

DESEMPENHO DO COMPUTADOR

Análise e Desenvolvimento de Sistemas | FAQI

Organização de Computadores | Aula 02



Tópicos

- Velocidade do Processador
- Balanço do desempenho
- Melhorias na organização e na arquitetura do chip
- Evolução da arquitetura Intel x86
- Sistemas embarcados e o ARM
- Avaliação de desempenho

PROJETANDO VISANDO AO DESEMPENHO

Evolução e desempenho do computador

Velocidade

- Com a lei de Moore os fabricantes oferecem uma nova geração de chips a cada 3 anos.
 - Com 4 vezes a quantidade de transistores.
 - Como?
 - Acrescentando novo circuitos e;
 - Diminuindo a distância entre os circuitos.
- Chips de memória DRAM a capacidade é quadruplicada a cada 4 anos.

Velocidade

- Velocidade do processador **não** depende somente do volume de circuitos contidos no chip.
- Interfere também o fluxo constante de trabalho para fazer na forma de instruções. Algumas das técnicas utilizadas:
 - Previsão de desvio
 - Análise de fluxo de dados
 - Execução especulativa

Velocidade - técnicas utilizadas

- Previsão de desvio

- O processador antecipa o código de instrução apanhado da memória e prevê quais desvios ou grupos de instruções, provavelmente serão processados.

- Análise de fluxo de dados

- O processador analisa quais instruções são dependentes dos resultados uma da outra, ou dos dados, para criar uma sequência otimizada de instruções.

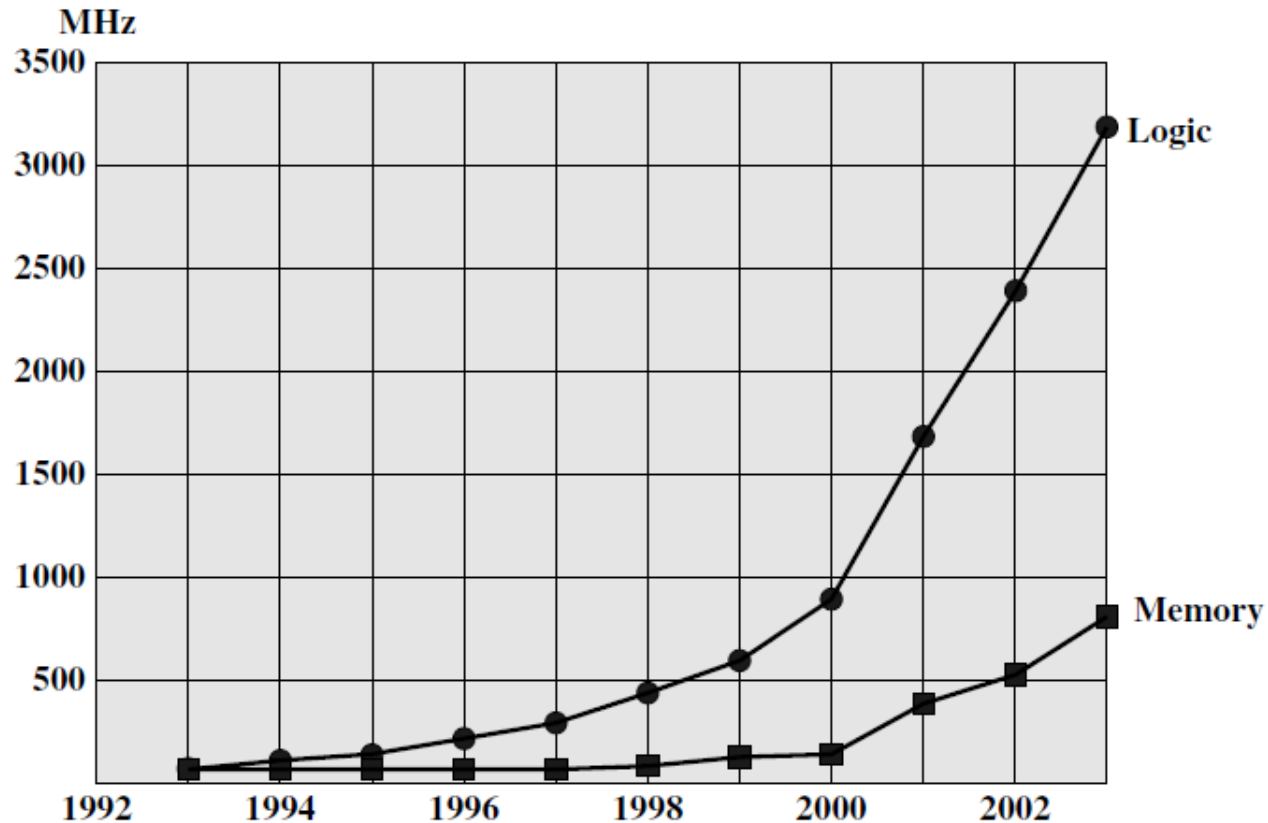
- Execução especulativa

- Alguns processadores especulativamente executam instruções antes do seu surgimento real na execução do programa e guardam o resultados em locais temporários.
- Permite os mecanismos de execução mais ocupados possíveis.

