CHƯƠNG 2: KIẾN TRÚC BỘ XỬ LÝ

TS. Trần Ngô Như Khánh

Yêu cầu cơ bản của CPU

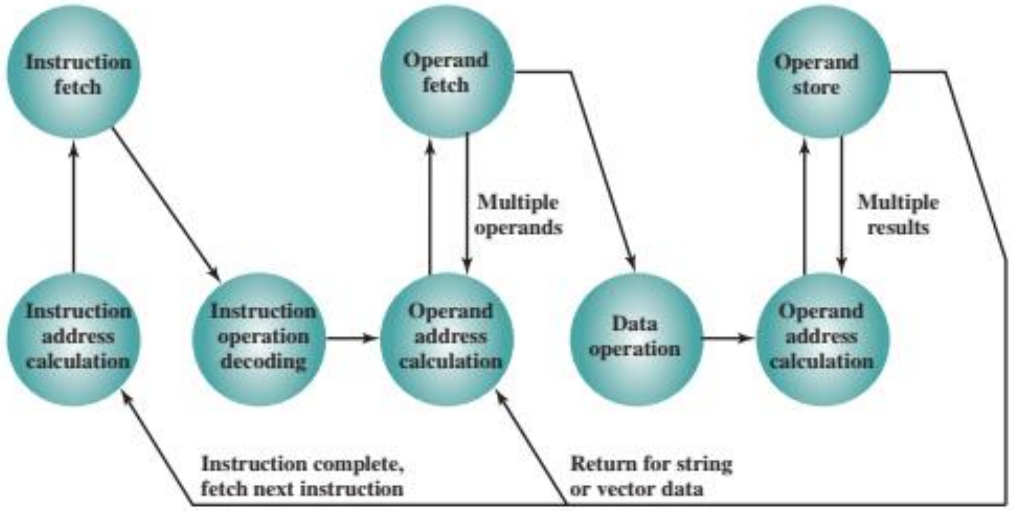
- Có kiến trúc hợp lý nhất để thoả mãn những yêu cầu tối thiểu trong chức năng xử lý dữ liệu.
- Có đầy đủ các tính năng để xây dựng một máy tính đáp ứng yêu cầu sử dụng.
- Có khả năng ứng dụng được trong thực tế.

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

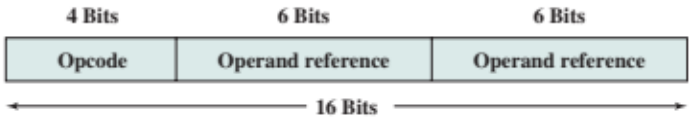
- Các thành phần của chỉ thị máy (Machine Instruction)
  - Toán tử (Operation code)
  - Toán hạng nguồn (Source operand reference)
  - Toán hạng kết quả (Result operand reference)
  - Lệnh tiếp theo (Next instruction reference)



Sơ đồ chu kỳ thực thi lệnh

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

- Biểu diễn lệnh:



  - Opcodes

ADD	Add
SUB	Subtract
MUL	Multiply
DIV	Divide
LOAD	Load data from memory
STOR	Store data to memory

  - Operand: Dùng kí hiệu (ví dụ A,R,…)
- Loại lệnh:
  - Data proccesing
  - Data storage
  - Data movement
  - Control

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

- Số lượng địa chỉ
  - Ví dụ thực hiện phép tính $Y = \frac{A - B}{C + (D \times E)}$ với số lượng địa chỉ trong lệnh khác nhau

Instruction		Comment
SUB	Y, A, B	Y ← A − B
MPY	T, D, E	T ← D × E
ADD	T, T, C	T ← T + C
DIV	Y, Y, T	Y ← Y ÷ T

Lệnh 3 địa chỉ

Instruction		Comment
MOVE	Y, A	Y ← A
SUB	Y, B	Y ← Y − B
MOVE	T, D	T ← D
MPY	T, E	T ← T × E
ADD	T, C	T ← T + C
DIV	Y, T	Y ← Y ÷ T

Lệnh 2 địa chỉ

Instruction		Comment
LOAD	D	AC ← D
MPY	E	AC ← AC × E
ADD	C	AC ← AC + C
STOR	Y	Y ← AC
LOAD	A	AC ← A
SUB	B	AC ← AC − B
DIV	Y	AC ← AC ÷ Y
STOR	Y	Y ← AC

Lệnh 1 địa chỉ

Tập lệnh(Instruction Sets): Các chế độ địa chỉ

- Các chế độ địa chỉ (Address mode)
  - Immediate
  - Direct
  - Indirect
  - Register
  - Register indirect
  - Displacement
  - Stack

