



Môn học: KIẾN TRÚC MÁY TÍNH

- Chương 1 • Tổng quan về máy tính
- Chương 2 • Biểu diễn số học trong máy tính
- Chương 3 • Hệ thống máy tính
- Chương 4 • CPU (Central Processing Unit)
- Chương 5 • Bộ nhớ máy tính (Memory)
- Chương 6 • Thiết bị giao tiếp – Thiết bị ngoại vi
- Chương 7 • Cài đặt máy tính
- Chương 8 • Sao lưu và phục hồi

Chương 4: Bộ xử lý trung tâm (CPU)

Cấu trúc cơ bản của CPU

Tập thanh ghi (Register File)

Giới thiệu tập lệnh của CPU

Các thông số đặc trưng của CPU

Một số công nghệ nổi bật của CPU

Các dòng sản phẩm Intel CPU

FACULTY OF INFORMATION TECHNOLOGY

Giới thiệu CPU

- **CPU (Central Processing Unit) - đơn vị xử lý trung tâm.**
 - ▶ CPU có thể được xem như não bộ, một trong những phần tử cốt lõi nhất của máy vi tính. Nhiệm vụ chính của CPU là xử lý các chương trình và dữ liệu.
 - ▶ CPU là một con chip với vài chục chân. Phức tạp hơn, CPU được ráp sẵn trong các bộ mạch với hàng trăm con chip khác.



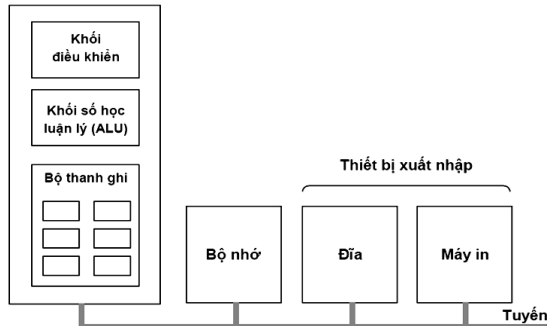
FACULTY OF INFORMATION TECHNOLOGY

- 4 -

Cấu trúc cơ bản của CPU

■ Các khối cơ bản bên trong CPU:

- *Arithmetic and Logic Unit (ALU)*: thực các phép toán và logic.
- *Control Unit (CU)*: điều khiển và thực thi lệnh
- **Registers**: bộ thanh ghi. Dùng chứa lệnh và dữ liệu cần thực thi
- **Bus**: các đường truyền dẫn tín hiệu (lệnh và dữ liệu)



FACULTY OF INFORMATION TECHNOLOGY

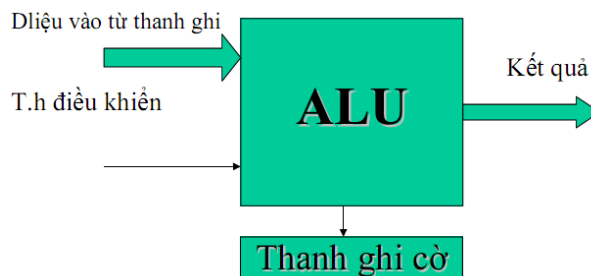
- 5 -

Cấu trúc cơ bản của CPU

■ *Arithmetic and Logic Unit (ALU)*:

- Thực thi các phép toán số học và logic
- **Số học**: cộng, trừ, nhân, chia, tăng, giảm, đảo,...
- **Logic**: AND, OR, XOR, NOT, dịch bit,...

■ Mô hình kết nối của ALU



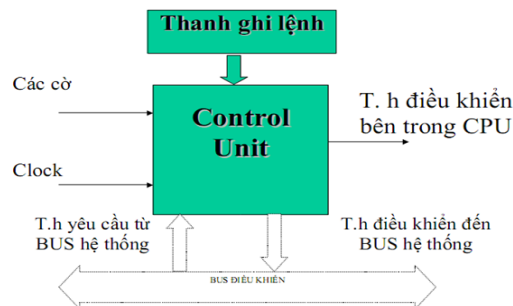
FACULTY OF INFORMATION TECHNOLOGY

- 6 -

Cấu trúc cơ bản của CPU

■ **Control Unit (CU):**

- Nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh.
- Tăng nội dung thanh ghi PC mỗi khi nhận lệnh xong
- Giải mã lệnh và xác định thao tác mà lệnh yêu cầu
- Phát ra tín hiệu điều khiển thực thi lệnh.
- Nhận các tín hiệu yêu cầu từ BUS hệ thống và giải quyết đáp ứng yêu cầu đó.