Problemas:

- Potência do processador com velocidade espantosa.
- Outros componentes do computador não a acompanharam.
- Necessidade:
 - Procurar o equilíbrio do desempenho.
 - Um ajuste da organização e da arquitetura para compensar a diferença entre as capacidades dos diversos componentes.

Problema 1: MP



- Exemplo:

- Velocidade do processador muito rápidas
- Velocidade de transferência de dados entre MP e processador não.:-(

- Resultado:

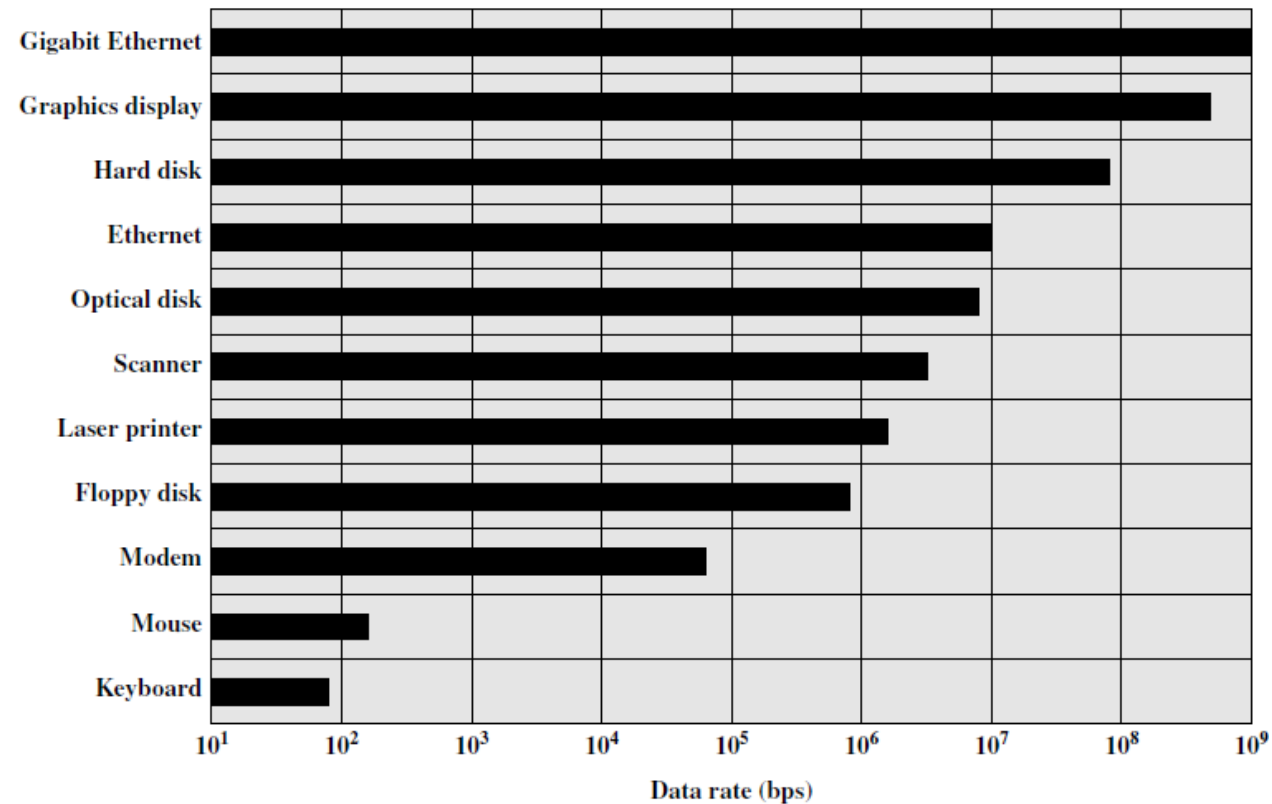
- Estado de espera → Perda de desempenho.

Balanço do Desempenho

- Maneiras de atacar o problema:
 1. Aumentar número de bits que são recuperados ao mesmo tempo.
 2. Alterar a interface da DRAM, incluindo uma cache ou outro esquema de *buffering* no chip de DRAM.
 3. Reduzir a frequência de acesso à memória incorporando estruturas de cache cada vez mais complexas e eficientes entre o processador e a memória cache.
 - Incorporar níveis de caches no chip do processador, ou fora do chip, próximo do processador.
 4. Aumentar a largura de banda de interconexão entre processador e MP.

