**Tổ chức bộ xử lý (CPU)**

**🔹 1. Bộ xử lý trung tâm (CPU)**

Là thành phần chính điều khiển toàn bộ hoạt động của máy tính. Gồm 2 khối chính:

* **Đường đi dữ liệu (Datapath)**: xử lý dữ liệu.
* **Bộ điều khiển (Control Unit)**: điều phối hoạt động các thành phần.

**🔹 2. Đường đi dữ liệu**

Gồm các thành phần:

* **ALU** (Arithmetic Logic Unit): thực hiện tính toán và logic.
* **Mạch dịch (Shifter)**
* **Thanh ghi (Registers)**: bao gồm nhiều loại:
  + **PC (Program Counter)**: giữ địa chỉ lệnh tiếp theo.
  + **IR (Instruction Register)**: giữ lệnh hiện tại.
  + **SR (Status Register)**: giữ cờ trạng thái.
  + **MAR (Memory Address Register)**: địa chỉ cần truy cập trong bộ nhớ.
  + **MBR (Memory Buffer Register)**: dữ liệu trao đổi với bộ nhớ.
  + **TEMP**: thanh ghi tạm.
* **MUX (Multiplexor)**: điều phối dòng dữ liệu.
* **Bus**: kết nối các bộ phận.

**🔹 3. Bộ điều khiển**

Chịu trách nhiệm tạo ra các **tín hiệu điều khiển**:

* Điều khiển luồng dữ liệu giữa các thanh ghi, bộ nhớ, ALU…
* Có 2 cách triển khai:
  1. **Bằng mạch điện tử (Hardwired Control)** – dùng cho kiến trúc RISC.
     + Dựa trên **FSM (Finite State Machine)**.
     + Nhanh nhưng khó thay đổi.
  2. **Bằng vi chương trình (Microprogrammed Control)** – dùng cho CISC.
     + Linh hoạt, dễ mở rộng.

**🔹 4. Diễn tiến thi hành lệnh mã máy (Instruction Cycle)**

Gồm 5 giai đoạn:

1. **IF** – Lấy lệnh từ bộ nhớ.
2. **ID** – Giải mã lệnh, đọc các thanh ghi nguồn.
3. **EX** – Thực thi lệnh (tính toán, địa chỉ nhảy...).
4. **MEM** – Truy cập bộ nhớ (nếu cần).
5. **WB/RS** – Ghi kết quả vào thanh ghi đích.

**🔹 5. Ngắt quãng (Interrupt)**

* Là cơ chế giúp **CPU phản hồi với các sự kiện bất ngờ** từ phần cứng hoặc phần mềm.
* Khi xảy ra:
  1. Hoàn thành lệnh hiện tại.
  2. Lưu trạng thái CPU.
  3. Nhảy đến **ISR (Interrupt Service Routine)**.
  4. Sau đó khôi phục trạng thái, tiếp tục chương trình đang chạy.
* Ứng dụng:
  1. I/O, lỗi phần cứng, chia thời CPU, gọi hàm hệ điều hành...

**🔹 6. Kỹ thuật ống dẫn (Pipelining)**

* Là kỹ thuật cho phép **các lệnh được xử lý song song** ở các giai đoạn khác nhau.
* Ví dụ: 5 lệnh qua 5 giai đoạn IF → RS, thực hiện trong **9 chu kỳ** thay vì **25** nếu tuần tự.
* **Ràng buộc:**
  + Mỗi giai đoạn cần phần cứng riêng.
  + Lệnh phải được giải mã nhanh, ALU đủ mạnh.

**Khó khăn:**

1. **Do cấu trúc**: thiếu phần cứng → giải quyết bằng thêm tài nguyên.
2. **Do dữ liệu**: lệnh sau phụ thuộc dữ liệu lệnh trước.
3. **Do điều khiển**: nhảy không dự đoán được.

**🔹 7. Siêu ống dẫn (Superpipelining)**

* Là mở rộng của pipeline bằng cách **chia nhỏ mỗi giai đoạn pipeline** thành nhiều giai đoạn con.
* Nhằm tăng tốc độ xử lý (mỗi chu kỳ nhỏ hơn, nhiều lệnh hơn).

**Tóm lại:**

* CPU vận hành theo chu kỳ lệnh với nhiều bước rõ ràng.
* Pipelining giúp tăng hiệu suất nhưng yêu cầu phần cứng và thiết kế tốt.
* Interrupt là chìa khóa để hệ thống phản hồi với các sự kiện thực tế.
* Kiến trúc CPU là nền tảng quan trọng để hiểu lập trình máy và hiệu suất hệ thống.