

Giai đoạn Phân tích Thiết kế

Công nghệ phần mềm

Tổng quan

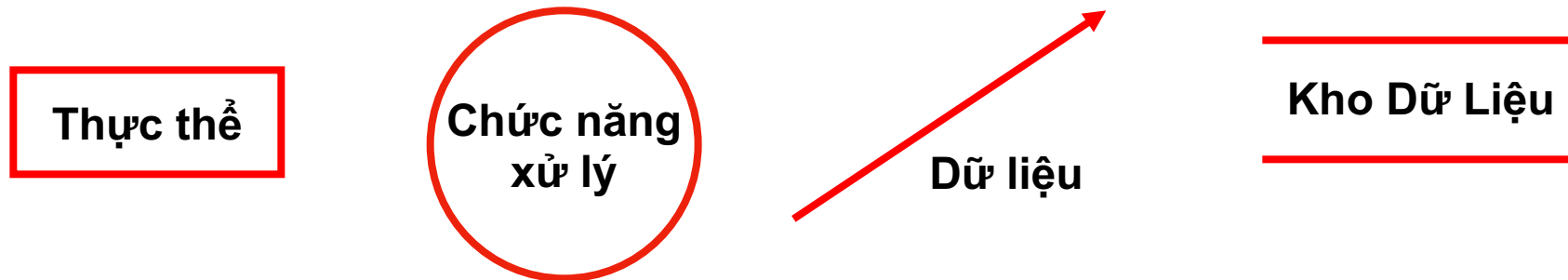
- Phân tích, thiết kế các xử lý của hệ thống
 - Theo hướng chức năng (DFD)
 - Theo hướng đối tượng (UseCase, Class)
- Phân tích, thiết kế dữ liệu (ERD)
- Phân tích, thiết kế giao diện
- Các công cụ hỗ trợ

I.1. Phân tích các xử lý của hệ thống theo hướng chức năng

- Dùng lược đồ dòng dữ liệu (DFD) để mô hình hóa các xử lý.
- Xây dựng DFD
- Các phương pháp phân tích.
- Các mức mô hình hóa
- Sơ đồ tổng quát
- Các ví dụ

LƯỢC ĐỒ DÒNG DỮ LIỆU (DFD)

- ▶ DFD được xây dựng từ 4 phần tử chính
 - **Thực thể**: tạo hoặc tiêu thụ thông tin, nằm bên ngoài phạm vi hệ thống
 - **Chức năng xử lý**: thực hiện chức năng nào đó, tiêu thụ và tạo thông tin, nằm bên trong phạm vi hệ thống
 - **Thông tin hay dữ liệu**
 - **Kho dữ liệu**: lưu trữ dữ liệu mà được sử dụng bởi nhiều chức năng xử lý



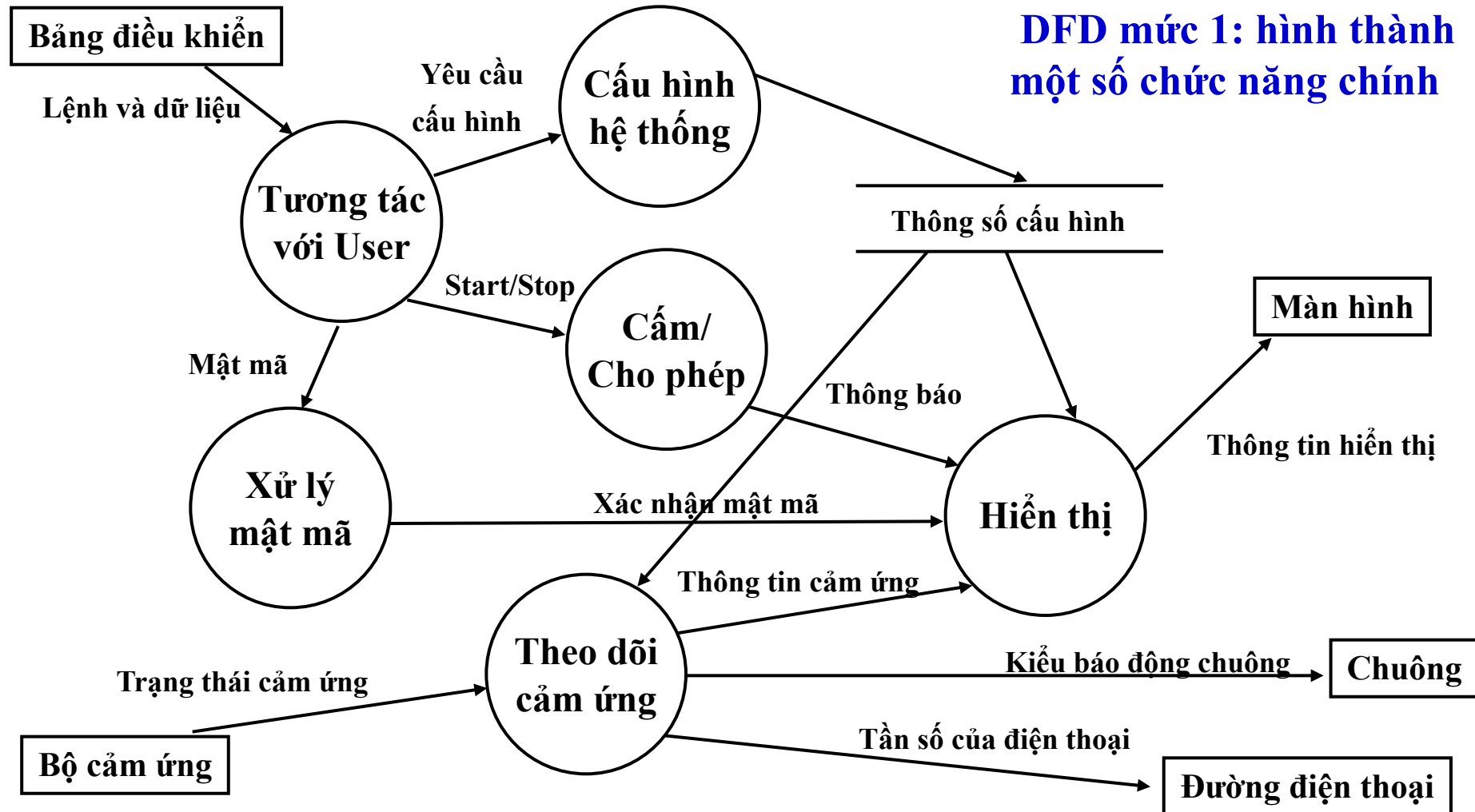
Xây dựng DFD

- Cái vào của một tiến trình phải khác với cái ra của nó.
- Các đối tượng phải có tên duy nhất. Tuy nhiên các tác nhân ngoài hay các kho dữ liệu có thể vẽ tại nhiều vị trí khác nhau để dễ quan sát.
- Mỗi tiến trình phải có một ít nhất một cái vào (input) và ra (output).
- Luồng dữ liệu không thể đi trực tiếp từ kho này đến kho khác.
- Dữ liệu không thể di chuyển trực tiếp từ một tác nhân đến một kho và ngược lại.

Xây dựng DFD

- Dữ liệu không thể đi trực tiếp từ tác nhân đến tác nhân.
- Một luồng dữ liệu không thể quay lại nơi nó vừa đi ra.
- Trong quá trình phân rã luôn luôn phải đảm bảo tính cân bằng: mọi luồng dữ liệu vào và ra, các tác nhân ngoài và các kho dữ liệu phải được bảo toàn trong sơ đồ ở mức sau.
- Quá trình phân rã dừng khi đạt được luồng dữ liệu sơ cấp (một tiến trình là một tính toán hay một thao tác dữ liệu đơn giản).

