



Universidade Tecnológica Federal do Paraná – UTFPR
Departamento de Ciência da Computação

Arquitetura e Organização de Computadores

Prof. Paulo C. Gonçalves

paulogoncalves@utfpr.edu.br

Aula de Hoje:

- Questões sobre desempenho de hardware
- Organização multicore
- Organização multicore x86 da Intel

Capítulo 18

William Stallings

Arquitetura e Organização de Computadores

8ª Edição

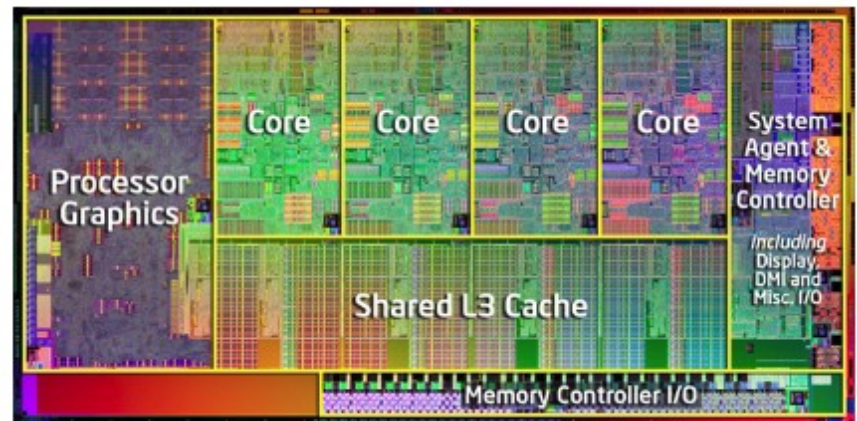
Computadores Multicore

- Definição:
 - Um computador **multicore**, conhecido também com **chip multiprocessador**
 - Combina dois ou mais processadores (chamados núcleos – core) em uma única peça de silício (chamada pastilha – die)

Computadores Multicore



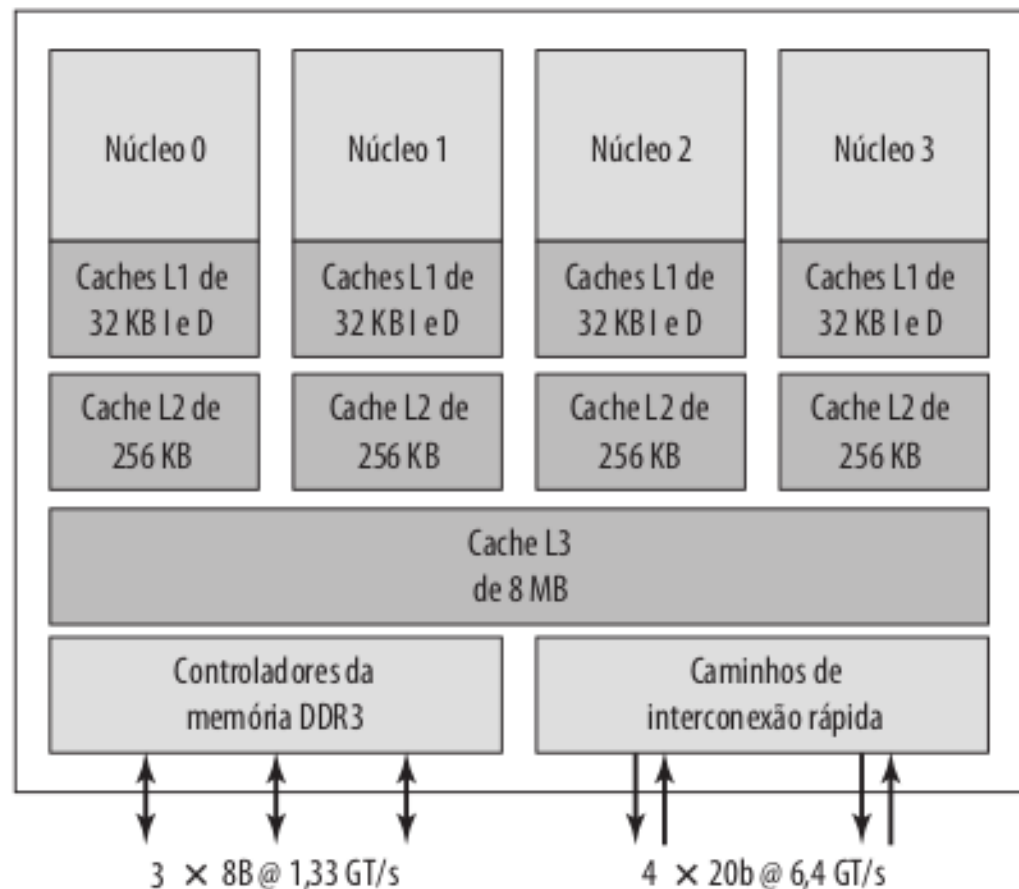
<https://www.zdnet.com/pictures/intel-celebrates-40-years-of-chipmaking-photos/34/>



<https://www.zdnet.com/pictures/intel-celebrates-40-years-of-chipmaking-photos/35/>

Organizações alternativas do chip

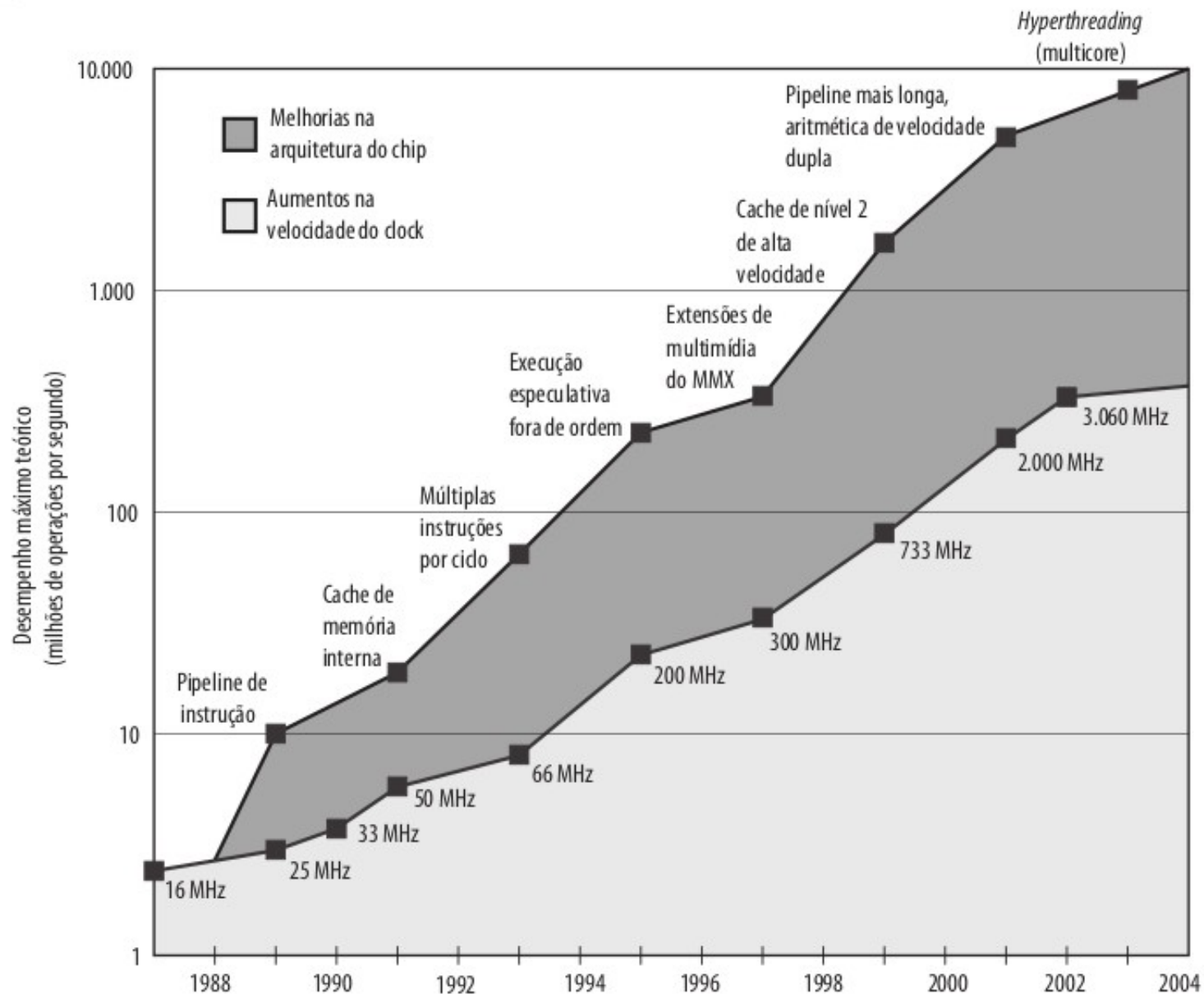
Figura 18.10 Diagrama de blocos do Intel Core i7



Questões de desempenho do hardware

- Microprocessadores viram um aumento exponencial no desempenho
 - Organização melhorada
 - Frequência de clock aumentada
- Aumento em paralelismo (ordem cronológica):
 - **Pipeline**
 - **Superescalar (pipeline paralelo)**
 - **Multithreading simultâneo (SMT)**
 - **Multicore**

Figura 2.12 Desempenho do microprocessador Intel (Gibbs, 2004ⁿ)

