**Định nghĩa**

• Kiến trúc máy tính(Computer architecture): nghiên cứu những thuộc tính của một hệ thống mà người lập trình có thể nhìn thấy được, những thuộc tính quyết định trực tiếp đến việc thực thi một chương trình tính toán, xử lý dữ liệu.

• Cấu trúc máy tính(Computer organization): nghiên cứu về các thành phần chức năng và sự kết nối giữa chúng để tạo nên một máy tính, nhằm thực hiện những chức năng và tính năng kỹ thuật của kiến trúc.

**Định nghĩa**

• Các thuộc tính liên quan đến kiến trúc:

• Tập lệnh cơ bản CPU có thể thực hiện

• Số bit được sử dụng để biểu diễn các loại dữ liệu khác nhau

• Cơ chế nhập/xuất dữ liệu

• Các kỹ thuật đánh địa chỉ ô nhớ

**Các khối chức năng cơ bản của máy tính**

Đơn vị xử lý trung tâm (CPU): Khối điều khiển và xử lý dữ liệu

• Bộ nhớ (Memory): Khối lưu trữ dữ liệu

• Thiết bị nhập xuất (I/O): Khối chức năng cung cấp dữ liệu cho máy tính xử lý, hoặc phản ánh dữ liệu đã được xử lý do máy tính cung cấp

• Kênh liên kết hệ thống (Bus): Các kênh truyền dẫn cung cấp sự liên lạc và trao đổi dữ liệu giữa các khối trên

**Kiến trúc máy tính Von Neumann**

• Có hệ thống phần cứng cơ bản: một CPU, một hệ thống bộ nhớ chính, một hệ thống I/O

• Lệnh (Instruction) và dữ liệu(Data) phải được lưu giữ trong một bộ nhớ ghi/đọc được.

• Từng ô nhớ trong bộ nhớ phải được định vị bằng địa chỉ. Sự định địa chỉ là tuần tự và không phụ thuộc vào nội dung của từng ô nhớ.

• Các lệnh được thực hiện tuần tự. CPU thi hành một tác vụ tại một thời điểm.

• Có một đường nối giữa bộ nhớ chính và đơn vị điều khiển của CPU, thường gọi là “cổ chai Von Neumann)

**Đánh giá kiến trúc máy tính**

• Tính phổ biến (generality)

• Tính ứng dụng (applicability)

• Tính hiệu quả (efficiency)

• Tính dễ dùng (ease of use)

• Tính dễ khiến (malleability)

• Tính dễ mở rộng (expandability)

**Biểu diễn số thực dấu chấm động**

• Tổng quát:

• S: phần định trị không dấu (Significand)

• E: phần số mũ (Exponent)

• B: cơ số (Base) – không cần lưu

• Đối với hệ nhị phân: ±𝟏. 𝐅 ∗ 𝟐�

**Ví dụ**

• Biểu diễn số thực sau theo dạng số chấm động 32 bit:

• X = -5.75

• Bước 1: Đổi X sang hệ nhị phân

• X = -5.7510 = -101.112

• Bước 2: Chuẩn hóa theo dạng ±1.F \* 2E

• X = -5.75 = -101.11 = -1.0111 \* 22

• Bước 3: Biểu diễn Floating Point

• Số âm: bit dấu Sign = 1

• Số mũ E = 2, phần mũ exponent với số thừa K=127 được biểu diễn:

• Exponent = E + 127 = 2 + 127 = 12910 = 1000 00012

• Phần định trị: 0111 0000 0000 0000 0000 000 (Thêm 19 số 0 cho đủ 23 bit)

• Kết quả: 1 1000 0001 0111 0000 0000 0000 0000 000

**Yêu cầu cơ bản của CPU**

• Có kiến trúc hợp lý nhất để thoả mãn những yêu cầu tối thiểu trong chức năng xử lý dữ liệu.

• Có đầy đủ các tính năng để xây dựng một máy tính đáp ứng yêu cầu sử dụng.

• Có khả năng ứng dụng được trong thực tế.

**Chức năng của CPU**

• CPU phải thực hiện được các phép tính số học và phép tính lôgic cơ bản

• Cộng (addition), Trừ (subtraction)

• VÀ logic (AND), HOẶC logic (OR), đảo giá trị (NOT) và các lệnh vào/ra dữ liệu (INP, OUT)

• Để thực hiện các phép tính CPU cần có ALU (Arithmetic – Logic Unit) và các thanh ghi (Register)

**Tập lệnh(Instruction Sets): Đặc điểm và chức năng**

• Các thành phần của chỉ thị máy (Machine Instruction)

• Toán tử (Operation code)

• Toán hạng nguồn (Source operand reference)

• Toán hạng kết quả (Result operand reference)

• Lệnh tiếp theo (Next instruction reference)

**Kiến trúc ALU**

• ALU là thành phần của máy tính chịu trách nhiệm thực hiện các phép toán số học và logic trên các dữ liệu được cung cấp.

• Các thành phần khác (bộ điều khiển, các thanh ghi, bộ nhớ, thiết bị vào ra) cung cấp dữ liệu để ALU xử lý và nhận dữ liệu đã xử lý từ ALU

• ALU thực chất là tổ hợp các vi mạch được thiết kế để lưu trữ dữ liệu và thực hiện các tính toán logic đơn giản.