Xây dựng DFD – Ví dụ



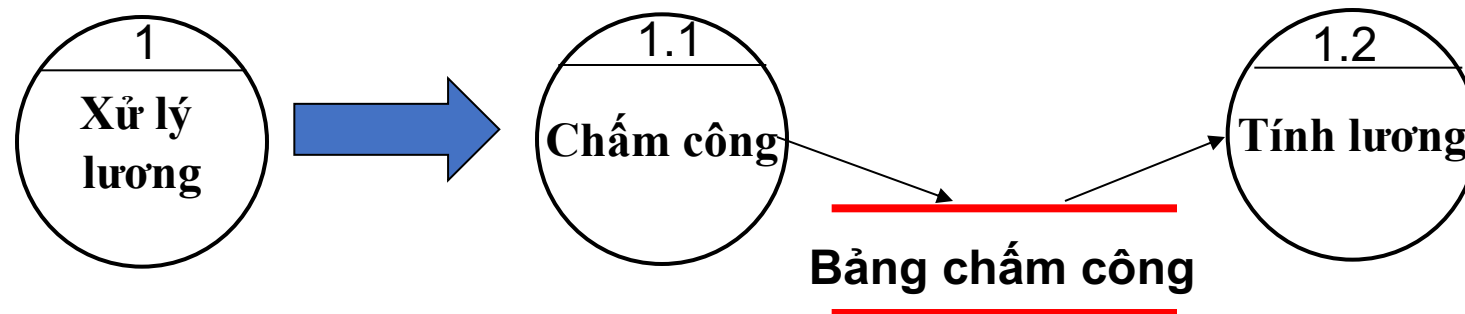
Các phương pháp phân tích

- Luật 1: Nếu xử lý đầu nhằm chuyển tiếp thông tin, và xử lý sau sử dụng thông tin đó.
- Thì xử lý ban đầu có thể phân tích thành một cặp xử lý kết nối với nhau bằng dòng dữ liệu.



Các phương pháp phân tích

- Luật 2: Một xử lý có thể phân tích thành một cặp xử lý và kho dữ liệu. Vì 2 xử lý có thời điểm khác nhau, nên dữ liệu chuyển đổi phải được lưu lại trong một kho dữ liệu.



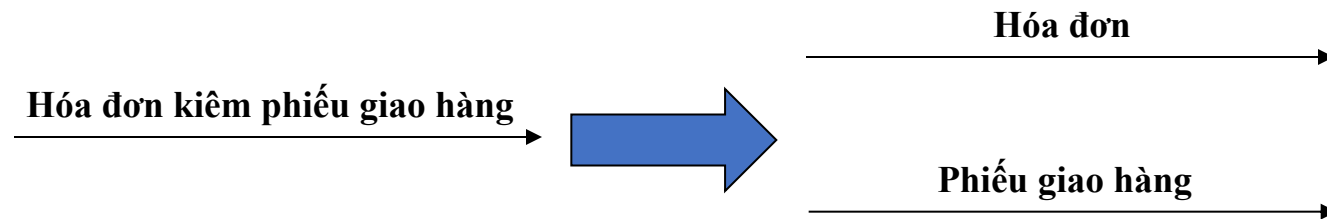
Các phương pháp phân tích

- Luật 3: Một xử lý có thể phân tích thành một cặp xử lý riêng biệt. Vì 2 xử lý không có nhu cầu trao đổi thông tin.



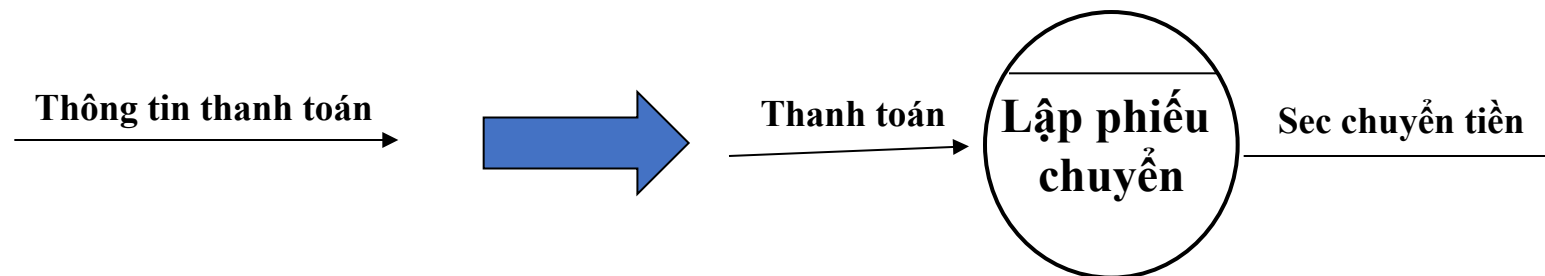
Các phương pháp phân tích

- Luật 4: Tách một dòng dữ liệu thành tập các dòng dữ liệu. Để chuyển một dòng dữ liệu tổng hợp thành các dòng dữ liệu có nội dung độc lập.



Các phương pháp phân tích

- Luật 5: Tách một dòng dữ liệu thành 2 dòng dữ liệu và một ô xử lý. Khi nhận ra dòng dữ liệu có chuyển đổi nội dung thông tin.

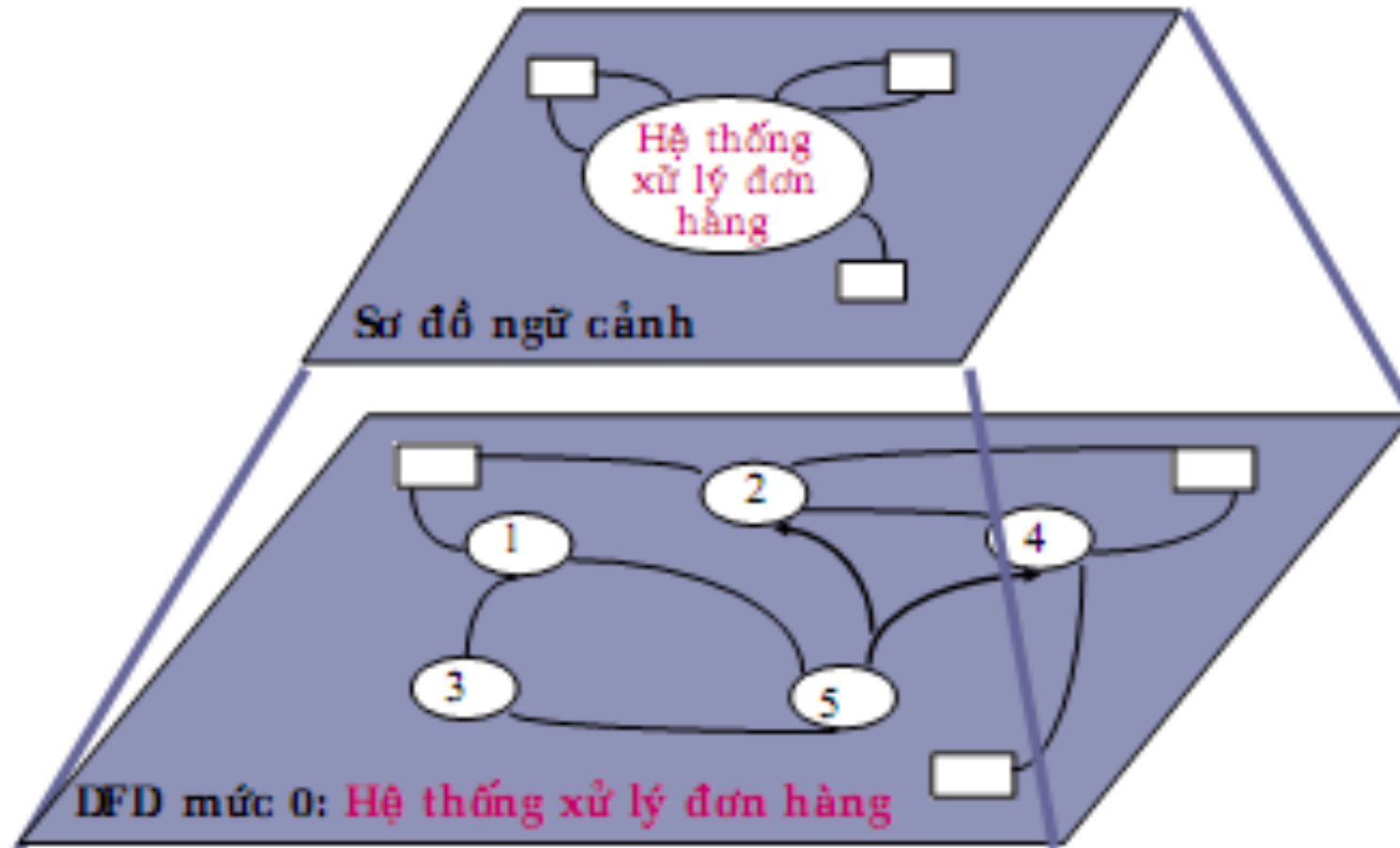


Các phương pháp phân tích

- Thông thường dùng phương pháp phân tích dữ liệu từ trên xuống.