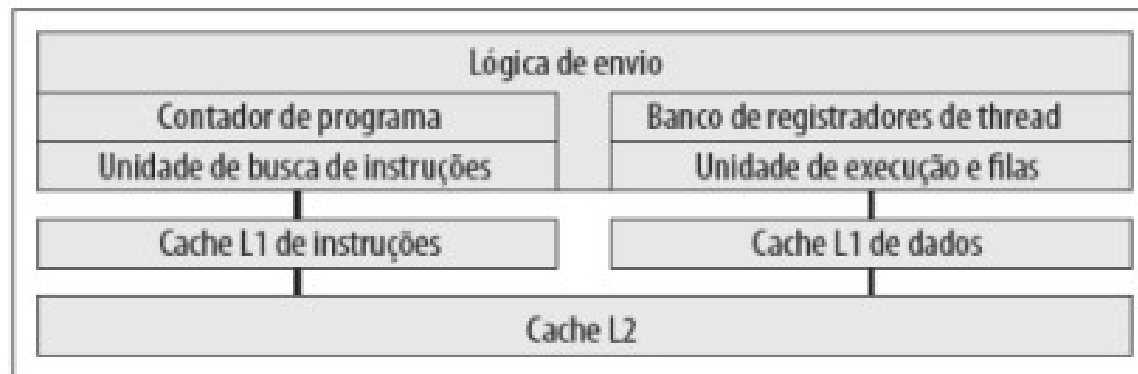


Questões de desempenho do hardware

Pipeline: instruções individuais são executadas por um pipeline de estágios de tal forma que, durante a execução de uma instrução em um estágio, outra instrução é executada em outro estágio do pipeline

Questões de desempenho do hardware

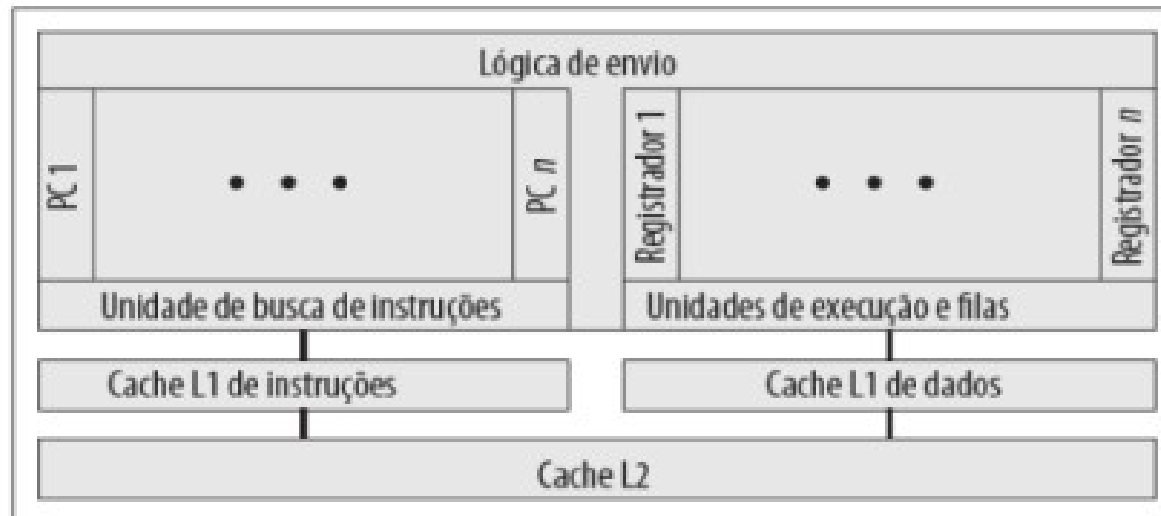
Superescalar: vários pipelines são construídos pela replicação de recursos da execução. Isto possibilita execução paralela de instruções em pipeline paralelos, assim que os hazards (conflitos) são evitados



(a) Superescalar

Questões de desempenho do hardware

Multithreading simultâneo (SMT): bancos de registradores são replicados para que várias threads possam compartilhar o uso dos recursos do pipeline

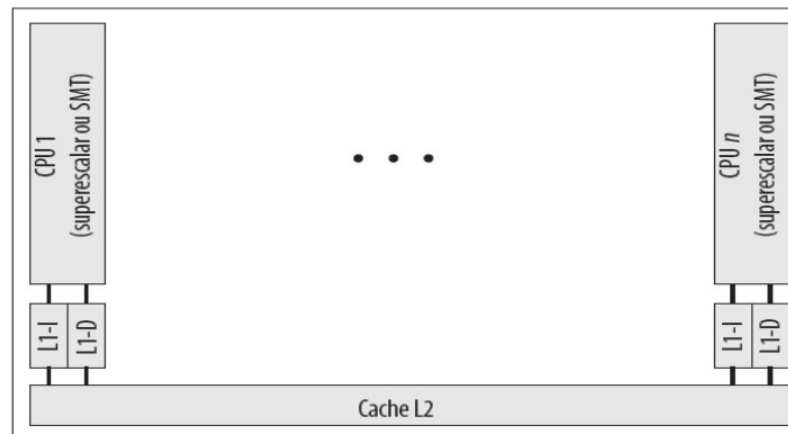


(b) Multithreading simultâneo

Organizações alternativas do chip

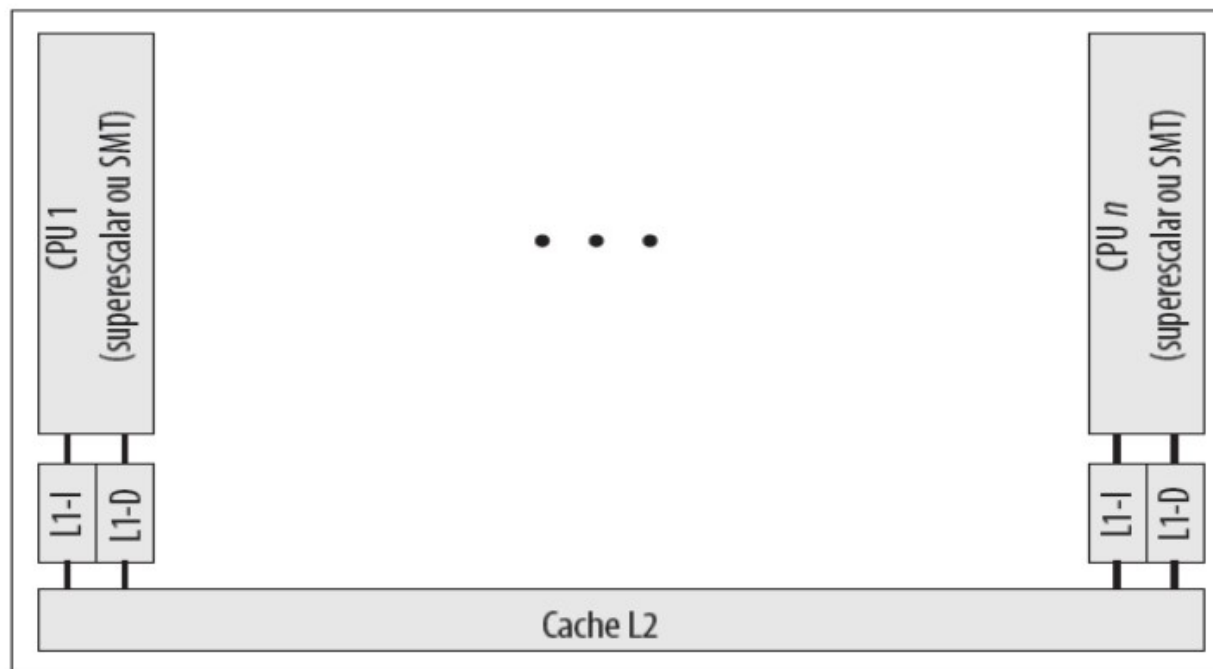
Multicore: cada núcleo (core) consiste de todos dos componentes de um processador independente

- Registradores
- Unidade Lógico e Aritmética - ULA
- Hardware de pipeline
- Unidade de Controle - UC
- Caches L1 de dados e instruções