FACULTY OF INFORMATION TECHNOLOGY

- 7 -

Cấu trúc cơ bản của CPU

■ **Các thông tin kết nối đến CU:**

- **Clock**: tín hiệu xung nhịp từ mạch tạo dao động.
- **Mã lệnh** từ thanh ghi lệnh đưa đến CU giải mã
- Các **trạng thái cờ** đưa đến cho biết trạng thái của CPU cũng như trạng thái thực hiện các phép toán trong ALU.
- Các tín hiệu **điều khiển từ BUS** điều khiển.
- Các tín hiệu **điều khiển bên trong CPU**: điều khiển thanh ghi, ALU.
- Các tín hiệu **điều khiển bên ngoài CPU** đó là Bộ nhớ hay cổng vào ra

FACULTY OF INFORMATION TECHNOLOGY

- 8 -

Tập thanh ghi (Register File)

■ Tập các thanh ghi (**Register File** - RF):

- ▶ Registers thực chất là vùng nhớ có tốc độ truy xuất cực nhanh, được đặt bên trong CPU, nhận biết qua *tên của thanh ghi* (ví dụ: thanh ghi AX, BX, CX...).
- ▶ Chứa thông tin tạm thời phục vụ cho hoạt động ở thời điểm hiện tại của CPU
- ▶ Số lượng thanh ghi tùy thuộc vào bộ vi xử lý cụ thể -> tăng hiệu năng CPU

■ Phân loại thanh ghi:

- ▶ Thanh ghi đa dụng (*General Register*).
- ▶ Thanh ghi đoạn (*Segment Register*).
- ▶ Thanh ghi con trỏ và chỉ số (*Pointer & Index register*).
- ▶ Thanh ghi Con trỏ lệnh (Đếm chương trình) và thanh ghi trạng thái (Cờ).

Tập thanh ghi (Register File)

■ Thanh ghi đa dụng (**General Register**).

- ▶ **AX** (*Accumulator*): Là thanh ghi tích lũy, số liệu tức thời, chứa kết quả tác vụ tính toán.
- ▶ **BX** (*Base address*): Thanh ghi nền, dùng để chứa địa chỉ ô nhớ.
- ▶ **CX** (*Count*): Là thanh ghi đếm, dùng để đếm số lần vòng lặp hoặc lệnh xử lý chuỗi ký tự.
- ▶ **DX** (*Data*): Thanh ghi dữ liệu, thường chứa giá trị địa chỉ của một số lệnh vào/ra, lệnh tính toán số học (kể cả lệnh nhân và chia).
- ▶ Với hệ xử lý 16 bit, mỗi thanh ghi có thể chia thành 2 thanh ghi 8 bit (*Low* và *High*). Ví dụ như:

AX = **AH** & **AL**;

BX = **BH** & **BL**;

CX = **CH** & **CL**;

DX = **DH** & **DL**.

Tập thanh ghi (Register File)

■ Thanh ghi đoạn (*Segment Register*):

- ▶ Dùng chứa địa chỉ đoạn của các ô nhớ - chuyển đổi địa chỉ tương đối thành địa chỉ tuyệt đối để truy xuất vào ô nhớ.
- ▶ Mỗi thanh ghi đoạn quản lý 1 đoạn tối đa 64K (2^{16} địa chỉ).
- ▶ **CS** (*Code Segment*): Thanh ghi đoạn mã lệnh, lưu địa chỉ đoạn chứa mã lệnh chương trình của người sử dụng
- ▶ **DS** (*Data Segment*): Thanh ghi đoạn dữ liệu, lưu địa chỉ đoạn chứa dữ liệu (các biến) trong chương trình.
- ▶ **ES** (*Extra data Segment*): Thanh ghi đoạn dữ liệu thêm, lưu địa chỉ đoạn chứa dữ liệu thêm trong chương trình.
- ▶ **SS** (*Stack Segment*): Thanh ghi đoạn ngăn xếp, lưu địa chỉ đoạn của vùng ngăn xếp.

Tập thanh ghi (Register File)

■ Thanh ghi con trỏ và chỉ số (*Pointer & Index register*):

- ▶ **SI** (*Source Index Reg*): trỏ đến ô nhớ trong đoạn dữ liệu được định địa chỉ bởi thanh ghi **DS**.
- ▶ **DI** (*Destination Index Reg*): tương tự SI.
- ▶ **BP** (*Base Pointer Reg*): con trỏ nền chỉ đến phần tử của ngăn xếp.
- ▶ **SP** (*Stack Pointer Reg*): trỏ đến vị trí đỉnh ngăn xếp.