Instruction Set Architecture(ISA)

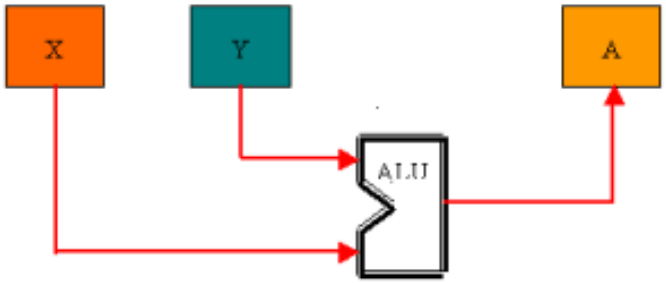
- Tập lệnh dành cho những bộ vi xử lý có kiến trúc tương tự nhau
- Một số ISA thông dụng:
  - Dòng vi xử lý 80x86 (gọi tắt x86) của Intel
    - IA-16: Dòng xử lý 16 bit (Intel 8086, 80186, 80286)
    - IA-32: Dòng xử lý 32 bit (Intel 80368 – i386, 80486 – i486, PentiumII, Pentium III …)
    - IA-64: Dòng xử lý 64 bit (Intel x86-64 như Pentium D…)
  - MIPS: Dùng nhiều trong các hệ thống nhúng
  - IBM PowerPC

Thiết kế ISA: CISC & RISC

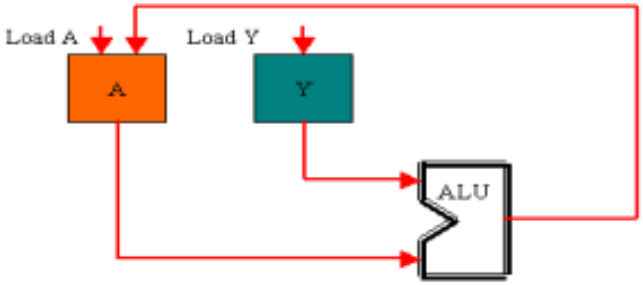
- Có 2 kiểu thiết kế bộ lệnh
  - Complete Instruction Set Computer (CISC): bộ lệnh gồm rất nhiều lệnh, từ đơn giản đến phức tạp
  - Reduced Instruction Set Computer (RISC): bộ lệnh chỉ gồm các lệnh đơn giản
- Nên chọn kiểu nào?

Kiến trúc ALU

- ALU với đầu vào X, Y và đầu ra A

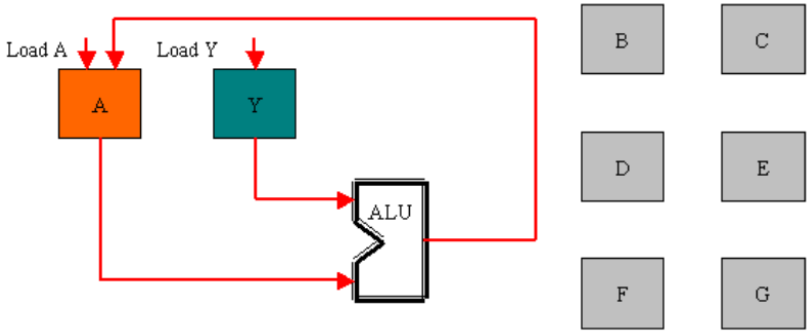


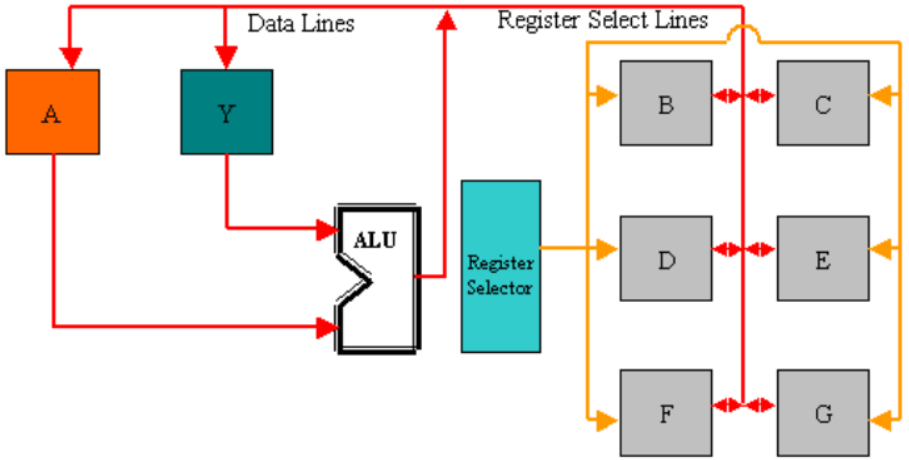
- ALU với đầu vào A, Y và đầu ra A



Kiến trúc ALU

- ALU với bộ nhớ trong (hoặc các thanh ghi)