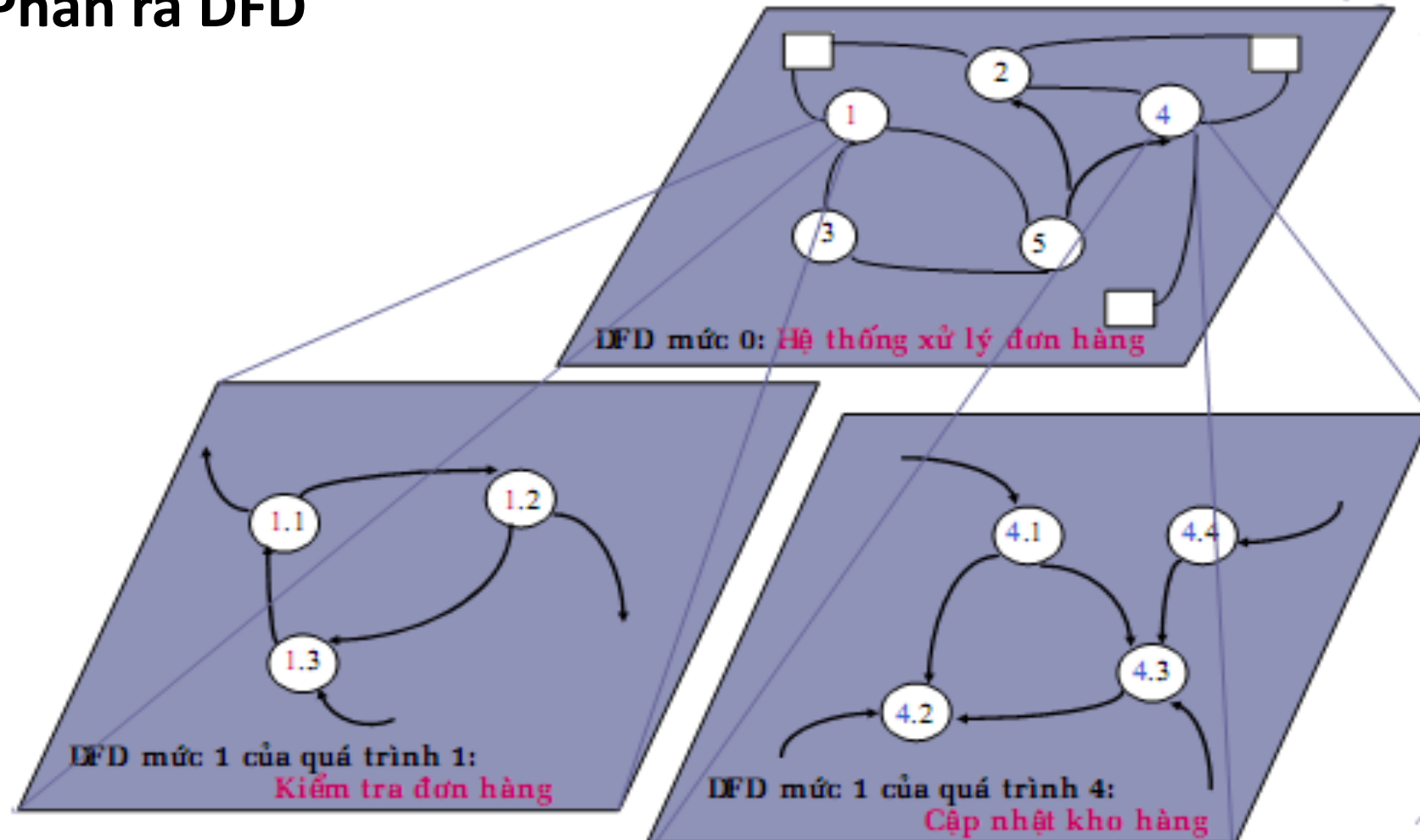
Phân tích từ trên xuống

Phân rã DFD

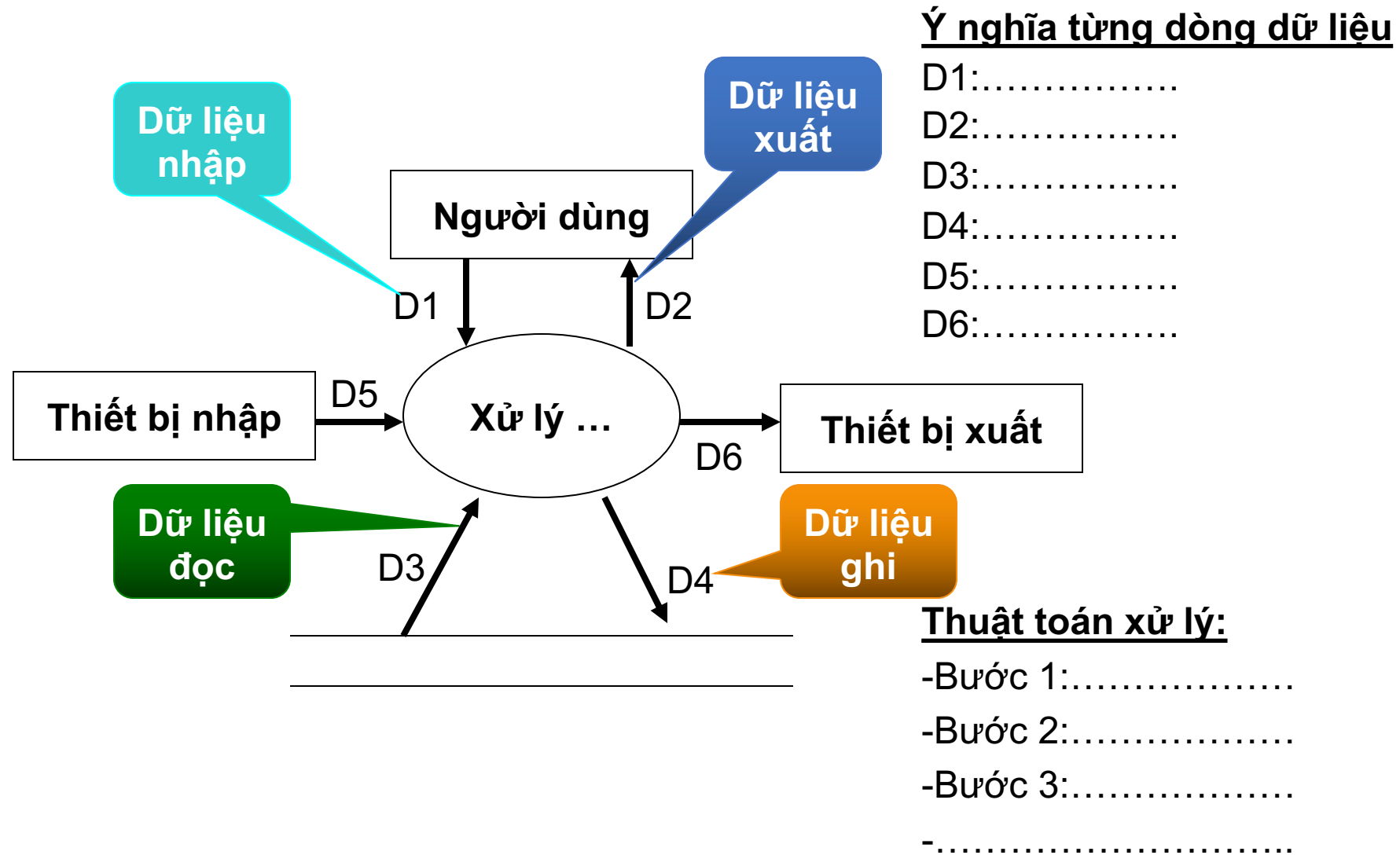


Phân tích từ trên xuống

Phân rã DFD



Sơ đồ tổng quát

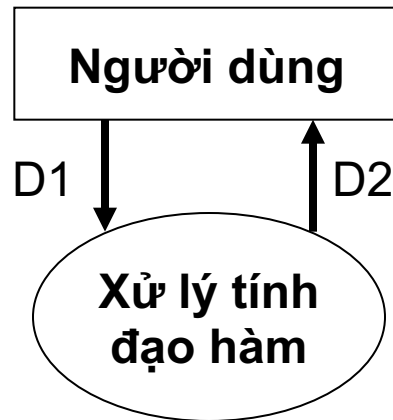


Ví dụ 1

- Xét chức năng tính đạo hàm của một đơn thức

1. Phân tích

Sơ đồ luồng dữ liệu

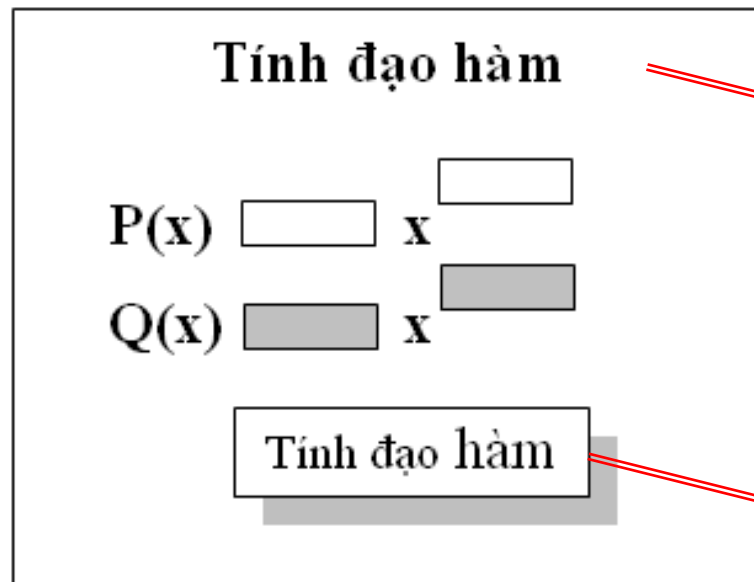


- ❖ D1: Đơn thức cần tính đạo hàm P
- ❖ D2: Đơn thức kết quả Q
- ❖ Thuật toán xử lý
 - Nhập và kiểm tra D1 ($a \neq 0$)
 - Tính và xuất D2

Ví dụ 1

2. Thiết kế

- Mô tả chi tiết cách thức giao diện



Khởi động giá trị ban đầu

Kiểm tra P hợp lệ và nhập giá trị cho P

Tính Q
Xuất Q

Ví dụ 1

2. Thiết kế

- Mô tả chi tiết kiểu dữ liệu: Sử dụng kiểu cấu trúc DON_THUC với hai thành phần:
 - Hệ số có kiểu số thực
 - Số mũ có kiểu số nguyên (không âm)
- Mô tả chi tiết các hàm xử lý
 - Hàm xử lý biến cố 1
 - Hàm kiểm tra hệ số, Hàm kiểm tra số mũ
 - Hàm nhập đơn thức
 - Hàm tính đạo hàm, Hàm xuất đơn thức

Ví dụ 1

3. Lập trình

‘Khai báo kiểu cấu trúc (C)

```
typedef struct DON_THUC
```

```
{
```

```
    int Heso;
```

```
    int Somu;
```

```
};
```

‘Khai báo biến

```
DON_THUC P, Q;
```

Ví dụ 1

3. Lập trình

