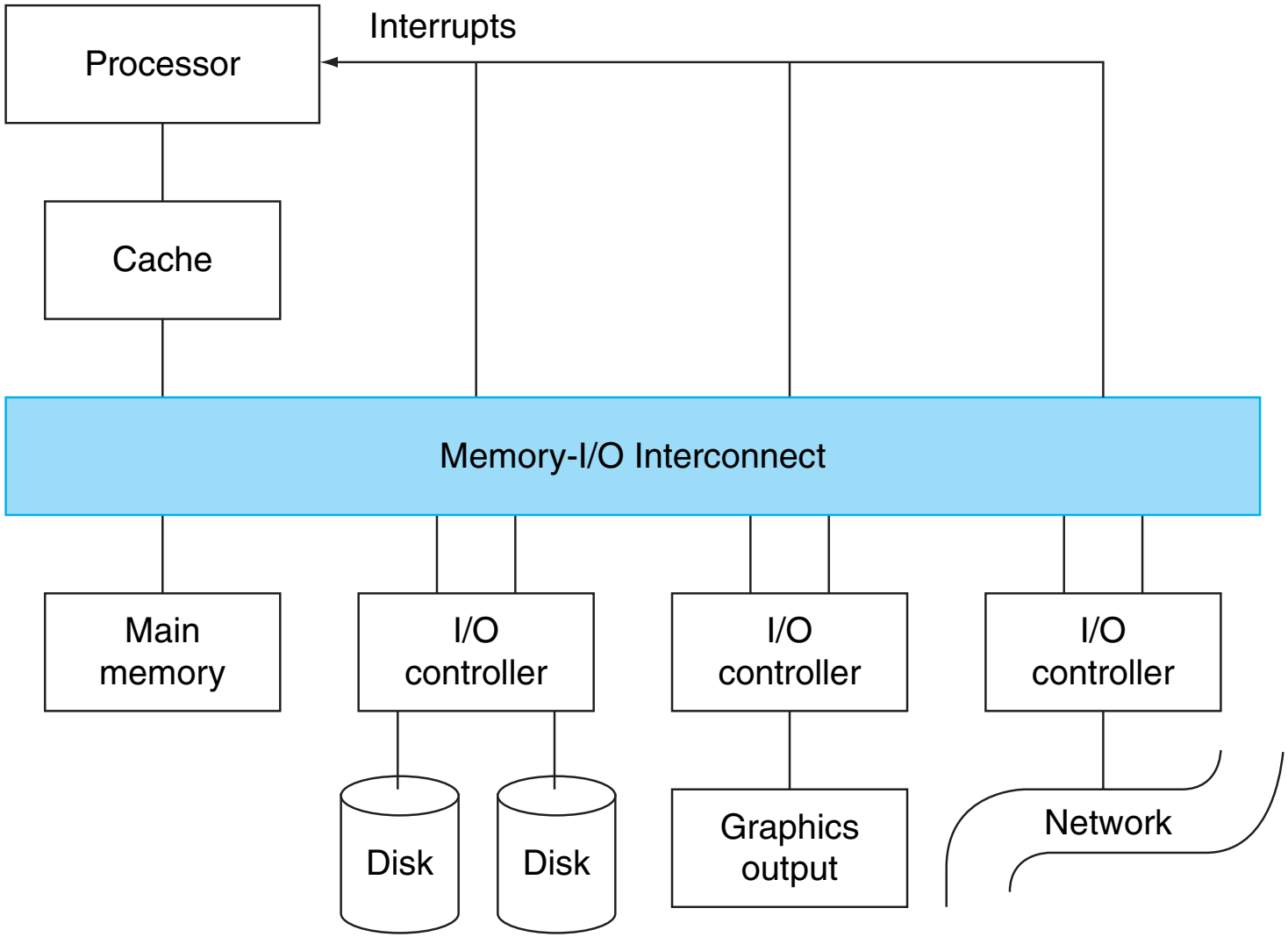
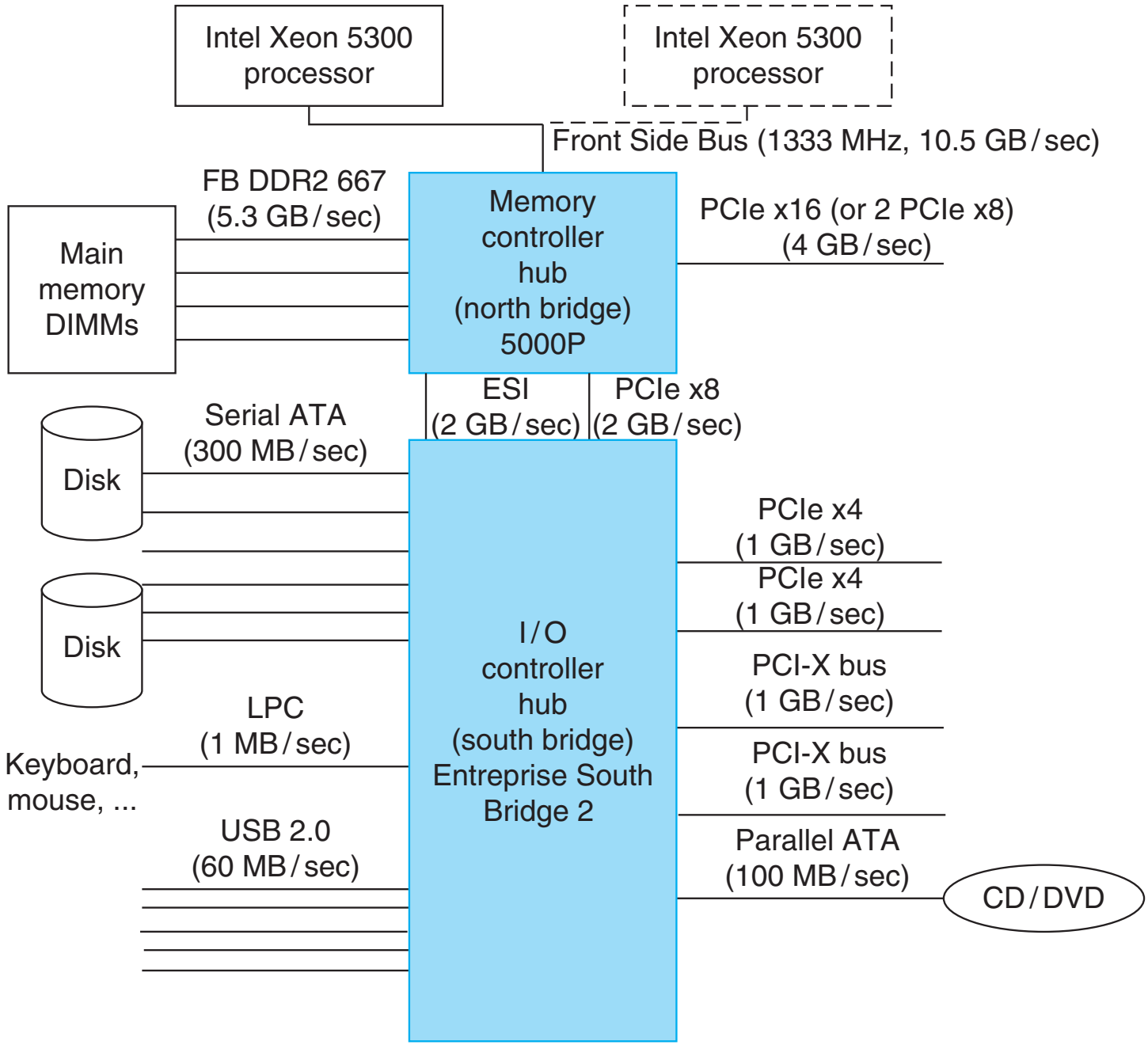
Input / Output

