



西南大学



软件工程

Software Engineering

吴浪

西南大学 计算机与信息科学学院

Email: wl0776@swu.edu.cn



西南大學



第五章 总体设计



西南大學

目录 CONTENTS

第一节

设计过程

第二节

设计原理

第三节

启发式规则

第四节

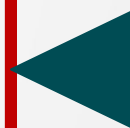
图像工具

第五节

面向数据流的设计

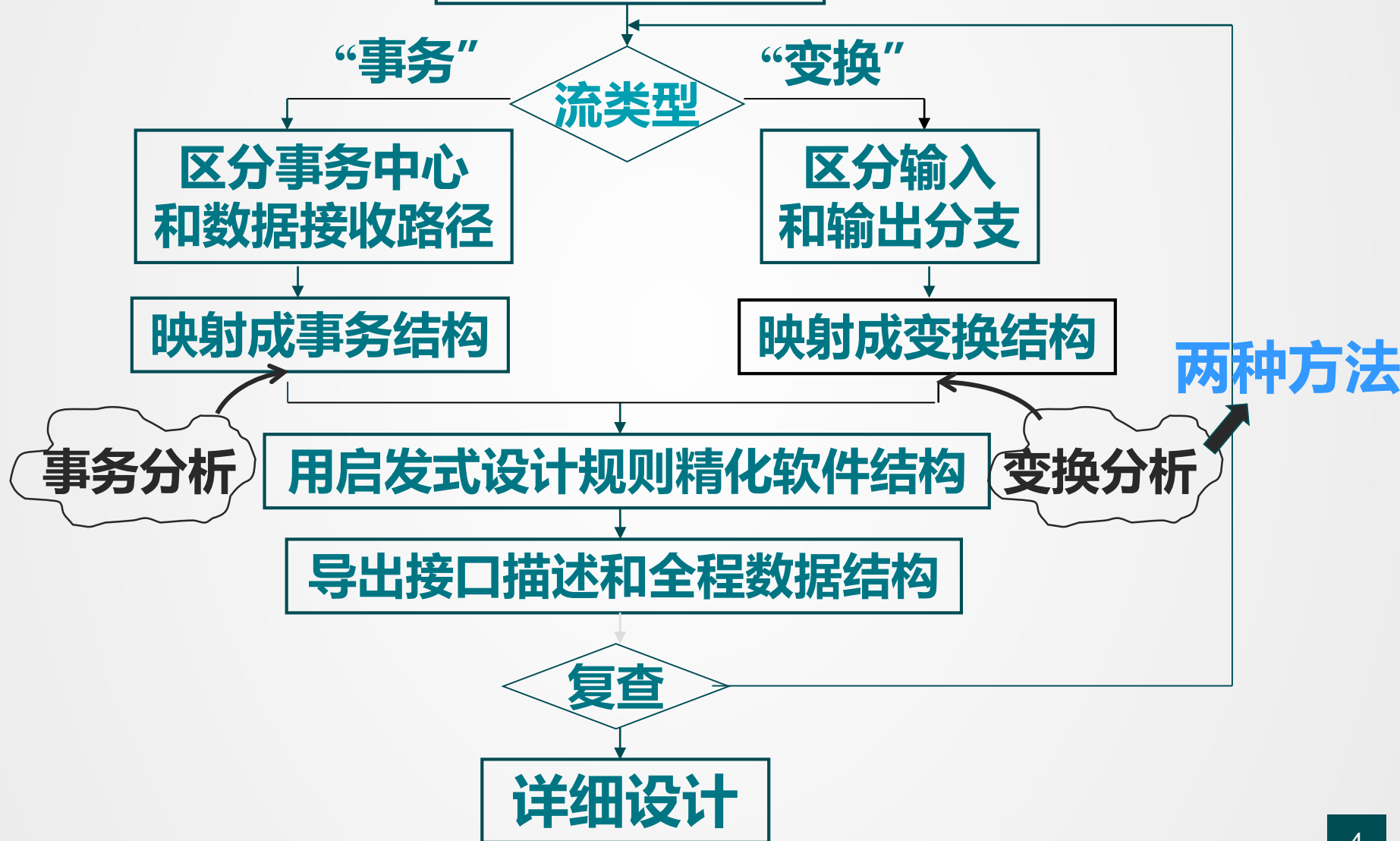
第六节

总体设计举例



SD设计过程

精化数据流图



1、变换分析 例子

设计步骤：

第1步 复查基本系统模型。

第2步 复查并精化数据流图。

第3步 确定数据流类型。

第4步 确定输入流和输出流的边界，从而孤立出变换中心。

第5步 设计系统的上层模块结构（完成第一级分解）。

第6步 设计下层模块（完成第二级分解）。

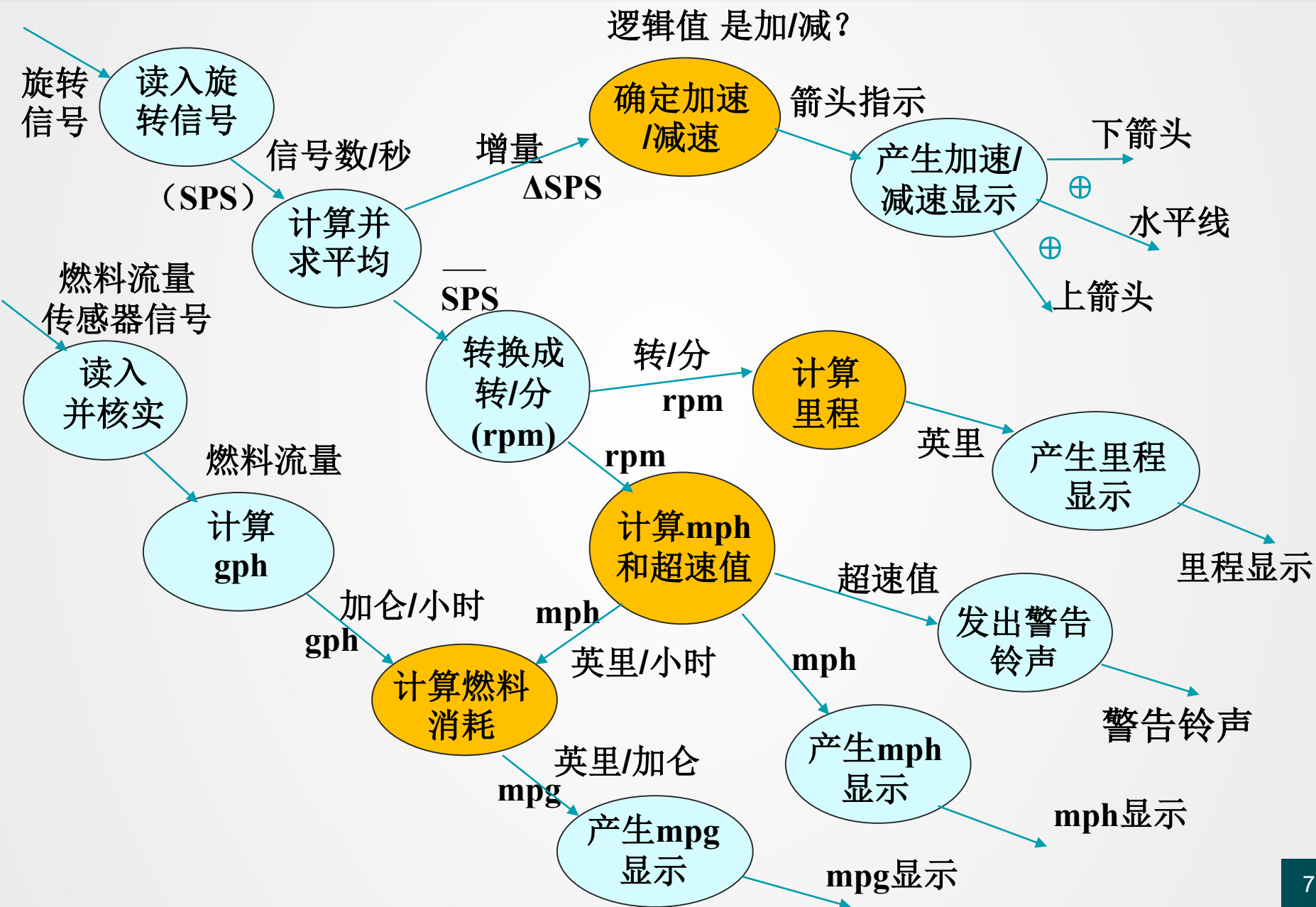
第7步 精化模块结构。

两种方法

重复举例

例：开发一个带有微处理器的汽车数字仪表盘控制系统。假设完成功能如下：

- (1) 通过模/数 (A/D) 转换实现传感器和微处理器接口。
- (2) 在发光二极管面板上显示如下数据：
 - 显示每小时行驶的英里数mph;
 - 显示每加仑油行驶的英里数mpg;
 - 显示汽车是加速或减速行驶;
 - 显示里程。
- (3) 如果汽车的速度超过55英里/小时，则发出超速警告铃声。



设计步骤:

第1步 复查基本系统模型。

确保系统的输入数据和输出数据符合实际。

第2步 复查并精化数据流图。

前面给出的数字仪表盘系统的数据流图已经足够详细，因此不需要精化。

第3步 确定数据流类型。

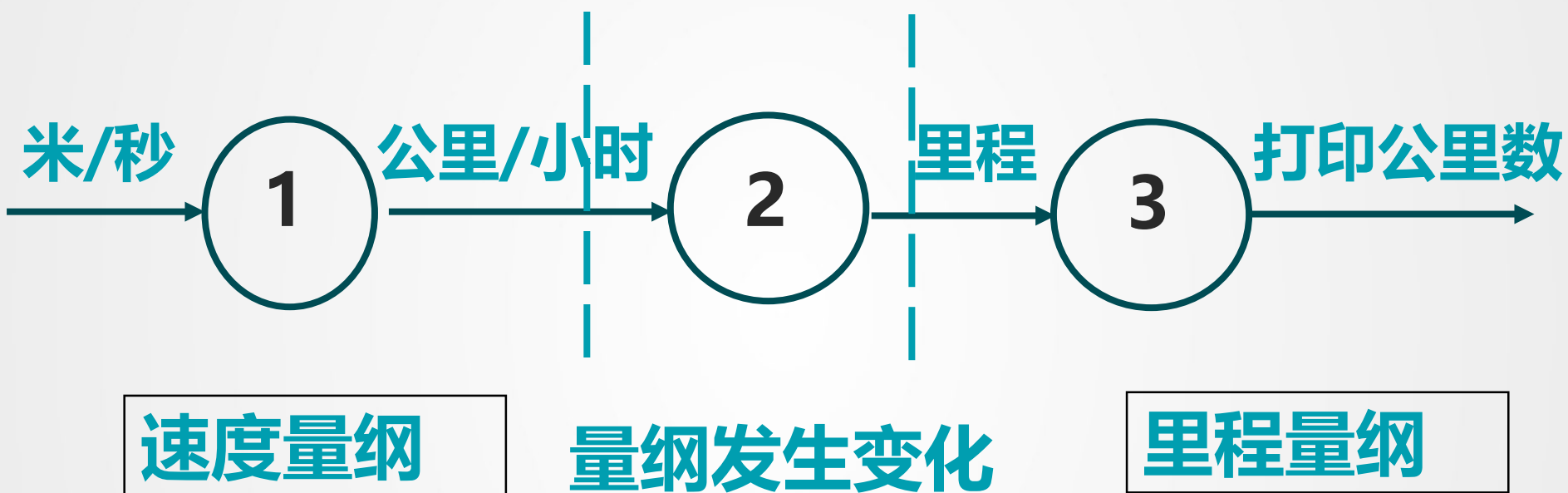
从图可以看出，没有明显的事务中心，因此认为数据流图的类型是变换型的数据流图。

第4步 确定输入流和输出流的边界，从而孤立出变换中心。

输入部分

变换中心

输出部分

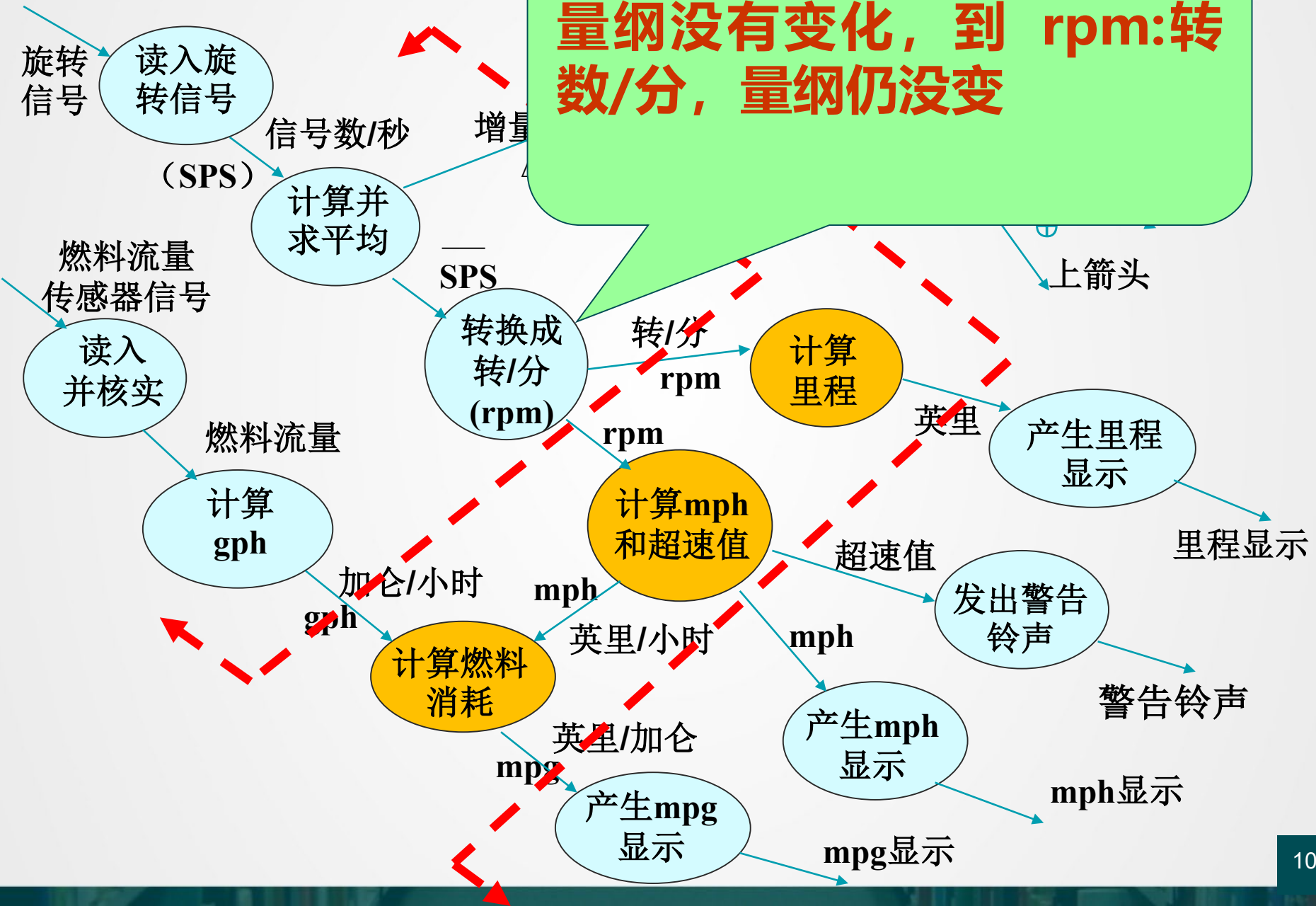


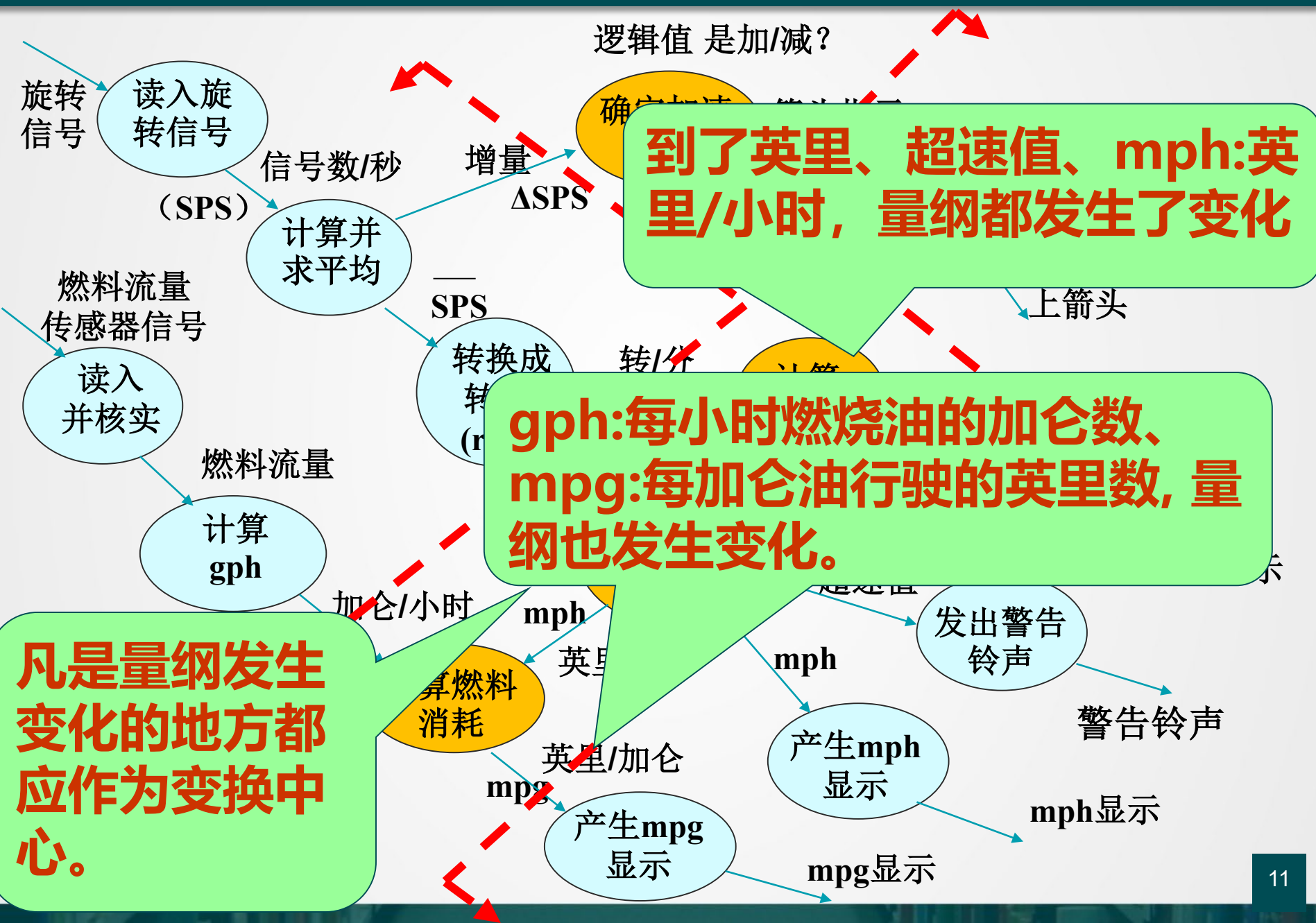
速度、里程发生了质的变化。

还有像流量、容积等也是如此。

面向数据流的设计举例

SPS:转数/秒和平均转数, 量纲没有变化, 到 rpm:转数/分, 量纲仍没变





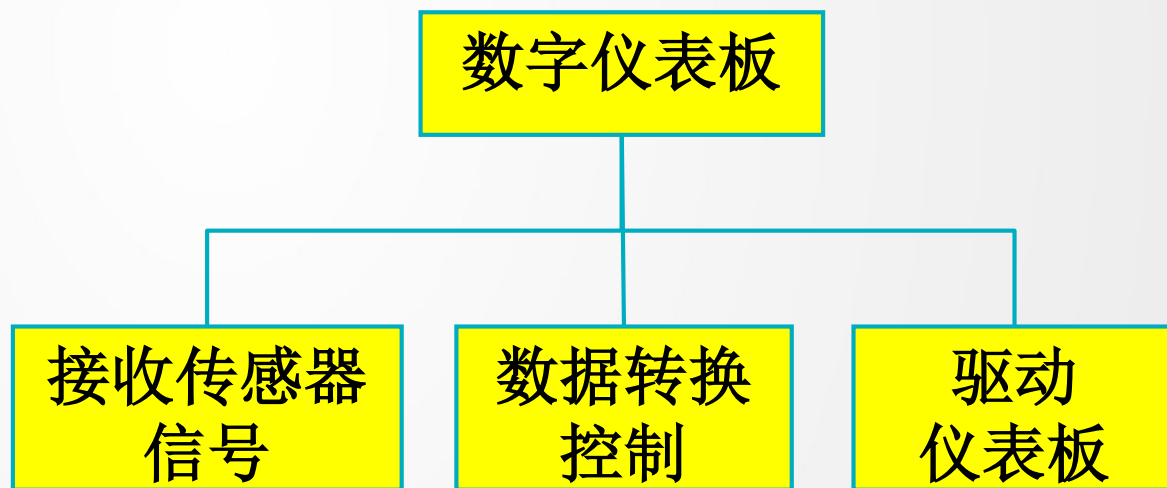
第5步 设计系统的上层模块结构 (完成第一级分解)