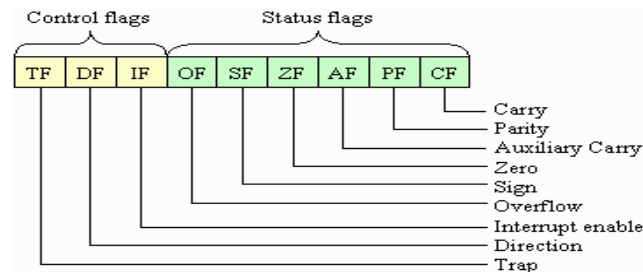
■ Thanh ghi Con trỏ lệnh (*IP - Instruction Pointer Reg*):

- ▶ Thanh ghi con trỏ lệnh **IP** (hay đếm chương trình – **PC** – *Program Counter*) là thanh ghi chứa địa chỉ của lệnh kế tiếp mà CPU sẽ thực hiện.
- ▶ Các lệnh của chương trình có địa chỉ đoạn trong CS.

Tập thanh ghi (Register File)

■ Thanh ghi cờ (*Flag Register*):

- ▶ Cờ (*flag*) là thể hiện một trạng thái tại một thời điểm trong quá trình thực thi chương trình
- ▶ Mỗi *flag* là một *bit*, biểu diễn bằng các ký hiệu gọi nhớ.
- ▶ *Flag Register* là thanh ghi **16 bit**. Vi xử lý 8088/8086 dùng **9 bit**.
 - *Status Flags*: các cờ phụ thuộc kết quả phép toán,
 - *Control Flags*: các cờ thiết lập bởi lệnh



FACULTY OF INFORMATION TECHNOLOGY

- 13 -

Tập thanh ghi (Register File)

■ Thanh ghi cờ (*Flag Register*):

- ▶ **CF** (*Carry Flag*): Cờ nhớ.
 - CF = 1 (CY - *Carry Yes*) khi kết quả phép toán giữa 2 bit có nhớ.
 - CF = 0 (NC - *No Carry*) trong trường hợp ngược lại. Cờ này thường được sử dụng khi thực hiện các phép cộng, trừ các số nhiều byte.
- ▶ **PF** (*Parity Flag*): Cờ chẵn lẻ.
 - PF = 1 (PE - *Parity Even*) khi số bit 1 trong kết quả phép toán (hay các phép vận chuyển dữ liệu) là chẵn.
 - PF = 0 (PO - *Parity Odd*) trong trường hợp ngược lại.
- ▶ **AF** (*Auxiliary Flag*): Cờ nhớ phụ (trần phụ - cờ hỗ trợ).
 - Cờ này có ý nghĩa khi ta làm việc với số BCD.
 - AF = 1 (AC - *Auxiliary Carry*) khi có nhớ hoặc mượn từ một số BCD thấp (4 bit thấp) sang một số BCD cao (4 bit cao)
 - AF = 0 (NA - *No Auxiliary carry*) trong trường hợp ngược lại.

FACULTY OF INFORMATION TECHNOLOGY

- 14 -

Tập thanh ghi (Register File)

■ Thanh ghi cờ (*Flag Register*):

- ▶ **ZF** (*Zero Flag*): Cờ rỗng.
 - ZF = 1 (ZR - *ZeRo*) khi kết quả phép toán = 0,
 - ZF = 0 (NZ - *Non Zero*) trong trường hợp ngược lại.
- ▶ **SF** (*Sign Flag*): Cờ dấu.
 - Trong bộ vi xử lý 8088/8086 các số âm được biểu diễn dưới dạng số bù 2, nên phải dùng cờ SF để chỉ thị dấu của kết quả.
 - SF = 1 (NG - *NeGative*), khi kết quả phép toán là một số **âm**,
 - SF = 0 (PL- *PLus*) khi kết quả phép toán là một số **dương**.
- ▶ **OF** (*Overflow Flag*): Cờ tràn.
 - OF = 1 (OV - *OverFlow*) khi kết quả là số bù 2 vượt khuôn khổ biểu diễn (tràn số học, hay nói cách khác: khi cộng hai số cùng dấu mà kết quả là một số trái dấu thì OF = 1),
 - OF = 0 (NV - *Non oVerflow*) trong trường hợp ngược lại (cờ này làm việc với số có dấu).