Sistema com dispositivos de I/O

Ligação por *bus*



Hubs de controlo de memória e de I/O



Interacção entre o processador e os dispositivos de I/O

Acções iniciadas pelo processador

1. Processador envia pedido para o dispositivo apropriado

2a. Espera a resposta (dispositivos rápidos, como a memória),

ou

2b. Continua com outras tarefas (e.g., execução de um programa)

No último caso, o processador pode obter a resposta ao pedido de dois modos

Polling (ou auscultação)

Periodicamente, o processador interroga o dispositivo sobre o estado do pedido

Interrupt-driven

O dispositivo gera uma interrupção quando estiver pronto para atender o pedido do processador

Interacção entre o processador e os dispositivos de I/O

Acções iniciadas pelos dispositivos

Por vezes, os dispositivos de I/O necessitam de comunicar com o processador

> Acontece com os dispositivos de *input*, quando há *input* disponível para ser processado (e.g., quando é premida uma tecla do teclado)

Tal como no caso das respostas aos pedidos do processador, a interacção pode efectuar-se de duas formas

Polling

Periodicamente, o processador pergunta a cada dispositivo se tem algo a comunicar

Em geral, o processador roda por todos os dispositivos (estratégia *round-robin*)

Interrupt-driven

Quando o dispositivo pretende comunicar com o processador, gera uma interrupção, que o processador tratará assim que puder

Comunicação entre o processador e os dispositivos de I/O

A comunicação entre o processador e um dispositivo de I/O pode ocorrer *aparentemente* através da memória

Memory-mapped I/O

É estabelecida a correspondência entre alguns endereços de memória e um dispositivo de I/O

Os acessos a esses endereços são interpretados como o envio de comandos para o dispositivo

Em alternativa ou em simultâneo, a arquitectura pode incluir instruções para acesso aos dispositivos de I/O

Direct memory access (DMA)

Transferência de dados de e para memória

O processador indica ao controlador de DMA

- O dispositivo a contactar
- A operação a realizar (leitura ou escrita)
- O número de *bytes* a transferir
- O endereço onde colocar, ou onde se encontram, os dados

