

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

# **KIẾN TRÚC MÁY TÍNH**

**NGUYỄN ĐÌNH VIỆT**

**-2009-**

## LỜI NÓI ĐẦU

Máy tính điện tử (MTĐT) ra đời vào đầu thập kỷ thứ tư của thế kỷ 20 và phát triển như vũ bão, làm nên một cuộc cách mạng trong lĩnh vực tính toán và xử lý thông tin, mở ra một kỷ nguyên mới trong lịch sử nhân loại - Kỷ nguyên MTĐT. Cấu tạo của MTĐT ngày càng tinh vi và phức tạp, bao gồm nhiều thành phần khác nhau. Kiến trúc máy tính (KTMT) là ngành khoa học nghiên cứu việc thiết kế các thành phần cấu thành nên MTĐT mà người lập trình có thể nhận thấy được.

MTĐT không chỉ có các thành phần vật lý, thường được gọi là phần cứng (hardware) mà còn có cả các chương trình để điều khiển sự hoạt động của các thành phần phần cứng, chúng được gọi là phần mềm (software). Việc nghiên cứu thiết kế MTĐT là một bài toán khổng lồ và phức tạp, cần không ngừng được cải tiến. Chính vì vậy mà người ta phải chia bài toán lớn này thành các bài toán nhỏ hơn, mỗi bài toán nhỏ sẽ giải quyết một vấn đề cụ thể, sử dụng một mức trừu tượng thích hợp.

Giáo trình KTMT này trình bày các vấn đề chung nhất, các thành phần cơ bản nhất cấu thành nên MTĐT hiện đại. Giáo trình được biên soạn nhằm phục vụ cho sinh viên ngành Công nghệ thông tin. Tuy nhiên tác giả hy vọng nó cũng hữu ích cho nhiều bạn đọc khác có nhu cầu tìm hiểu lĩnh vực chuyên môn này, nhất là sinh viên các ngành khoa học tự nhiên.

Giáo trình gồm 9 chương, nội dung chính của từng chương như sau:

Chương 1 trình bày các khái niệm và nguyên lý cơ bản; Phác họa kiến trúc của MTĐT hiện đại như là một dãy các mức. Mức dưới cùng là mức vật lý, đó là phần cứng, bên trên là các mức phần mềm, mỗi mức này thể hiện ra như một cái máy ảo, mỗi máy ảo sử dụng một ngôn ngữ riêng của nó. Chương 1 cũng đi qua các cột mốc trong lịch sử phát triển của MTĐT.

Chương 2: Tổ chức hệ thống máy tính, trình bày các thành phần căn bản nhất của MTĐT là Bộ xử lý (Processor), Bộ nhớ (Memory) và Hệ thống vào/ra (Input/Output).

Chương 3: Mức logic số, đây là mức thấp nhất trong kiến trúc MTĐT. Tại mức này chúng ta nhìn thấy các phần tử logic và các phần tử nhớ cơ bản, chúng là những loại tế bào cấu thành MTĐT, được điều khiển bởi các tín hiệu điện biểu diễn các giá trị logic 0 và 1. Mức logic số chính là cái máy vật lý, là phần cứng của MTĐT.

Chương 4: Mức vi chương trình. Máy vật lý bao gồm nhiều đơn vị chức năng khác nhau, mỗi đơn vị này thực hiện một vài thao tác hết sức đơn giản, được coi là sơ cấp. Chương này nghiên cứu cách thức điều khiển và phối hợp hoạt động của các đơn vị chức năng, để máy vật lý thực hiện được các thao tác sơ cấp một cách hiệu quả nhất.

Chương 5: Mức máy thông thường. Tại mức này có một ngôn ngữ thường được các nhà sản xuất MTĐT gọi là "Ngôn ngữ máy". Thực chất đây không phải là ngôn ngữ để "nói chuyện" trực tiếp với phần cứng của MTĐT, mà nó được xây dựng dựa trên các vi chương trình, được nhà sản xuất nhúng sẵn vào trong phần cứng

(firmware). Mục đích của việc này là để đưa ra cho người lập trình một ngôn ngữ dễ hiểu hơn so với ngôn ngữ lập vi chương trình.

