LẬP TRÌNH HỢP NGỮ

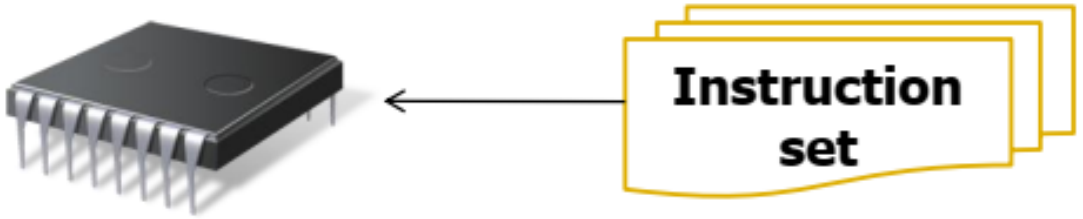
TS. Trần Ngô Như Khánh

Ngôn ngữ lập trình

- Là ngôn ngữ nhân tạo, được chuẩn hóa và thiết kế theo một hệ thống nguyên tắc riêng để truyền các chỉ thị các máy tính có bộ xử lý (CPU)
- Cấu thành bởi 2 yếu tố Từ vựng và Ngữ pháp
  - Ví dụ C/C++
    - Keyword: struct, enum, if, int…
    - Sytax: (if(…){} else{}, do{} while()…
- Độ phức tạp (trừu tượng) của cách truyền chỉ thị quyết định thứ bậc của ngôn ngữ
  - Độ phức tạp càng cao thì bậc càng thấp

Ngôn ngữ máy (Machine Language)

- Là một tập các chỉ thị (instruction) được CPU của máy tính trực tiếp thực thi
- Mỗi bộ vi xử lý (CPU) có 1 ngôn ngữ riêng, gọi là bộ lệnh (instruction set)
- Trong cùng môt dòng vi xử lý (processor family) bộ lệnh gần giống nhau



Hợp ngữ

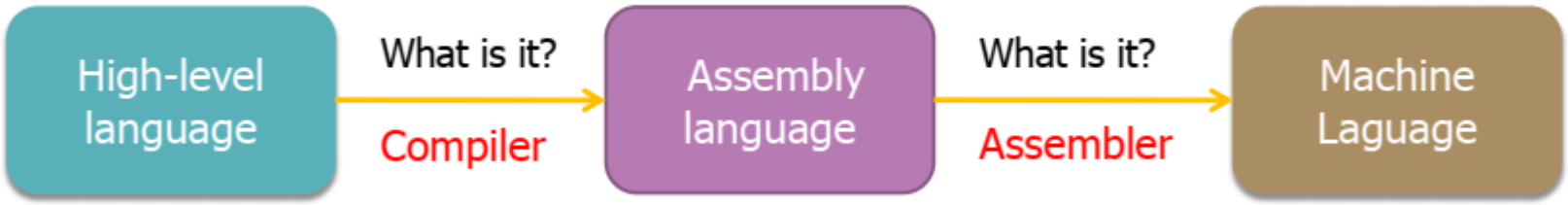
- Các mã máy chỉ là các con số (0 / 1)
- Trong ngôn ngữ máy không có khái niệm biến → thay vào đó là địa chỉ ô nhớ, thanh ghi (lưu trữ mã lệnh, dữ liệu)
- Hợp ngữ rất gần với ngôn ngữ máy nhưng lại đủ để con người hiểu và sử dụng tốt hơn ngôn ngữ máy
  - Ví dụ: Gán giá trị 10h cho thanh ghi AX

    Ngôn ngữ máy: 10111000 00010000 00000000

    Hợp ngữ: MOV AX, 10h
- Vì mỗi bộ vi xử lý có 1 cấu trúc thanh ghi và tập lệnh riêng nên khi lập trình hợp ngữ phải nói rõ là lập trình cho bộ vi xử lý nào, hay dòng (family) vi xử lý nào
  - Hợp ngữ MIPS
  - Hợp ngữ dòng vi xử lý Intel 80x86

