

**BỘ LAO ĐỘNG THƯƠNG BINH VÀ XÃ HỘI
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG**



GIÁO TRÌNH

CẤU TRÚC MÁY TÍNH

(COMPUTER STRUCTURE)

BIÊN SOẠN
Nguyễn Duy Phúc

Vĩnh long – 10/2014

**BỘ LAO ĐỘNG THƯỜNG BINH VÀ XÃ HỘI
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG**



GIÁO TRÌNH

CẤU TRÚC MÁY TÍNH

(COMPUTER STRUCTURE)

Mã số học phần: TH1205

Số tín chỉ: 3

BIÊN SOẠN

Nguyễn Duy Phúc

Vĩnh long – 10/2014

LỜI NÓI ĐẦU

Để đáp ứng nhu cầu học tập và nghiên cứu của các bạn sinh viên, đặc biệt là sinh viên chuyên ngành Công nghệ thông tin, Khoa Công nghệ thông tin - Trường Đại học Sư phạm Kỹ thuật Vĩnh Long đã tiến hành biên soạn các giáo trình, bài giảng chính trong chương trình học. Giáo trình môn Cấu trúc máy tính được biên soạn dựa trên:

1. "Giáo trình cấu trúc máy tính" của các tác giả Tống Văn On, Hoàng Đức Hải do nhà xuất bản Lao động xã hội phát hành.
2. "Giáo trình Kiến trúc máy tính" của các tác giả Võ Văn Chín, Nguyễn Hồng Vân, Phạm Hữu Tài – Trường đại học Cần Thơ
3. "Giáo trình lập trình hợp ngữ" của nhà xuất bản Thống kê

Giáo trình này cũng được biên soạn dựa trên kinh nghiệm giảng dạy nhiều năm môn Cấu trúc máy tính của các giáo viên trong khoa chúng tôi. Ngoài ra chúng tôi cũng đã tham khảo rất nhiều tài liệu của các trường đại học trong và ngoài nước.

Cấu trúc tài liệu này được biên soạn dựa theo đề cương chi tiết môn học Cấu trúc máy tính của Khoa Công nghệ thông tin Trường Đại học Sư phạm Kỹ thuật Vĩnh Long dùng cho sinh viên chuyên ngành Công nghệ thông tin. Nội dung gồm 8 chương:

- Chương 1: Tổng quan
- Chương 2: Kiến trúc phần mềm bộ xử lý
- Chương 3: Tổ chức bộ xử lý
- Chương 4: Các cấp bộ nhớ
- Chương 5: Thiết bị nhập xuất
- Chương 6: Tổng quan về lập trình hợp ngữ
- Chương 7: Tổ chức bộ xử lý Intel 8086/8088
- Chương 8: Lập trình Turbo Assembler

Mục tiêu của giáo trình nhằm giúp các bạn sinh viên chuyên ngành có một tài liệu cô đọng dùng để học tập và nghiên cứu. Ngoài ra, chúng tôi nghĩ rằng các bạn sinh viên thuộc các chuyên ngành khác và những người quan tâm, tìm hiểu về hoạt động của hệ thống, lập trình hệ thống máy tính cũng sẽ tìm được những điều bổ ích trong quyển giáo trình này.

Mặc dù chúng tôi đã rất cố gắng trong quá trình biên soạn nhưng chắc chắn giáo trình này sẽ còn nhiều thiếu sót và hạn chế. Rất mong nhận được sự đóng góp ý kiến quý báu của sinh viên và các bạn đọc để giáo trình ngày càng hoàn thiện hơn.

Chân thành cảm ơn!

Vĩnh Long, ngày 20 tháng 10 năm 2014

MỤC LỤC

Chương 1	1
TỔNG QUAN	1
1.1. Ngôn ngữ, cấp máy và máy ảo	1
1.1.1. Giới thiệu.....	1
1.1.2. Máy nhiều cấp	2
1.1.3. Quá trình phát triển của máy nhiều cấp	4
1.2. Phần cứng và phần mềm	4
1.3. Lịch sử phát triển của máy tính.....	5
1.3.1. Máy tính cơ khí	5
1.3.2. Máy tính đèn điện tử - thế hệ thứ nhất	6
1.3.3. Máy tính transistor – thế hệ thứ hai	7
1.3.4. Máy tính IC – thế hệ thứ ba	8
1.3.5. Máy tính cá nhân và VLSI – thế hệ thứ tư.....	8
1.3.6. Khuynh hướng hiện tại.....	9
1.4. Biểu diễn thông tin trong máy tính	9
1.4.1. Khái niệm thông tin.....	9
1.4.2. Lượng thông tin và sự mã hoá thông tin	10
1.4.3. Biểu diễn các số	11
1.4.4. Số nguyên có dấu	12
1.4.5. Biểu diễn số với dấu chấm động.....	14
1.4.6. Số BCD (Binary Coded Decimal).....	16
1.4.7. Biểu diễn các ký tự.....	17
1.4.8. Đơn vị đo thông tin	18
CÂU HỎI ÔN TẬP VÀ BÀI TẬP.....	19
Chương 2.....	21
KIẾN TRÚC PHẦN MỀM BỘ XỬ LÝ	21
2.1. Thành phần cơ bản của một máy tính	21
2.2. Định nghĩa kiến trúc máy tính.....	22
2.3. Tập lệnh.....	23
2.3.1. Gán trị.....	23
2.3.2. Lệnh có điều kiện.....	24
2.3.3. Vòng lặp	25