**Kiến trúc ALU**

• Kiến trúc bên trong của ALU cần có các thành phần chính

• Các mạch số học và logic: Thực hiện các phép toán số học và logic trên dữ liệu được cung cấp.

• Mạch lấy phần bù (Complemeter): Lấy phần bù 2 của một số được lưu trong một thanh ghi để hỗ trợ cho việc tính toán.

• Mạch dịch (Shifter): Dịch chuyển các bit được lưu trong một thanh ghi sang trái hoặc sang phải một số bit xác định trước để hỗ trợ cho việc tính toán.

• Thanh ghi cờ: (Status Flag): Lưu các trạng thái xử lý dữ liệu (thành công hay không thành công, lý do...) để thông báo cho đơn vị điều khiển.

**Tổ chức thanh ghi trong CPU họ x86**

• Nhóm thanh ghi đoạn (Segment Register)

• Thanh ghi đoạn mã lệnh CS(Code Segment)

• Thanh ghi đoạn ngăn xếp SS (Stack Segment)

• Thanh ghi đoạn mở rộng ES (Extra Segment)

• Thanh ghi đoạn dữ liệu DS (Data Segment).

• Nhóm thanh ghi đa dụng (General Register)

• Thanh ghi 16 bit (AX, BX, CX, DX)

• Hoặc hai thanh ghi 8 bit (AH, AL, BH, BL, CH, CL, DH, DL)

• Từ CPU 80386, các thanh ghi này có thể kéo dài đến 32 bit, tạo thành các thanh ghi EAX, EBX, ECX, EDX

**Assembler**

• Trình biên dịch hợp ngữ → ngôn ngữ máy

• Một bộ vi xử lý (đi kèm 1 bộ lệnh xác định) có thể có nhiều Assembler của nhiều nhà cung cấp khác nhau chạy trên các OS khác nhau

• Ví dụ: Cùng là kiến trúc x86, nhưng có thể dùng A86, GAS, TASM, MASM, NASM

• Assembly program phụ thuộc vào Assembler mà nó sử dụng (do các mở rộng, đặc điểm khác nhau giữa các Assembler)

**Thanh ghi cờ**

**1) Cờ gửi (CF)**

• Có giá trị 1 khi có nhớ hoặc mượn từ bit MSB (tràn không dấu) trong phép cộng hoặc trừ, ngược lại là 0 • Có thể thay đổi theo lệnh dịch hoặc quay

**2) Cờ chẵn lẻ (PF)**

• Có giá trị 1 khi byte thấp của kết quả là chẵn

• Có giá trị 0 khi byte thấp là lẻ

• Một từ gọi là chẵn/lẻ khi số bit 1 của từ là chẵn/lẻ

• Ví dụ: sau khi thực hiện một lệnh kết quả chứa trong AL là 11010010b thì PF=1

**3) Cờ phụ (AF)**

• Có giá trị 1 khi có nhớ hoặc mượn từ 3 bit (tức có nhớ hoặc mượn từ 4 bit thấp), ngược lại bằng 0 • Dùng trong các lệnh với số BCD

**4) Cờ không (ZF)**

• Có giá trị 1 khi kết quả bằng 0

• Có giá trị 0 khi kết quả khác 0

**5) Cờ dấu (SF)**

• Có giá trị 1 khi kết quả là âm (bit MSB là 1)

• Có giá trị 0 khi kết quả là dương (bit MSB là 0)

**6) Cờ tràn (OF)**

• Có giá trị 1 khi xảy ra trạng thái tràn tức giá trị (có dấu) vượt quá phạm vi giá trị cho phép

• Ví dụ: AX=BX=7FFFh=32767, nếu thực hiện lệnh ADD AX, BX thì OF=1

• CPU đặt OF=1 theo qui tắc: “nhớ ra và vào MSB xảy ra không đồng thời”

**Ví dụ**

**1**

MOV AX,

0FFFFh MOV BX,

0FFFFh

ADD AX, BX

**;AX=FFFEh**

**;SF=1 vì MSB=1**

**;PF=0 vì byte thấp kết quả là lẻ**

**;ZF=0 vì kết quả khác không**

**;CF=1 vì có nhớ**

**;OF=0 vì có nhớ vào và ra MSB**

**2**

MOV AL,

80h MOV BL,

80h ADD AL, BL

**;AL=0**

**;SF=0 vì MSB=0**

**;PF=1 vì byte kết quả là chẵn**

**;ZF=1 vì kết quả bằng không**

**;CF=1 vì có nhớ**

**;OF=1 vì có nhớ ra nhưng không nhớ vào MSB**

**3**

MOV AX,

8000h MOV BX,

1 SUB AX, BX

**;AX=7FFFh**

**;SF=0 vì MSB=0**

**;PF=1 vì byte kết quả là chẵn**

**;ZF=0 vì kết quả khác không**

**;CF=0 vì không nhớ**

**;OF=1 vì có mượn nhưng không trả từ MSB (trừ âm cho dương nhưng kết quả lại dương)**