Complier

- Trình biên dịch ngôn ngữ cấp cao → hợp ngữ
- Compiler phụ thuộc vào:
  - Ngôn ngữ cấp cao được biên dịch
  - Kiến trúc hệ thống phần cứng bên dưới



Assembler

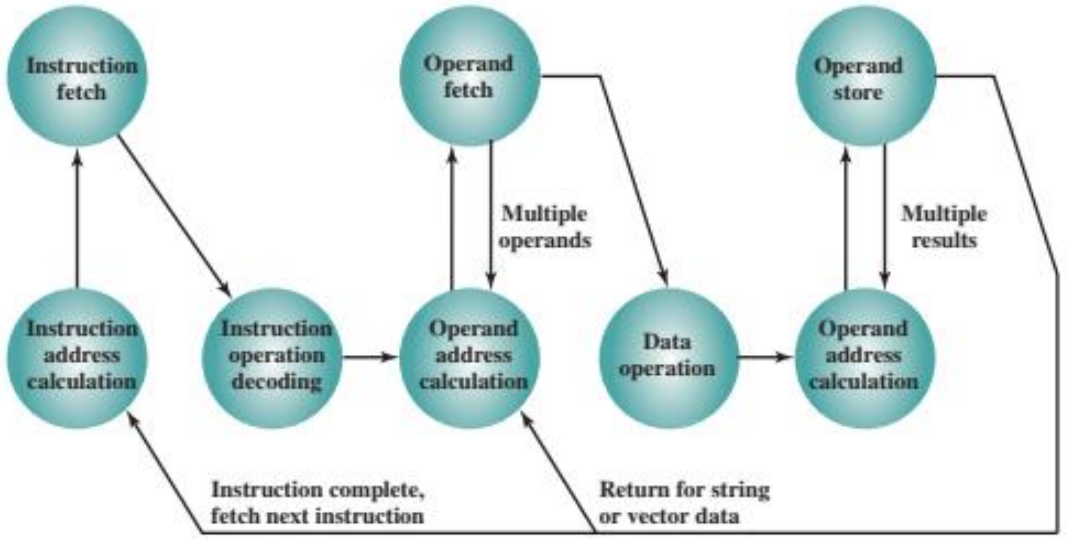
- Trình biên dịch hợp ngữ → ngôn ngữ máy
- Một bộ vi xử lý (đi kèm 1 bộ lệnh xác định) có thể có nhiều Assembler của nhiều nhà cung cấp khác nhau chạy trên các OS khác nhau
  - Ví dụ: Cùng là kiến trúc x86, nhưng có thể dùng A86, GAS, TASM, MASM, NASM
- Assembly program phụ thuộc vào Assembler mà nó sử dụng (do các mở rộng, đặc điểm khác nhau giữa các Assembler)

Hoạt động Xử lý lệnh của CPU

- Gồm 2 bước:
  - Nạp lệnh (Fetch): Di chuyển lệnh từ memory vào thanh ghi (register) trong CPU
  - Thực thi lệnh (Excute): Giải mã lệnh và thực thi thao tác yêu cầu

Quy trình thực thi lệnh

- Gọi là Instruction Cycle – vòng lặp xử lý lệnh , gồm các bước được lặp đi lặp lại cho tất cả các lệnh tiếp theo



- Tính địa chỉ lệnh
- Nạp lệnh
- Giải mã lệnh
- Tính địa chỉ của toán hạng
- Nạp toán hạng
- Thực hiện lệnh
- Tính địa chỉ của toán hạng chứa kết quả
- Ghi kết quả

LẬP TRÌNH HỢP NGỮ:

CÁC KIỂU ĐỊNH VỊ & THANH GHI CỜ

Các kiểu định vị

1) Định vị thanh ghi
2) Định vị lấy ngay (tức thì)
3) Định vị trực tiếp
4) Định vị gián tiếp thanh ghi
5) Định vị chỉ số
6) Định vị cơ sở
7) Định vị chỉ số cơ sở

1) Định vị thanh ghi

- Toán hạng là các thanh ghi, tức dữ liệu chứa trong thanh ghi.
- Lệnh được thực hiện nhanh hơn vì không mất thời gian truy cập bộ nhớ
- Ví dụ
  - `MOV AL, BL` `;AL← BL`
  - `INC BX` `;BX ← BX+1`
  - `DEC AL` `;AL ← AL -1`
  - `SUB DX,CX` `;DX ← DX-CX`