MỤC LỤC

2.3.4. Thâm nhập bộ nhớ ngăn xếp	26
2.3.5. Các thủ tục	26
2.4. Kiến trúc RISC và CISC.....	28
2.5. Toán hạng	30
CÂU HỎI ÔN TẬP VÀ BÀI TẬP	30
Chương 3	31
TỔ CHỨC BỘ XỬ LÝ	31
3.1. Đường đi dữ liệu.....	31
3.2. Bộ điều khiển.....	33
3.2.1. Bộ điều khiển mạch điện tử	33
3.2.2. Bộ điều khiển vi chương trình	34
3.3. Diễn tiến thi hành lệnh mã máy.....	35
3.3.1. Đọc lệnh.....	35
3.3.2. Giải mã lệnh và đọc các thanh ghi nguồn.....	35
3.3.3. Thi hành lệnh	35
3.3.4. Thâm nhập bộ nhớ trong hoặc nhảy lần cuối	36
3.3.5. Lưu trữ kết quả	36
3.4. Ngắt quãng (Interrupt)	36
3.5. Kỹ thuật ống dẫn.....	37
3.6. Khó khăn trong kỹ thuật ống dẫn	38
3.6.1. Khó khăn do cấu trúc.....	38
3.6.2. Khó khăn do số liệu	38
3.6.3. Khó khăn do điều khiển.....	39
3.7. Siêu ống dẫn	40
3.8. Siêu vô hướng.....	41
3.9. Máy tính song song.....	42
CÂU HỎI ÔN TẬP VÀ BÀI TẬP	47
Chương 4	49
CÁC CẤP BỘ NHỚ.....	49
4.1. Các loại bộ nhớ.....	49
4.2. Các cấp bộ nhớ	50
4.3. Bộ nhớ cache	52
4.3.1. Khái niệm.....	52
4.3.2. Vận hành của cache	52
4.4. Hiệu quả của cache và các mức cache.....	58

4.5. Bộ nhớ trong.....	59
4.6. Bộ nhớ ảo	60
CÂU HỎI ÔN TẬP VÀ BÀI TẬP.....	63
Chương 5.....	65
THIẾT BỊ NHẬP XUẤT	65
5.1. Dẫn nhập	65
5.2. Đĩa từ.....	65
5.3. Đĩa quang	68
5.4. Các loại thẻ nhớ.....	69
5.5. Băng từ	70
5.6. Các chuẩn bus	71
5.7. Một số biện pháp an toàn dữ liệu trong việc lưu trữ thông tin	71
CÂU HỎI ÔN TẬP VÀ BÀI TẬP.....	75
Chương 6.....	77
TỔNG QUAN VỀ LẬP TRÌNH HỢP NGỮ.....	77
6.1. Ngôn ngữ máy.....	77
6.2. Hợp ngữ (Assembly Language)	77
6.2.1. Khái niệm	77
6.2.2. Trình hợp dịch (Assembler)	78
6.3. Ứng dụng của hợp ngữ.....	78
6.4. Các kiểu dữ liệu	79
6.4.1. Số học.....	79
6.4.2. Ký tự.....	80
6.4.3. Chú ý	81
CÂU HỎI ÔN TẬP VÀ BÀI TẬP.....	81
Chương 7	83
TỔ CHỨC BỘ XỬ LÝ INTEL 8086/8088.....	83
7.1. Tổ chức bộ nhớ	83
7.2. Tổ chức thanh ghi.....	84
7.2.1. Nhóm thanh ghi đoạn (segment register).....	84
7.2.2. Nhóm thanh ghi đa dụng (general register)	84
7.2.3. Nhóm thanh ghi con trỏ và chỉ mục (pointer and index register)	85
7.2.4. Thanh ghi con trỏ lệnh chương trình IP (Instruction Pointer)	88
7.2.5. Thanh ghi cờ hiệu (Flag register).....	88
7.3. Ngắt quãng (Interrupt)	89

MỤC LỤC

7.3.1. Định nghĩa	89
7.3.2. Phân loại ngắt	90
7.3.3. Bảng vector ngắt	90
7.4. Các kiểu định vị	91
7.4.1. Phép định địa chỉ thanh ghi	91
7.4.2. Phép định địa chỉ hằng.....	91
7.4.3. Phép định địa chỉ bộ nhớ	91
CÂU HỎI ÔN TẬP VÀ BÀI TẬP	94
CHƯƠNG 8	95
LẬP TRÌNH TURBO ASSEMBLER	95
8.1. Các thành phần cơ bản của hợp ngữ	95
8.1.1. Bộ ký tự của hợp ngữ, từ khoá, tên	95
8.1.2. Cấu trúc một lệnh hợp ngữ	95
8.1.3. Chỉ dẫn khai báo dữ liệu	98
8.2. Cấu trúc chương trình hợp ngữ	100
8.2.1. Tập tin dạng .COM	100
8.2.2. Tập tin dạng .EXE	103
8.3. Các lệnh cơ bản của Bộ xử lý Intel 8086/8088	106
8.3.1. Nhóm lệnh chuyển dữ liệu	107
8.3.2. Nhóm lệnh cờ hiệu	110
8.3.3. Nhóm lệnh dữ liệu qua cổng	111
8.3.4. Nhóm lệnh tính toán số học	112
8.3.5. Nhóm các lệnh dịch chuyển và quay (shift and rotate)	117
8.3.6. Nhóm các lệnh logic	119
8.3.7. Nhóm lệnh chuyển điều khiển	120
8.4. Chương trình con	126
8.4.1. Khái niệm	126
8.4.2. Khai báo chương trình con	126
8.4.3. Gọi chương trình con	126
8.4.4. Truyền tham số	128
8.5. Macro	129
8.5.1. Khái niệm	129
8.5.2. Khai báo và gọi Macro	130
8.5.3. Macro và chương trình con	131
8.5.4. Chỉ dẫn LOCAL	131

