Bài làm

Câu 1:

* Câu lệnh là các lệnh được viết bằng ngôn ngữ hợp ngữ hoặc ngôn ngữ máy để điều khiển vi xử lý thực hiện các thao tác cụ thể. Câu lệnh bao gồm mã gợi nhớ (mnemonic), toán hạng nguồn, và toán hạng đích. Ví dụ, câu lệnh "ADD R0, R1, R2" trong hợp ngữ ARM là lệnh cộng giá trị trong các thanh ghi R1 và R2, sau đó lưu kết quả vào thanh ghi R0
* Mã lệnh là phần của câu lệnh mà bộ xử lý nhận diện để thực thi hành động cụ thể. Trong tài liệu, mã lệnh được tách ra từ thanh ghi lệnh và được đưa vào bộ giải mã lệnh. Bộ giải mã sau đó xác định chức năng cần thực hiện dựa trên mã lệnh này​
* Mã thanh ghi (Opcode) là một phần của mã lệnh xác định loại thao tác mà vi xử lý phải thực hiện. Nó có thể được đưa vào các chân của bộ giải mã lệnh để chọn mạch chức năng tương ứng trong ALU để thực thi​
* Thanh ghi lệnh là một thanh ghi trong CPU, lưu trữ câu lệnh hiện tại mà vi xử lý cần thực hiện. Mã thao tác từ thanh ghi lệnh sẽ được giải mã để xác định hành động cụ thể mà CPU cần thực hiện​
* Thanh ghi con trỏ lệnh (Program Counter) lưu trữ địa chỉ của câu lệnh tiếp theo cần thực hiện. Khi một lệnh được thực thi, thanh ghi này sẽ cập nhật để trỏ đến câu lệnh kế tiếp trong chuỗi lệnh

Câu 2:

* **Quá trình thực hiện một câu lệnh trong CPU**

1. Lấy lệnh (Fetch): CPU đọc lệnh từ bộ nhớ và lưu trữ vào thanh ghi lệnh (IR).

* Địa chỉ của lệnh cần lấy được xác định bởi thanh ghi con trỏ lệnh (PC). Sau khi lệnh được nạp, PC được tăng lên để trỏ đến lệnh tiếp theo.

1. Giải mã lệnh (Decode): Lệnh vừa được nạp vào IR sẽ được giải mã để xác định các thao tác cần thực hiện.
2. Tính toán địa chỉ toán hạng (Operand Address Calculation): Nếu lệnh yêu cầu truy cập vào dữ liệu từ bộ nhớ, CPU sẽ tính toán địa chỉ của dữ liệu này.
3. Lấy toán hạng (Operand Fetch): CPU lấy toán hạng từ bộ nhớ hoặc từ các thanh ghi nội bộ.
4. Thực thi (Execute): CPU thực hiện các thao tác đã được xác định trong lệnh, ví dụ như cộng, trừ, dịch chuyển, v.v.
5. Lưu trữ kết quả (Store): Kết quả của lệnh sẽ được lưu trữ lại vào thanh ghi hoặc bộ nhớ, tùy thuộc vào lệnh.

Quá trình này có thể khác nhau tùy vào kiến trúc CPU. Ví dụ, vi xử lý họ 80x86 thực hiện đầy đủ cả sáu giai đoạn như đã mô tả, trong khi vi xử lý ARM với kiến trúc RISC (Reduced Instruction Set Computing) chỉ chia quá trình thực hiện thành ba giai đoạn: lấy lệnh, thực thi, và lưu kết quả​

* **Tại sao mỗi họ CPU lại có tập lệnh khác nhau**
* Mỗi họ CPU có tập lệnh khác nhau do các thiết kế kiến trúc khác biệt, nhằm tối ưu hóa cho các mục đích khác nhau như hiệu suất, hiệu năng, hay tiết kiệm năng lượng. Ví dụ:
* CISC (Complex Instruction Set Computing): Như vi xử lý họ 80x86, tập lệnh phức tạp với nhiều lệnh đa dạng, phù hợp cho các ứng dụng cần hiệu năng cao nhưng chấp nhận mức độ tiêu thụ năng lượng lớn.
* RISC (Reduced Instruction Set Computing): Như vi xử lý ARM, sử dụng tập lệnh đơn giản hơn, giúp tăng tốc độ xử lý và giảm tiêu thụ năng lượng, phù hợp với các thiết bị di động và nhúng​

Câu 3:

* **Mô hình lưu trữ phân cấp**
* Mô hình lưu trữ phân cấp trong hệ thống máy tính bao gồm các mức bộ nhớ khác nhau, mỗi mức có các đặc điểm về giá thành, dung lượng, và tốc độ truy xuất khác nhau. Từ bộ nhớ đệm (cache) đến bộ nhớ chính và cuối cùng là bộ nhớ thứ cấp, sự gần gũi với lõi vi xử lý tăng tốc độ truyền dữ liệu, nhưng giá thành cũng tăng và dung lượng giảm dần từ bộ nhớ thứ cấp đến bộ nhớ đệm​
* **Bộ nhớ vật lý, địa chỉ vật lý, không gian địa chỉ vật lý**
* Bộ nhớ vật lý: Là một chuỗi các phần tử nhớ liên tiếp với thứ tự tăng dần. Mỗi phần tử có thể có kích thước là một byte, từ 2 byte, hoặc 4 byte, tùy theo hệ thống máy tính.
* Địa chỉ vật lý: Là số thứ tự của từng byte trong bộ nhớ. Địa chỉ vật lý được sử dụng để truy cập trực tiếp vào các phần tử trong bộ nhớ.
* **Không gian địa chỉ vật lý:**
* Là phạm vi các địa chỉ mà hệ thống có thể sử dụng để truy cập vào bộ nhớ. Độ rộng của bus địa chỉ (S) xác định không gian địa chỉ này. Ví dụ, với bus địa chỉ 32 bit, không gian địa chỉ vật lý sẽ có kích thước tối đa là 2322^{32}232 byte, tức là 4 GB, với các địa chỉ từ 00000000 đến FFFFFFFF trong hệ thống số hexa, hoặc từ 0 đến 4294967295 trong hệ thống thập phân
* **Bộ giải mã địa chỉ với hệ thống có 32 bit địa chỉ**
* Bộ giải mã địa chỉ trong một hệ thống có 32 bit địa chỉ sẽ phải giải mã được 32 bit thông tin, tương đương với 4,294,967,296 (2^32) địa chỉ có thể. Không gian địa chỉ vật lý trong trường hợp này là từ 0 đến 4294967295 (thập phân), 00000000 đến FFFFFFFF (hexa), và 00000000 đến 11111111\_11111111\_11111111\_11111111 (nhị phân)​
* **Có thể dùng bộ giải mã địa chỉ 64 bit cho hệ thống 32 bit không?**
* Có thể sử dụng bộ giải mã địa chỉ 64 bit cho hệ thống 32 bit, nhưng điều này sẽ không tận dụng hết khả năng của bộ giải mã 64 bit. Trong trường hợp này, chỉ có 32 bit của bộ giải mã được sử dụng, và phần còn lại sẽ không có tác dụng, do đó không có lợi thế nào đáng kể từ việc sử dụng bộ giải mã lớn hơn yêu cầu.