* 第一种方法

上层模块:

- 顶层: 系统模块
- 第一层:
 - 输入模块
 - 变换模块
 - 输出模块

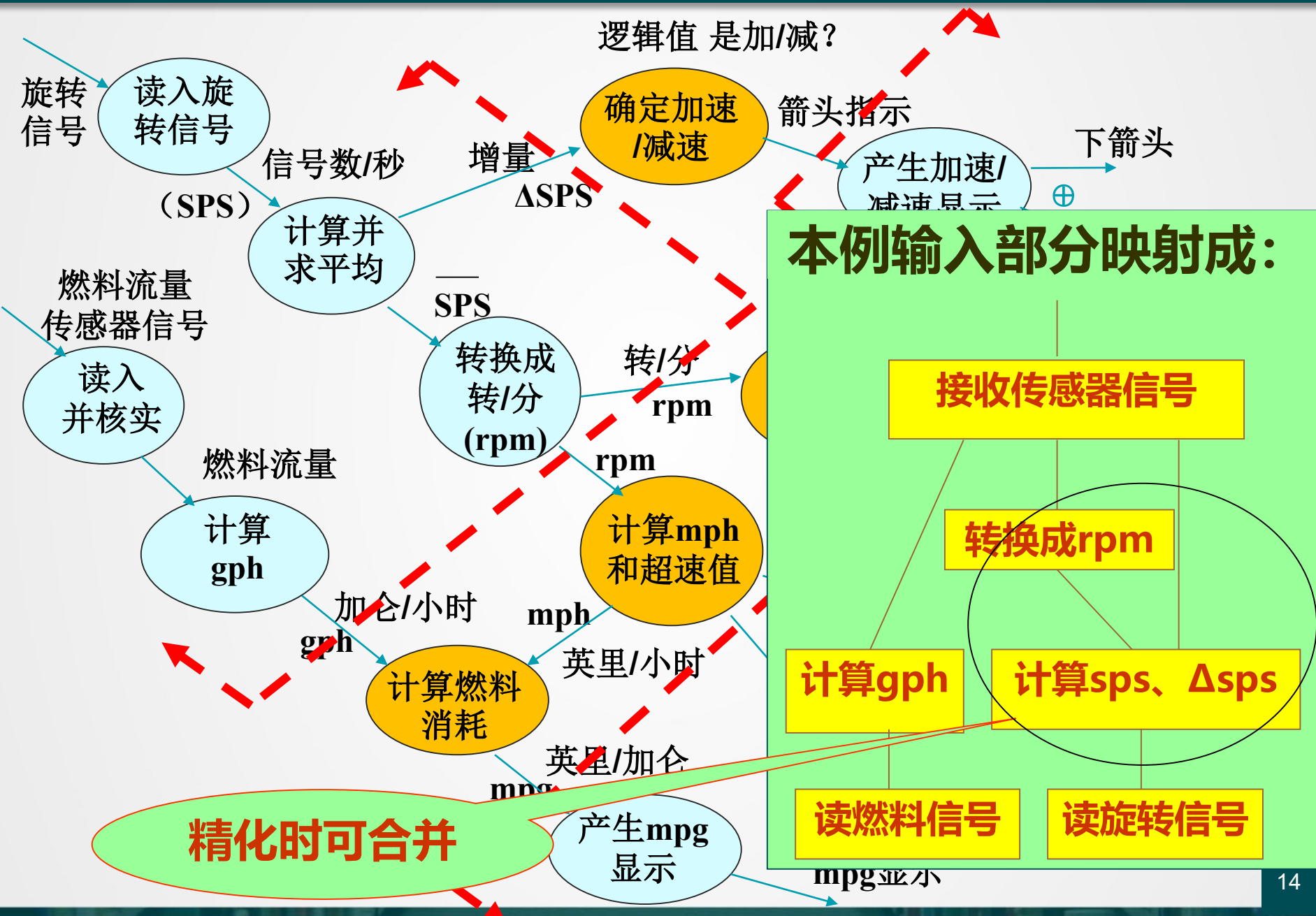
本例则为:



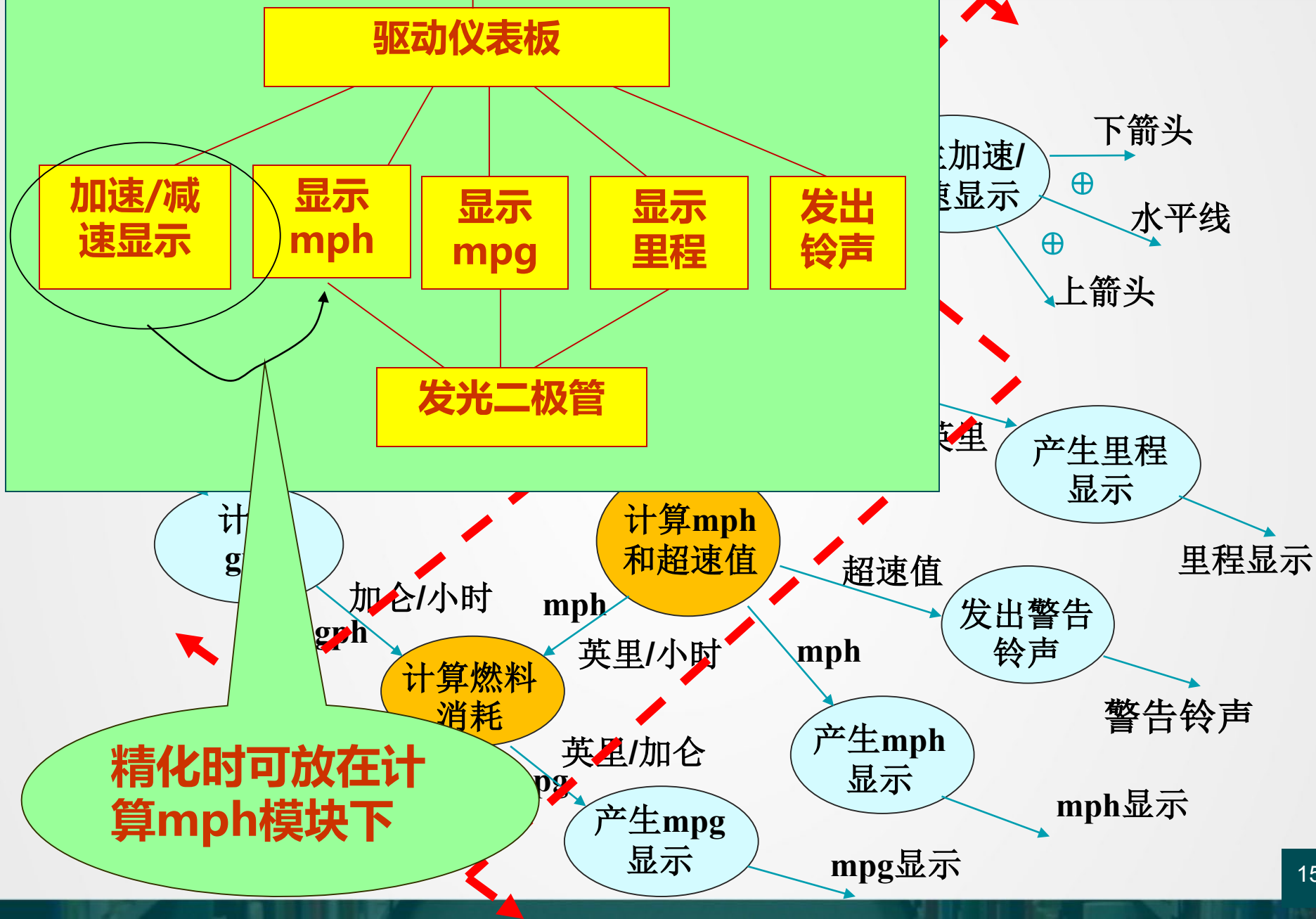
第6步 设计下层模块（完成第二级分解）

* 第一种方法

- 沿着输入通路向外移动，把输入通路中每个处理映射成软件结构中Ca控制下的一个低层模块。
- 沿着输出通路向外移动，把输出通路中的每一个模块映射成Ce控制下的一个低层模块。
- 把变换部分中每个处理，映射成软件结构中Ct控制下的若干模块。



本例输出部分映射成:



变换部分映射成:

变换模块

确定加速
/减速

计算
mph

计算
mpg

计算
里程

逻辑值 是加/减?