2) Định vị lấy ngay

- Dữ liệu xử lý được lưu ngay trong lệnh
- Ví dụ
  - `MOV CL, 61h ;CL← 61h`
- Mã máy trong ví dụ trên là B161 với 0B1 là mã toán tử
- Chế độ định vị này cũng được thực hiện nhanh vì dữ liệu được lấy cùng lúc với lệnh

3) Định vị trực tiếp

- Địa chỉ toán hạng được ghi trong lệnh thay vì là dữ liệu (địa chỉ trong bộ nhớ của dữ liệu chứa trong lệnh)
- Ví dụ:
  - `MOV [7000h], AX`
- Trong ví dụ trên:
  - Giả sử DS chứa trị 1000h, còn AX chứa 1234h
  - Nội dung của AX sẽ được đưa vào từ nhớ có địa chỉ là 17000h
  - Byte 17000h chứa trị 34h và byte 17001h chứa 12h

4) Định vị gián tiếp thanh ghi

- Địa chỉ ô của toán hạng chứa trong các thanh ghi BX, BP, SI hoặc DI
  - Địa chỉ đoạn chứa trong DS tương ứng với BX, SI, DI
  - SS tương ứng với BP
- Ví dụ:
  - `SUB DX, [BX] ; Trừ nội dung DX với vùng nhớ có địa chỉ chứa trong BX`

4) Định vị gián tiếp thanh ghi

- Cho bảng nội dung thanh ghi và vùng nhớ:

Thanh ghi	Nội dụng thanh ghi	Nội dung vùng nhớ	Địa chỉ vùng nhớ
BX	1000	1BAC	DS:1000
SI	2000	20FE	DS:2000
DI	3000	031D	DS:3000

- Các lệnh sau cho kết quả

```
MOV BX, [BX]        ;BX=1BAC
MOV CX, [SI]        ;CX=20FE
MOV BX, [AX]        ; Toán hạng không hợp lệ
ADD [SI],[DI]       ; Lệnh không hợp lệ
INC [DI]            ;[DS:3000]=031E
```

5) Định vị chỉ số

- Dữ liệu chứa trong bộ nhớ
- Địa chỉ toán hạng là tổng nội dung thanh ghi chỉ số SI hoặc DI với một số 8 hoặc 16 bit
- Ví dụ:
  - `MOV AX, [SI+2]`
  - `MOV AX, [2+SI]`
  - `ADD [DI-6] CX`
  - `ADD [DI]-6, CX`
  - `MOV [SI+ACCOUNT], AX`

6) Định vị cơ sở

- Tương tự định vị chỉ số nhưng sử dụng thanh ghi BX hoặc BP thay vì DI hoặc SI
- Cho bảng nội dung thanh ghi và vùng nhớ

Thanh ghi	Nội dụng thanh ghi	Nội dung vùng nhớ	Địa chỉ vùng nhớ
BX	2	1084	DS:0002
SI	4	2BAC	DS:0004
DI	1	031D	DS:3000

```
; Cho khai báo:
ALPHA DW 0123h, 0456h, 0789h, 0ABCh
```

