Cheatsheet: Computer Abstractions and Technology - Chương 1

Kiến trúc Máy tính - Giữa kỳ HK241

1. Tổng quan về sự phát triển máy tính

- **Thế hệ thứ nhất (1945-1955)**:
  - Sử dụng đèn chân không
  - Lập trình bằng ngôn ngữ máy
  - Kích thước lớn, tốn nhiều điện năng
  - Ví dụ: ENIAC (1946)
- **Thế hệ thứ hai (1955-1965)**:
  - Sử dụng transistor
  - Xuất hiện ngôn ngữ lập trình bậc cao (FORTRAN, COBOL)
  - Hệ thống xử lý theo lô (batch processing)
  - Ví dụ: IBM 7094
- **Thế hệ thứ ba (1965-1980)**:
  - Sử dụng mạch tích hợp (IC)
  - Đa chương trình (multiprogramming)
  - Hệ điều hành phát triển
  - Ví dụ: IBM System/360
- **Thế hệ thứ tư (1980-nay)**:
  - Vi xử lý (microprocessor)
  - Máy tính cá nhân phổ biến
  - Internet và World Wide Web
  - Điện toán di động và đám mây

2. Luật Moore

- Được đưa ra bởi Gordon Moore (đồng sáng lập Intel) năm 1965
- Dự đoán: Số lượng transistor trên chip gấp đôi mỗi 18-24 tháng
- Ảnh hưởng:
  - Tăng hiệu năng máy tính
  - Giảm chi phí sản xuất
  - Khả năng thu nhỏ kích thước thiết bị
- Ứng dụng mới: điện thoại thông minh, IoT, AI, xe tự lái
- Thách thức: giới hạn vật lý, tiêu thụ năng lượng

3. Các loại máy tính

- **Máy tính cá nhân (PC)**:
  - Đa năng, phổ biến
  - Desktop, laptop, tablet
  - Cân bằng giữa hiệu năng và chi phí
- **Máy tính nhúng (Embedded)**:
  - Tích hợp trong các thiết bị
  - Ví dụ: điều khiển ô tô, thiết bị y tế, đồ gia dụng thông minh
  - Yêu cầu tiết kiệm năng lượng, chi phí thấp
- **Máy chủ (Server)**:
  - Phục vụ nhiều người dùng qua mạng
  - Yêu cầu hiệu năng cao, độ tin cậy lớn
  - Ví dụ: máy chủ web, cơ sở dữ liệu, đám mây
- **Siêu máy tính (Supercomputer)**:
  - Hiệu năng cực cao cho tính toán phức tạp
  - Ứng dụng: dự báo thời tiết, mô phỏng vật lý, nghiên cứu gen
  - Ví dụ: IBM Summit, Fugaku

4. Hiểu về hiệu năng máy tính

- **Thuật toán**:
  - Ảnh hưởng trực tiếp đến số lượng phép tính
  - Độ phức tạp thuật toán: O(n), O(n log n), $O(n^2)$, ...
- **Ngôn ngữ lập trình**:
  - Ngôn ngữ bậc cao vs. bậc thấp
  - Ảnh hưởng đến số lượng mã máy được tạo ra
- **Trình biên dịch**:
  - Tối ưu hóa mã
  - Sinh mã hiệu quả cho kiến trúc cụ thể
- **Kiến trúc máy tính (ISA)**:
  - CISC vs. RISC
  - Ảnh hưởng đến số chu kỳ đồng hồ cần thiết cho mỗi lệnh
- **Bộ xử lý và hệ thống bộ nhớ**:

  - Tốc độ xử lý
  - Hiệu quả của bộ nhớ cache
  - Băng thông bộ nhớ
- **Hệ thống I/O**:
  - Tốc độ đọc/ghi ổ cứng
  - Băng thông mạng

5. Các lớp phần mềm chương trình

- **Phần mềm ứng dụng**:
  - Viết bằng ngôn ngữ bậc cao (Python, Java, C++, ...)
  - Giao diện người dùng
  - Xử lý nghiệp vụ cụ thể
- **Phần mềm hệ thống**:
  - *Hệ điều hành*:
    * Quản lý tài nguyên máy tính
    * Cung cấp giao diện cho phần mềm ứng dụng
    * Ví dụ: Windows, Linux, macOS
  - *Trình biên dịch*:
    * Chuyển đổi mã nguồn thành mã máy
    * Tối ưu hóa mã
    * Ví dụ: GCC, Clang, MSVC
- **Phần cứng**:
  - Bộ xử lý (CPU, GPU)
  - Bộ nhớ (RAM, ROM)
  - Thiết bị lưu trữ (HDD, SSD)
  - Bộ điều khiển I/O