‘Hàm xử lý biến cố tính đạo hàm trên màn hình

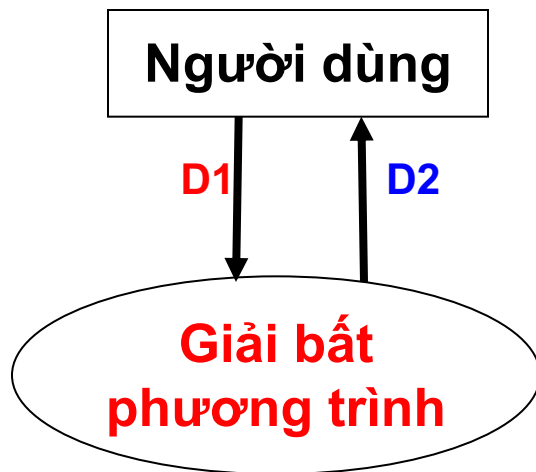
```
void cmdDaoham_Click()  
    if (Kiem_Tra_He_So() && Kiem_Tra_So_Mu()){  
        Nhap();  
        DaoHam();  
        Xuat();  
    }  
}
```

Ví dụ 2

- Xét chức năng giải bất phương trình bậc nhất có dạng $ax + b \geq 0$, với $a \neq 0$
- Hãy phân tích, thiết kế và lập trình

Ví dụ 2

- Sơ đồ luồng dữ liệu



❖ **D1**: Các hệ a, b của bất phương trình

❖ **D2**: Nghiệm của bất phương trình thuộc một trong hai dạng sau:

- Dạng 1: $(-\infty, x_0]$, Dạng 2: $[x_0, +\infty)$

❖ **Xử lý**:

- Nhập và kiểm tra D1 ($a \neq 0$)
- Tính D2 theo qui tắc:
 - $a > 0$: nghiệm thuộc dạng 2
 - $a < 0$: nghiệm thuộc dạng 1
 - Với $x_0 = -b/a$

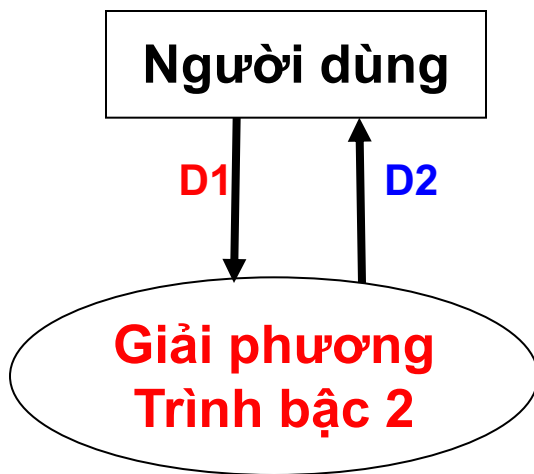
- Xuất D2

Ví dụ 3

- Xét chức năng giải phương trình bậc hai:
 $ax^2 + bx + c = 0$, với $a \neq 0$
- Hãy phân tích, thiết kế và lập trình

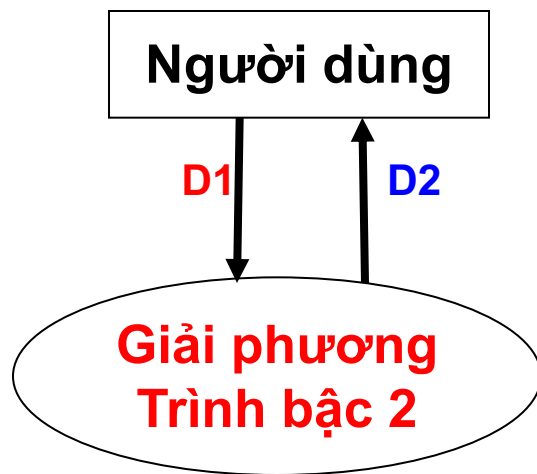
Ví dụ 3

- Sơ đồ luồng dữ liệu



- ❖ **D1**: Các hệ a, b, c của tam thức bậc 2
 - $P(x) = ax^2 + bx + c$
- ❖ **D2**: Nghiệm của phương trình $P(x) = 0$ thuộc một trong ba loại sau:
 - Vô nghiệm
 - Nghiệm kép $x_1 = x_2$
 - Hai nghiệm phân biệt x_1 và x_2