确定加速
/减速

箭头指示

产生加速/
减速显示

下箭头

水平线

上箭头

转/分
rpm

计算
里程

英里

产生里程
显示

里程显示

计算mph
和超速值

超速值

发出警告
铃声

警告铃声

产生mph
显示

mph显示

产生mpg
显示

mpg显示

加仑/小时
gph

计算燃料
消耗

英里/小时
mph

英里/加仑
mpg

精化时可放
在计算mph
模块下

第7步 精化模块结构

在输入结构中：

计算rpm、计算 Δ sps、采集sps可以合并。

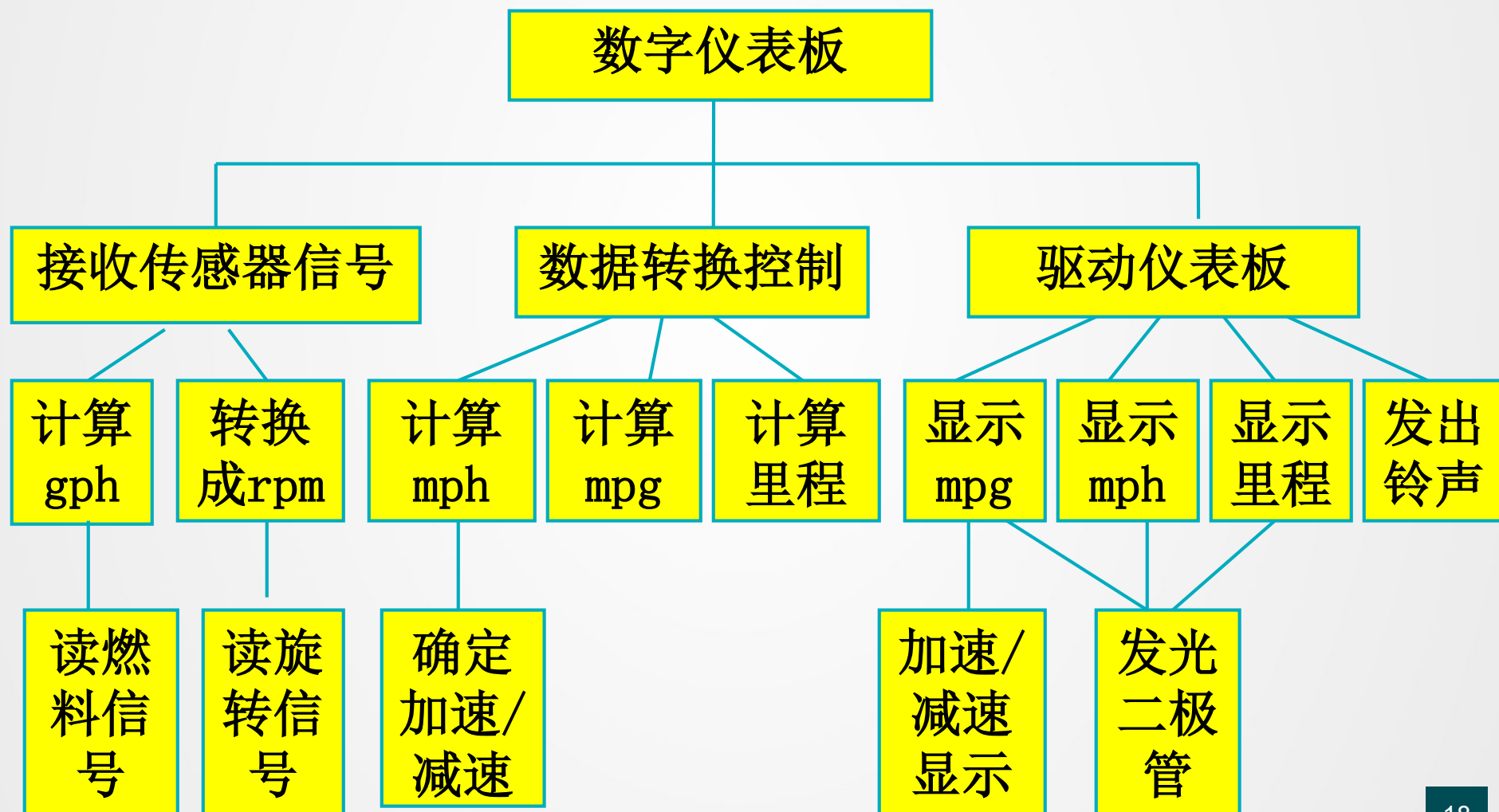
在输出结构中：

加速/减速模块可以放在显示mph模块下。

在变换结构中：

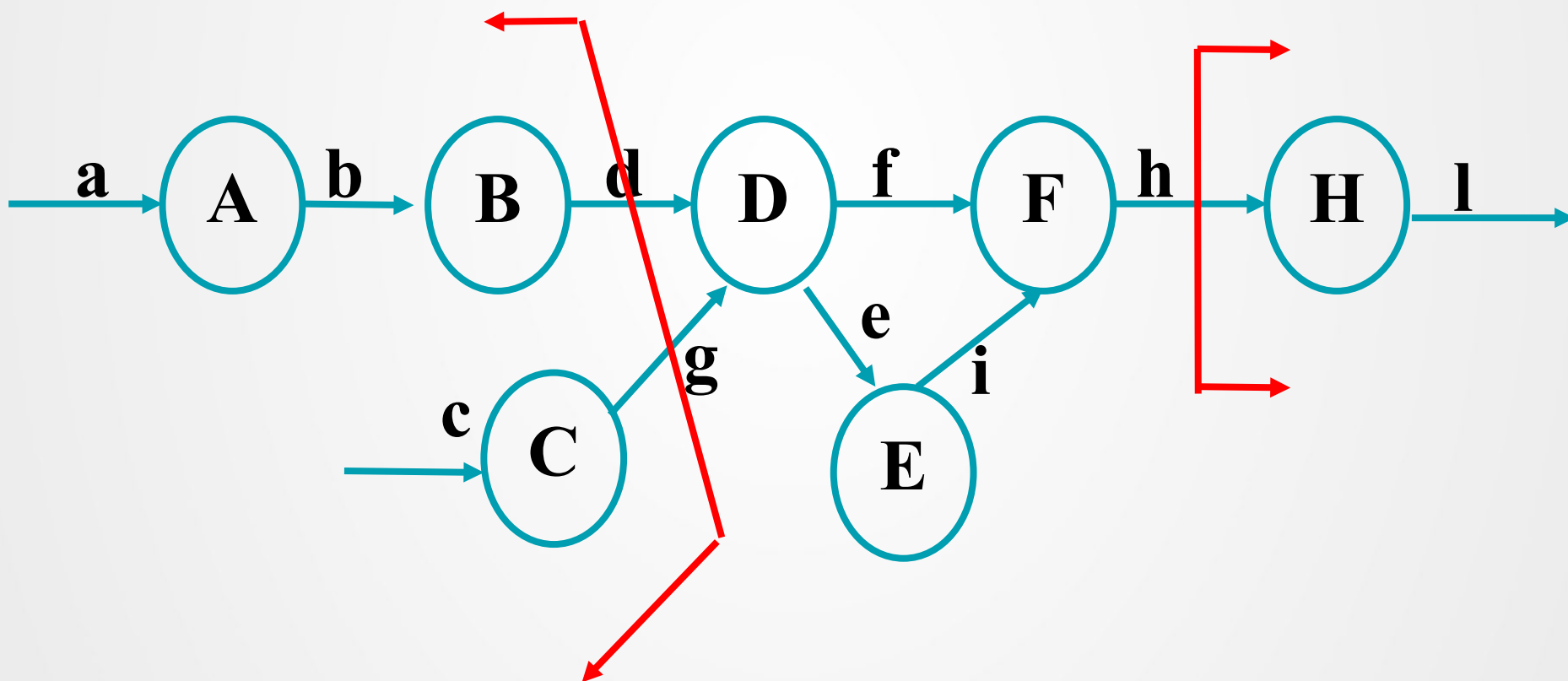
确定加速/减速模块可以放在计算mph模块之下，以减少耦合。

汽车数字仪表盘系统的软件结构

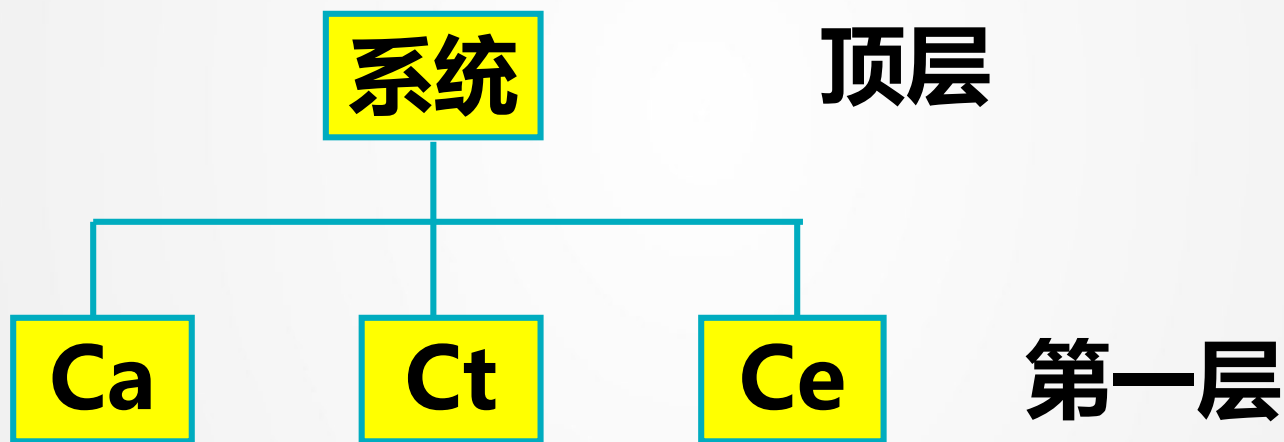
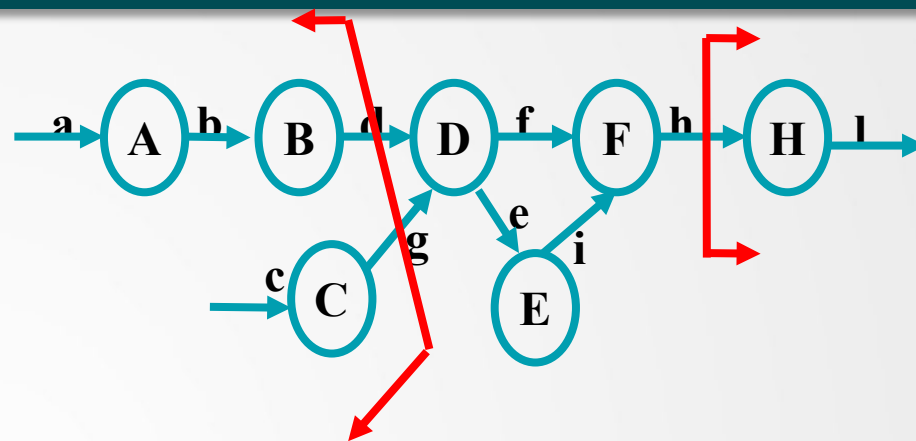


课堂练习1

将以下变换型数据流图映射成软件结构



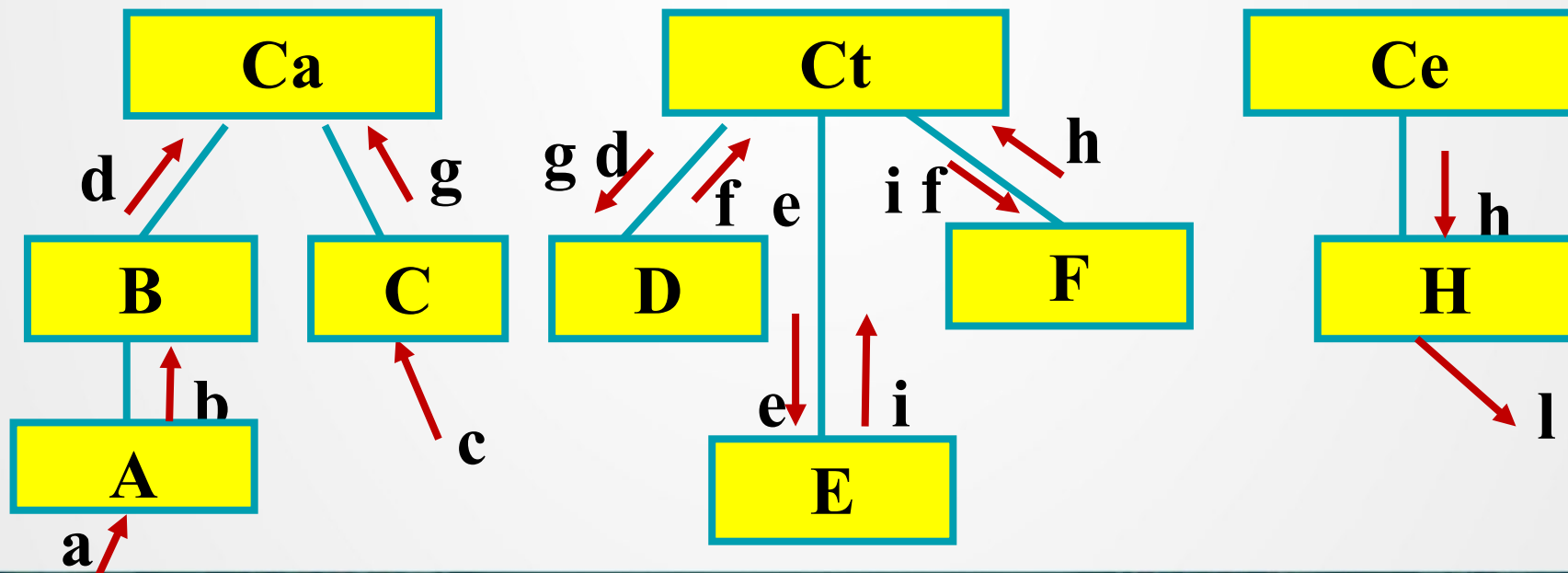
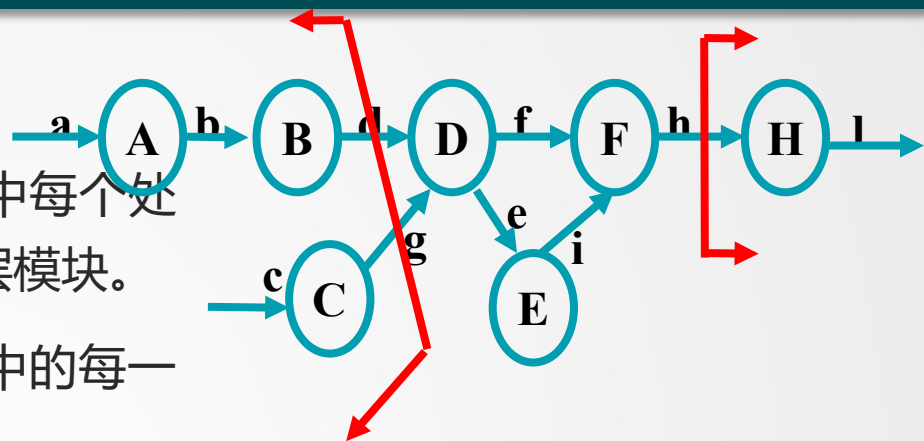
第5步 设计系统的上层模块 (完成第一级分解)



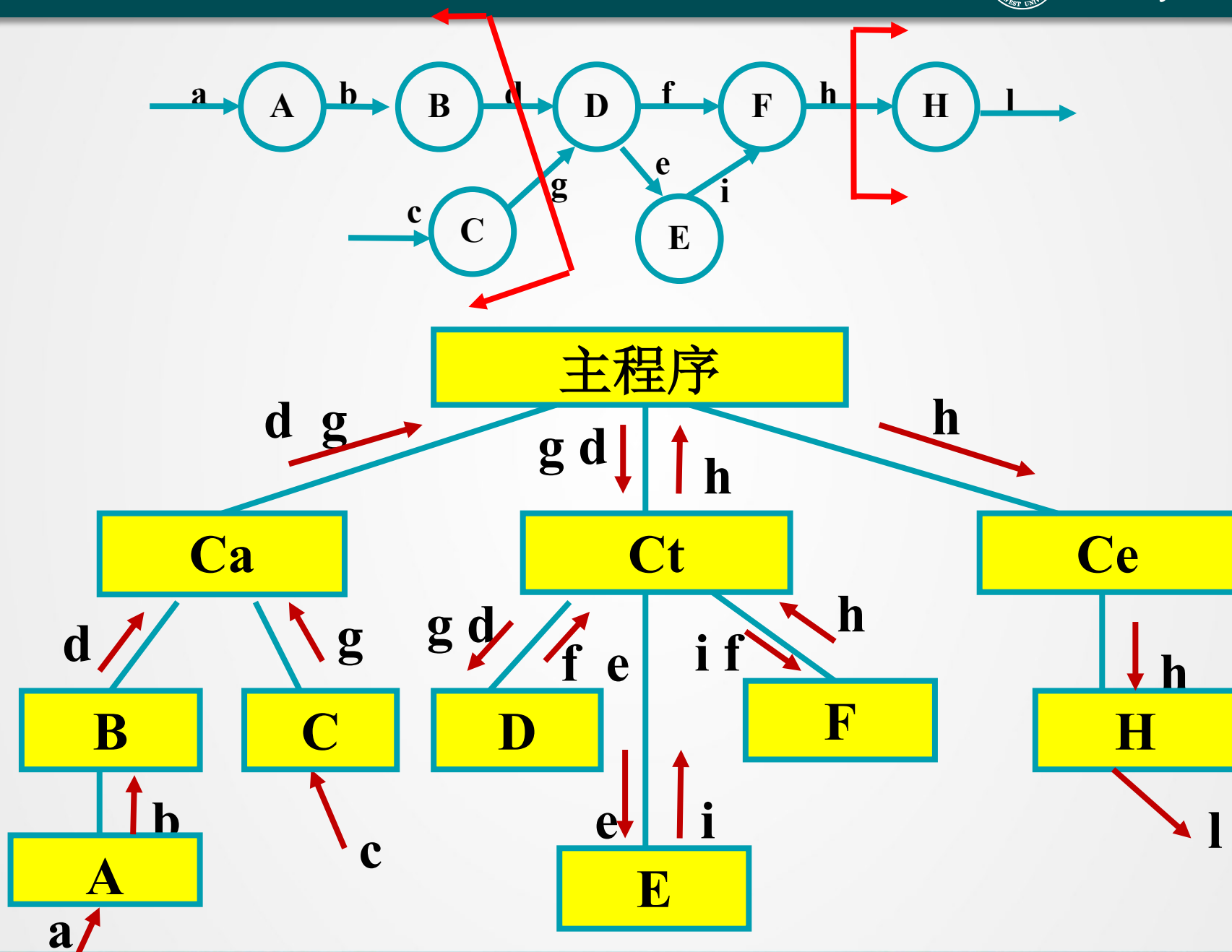
上层模块的作用是协调和控制下层模块并接收和发送数据，所以上层模块通常称为控制模块。

第6步 设计下层模块

- 沿着输入通路向外移动，把输入通路中每个处理映射成软件结构中Ca控制下的一个低层模块。
- 沿着输出通路向外移动，把输出通路中的每一个模块映射成Ce控制下的一个低层模块。
- 把变换部分中每个处理，映射成软件结构中Ct控制下的若干模块。

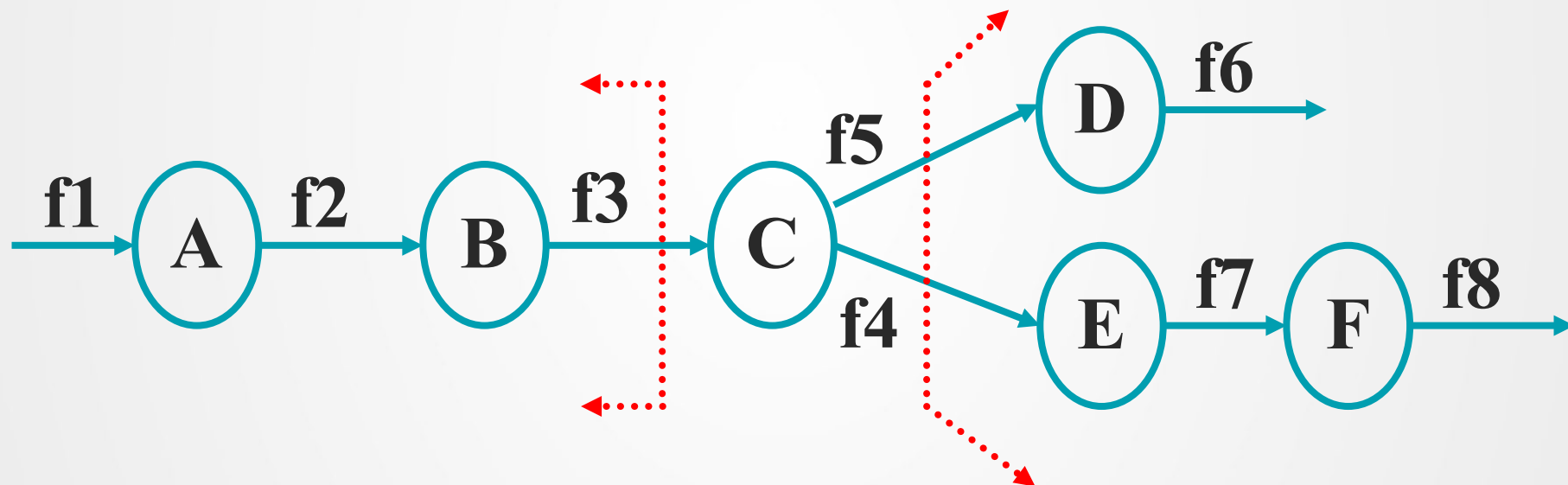


面向数据流的设计举例



课堂练习2

将下面的数据流图映射为软件结构图。



第5步 设计系统的上层模块 (完成第一级分解)

* 第二种方法

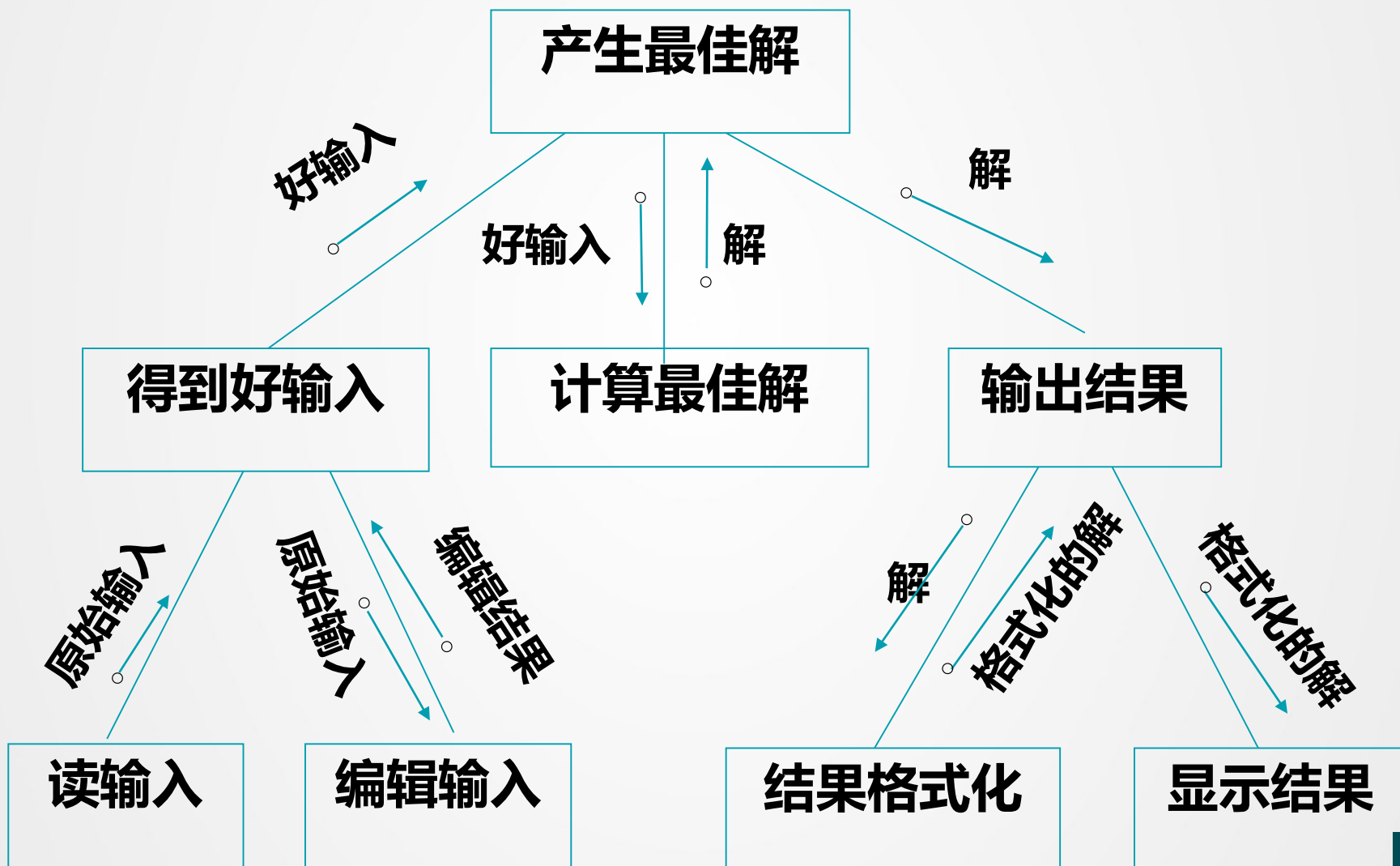
1. 顶层模块的设计同第一种方法。
2. 第一层模块的设计
 - 为每个逻辑输入设计一个输入模块，负责向顶层模块提供数据(逻辑输入)。
 - 为变换流设计一个变换控制模块，以实现所要求的变换功能。
 - 为每个逻辑输出设计一个输出模块，为顶层模块实现数据的输出(逻辑输出)。

