada vez maiores
- Quanto mais rápidas mais caras são

Tipos de Memórias

- Dynamic random access memory (DRAM)
- Static random access memory (SRAM)
- Random – vc pode acessar em qualquer ordem
  - Contra – Ex.: fitas magnéticas

- Dinâmica
  - Os bits são armazenados em pequenos capacitores que perdem sua carga, e portanto precisam ser "recarregados" de tempos em tempos.
  - Maior densidade
  - melhor preço / desempenho
  - Pode chegar a 50 ns

- Estática
  - não precisa ser "recarregada"
  - Utilizada em caches
  - É mais rápida podendo chegar a 7ns

Tempo de acesso

- Definição:
  - Tempo para acessar um dado na memória
- Tempo do ciclo da memória
  - Tempo para a memória poder repetir um acesso
- Ex.:
  - 50 ns para acessar um dado
  - 100 ns para poder pedir para acessar outro dado

Comparação Proc. x Mem.

- 1980 PC XT 4.77 Mhz
  - Proc. 210 ns
  - Mem. 200 ns
- 199...(alto) Pentium II 300 Mhz
  - Proc. 3 ns
  - Men. 50 ns

Hierarquia de memória

- Razões econômicas
  - Fica caro utilizar só SRAM

Razões econômicas

- Fica caro utilizar só SRAM

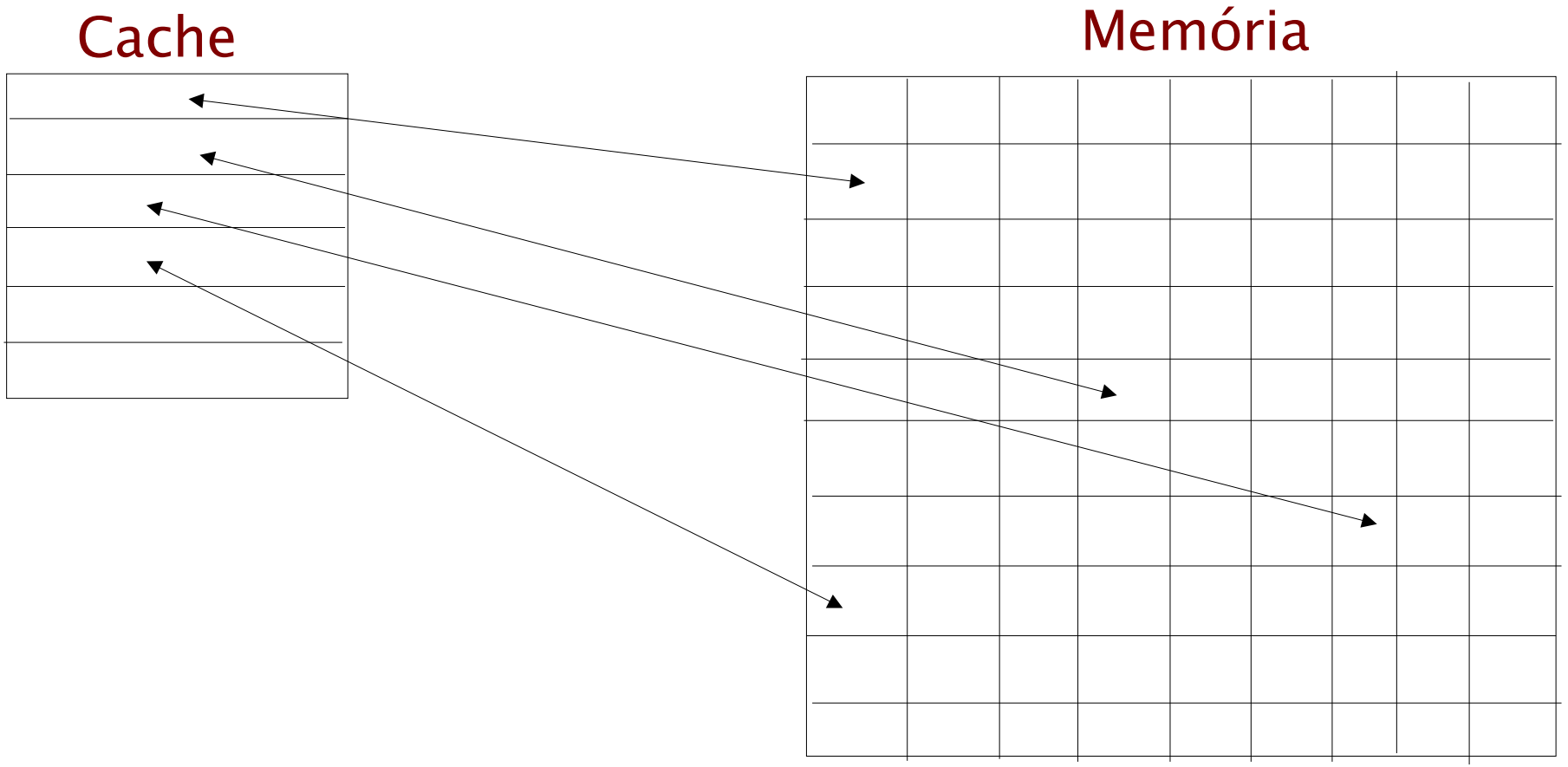
- Ex.: DEC ALPHA 21164
  - Registradores – 2 ns
  - L1 – 4 ns
  - L2 – 5 ns
  - L3 – 30 ns
  - RAM – 220 ns

Registradores

- Operam na frequência do Proc.
- Desempenho significa manter os "dados" o maior tempo possivel nos registradores
  - Evita o movimento entre caches e RAM
- Não é pratico colocar mais registradores na CPU

Caches

- Memória mais rápida SRAM



- L1 é interno L2 pode ser interno tb.
- L2 externo não funciona a mesma velocidade que o processador
- L3 é sempre externo
- Toda referência a uma posição da memória, o dado é copiado para o cache. É necessário manter a coerência entre cache e memória
- ***Hit*** – quando uma referência é encontrado no cache
  - Valores considerados razoavéis
    - L1 – 90% :L2 – 50%
    - Ex.: L1 – 10ns (75%) L2 – 30ns(20%) RAM – 300ns (5%)
    - Desempenho médio:
      - 0.75*10 + 0.20*30 + 0.05*300 = 28.5ns

- Quando um dado não é encontrado no cache é necessário "pegar" na memória. Como o cache é limitado é necessário eliminar uma linha e substituí-la pela nova. (MISS)
- Reorganizar o algoritmo para utilizar o máximo possível dos dados no cache.

Memória Virtual

- A memória pode não ser suficiente para rodar um programa.
- Ilusão de que o endereçamento é contínuo.
- Na verdade a memória é dividida em páginas.
- Quando um dado é requisitado é calculado o seu endereço e a página (tabela de páginas) qual ele pertence: ai é determinada a localização física.
- O gerenciamento das páginas é feito pela TBL
  - Translation Lookaside Buffer
    - "Cache para a possição das variaveis nas páginas"
-

- Se o dado não exite na pág. é necessário carregar uma outra página (TBL miss)
- Se o programa for orientado para reutilizar o cache a TBL vai se comportar bem!!!!
- Quando um dado é requisitado e não está no cache (cache miss), se o endereço não está na TBL (TBL miss), consulta-se a tabela de páginas. Se a pág. não estiver na memória é necessária a escrita da pagina atual no disco e a leitura ou a criação de outra página.
  - É chamado de paginação (Page Faults - swap)
  - Isso é um acidente!!!!!!