O controlador de DMA transfere dados entre a memória e os dispositivos de I/O sem intervenção posterior do processador

Quando a transferência termina, o controlador gera uma interrupção para notificar o processador

Acesso a um disco magnético

Características

Velocidade de rotação Velocidade a que o disco gira

Mede-se em rotações por minuto (rpm)

***Seek time* (tempo de colocação)** Tempo que demora a colocar a cabeça de leitura/escrita na pista a aceder

Latência rotacional Tempo que é necessário esperar, em média, até que o sector pretendido esteja sob a cabeça de leitura/escrita

Depende da velocidade de rotação

Corresponde ao tempo de meia rotação

Taxa de transferência Velocidade a que é possível ler (ou escrever) informação do (ou no) disco

Mede-se em MB/s*

Controlador Circuito que controla o funcionamento do disco

Introduz algum *overhead* nos acessos

* $1\,\text{MB} = 10^6$ *bytes*

Acesso a um disco magnético

Tempo de acesso

$$\text{Tempo de acesso} = \textit{Seek time} + \text{Latência rotacional} + \frac{\text{Tempo de}}{\text{transferência}} + \frac{\textit{Overhead}\text{ do}}{\text{controlador}}$$

$$\text{Latência rotacional} = \frac{1}{2} \times \frac{60}{\text{Velocidade de rotação}}$$

Acesso a um disco magnético (1)

Exemplo

Exemplo

Qual o tempo necessário para ler um bloco de 512 *bytes* de um disco com as seguintes características:

- *Seek time* médio = 4,0 ms
- Velocidade de rotação = 15 000 rpm
- Taxa de transferência = 100 MB/s
- *Overhead* do controlador = 0,2 ms

$$\text{Latência rotacional} = \frac{1}{2} \times \frac{60}{15\,000} = 2{,}0\,\text{ms}$$

$$t_{\text{transferência}} = \frac{\textit{bytes}\ \text{a transferir}}{\text{taxa de transferência}} = \frac{512}{100 \times 10^6} = 0{,}00512\,\text{ms}$$

Acesso a um disco magnético (2)

Exemplo

Exemplo (cont.)

$$t_{\text{acesso}} = 4{,}0 + 2{,}0 + 0{,}00512 + 0{,}2 = 6{,}20512 \approx 6{,}2\,\text{ms}$$

Se o período do relógio do processador for de 0,5 ns, quantos ciclos de relógio leva a leitura?

$$\text{ciclos}_{\text{acesso}} = \frac{t_{\text{acesso}}}{T} \approx \frac{6{,}2 \times 10^{-3}}{0{,}5 \times 10^{-9}} = 12{,}4 \times 10^{6}$$

A leitura demora cerca de 12,4 milhões de ciclos de relógio

Multiprocessamento

Paralelismo *vs* concorrência

		Software	
		Sequential	**Concurrent**
Hardware	Serial	Matrix Multiply written in MatLab running on an Intel Pentium 4	Windows Vista Operating System running on an Intel Pentium 4
	Parallel	Matrix Multiply written in MATLAB running on an Intel Core i7	Windows Vista Operating System running on an Intel Core i7

Serial é traduzido para sequencial ou, menos frequentemente, série

Temos processamento paralelo, ou um programa paralelo, quando o programa utiliza múltiplos processadores em simultâneo

O desafio do paralelismo

A situação ideal

$$\text{Tempo depois da paralelização} = \frac{\text{Tempo antes da paralelização}}{\text{Número de processadores}}$$

$$\begin{aligned} \textit{Speedup} &= \frac{\text{Tempo antes da paralelização}}{\text{Tempo depois da paralelização}} \\ &= \text{Número de processadores} \end{aligned}$$

O desafio do paralelismo

Lei de Amdahl (no contexto da paralelização)

$$\text{Tempo depois da paralelização} = \frac{\text{Tempo antes da paralelização} - \text{Tempo não afectado pela paralelização}}{\text{Número de processadores}} + \text{Tempo não afectado pela paralelização}$$

$$\textit{Speedup} = \frac{\text{Tempo antes}}{\frac{\text{Tempo antes} - \text{Tempo não afectado}}{\text{Número de processadores}} + \text{Tempo não afectado}}$$

$$= \frac{1}{\frac{1 - \%\text{ tempo não afectado}}{\text{Número de processadores}} + \%\text{ tempo não afectado}}$$

Escalabilidade

Aumento do número de processadores pode não se traduzir directamente no aumento do desempenho, devido a

- Tempo da parte sequencial
- Maiores necessidades de sincronização ou de comunicação
- Distribuição do trabalho (*load balancing*) desigual entre os processadores

A escalabilidade diz respeito ao modo como evolui o *speedup* de um programa

- Escalabilidade forte (*strong scaling*)
  Evolução do *speedup* com o número de processadores
- Escalabilidade fraca (*weak scaling*)
  Evolução do *speedup* quando a dimensão do problema aumenta com o número de processadores