(c) Multicore

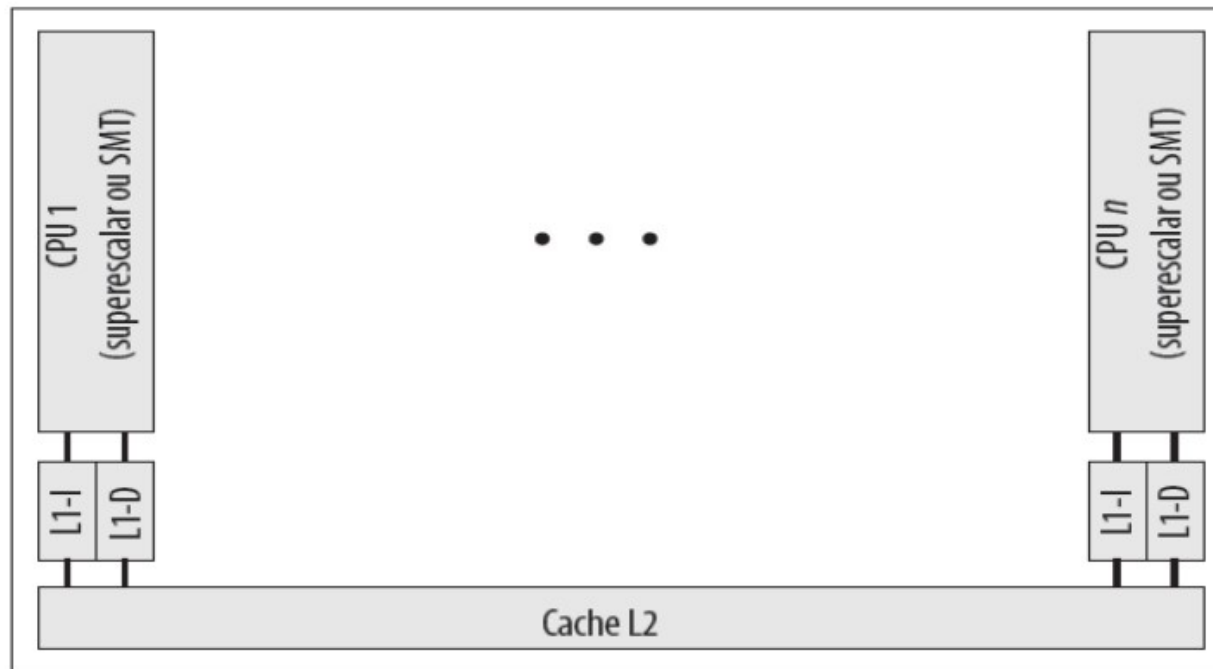
Organizações alternativas do chip



(c) Multicore

Organizações alternativas do chip

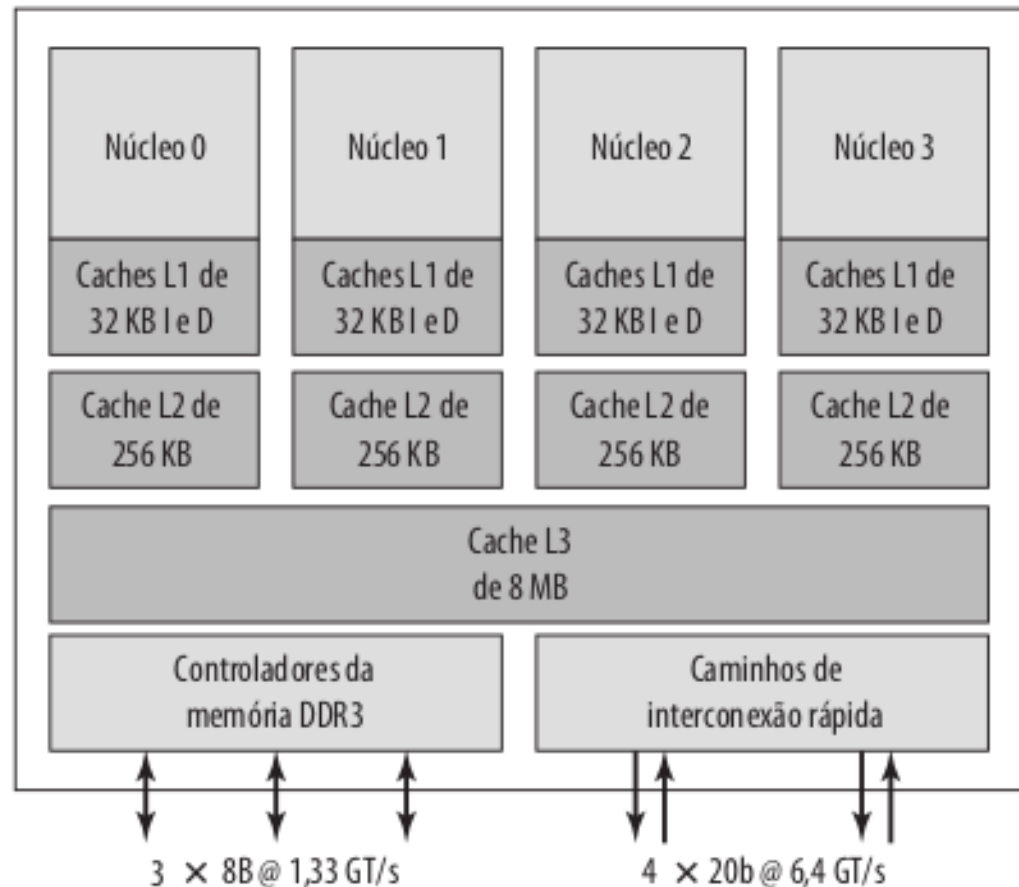
- Além de vários núcleos os chips atuais incluem também cache L2 e, em alguns casos, cache L3, em uma única pastilha de silício



(c) Multicore

Organizações alternativas do chip

Figura 18.10 Diagrama de blocos do Intel Core i7



Complexidade aumentada

- Há um conjunto de problemas relacionados ao projeto e fabricação de chips de computadores
- **O aumento na complexidade para lidar com todas as questões de lógica relacionas com:**
 - Pipelines muito longos
 - Vários pipelines superescalares
 - Vários bancos de registradores SMT
- Consequência:
 - **Grande parte do chip é ocupada com lógica de coordenação e transferência de sinais**
 - Isso aumenta a dificuldade de projeto, fabricação e depuração de chips

Utilização de transistores do chip

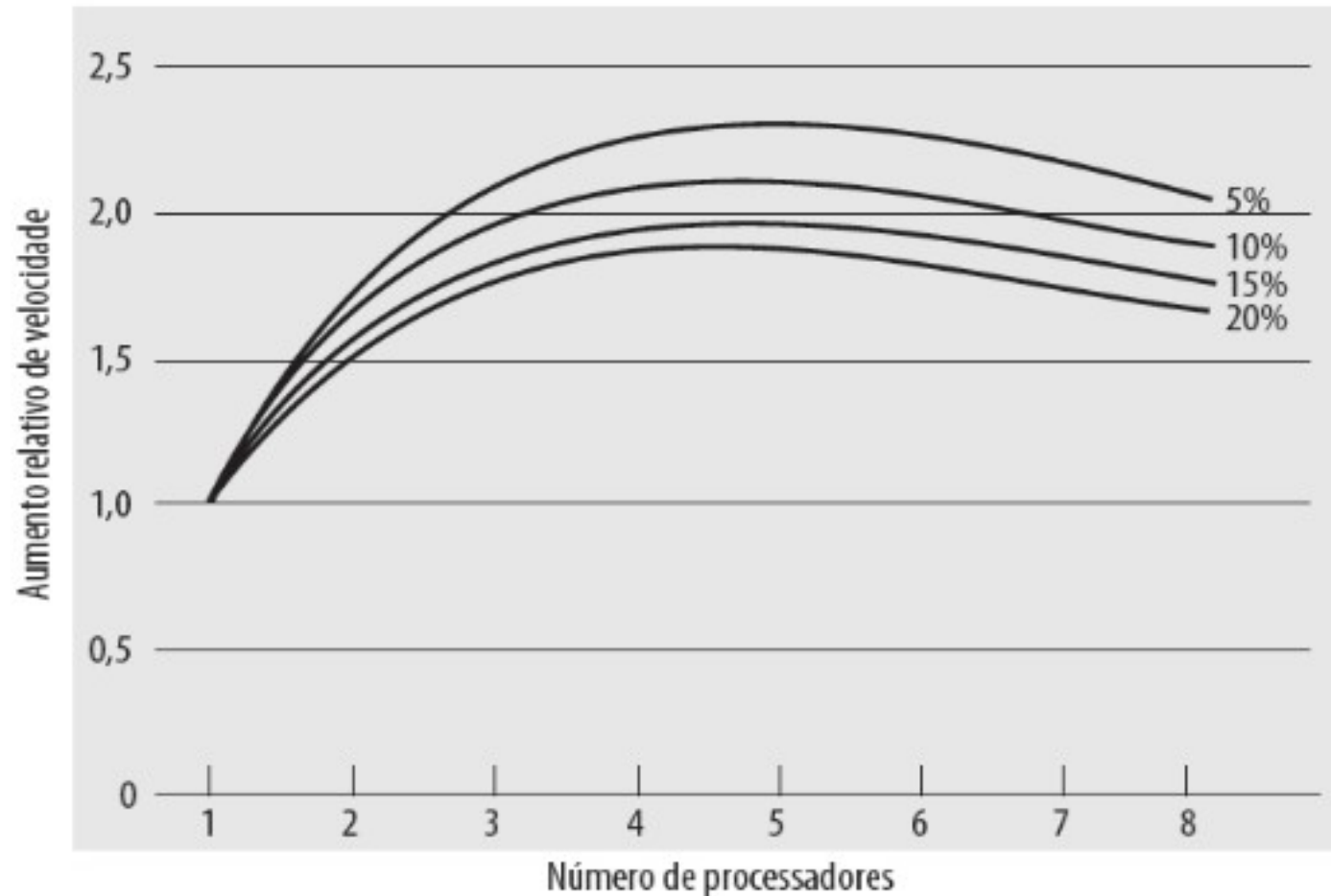
Processor	Transistor count	Date of introduction	Designer	Process	Area
Xbox One X (Project Scorpio) main SoC	7,000,000,000 ^[85]	2017	Microsoft/AMD	16 nm	360 mm ² ^[85]
IBM z13 Storage Controller	7,100,000,000	2015	IBM	22 nm	678 mm ²
28-core Xeon Platinum 8180	8,000,000,000 ^[86]	2017	Intel	14 nm	
22-core Xeon Broadwell-E5	7,200,000,000 ^[87]	2016	Intel	14 nm	456 mm ²
POWER9	8,000,000,000	2017	IBM	14 nm	695 mm ²
72-core Xeon Phi	8,000,000,000	2016	Intel	14 nm	683 mm ²
IBM z14 Storage Controller	9,700,000,000	2017	IBM	14 nm	696 mm ²
Freedom U500 Base Platform Chip (E51, 4xU54) RISC-V	250,000,000 ^[88]	2017	SiFive	28 nm	~30 mm ²
32-core SPARC M7	10,000,000,000 ^[89]	2015	Oracle	20 nm	
SPARC64 XII (12-core)	5,450,000,000 ^[90]	2017	Fujitsu	20 nm	795 mm ²
Apple A12X Bionic (octa-core ARM64 "mobile SoC")	10,000,000,000 ^[91]	2018	Apple	7 nm	122 mm ²
Apple A10X Fusion (hexa-core ARM64 "mobile SoC")	4,300,000,000 ^[92]	2017	Apple	10 nm	96.40 mm ²
Centriq 2400	18,000,000,000 ^[93]	2017	Qualcomm	10 nm	398 mm ²
32-core AMD Epyc	19,200,000,000	2017	AMD	14 nm	768 mm ²
Fujitsu A64FX	8,876,000,000 ^[94]	2018 ^[95]	Fujitsu	7 nm	
GC2 IPU	23,600,000,000	2018	Graphcore	16 nm	825 mm ²
Tegra Xavier SoC	9,000,000,000 ^[96]	2018	Nvidia	12 nm	350 mm ²

http://en.wikipedia.org/wiki/Transistor_count

Questões sobre desempenho de software

- Software normalmente provoca sobrecarga como:
 - resultado de comunicação e a distribuição de trabalho para vários processadores
 - Sobrecarga de coerência de cache
 - Coerência de cache será explicado mais adiante
- O desempenho alcança picos e depois começa a degradar por causa do aumento da sobrecarga de uso de vários processadores