Problema 2: E/S

Taxas de dados típicas dos dispositivos de E/S



Exemplo: Dispositivos de E/S

- Criam demandas consideráveis de vazão de dados.
- Problema de movimentar os dados entre o processador e o periférico.

Balanço do Desempenho

- Maneiras de atacar o problema:
 1. Esquemas de *caching* e *buffering*.
 2. Utilização de interconexão de maior velocidade e mais elaboradas.
 3. Utilização de configurações de processador múltiplo pode auxiliar a satisfazer as demandas de E/S.

Balanço do Desempenho

- O projeto de um computador precisa ser constantemente repensado para lidar com dois fatores em constante evolução:
 - A taxa de desempenho que difere bastante de um dispositivo para outro.
 - Novas aplicações e novos dispositivos periféricos.

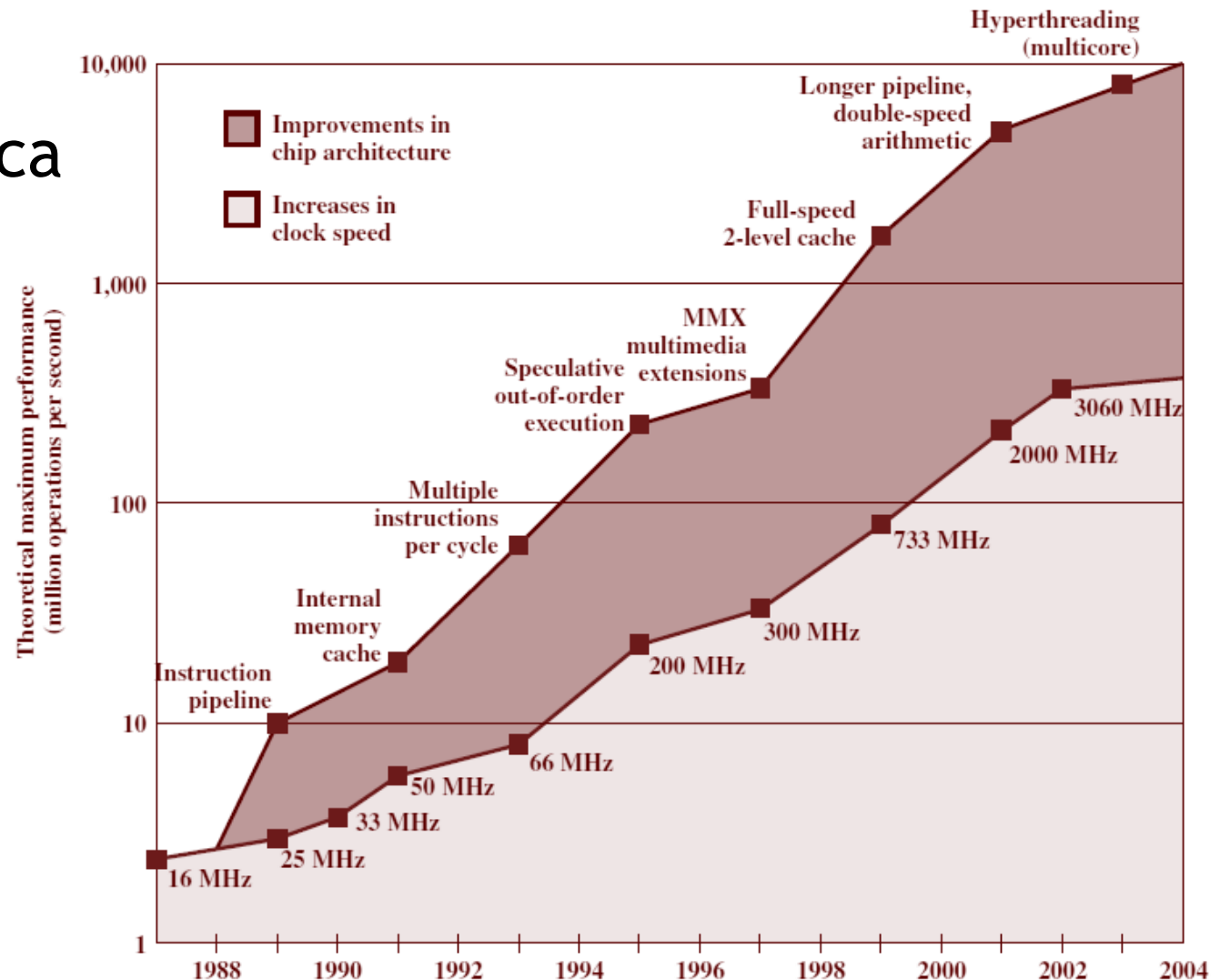
Melhorias na organização e na arquitetura do chip

- Como aumentar a velocidade do processador e balancear o desempenho do processador com a memória principal e outros componentes do computador?
 - Aumentar a velocidade do hardware.
 - Aumentar o tamanho e a velocidade das caches:
 - Fazer mudanças na organização e na arquitetura do processador, que aumentam a velocidade efetiva da execução da instrução. Basicamente, paralelismo.