CÂU HỎI ÔN TẬP VÀ BÀI TẬP.....	133
PHỤ LỤC	141
Các hàm của DOS với ngắt 21h.....	141
1. Hàm 01h.....	141
2. Hàm 02h.....	141
3. Hàm 08h.....	141
4. Hàm 09h.....	141
5. Hàm 0Ah.....	142
TÀI LIỆU THAM KHẢO	143

TaiLieu.vn

Chương 1

TỔNG QUAN

1.1. NGÔN NGỮ, CẤP MÁY VÀ MÁY ẢO

1.1.1. Giới thiệu

Máy tính số (digital computer) là máy giải quyết các vấn đề bằng cách thực hiện các chỉ thị do con người cung cấp. Chuỗi các chỉ thị này gọi là chương trình (program). Các mạch điện tử trong một máy tính số sẽ thực hiện một số giới hạn các chỉ thị đơn giản cho trước. Tập hợp các chỉ thị này gọi là tập lệnh của máy tính. Tất cả các chương trình muốn thực thi đều phải được biến đổi sang tập lệnh trước khi được thi hành. Các lệnh cơ bản là:

- Cộng 2 số
- So sánh số với 0
- Di chuyển dữ liệu

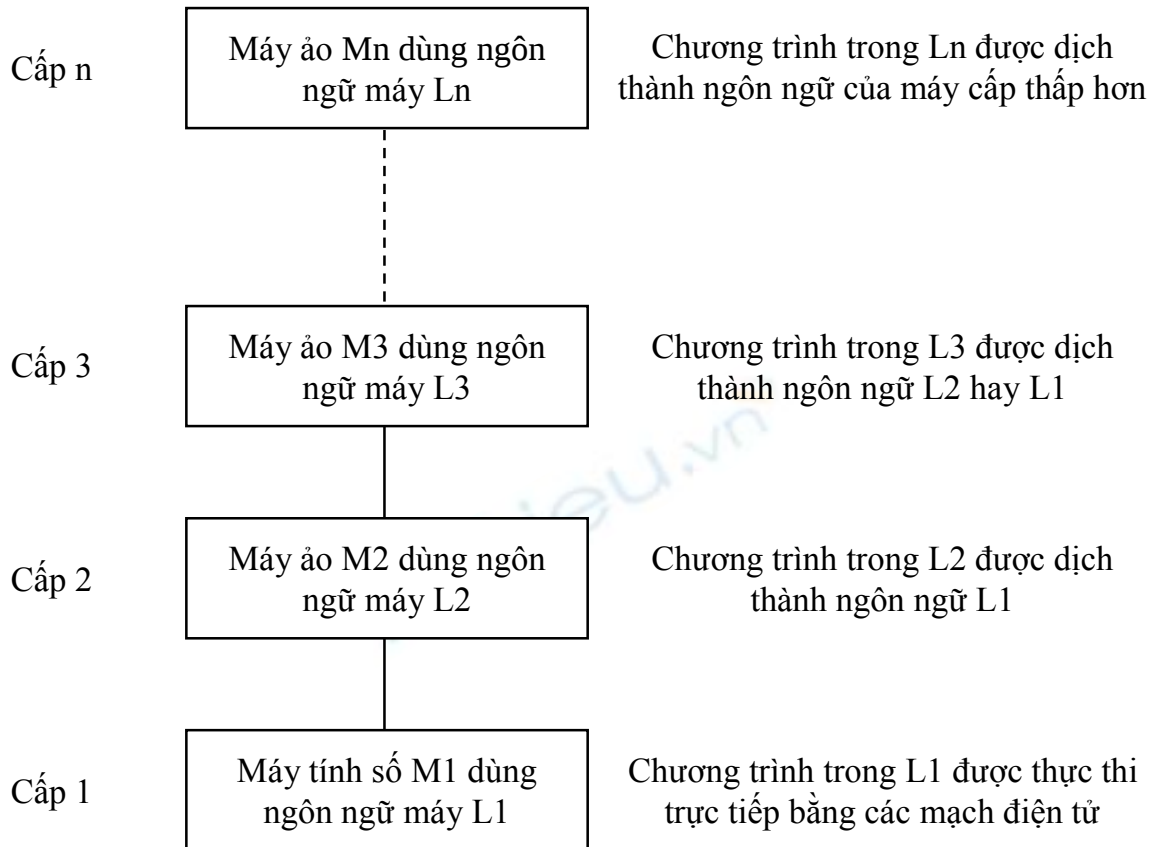
Tập lệnh của máy tính tạo thành một ngôn ngữ giúp con người có thể tác động lên máy tính, ngôn ngữ này gọi là ngôn ngữ máy (machine language). Tuy nhiên, hầu hết các ngôn ngữ máy đều đơn giản nên để thực hiện một yêu cầu nào đó, người thiết kế phải thực hiện công việc một cách phức tạp. Đó là chuyển các yêu cầu này thành các chỉ thị có chứa trong tập lệnh của máy. Vấn đề này có thể giải quyết bằng cách thiết kế một tập lệnh mới thích hợp cho con người hơn tập lệnh đã cài đặt sẵn trong máy tính (built-in). Ngôn ngữ máy sẽ được gọi là ngôn ngữ cấp 1 (L1) và ngôn ngữ vừa được hình thành gọi là ngôn ngữ cấp 2 (L2).