Chương 6: Mức máy hệ điều hành. Việc “nói chuyện” với MTĐT bằng ngôn ngữ Mức máy thông thường vẫn còn phức tạp, nhất là khi chúng ta muốn máy thực hiện các thao tác vào/ra. Hệ điều hành sẽ khắc phục vấn đề đó, ngoài ra nó còn thực hiện chức năng quản lý bộ nhớ và các tiến trình.

Chương 7: Mức ngôn ngữ assembly. Chương này tác giả chỉ giới thiệu sơ lược vì các bạn đọc là sinh viên ngành Công nghệ Thông tin chắc chắn sẽ có một môn học bắt buộc về lập trình bằng ngôn ngữ assembly.

Chương 8: Trình bày về nguyên lý hoạt động của một số thiết bị ngoại vi thông dụng như bàn phím, màn hình, máy in, ổ đĩa.

Chương 9: Trình bày khá chi tiết về máy vi tính IBM PC, trong đó có các thành phần cơ bản của hệ điều hành DOS và cách DOS tổ chức thông tin trên đĩa. Chương này liên quan đến các kiến thức cụ thể, mang nhiều tính ứng dụng thực tế.

Giáo trình KTMT này đã được sử dụng làm tài liệu giảng dạy từ năm 1989 ở Khoa Toán-Cơ-Tin học và sau đó là Khoa Công nghệ Thông tin, Trường Đại học Khoa học Tự nhiên thuộc Đại học Quốc gia Hà Nội. Từ năm 1998 đến nay, giáo trình được sử dụng cho sinh viên ngành Công nghệ Thông tin, Khoa Công nghệ nay là Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Tác giả xin bày tỏ sự biết ơn giáo sư Nguyễn Khang Cường đã cho nhiều nhận xét, chỉ bảo xác đáng trong quá trình biên soạn lần thứ nhất, năm 1998. Tác giả cũng xin cảm ơn giảng viên Hồ Đắc Phương, Khoa Công nghệ, Đại học Quốc gia Hà Nội, đã đọc giúp bản thảo và cho nhiều ý kiến đóng góp trong lần xuất bản này.

Mặc dù đã cố gắng nhưng do kiến thức còn hạn hẹp, chắc chắn không tránh khỏi việc vẫn còn thiếu sót. Tác giả chân thành mong muốn nhận được các ý kiến đóng góp của các thầy, các bạn đồng nghiệp, các bạn sinh viên và bạn đọc về mọi mặt của giáo trình.

Các ý kiến đóng góp xin gửi về địa chỉ:

Nguyễn Đình Việt, Trường Đại học Công nghệ, ĐHQGHN

144 Đường Xuân Thủy, Quận Cầu Giấy, Hà Nội

Tác giả

# CHƯƠNG 1: MỞ ĐẦU

## 1.1 NHỮNG KHÁI NIỆM VÀ NGUYÊN LÝ CƠ BẢN

### 1.1.1 Máy tính

Máy tính theo nghĩa chung nhất dùng để chỉ mọi phương tiện được sử dụng để thực hiện các phép biến đổi toán học, thí dụ bàn tính, máy tính bỏ túi, máy tính điện tử... Tuy nhiên trong thời đại ngày nay, khi nói đến máy tính người ta thường ngầm chỉ máy tính điện tử là máy tính được cấu tạo bởi các linh kiện điện tử.

### 1.1.2 Nguyên lý xây dựng và sự phân loại máy tính điện tử

#### 1.1.2.1 Các nguyên lý xây dựng máy tính điện tử

Máy tính điện tử làm việc theo hai nguyên lý cơ bản, đó là nguyên lý số và nguyên lý tương tự.

Nguyên lý số (Digital): sử dụng các trạng thái rời rạc của một đại lượng vật lý để biểu diễn số liệu, nguyên lý này còn được gọi là nguyên lý đếm.

Thí dụ về tình trạng rời rạc của đại lượng vật lý theo nguyên lý số được thể hiện trong bảng 1-01.

Bảng 1-01

Linh kiện	Đại lượng vật lý	Trạng thái 1	Trạng thái 2
Chuyển mạch điện tử	Dòng điện	Có (nối mạch)	Không (ngắt mạch)
Lõi ferit	Trường từ tính	Tồn tại	Đảo từ (đảo hướng)
Điôt/transistor	Dòng điện	Dẫn điện	Không dẫn điện

Nguyên lý tương tự (Analog): sử dụng một đại lượng vật lý biến đổi liên tục để biểu diễn số liệu, nguyên lý này còn có tên gọi là nguyên lý đo.