Tập thanh ghi (Register File)

■ Thanh ghi cờ (*Flag Register*):

- ▶ **IF** (*Interrupt Flag*): Cờ điều khiển ngắt.
 - IF = 1 (EI - *Enable Interrup*), CPU cho phép ngắt,
 - IF = 0 (DI - *Disable Interrup*) CPU không cho phép ngắt.
- ▶ **DF** (*Direction Flag*): Cờ hướng - Điều khiển hướng xử lý đối với thao tác chuỗi
 - DF = 1 (DN- *DowN*) thì các lệnh vận chuyển dữ liệu hay xử lý chuỗi sẽ thao tác lùi từ phải đến trái (địa chỉ cao đến địa chỉ thấp).
 - DF=0 (*UP*) trong trường hợp ngược lại (thao tác các phần tử từ địa chỉ thấp đến địa chỉ cao).
- ▶ **TF** (*Trap Flag*): Cờ bẫy.
 - TF = 1 bật chế độ CPU chạy từng lệnh. Thường dùng để gỡ rối chương trình (*debug*). Sau khi thực hiện xong mỗi lệnh, bộ vi xử lý sẽ phát ra một lệnh ngắt (INT) để có kiểm tra chương trình.

Giới thiệu tập lệnh của CPU

■ Giới thiệu tập lệnh:

- ▶ Tập lệnh là ngôn ngữ cho CPU hiểu và thực thi.
- ▶ Chương trình chạy trên máy tính là sự kết hợp các lệnh theo một cách thức nào đó nhằm làm thực hiện việc cụ thể.

■ Một lệnh máy thường có 2 phần:

Mã thao tác

Địa chỉ toán hạng

- ▶ Mã thao tác (*Operation Code: Opcode*): Mã chỉ ra thao tác mà bộ vi xử lý cần phải thực hiện.
- ▶ Địa chỉ toán hạng (*Operand Addressing*): Chỉ ra nơi chứa các toán hạng mà mã thao tác sẽ tác động. Bao gồm:
 - *Toán hạng đích*: dữ liệu đích và là nơi chứa dữ liệu kết quả thao tác.
 - *Toán hạng nguồn*: dữ liệu tương tác với đích trong thao tác
 - CPU chỉ chấp nhận lệnh khi có ít nhất 1 toán hạng là thanh ghi.

FACULTY OF INFORMATION TECHNOLOGY

- 17 -

Giới thiệu tập lệnh của CPU

■ Các lệnh chuyển dữ liệu

- ▶ Lệnh **MOV**: Sao chép dữ liệu từ toán hạng nguồn -> đích
 - Cú pháp: **Mov** *đích, nguồn*
- ▶ Các lệnh **Load** nạp dữ liệu từ bộ nhớ -> thanh ghi
 - **LES** *reg, mem* nạp byte dữ liệu từ vùng nhớ *mem* vào *reg*, byte dữ liệu kế tiếp trong vùng *mem* vào thanh ghi **ES**
 - **LDS** *reg, mem* nạp byte dữ liệu từ vùng nhớ *mem* vào *reg*, byte dữ liệu kế tiếp trong vùng *mem* vào thanh ghi **DS**
- ▶ Lệnh **XCHG**: Trao đổi nội dung của hai toán hạng cho nhau
 - Cú pháp: **XCHG** *toán_hạng1, toán_hạng2*
- ▶ Lệnh **PUSH**: Cất nội dung vào stack
 - Cú pháp: **PUSH** *nguồn*
- ▶ Lệnh **POP**: Lấy nội dung ở đỉnh *Stack* ra toán hạng đích
 - Cú pháp: **POP** *đích*

FACULTY OF INFORMATION TECHNOLOGY

- 18 -

Giới thiệu tập lệnh của CPU

■ Các lệnh tính toán:

- ▶ Lệnh **ADD**: cộng dữ liệu từ giữa toán hạng *nguồn* và *đích*. Lưu kết quả vào *đích*.
 - Cú pháp: **ADD** *đích, nguồn*
- ▶ Lệnh **SUB**: phép trừ. *Đích* là số bị trừ. *Nguồn* là số trừ. Lưu kết quả vào *đích*.
 - Cú pháp: **SUB** *đích, nguồn*
- ▶ Lệnh **MUL**: nhân dữ liệu từ giữa toán hạng *nguồn* và *đích*. Lưu kết quả vào *đích*
 - Cú pháp: **MUL** *đích, nguồn*
- ▶ Lệnh **DIV**: phép chia. *Đích* là số bị chia. *Nguồn* là số chia. Lưu kết quả vào *đích*
 - Cú pháp: **DIV** *đích, nguồn*

Giới thiệu tập lệnh của CPU

■ Các lệnh dịch bit:

- ▶ Lệnh **ShR**: dịch sang phải (*Shift Right*) *n* bit. Hủy các bit đã dịch ra khỏi thanh ghi.
 - Cú pháp: **ShR** [**Toán hạng đích**], <*n*>
- ▶ Lệnh **ShL**: dịch sang trái (*Shift Left*) *n* bit. Hủy các bit đã dịch ra khỏi thanh ghi
 - Cú pháp: **ShL** [**Toán hạng đích**], <*n*>
- ▶ Lệnh **RcR**: quay vòng sang phải (*Rotate through Carry Right*) *n* bit.
 - Cú pháp: **RcR** [**Toán hạng đích**], <*n*>
- ▶ Lệnh **RcL**: quay vòng sang trái (*Rotate through Carry Left*) *n* bit.
 - Cú pháp: **RcL** [**Toán hạng đích**], <*n*>

Giới thiệu tập lệnh của CPU

■ Các lệnh vào ra ngăn xếp (Stack)

- Lệnh **POP**: lấy dữ liệu từ ngăn xếp (*Stack Segment*) cho vào *đích*.
Cú pháp: **POP** *đích*.
- Lệnh **PUSH**: đẩy dữ liệu từ *đích* vào ngăn xếp (*Stack Segment*).
Cú pháp: **PUSH** *đích*.

■ Các lệnh Nhập / Xuất

- Lệnh **Input**
Cú pháp: **Input** *địa_chỉ_cổng_đích, nguồn*
- Lệnh **Output**
Cú pháp: **Output** *địa_chỉ_cổng_đích, nguồ*

Giới thiệu tập lệnh của CPU

■ Các lệnh điều khiển hệ thống

- Lệnh **Halt**: dừng thực hiện chương trình
- Lệnh **Wait**: tạm dừng thực hiện chương trình, lặp kiểm tra cho đến khi thoả mãn thì tiếp tục thực hiện
- Lệnh **No Operation**: không thực hiện gì cả
- Lệnh **Lock**: Cấm không cho chuyển nhượng BUS
- Lệnh **Unlock**: cho phép chuyển nhượng BUS

Giới thiệu tập lệnh của CPU

■ Các chế độ địa chỉ (*Addressing modes*):

► 1. Chế độ địa chỉ tức thời (*Immediate Addressing Mode*)

- Toán hạng đích là thanh ghi, hoặc ô nhớ - Toán hạng nguồn là 1 giá trị cụ thể (hằng số).
- Ví dụ:
`MOV AX, 04FCh` ; kết quả: **AX = 04FCh** (**AH=04h, AL=FCh**)
- Lệnh này thường được dùng để nạp dữ liệu cần thao tác vào thanh ghi nào đó.

► 2. Chế độ địa chỉ thanh ghi (*Register Addressing Mode*)

- Dùng thanh ghi như là các toán hạng chứa dữ liệu cần thao tác
- Tốc độ thực hiện lệnh rất nhanh vì CPU không mất thời gian tìm dữ liệu.
- Ví dụ:
`MOV AL, BL` ; kết quả: **AL = BL**
`MOV DS, BX` ; kết quả: **DS = BX**
`ADD AL, DL` ; kết quả: **AL = AL + DL**

FACULTY OF INFORMATION TECHNOLOGY

- 23 -

Giới thiệu tập lệnh của CPU

■ Các chế độ địa chỉ (*Addressing modes*):

► 3. Chế độ địa chỉ trực tiếp (*Direct Memory Addressing Mode*)

- Một toán hạng là địa chỉ offset của ô nhớ chứa giá trị cần thao tác. Toán hạng kia là thanh ghi (không được là ô nhớ).
- Ví dụ:
`MOV AL, [04FCh]` ; đưa vào thanh ghi **AL** nội dung ô nhớ có địa chỉ logic: **04FCh**

► 4. Chế độ địa chỉ gián tiếp qua thanh ghi (*Register Indirect Memory Addressing Mode*)

- Một toán hạng là thanh ghi chứa địa chỉ offset của ngăn nhớ chứa giá trị cần thao tác. Toán hạng kia là thanh ghi (không được là ngăn nhớ).
- Ví dụ:
`MOV [BX], AL` ; Đưa nội dung thanh ghi **AL** vào ô nhớ có địa chỉ *offset* là giá trị của thanh ghi **BX**

FACULTY OF INFORMATION TECHNOLOGY

- 24 -

Giới thiệu tập lệnh của CPU

■ Các chế độ địa chỉ (*Addressing modes*):

► 5. Chế độ địa chỉ tương đối cơ sở: (*Base Relative Addressing Mode*)