Ví dụ 3



❖ Xử lý:

- Nhập và kiểm tra D1 ($a \neq 0$)
- Tính D2 theo các bước:
 - $\Delta = b^2 - 4ac$
 - Nếu $\Delta < 0$: nghiệm loại 1
 - Nếu $\Delta = 0$: nghiệm loại 2, với $x_1 = x_2 = -b/2a$
 - Nếu $\Delta > 0$: nghiệm loại 3, với $x_1 = \frac{-b - \sqrt{\Delta}}{2a}$
 $x_2 = \frac{-b + \sqrt{\Delta}}{2a}$
- Xuất D2

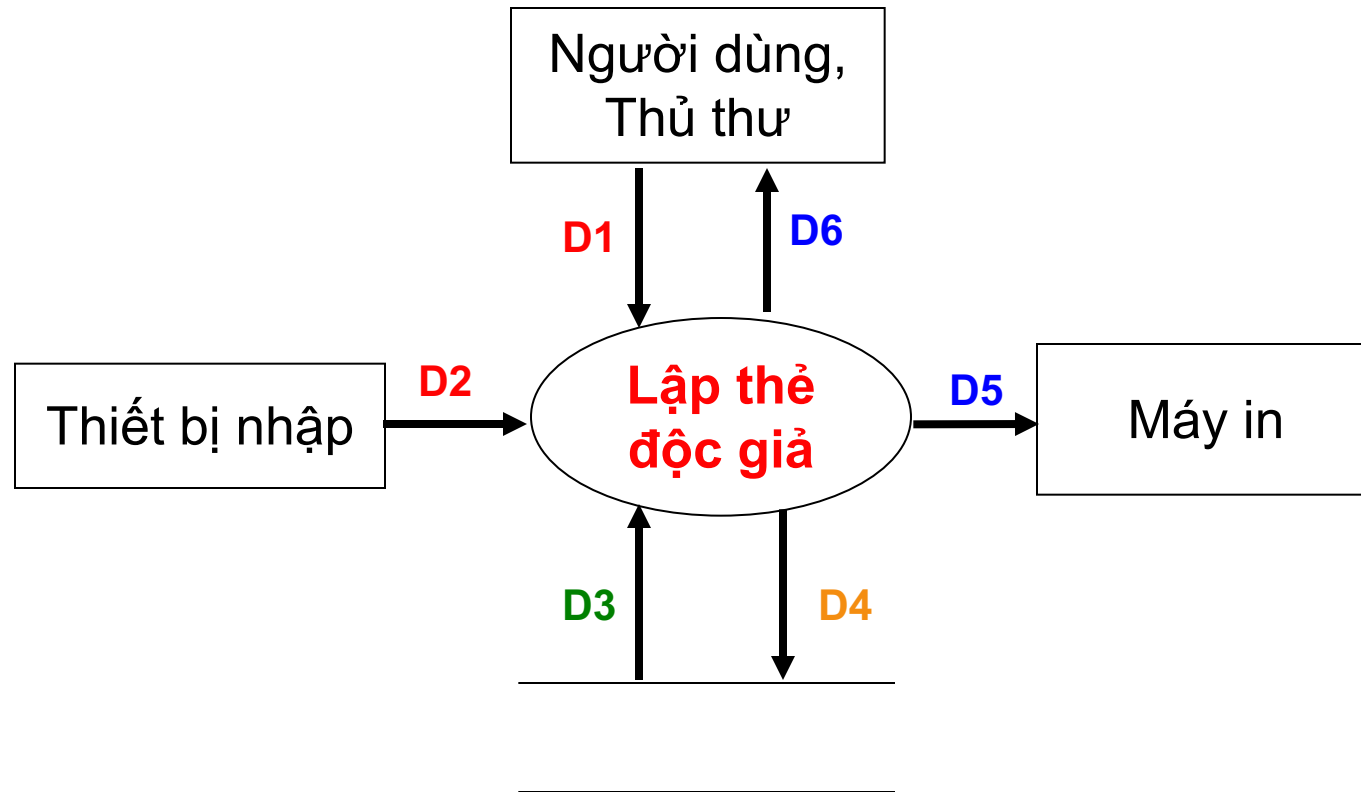
Ví dụ 4

- Xét phần mềm quản lý thư viện, hãy lập sơ đồ luồng dữ liệu cho yêu cầu

BM2:	Thẻ Độc Giả		
Họ và tên:	Loại độc giả:.....	Ngày sinh:	
Địa chỉ:	Email:	Ngày lập thẻ:	
Người lập:			

QĐ2: Có 2 loại độc giả (X,Y). Tuổi độc giả từ 18 đến 55. Thẻ có giá trị 6 tháng.

Ví dụ 4



Ví dụ 4

- Giải thích:
 - D1: Thông tin về thẻ độc giả: Họ tên, Loại độc giả, Ngày sinh, Địa chỉ, E-Mail, Ngày Lập Thẻ.
 - D2: Không có
 - D3: Danh sách các loại độc giả, Tuổi tối thiểu, Tuổi tối đa, Thời hạn sử dụng.
 - D4: D1
 - D5: D4
 - D6: Danh mục loại độc giả

Ví dụ 4

- Thuật toán:

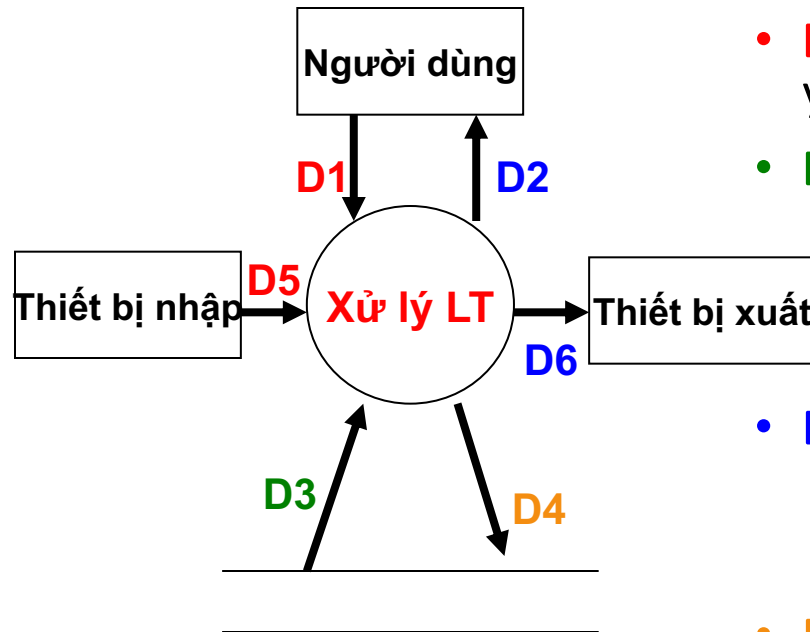
- Bước 01: Kết nối dữ liệu
- Bước 02: Đọc D3 từ bộ nhớ phụ
- Bước 03: Nhận D1 từ người dùng
- Bước 04: Kiểm tra “Loại độc giả” có thuộc “danh sách các loại độc giả” hay không?
- Bước 05: Tính tuổi độc giả.
- Bước 06: Kiểm tra qui định “Tuổi tối thiểu”
- Bước 07: Kiểm tra qui định “Tuổi tối đa”

Ví dụ 4

- Thuật toán:

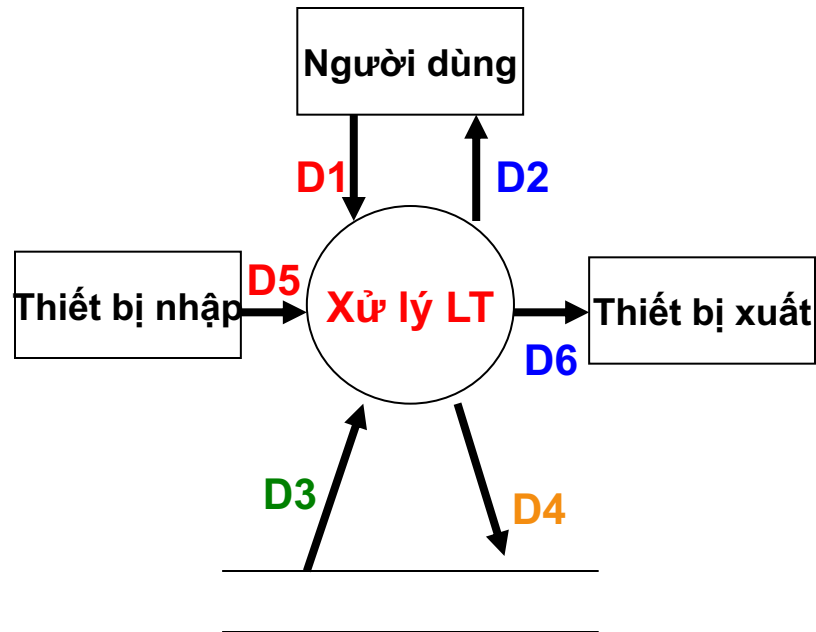
- Bước 08: Nếu không thỏa tất cả các qui định trên thì tới bước 12
- Bước 09: Tính ngày hết hạn của thẻ.
- Bước 10: Lưu D4 xuống bộ nhớ phụ
- Bước 11: Xuất D5 ra máy in
- Bước 12: Đóng kết nối cơ sở dữ liệu
- Bước 13: Kết thúc.

Sơ đồ tổng quát cho Yêu cầu lưu trữ



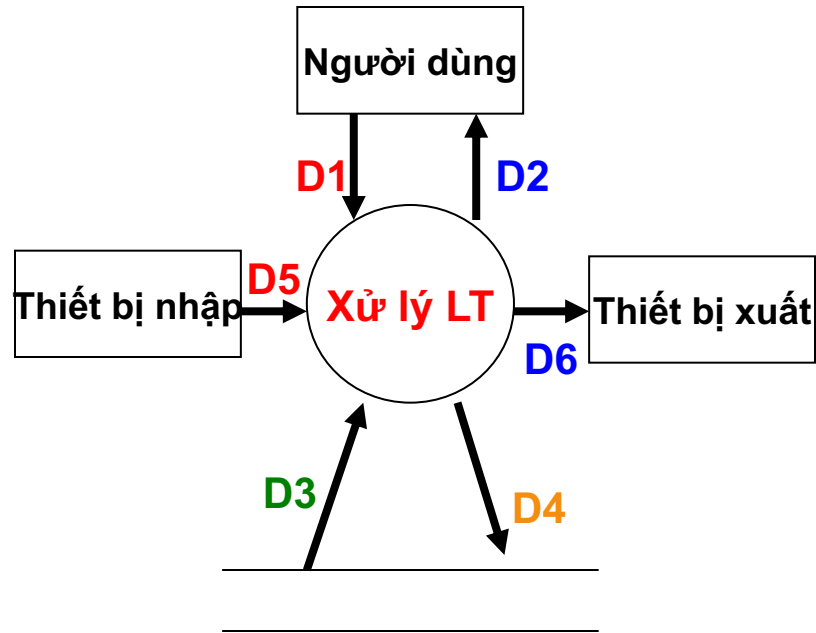
- **D1**: Thông tin cần lưu trữ (dựa vào biểu mẫu liên quan)
- **D5**: Thông tin cần lưu trữ (chỉ có trong một số yêu cầu đặc biệt)
- **D3**:
 - Các danh mục để chọn lựa
 - Dữ liệu cần thiết cho việc kiểm tra tính hợp lệ (dựa vào quy định)
- **D2**:
 - Các danh mục để chọn lựa
 - Kết quả thành công/thất bại
- **D4**: Dữ liệu được lưu trữ (dựa vào biểu mẫu).
 - Ghi chú: Thông thường
 $D4 = D1 (+ D5) (+ ID \text{ tự phát sinh})$
- **D6**: Dữ liệu kết xuất (chỉ có trong một số yêu cầu đặc biệt)

Sơ đồ tổng quát cho Yêu cầu lưu trữ



- Xử lý lưu trữ
 - **Đọc D3** để lấy các tham số, quy định và danh mục
 - **Hiển thị D2** (các danh mục)
 - **Nhận thông tin D1, D5** (nếu cần)
 - Kiểm tra các thông tin D1, D5 có thỏa quy định liên quan hay không (dựa vào D3 nếu cần thiết)
 - Nếu thỏa quy định, **ghi D4**, **thông báo kết quả D2** (nếu cần) và **xuất D6** (nếu cần thiết)

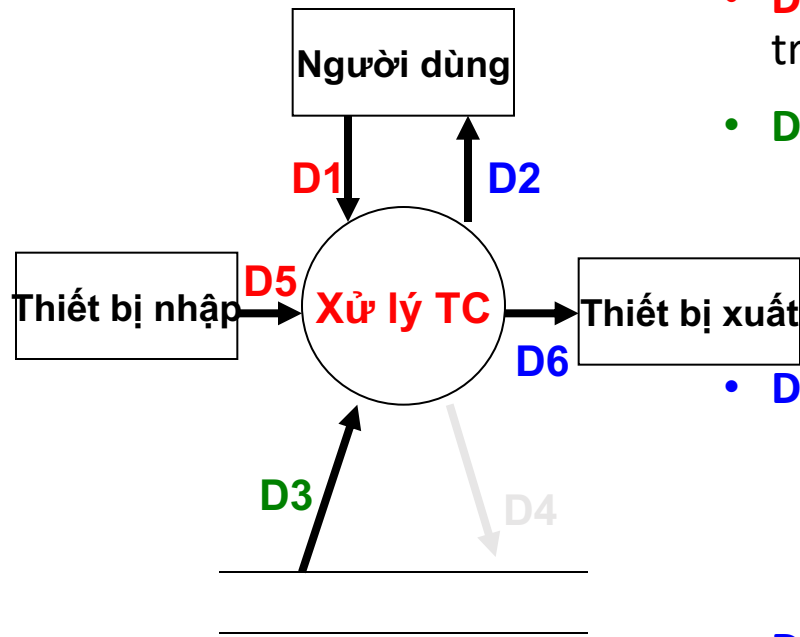
Sơ đồ tổng quát cho Yêu cầu lưu trữ



- Ghi chú:

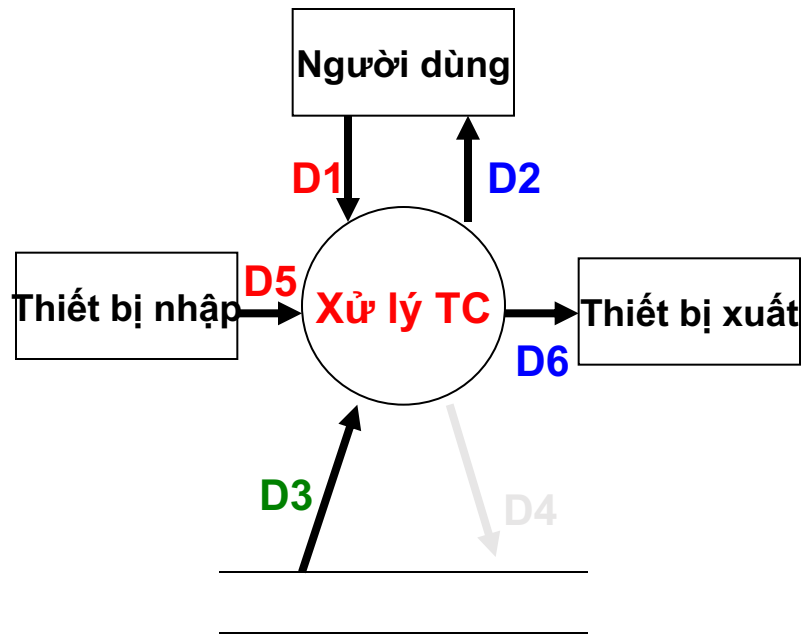
- D1 không nhất thiết chứa toàn bộ thông tin trong biểu mẫu liên quan
- Tùy theo quy định có thể có hay không có D5
- D4 hoặc D6 không nhất thiết phải trùng với D1 hoặc D5
- D2 không nhất thiết phải trùng với D3