Thí dụ về đại lượng vật lý biến đổi liên tục theo nguyên lý tương tự được thể hiện trong bảng 1-02.

Bảng 1-02

Thiết bị	Đại lượng vật lý
Thước tính	Chiều dài
Máy tính điện tử tương tự	Điện áp

### 1.1.2.2 Phân loại máy tính điện tử

Có nhiều phương pháp và cách phân loại khác nhau, ở đây ta nêu phương pháp phân loại theo nguyên lý xây dựng máy tính điện tử. Theo phương pháp này máy tính được phân chia thành hai lớp lớn là máy tính tương tự và máy tính số. Mỗi lớp lớn này lại có thể được chia thành các lớp con, thí dụ máy tính đa năng và máy tính chuyên dụng...

#### Máy tính số (digital computer)

Máy tính số là loại máy tính sử dụng các đại lượng vật lý biến thiên rời rạc (dạng số) để biểu diễn các đại lượng cần tính toán. Những thông số cơ bản của máy tính số là: tốc độ hoạt động, hệ thống lệnh và số địa chỉ của các lệnh, các thiết bị nhớ và dung lượng tin của chúng, tổ hợp các thiết bị vào/ra số liệu, kích thước.v.v.

Người ta có thể phân loại máy tính số dựa trên một số cơ sở khác nhau, đó có thể là cách thức thi hành chương trình, là nhiệm vụ mà người thiết kế định ra cho máy tính.v.v. Sau đây là hai thí dụ.

1. Phân loại máy tính số (MTS) theo cách thức thi hành chương trình:

- MTS liên tiếp: Là MTS trong đó chương trình được thi hành từng lệnh một, hết lệnh này đến lệnh khác.
- MTS song song: Là MTS có thể thi hành đồng thời nhiều chương trình. MTS song song cần nhiều trang thiết bị hơn và phức tạp hơn MTS liên tiếp nhưng có tốc độ tác động cao hơn.
- MTS liên tiếp-song song: Là loại MTS trung gian giữa hai loại MTS nêu trên, trong đó các phép tính theo mã của chương trình được liên tiếp đưa vào các bộ phận của máy, còn các bộ phận thi hành các phép tính một cách song song.

Ngày nay trong tất cả các máy tính, kể cả loại máy tính được gọi là liên tiếp, người ta cũng áp dụng các cơ chế thực hiện song song ở các mức độ khác nhau để nâng cao tốc độ hoạt động chung của máy tính điện tử.

2. Phân loại máy tính số theo nhiệm vụ mà người thiết kế định ra cho nó:

- MTS chuyên dụng: Là loại MTS được chế tạo ra để giải một loại bài toán nhất định, nó thường đơn giản và rẻ hơn MTS đa năng nhờ việc có thể giảm bớt một số thành phần của máy và thậm chí cả việc rút gọn tập lệnh của bộ xử lý của máy.

Các MTS chuyên dụng được sử dụng rộng rãi hiện nay là:

- Các máy tính điều khiển, nằm trong các hệ thống điều khiển những đối tượng và quá trình thực.
- Các máy tính để giải một loại bài toán nhất định, chẳng hạn tính tổng các tích 2 số trong các bài toán về dự báo khí tượng, giải hệ phương trình vi phân thường...

- Các máy tính tìm thông tin, dùng để xử lý những lượng lớn các thông tin chữ và số.
- MTS đa năng: Là loại MTS được chế tạo ra để giải một lớp lớn các bài toán mà thành phần của lớp bài toán có thể còn chưa được nêu đầy đủ khi thiết kế máy.

Những tính chất cơ bản của MTS đa năng là:

- Có tập lệnh tương đối lớn, bao gồm nhiều nhóm, mỗi nhóm phục vụ cho một loại thao tác nhất định, để có thể thi hành nhiều loại chương trình khác nhau một cách có hiệu quả.
- Kích thước các thanh ghi trong bộ xử lý và độ lớn của một đơn vị bộ nhớ (mà sau này chúng ta sẽ gọi là word) phải không quá lớn cũng không quá nhỏ, thường bằng bội số của 8 bit. Như vậy chúng có thể chứa vừa vặn một số ký tự ASCII 8 bit hoặc chứa được con số đủ lớn cho hầu hết các chương trình ứng dụng sẽ chạy trên máy này. Các chương trình soạn thảo văn bản, chương trình quản lý.v.v. nói chung đều có dữ liệu kiểu ký tự.
- Các thiết bị nhớ có thể nhận vào, lưu trữ và đưa ra các số liệu, các kết quả và các chương trình... Còn trong MTS chuyên dụng các chương trình, hằng số... thường được cứng hoá.
- Hệ thống trao đổi thông tin (giao diện) giữa máy và người thuận tiện, giảm nhẹ các công việc mà con người phải thực hiện trong quá trình máy tính hoạt động.