- Dùng *Base register* (thanh ghi **BX** hoặc **BP**) để chỉ đến địa chỉ khởi đầu trên vùng nhớ.
- Dùng thêm số độ dời (gọi là *offset*) để dời đến ô nhớ cần truy xuất, tính từ vị trí *Base*.
- Ví dụ 1: đã có **BX** = **82h**
`MOV [BX + 3], CL` ; Đưa nội dung của thanh ghi **CL** vào ô nhớ tại địa chỉ **85h** (*Base* = **82**, *offset* = **3**).
- Ví dụ 2: đã có **BP** = **02h**
`MOV AX, [BP + 3]` ; Đưa nội dung của phần tử thứ **5** trong ngăn xếp vào thanh ghi **AX**.
 Ghi chú: **BP** là con trỏ nền chỉ đến phần tử nào đó trong *Stack*.

Giới thiệu tập lệnh của CPU

■ Các chế độ địa chỉ (*Addressing modes*):

► 6. Chế độ địa chỉ tương đối chỉ số: (*Index Addressing Mode*)

- Dùng *Index register* (thanh ghi **SI** hoặc **DI**) để chỉ đến địa chỉ nào đó trong vùng nhớ xác định bởi thanh ghi **DS** (*Data segment*).
- Dùng thêm số độ dời (gọi là *offset*) để dời đến ô nhớ cần truy xuất, tính từ vị trí của **SI** (hoặc **DI**).
- Ví dụ 1: đã có **DI** = **82** (đang trỏ ô nhớ 82 trong đoạn nhớ **DS**)
`MOV [DI + 3], AH` ; Đưa nội dung của thanh ghi **AH** vào ô nhớ thứ **85** trong vùng nhớ chỉ định bởi **DS**.
- Ví dụ 2: đã có **SI** = **02h**
`MOV AX, [SI + 3]` ; Đưa nội dung của ô nhớ thứ **5** trong vùng nhớ chỉ định bởi **DS** vào thanh ghi **AX**.

Giới thiệu tập lệnh của CPU

■ Các chế độ địa chỉ (Addressing modes):

► 7. Chế độ địa chỉ tương đối chỉ số cơ sở: (Base-Index Addressing Mode)

- Chế độ địa chỉ này là sự kết hợp cả 2 chế độ địa chỉ Base và Index.
- Dùng cả thanh ghi cơ sở và thanh ghi chỉ số để tính địa chỉ toán hạng.
- Ví dụ:

MOV **[BX + DI + N], AX** ; Đưa nội dung thanh ghi AX vào ngăn nhớ có địa chỉ offset tại BX + SI + N (DS:BX + SI + N).

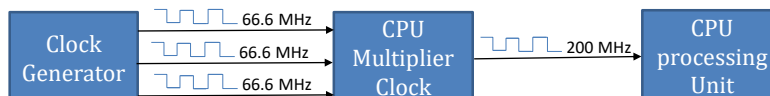
MOV **CL, [BP + SI + N]** ; Đưa nội dung ngăn nhớ có địa chỉ offset tại ; BP + DI + N (DS:BP + DI + N) vào thanh ghi CL.

CÁC THÔNG SỐ ĐẶC TRƯNG CỦA CPU

■ CPU Clock và Ratio Frequency

- **CPU Speed**: tần số hoạt động theo thiết kế của CPU.
- **CPU Clock**: tần số của xung nhịp (clock) cơ sở.
- **Ratio Frequency** (hay **Multiplier**): tỷ lệ của bộ nhân tần số để tạo ra xung nhịp lớn cấp phát cho CPU hoạt động.

■ $CPU\ speed = CPU\ clock \times Ratio\ Frequency$



CÁC THÔNG SỐ ĐẶC TRƯNG CỦA CPU

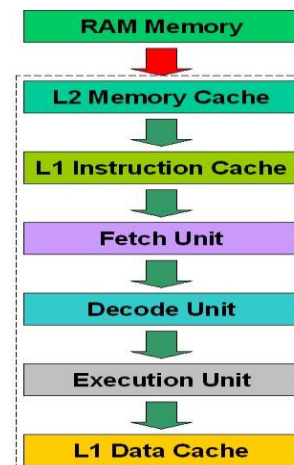
■ FSB – QPI – DMI:

- ▶ **Front Side Bus (FSB)**: trong cấu trúc máy tính dùng 2 chipset, FSB là tên gọi của đường truyền dữ liệu giữa CPU và chip cầu bắc (*NorthBridge*).
 - **FSB freq = CPU clock x 4**
- ▶ **Quick Path Interconnect (QPI)**: trong cấu trúc máy tính dùng 2 chipset, với “bộ điều khiển RAM” (*Memory Controller*) được chuyển từ chip bắc lên CPU, QPI là tên gọi của đường truyền dữ liệu giữa CPU và chip cầu bắc (*NorthBridge*).
- ▶ **Direct Media Interface (DMI)**: trong cấu trúc máy tính dùng 1 chipset, DMI là tên gọi của đường truyền dữ liệu giữa CPU và chipset trên Mainboard.