Melhorias na organização e na arquitetura do chip

- A medida que a velocidade do *clock* e a densidade lógica aumentam, diversos obstáculos se tornam mais significativos:

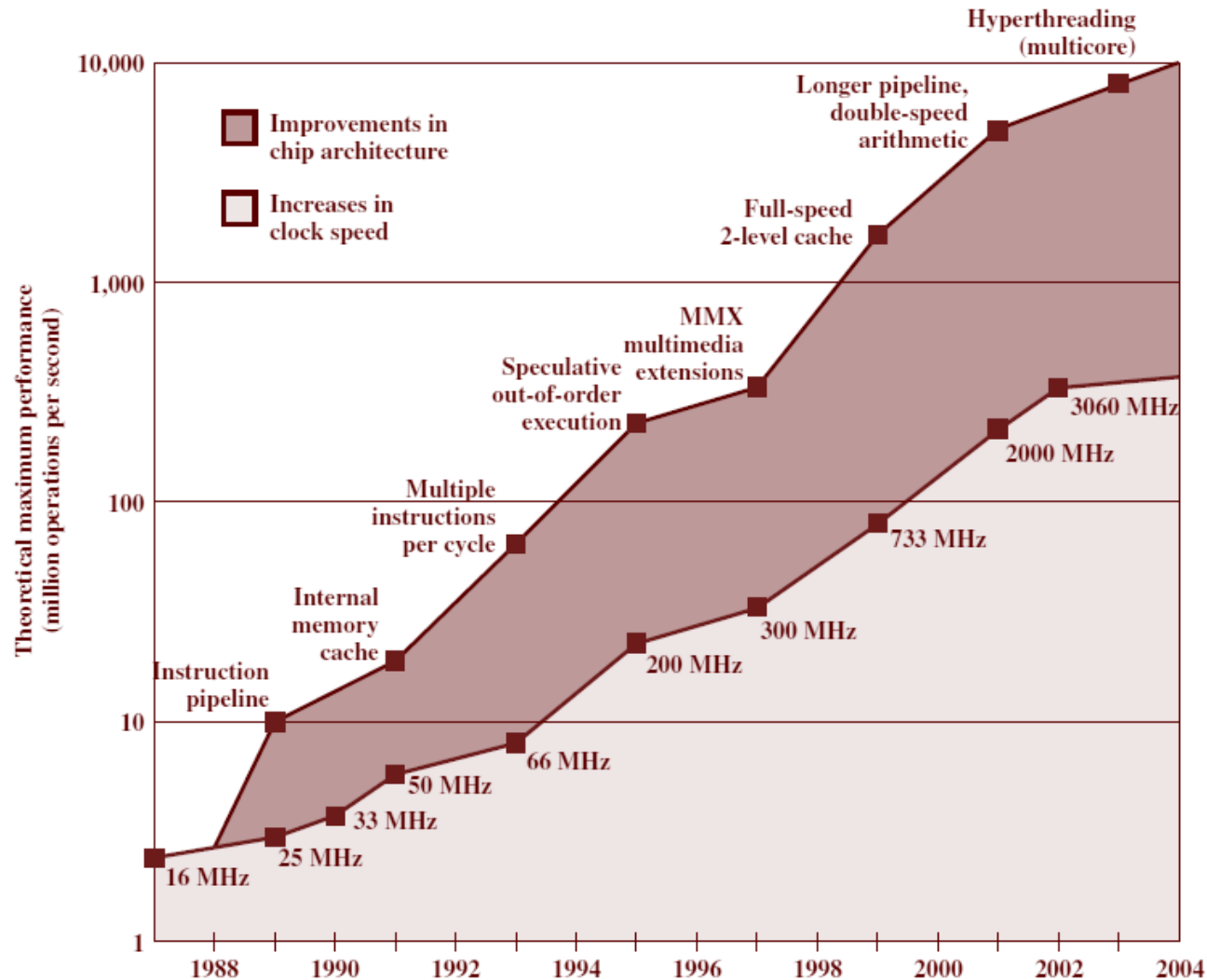
- Potência
- Atraso de RC
- Latência de memória



Melhorias na organização e na arquitetura do chip

- Potência:
 - À medida que a densidade da lógica e a velocidade do clock em um chip aumentam, também aumenta a densidade de potência (Watts/cm²)
- Atraso de RC:
 - A velocidade que os elétrons podem fluir em um chip entre os transistores é limitada pela resistência (R) e capacitância.
 - À medida que os componentes no chip diminuem de tamanho:
 - As interconexões de fio se tornam mais finas, aumentando a resistência.
 - Os fios mais próximos aumentam a capacitância.
- Latência de memória
 - As velocidades de memória limitam as velocidades do processador.