Questões sobre desempenho de software



(b) Aumento de velocidade com sobrecargas

Dificuldade de criar programa paralelo em multiprocessador

Analogia:

- Oito reporteres tentando escrever um único artigo na esperança de realizar o trabalho oito vezes mais rápido
 - Para ter sucesso, a tarefa precisa ser dividida em oito partes de mesmo tamanho, pois senão alguns repórteres estariam ociosos enquanto esperam aqueles com partes maiores terminem
 - Outro perigo do desempenho seria que os repórteres gastariam muito tempo comunicando entre si em vez de escrever suas partes do artigo
 - Para essa analogia e para a programação paralela, os desafios incluem **escalonamento, balanceamento de carga, tempo para sincronização e overhead para a comunicação entre as partes**

Dificuldade de criar programa paralelo em multiprocessador

Analogia:

- Oito reporteres tentando escrever um único artigo na esperança de realizar o trabalho oito vezes mais rápido
 - Para ter sucesso, a tarefa precisa ser dividida em oito partes de mesmo tamanho, pois senão alguns repórteres estariam ociosos enquanto esperam aqueles com partes maiores terminem
 - **escalonamento e balanceamento**
 - Outro perigo do desempenho seria que os repórteres gastariam muito tempo comunicando entre si em vez de escrever suas partes do artigo
 - **tempo para sincronização e overhead para a comunicação entre as partes**

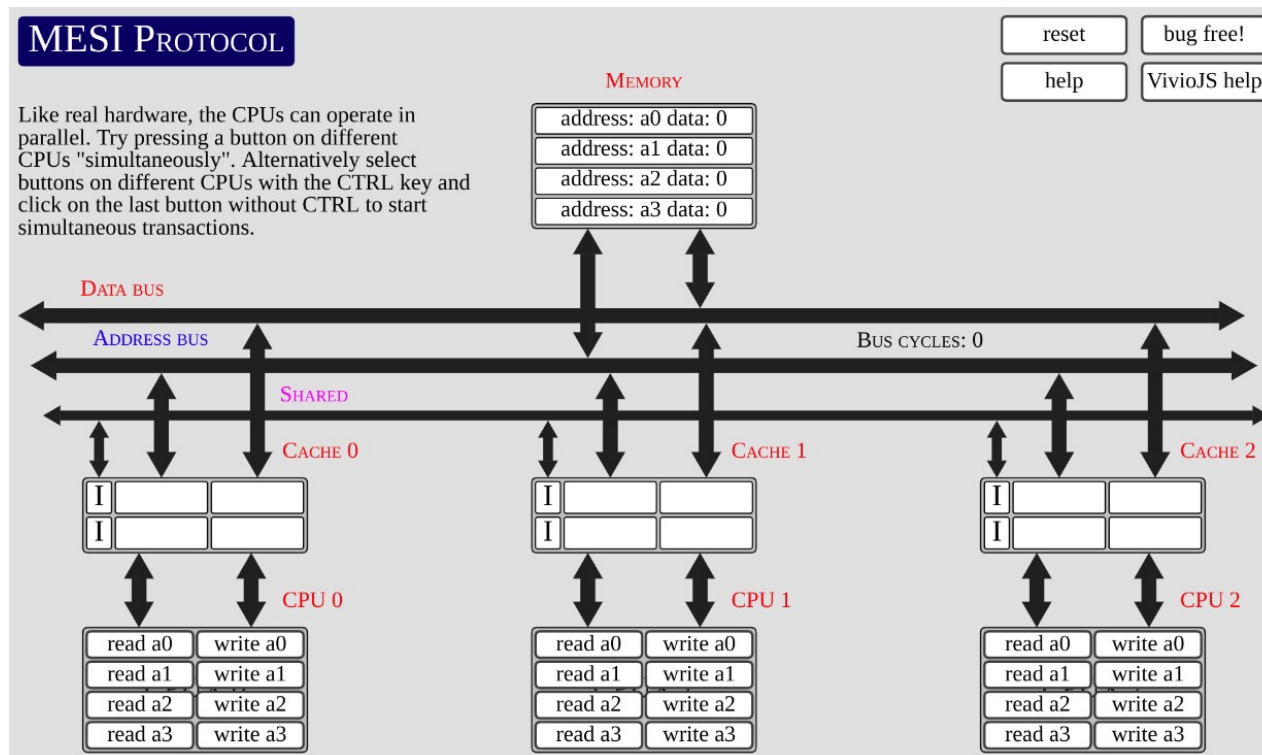
Coerência de cache

- **Coerência de cache e protocolo MESI**
 - Protocolo **MESI**, do inglês **M**odified, **E**xclusive, **S**hared, **I**nvalid
 - Nos atuais **sistemas multiprocessadores**, é comum haver um ou dois níveis de cache associados a cada processador
 - Esta organização é essencial para alcançar um desempenho razoável, no entanto, isso cria um problema conhecido como problema de **coerência de cache**

Protocolo de coerência de cache MESI

- Esta animação do VivioJS foi projetada para ajudar você a entender o protocolo de coerência do cache MESI

<https://www.scss.tcd.ie/Jeremy.Jones/VivioJS/caches/MESIHelp.htm>



MESI PROTOCOL

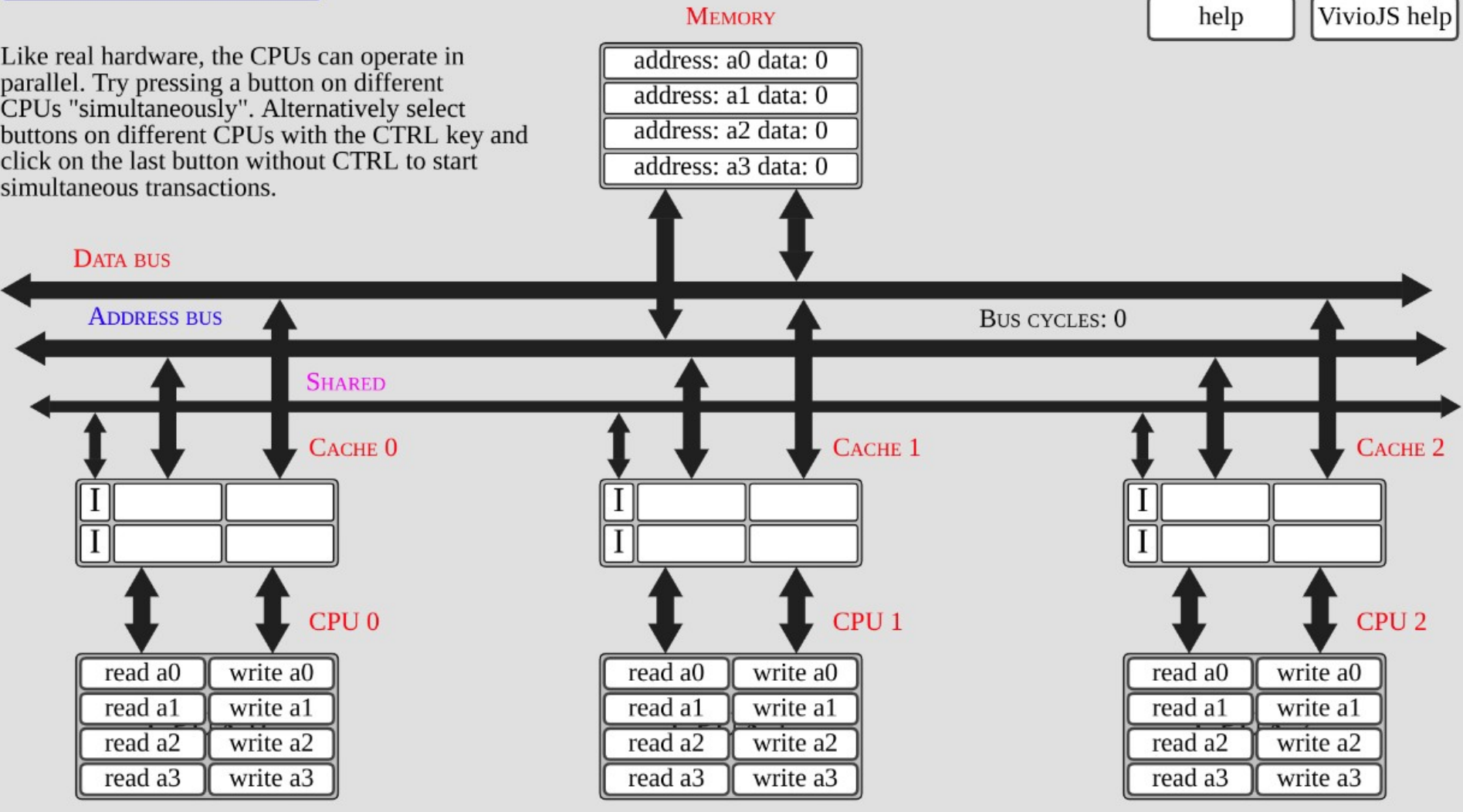
reset

bug free!

help

VivioJS help

Like real hardware, the CPUs can operate in parallel. Try pressing a button on different CPUs "simultaneously". Alternatively select buttons on different CPUs with the CTRL key and click on the last button without CTRL to start simultaneous transactions.