第6步 设计下层模块（完成第二级分解）

* 第二种方法

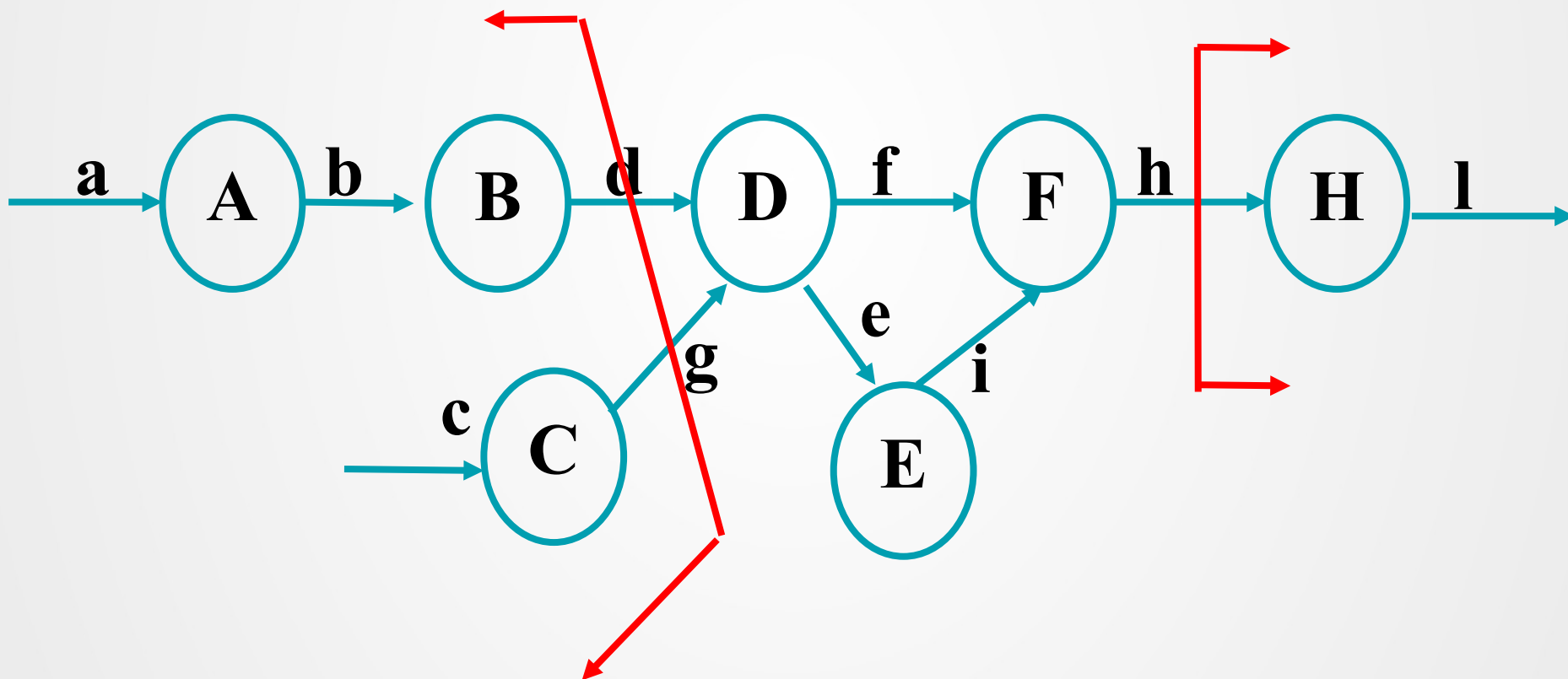
- 为每个**输入**模块设计两个下层模块：
 - 接收模块 • 变换模块
- 为每个**输出**模块设计两个下层模块：
 - 变换模块 • 输出模块
- **变换模块**下属模块的设计：
一个处理建立一个功能模块。

例：产生最佳解的结构图



例：课堂练习1

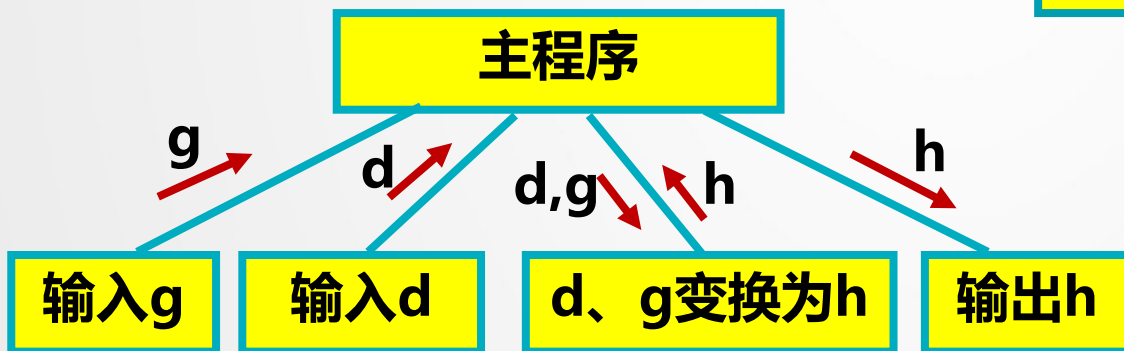
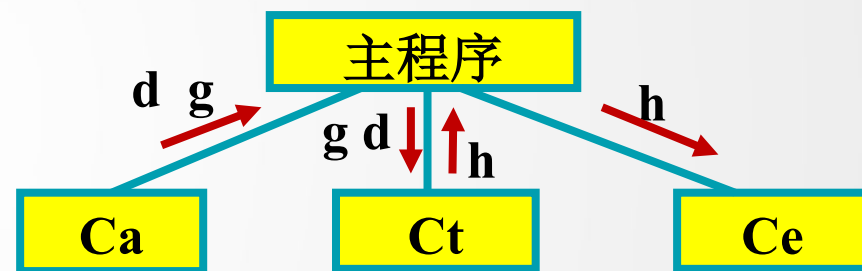
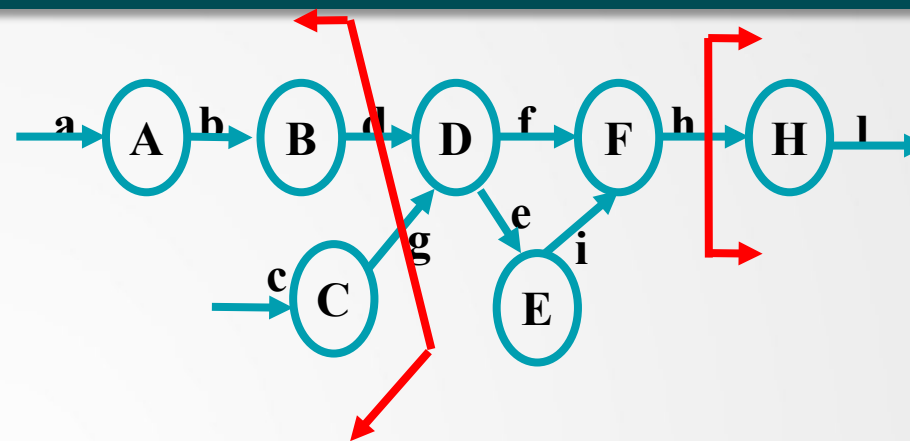
用第二种方法将以下变换型数据流图映射成软件结构



面向数据流的设计举例

第二种方法:

1. 顶层模块的设计同第一种方法。
2. 第一层模块的设计:
 - 为**每个逻辑输入**设计一个输入模块, 负责向顶层模块提供数据(逻辑输入)。
 - 为**变换流**设计一个变换控制模块, 以实现所要求的变换功能。
 - 为**每个逻辑输出**设计一个输出模块, 为顶层模块实现数据的输出(逻辑输出)。

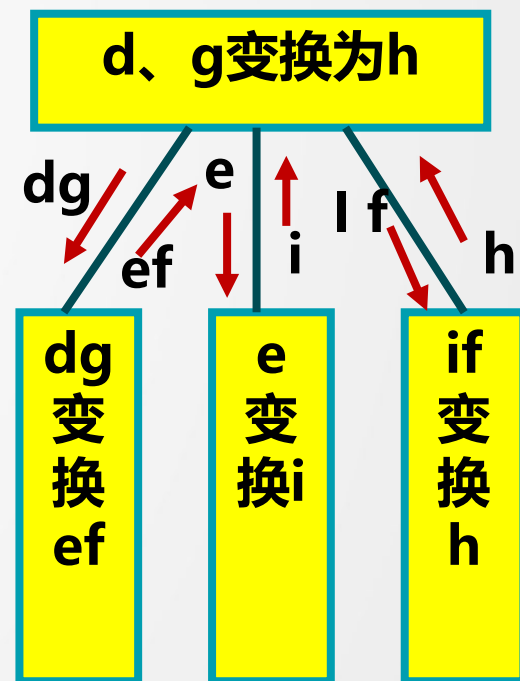
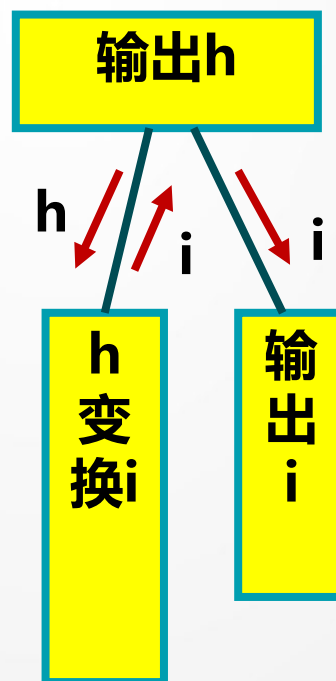
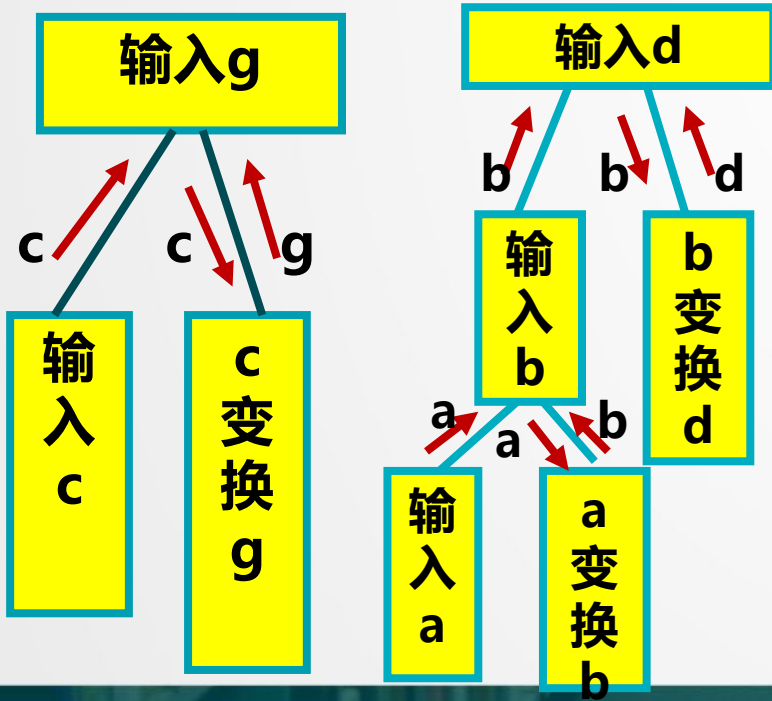
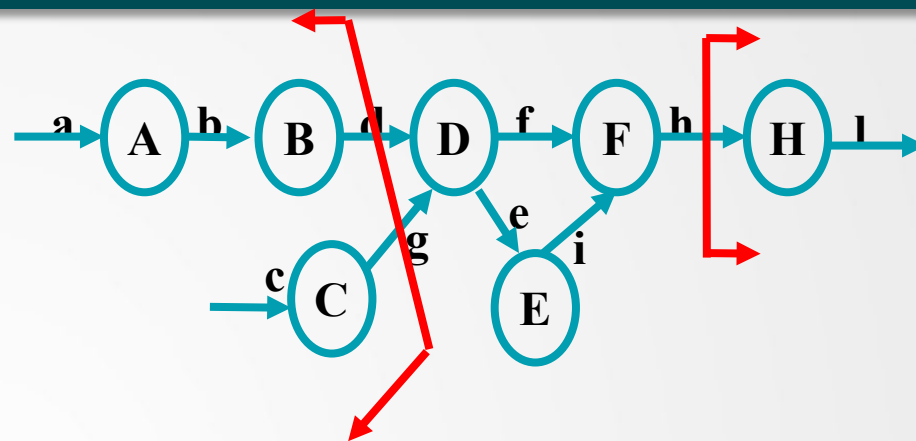


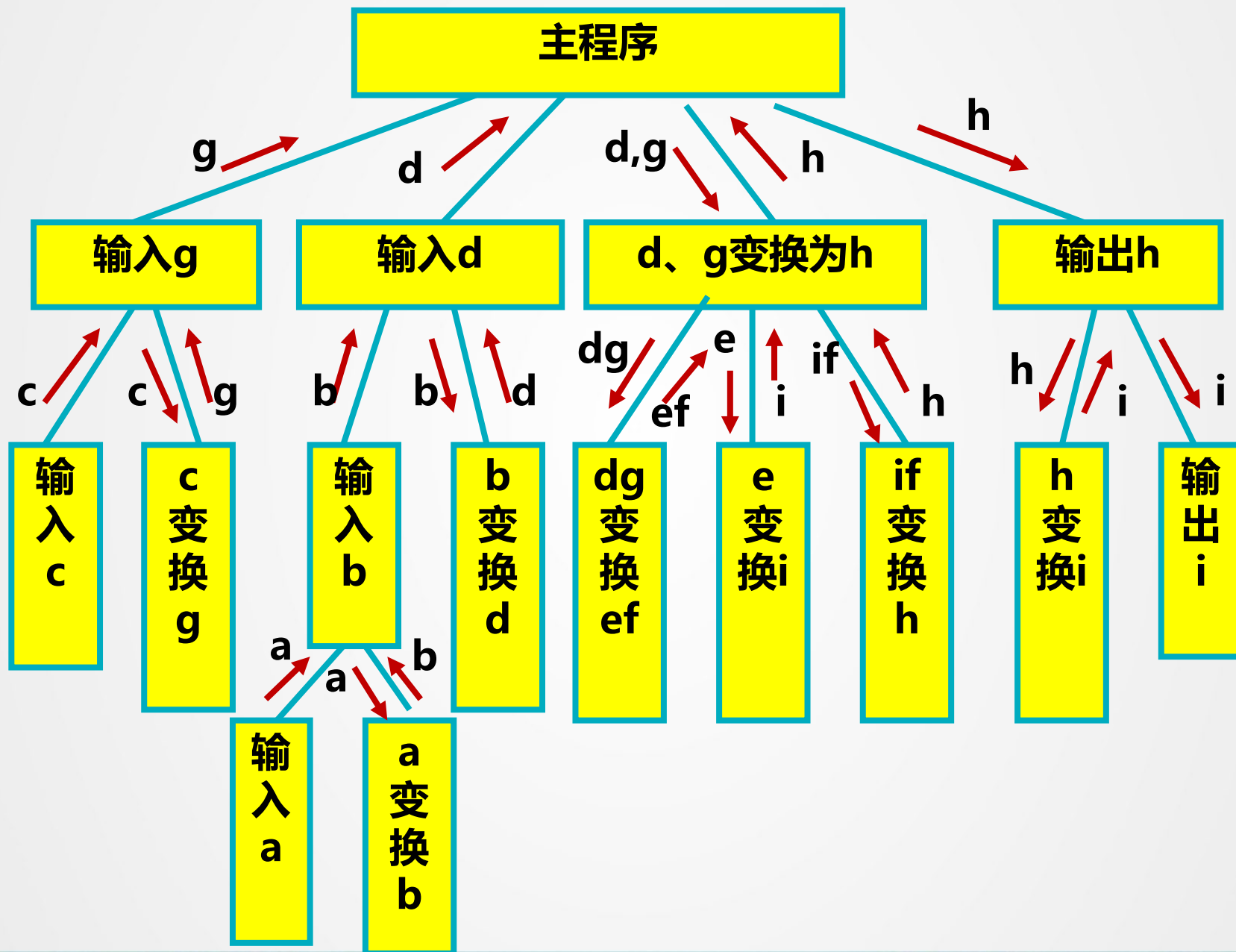
面向数据流的设计举例

第二种方法:

3. 设计下层模块:

- 为每个**输入**模块设计两个下层模块:
 - 接收模块 • 变换模块
- 为每个**输出**模块设计两个下层模块:
 - 变换模块 • 输出模块
- **变换模块**下属模块的设计:
 - 一个处理建立一个功能模块。





两种映射方法的比较：

设计上层时：

当**数据流图的逻辑输入（出）多时**，第一种方法优于第二种方法。因为它可以减少顶层模块的扇出并降低主控程序的复杂性。

当**逻辑输入（出）少时**，第二种方法优于第一种方法。因为Ca、Ce过于简单，优化时需要与主控模块合并。

两种映射方法的比较：

设计下层时：

第一种方法**对输入流和输出流的每一个处理只直接映射成一个模块**，第二种方法对每个处理要设计两个下层模块（输入、变换或变换、输出）。

第一种方法当功能复杂时，在优化时可能不得不进一步分解，而第二种方法，输入模块和变换模块，或变换模块和输出模块过于简单，在优化时不得不合并。因此根据具体情况，两种方法灵活使用为宜。