Sơ đồ tổng quát cho Yêu cầu tra cứu



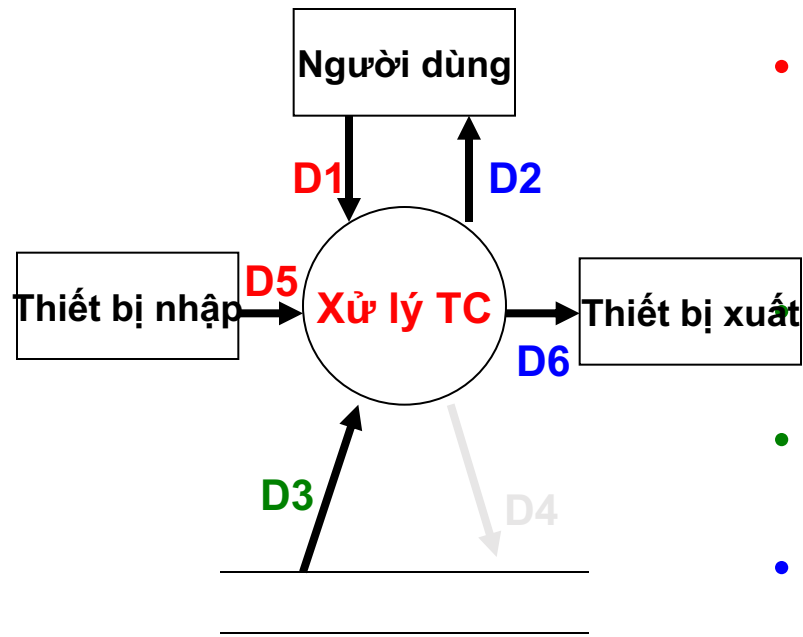
- **D1:** Thông tin về đối tượng muốn tìm kiếm (dựa vào **biểu mẫu liên quan đến đối tượng cần tìm kiếm**)
- **D5:** Thông tin về đối tượng muốn tìm kiếm (chỉ có trong một số yêu cầu đặc biệt)
- **D3:**
 - Các danh mục để chọn lựa
 - Dữ liệu về đối tượng khi tìm thấy (dựa vào **biểu mẫu liên quan đến đối tượng cần tìm kiếm**)
- **D2:**
 - Các danh mục để chọn lựa
 - Dữ liệu về đối tượng khi tìm thấy (dựa vào **biểu mẫu liên quan đến đối tượng cần tìm kiếm**)
- **D6:** Dữ liệu kết xuất (thông thường là cần thiết)
- **D4:** Dữ liệu cần lưu trữ lại
 - Thông thường không cần thiết
 - Cần thiết khi nào???

Sơ đồ tổng quát cho Yêu cầu tra cứu



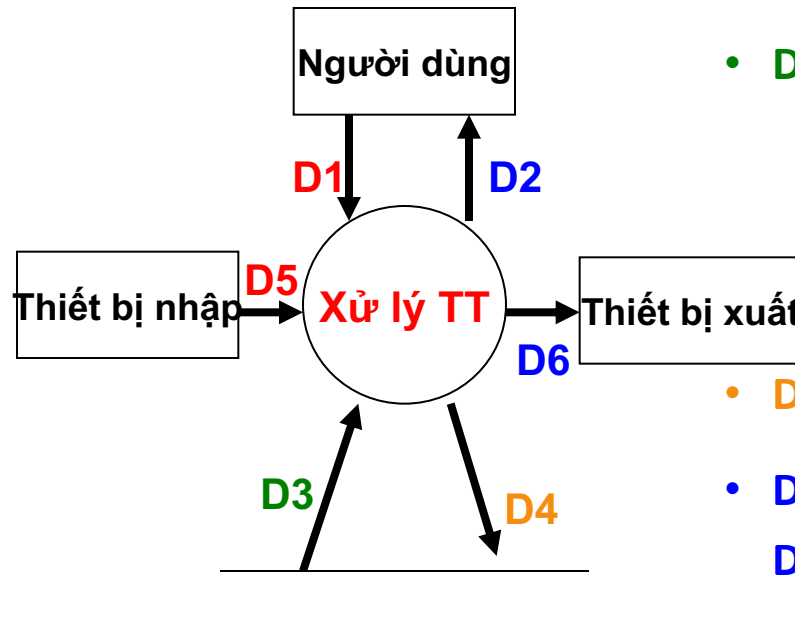
- Xử lý tra cứu
 - **Đọc** để lấy các danh mục (**D3**)
 - **Hiển thị D2** (các danh mục)
 - **Nhận thông tin về tiêu chí tìm kiếm D1, D5** (nếu cần)
 - Tìm kiếm theo các tiêu chí **D1, D5**, nhận được **danh sách các đối tượng tìm được (D3)**
 - **Hiển thị** thông tin kết quả (**D2**) và kết xuất **D6** (nếu cần)

Sơ đồ tổng quát cho Yêu cầu tra cứu



- Ghi chú:
 - Có rất nhiều mức độ khác nhau từ rất đơn giản đến rất phức tạp để xác định **D1**
 - **D1** chứa nhiều thông tin thì việc tìm kiếm sẽ dễ dàng cho người dùng và ngược lại sẽ khó khăn cho phần thiết kế và cài đặt chức năng này
 - **D3** thông thường là danh sách các đối tượng tìm thấy cùng với thông tin liên quan.
 - **D3** cũng có rất nhiều mức độ khác nhau để xác định các thông tin của đối tượng tìm thấy
 - **D2** và **D6** thường trùng với **D3** (nhưng không nhất thiết)

