

# Arquitetura de Computadores Multicore

Ríad Mattos Nassiffe

# Problemas do Single Core?

- Limite de tecnologia
  - Frequência já estava no limite
  - Consumo excessivo de energia



# Multicore

- O que é multicore?



# Multicore

- Executar vários aplicativos de vez?



# Multicore

- Executar várias instruções em paralelo
  - Pipeline
    - Execução superescalar
    - Análise de fluxo de dados

# Computação Paralela – Conceitos

- Permite a execução das tarefas em menor tempo, através da execução em paralelo de diversas tarefas.
- O paralelismo pode ser obtido em diversos níveis, com ou sem o uso de linguagens de programação paralela.
- Arquiteturas de diversos tipos, elaboradas para aplicações específicas, podem ser utilizadas para acelerar a execução dessas aplicações.

# Computação Paralela – Conceitos

- Seu programa pode ser:
  - Programação Seqüencial
  - Programação Concorrente
    - Um servidor, atendendo vários clientes através de uma política de escalonamento no tempo.
  - Programação Paralela
    - Vários servidores, atendendo vários clientes simultaneamente no tempo

# Computação Paralela – Conceitos

- Tarefa
  - É a execução das instruções definidas no programa (Visualizar uma figura, editar ou aplicar um efeito a uma figura.)
- Processo
  - Uma tarefa pode conter vários processos ou até mesmo compartilhar
  - Contêiner de recursos utilizados para executar tarefas



# Como Paralelizar?

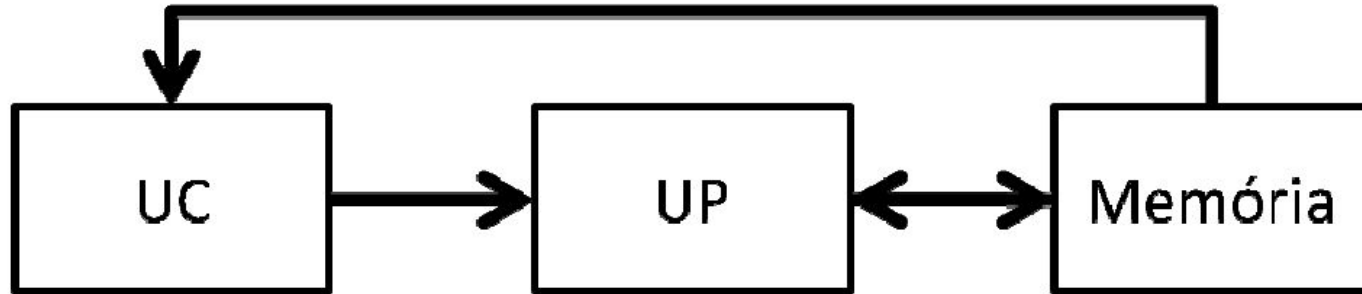
- Níveis de paralelismo
  - Instrução (granulosidade fina)
    - Paralelismo entre as instruções
    - Arquiteturas Pipeline, Superescalar
  - Tarefas (granulosidade média)
    - Paralelismo entre as threads
    - Arquiteturas SMT (Simultaneous MultiThreading)
  - Processos (granulosidade grossa)
    - Paralelismo entre os processos
    - Computação Paralela
    - Arquiteturas multiprocessadores e multicomputadores



# Arquiteturas Paralelas

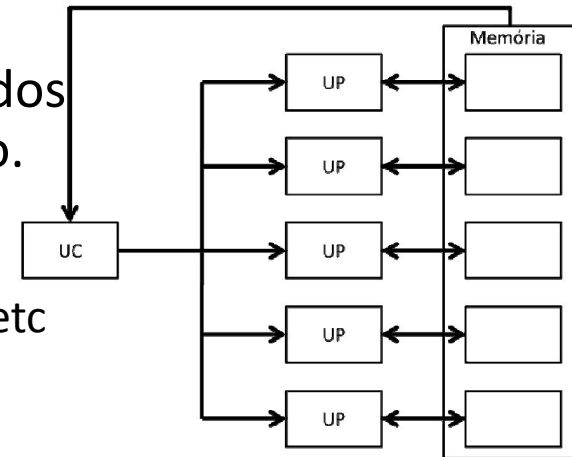
- A taxonomia de Flynn
  - É uma classificação de arquiteturas de computador que se baseia em **fluxo de instruções** e o **fluxo de dados**.