Một phương pháp thực thi chương trình L2 là chuyển một lệnh trong L2 bằng một chuỗi các lệnh tương đương trong L1. Kết quả là sẽ tạo thành một chương trình L1 và máy tính sẽ thực hiện chương trình tương đương L1 thay vì thực hiện chương trình L2. Kỹ thuật này gọi là biên dịch (compile). Cách khác là một lệnh trong chương trình L2 sẽ được xem như dữ liệu ngõ vào của chương trình L1 và toàn bộ chương trình L2 sẽ được thực thi tuần tự. Kỹ thuật này gọi là thông dịch (interpret), nó không yêu cầu tạo ra một chương trình mới trong L1.

Biên dịch và thông dịch đều thực hiện chương trình L2 thông qua tập lệnh trong chương trình L1. Chúng khác nhau ở chỗ là khi biên dịch thì toàn bộ chương trình L2 sẽ được chuyển thành chuỗi lệnh L1 rồi sau đó mới được thực thi còn đối với phương pháp thông dịch thì sẽ thực thi từng lệnh trong L2. Để thuận tiện hơn, ta giả sử tồn tại một máy tính sử dụng ngôn ngữ máy là L2, ta gọi máy tính này là máy ảo (virtual machine).

Tuy nhiên, trong thực tế, để có thể thực hiện biên dịch và thông dịch, các ngôn ngữ L1 và L2 không được khác nhau nhiều. Như vậy, ngôn ngữ L2 cũng không thật sự giúp ích nhiều cho người thiết kế. Do đó, một tập lệnh kế tiếp được hình thành sẽ hướng về con người nhiều hơn là máy tính, tập lệnh này sẽ tạo thành một ngôn ngữ và ta gọi là ngôn ngữ L3. Ta có thể viết các chương trình trong L3 như là đã tồn tại máy tính sử dụng ngôn ngữ L3 (máy ảo L3). Các chương trình này sẽ được dịch sang ngôn ngữ L2 và được thực thi bằng một chương trình dịch L2.

Việc xây dựng toàn bộ chuỗi các ngôn ngữ, mỗi ngôn ngữ được tạo ra sẽ thích hợp hơn ngôn ngữ trước đó sẽ có thể tiếp tục cho đến khi nhận được ngôn ngữ thích hợp nhất. Sơ đồ một máy ảo n cấp có thể biểu diễn như sau:



Hình 1.1. Máy ảo n cấp

Một máy tính số có n cấp có thể xem như có $n-1$ máy ảo khác nhau, mỗi máy ảo có một ngôn ngữ máy riêng. Các chương trình viết trên các máy ảo này không thể thực thi trực tiếp mà phải dịch thành các ngôn ngữ máy cấp thấp hơn. Chỉ có máy thật dùng ngôn ngữ máy L_1 mới có thể thực thi trực tiếp bằng các mạch điện tử. Một lập trình viên sử dụng máy ảo cấp n không cần biết tất cả các trình dịch này. Chương trình trong máy ảo cấp n sẽ được thực thi bằng cách dịch thành ngôn ngữ máy cấp thấp hơn và ngôn ngữ máy này sẽ được dịch thành ngôn ngữ máy thấp hơn nữa hay dịch trực tiếp thành ngôn ngữ máy L_1 và thực thi trực tiếp trên các mạch điện tử.

1.1.2. Máy nhiều cấp

Hầu hết các máy tính hiện nay đều có 6 cấp.

Cấp 0 chính là phần cứng của máy tính. Các mạch điện tử của cấp này sẽ thực thi các chương trình ngôn ngữ máy của cấp 1. Trong cấp logic số, đối tượng quan tâm là các cổng logic. Các cổng này được xây dựng từ một nhóm các transistor.

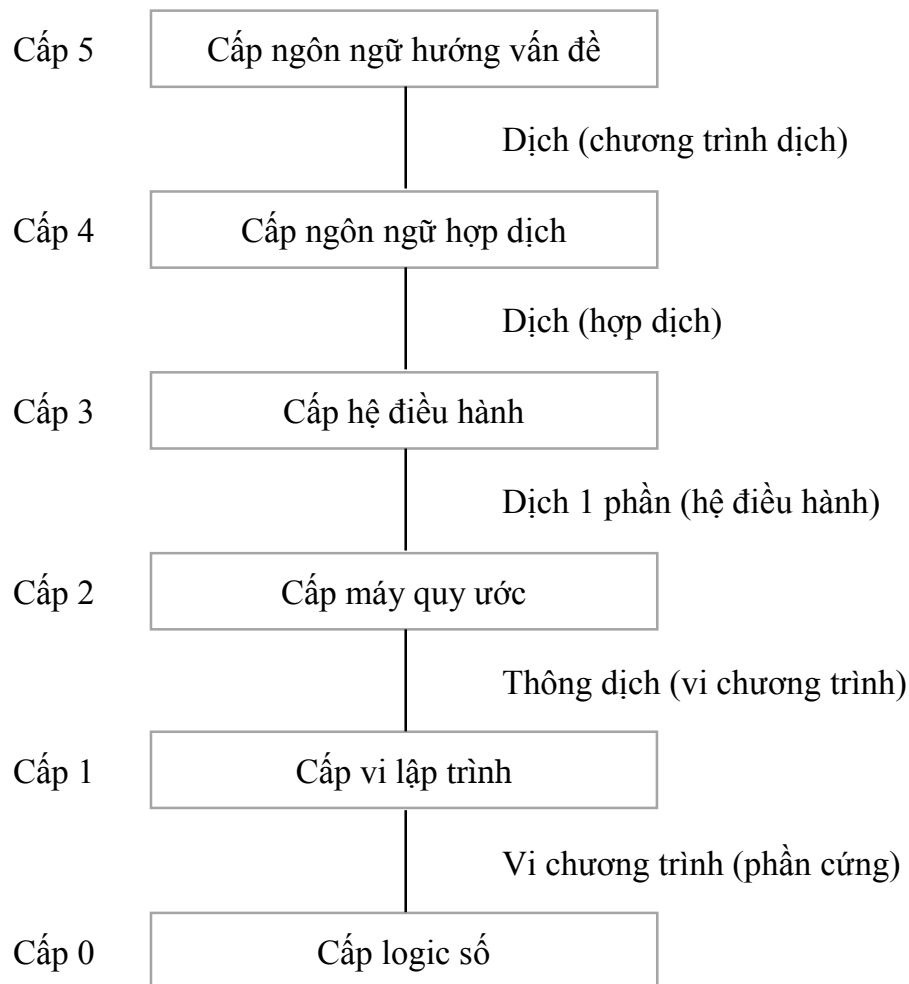
Cấp 1 là cấp ngôn ngữ máy thật sự. Cấp này có một chương trình gọi là vi chương trình (microprogram), vi chương trình có nhiệm vụ thông dịch các chỉ thị của cấp 2. Hầu hết các lệnh trong cấp này là di chuyển dữ liệu từ phần này đến phần khác của máy hay thực hiện việc một số kiểm tra đơn giản.