Melhorias na organização e na arquitetura do chip



- Aumento da capacidade de cache:

- A medida que a densidade aumenta - mais cache é incorporada no chip.

- Lógica de execução das instruções:

- Execuções paralelas dentro do processador.

Melhorias na organização e na arquitetura do chip

- Problema:

- Aumento da taxa de *clock* para aumentar o desempenho faz com que o problema da dissipação de calor torne-se um grande problema.
- Quanto maior a taxa de *clock*, maior a quantidade de potência dissipada.

- Solução:

- Colocar múltiplos processadores no mesmo chip → *Multicore*
- Oferece o potencial de aumentar o desempenho sem aumentar a taxa de *clock*.

Melhoria na organização e na arquitetura do chip

- Vídeo.
 - Processadores 3D Intel
 - <http://youtu.be/j0y2aUAh9GE>

Evolução da arquitetura Intel x86

- Tarefa: Leitura:
- <http://datacenter10.blogspot.com.br/2008/12/evolucao-dos-processadores-x86.html>
- <http://www.creci-rs.gov.br/gestor/php/noticias/pdf/Evolu%E7%E3o%20dos%20Processadores.pdf>

Sistemas embarcados e o ARM

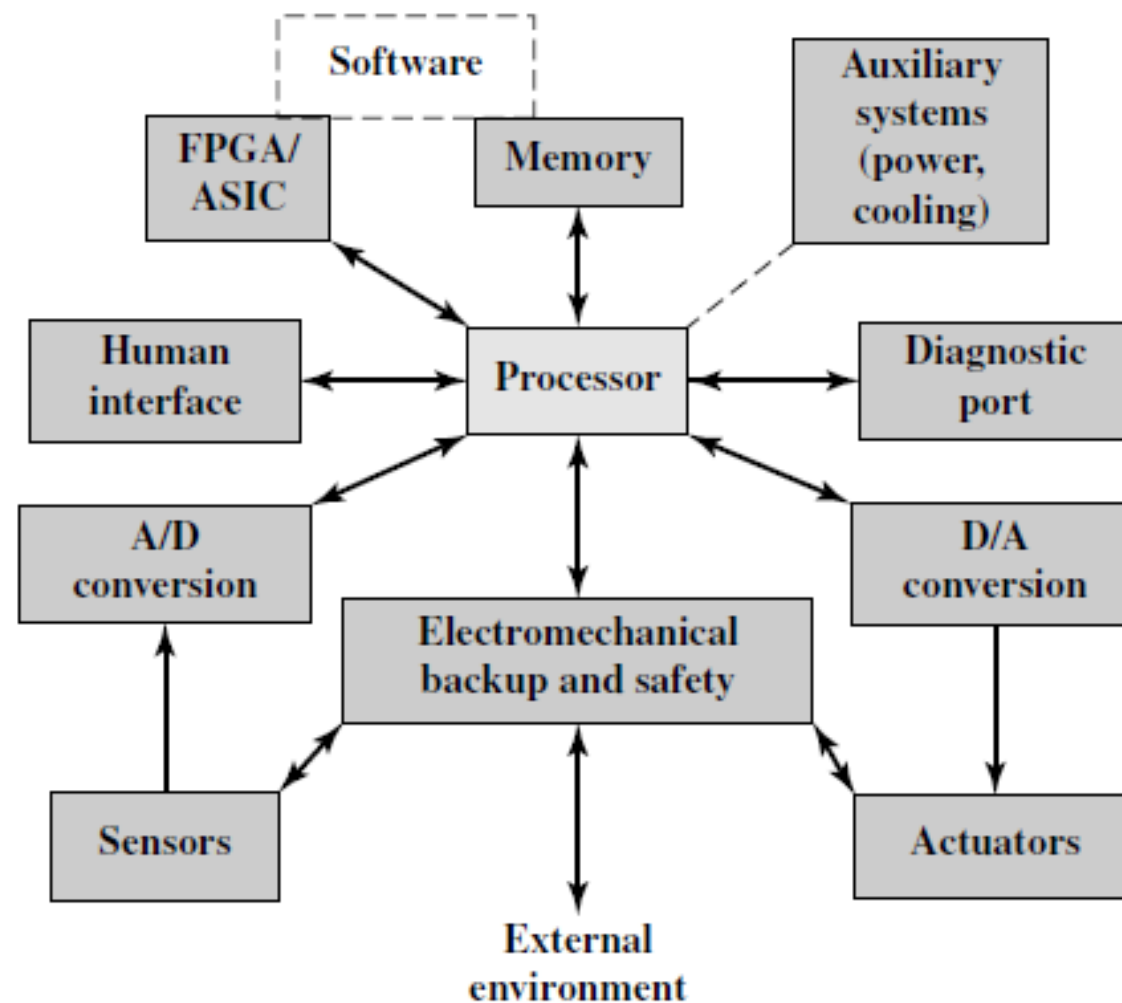
- Arquitetura ARM refere-se a uma arquitetura RISC.
- É utilizado em sistemas embarcados.
 - Uso de eletrônica e software dentro de um produto.
- Possuem requisitos:
 - Sistemas de pequenos a grandes, implicando restrições de custo muito diferentes.
 - Tempo de vida de curto a longo prazo.
 - Diferentes condições ambientais: radiação, vibrações e umidade.
 - Diferentes características de aplicação.
 - Diferentes modelos de computação