- SISD (Single Instruction Single Data): Fluxo único de instruções sobre um único conjunto de dados.
- Qualquer máquina com a arquitetura von Neumann



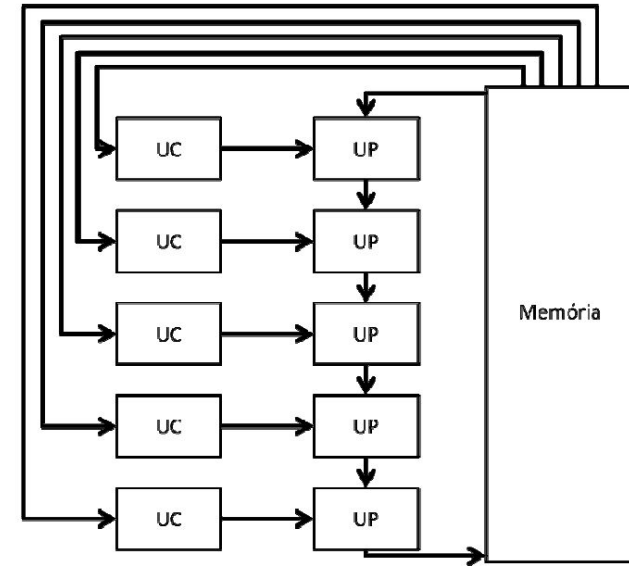
# SIMD

- SIMD (Single Instruction Multiple Data): Fluxo único de instruções em múltiplos conjuntos de dados.
- **Única** instrução executada sobre diferentes dados
- Bom para aplicar várias operações em um dado.
- Exemplos de aplicações:
  - Processamento de sinais e multimídia
  - Soma e multiplicação de constante em um vetor, etc



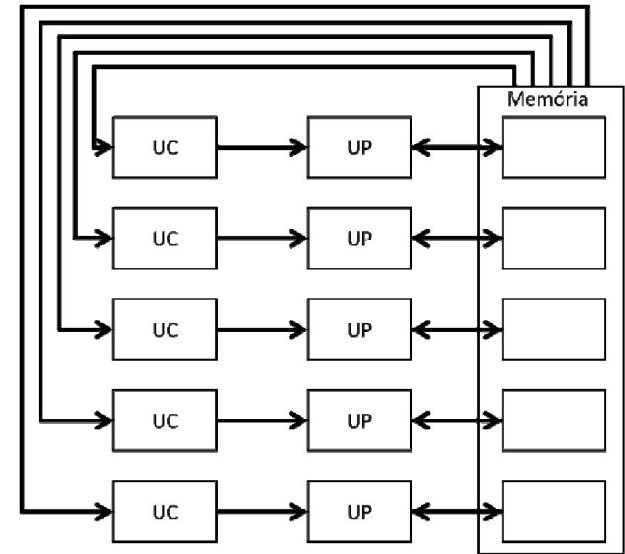
# MISD

- MISD (Multiple Instruction Single Data): Fluxo múltiplo de instruções em um único conjunto de dados.
- Executam a mesma instrução em diferentes dados.
- Exemplo:
  - Multiplicação e transposição de matrizes.
  - Sistemas com tolerância a falha.



# MIMD

- MIMD (Multiple Instruction Multiple Data): Fluxo múltiplo de instruções sobre múltiplos conjuntos de dados.
- Exemplo:
  - Multi-core superescalares
  - Sistemas distribuídos.