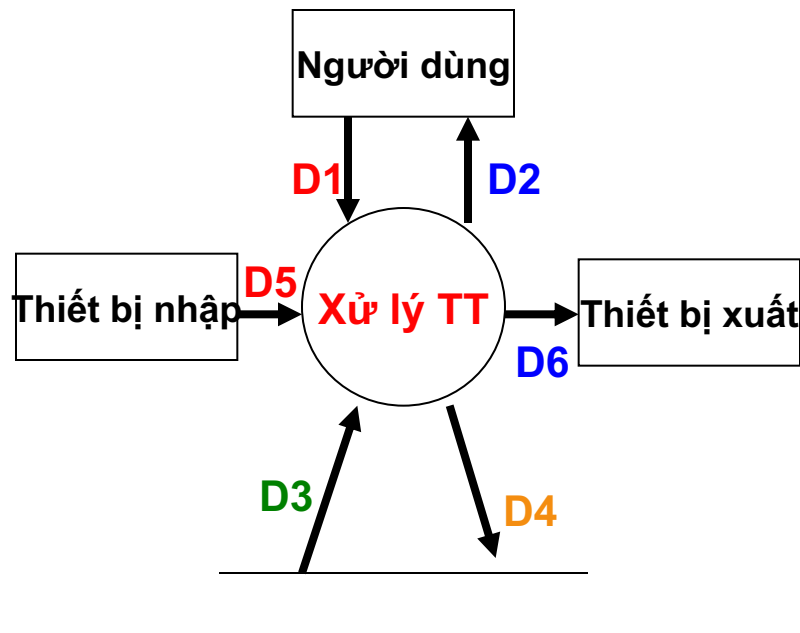
Sơ đồ tổng quát cho Yêu cầu tính toán



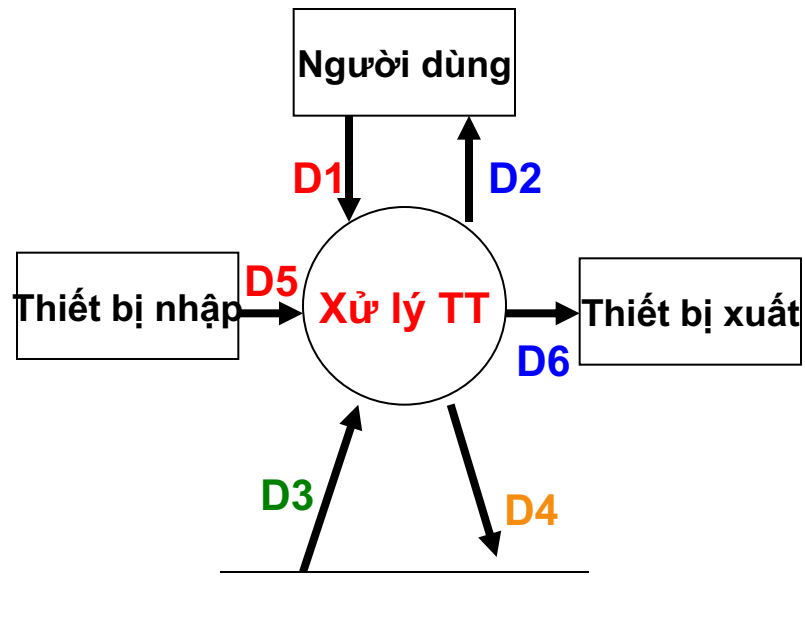
- **D1**: Thông tin về đối tượng cần thực hiện việc xử lý tính toán (dựa vào *các biểu mẫu liên quan*)
- **D5**: Thông tin về đối tượng cần thực hiện việc xử lý tính toán (chỉ có trong một số yêu cầu đặc biệt)
- **D3**:
 - Dữ liệu cần thiết cho việc xử lý tính toán (dựa vào biểu mẫu và quy định liên quan)
 - Các tham số tính toán
- **D4**: Kết quả của xử lý tính toán
- **D2**: Kết quả của xử lý tính toán (thường gồm cả **D3** và **D4**)
- **D6**: Dữ liệu kết xuất (thường gồm cả **D3** và **D4**)

Sơ đồ tổng quát cho Yêu cầu tính toán

- Xử lý tính toán
 - Nhận thông tin D1, D5 (nếu cần)
 - Đọc D3 để lấy các dữ liệu cần thiết cho việc tính toán (kể cả các tham số)
 - Sử dụng D1, D3, D5 và quy định liên quan để tính kết quả D4
 - Ghi kết quả D4
 - Hiển thị thông tin kết quả D2 và kết xuất D6



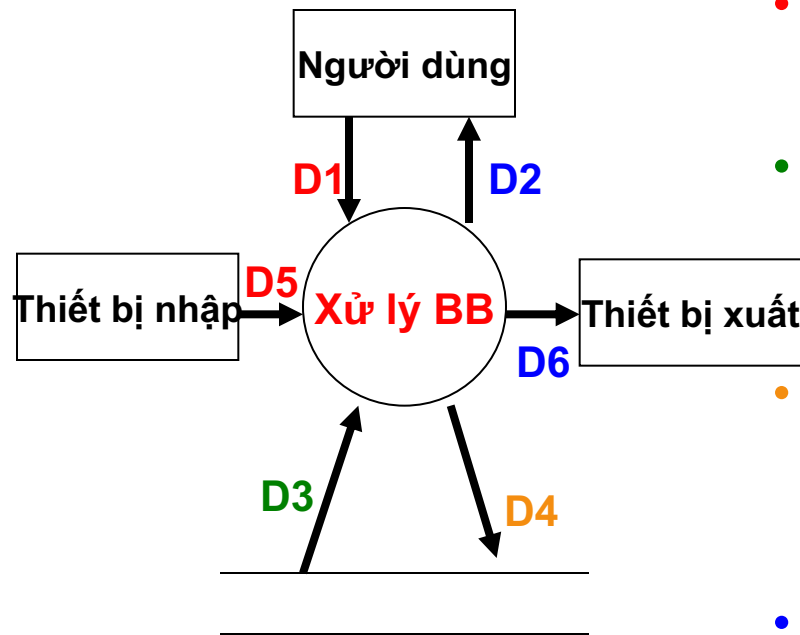
Sơ đồ tổng quát cho Yêu cầu tính toán



- Ghi chú:

- **D1** thường có chứa yếu tố thời gian thực hiện xử lý tính toán
- Có nhiều mức độ khác nhau xác định **D1** trong xử lý tính toán (để tăng tính tiện dụng)
- **D1** có thể rỗng (tính toán cho mọi đối tượng trong tất cả cột mốc thời gian liên quan)
- **D4** có thể có hay không có
=> Khi nào cần **D4**?
- Thông thường **D2** và **D6** bao gồm **D3** và **D4**

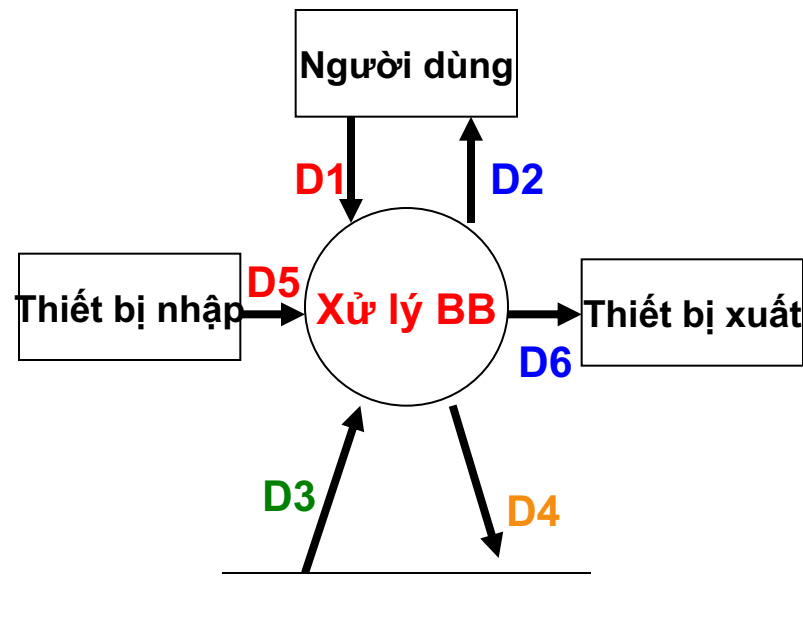
Sơ đồ tổng quát cho Yêu cầu báo biểu



- **D1**: Thông tin về báo biểu muốn thực hiện (dựa vào **biểu mẫu liên quan**)
- **D5**: Thông tin về báo biểu muốn thực hiện (chỉ có trong một số yêu cầu đặc biệt)
- **D3**: Dữ liệu cần thiết cho việc thực hiện báo biểu (dựa vào biểu mẫu và quy định liên quan)
- **D4**: Thông tin có trong báo biểu liên quan (cần thiết phải lưu lại) nhưng chưa được xử lý và ghi nhận lại (yêu cầu xử lý tính toán)
- **D2**: Thông tin về báo biểu được lập (biểu mẫu liên quan)
- **D6**: Dữ liệu kết xuất (thường giống **D2**)

Sơ đồ tổng quát cho Yêu cầu báo biểu

- Xử lý báo biểu



- Nhận thông tin D1, D5 (nếu cần)
- Đọc D3 để lấy các dữ liệu cần thiết cho việc lập báo biểu
- Nếu có D4 thì tính toán theo quy định và Ghi kết quả D4
- Hiển thị thông tin báo biểu D2 và kết xuất D6

Sơ đồ tổng quát cho Yêu cầu báo biểu

- Ghi chú:

- **D1** thường có chứa yếu tố thời gian của báo biểu
- Có nhiều mức độ khác nhau xác định **D1** trong xử lý tính toán (để tăng tính tiện dụng)
- **D4** có thể có hay không có
=> Khi nào cần **D4**?
- Thông thường **D2** và **D6** bao gồm **D3** và **D4**