Sistemas embarcados e o ARM

- Normalmente são sistema fortemente acoplados.
- Possuem restrições:
 - Tempo real - necessidade de interagir com o ambiente.
 - Velocidades de movimento exigidas.
 - Precisão de medição e duração de tempo exigidas.
 - Se múltiplas atividades estiverem sendo gerenciadas simultaneamente - restrições de tempo real complexas.

Organização de sistema embarcado

- Além de processador e memória existem outros elementos que diferem de um computador típico:
 - O software normalmente tem uma função fixa e é específico à aplicação.
 - Pode haver uma variedade de interfaces que permitem que o sistema meça, manipule e interaja de outras maneiras com o ambiente externo.



Evolução do ARM

- Tarefa:
 - Pesquise sobre a evolução deste tipo de processador.
 - Identifique atualmente que tipo de dispositivos utiliza esta tecnologia.

AVALIAÇÃO DE DESEMPENHO

Evolução e desempenho do computador

Avaliação de desempenho

- Desempenho é dos principais **parâmetros** a se considerar na definição de requisitos de sistemas computacionais.
- Dificuldade de fazer comparações de desempenho.
 - Mesmo entre os processadores da mesma família.
 - Velocidade bruta é menos importante do que como o processador funciona.
 - Desempenho de uma aplicação depende do conjunto de instruções.
 - Escolha da linguagem de programação
 - Eficiência do compilador
- **Vamos compreender as razões!**

Velocidade do *clock* e instruções por segundo

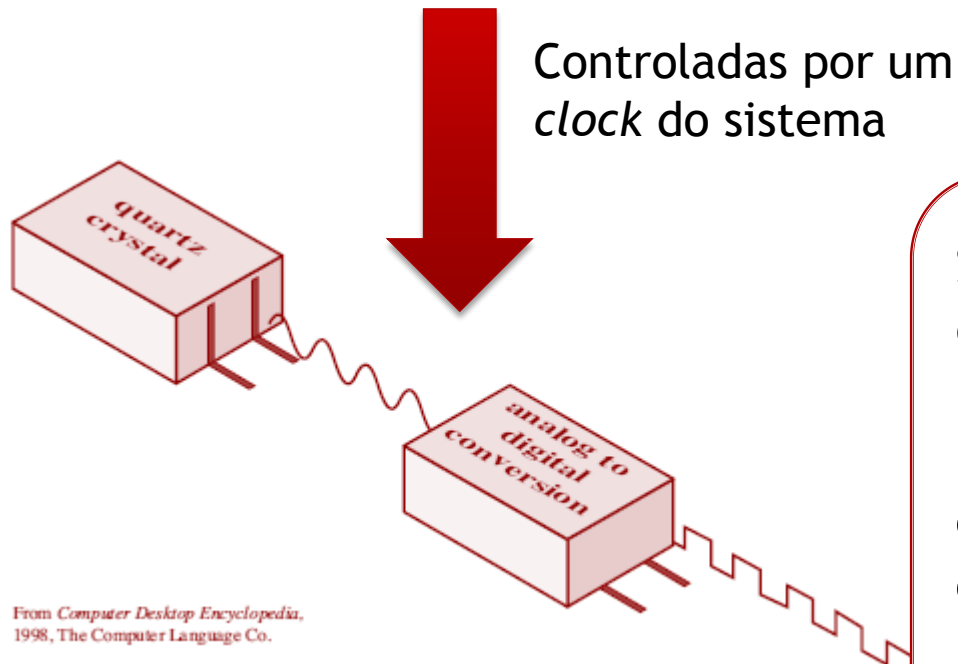
- Clock do sistema:

As operações realizadas por um processador:

- Busca e decodificação de uma instrução.
- Realização de uma operação aritmética.

