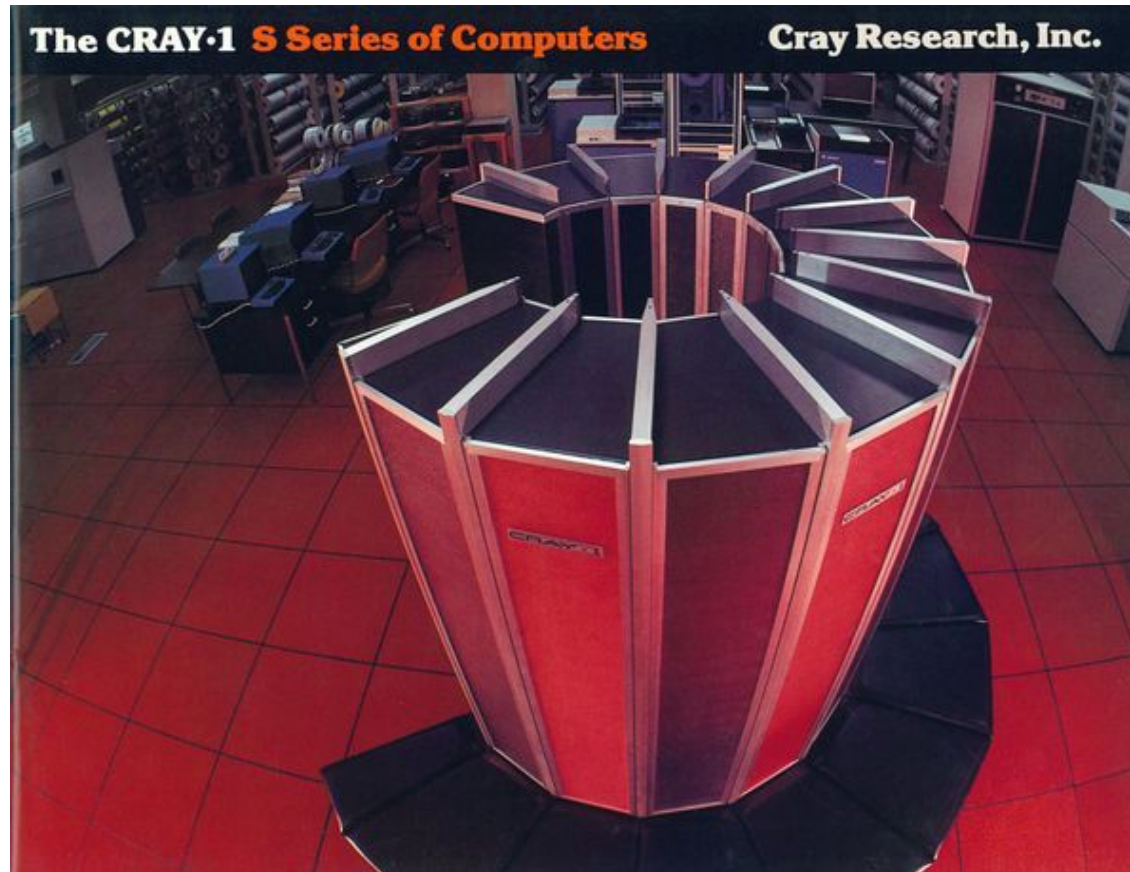


# Microprocessadores e Computadores Pessoais 2019/20

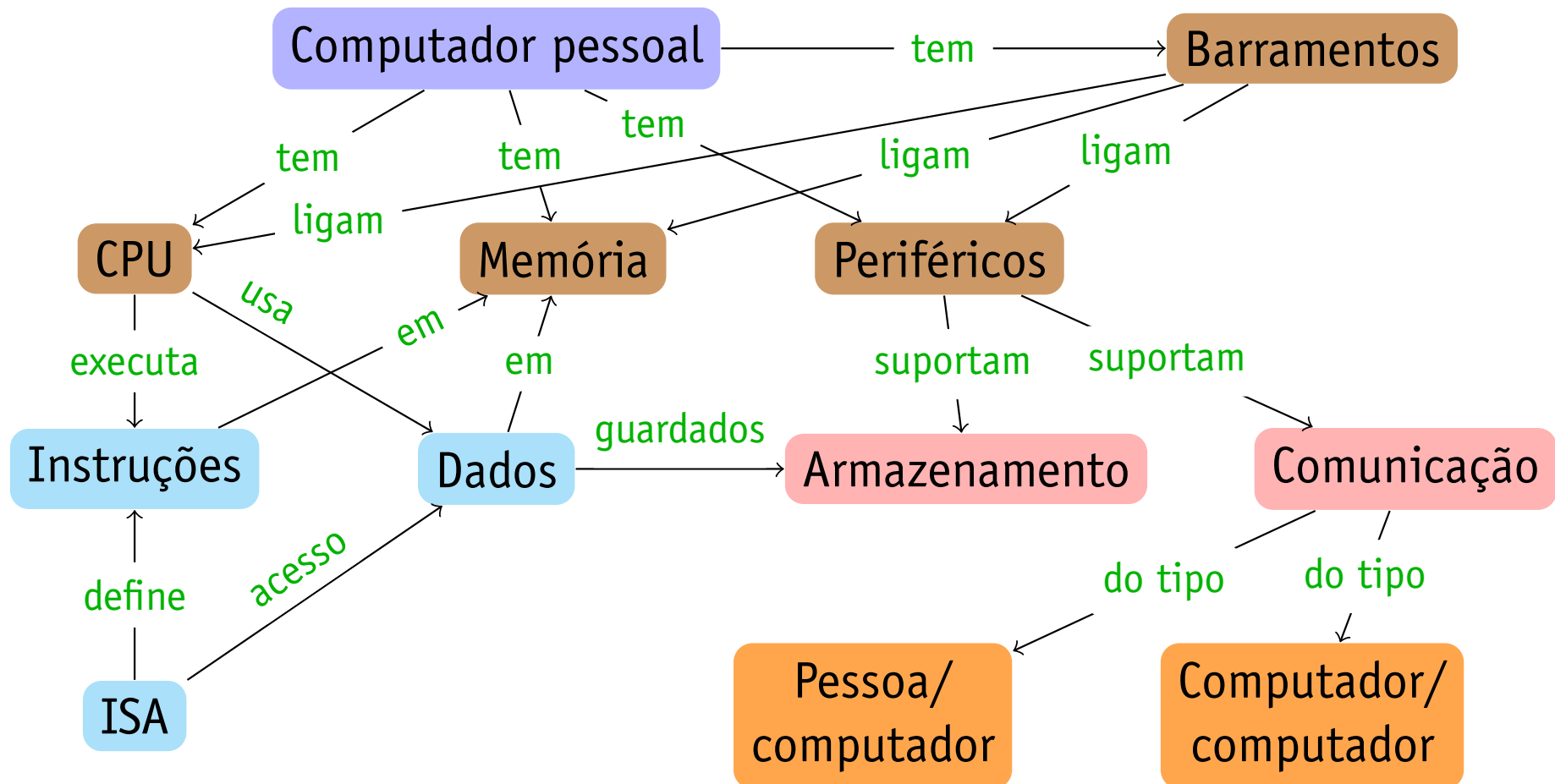


Processador de 64 bits, 160 MFLOPS,  $2^{20}$  palavras de 72 bits (64 bits de dados + 8 bits de paridade, equivalente a 8 MiB úteis), 80 MHz, 5,5 toneladas, consumo de 115 kW, US\$ 8,8 milhões em 1977 (US\$ 38,7 milhões em 2019, cerca de € 35,4 milhões).

# Assuntos

- ▶ Periféricos
  - ▶ Sistemas de armazenamento (discos magnéticos, SSD)
  - ▶ Barramentos para periféricos
  - ▶ Gestão de periféricos (exemplos)
  - ▶ Avaliação de desempenho de sistemas “completos”
- ▶ Arquitetura ARMv8 (AArch64, 64-bit)
  - ▶ Arquitetura do conjunto de instruções
- ▶ Programação em *assembly*
  - ▶ Arquitetura do conjunto de instruções
  - ▶ Programação modular: sub-rotinas e passagem de parâmetros
  - ▶ Programação mista C++/*assembly*
  - ▶ Tratamento de dados em vírgula flutuante
  - ▶ Processamento paralelo (SIMD: single-instruction multiple-data)
- ▶ Introdução aos multiprocessadores (*multi-core*)

# Mapa de conceitos



## Material de consulta

- ▶ David Patterson, John Hennessy  
Computer Organization and Design: The Hardware/Software Interface  
**ARM Edition**,  
Elsevier / Morgan Kaufmann, 2016
- ▶ Bruce Jacob; Spencer W. Ng ; David T.Wang  