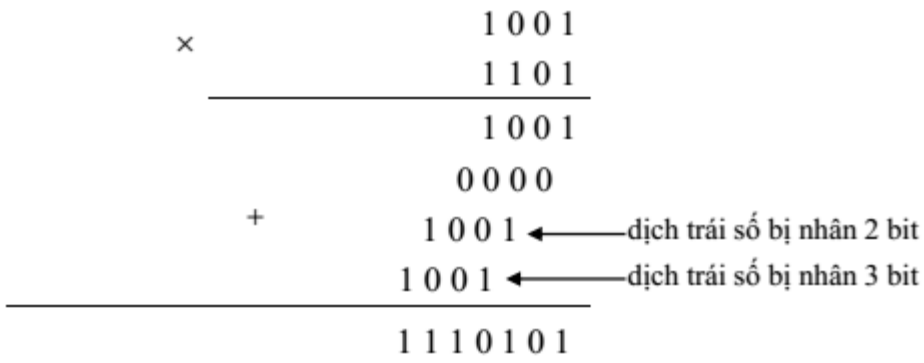


Kiến trúc ALU

- 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

- Thành phần quan trọng nhất của ALU là các bộ cộng số học và bộ cộng logic.
- Xét ví dụ thực hiện phép tính nhân



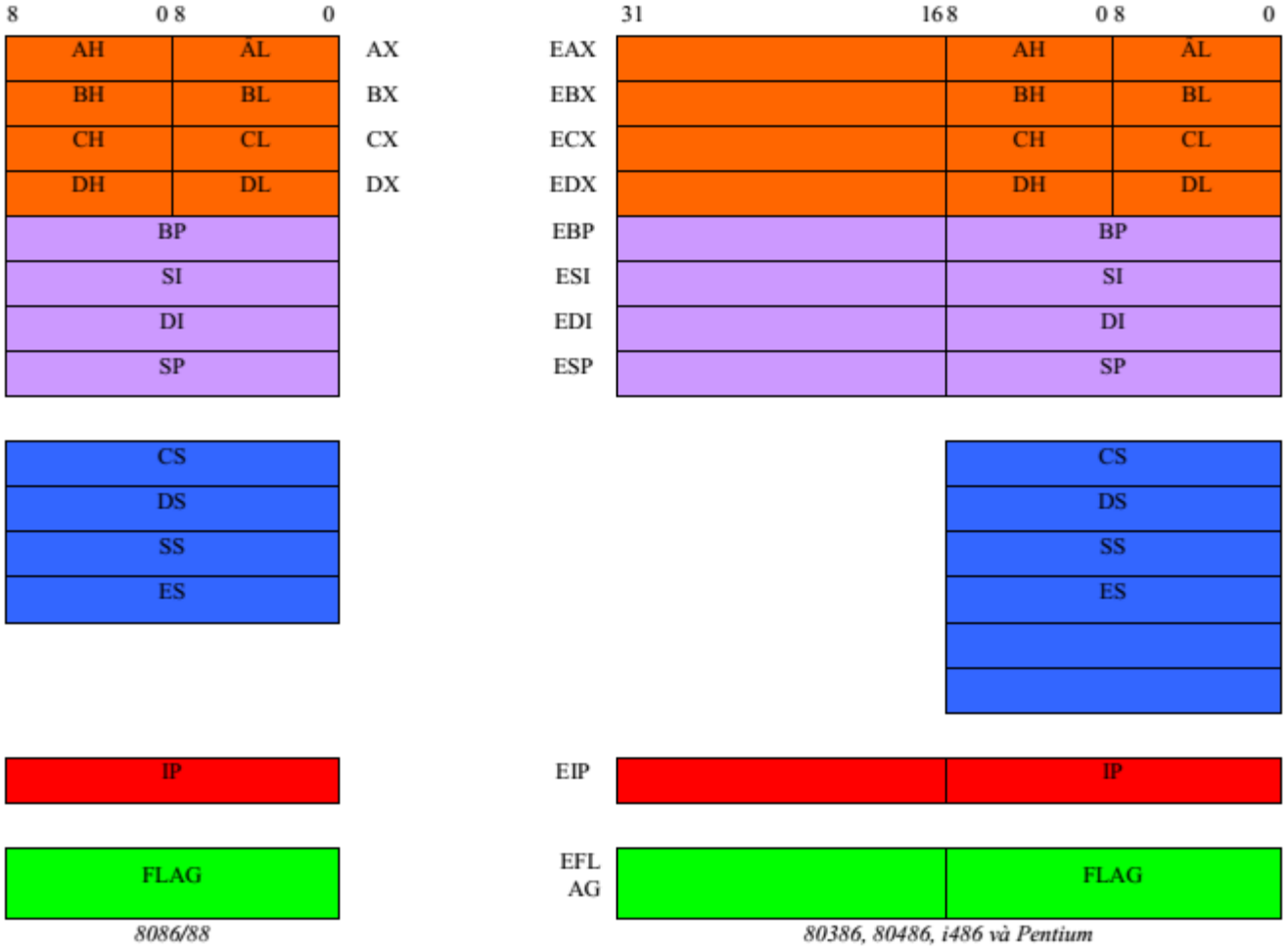
Các thao tác cần thực hiện:

- Đưa dữ liệu vào thanh ghi, nếu nhân với 1; hoặc nạp 0 nếu nhân với 0;
- Dịch chuyển dữ liệu sang trái một số bit;
- Cộng dồn các dữ liệu trong từng bước

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



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

Tổ chức thanh ghi trong CPU họ x86

- Nhóm thanh ghi con trỏ và chỉ số (Pointer and Index Register)
  - Gồm 4 thanh ghi
  - Chỉ số nguồn (SI), chỉ số đích (DI)
  - Con trỏ ngăn xếp (SP)
  - Con trỏ cơ sở (BP)
  - SI và DI chứa địa chỉ ô (offset) tương ứng với địa chỉ đoạn chứa trong DS hoặc ES
  - SP và BP chứa địa chỉ ô (offset) tương ứng với địa chỉ đoạn chứa trong DS hoặc ES
  - Từ CPU 80386, các thanh ghi này có thể kéo dài đến 32 bit tạo thành các thanh ghi ESI, EDI, ESP, EBP.

Tổ chức thanh ghi trong CPU họ x86

- Nhóm thanh ghi cờ và con trỏ lệnh
  - Thanh ghi cờ (Flag Register) dài 16 bit ghi nhận các thông tin về điều khiển và trạng thái của CPU
    - Cờ trạng thái: CF (carry flag), PF (parity flag) , AF (auxiliary flag), ZF (zero flag) , SF (Sign flag), OF (Overflow flag).
    - Cờ điều khiển: DF (direction flag), IF (Interrupt enable flag), TF (Trap flag).
  - Thanh ghi lệnh (IP-Instruction Pointer) dài 16 bit chứa địa chỉ ô của lệnh tiếp theo
    - Địa chỉ đoạn ứng với IP là CS
  - Từ CPU 80386, các thanh ghi này có thể kéo dài đến 32 bit tạo thành các thanh ghi EFLAGS và EIP

Tổ chức bộ nhớ

- Chế độ thực
  - 16 bit (8086)
  - Truy cập 1Mb bộ nhớ
  - Hệ điều hành: DOS
- Chế độ bảo vệ:
  - 32 bit
  - Truy cập 4Gb bộ nhớ
  - Hệ điều hành: Windows, Linux
- Chế độ quản lý hệ thống
  - Quản lý nguồn cung cấp
  - Chẩn lỗi và bảo mật hệ thống

Tổ chức bộ nhớ chế độ thực

- Việc truy cập bộ nhớ phải theo từng đoạn (segment)
  - Mỗi đoạn bắt đầu từ địa chỉ chia chẵn cho 16.
  - Mỗi đoạn có độ lớn 64 Kb
- Địa chỉ một byte trong bộ nhớ được xác định bằng giá trị Đoạn:Ô (Segment:Offset)
- Đoạn hoặc ô là một số nhị phân 16 bit có giá trị từ 0000h đến FFFFh
  - Ví dụ Địa chỉ logic của một ô nhớ: 1123:0001
- Cách xác định địa chỉ vật lý

$$\textit{Địa chỉ vật lý} = (\textit{Segment}) \times 10H + (\textit{offset})$$

Tổ chức bộ nhớ chế độ thực