# Resumo da Taxonomia de Flynn

Fluxo de Instruções

Fluxo de Dados

	Único	Múltiplo
Único	SISD Single Instruction Single Data Instrução Simples de Dados Simples	SIMD Single Instruction Multiple Data Instrução Simples de Múltiplos Dados
Múltiplo	MISD Multiple Instruction Single Data Instrução Múltiplas de Dados Simples	MIMD Multiple Instruction Multiple Data Instrução Múltiplas de Dados Múltiplos

# Memória

- Classificação da memória:
  - Memória distribuída (distributed memory)
    - Memória implementada com vários módulos
    - Cada módulo fica próximo de um processador
  - Memória centralizada (centralized memory)
    - Memória encontra-se à mesma distância de todos os processadores  
Independentemente de ter sido implementada com um ou vários módulos



# Compartilhamento da Memória

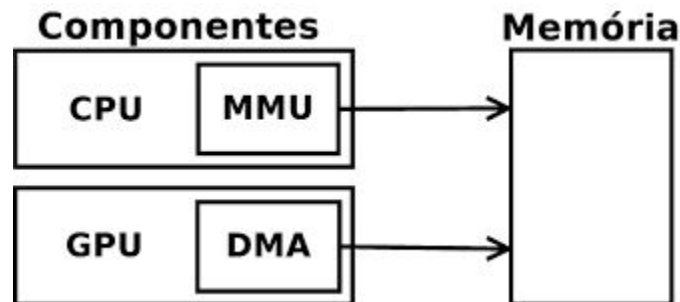
- Refere-se ao espaço de endereçamento dos processadores
  - Memória compartilhada (shared memory)
    - Único espaço de endereçamento usado para comunicação entre processadores
    - Processos acessam variáveis compartilhadas
    - Operações de load e store
  - Memória não compartilhada
    - Múltiplos espaços de endereçamento privados um para cada processador
    - Comunicação através de troca de mensagens operações send e receive
    - Conforme compartilhamento de memória as máquinas podem ser

# Formas de Acesso a Memória

- UMA (Uniform Memory Access)
- NUMA(non Uniform Memory Access)
- COMA (Cache-only memory architecture)

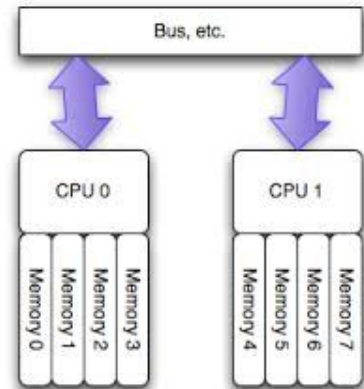
# UMA

- Memória centralizada
- Encontra-se à mesma distância de todos processadores
- Maior latência de acesso à memória
- Igual para todos processadores
- Infra-estrutura de comunicação
  - Barramento é a mais usada suporta apenas uma transação por vez
  - Outras infra-estruturas também se enquadram nesta categoria, se mantiverem
- Tempo uniforme de acesso



# Numa

- Memória Distribuída
- Espaço de endereçamento único
- Memória implementada com múltiplos módulos associados a cada processador
- Comunicação processador-memórias não locais através da infra-estrutura de comunicação
- Acesso à memória local mais rápido às demais
- Acesso não uniforme
- Distância das memórias variável depende do endereço



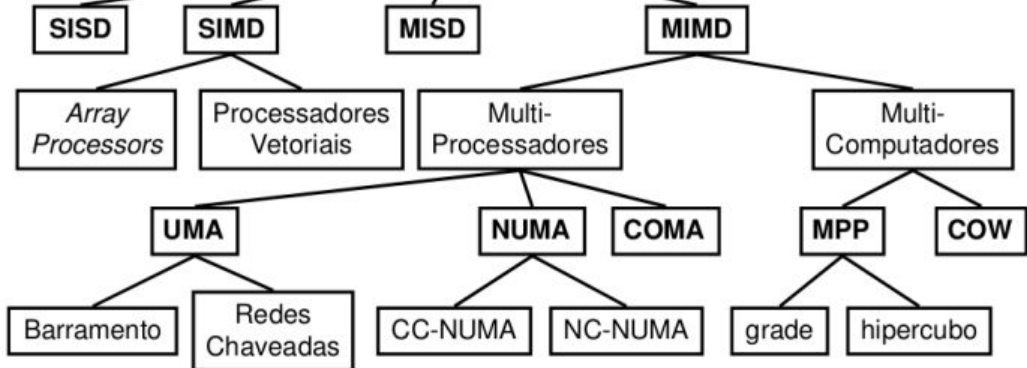
# COMA

- Memórias locais estão estruturadas como memórias cache
  - São chamadas de COMA caches
  - COMA caches têm muito mais capacidade que uma cache tradicional
- Arquiteturas COMA têm suporte de hardware para a replicação efetiva do mesmo bloco de cache em múltiplos nós
  - Reduz tempo global para pegar informações