### Máy tính tương tự (Analog Computer)

Máy tính tương tự (MTTT) là loại máy tính sử dụng các đại lượng vật lý biến thiên liên tục để biểu diễn các đại lượng cần tính toán. Đại lượng vật lý đó thường là điện áp hoặc dòng điện.

Mô hình hoá (modelling) là cơ sở cho sự hoạt động của MTTT, trong đó một quá trình vật lý thực sự hay các thành phần của nó được thay thế bằng một mô hình điện có các đặc tính tương tự. Nhờ có mô hình điện mà việc nghiên cứu có thể tiến hành đơn giản, thuận tiện và rẻ hơn.

Các MTTT vận hành rất thuận tiện, thường đưa ra kết quả dưới dạng đồ thị, đặc biệt là với thời gian cực kỳ ngắn.

MTTT có các nhược điểm chính sau: kết quả có độ chính xác không cao lắm, sự hoạt động của nó không mềm dẻo như MST, khả năng giải bài toán phụ thuộc mạnh vào chính phần cứng của máy.

Sự khác nhau cơ bản nhất giữa MTTT và MTS là MTS chỉ làm được các phép tính số học cổ điển như cộng, trừ, nhân, chia; để thực hiện các tổ hợp gồm các phép tính cộng và nhân.v.v. những cái mà bộ cộng của MTTT chỉ phải làm trong nháy mắt thì ở MTS phải có một chương trình đặc biệt để sắp xếp dần dần các phép tính số học chủ yếu thành các tổ hợp cần thiết.

Các MTS ngày nay có khả năng giải được hầu như mọi loại bài toán toán học và các tính toán logic phức tạp với tốc độ thực hiện, độ chính xác và độ tin cậy ngày càng cao. Chính vì vậy mà MTTT với các nhược điểm trên chỉ được sử dụng cho những ứng dụng có tính chất chuyên biệt, không phổ biến như MTS.

### Máy tính lai (Hybrid Computer)

Đó là loại máy tính kết hợp cả hai nguyên lý số và tương tự, trong hệ thống này có một nửa là số và một nửa là tương tự. Nửa số về thực chất là một máy tính số hoặc là một tập hợp các phần tử tính toán số. Nửa tương tự là một máy tính tương tự hoặc là một tập hợp các phần tử tính toán tương tự. Trong quá trình tính toán, hai nửa này truyền dữ liệu cho nhau thông qua các bộ chuyển đổi (converter). Việc đồng bộ hoạt động của hai nửa có thể do một đơn vị điều khiển riêng hoặc do đơn vị điều khiển của máy tính số đảm nhiệm.

#### 1.1.3 Ngôn ngữ máy, mức và máy ảo

##### • **Lệnh máy, ngôn ngữ máy và chương trình**

Các mạch điện tử của máy tính có thể hiểu và thực hiện trực tiếp được một tập hợp hữu hạn các lệnh rất đơn giản, chẳng hạn:

- Cộng hai số với nhau.
- Kiểm tra xem một số có bằng không hay không.
- Vận chuyển một nhóm dữ liệu từ vùng này của bộ nhớ sang một vùng khác.

Các lệnh rất đơn giản này thường được gọi là **chỉ thị** (instruction) **máy** hay **lệnh máy**. Tập các chỉ thị rất đơn giản đó tạo nên một ngôn ngữ, con người có thể sử dụng ngôn ngữ này để giao tiếp với máy tính. Một ngôn ngữ như vậy được gọi là **ngôn ngữ máy** (**machine language**). Một dãy các chỉ thị mô tả việc thực hiện một nhiệm vụ cụ thể như thế nào được gọi là **chương trình** (**program**).