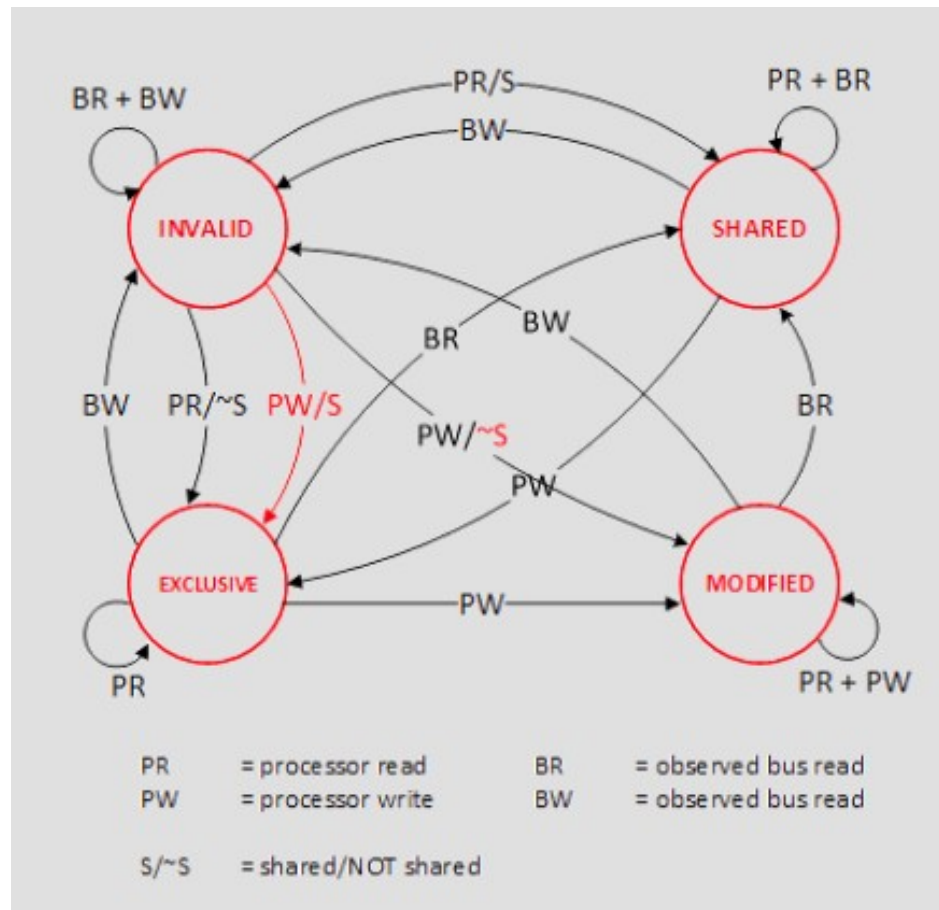
<https://www.scss.tcd.ie/Jeremy.Jones/VivioJS/caches/MESI.htm>

Protocolo de coerência de cache MESI

- Uma linha de cache pode estar em um dos quatro estados
 - **INVÁLIDO:** linha de cache não presente no cache
 - **EXCLUSIVO:** linha de cache presente somente neste cache e idêntica à cópia na memória
 - **MODIFICADO:** linha de cache presente somente neste cache e cópia da memória desatualizada (obsoleta)
 - **COMPARTILHADO:** linha de cache nesse cache e, possivelmente, outros caches, todas as cópias são idênticas às da memória

Protocolo de coerência de cache MESI

- Aqui está o diagrama de transição de estado para uma linha de cache:



Protocolo de coerência de cache MESI

- Sequência de amostra para tentar

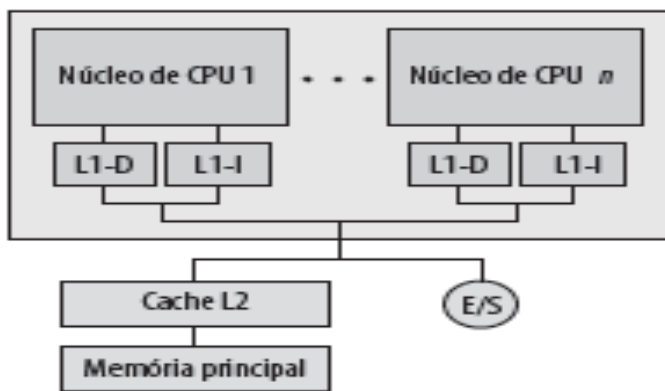
1	CPU0: leia a0	<i>CPU0 lê a0 da memória [não compartilhada] - estado E</i>
2	CPU0: leia a0	<i>CPU0 lê a0 do cache - estado E</i>
3	CPU0: escrever a0	<i>CPU0 atualiza a0 APENAS no cache - estado M</i>
4	CPU0: escrever a0	<i>CPU0 atualiza a0 APENAS no cache - estado M</i>
5	CPU1: leia a0	<i>CPU1 lê a0, o cache da CPU0 intervém e fornece dados ao cache e à memória - estado S</i>
6	CPU1: escrever a0	<i>A CPU1 atualiza a0 no cache e na memória e invalida todos os outros caches com o endereço a0 - estado E</i>
7	CPU1: escrever a0	<i>CPU1 atualiza apenas a0 no cache - estado M</i>
8	CPU0: escrever a0	<i>CPU0 lê a0, o cache da CPU1 intervém e fornece dados ao cache e à memória (S); a CPU0 grava em a0 no cache e na memória, invalidando todos os outros caches com o endereço a0 - state</i>
9	CPU0: escrever a2	<i>CPU0 lê a2 da memória (E) e depois grava no a2 - estado M</i>
10	CPU0: escrever a0	<i>CPU0 libera a2 na memória, lê a2 na memória (E) e depois grava no a0 - estado M</i>

Organização multicore

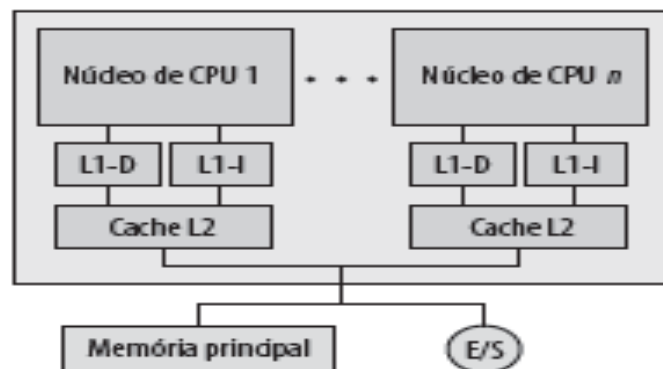
- No nível mais alto da descrição, as principais variáveis em uma organização multicore são as seguintes:
 - Número de núcleos processadores no chip
 - Número de níveis de cache no chip
 - Quantidade de cache compartilhada
- Próximo *slide* com exemplos de cada organização:
 - (a) ARM11 MPCore
 - O MPCore ARM11 - HTC, iPhone 3G, iPod Touch, Kindle, 3DS Nintendo e o tablet Nokia N800
 - (b) AMD Opteron
 - (c) Intel Core Duo
 - (d) Intel Core i7

Alternativas da organização multicore

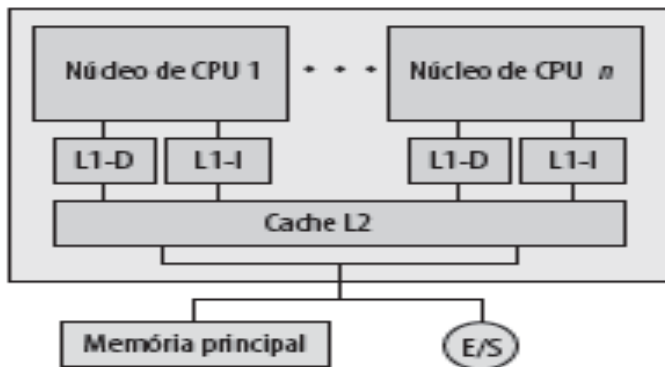
- (a) ARM11 MPCore
- (b) AMD Opteron
- (c) Intel Core Duo
- (d) Intel Core i7



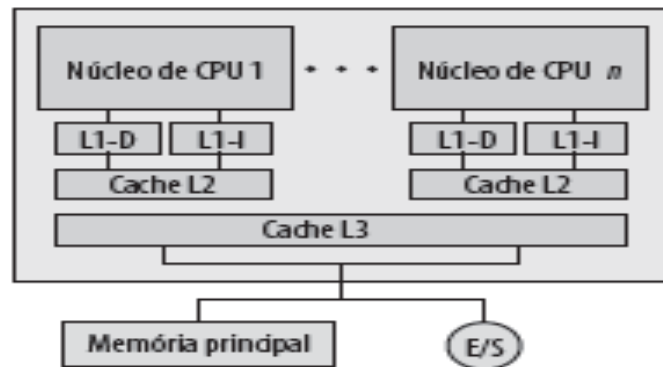
(a) Cache L1 dedicada



(b) Cache L2 dedicada



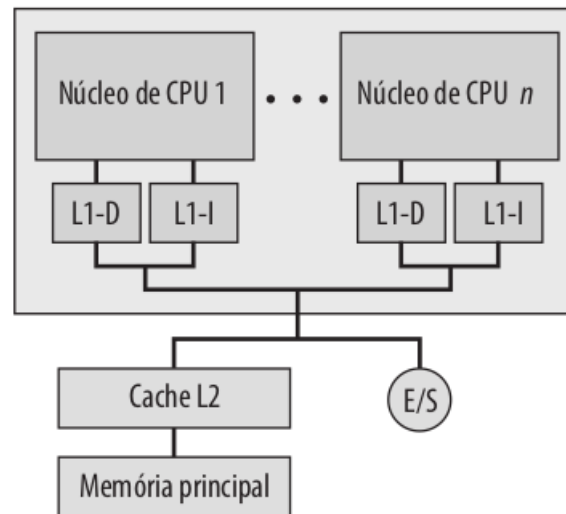
(c) Cache L2 compartilhada



(d) Cache L3 compartilhada

Alternativas da organização multicore

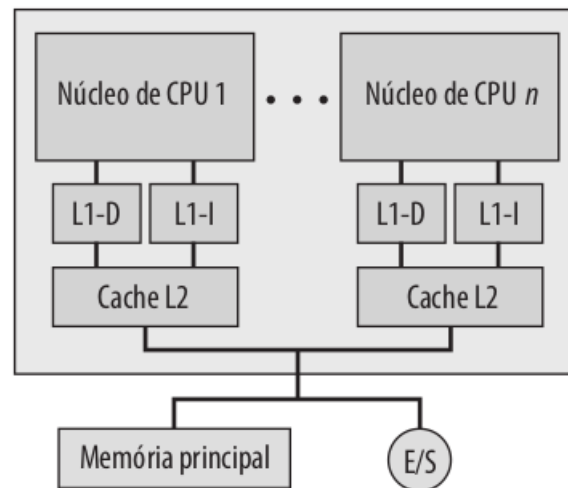
- A única cache no chip é L1, com cada núcleo tendo a sua cache L1 dedicada
- Quase invariavelmente, a cache L1 é dividida em caches de dados e instruções
 - Um exemplo desta organização é ARM11 MPCore