# Multicomputadores

- São construídos a partir da replicação de toda arquitetura convencional, não apenas do processador, como nos multiprocessadores
- Tipicamente composto por “armários” compostos por dezenas de “gavetas” onde cada uma possui diversos processadores.
- Non-Remote Memory Access - NORMA

## ARQUITETURAS PARALELAS



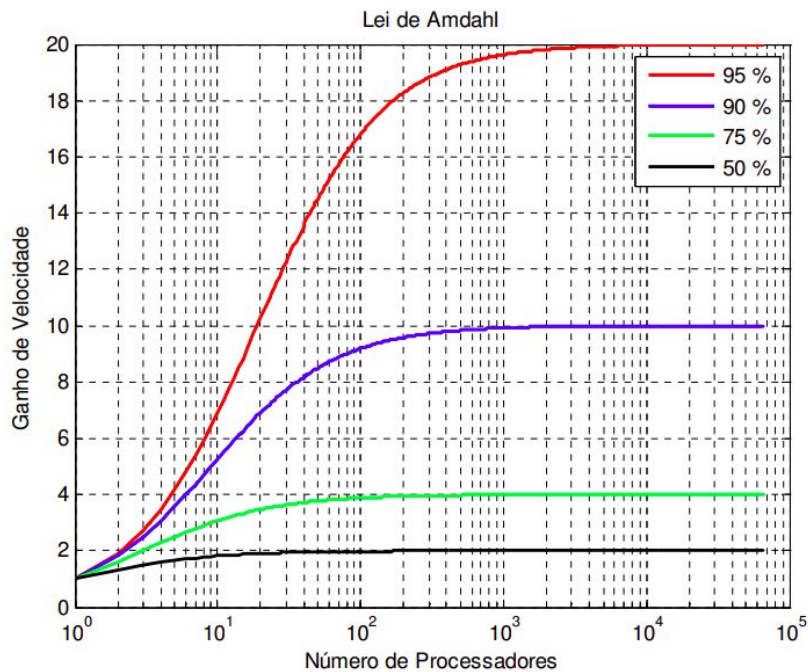
# Lei de Amdahl

- “O ganho de velocidade de um algoritmo rodando em múltiplos processadores é limitado pelo tempo consumido pela fração seqüencial do programa.”