Mỗi máy cấp 1 có một hay nhiều vi chương trình chạy trên chúng. Mỗi vi chương trình xác định một ngôn ngữ cấp 2. Các máy cấp 2 đều có nhiều điểm chung ngay cả các máy cấp 2 của các hãng sản xuất khác nhau. Các lệnh trên máy cấp 2 được thực thi bằng cách thông dịch bởi vi chương trình mà không phải thực thi trực tiếp bằng phần cứng.

Cấp thứ 3 thường là cấp hỗn hợp. Hầu hết các lệnh trong ngôn ngữ của cấp máy này cũng có trong ngôn ngữ cấp 2 và đồng thời có thêm một tập lệnh mới, một tổ chức bộ nhớ khác và khả năng chạy 2 hay nhiều chương trình song song. Các lệnh mới thêm vào sẽ được thực thi bằng một trình thông dịch chạy trên cấp 2, gọi là hệ điều hành. Nhiều lệnh cấp 3 được thực thi trực tiếp do vi chương trình và một số lệnh khác được thông dịch bằng hệ điều hành (do đó, cấp này là cấp hỗn hợp).

Cấp 4 thật sự là dạng tượng trưng cho một trong các ngôn ngữ. Cấp này cung cấp một phương pháp viết chương trình cho các cấp 1, 2, 3 dễ dàng hơn. Các chương trình viết bằng hợp ngữ được dịch sang các ngôn ngữ của cấp 1, 2, 3 và sau đó được thông dịch bằng các máy ảo hay thực tương ứng.

Cấp 5 bao gồm các ngôn ngữ được thiết kế cho người lập trình nhằm giải quyết một vấn đề cụ thể. Các ngôn ngữ này được gọi là cấp cao. Một số ngôn ngữ cấp cao như Basic, C, Cobol, Fortran, Lisp, Prolog, Pascal và các ngôn ngữ lập trình hướng đối tượng như C++, J++, ... Các chương trình viết bằng các ngôn ngữ này thường được dịch sang cấp 3 hay 4 bằng các trình biên dịch (compiler).



Hình 1.2. Các cấp trên máy tính số

1.1.3. Quá trình phát triển của máy nhiều cấp

Các máy tính đầu tiên trong thập niên 40 chỉ có 2 cấp: cấp máy quy ước và cấp logic số. Các lập trình viên phải làm việc trên cấp máy quy ước và chương trình được thực thi trên cấp logic số. Trong thập niên 50, Wikes đề xuất ý tưởng thiết kế máy tính 3 cấp. Máy tính này có một trình thông dịch cài đặt sẵn, không thay đổi, có nhiệm vụ thực thi các chương trình trong cấp máy quy ước. Như vậy, phần cứng chỉ thực thi các vi chương trình với số lệnh giới hạn nên các mạch điện tử cũng đơn giản hơn.

Trình dịch hợp ngữ (assembler) và các trình biên dịch cho ngôn ngữ cấp cao (compiler) phát triển vào những năm 50 tạo điều kiện dễ dàng hơn cho lập trình viên. Tuy nhiên, vào lúc này, lập trình viên phải tự điều hành máy. Vào những năm 60, việc tự động hóa công việc điều hành bắt đầu được thực hiện. Một chương trình gọi là hệ điều hành (operating system) luôn được lưu trữ bên trong máy tính. Lập trình viên cung cấp các thẻ điều khiển và chương trình, chúng sẽ được đọc và thực thi bằng hệ điều hành.

Trong nhiều năm tiếp theo, hệ điều hành càng trở nên phức tạp. Các lệnh, tiện ích và đặc trưng mới được thêm vào cấp máy quy ước cho đến khi xuất hiện một cấp mới. Một số lệnh của cấp mới này giống như cấp máy quy ước nhưng một số lệnh lại hoàn toàn khác, nhất là các lệnh xuất nhập. Vào những năm đầu thập niên 60, các nghiên cứu ở đại học Dartmouth, MIT đã phát triển các hệ điều hành cho phép lập trình viên có thể tác động trực tiếp lên máy tính. Trong các hệ thống này, thiết bị đầu cuối từ xa được nối với máy tính trung tâm qua các đường điện thoại. Một lập trình viên có thể gõ chương trình và nhận kết quả trả về tức thời ở bất cứ nơi nào có thiết bị đầu cuối. Các hệ thống này gọi là hệ thống chia sẻ thời gian (time-sharing system).