2、事务分析 例子

设计步骤：

第1步 复查基本系统模型。

第2步 复查并精化数据流图。

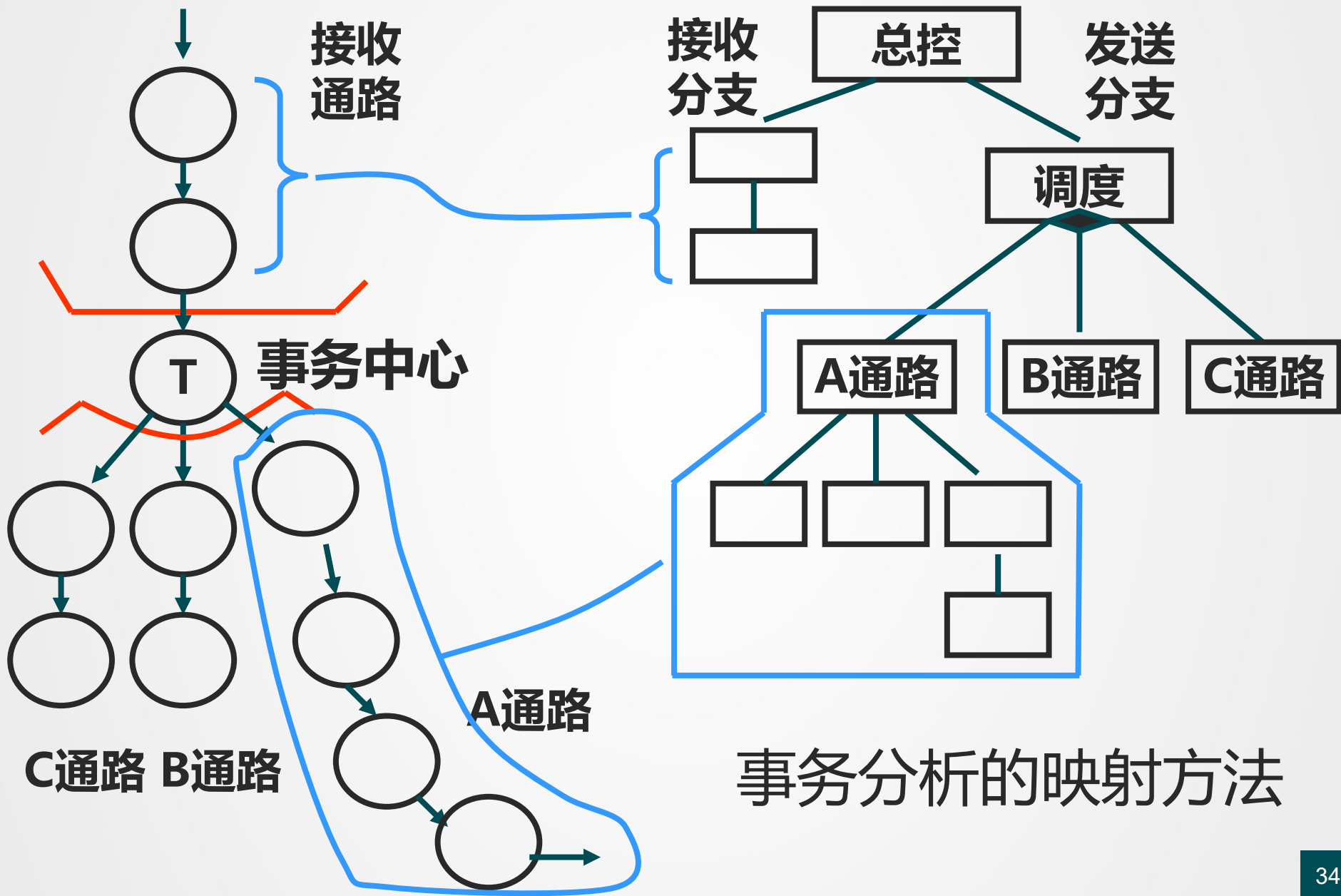
第3步 确定数据流类型。

第4步 确定接收部分和发送部分的边界。

第5步 设计系统的上层模块结构（第一级分解）。

第6步 设计下层模块（完成第二级分解）。

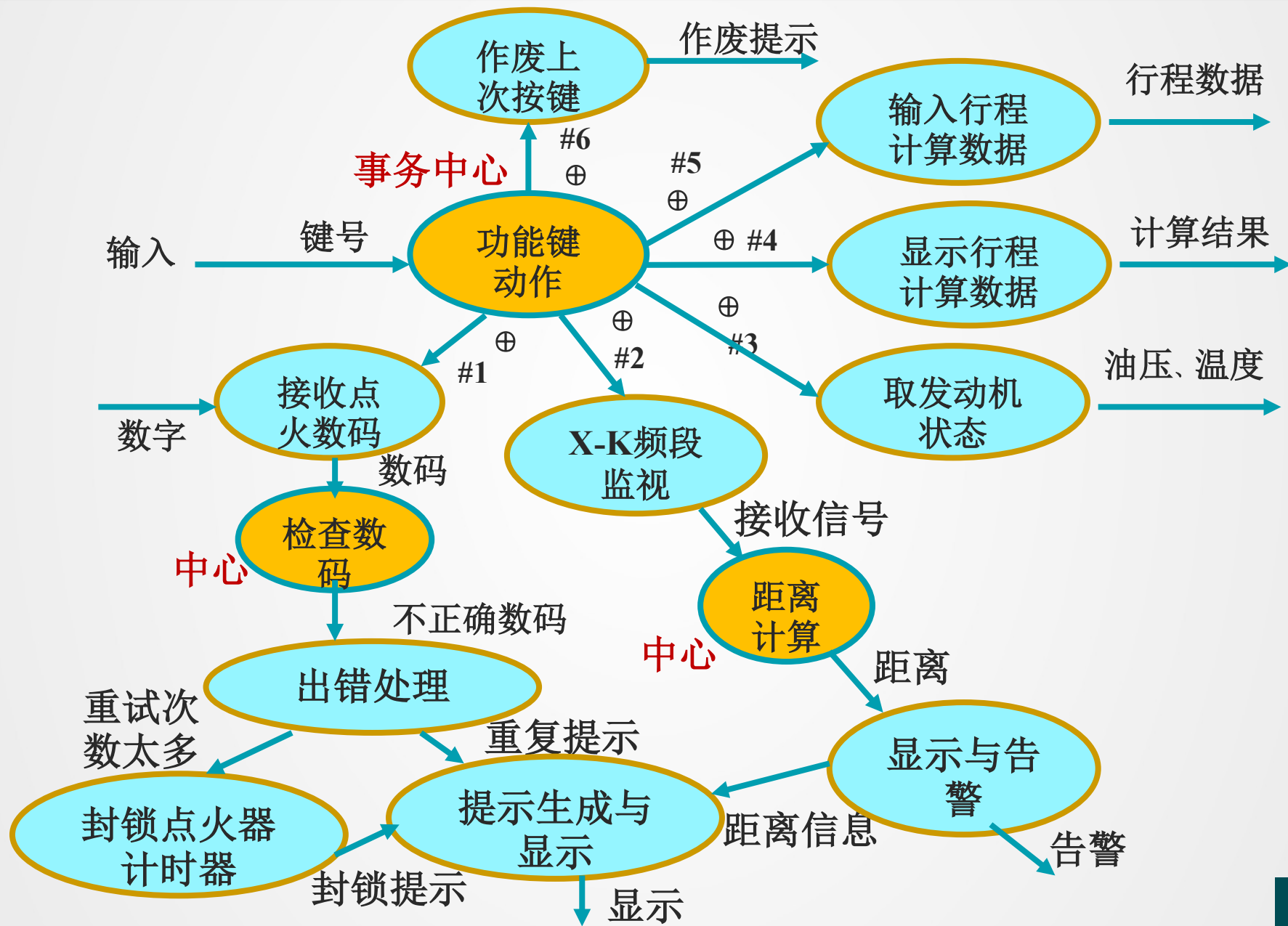
第7步 精化模块结构



事务分析-例子

例：汽车仪表板系统有一个电子点火钥匙和一个功能选择与显示装置，用来选择某些性能。通过一组按键产生一串点火启动数字来代替普通的点火钥匙，这组按键的选择功能如下：

1. 点火，启动汽车。除了这个点火按钮外，还有一排数字按钮，事先可设置一个数字密码，只有密码对上了，汽车才能安全启动。
2. 雷达探测。它可以给出障碍物的距离，如果距离太近就发出警告，以防止事故的发生。
3. 显示发动机的工作状态。
4. 启动行程计算器，输入行程计算数据。
5. 显示行程，计算信息。
6. 清除上一键。



设计步骤：

第1步 复查基本系统模型

确保系统的输入数据和输出数据符合实际。

第2步 复查并精化数据流图。

物理输入是欲选择的某种功能键，按键并检查后，数据流沿六条路径中的一条移动，本题只给出详细的按键1和按键2的路径，其它路径则只简化为一个处理表示。

第3步 确定数据流类型。

由图可知，**功能键**动作是事务中心，因此属于事务型结构的数据流图。

第4步 确定接收部分和发送部分的边界。

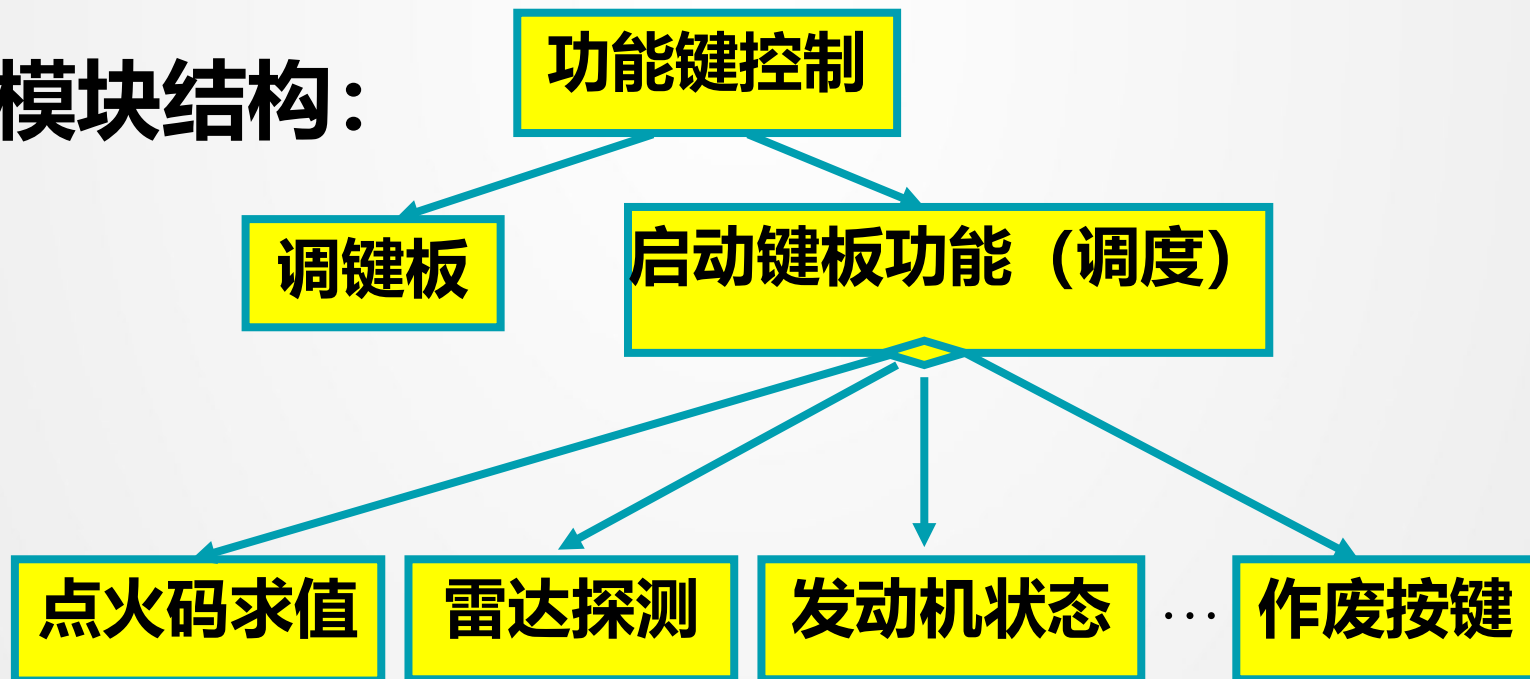
第一、二条路径，又是两个变换流，可按变换流进一步划分子图边界。

设计步骤:

第5步 系统的上层模块结构

事务型结构的数据流图在映射软件结构时包括两个分支，一个为接收分支，一个为发送分支，且发送分支在调度模块的控制下。

上层模块结构:

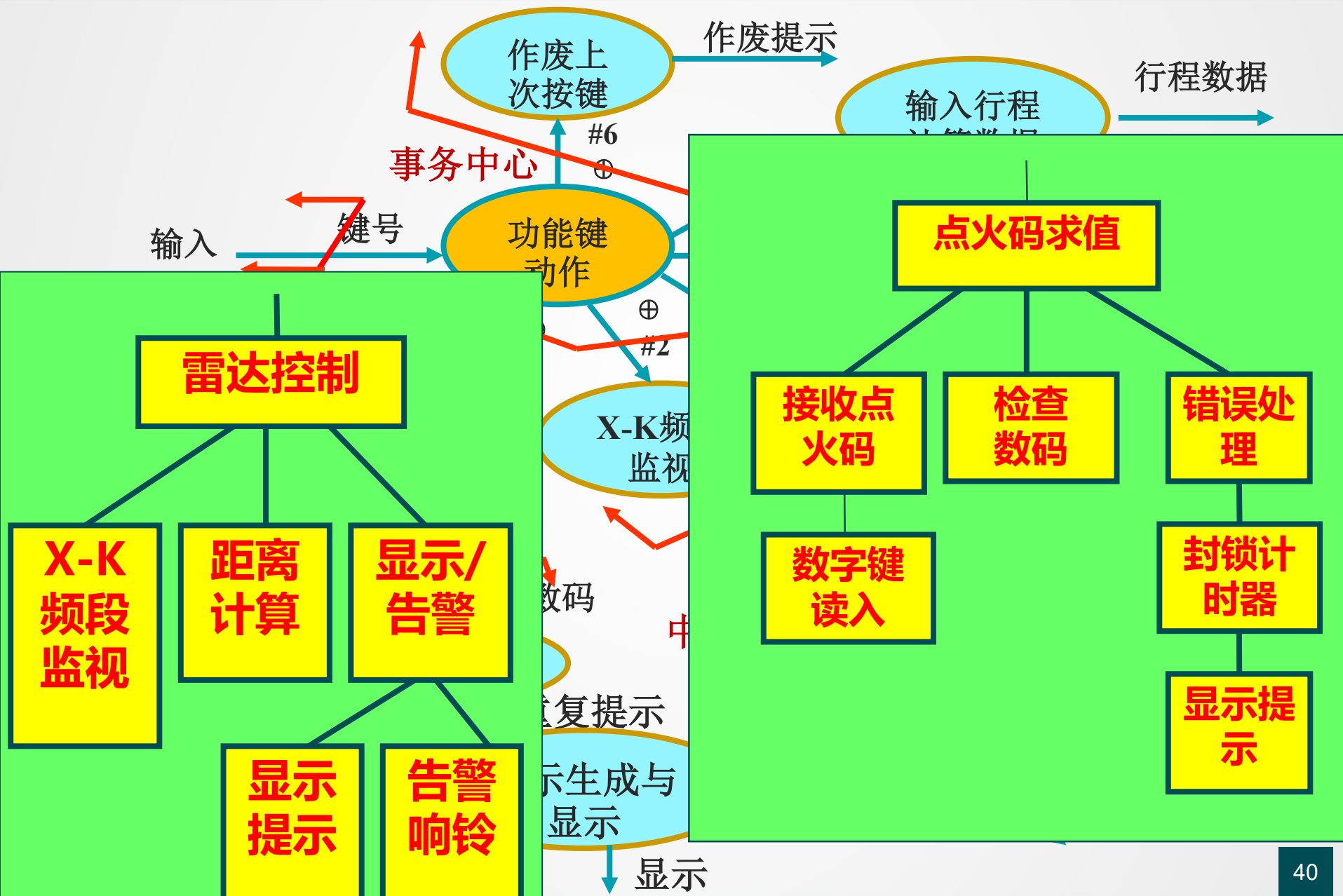


设计步骤：

第6步 设计系统的下层模块

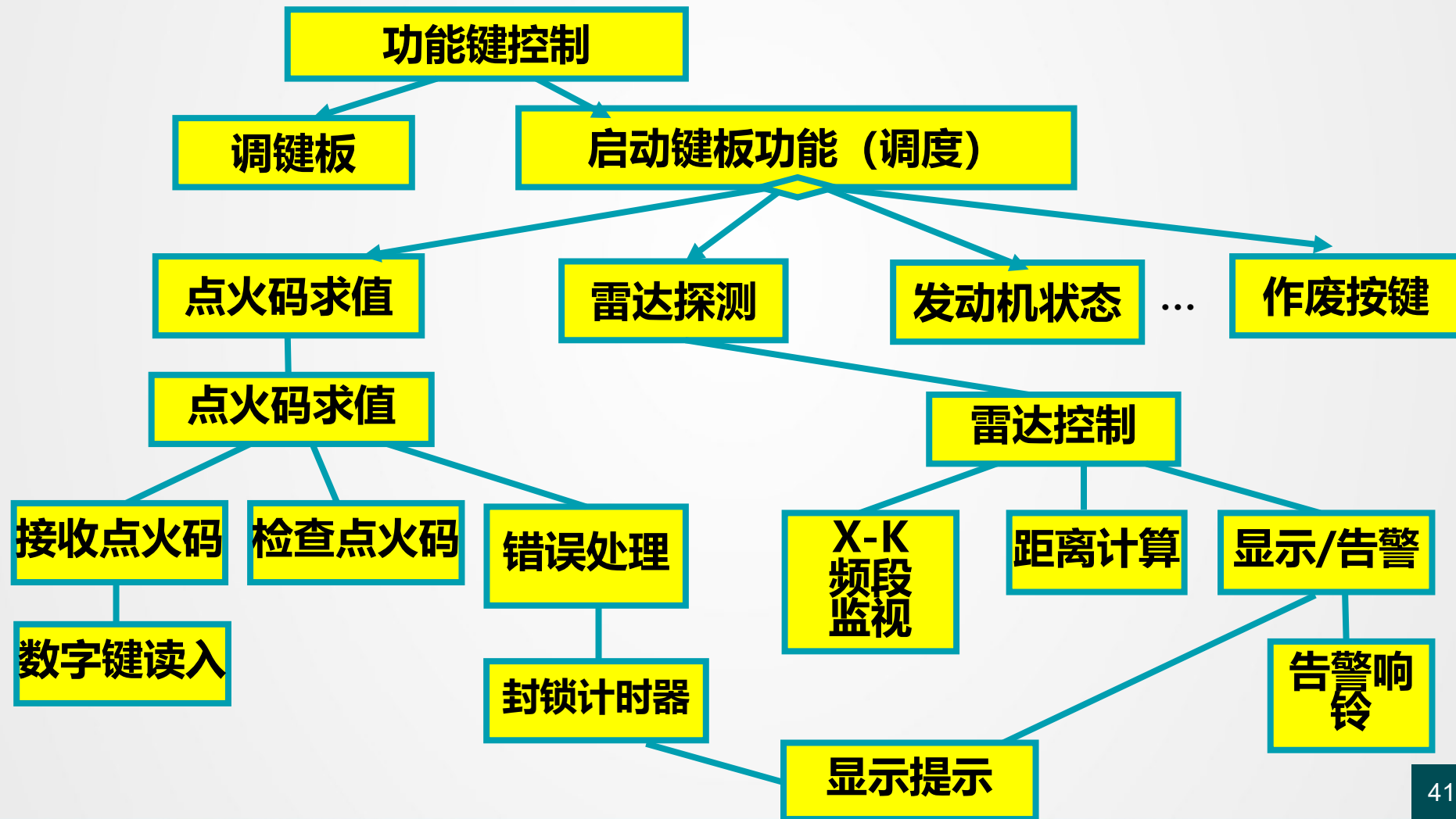
(1) 接收部分的下层模块的设计从事务中心的边界开始，沿着接收通路向外，将处理映射成模块。

(2) 发送分支是在调度模块控制下的下层模块，从事务中心的边界开始，沿着发送通路向外把每个活动流通路径映射成与它的流特征相对应的结构，本题中的1、2两个分支又是变换流。（只给出两个分支）



设计步骤:

第7步 精化模块结构 (将显示模块合并)