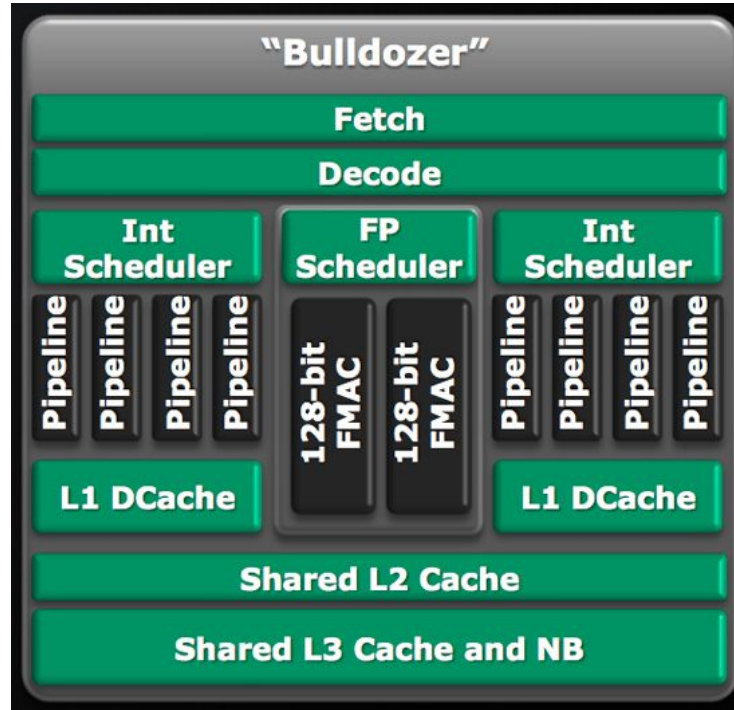
$$S = \frac{1}{1 - P + \frac{P}{N}}$$



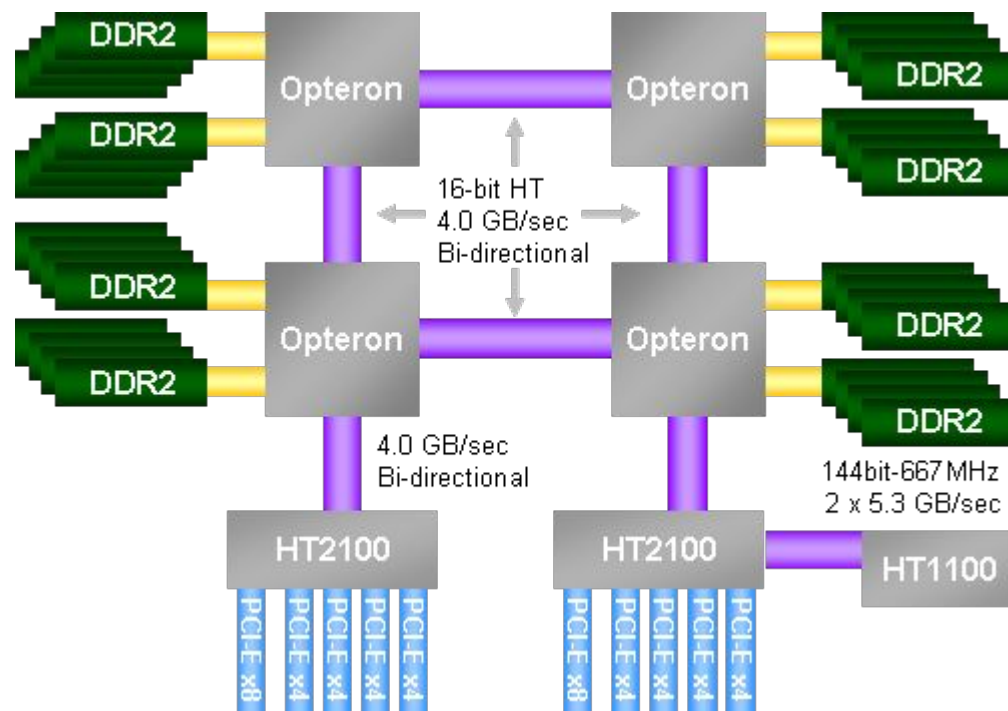
# Lei de Amdahl

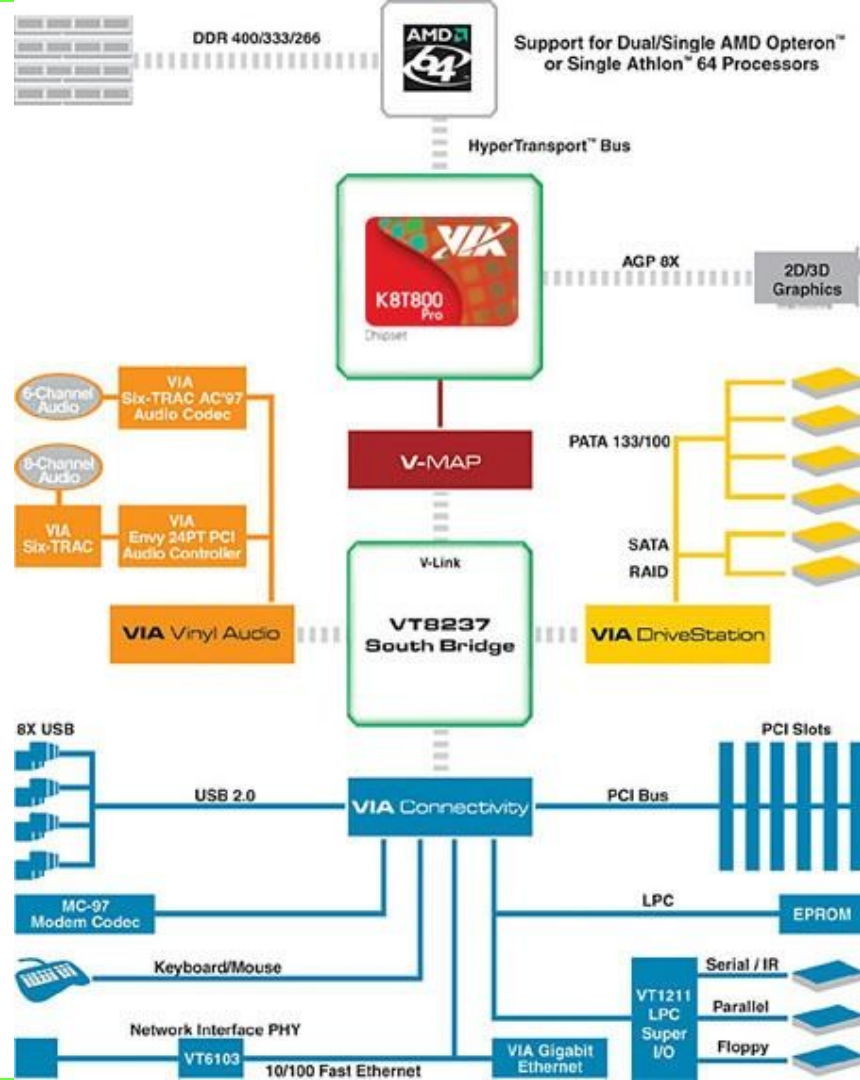