(a) Cache L1 dedicada

Alternativas da organização multicore

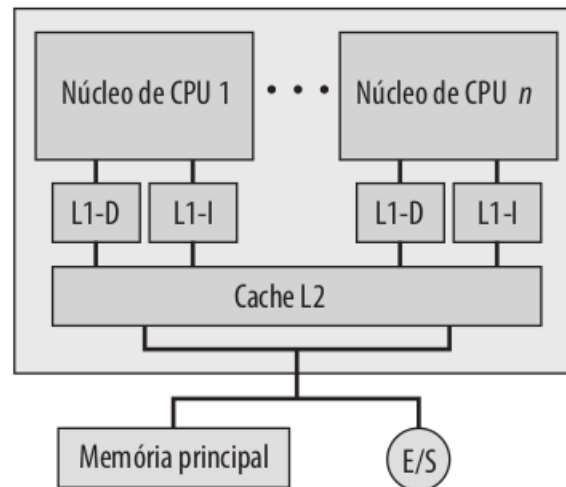
- Não há compartilhamento da cache no chip
- Há bastante área disponível no chip para permitir a cache L2
 - Um exemplo desta organização é o AMD Opteron



(b) Cache L2 dedicada

Alternativas da organização multicore

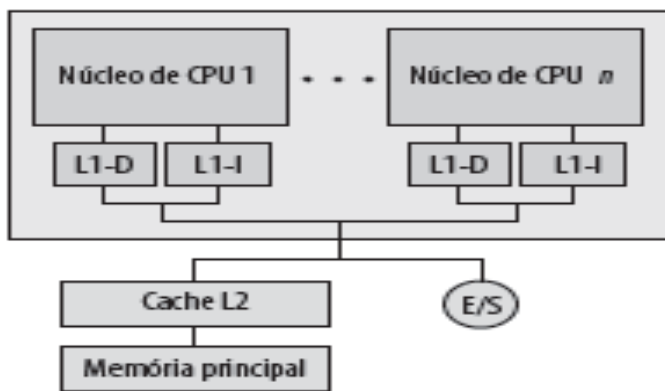
- Uma alocação semelhante de espaço do chip para memória, porém com **uso de cache L2 compartilhada**
 - O Core Duo da Intel tem essa organização



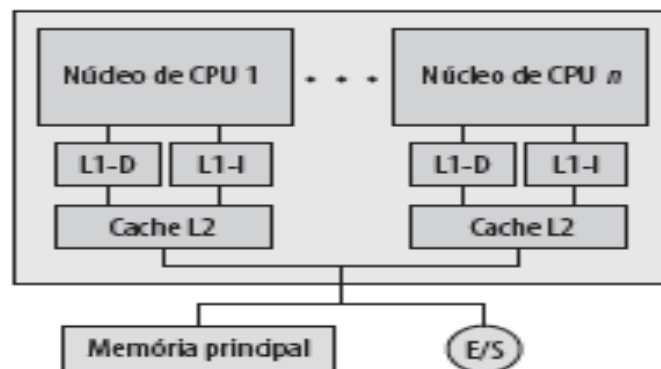
(c) Cache L2 compartilhada

Alternativas da organização multicore

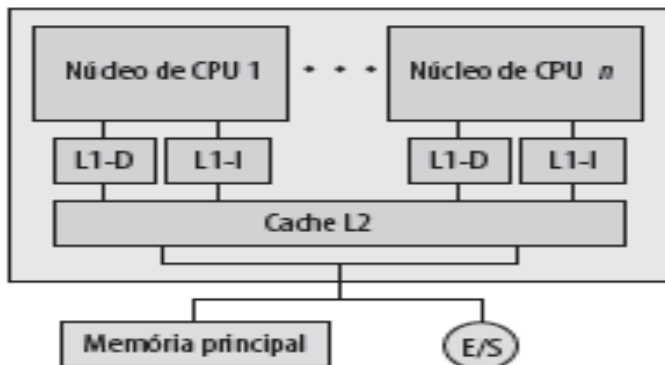
- (a) ARM11 MPCore
- (b) AMD Opteron
- (c) Intel Core Duo
- (d) Intel Core i7



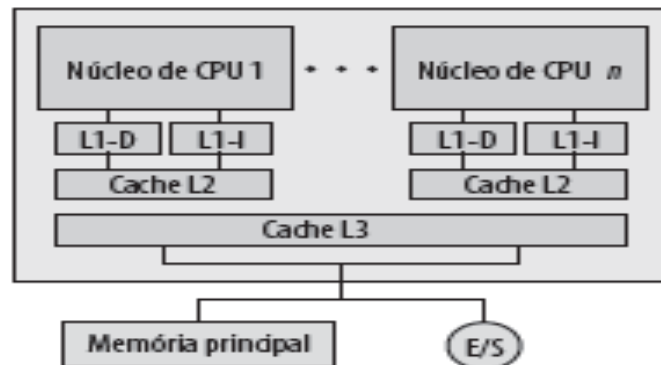
(a) Cache L1 dedicada



(b) Cache L2 dedicada



(c) Cache L2 compartilhada



(d) Cache L3 compartilhada

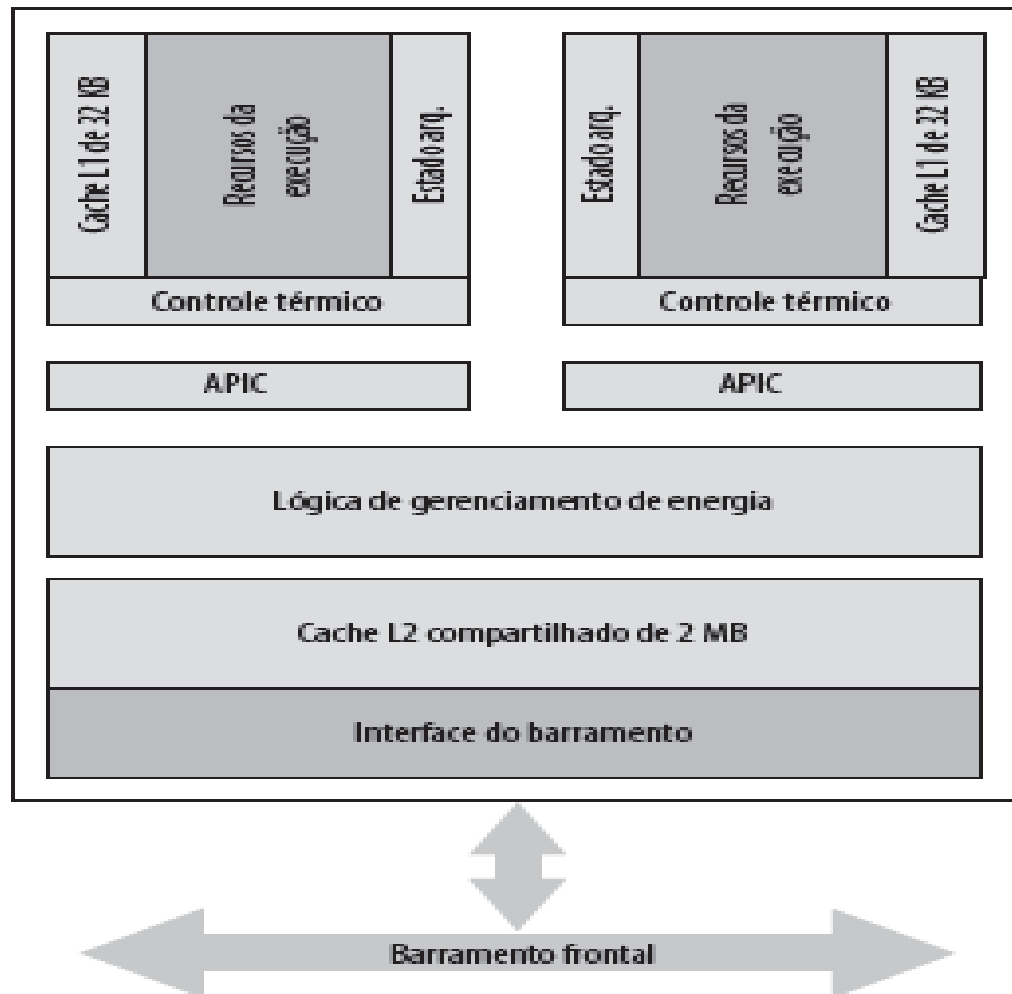
Arquitetura de núcleo individual

- Intel Core Duo usa núcleos superescalares
- Intel Core i7 usa multithreading simultâneo (SMT)
 - Aumenta número de threads suportadas
 - 4 núcleos SMT, cada um suportando 4 threads, aparece como um multicore de 16 núcleos.