1.2. PHẦN CỨNG VÀ PHẦN MỀM

Các chương trình viết bằng ngôn ngữ máy (cấp 1) được thực thi trực tiếp bằng các mạch điện tử của máy tính, không có trình thông dịch và biên dịch nào can thiệp vào. Các mạch điện tử cùng với bộ nhớ và các thành phần xuất / nhập tạo nên phần cứng máy tính. Phần cứng bao gồm các mạch tích hợp, các board mạch in, cable, nguồn cung cấp, bộ nhớ, thiết bị đầu cuối, ...

Phần mềm bao gồm các giải thuật và các biểu diễn của các giải thuật này gọi là chương trình. Nó chính là tập hợp các lệnh tạo thành một chương trình, chứ không phải là các phương tiện vật lý lưu trữ chúng.

Một dạng trung gian giữa phần mềm và phần cứng gọi là phần dẻo (firmware). Nó chính là thành phần bao gồm phần mềm được đặt vào bên trong các mạch điện tử trong quá trình sản xuất. Phần dẻo được dùng khi chương trình không thay đổi hay hiếm khi phải thay đổi như chương trình điều khiển đặt trong ROM BIOS.

Một thao tác bất kỳ thực thi bằng phần mềm có thể được gắn trực tiếp vào phần cứng và một lệnh bất kỳ thực thi bằng phần cứng cũng có thể được mô phỏng bằng phần mềm. Quyết định đặt một số chức năng vào phần mềm và các chức năng khác vào phần cứng dựa trên các yếu tố giá thành, tốc độ, độ tin cậy. Trên nhiều máy tính đầu tiên, phần cứng và phần mềm được phân biệt rõ ràng. Phần cứng thực hiện vài lệnh đơn giản như cộng và nhảy, các thủ tục khác phải do lập trình viên tự thiết kế. Sau đó, một số thao tác thường xuyên thực thi đòi hỏi các nhà thiết kế hướng đến yêu cầu xây dựng các mạch điện tử thực thi các thao tác này. Kết quả là hình thành xu hướng di chuyển các

thao tác theo hướng từ cấp cao xuống cấp thấp hơn. Một số thao tác trước đây được lập trình ở cấp máy quy ước, sau đó được chuyển xuống thực thi ở phần cứng.

Tuy nhiên, khi xuất hiện thế hệ máy tính dùng vi lập trình và thế hệ máy tính nhiều cấp, lại xuất hiện xu hướng ngược lại, nghĩa là di chuyển các thao tác từ cấp thấp lên cấp cao hơn. Ví dụ như lệnh cộng sẽ được thực hiện trực tiếp bằng phần cứng ở các máy trước kia. Đối với máy tính được vi lập trình hóa, lệnh cộng của cấp máy quy ước được thông dịch bằng một vi chương trình chạy trên cấp thấp nhất và được thực thi bằng một chuỗi các bước nhỏ: tìm lệnh, nạp lệnh, xác định lệnh, định vị dữ liệu, tìm và nạp dữ liệu từ bộ nhớ, thực thi phép cộng và lưu trữ kết quả.

Một số đặc trưng trước đây được lập trình ở cấp máy quy ước, sau đó được thực hiện bằng phần cứng hay vi chương trình:

- Các lệnh nhân, chia số nguyên.
- Các lệnh xử lý dấu chấm động.
- Các lệnh gọi thủ tục và quay về từ lệnh gọi thủ tục.
- Các lệnh đếm.
- Các lệnh quản lý chuỗi ký tự.
- Các đặc trưng làm tăng tốc độ tính toán chuỗi: định địa chỉ chỉ số và định địa chỉ gián tiếp.
- Các đặc trưng cho phép chương trình di chuyển trong bộ nhớ sau khi đã thực thi (cấp phát lại bộ nhớ).
- Các xung clock cho thủ tục định thời.
- Các ngắt báo hiệu cho máy tính.
- Khả năng chuyển đổi quá trình.

Như vậy, ta thấy ranh giới giữa phần cứng và phần mềm là không nhất định và thường xuyên thay đổi. Theo quan điểm của lập trình viên, cách thức thực thi một lệnh là không quan trọng, ngoại trừ tốc độ thực thi. Như vậy, phần cứng của người này có thể là phần mềm của người kia. Từ đó dẫn đến ý tưởng thiết kế máy tính có cấu trúc (structured computer). Đó là cấu trúc một máy tính thành một chuỗi các cấp, lập trình viên làm việc trên cấp n không quan tâm đến các cấp khác.

1.3. LỊCH SỬ PHÁT TRIỂN CỦA MÁY TÍNH

1.3.1. Máy tính cơ khí

Năm 1942, nhà khoa học Pháp Blaise Pascal xây dựng một máy đầu tiên thực hiện công việc tính toán. Đây là thiết bị hoàn toàn bằng cơ khí sử dụng các bánh răng và cung cấp lực bằng một cánh tay quay. Nó chỉ thực hiện được các phép toán cộng và trừ. Sau đó 30 năm, nhà toán học Đức Baron Gottfried Wilhelm von Leibniz xây dựng một máy cơ khí làm được phép nhân và chia.