*(Trong giáo trình này hai thuật ngữ lệnh và chỉ thị thường được sử dụng với nghĩa gần giống nhau. Thuật ngữ lệnh được sử dụng với nghĩa chung nhất, còn thuật ngữ chỉ thị thường dùng để chỉ các lệnh rất đơn giản. Tuy nhiên lệnh nào được coi là rất đơn giản lại tùy thuộc vào mức máy mà chúng ta đang nghiên cứu.)*

Khi nghiên cứu thiết kế một máy tính mới, người thiết kế phải quyết định ngôn ngữ máy của máy tính đó sẽ bao gồm những chỉ thị nào. Thông thường để làm giảm độ phức tạp và giá thành của các mạch điện tử, người ta cố gắng làm cho các chỉ thị càng đơn giản càng tốt, phù hợp với mục đích sử dụng của máy tính này và các đòi hỏi về hiệu suất máy.

##### • **Sự cần thiết của các ngôn ngữ cao hơn ngôn ngữ máy và vấn đề dịch**

Vì hầu hết các ngôn ngữ máy rất đơn giản, nên việc sử dụng chúng là khó và tẻ nhạt. Có thể khắc phục vấn đề này bằng cách thiết kế một tập chỉ thị mới để sử dụng hơn tập chỉ thị máy đã được xây dựng ngay bên trong máy. Tập các chỉ thị này cũng tạo thành một ngôn ngữ mà chúng ta sẽ gọi là ngôn ngữ mức (Level) 2 -

L2, còn tập chỉ thị được xây dựng ngay bên trong máy chúng ta sẽ gọi là ngôn ngữ L1.

Có hai cách để máy tính thi hành được các chương trình viết bằng L2, mà xét cho cùng về thực chất máy tính cũng chỉ có thể thực hiện được các chương trình được viết bằng ngôn ngữ máy của nó là ngôn ngữ L1.

Cách thứ nhất để máy tính thi hành chương trình được viết bằng L2 là đầu tiên thay thế mỗi chỉ thị trong chương trình này bằng một dãy tương đương gồm các chỉ thị trong ngôn ngữ L1. Kết quả thu được một chương trình gồm toàn các chỉ thị thuộc ngôn ngữ L1. Sau đó máy tính thực hiện chương trình mới bằng ngôn ngữ L1 chứ không phải chương trình cũ bằng ngôn ngữ L2. Kỹ thuật này được gọi là **dịch** hay **biên dịch - Compilation** (hay **translation**), còn chương trình thực hiện việc này được gọi là Trình biên dịch - **Compiler**.

Cách thứ hai là viết một chương trình bằng ngôn ngữ L1, có nhiệm vụ làm cho máy tính thi hành chương trình bằng ngôn ngữ L2. Nó lấy chương trình L2 làm dữ liệu vào, đọc và kiểm tra từng chỉ thị L2 một và thực hiện một dãy tương đương các chỉ thị L1 một cách trực tiếp. Cách này không đòi hỏi việc trước tiên phải sinh ra một chương trình mới bằng ngôn ngữ L1, nó được gọi là **thông dịch - interpretation**, còn chương trình thực hiện việc này có tên gọi là Trình thông dịch - **interpreter**.

Biên dịch và thông dịch giống nhau ở chỗ trong cả hai phương pháp, các chỉ thị bằng ngôn ngữ L2 cuối cùng được thực hiện bằng cách thi hành một dãy tương đương các chỉ thị L1. Điều khác nhau là ở chỗ, trong phương pháp dịch, toàn bộ chương trình L2 ban đầu được biến đổi thành chương trình L1, còn chương trình L2 thì có thể bỏ đi, sau đó chương trình L1 mới sinh ra này được thi hành. Trong phương pháp thông dịch, sau khi từng chỉ thị L2 được kiểm tra và giải mã (decoded), nó sẽ được thi hành ngay lập tức, không có việc sinh ra một chương trình L1 mới.

Cả hai phương pháp kể trên đều được sử dụng rộng rãi.

Để cho thuận tiện, chúng ta hình dung có một máy tính tương tượng hay gọi là máy tính ảo (virtual machine), mà ngôn ngữ máy là L2. Nếu có thể xây dựng một máy tính như vậy với giá đủ rẻ, thì chúng ta không cần phải có ngôn ngữ L1 và cái máy để thực hiện các chương trình bằng ngôn ngữ L1 nữa, chúng ta chỉ việc viết các chương trình bằng ngôn ngữ L2 và cho máy tính thực hiện trực tiếp. Việc xây dựng máy tính có ngôn ngữ L2 bằng các mạch điện tử là rất đắt, không khả thi, tuy nhiên chúng ta vẫn có thể viết các chương trình cho chiếc máy ảo này. Các chương trình này có thể được thông dịch hoặc dịch bằng một chương trình được viết bằng ngôn ngữ L1, chính chương trình này được thi hành trực tiếp bởi cái máy tính đang tồn tại. Nói theo cách khác, người ta có thể viết chương trình cho các máy tính ảo, cứ như thể chúng tồn tại thực sự.