Organização multicore x86 da Intel – Core Duo

- O Intel Core Duo, introduzido em 2006
- Dois x86 superscalares, cache L2 compartilhada
- Cache L1 dedicado por núcleo
 - Instrução de 32 KB e dados de 32KB
- Unidade de controle térmico por núcleo:
 - Controla dissipação de calor no chip
 - Maximiza desempenho dentro das restrições
 - Melhora a ergonomia (menor barulho do ventilador)
 - **Importante para Notebooks e sistemas móveis**

Diagrama de blocos do Intel Core Duo



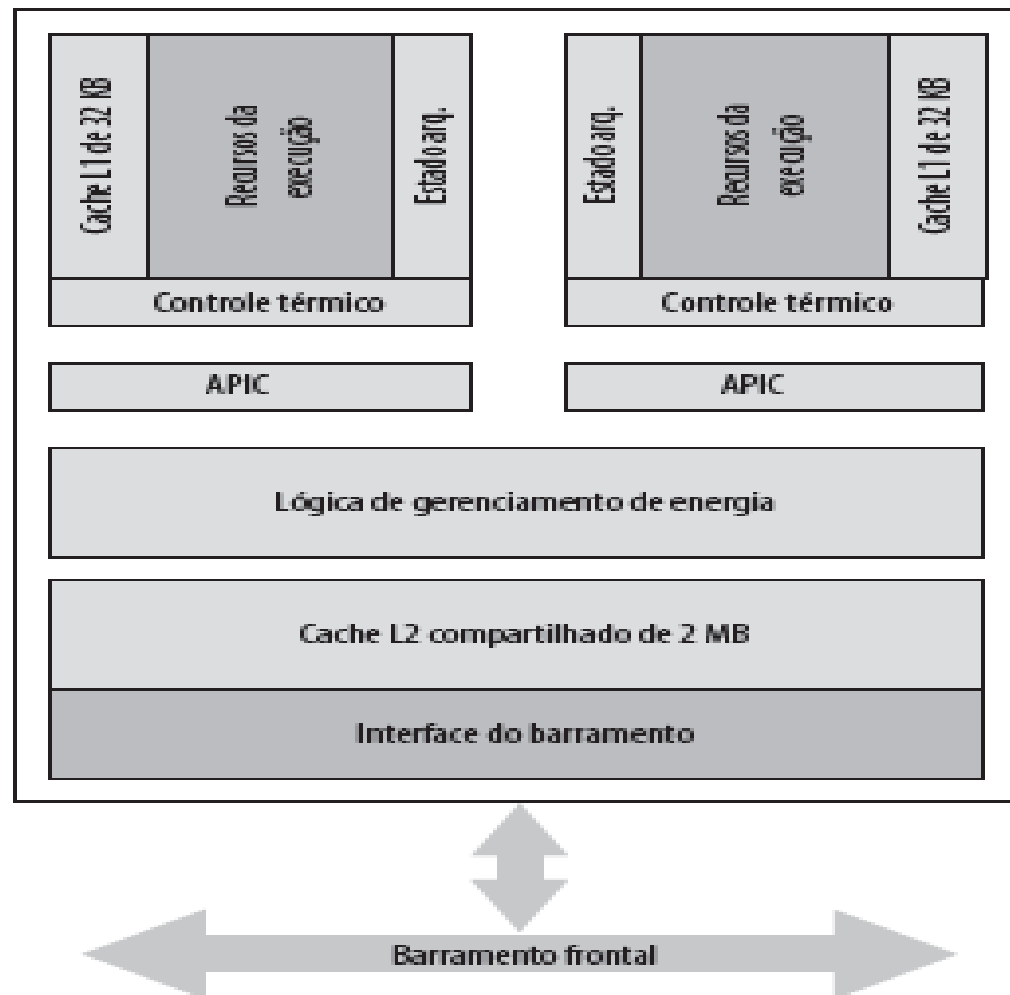
Organização multicore x86 da Intel – Core Duo

- **Lógica de gerenciamento de energia:**
 - Monitora condições térmicas e atividade da CPU
 - Ajusta voltagem e consumo de energia
 - Pode chavear subsistemas lógicos individuais
 - **Objetivo: reduzir consumo sempre que possível**

Organização multicore x86 da Intel – Core Duo

- **Cache L2 de 2 MB compartilhada:**
 - Alocação dinâmica:
 - Aloca espaço da cache com base nas necessidades atuais do núcleo.
 - A um núcleo pode ser atribuído até 100 por cento da cache L2.
 - Suporte **MESI** para caches L1 anexadas:
 - Uma linha de cache obtém o estado M quando um processador escreve nela
 - Se a linha não esteve no estado E ou M antes de escrever nela, a cache envia uma requisição Leitura-Para-Posse que garante que a linha existe na cache L1 e está no estado I na outra cache L1

Diagrama de blocos do Intel Core Duo



Organização multicore x86 da Intel – Core Duo

- **Cache L2 de 2 MB compartilhada:**
 - Estendido para dar suporte a Core Duo múltiplo em SMP (Multiprocessador Simétrico)
 - Dados L2 compartilhados entre núcleos locais ou externos

Organização Multiprocessador Simétrico

Figura 17.5 Organização de um multiprocessador simétrico

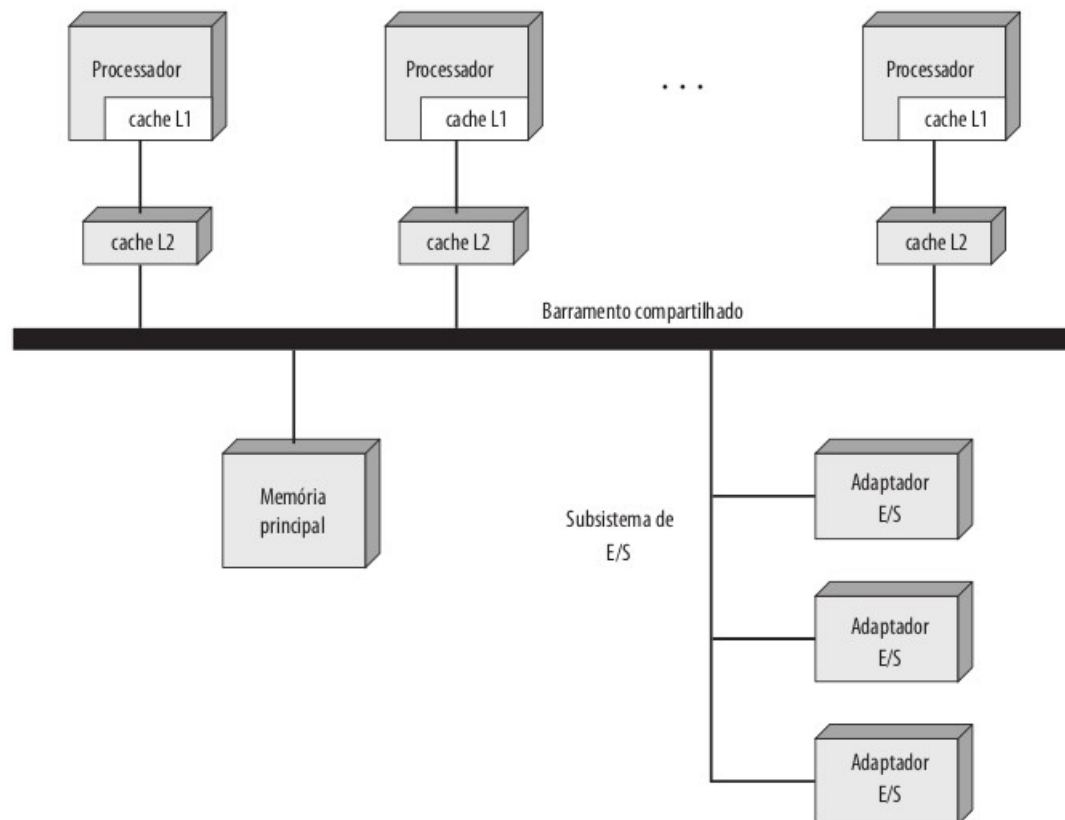
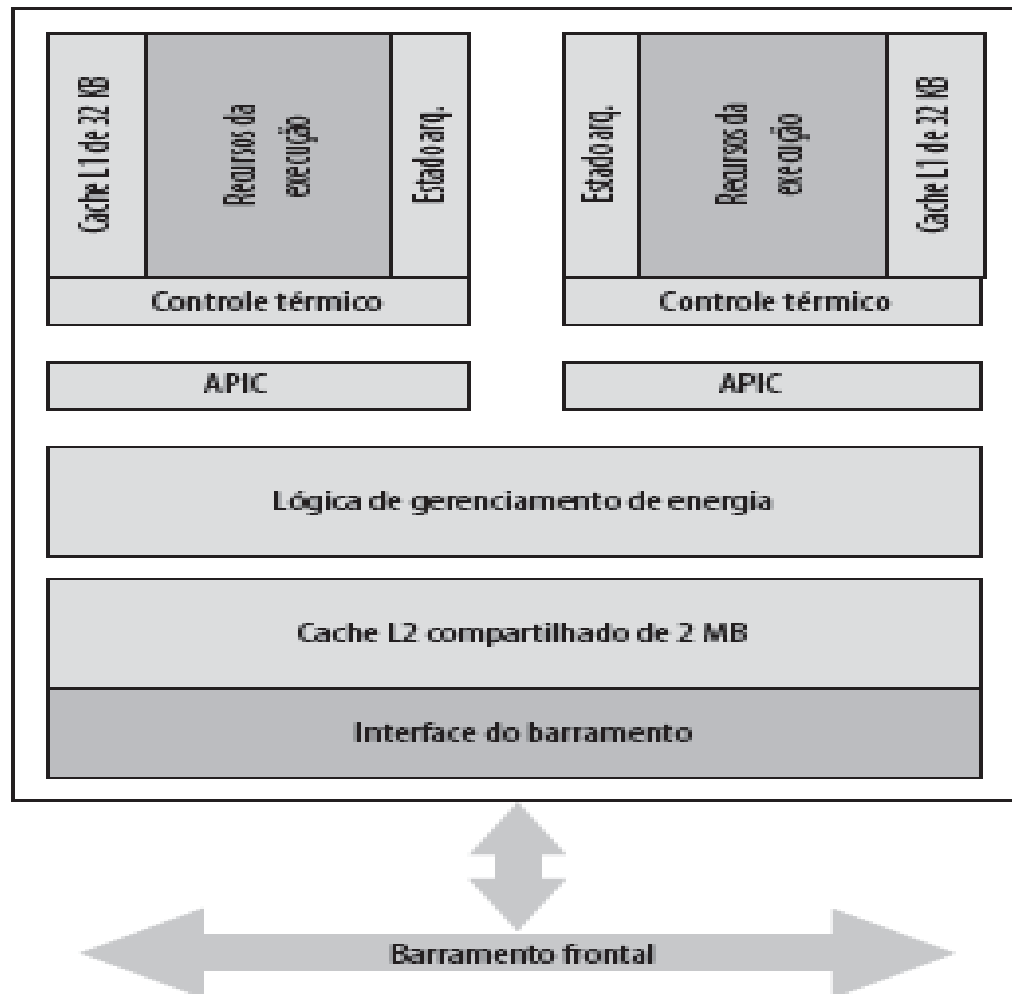