CÁC THÔNG SỐ ĐẶC TRƯNG CỦA CPU

■ Cache và vai trò của Cache:

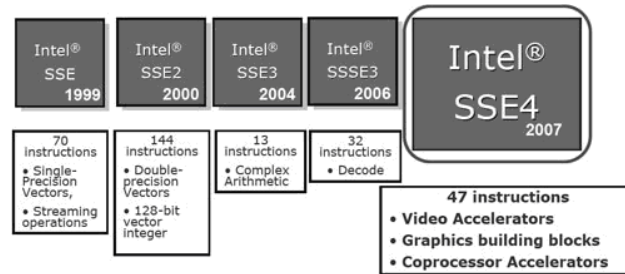
- ▶ Quá trình xử lý dữ liệu được thực hiện qua các giai đoạn:
 - Đọc lệnh (*Instruction Fetch*).
 - Giải mã (*Instruction Decode*).
 - Thi hành (*Execute*).
 - Thâm nhập bộ nhớ hoặc chuyển hướng.
 - Lưu trữ kết quả (*Result Writeback*).
- ▶ **L2 Cache**: đệm chương trình (lệnh) và dữ liệu từ RAM vào CPU.
- ▶ **L1 Instruction Cache**: đệm bộ lệnh.
- ▶ **L1 Data Cache**: đệm dữ liệu.



CÁC THÔNG SỐ ĐẶC TRƯNG CỦA CPU

■ Tập lệnh hỗ trợ tăng tốc xử lý đồ họa và âm thanh:

- ▶ Tập lệnh **MMX**: tăng cường xử lý âm thanh và đồ họa.
- ▶ Tập lệnh **SSE**: bao gồm các lệnh MMX, được tăng cường thêm 70 lệnh hỗ trợ.



CÁC THÔNG SỐ ĐẶC TRƯNG CỦA CPU

■ Socket (hoặc Slot):

- ▶ Socket (để cắm), slot (khe cắm): là module lắp đặt CPU vào Mainboard.
- ▶ Cho phép người dùng lựa chọn nhiều loại CPU để lắp đặt.



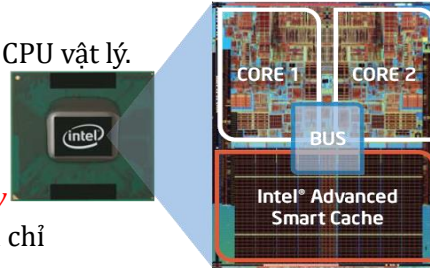
CÁC THÔNG SỐ ĐẶC TRƯNG CỦA CPU

■ Multi-Core:

- Tích hợp nhiều bộ xử lý trong 1 CPU vật lý.
- *Dual Core / Quad Core / ...*

■ EM64T:

- *Extended Memory 64 Technology* (EM64T): công nghệ mã hoá địa chỉ có độ dài 64-bit.
- *EM64T* cho phép CPU truy cập bộ nhớ (RAM) có dung lượng lớn (2^{64} bit – tương đương = 17.179.869.184 GB hay 16 ExaBytes).
- Công nghệ *32bit* hỗ trợ truy cập bộ nhớ tối đa 4GB (**3.2GB** RAM + **0.8GB** VGA RAM)



Một số công nghệ nổi bật của CPU

■ Lithography (công nghệ in thạch bản)

- *Lithography*: công nghệ in bản mạch *chip* ngay trên hợp chất bán dẫn (in thạch bản)
- Thu nhỏ kích thước *Lithography* nhằm:
 - Gia tăng số lượng transistors trên cùng một bề mặt bản in => tăng *Core*, tăng *cache*...
 - Giảm quãng đường di chuyển của dòng điện giữa các transistors => giảm nhiệt, giảm công suất tiêu thụ điện (**TDP**), tăng tốc độ xử lý...

■ Intel® Hyper-Threading

- Công nghệ “Siêu phân luồng”: sử dụng thời gian nhàn rỗi của CPU để thực thi luồng (*thread*) chương trình khác
- *Hyper Threading* tạo ra 2 nhân (*core*) ảo trên một CPU vật lý.