6. Các thành phần của máy tính

- **CPU (Central Processing Unit)**:
  - *Đơn vị điều khiển (Control Unit)*:
    * Giải mã và thực thi lệnh
    * Điều phối hoạt động của các thành phần khác
  - *Đơn vị số học và logic (ALU)*:
    * Thực hiện các phép tính số học và logic
  - *Thanh ghi (Registers)*:
    * Lưu trữ dữ liệu tạm thời
    * Truy cập nhanh nhất
  - *Bộ nhớ cache*:
    * Lưu trữ dữ liệu và lệnh thường xuyên sử dụng
    * Giảm thời gian truy cập bộ nhớ chính
- **Bộ nhớ (Memory)**:
  - *RAM (Random Access Memory)*:
    * Bộ nhớ chính, truy cập nhanh
    * Mất dữ liệu khi mất điện
  - *ROM (Read-Only Memory)*:
    * Lưu trữ firmware, BIOS
    * Không mất dữ liệu khi mất điện
- **Thiết bị vào/ra (I/O devices)**:
  - *Thiết bị đầu vào*: Bàn phím, chuột, máy quét, ...
  - *Thiết bị đầu ra*: Màn hình, loa, máy in, ...
  - *Thiết bị lưu trữ*: Ổ cứng (HDD, SSD), USB, ...
  - *Thiết bị mạng*: Card mạng, Wi-Fi, Bluetooth, ...

7. 8 nguyên tắc vàng trong thiết kế máy tính

1. **Thiết kế theo Luật Moore**:
   - Tận dụng sự tăng trưởng về số lượng transistor
   - Dự đoán và chuẩn bị cho công nghệ trong tương lai
2. **Sử dụng trừu tượng để đơn giản hóa thiết kế**:
   - Chia nhỏ hệ thống thành các module
   - Sử dụng các lớp trừu tượng (ISA, ABI)
3. **Làm nhanh các trường hợp phổ biến**:
   - Tối ưu hóa cho các tác vụ thường xuyên thực hiện
   - Ví dụ: Tối ưu hóa bộ nhớ cache cho các lệnh phổ biến
4. **Hiệu năng thông qua song song (Parallelism)**:
   - Thực hiện nhiều tác vụ cùng lúc
   - Ví dụ: Đa nhân (multi-core), đa luồng (multi-threading)
5. **Hiệu năng thông qua đường ống (Pipelining)**:
   - Chia nhỏ quá trình xử lý thành các giai đoạn
   - Thực hiện đồng thời các giai đoạn khác nhau
6. **Hiệu năng thông qua dự đoán (Prediction)**:
   - Dự đoán trước các lệnh và chuẩn bị sẵn tài nguyên
   - Ví dụ: Dự đoán nhánh (Branch Prediction)
7. **Cấp bậc bộ nhớ (Hierarchy of memories)**:
   - Tổ chức bộ nhớ thành nhiều lớp (Cache, RAM, SSD/HDD)
   - Giảm thời gian truy cập dữ liệu bằng cách lưu trữ tạm thời các dữ liệu thường dùng
8. **Độ tin cậy qua dư thừa (Dependability via redundancy)**:
   - Sử dụng nhiều bản sao của dữ liệu hoặc phần cứng để phòng ngừa hư hỏng
   - Ví dụ: RAID, ECC memory

8. Tính toán hiệu năng

- **Thời gian CPU** = `Số chu kỳ xung nhịp` × `Thời gian mỗi chu kỳ`.
- **Tần số xung nhịp (Clock Rate)**: Số chu kỳ mỗi giây, đơn vị là Hertz (Hz) .
- **Thời gian thực thi (Execution Time)** = $\frac{1}{\text{Hiệu năng}}$.

9. Thời gian CPU và CPI

- **CPI (Cycles per Instruction)**: Số chu kỳ cần thiết để thực thi mỗi lệnh .
- Công thức tính:

$$\text{Thời gian CPU} = \frac{\text{Số lệnh} \times \text{CPI}}{\text{Tần số xung nhịp}}.$$

$$= \text{Số lệnh} \times \text{CPI} \times \text{Thời gian mỗi chu kỳ}$$

10. Đo lường hiệu năng

- Hiệu năng được đo lường bằng cách so sánh thời gian thực thi chương trình .
- Tỉ lệ hiệu năng giữa hai máy A và B:

$$\text{Hiệu năng của A so với B} = \frac{\text{Thời gian thực thi của B}}{\text{Thời gian thực thi của A}}.$$