

堆工部软件室	文档编号		版本	A1	密级	商密 A
	项目名称	MOSCA 堆芯子通道热工水力软件				
	项目来源					

MOSCA 堆芯子通道热工水力软件  
详细设计说明书

编	写：		日 期：	
检	查：		日 期：	
审	核：		日 期：	
批	准：		日 期：	

中国原子能科学研究院



## 目 录

1. 引言.....	5
1.1 背景.....	5
1.2 编写目的和范围.....	5
1.3 术语表.....	5
1.4 参考资料.....	错误!未定义书签。
1.5 开发环境和工具.....	6
2. 设计概述.....	7
2.1 任务和目标.....	7
2.1.1 需求概述.....	7
2.1.2 运行环境概述.....	7
2.1.3 条件与限制.....	7
2.1.4 详细设计方法和工具.....	7
3. 系统详细需求分析.....	7
3.1 详细需求分析.....	7
3.1.1 详细功能需求分析.....	7
3.1.2 详细性能需求分析.....	7
3.1.3 详细资源需求分析.....	7
3.1.4 详细接口需求分析.....	7
3.1.5 详细系统运行环境及限制条件分析.....	7
4. 总体方案确认.....	7
4.1 系统总体结构确认.....	8
4.1.1 系统组成、逻辑结构及层次确认.....	8
4.1.2 应用系统结构确认.....	8
4.1.3 支撑系统结构确认.....	8
4.1.4 系统集成确认.....	8
4.1.5 系统工作流程确认.....	8
4.2 系统详细界面划分.....	8
4.2.1 应用系统与支撑系统的详细界面划分.....	8
4.2.2 系统内部详细界面划分.....	8
5. 全局数据结构说明.....	8
5.1 常量.....	8
5.2 变量.....	9
5.3 数据结构.....	9
6. 系统详细设计.....	9
6.1 功能结构图.....	9
6.2 系统结构设计及子系统划分.....	9
6.3 系统功能模块详细设计.....	9
6.4 系统界面详细设计.....	10
6.4.1 外部界面设计.....	10
6.4.2 内部界面设计.....	10
6.4.3 用户界面设计.....	10
7. 开发和测试生产环境说明.....	10
7.1 开发环境.....	10

7.2 测试及产品环境.....	10
8. 模块设计.....	10
8.1 用例图.....	10
8.2 功能设计说明.....	11
8.2.1 模块 1.....	11
8.2.2 模块 2.....	12
9. 接口设计.....	13
9.1 内部接口.....	13
9.2 外部接口.....	13
9.2.1 接口说明.....	13
9.2.2 调用方式.....	13
10. 数据库设计.....	13
11. 软件安全保密.....	13
11.1 说明.....	13
12. 系统性能设计.....	13
13. 出错处理.....	14
14. 设计和开发规范.....	14
14.1 数据库设计规范.....	错误!未定义书签。
14.2 编码规范.....	14
14.3 代码目录结构.....	16

# 1. 引言

## 1.1 背景

核反应堆堆芯热工水力分析软件一直是核反应堆设计中所使用的核心软件,堆芯热工水力设计国际公认的设计方法为子通道计算方法.该方法通过将堆芯划分子通道对堆芯燃料棒的传热状态进行相对精确的分析,常见的子通道软件为 COBRA,FLICA 等.

MOSCA(Model Objectified Sub Channel Analysis)软件下称本软件,依托于燕龙低温堆项目走出去配套软件知识产权自主化需求,由中国原子能科学研究院软件室牵头,联合西安交通大学核工程小组联合开发开发,其中绝大部分工作由软件室完成.

本软件使用现在语言以模块化开发方式开发,并完全自主化,极大提高了软件的可维护性和耦合性,大大提高了输入的可识别性和输出后处理.

## 1.2 编写目的和范围

本详细设计说明书编写的目的是说明程序模块的设计考虑,包括程序描述、输入/输出、算法和流程逻辑等,为软件编程和系统维护提供基础.本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

## 1.3 术语表

术语或缩略语	说明性定义
子通道	一种堆芯热工水力计算方法
动态链接库	DLL 文件, 通常作为软件的组件使用, 为封装好的带接口的可执行程序
W3C	万维网联盟组织
Xml	万维网联盟提出的计算机中通用的包含标签的数据文件
序列化	将内存中的数据对象转换为 xml 文本
反序列化	将 xml 文本转换为内存中数据对象
BS 架构	主要用浏览器调用以 http 协议服务器的架构
Web Service	BS 架构中服务器提供的可远程调用的网络服务
面向对象	具象化的编程思想
.Net FrameWork	微软发布的软件开发框架
Asp. net	微软发布的网站设计系统
IIS6	网络服务器代码托管程序
C#	C Sharp, 微软发布的面向对象语言
属性	面向对象中封装的字段
方法	面向对象中封装的函数
继承	面向对象中一个对象获取另一个对象的属性和方法的一

	种方式
接口	用于继承的定义某种功能的类
签名	用于版权保护的数字签名
单元测试	
静态测试	静态测试是指被测试程序不在机器上运行, 而是采用人工检测和计算机辅助静态分析的
模块测试	
系统测试	
热更新	代码维护更新时只需更改服务器代码即可用户使用功能或逻辑的更改
UML	类关系设计图 Unified Modeling Language
SignalR	微软集成了一个 web socket 的通讯插件
用例图	从系统的外部观看系统功能, 并不描述系统内部
可维护性	静态测试是指被测试程序不在机器上运行, 而是采用人工检测和计算机辅助静态分析的

## 1.4 开发环境和工具

代码编译器: Visual Studio 2012 及以上

开发环境: .Net Framework 4.5

开发语言:以 C#为主,包含部分 C++混合编程.