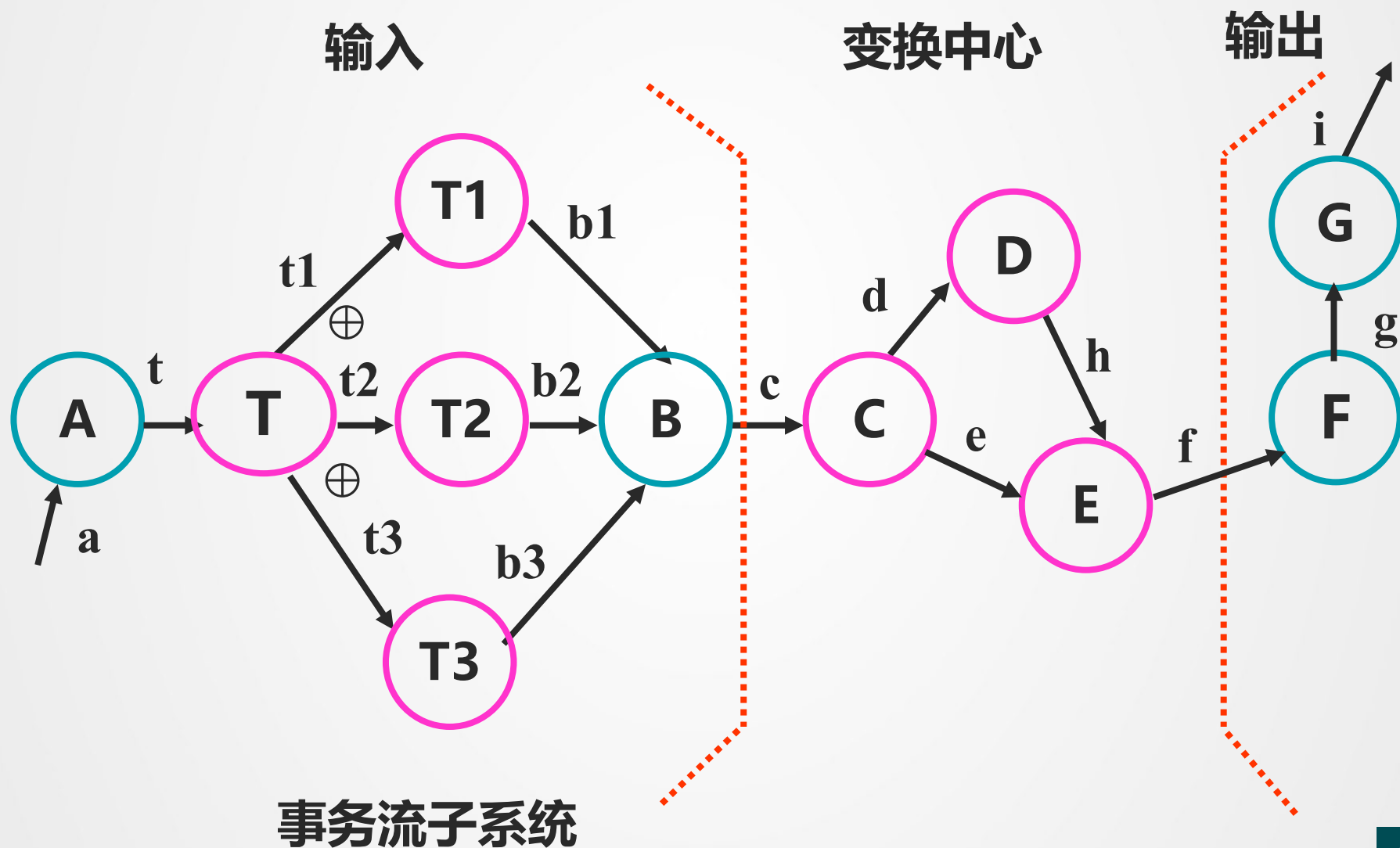
3、综合型数据流图的软件结构设计

设计步骤：

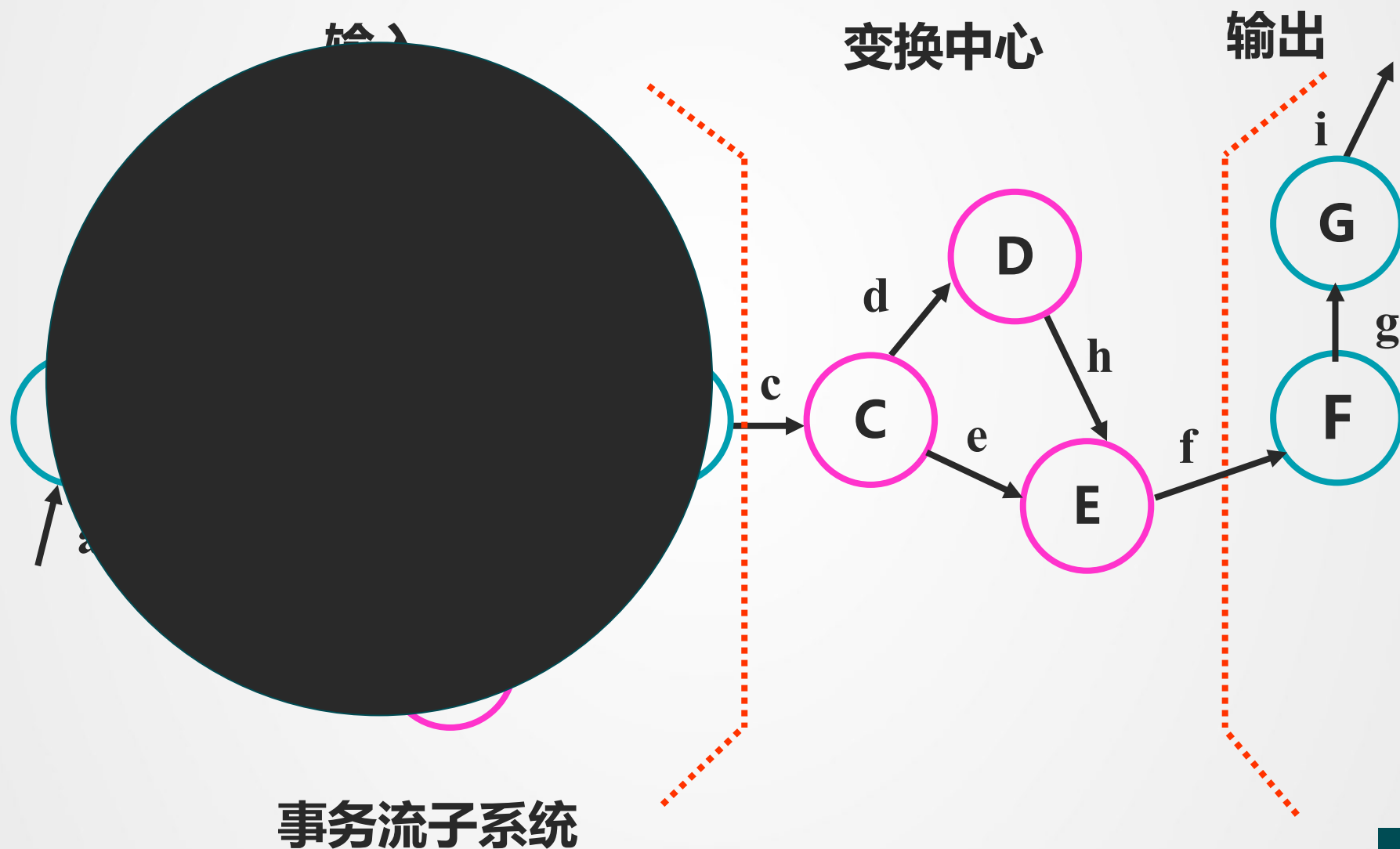
当一个系统的DFD中，既有变换流又有事务流时，这就是综合的数据流图。其软件结构设计方法如下：

- (1) 确定DFD整体上的类型。
- (2) 标定局部DFD的范围，确定其类型。
- (3) 按整体和局部的DFD特征，设计出软件结构。

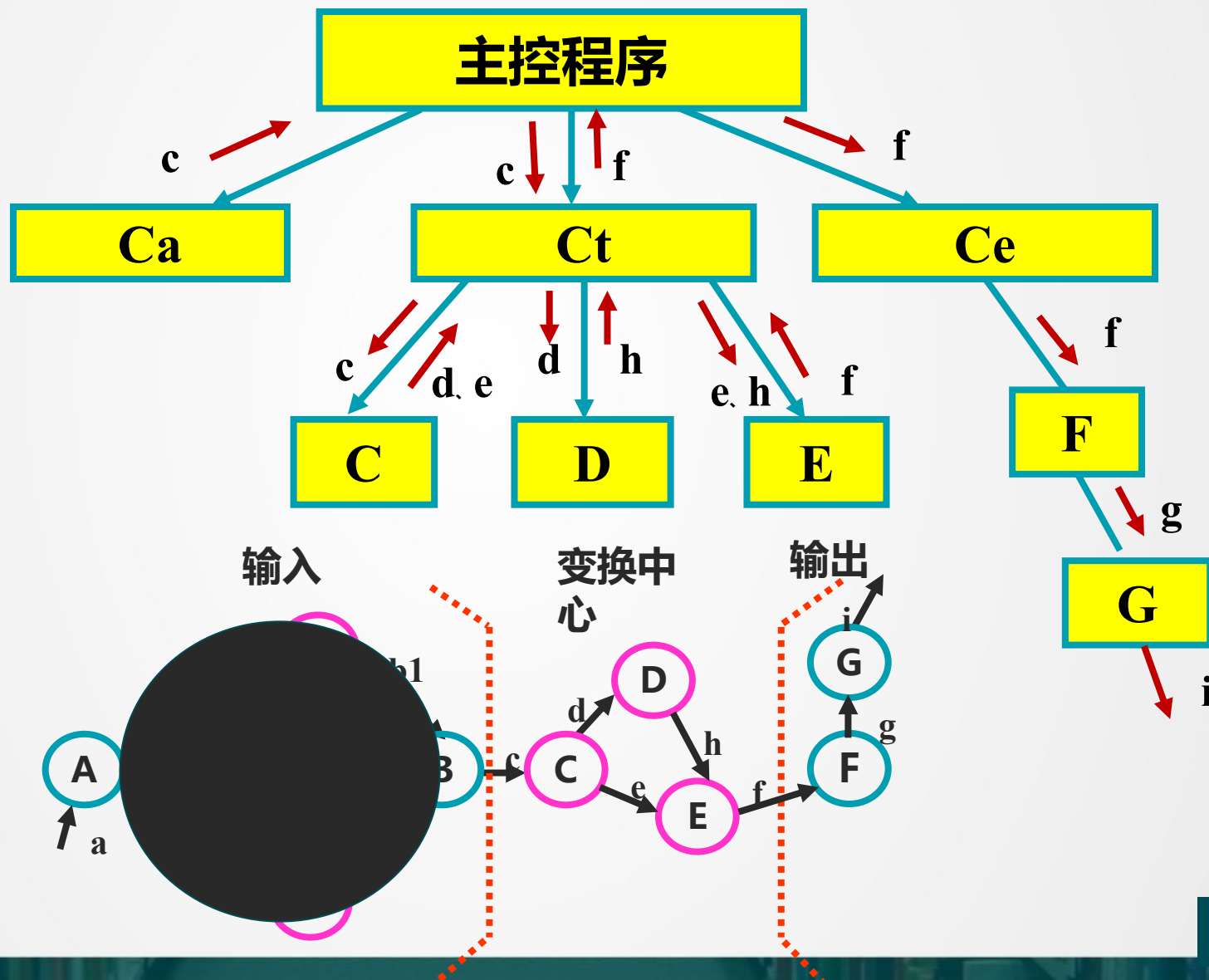
例1：主图变换型 子图事务型



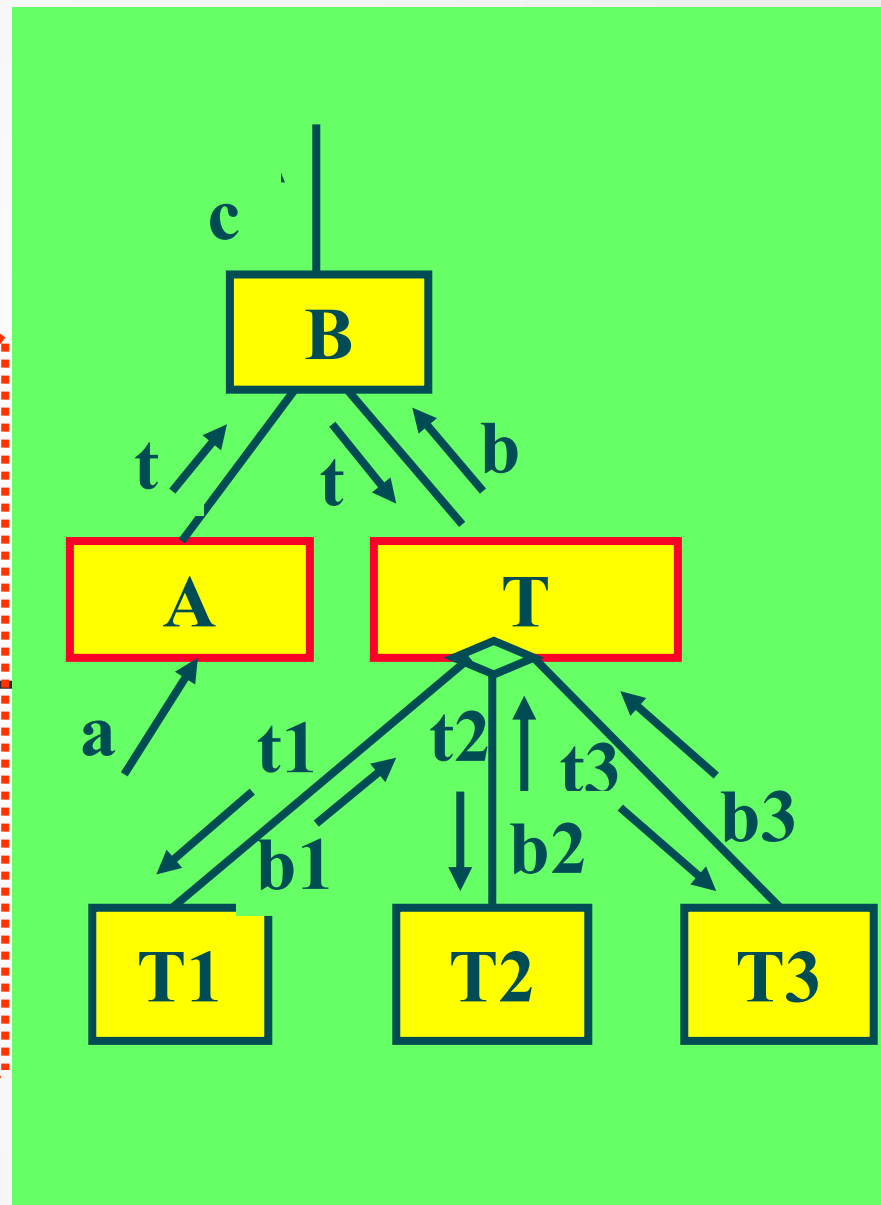
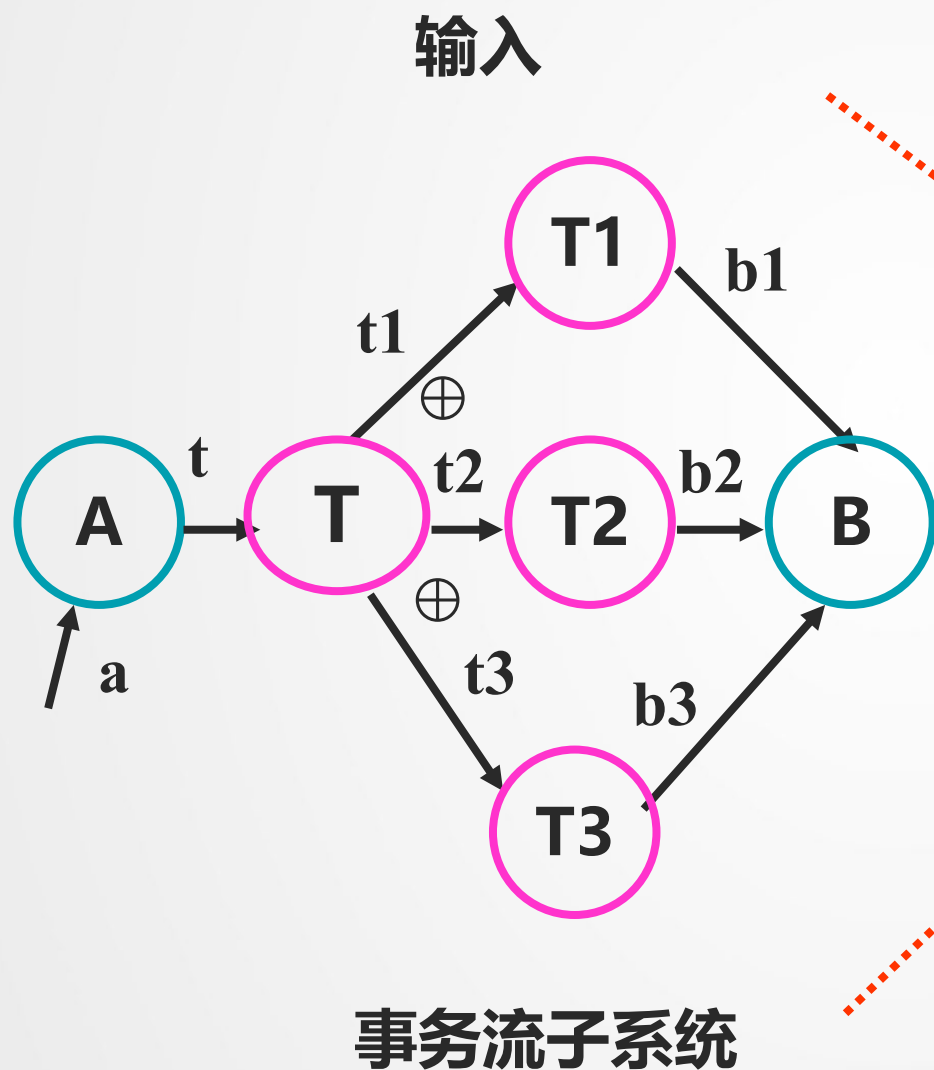
例1：主图变换型 子图事务型