- Velocidade do processador:

- Ditada pela frequência de pulso produzida pelo clock.
- Medida em ciclos por segundos ou *Hertz* (Hz)



Sinais de *clock* são gerados por um cristal de quartzo, que gera uma onda de sinal constante.

Esta onda é convertida em um *stream* de pulsos em voltagem digital, que é fornecido em um fluxo constante aos circuitos do processador.

Velocidade do *clock* e instruções por segundo

Exemplo:

- Um processador de 1 GHz recebe 1 bilhão de **pulsos** por segundo.

Conceitos:

- Taxa de **pulsos** é conhecida como:
 - taxa de *clock*, ou velocidade de *clock*.
- Pulso de *clock* é conhecido como:
 - Um ciclo de *clock* ou um *clock tick*.
- O tempo entre os pulsos é o **tempo de ciclo**.

Velocidade do *clock* e instruções por segundo

- **Estabilidade do sinal:** Os sinais emitidos pela taxa de *clock* precisam de uma quantidade finita de tempo para os níveis de voltagem se estabilizem, para o valor preciso (0 ou 1)
- **Sincronização:** Alguns sinais, dependendo do layout físico dos circuitos do processador mudam mais rapidamente do que outros.
 - Operações precisam ser sincronizadas e ritmadas de modo que valores de sinal elétrico (voltagem) apropriados estejam disponíveis para cada operação.

Velocidade do *clock* e instruções por segundo

- **Execução de instruções:**
 - Envolve uma série de etapas discretas.
 - Grande parte das instruções requer múltiplos ciclos de *clock* para completar.
 - Algumas instruções utilizam apenas alguns ciclos.
 - Outras, exigem dezenas.
- Por estas razões comparar diretamente a velocidade do processador não diz a história toda sobre o desempenho.

Velocidade do *clock* e instruções por segundo

- Taxa de execução de instrução:

- Processador é controlado por um *clock* com uma frequência constante f .
- Ou, tempo de ciclo constante τ , onde:

$$\tau = 1/f$$

- Média de ciclos por instrução(CPI):

- Se todas as instruções exigissem o mesmo número de ciclos de clock, então o CPI seria um valor constante para um processador.
- Porém, o número de ciclos de *clock* exigidos varia para diferentes tipos de instruções.

Velocidade do *clock* e instruções por segundo

- Considere que CPI_i , seja o número de ciclos exigidos para a instrução tipo i ;
- I_i seja o número de instruções executadas de tipo i para determinado programa.
- I_C = número de instruções de máquina executadas para esse programa até que ele rode até o fim ou por algum intervalo de tempo definido.
- Logo, pode-se calcular um CPI geral como a seguir:

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_C}$$

Velocidade do *clock* e instruções por segundo

- **Problema 1:**

- Um projetista de compilador está tentando decidir entre duas sequências de código para uma máquina específica. Com base na implementação de hardware, existem três classes diferentes de instruções: classe A, classe B e classe C, e exigem um, dois e três ciclos por instrução, respectivamente.
- A primeira sequência de código possui 5 instruções:
 - 2 de A
 - 1 de B
 - 2 de C
- A segunda sequência possui 6 instruções:
 - 4 de A
 - 1 de B
 - 1 de C

Primeiro sempre descobrir o número de ciclos. A parte de cima da fórmula lhe dá esta resposta.

A parte de baixo da fórmula sempre necessita do número de instruções.

Velocidade do *clock* e instruções por segundo

- **Problema 1:**

- Calcular os ciclos de CPU para cada sequência. Qual sequência é mais rápida?
- Qual é o CPI para cada sequência?
- Lembre-se que a fórmula é um somatório das execuções das instruções.

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_C}$$

- IC 1ª sequência: 5
- IC 2ª sequência: 6

- **SOLUÇÃO:**

- Primeira sequência: $(2 \times 1) + (1 \times 2) + (2 \times 3) = 10$ ciclos. $CPI = 10/5 \rightarrow 2,0$
- Segunda sequência: $(4 \times 1) + (1 \times 2) + (1 \times 3) = 9$ ciclos. **$CPI = 9/6 \rightarrow 1,5$**

Velocidade do *clock* e instruções por segundo

- **Problema 2:**

- Dado um mix de instruções de um programa em um processador RISC:
- Qual é a CPI média?
- Qual é o **percentual** de tempo utilizado por cada classe de instrução?

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_C}$$

Classe _i	Frequência _i	CPI _i	CPI _i x Frequência _i	% Tempo
ALU	50 %	1		
LOAD	20 %	5		
STORE	10 %	3		
BRANCH	20%	2		

Velocidade do *clock* e instruções por segundo

• Problema 2:

- Dado um mix de instruções de um programa em um processador RISC:
 - Qual é a CPI média? $0.5+1.0+0.3+0.4 \rightarrow 2,2$
 - Qual é o percentual de tempo utilizado por cada classe de instrução?

