

## 变换与事务分析

授课教师: 蓝天 电子邮箱: lantian1029@uestc.edu.cn

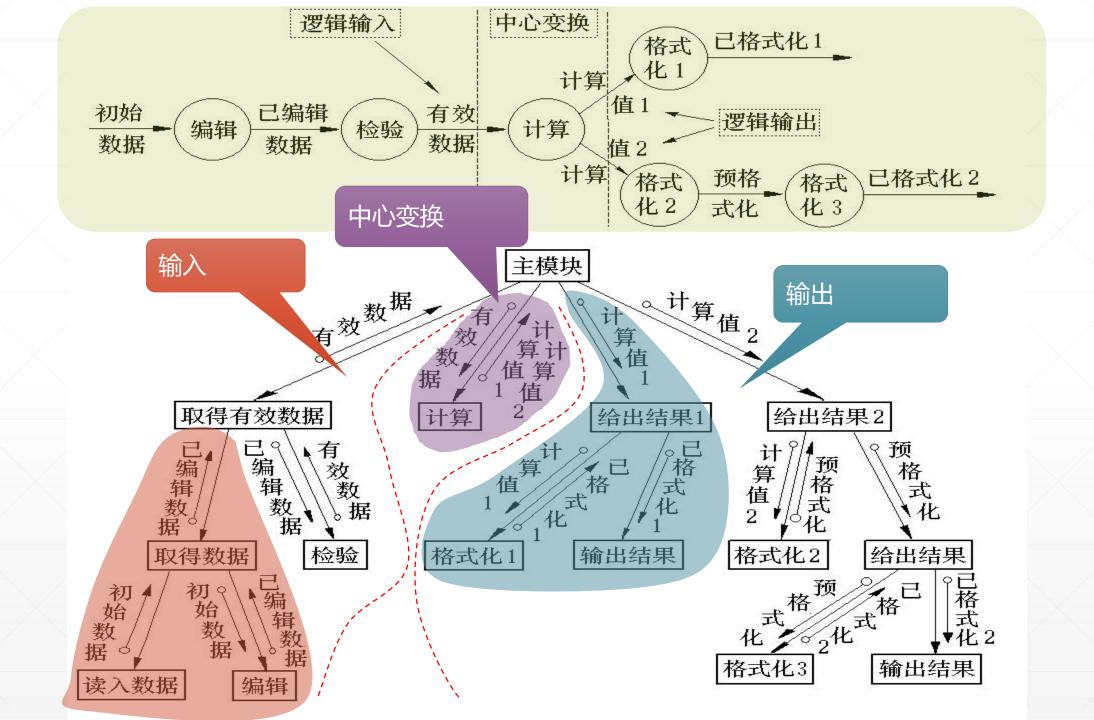
## 变换分析

## 重画数据流图

区分有效(逻辑)输入、有效(逻辑)输出和中心变换部分

进行一级分解,设计上层模块

进行二级分解,设计输入、输出和中心变换部分的中、下层模块



## 变换分析注意事项

## 深度遍历模块设计

• 在选择模块设计的次序时,必须对一个模块的全部直接下属模块都设计完成之后,才能转向另一个模块的下层模块的设计。

## 耦合与内聚

• 在设计下层模块时,应考虑模块的耦合和内聚问题,以提高初始结构图的质量。

## 使用"黑箱"技术

• 在设计当前模块时,先把这个模块的所有下层模块定义成"黑箱",在设计中利用它们时,暂时不考虑其内部结构和实现。在这一步定义好的"黑箱",在下一步就可以对它们进行设计和加工。这样,又会导致更多的"黑箱"。最后,全部"黑箱"的内容和结构应完全被确定。