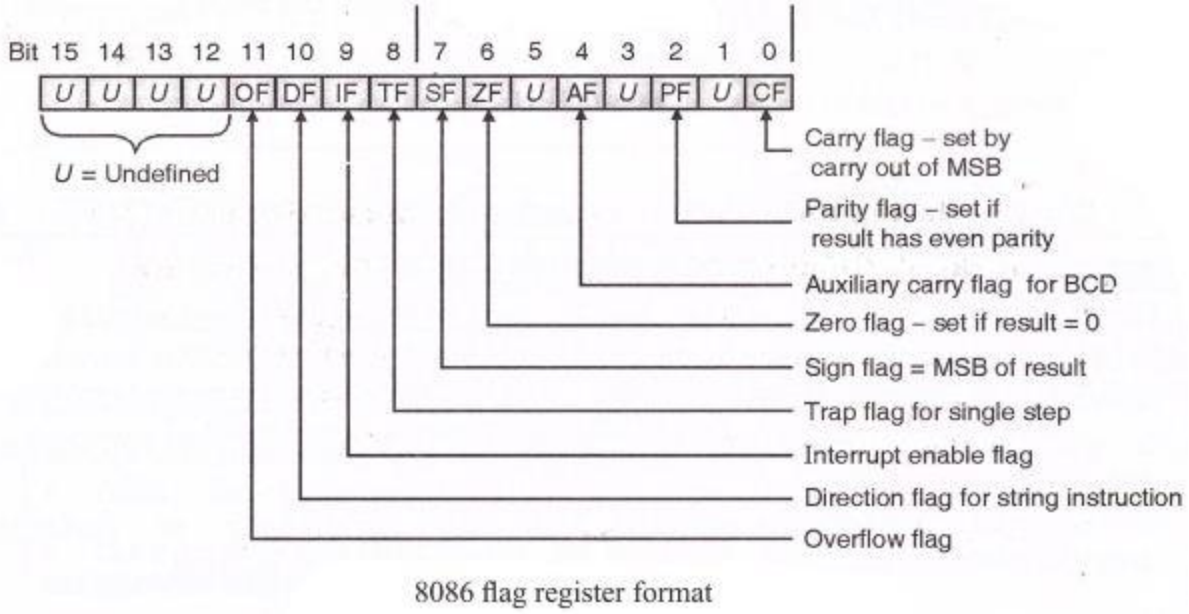
- `MOV AX, [ALPHA+BX]` ;AX=0456h
- `MOV BX, [BX+2]` ;BX=2BACh
- `MOV CX, ALPHA[SI]` ;CX=0789h
- `MOV AX, -2[SI]` ;AX=1084h
- `MOV BX, [ALPHA+3+DI]` ;BX=0789h
- `ADD BX, [ALPHA+AX]` ;Thanh ghi nguồn không hợp lệ

7) Định vị chỉ số cơ sở

- Địa chỉ toán hạng = nội dung thanh ghi cơ sở cộng với nội dung thanh ghi chỉ số và có thể cộng thêm một trị 8 hoặc 16 bit
- Ví dụ:
  - `MOV AX, [BX+DI]`
  - `MOV [BX+SI+2], AX`
  - `MOV AX, [BX+SI+2]`
  - `MOV AX,[SI+BX+2]`
  - `MOV AC, [SI][BX+2]`
  - `MOV AC, [BX][SI+2]`
  - `MOV AC, 2[BX][SI]`

Thanh ghi cờ

- Thanh ghi cờ dài 16 bit (8086) ghi nhận các thông tin điều khiển, trạng thái CPU và kết quả thực hiện lệnh sau cùng
- Mỗi bit gọi là một cờ: giá trị 1 (set), 0 (clear)
- Gồm 2 nhóm:
  - Cờ điều khiển
  - Cờ trạng thái



8086 flag register format

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

Thanh ghi cờ

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)
```

Bài tập

Xác định giá trị các cờ (CF, SF, ZF, PF và OF) và giải thích kết quả

1

```
MOV AL,0FFh
INC AL
```

2

```
MOV AX,8000h
NEG AX
```

3

```
MOV AX, 8000h
MOV BX, 1
SUB AX, BX
```

Trị đặt/xóa của một số cờ thông dụng

Cờ	Đặt	Xóa	Ý nghĩa
OF	OV	NV	Tràn / Không tràn (Overflow / Not overflow)
DF	DN	UP	Xuống / Lên (Down / Up)
IF	EI	DI	Cho ngắt / Cấm ngắt (Enable / Disable)
SF	NG	PL	Âm / Dương (Negative / Plus
ZF	ZR	NZ	Không / Khác không (Zero / Not zero)
AF	AC	NA	Gửi phụ / Không gửi phụ (Auxiliary Carry / Not auxiliary Carry)
PF	PE	PO	Chẵn / Lẻ (Parity Even / Parity Odd)
CF	CY	NC	Gửi / Không gửi (Carry yes / Not carry)