Sau đó, giáo sư Charles Babbage đã thiết kế và xây dựng máy sai phân (difference engine). Nó được thiết kế để chạy một giải thuật đơn: phương pháp sai phân hữu hạn sử dụng các đa thức và cũng chỉ thực hiện các phép toán cộng và trừ. Năm 1834, Babbage thiết kế và xây dựng máy phân tích (analytical engine). Máy phân tích có 4 thành phần: bộ lưu trữ (bộ nhớ), bộ tính toán, thành phần nhập (đầu đọc thẻ đục lỗ) và thành phần

xuất (in và đục lỗ). Bộ tính toán có thể nhận các toán hạng từ bộ lưu trữ, thực hiện phép toán cộng, trừ, nhân hay chia chúng và trả kết quả về bộ lưu trữ.

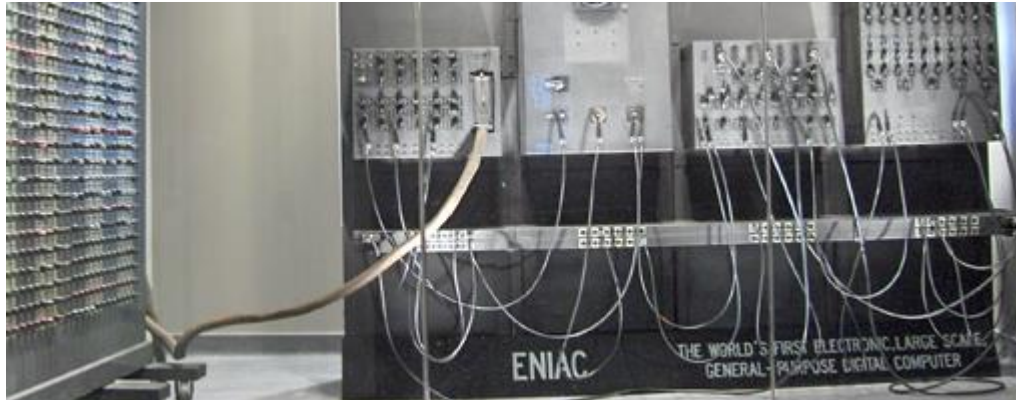
Phát triển tiếp theo của máy phân tích là máy đa năng. Máy đọc lệnh từ các thẻ đục lỗ và thực thi chúng. Bằng cách đục lỗ một chương trình khác trên thẻ nhập, máy phân tích có khả năng thực hiện các tính toán khác. Lập trình viên máy tính đầu tiên là Ada Lovelace đã tạo ra phần mềm cho máy phân tích.

Vào những năm 1930, Konrad Zuse xây dựng một chuỗi các máy tính toán tự động bằng cách sử dụng các relay từ. Sau đó, John Atanasoff và George Stibbitz đã thiết kế các máy tính (calculator). Máy của Atanasoff sử dụng số nhị phân và có các tụ điện làm cho bộ nhớ được làm tươi theo chu kỳ. Tuy nhiên, máy này bị thất bại do công nghệ phần cứng không tương xứng với ý tưởng thiết kế.

Năm 1944, Aiken hoàn tất máy tính Mark 1, có tất cả 72 từ, mỗi từ 23 số thập phân và có thời gian một chu kỳ là 6 giây. Việc nhập và xuất thực hiện bằng các băng giấy đục lỗ.

1.3.2. Máy tính đèn điện tử - thế hệ thứ nhất

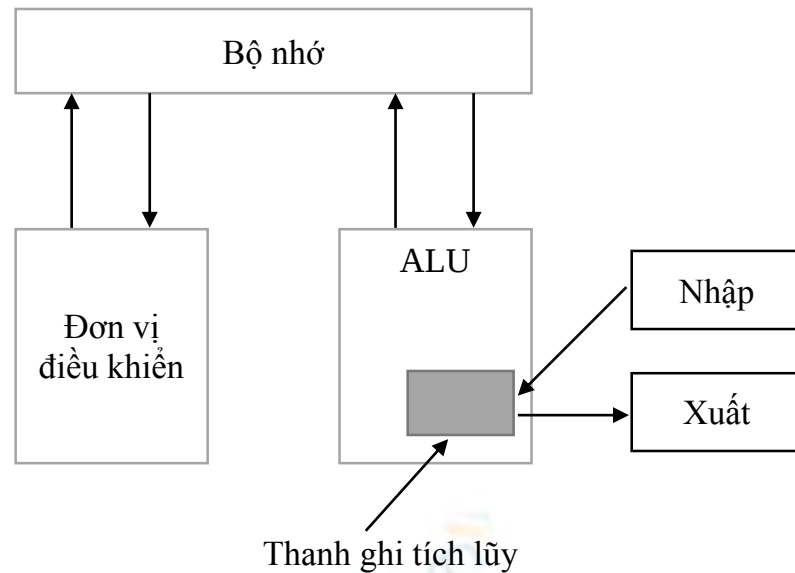
Năm 1943, máy tính số điện tử đầu tiên trên thế giới bắt đầu hoạt động, máy Colossus. Colossus do Alan Turing thiết kế nhằm thực hiện giải mã các thông điệp đã mã hóa trong chiến tranh thế giới thứ 2. Cũng trong năm 1943, Mauchley và Presper Eckert bắt đầu tiến hành xây dựng máy tính ENIAC (Electronic Numerical Integrator And Computer). ENIAC gồm 1800 đèn điện tử và 1500 rơ le (relay), cân nặng 30 tấn, công suất tiêu thụ 140 kWh. Nó có tất cả 20 thanh ghi, mỗi thanh ghi có thể lưu trữ một số thập phân 10 chữ số.