# Arquitetura do Opteron

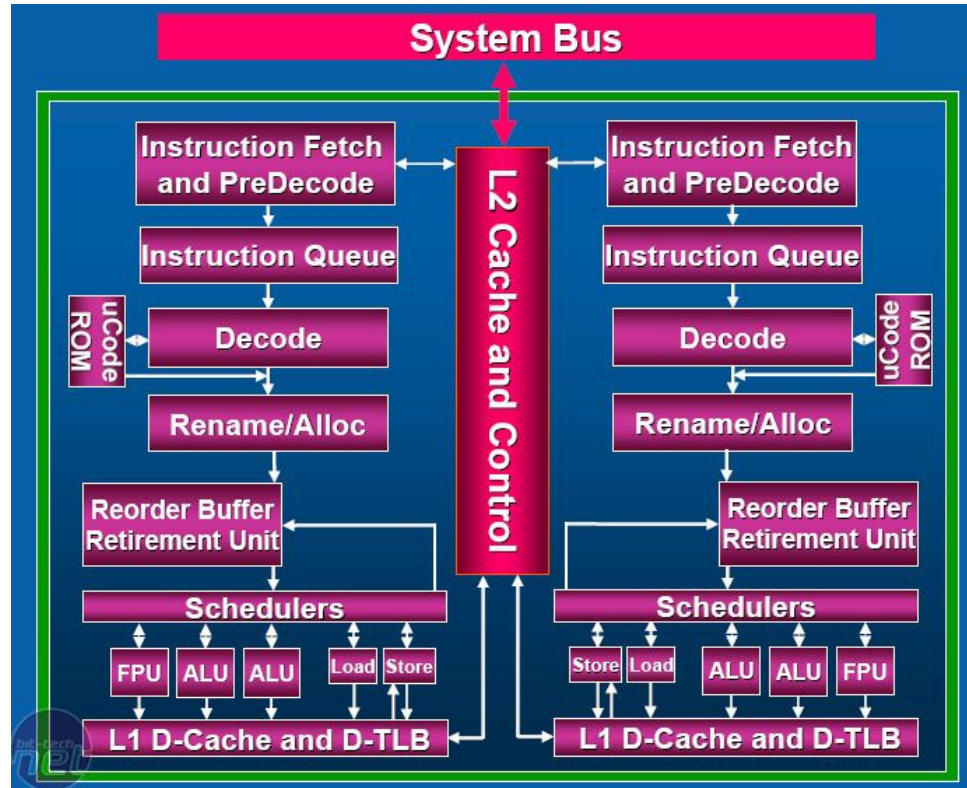


# Arquitetura do Opteron

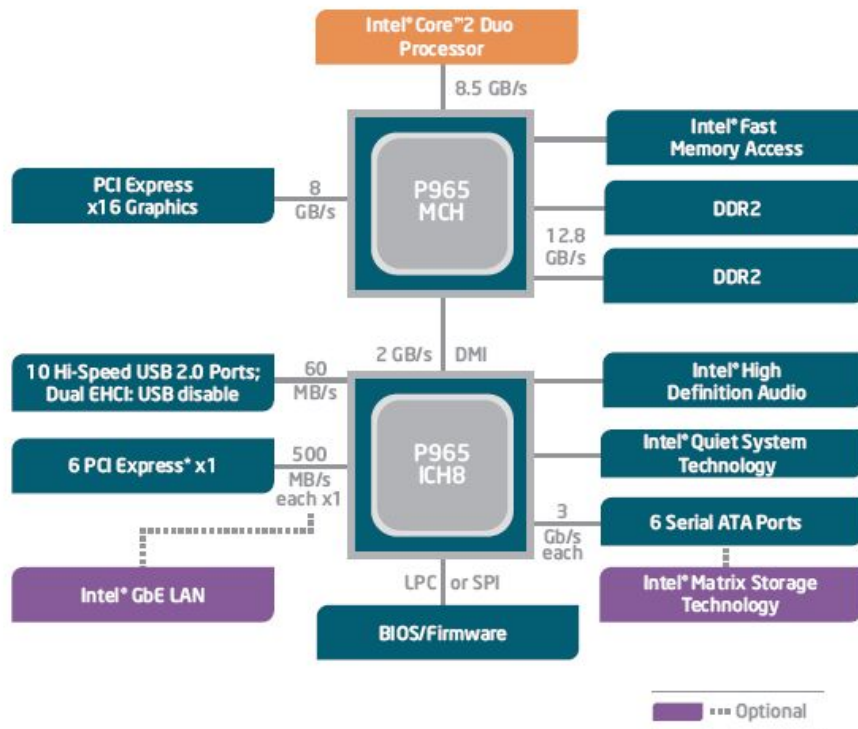


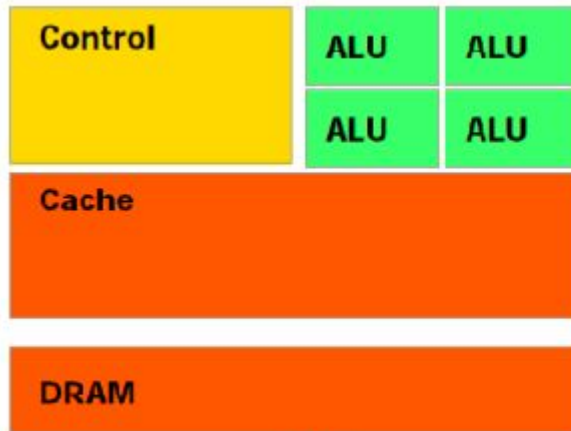


# Intel Core 2 Duo

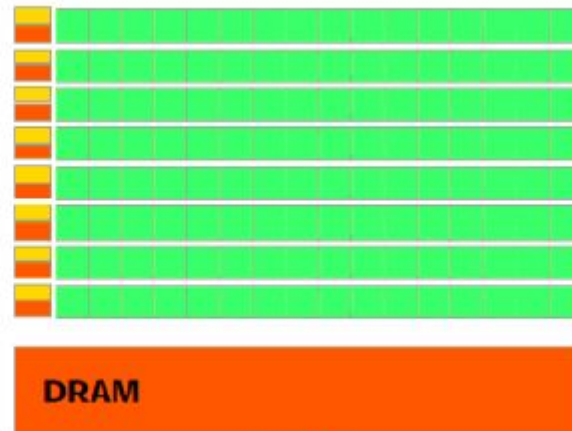


# Intel Core 2 Duo





**CPU**



**GPU**



# Dúvidas

- **Encontros presenciais.**
- **Encontros síncronos no google meet, marcados no calendário da turma e avisados no SIGAA.**