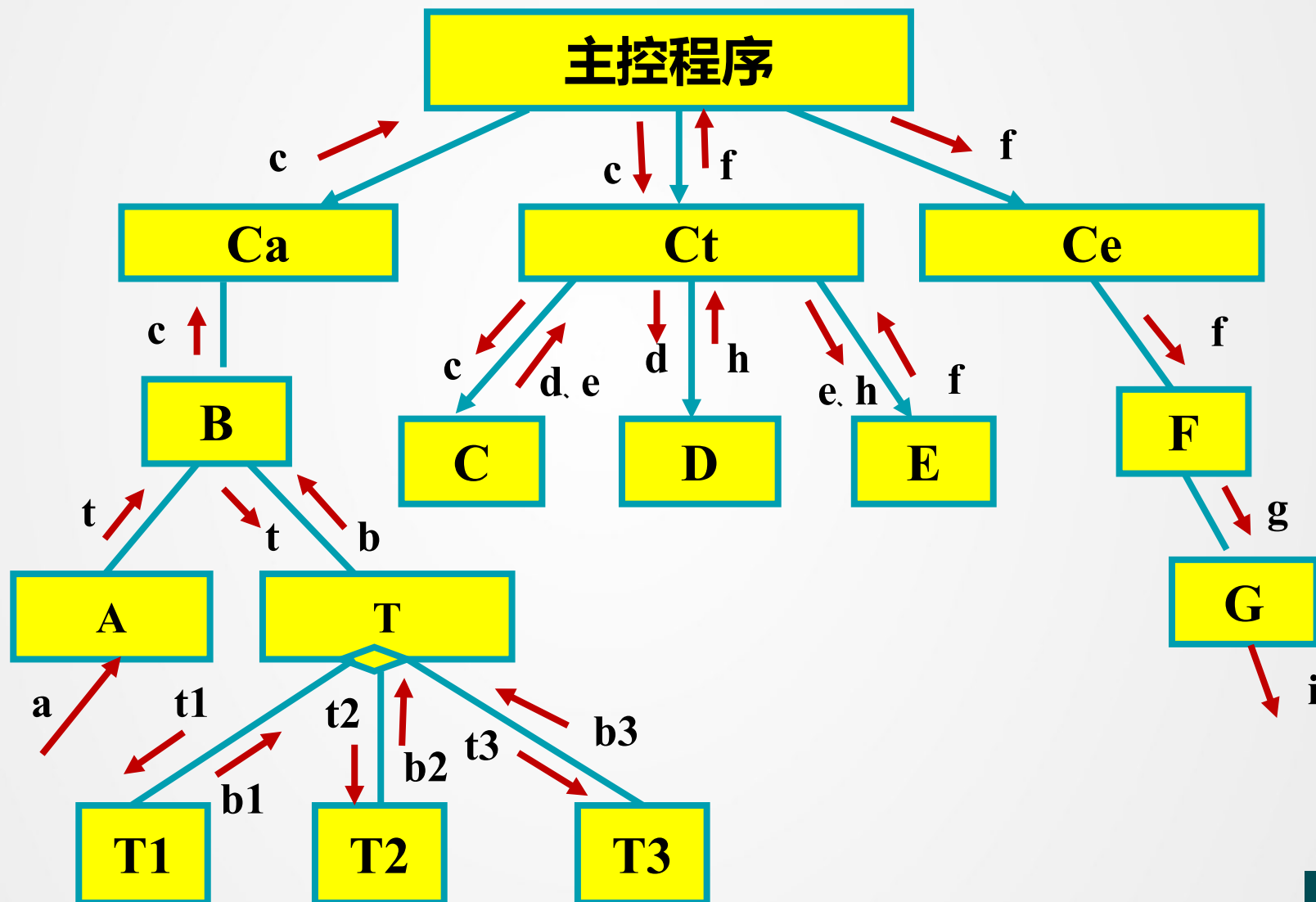
例1：主图变换型 子图事务型



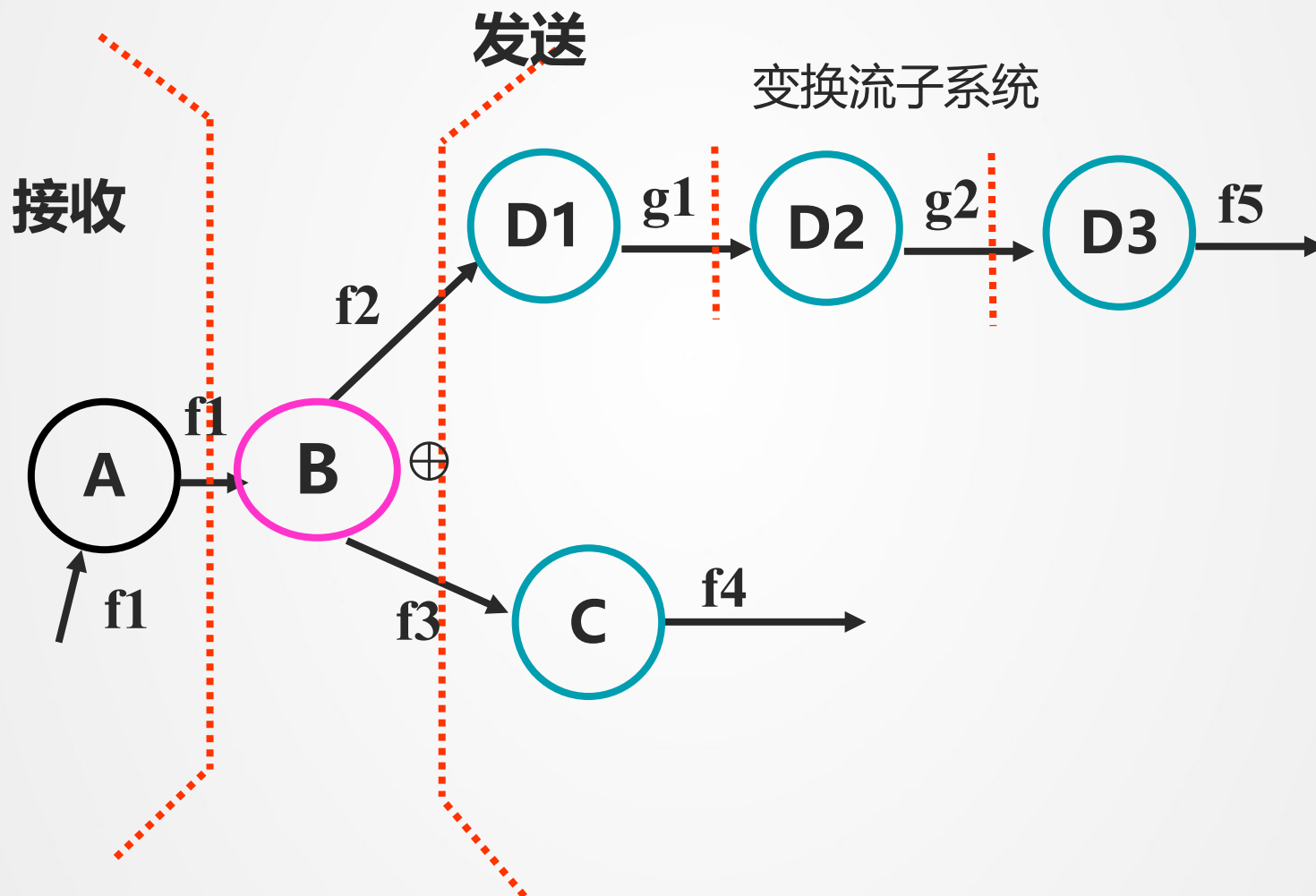
例1：主图变换型 子图事务型



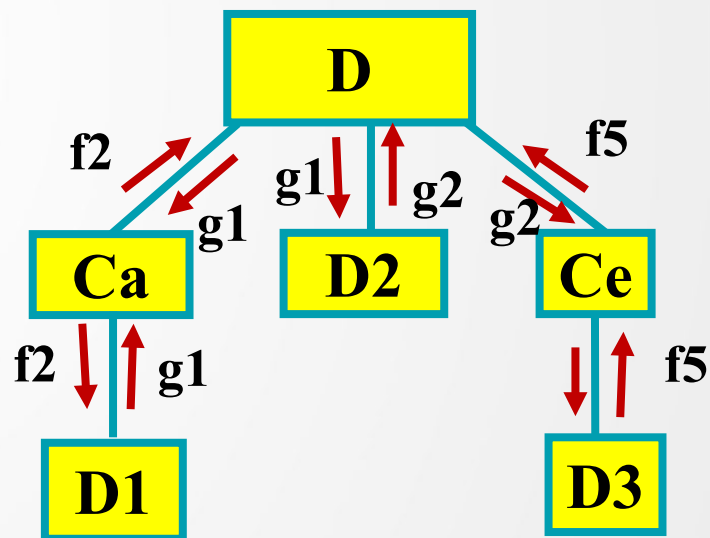
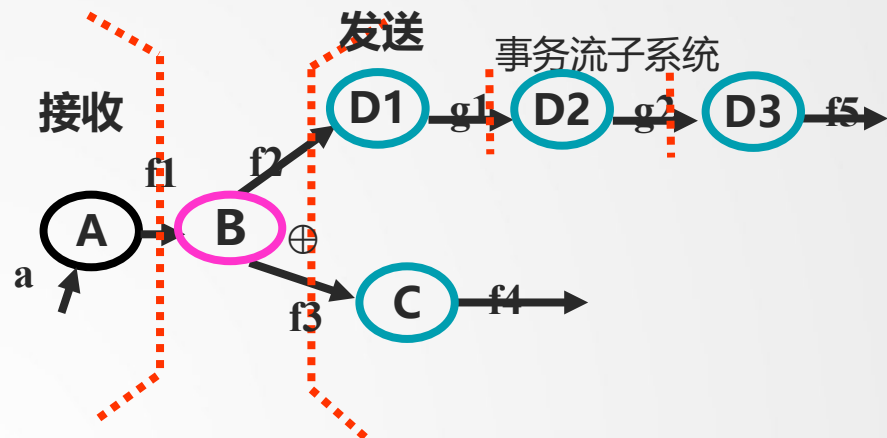
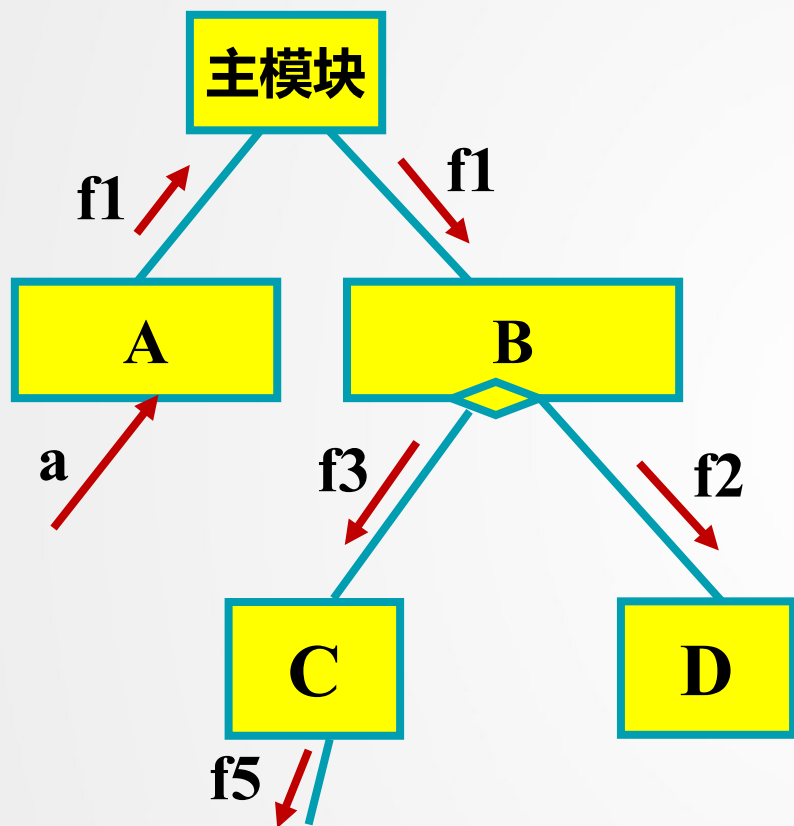
例1：主图变换型 子图事务型