Hình 1.3. Một bảng điều khiển của máy tính ENIAC

Sau đó, John von Neumann thiết kế máy IAS dựa cơ sở trên máy EDVAC, là một phiên bản nâng cao của ENIAC. Máy von Neumann có 5 phần cơ bản: bộ nhớ, đơn vị luận lý số học (ALU – Arithmetich Logic Unit), đơn vị điều khiển chương trình, thiết bị nhập và thiết bị xuất. Bộ nhớ có tất cả 4096 từ, mỗi từ lưu trữ 40 bit. Mỗi từ chứa 2 lệnh 20 bit hay một số nguyên có dấu 39 bit. Mỗi lệnh 20 bit gồm có 8 bit xác định loại lệnh và 12 bit xác định 1 trong 4096 từ nhớ.

Vào cùng thời gian của máy IAS, các nhà nghiên cứu ở MIT cũng đang xây dựng máy tính Whirlwind 1. Từ nhớ của nó dài 16 bit và thiết kế để điều khiển thời gian thực.



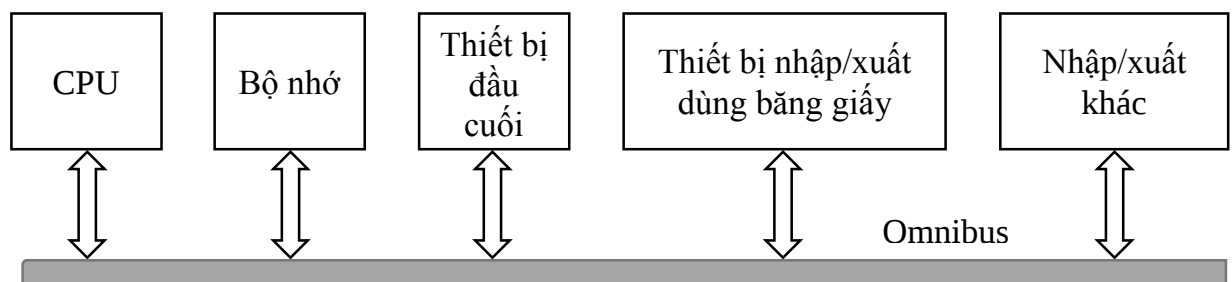
Hình 1.4. Máy von Neumann

1.3.3. Máy tính transistor – thế hệ thứ hai

Năm 1948, John Bardeen, Walter Brattain và William Shockley phát minh ra transistor đã làm cuộc cách mạng trong lĩnh vực máy tính. Máy tính transistor đầu tiên được xây dựng tại MIT, máy TX-0 (Transistorized experimental computer 0), có 16 bit tương tự như Whirlwind 1.

Năm 1961, máy tính PDP-1 xuất hiện có 4K từ 18 bit và khoảng thời gian một chu kỳ là 5 μ s. Vài năm sau, PDP-8 ra đời có 12 bit nhưng giá thành rẻ hơn PDP-1 rất nhiều (16.000 USD so với 120.000 USD). PDP-8 có một đổi mới đó là hình thành một bus đơn gọi là omnibus trong đó bus là tập hợp các dây nối song song dùng để kết nối các thành phần của máy tính.

Trong khi đó, IBM xây dựng một phiên bản của 709 bằng transistor, đó là máy tính 7094 có thời gian một chu kỳ là 2 μ s và bộ nhớ 32K từ 36 bit. Năm 1964, công ty CDC giới thiệu máy 6600 có tốc độ nhanh hơn 7094 do bên trong CPU có một cơ chế song song. CPU có vài đơn vị thực hiện phép cộng, các đơn vị khác thực hiện phép nhân, chia và tất cả chúng đều hoạt động song song. Với một công việc, máy có khả năng thực thi 10 lệnh đồng thời.



Hình 1.5. Omnibus của PDP-8

1.3.4. Máy tính IC – thế hệ thứ ba

Vi mạch được phát minh cho phép đặt vài chục transistor trong một chip đơn. Việc này giúp cho các máy tính xây dựng trên IC nhỏ hơn, nhanh hơn và rẻ hơn so với các máy tính transistor. Lúc này, IBM giới thiệu một sản phẩm đơn, máy System 360, được thiết kế dựa trên các vi mạch. Điểm mới quan trọng trong 360 là khả năng đa lập trình (multiprogramming), có vài chương trình trong bộ nhớ đồng thời để khi một chương trình đang chờ xuất / nhập dữ liệu thì chương trình khác có thể tính toán. Một đặc trưng khác của 360 là không gian địa chỉ lớn (thời điểm lúc đó), với 224 byte nhớ (16 MB).

1.3.5. Máy tính cá nhân và VLSI – thế hệ thứ tư

Vào thập niên 80, vi mạch VLSI (Very Large Scale Integrate) có khả năng chứa vài chục ngàn, vài trăm ngàn và vài triệu transistor trên một chip đơn đã được chế tạo. Sự phát triển này dẫn đến việc sản xuất các máy tính nhỏ hơn và nhanh hơn. Do đó, giá cả đã giảm xuống đến mức một cá nhân có thể sở hữu một máy tính. Các máy tính trong thế hệ này có thể chia thành 5 loại: máy tính cá nhân, máy tính mini, siêu máy tính mini, mainframe, siêu máy tính.



Hình 1.6. Máy tính IBM PC 5050 (1981)



Hình 1.7. Siêu máy tính IBM Blue Gene/P (2007)