Memory Systems - Cache, DRAM, Disk (portal Knovel)
- ▶ Apresentações das aulas teóricas  
(sucintas; **tirar apontamentos**)
- ▶ Resumos de instruções

## Avaliação

- ▶ Frequência: não ter mais de 3 faltas às aulas TP.  
**Apenas frequência de 2018/19 válida!**
- ▶ Dois minitestest: M1, M2 (com correção automática de programas)
- ▶ Exame: E (com correção automática de programas )

Cada miniteste: questões de escolha múltipla + questões de desenvolvimento (M1) ou exercício de programação(M2).

### Avaliação distribuída

$$NDist = \frac{M1 + M2}{2}$$

Limite mínimo:  **$NDist \geq 5,0$**

### Nota final

$$NMed = 0,5 \times NDist + 0,5 \times E$$

$$NFinal = \max(NMed, E)$$

Limites mínimos:  **$E \geq 5,0$**  e  **$NFinal \geq 10$** .

# Correção automática de programas

## ■ Procedimento:

1. Escrever e testar programa no ambiente DS-5
2. Submeter o texto do programa ao servidor de correção
3. Obter resultados de testes básicos
4. Se necessário, recomeçar a partir de 1
5. Submeter versão definitiva.

■ Em teste ou exame, a versão definitiva será posteriormente submetida a mais testes e/ou avaliada manualmente.

■ Utilização: teste M2 e Exame.

■ O sistema de correção será usado nas aulas TP e estará disponível para uso durante a parte final do semestre.