Để có thể thực hiện được trên thực tế việc dịch và thông dịch, các ngôn ngữ L1 và L2 không được khác nhau nhiều quá. Điều hạn chế này thường có nghĩa là L2



mặc dầu tốt hơn L1, nhưng nó cũng chưa thể là một ngôn ngữ lý tưởng để viết hầu hết các chương trình ứng dụng.

Một điều rất tự nhiên là chúng ta có thể tạo ra một tập chỉ thị nữa thuận lợi hơn cho con người so với tập chỉ thị L2. Tập thứ ba này cũng tạo thành một ngôn ngữ, chúng ta sẽ gọi nó là L3. Sau đó chúng ta có thể viết chương trình bằng ngôn ngữ L3 như thể có một máy tính ảo với ngôn ngữ máy L3 tồn tại thực sự. Các chương trình như vậy có thể hoặc là được dịch sang ngôn ngữ L2 hoặc được cho thi hành bằng một trình thông dịch viết bằng ngôn ngữ L2.

Theo ý tưởng trên chúng ta có thể tạo ra cả một dãy các ngôn ngữ, mỗi ngôn ngữ sử dụng ngôn ngữ đứng trước nó làm cơ sở, ngôn ngữ ra đời sau thuận tiện hơn ngôn ngữ ra đời trước nó, cứ tiếp tục mãi cho tới khi có được một ngôn ngữ thích hợp. Vì vậy chúng ta có thể coi một máy tính được xây dựng theo ý tưởng này như một dãy các lớp hay các mức, lớp này nằm trên lớp kia, như minh hoạ ở hình 1-01. Ngôn ngữ (hay lớp) dưới cùng là đơn giản nhất còn ngôn ngữ ở cao nhất là phức tạp nhất.

Giải thích:

Các chương trình viết bằng Ln hoặc là được thông dịch bởi một trình thông dịch chạy trên một máy nằm bên dưới, hoặc là được dịch sang ngôn ngữ máy của một máy tính nằm bên dưới.

Các chương trình viết bằng L3 hoặc là được thông dịch bởi một trình thông dịch chạy trên M2 hoặc M1, hoặc là được dịch sang ngôn ngữ máy L2 hoặc L1.

Các chương trình viết bằng L2 hoặc là được thông dịch bởi một trình thông dịch chạy trên M1, hoặc là được dịch sang ngôn ngữ máy L1.

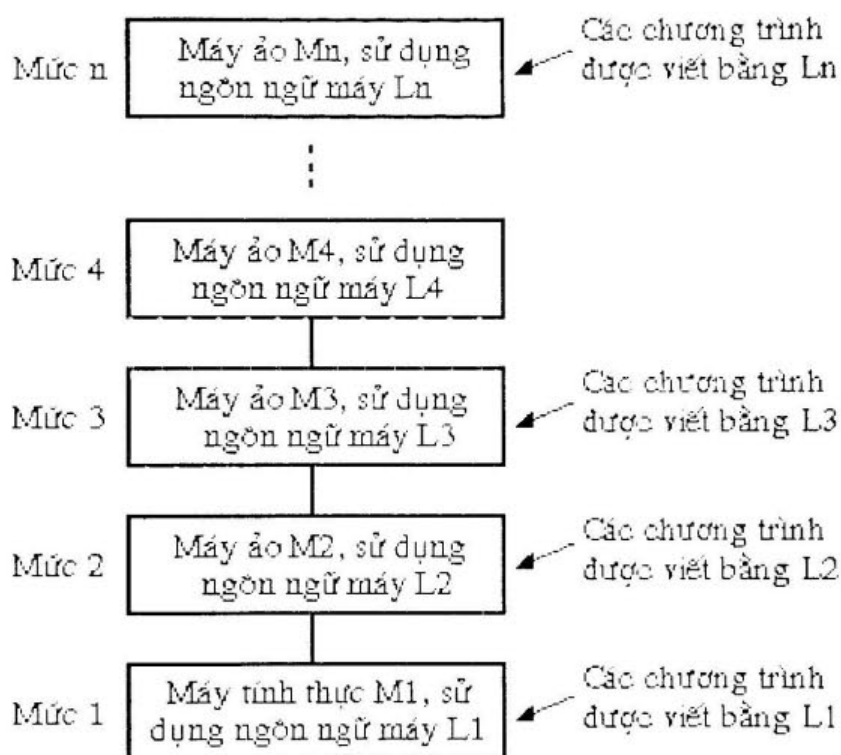
Các chương trình viết bằng L1 được thực hiện trực tiếp bằng các mạch điện tử.