## 控制直接下属模块数

• 在模块划分时,一个模块的直接下属模块一般在5个左右。如果直接下属模块超过10个,可设立中间层次。

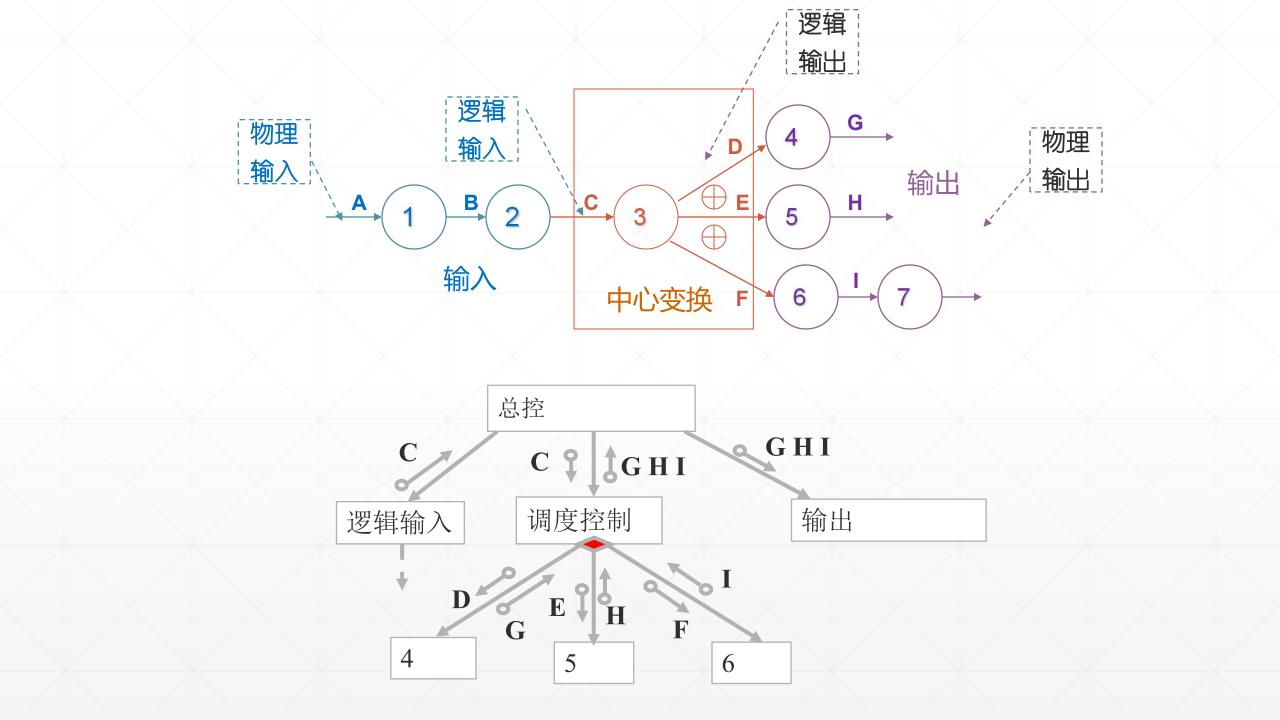
## 停止模块功能分解的情况

- 模块不能再细分为明显的子任务;
- 分解成用户提供的模块或程序库的子程序;
- 模块的界面是输入 / 输出设备传送的信息;
- 模块不宜再分解得过小。

## 事务分析

在很多软件应用中,存在某种作业数据流,它可以引发一个或多个处理,这些处理能够完成该作业要求的功能。这种数据流就叫做事务。

与变换分析一样,事务分析也是从分析数据流图开始,自顶向下, 逐步分解,建立系统结构图。



## 事务分析过程

#### 识别事务源

利用数据流图和数据词典,从问题定义和需求分析的结果中,找出各种需要处理的事务。通常,事务来自物理输入装置。有时,设计人员还必须区别系统的输入、中心加工和输出中产生的事务。

### 规定适当的事务型结构

 在确定了该数据流图具有事务型特征之后, 根据模块划分理论,建立适当的事务型结构。

### 识别各种事务和它们定义的操作

 从问题定义和需求分析中找出的事务及其 操作所必需的全部信息,对于系统内部产 生的事务,必须仔细地定义它们的操作。

#### 注意利用公用模块

在事务分析的过程中,如果不同事务的一些中间模块可由具有类似的语法和语义的若干个低层模块组成,则可以把这些低层模块构造成公用模块。

## 事务处理模块

• 对每一事务,或对联系密切的一组事务, 建立一个事务处理模块;如果发现在系统中有类似的事务,可以把它们组成一个事 务处理模块。

#### 操作模块

• 对事务处理模块规定它们全部 的下层操作模块

#### 细节模块

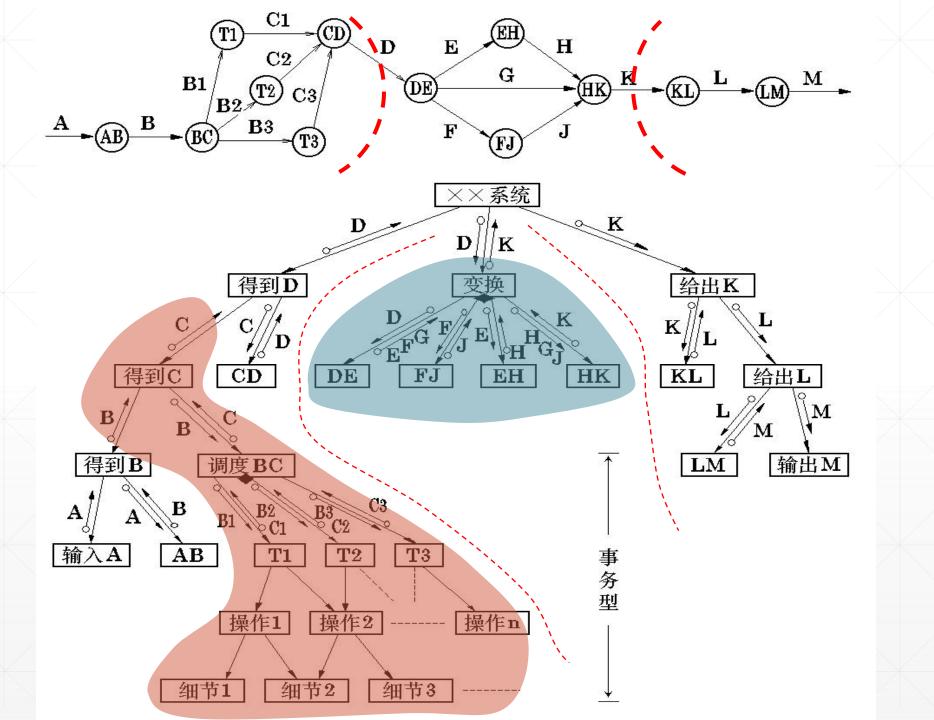
• 对操作模块规定它们的全部细 节模块

## 混合结构分析

变换分析是软件系统结构设计的主要方法。

一般,一个大型的软件系统是变换型结构和事务型结构的混合结构。

所以,我们通常利用以变换分析为主、事务分析为辅的方式进行软件结构设计。





# 感谢观看!

授课教师: 蓝天 电子邮箱: lantian1029@uestc.edu.cn