Diagrama de blocos do Intel Core Duo



Organização multicore x86 da Intel – Core i7

- Novembro de 2008
- Quatro processadores SMT x86
- Cache L2 dedicada, L3 compartilhada
- Pré-busca especulativa para caches
 - O hardware analisa padrões de acesso e tenta preencher as caches de forma especulativa com dados que provavelmente serão requisitados logo
 - É interessante comparar o desempenho desta organização. O Core 2 Quad tem uma cache L2 compartilhada, semelhante ao Core Duo

Tabela 18.1 Latência de cache (em ciclos de clock)

CPU	Frequência de clock	Cache L1	Cache L2	Cache L3
Core 2 Quad	2,66 GHz	3 ciclos	15 ciclos	—
Core i7	2,66 GHz	4 ciclos	11 ciclos	39 ciclos

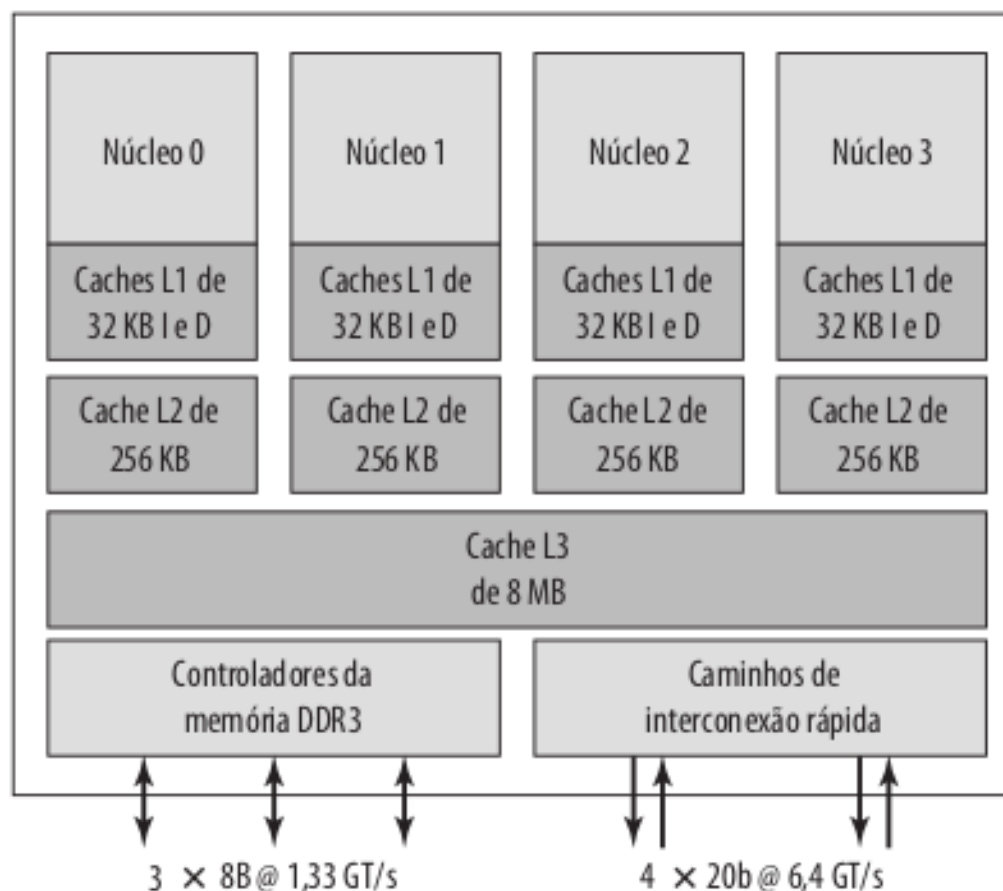
Organização multicore x86 da Intel – Core i7

- Controlador de memória DDR3 no chip:
 - Três canais de 8 bytes (192 bits) gerando 32 GB/s
 - Sem barramento frontal
- Caminho de interconexão rápida(QPI – QuickPath Interconnect):
 - Interconexão ponto a ponto coerente com cache (MESI)
 - Comunicações de alta velocidade entre chips processadores
 - Transferências de 6,4 G por segundo, 16 bits por transferência
 - Pares bidirecionais dedicados.
 - Largura de banda total de 25,6 GBps

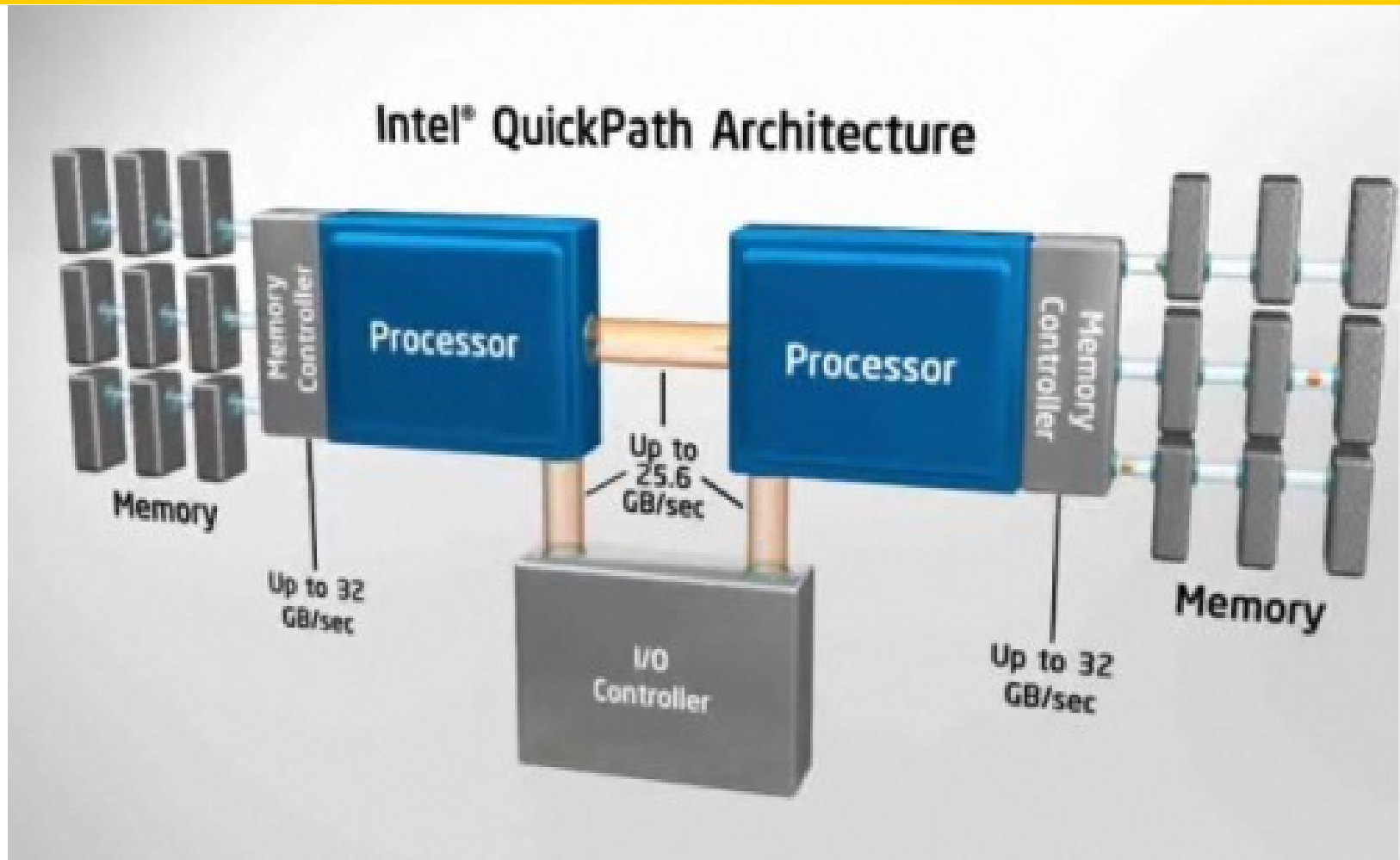
<https://www.intel.ca/content/dam/doc/white-paper/quick-path-interconnect-introduction-paper.pdf>

Diagrama em blocos do Intel Core i7

Figura 18.10 Diagrama de blocos do Intel Core i7



Arquitetura QuickPath



<https://www.intel.ca/content/dam/doc/white-paper/quick-path-interconnect-introduction-paper.pdf>

Leitura recomendada

Capítulo 18

William Stallings

Arquitetura e Organização de Computadores

8ª Edição