Một số công nghệ nổi bật của CPU

FACULTY OF INFORMATION TECHNOLOGY

- 35 -

Một số công nghệ nổi bật của CPU

■ Intel SpeedStep

- ▶ Công nghệ cho phép CPU tự động giảm tỉ lệ nhân tần số (*Ratio Frequency*) khi CPU nhàn rỗi.
- ▶ Ví dụ: khi nhàn rỗi, CPU có tốc độ thiết kế là **2.4 GHz** (*100 MHz x 24*) sẽ hoạt động ở tốc độ **1.0 GHz** (*100 MHz x 10*)
- ▶ *Speedstep* giúp giảm nhiệt, giảm điện năng tiêu thụ...

■ Intel® Turbo Boost

- ▶ Công nghệ tự động tăng tốc vượt thiết kế (gọi là ép xung – *Over clock*) khi nhu cầu xử lý tăng cao
- ▶ Ví dụ: CPU có tốc độ thiết kế là **2.4 GHz** (*100 MHz x 24*), khi *turbo boost* có thể hoạt động ở tốc độ **2.8 GHz** (*120 MHz x 24*)

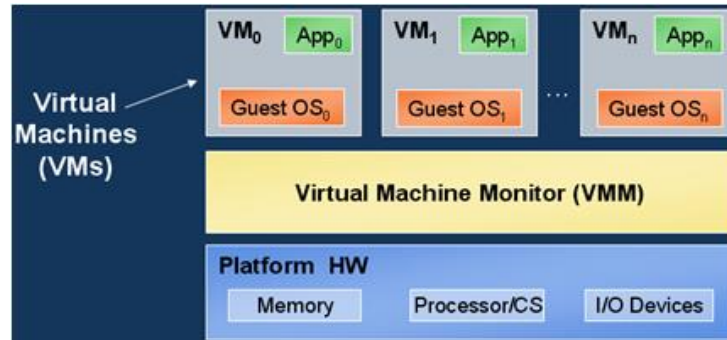
FACULTY OF INFORMATION TECHNOLOGY

- 36 -

Một số công nghệ nổi bật của CPU

■ Intel Virtualization Technology:

- ▶ Công nghệ ảo hóa CPU.
- ▶ Cho phép ảo hóa một CPU vật lý thành nhiều CPU ảo => xử lý được nhiều Hệ điều hành (máy ảo) trên cùng một máy thật.



FACULTY OF INFORMATION TECHNOLOGY

- 37 -

Các dòng sản phẩm Intel CPU

■ Thế hệ máy tính dùng 2 chip (*Northbridge và Southbridge*)

- ▶ ... 286 → 386 → 486 → 586 (Pentium)
- ▶ Pentium MMX → Pentium II → Pentium III → Pentium 4 → Pentium 4 – HT
- ▶ Pentium-D → Pentium Dual Core → *Core™ Solo/Duo* → *Core™ 2 Duo/Quad*.

■ Thế hệ máy tính dùng 2 chip – CPU đảm nhận *Memory Controller*

- ▶ *Core™ i7 9xx* (1st Generation) – dành cho Desktop
- ▶ *Core™ i7 7xx* (1st Generation) – dành cho Mobile (Laptop)

■ Thế hệ máy tính dùng 1 chip:

- ▶ *Core™ i7 8xx* và *Core™ i5*, *Core™ i3* (1st Generation)
- ▶ *Core™ i7*, *Core™ i5*, *Core™ i3* của các thế hệ 2 (2nd Generation) trở về sau.

FACULTY OF INFORMATION TECHNOLOGY

- 38 -

Các dòng sản phẩm Intel CPU

■ Sản phẩm Intel CPU giá thấp:

- ▶ Song song với dòng CPU chính thức, Intel còn sản xuất dòng CPU giá rẻ dành cho người dùng phổ thông.
- ▶ Tương đương các dòng sản phẩm từ *Pentium MMX* → *Core™ 2*, dòng sản phẩm CPU giá rẻ được gọi tên là **Celeron**.
- ▶ Tương đương các dòng sản phẩm từ *Core™ i7, i5, i3* (từ thế hệ 2 trở về sau) dòng sản phẩm CPU giá rẻ được gọi tên là **Pentium**.

■ Các dòng sản phẩm Intel CPU khác:

- ▶ **Xeon** CPU: dùng cho các máy chủ chuyên dụng (*Server*).
- ▶ **Atom** CPU: dùng cho các máy tiết kiệm điện, tốc độ thấp... (*Netbook / Smartphone*)
- ▶ **Quark** CPU: dùng cho các thiết bị điều khiển thông minh.

Thảo Luận