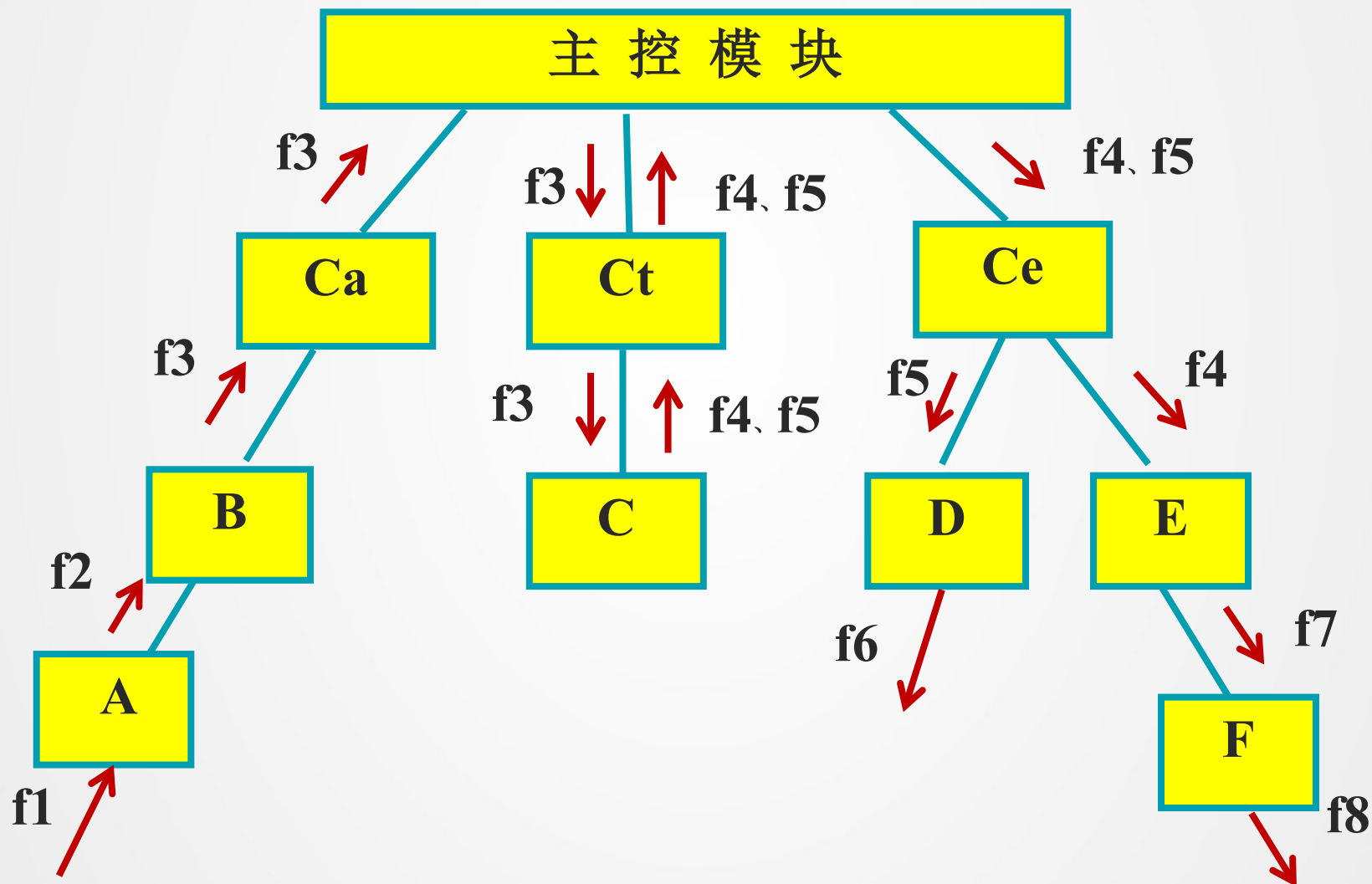
例2：主图事务型 子图变换型



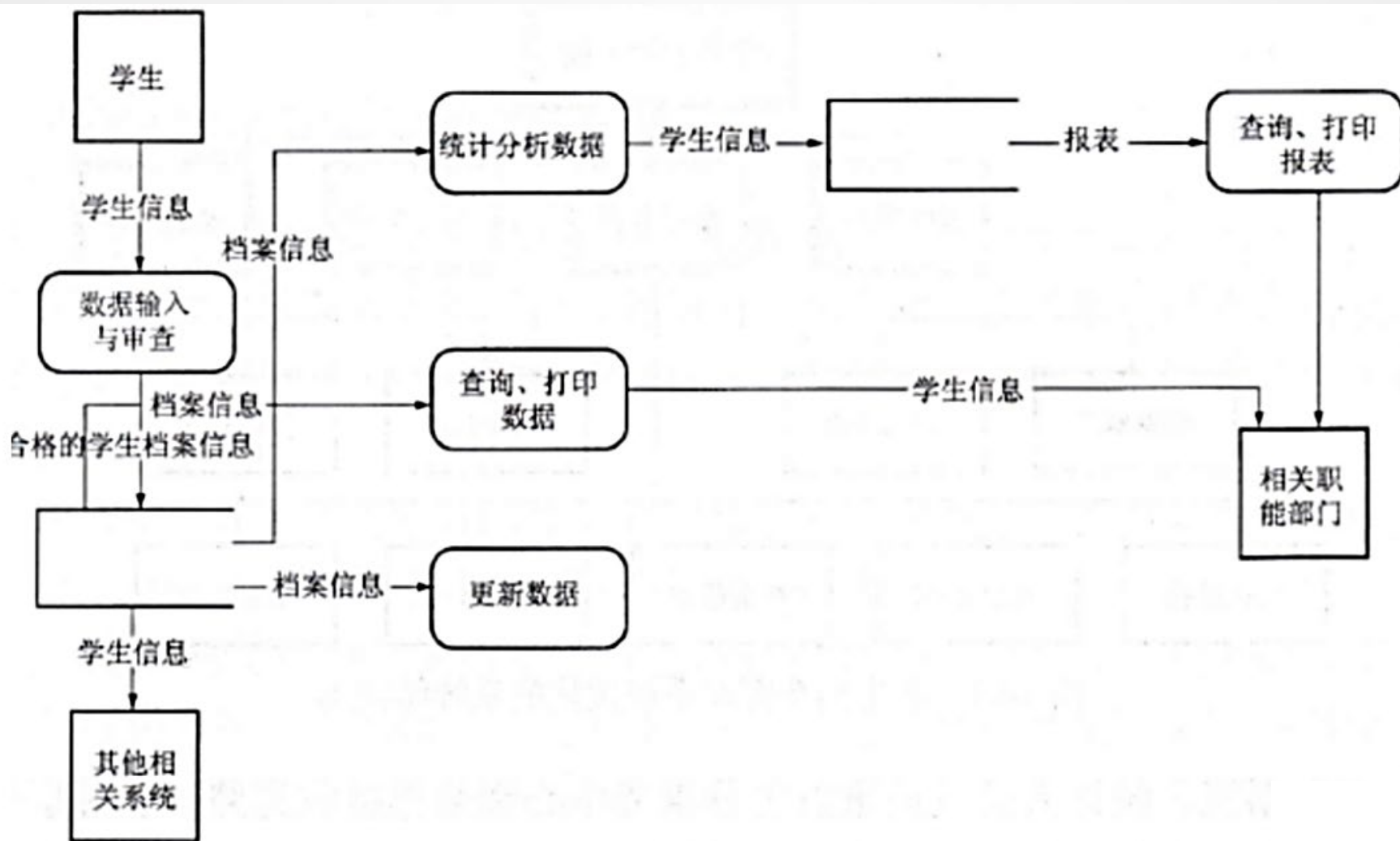
例2：主图事务型 子图变换型



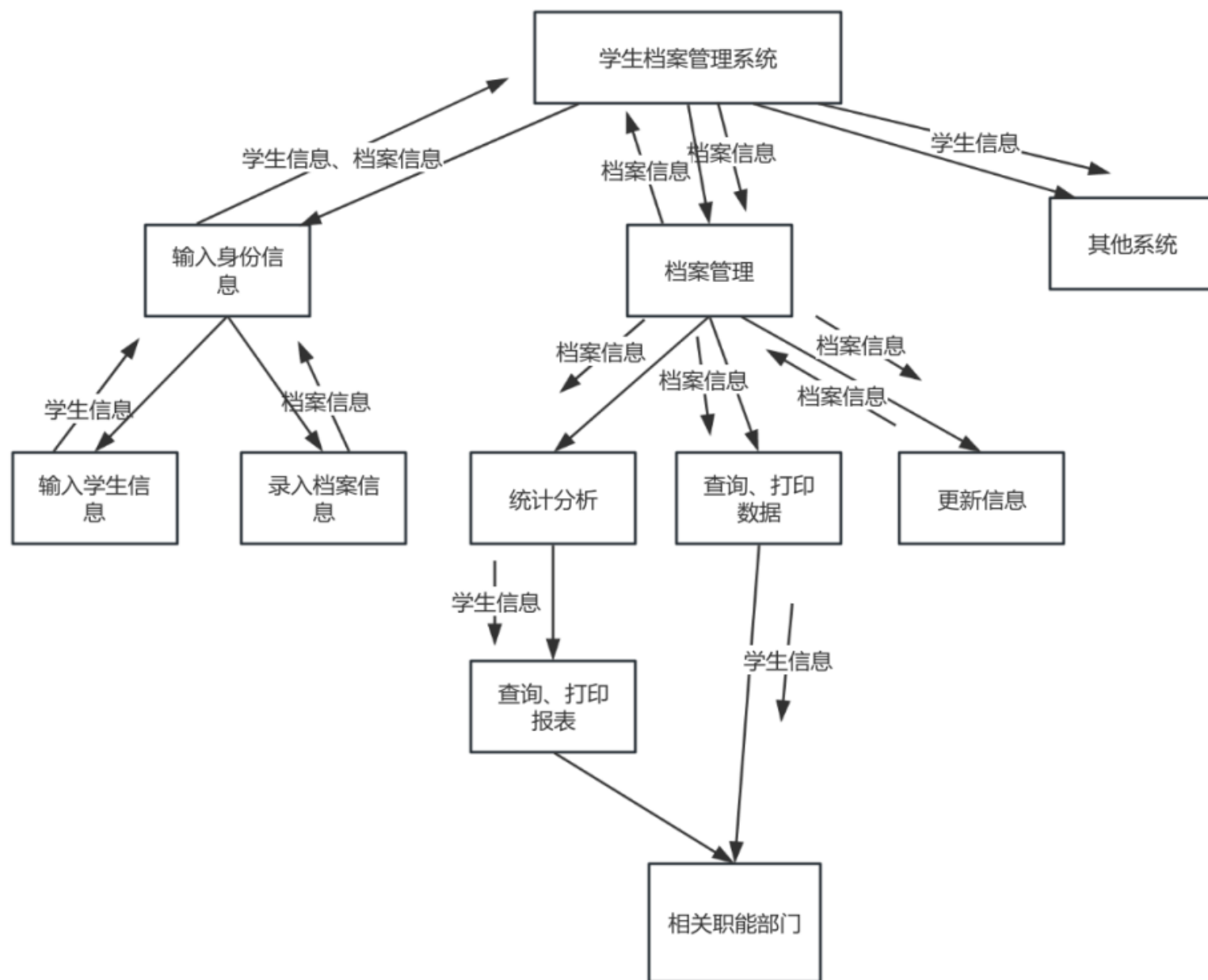
课堂练习2 答案



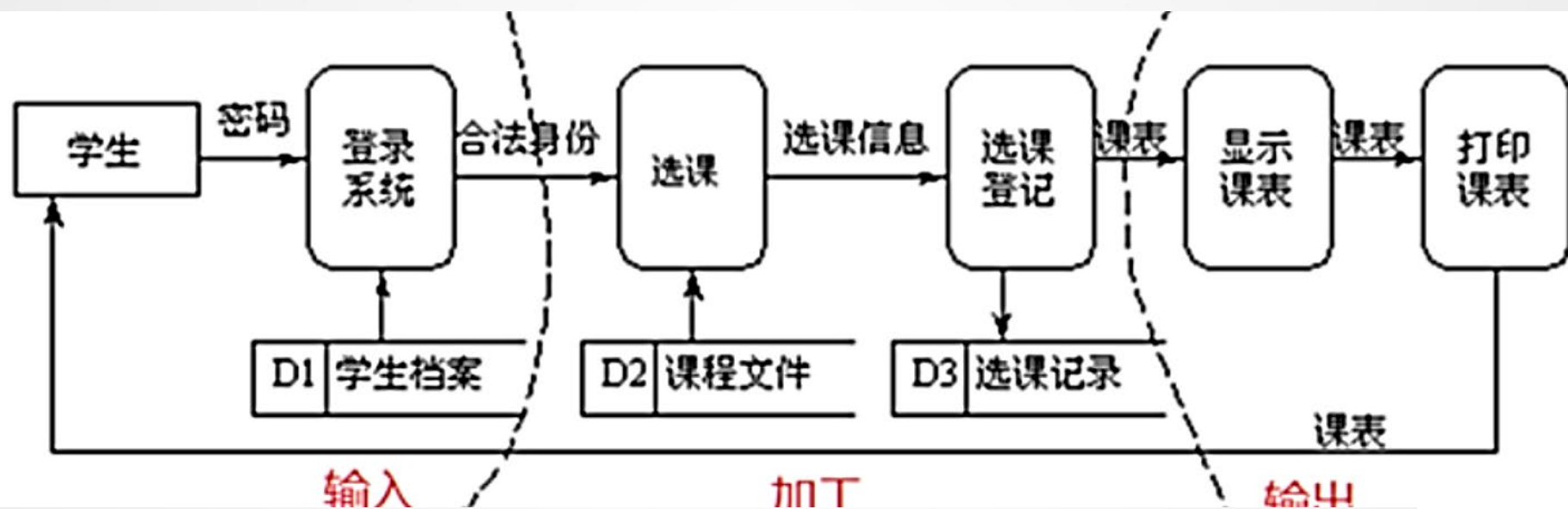
详细示例1：学生档案管理系统



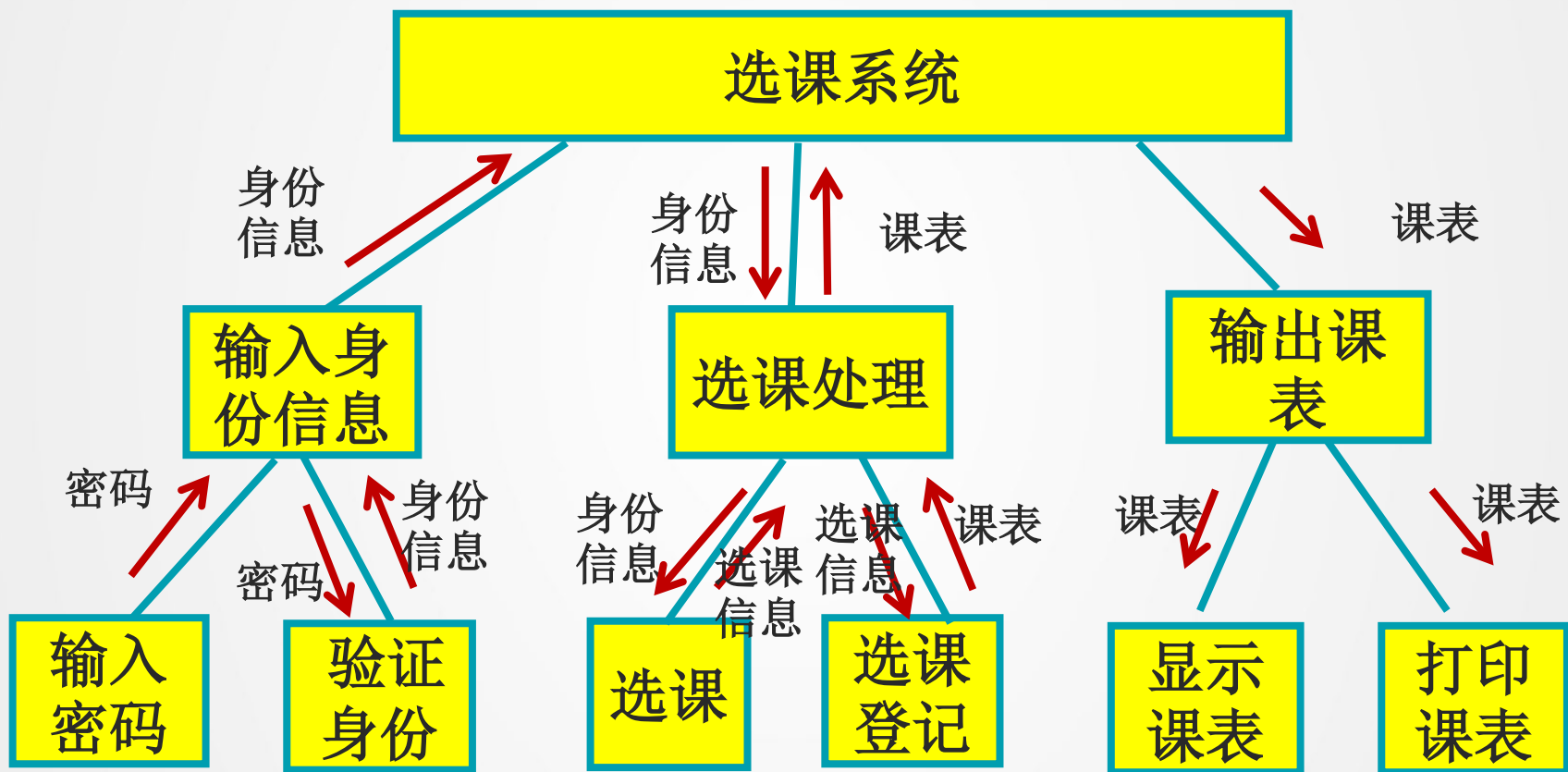
面向数据流的设计举例



详细示例2：学生选课



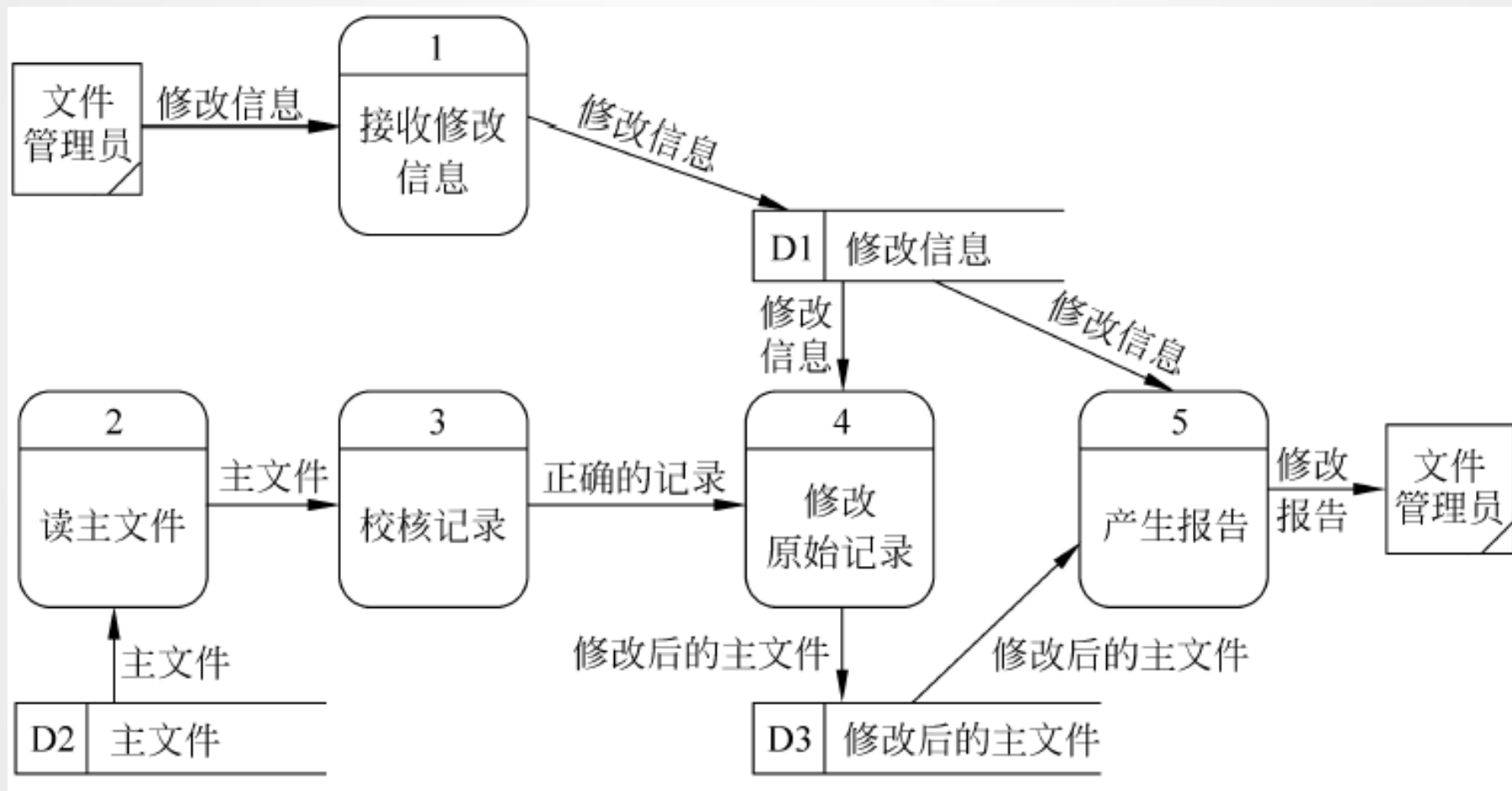
详细示例2：学生选课



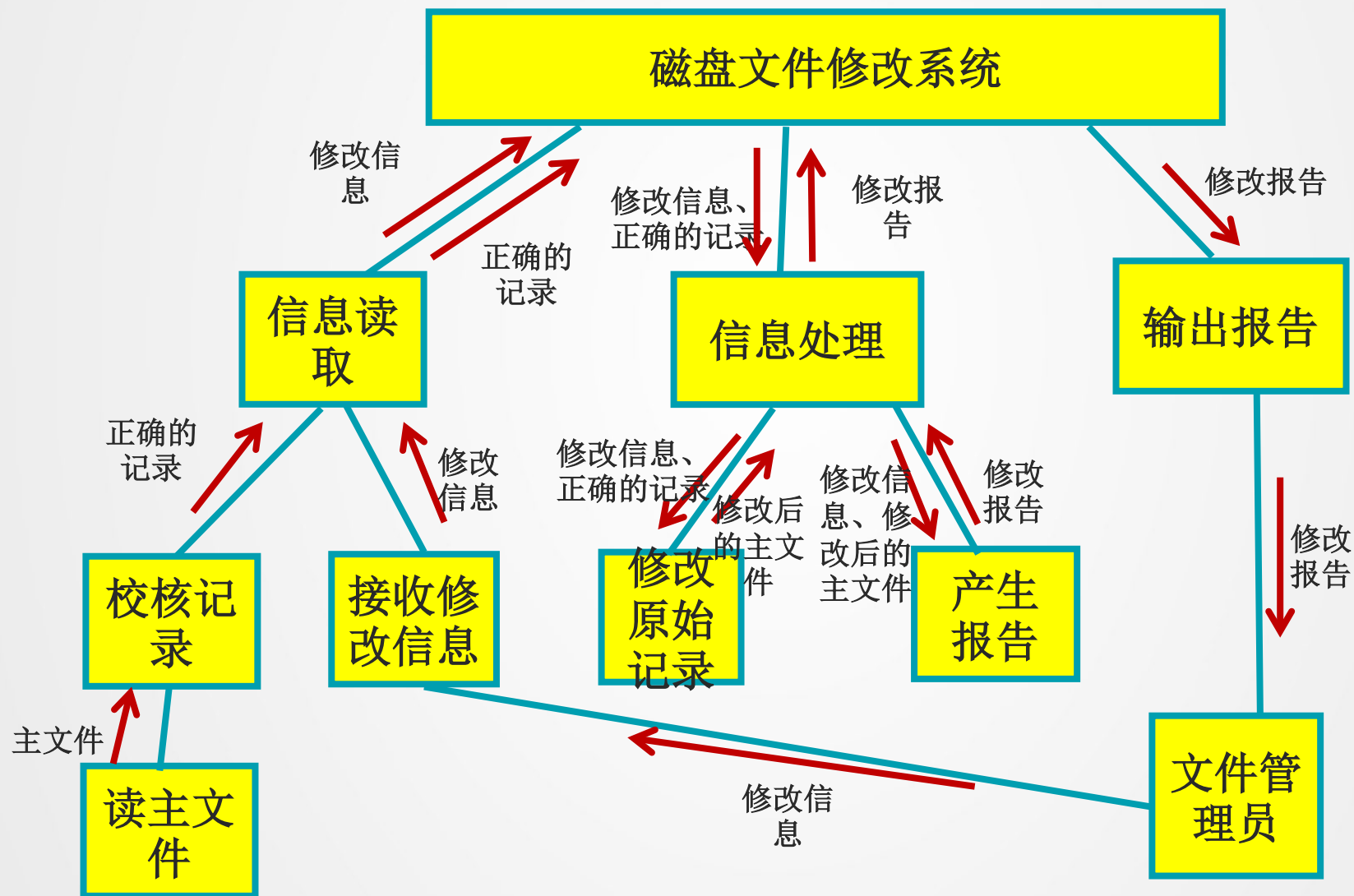
作业：

有一个修改磁带上主文件的系统，文件管理员把修改信息穿孔在卡片上，系统读入穿孔卡片上的信息并按照记录号把修改信息顺序排列好。然后系统逐个读入主文件上的记录，根据记录上的校验码校核每个读入的记录，丢掉出错的记录，按照修改信息修改余下的记录，产生的新文件存储在磁盘上并打印出一份修改报告供文件管理员参阅，**请画出系统的数据流图，并将数据流图用SD方法映射为软件结构图。**

作业答案：数据流图



作业答案：软件结构图



本章結束