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_c}$$

Classe _i	Frequência _i	CPI _i	CPI _i x Frequência _i	% Tempo
ALU	50 %	1	0,5 X 1 = 0,5	0,5/2,2 = 23 %
LOAD	20 %	5	0,2 X 5 = 1,0	1,0/2,2 = 45 %
STORE	10 %	3	0,1 X 3 = 0,3	0,3/2,0 = 14 %
BRANCH	20%	2	0,2 X 2 = 0,4	0,4/2,2 = 18%

SOMA = 2,2

Velocidade do *clock* e instruções por segundo

- **Problema 3:** um dado programa consiste de um laço de 100 instruções que é executado 42 vezes. Se demora 16.000 ciclos para executar o programa em um dado sistema, qual o valor de CPI do sistema para este programa?
- **Solução:**
 - Total de instruções: $42 \times 100 = 4200$
 - Demora 16.000 ciclos para executar o programa.
 - Logo, $16.000 \text{ ciclos} / 4.200 \text{ instruções} = 3,80 \text{ ciclos por instrução (CPI)}$

Velocidade do *clock* e instruções por segundo

- Tempo de processador T necessário para executar determinado programa pode ser expresso como:

$$T = I_c \times CPI \times \tau$$

Velocidade do *clock* e instruções por segundo

- *Millions of instructions per second (MIPS)*:
 - Uma medida comum do desempenho de um processador, conhecida como taxa de MIPS.
 - É expressa em termos da taxa de clock e do CPI da seguinte forma:

$$Taxa\ MIPS = \frac{I_i}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

- Exemplo: Considere a execução de um programa que resulta na execução de 2 milhões de instruções em um processador de 400 MHz. O programa consiste em quatro tipos principais de instruções. A mistura de instruções e o CPI para cada tipo de instrução aparecem na tabela.

Tipo de instrução	CPI	Número de Instruções (%)
Aritmética e lógica	1	60 %
Load/Store com acerto de cache	2	18 %
Desvio	4	12 %
Referência de memória com falha de cache	8	10 %

Velocidade do *clock* e instruções por segundo

- *Millions of instructions per second (MIPS)*:
 - Solução:

$$Taxa\ MIPS = \frac{I_i}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

Tipo de instrução	CPI	Número de Instruções (%)
Aritmética e lógica	1	60 %
Load/Store com acerto de cache	2	18 %
Desvio	4	12 %
Referência de memória com falha de cache	8	10 %

- $CPI = (1 \times 0,6) + (2 \times 0,18) + (4 \times 0,12) + (8 \times 0,10) = 2,24$
- $Taxa\ de\ MIPS = 400 \times 10^6 / 2,24 \times 10^6 = 175,57$

Velocidade do *clock* e instruções por segundo

- *Million of floating-point operation per second* (MFLOPS):
 - Outra medida de desempenho comum lida apenas com instruções de ponto flutuante.
 - Comuns em muitas aplicações científicas e de jogos.
- O desempenho do ponto flutuante é expresso como milhões de operações de ponto flutuante por segundo da seguinte forma:

$$\text{Taxa MFLOPS} = \frac{\text{Número de operações de ponto flutuante executadas em um programa}}{\text{Tempo de execução} \times 10^6}$$

Lei de Amdahl

- Ao considerar o desempenho do sistema, os projetistas de sistemas de computação procuram maneiras de melhorar o desempenho aperfeiçoando a tecnologia ou mudando o projeto.
- Exemplos:
 - Processadores paralelos
 - Uso de hierarquia de cache de memória
 - *Speedup* no tempo de acesso da memória e na taxa de transferência de E/S

Lei de Amdahl

- O alcance de speedup em um aspecto da tecnologia não resulta em uma melhoria correspondente no desempenho.
- Essa limitação é expressa de forma sucinta pela lei de Amdahl.
- A lei de Amdahl foi proposta inicialmente por Gene Amdahl em 1967 e lida com o potencial speedup de um programa usando múltiplos processadores em comparação com um único processador.

Benchmarks

- Medidas como MIPS e MFLOPS provaram ser inadequadas para avaliar o desempenho dos processadores.

Vídeos

- Velocidade dos Processadores Intel Core i3 i5 3 i7:
 - <http://youtu.be/ax38p4mJ-Do>
- Novos processadores 3D Intel
 - <http://youtu.be/j0y2aUAh9GE>
- Intel® 22nm 3D Tri-Gate Transistor Technology
 - <http://www.intel.com/content/www/us/en/energy/intel-22nm-3-d-tri-gate-transistor-technology.html>

DESEMPENHO DO COMPUTADOR

Análise e Desenvolvimento de Sistemas | FAQI

Organização de Computadores | Aula 02