Giữa ngôn ngữ và máy có một mối liên hệ quan trọng, mỗi máy có một ngôn ngữ máy nào đó, bao gồm tất cả các chỉ thị mà máy có thể thi hành. Máy xác định ngôn ngữ và một ngôn ngữ cũng xác định một máy, đó là cái máy có thể thi hành mọi chương trình được viết bằng ngôn ngữ này.

Nếu xây dựng một máy với một ngôn ngữ nhất định trực tiếp bằng các mạch điện tử thì có thể sẽ rất phức tạp và đắt tiền, tuy nhiên chúng ta vẫn có thể hình dung ra được chiếc máy này. Vào thời điểm hiện nay việc xây dựng một máy có ngôn ngữ máy là ngôn ngữ C, Pascal hoặc COBOL có thể coi là quá khó và phi thực tế nhưng rất có thể trong tương lai không xa những máy tính kiểu như vậy sẽ ra đời.

Một máy với n mức có thể được xem như n máy ảo khác nhau, mỗi máy ảo có một ngôn ngữ máy tương ứng. **Chúng ta sẽ sử dụng thuật ngữ "mức" và "máy ảo" với cùng một ý nghĩa.** Chỉ có các chương trình được viết bằng ngôn ngữ L1 là có thể được thực hiện trực tiếp bằng các mạch điện tử mà không cần sự can thiệp trung gian của việc dịch hoặc thông dịch. Các chương trình được viết bằng L2, L3,..., Ln hoặc là phải được thông dịch bằng một trình thông dịch chạy ở một mức

thấp hơn hoặc là phải được dịch sang một ngôn ngữ khác tương ứng với một mức thấp hơn.



Hình 1-01 Một máy tính nhiều mức

Một người làm công việc viết chương trình cho máy tính ảo mức  $n$  không nhất thiết phải biết các trình thông dịch hoặc trình dịch nằm ở mức bên dưới. Cấu trúc của máy đảm bảo rằng các chương trình này sẽ được thi hành bằng một cách nào đó. Người viết chương trình không phải bận tâm nhiều về việc chương trình sẽ được thực hiện trực tiếp bởi các mạch điện tử hay là chúng sẽ được thực hiện bởi một trình thông dịch, còn chính trình thông dịch này cũng sẽ lại được thực hiện bởi một trình thông dịch khác. Hầu hết những nhà lập trình sử dụng máy tính mức  $n$  chỉ quan tâm đến mức trên cùng, là mức khác xa nhất so với mức ngôn ngữ máy ở dưới cùng. Tuy nhiên những người muốn hiểu biết máy tính thực sự hoạt động như thế nào phải nghiên cứu tất cả các mức. Những người quan tâm đến việc nghiên cứu thiết kế các máy tính mới hoặc là thiết kế các mức mới, nghĩa là máy tính ảo mới phải làm quen với tất cả các mức chứ không phải chỉ riêng mức trên cùng.

#### 1.1.4 Các máy tính nhiều mức hiện đại

Hầu hết các máy tính hiện đại có từ hai mức trở lên. Các máy tính có đến 6 mức như minh họa ở hình 1-02 không phải là hiếm. Mức 0 nằm dưới cùng là phần cứng thực sự của máy. Các mạch điện của máy này thực hiện các chương trình ngôn ngữ máy ở mức 1. Thực tế thì còn có một mức nữa nằm bên dưới mức 0, mức này không được vẽ trên hình 1-02 bởi vì nó thuộc lĩnh vực điện tử, không thuộc nội dung giáo trình này, mức đó có tên gọi là mức **thiết bị (device-level)**. Tại mức này người thiết