UML 绘图工具:Visual Studio 类设计器

## 2. 设计概述

### 2.1 任务和目标

#### 2.1.1 需求概述

#### 2.1.2 运行环境概述

#### 2.1.3 条件与限制

#### 2.1.4 详细设计方法和工具

## 3. 系统详细需求分析

*主要对系统级的需求进行分析。首先应对需求分析提出的企业需求进一步确认，并对由于情况变化而带来的需求变化进行较为详细的分析。*

### 3.1 详细需求分析

#### 3.1.1 详细功能需求分析

#### 3.1.2 详细性能需求分析

#### 3.1.3 详细资源需求分析

#### 3.1.4 详细接口需求分析

#### 3.1.5 详细系统运行环境及限制条件分析

## 4. 总体方案确认

*着重解决系统总体结构确认及界面划分问题。*

## 4.1 系统总体结构确认

*对系统组成、逻辑结构及层次进行确认，对应用系统、支撑系统及各自实现的功能进行确认，细化集成设计及系统工作流程，特别要注意因软件的引进造成的系统本身结构和公司其他系统的结构变化。包括：*

### 4.1.1 系统组成、逻辑结构及层次确认

### 4.1.2 应用系统结构确认

### 4.1.3 支撑系统结构确认

### 4.1.4 系统集成确认

### 4.1.5 系统工作流程确认

## 4.2 系统详细界面划分

### 4.2.1 应用系统与支撑系统的详细界面划分

*应用系统与支撑系统之间的界面包括系统主服务器与其他服务器的服务范围及访问方式，网络及数据库对应用系统的支撑方式，全局数据的管理与存取方式等。*

### 4.2.2 系统内部详细界面划分

*系统各功能之间的界面包括覆盖范围，模块间功能调用涉及到的系统模块及方法，全局数据格式，系统性能要求等。*

## 5. 全局数据结构说明

*本章说明本程序系统中使用的全局数据常量、变量和数据结构。*

### 5.1 常量

*包括数据文件名称及其所在目录，功能说明，具体常量说明等。*



## 5.2 变量

本章说明本程序系统中使用的全局数据常量、变量和数据结构。

## 5.3 数据结构

包括数据结构名称，功能说明，具体数据结构说明（定义、注释、取值）等。

# 6. 系统详细设计

## 6.1 功能结构图

## 6.2 系统结构设计及子系统划分

对系统的组成及逻辑结构进行设计前确认。

划分系统功能模块或子系统（如果有或者有必要，特别是大型的软件系统）。

## 6.3 系统功能模块详细设计

按结构化设计方法，在系统功能逐层分解的基础上，对系统各功能模块或子系统进行设计。此为详细设计的主要部分之一。

用层次图描述系统的总体结构、功能分解及各个模块之间的相互调用关系和信息交互，用 IPO 图或其他方法描述各模块完成的功能。以上建议采用 HIPO 图进行功能分解与模块描述，更高的要求建议采用 IDEF0 方法进行功能模型设计。

详细设计应用系统的各个构成模块完成的功能及其相互之间的关系，用 IPO 或结构图描述各模块的组成结构、算法、模块间的接口关系，以及需求、功能和模块三者之间的交叉参照关系。

每个模块的描述说明可参照以下格式：

模块编号：

模块名称：

输入：

处理：

算法描述：

输出：

其中处理和算法描述部分主要采用伪码或具体的程序语言完成。

对详细设计更高的要求建议用 IDEF0 图进行各功能模块的设计。

如果对软件需进行二次开发（包括功能扩展、功能改造、用户界面改造等），则相应的设计工作应该设立子课题完成。

## 6.4 系统界面详细设计

*系统界面说明应用系统软件的各种接口。整个系统的其他接口（如系统硬件接口、通讯接口等）在相应的部分说明。*

### 6.4.1 外部界面设计

*根据系统界面划分进行系统外部界面设计，对系统的所有外部接口（包括功能和数据接口）进行设计。*

### 6.4.2 内部界面设计

*设计系统内部各功能模块间的调用关系和数据接口。*

### 6.4.3 用户界面设计

*规定人机界面的内容、界面风格、调用方式等，包括所谓的表单设计、报表设计和用户需要的打印输出等设计。此部分内容可能比较多。*

## 7. 开发和测试生产环境说明

### 7.1 开发环境

### 7.2 测试及产品环境

- 开发环境:c#

Vs 2012 - - IIS 6.0 + - Access 2007 - .net framwork 4.5

- 测试环境: Win7 或更高版本 Windows, Windwos Server 2012, IIS 8.0

## 8. 模块设计

### 8.1 用例图

## 8.2 功能设计说明

### 8.2.1 模块 1

*模块 1 主要分为以下几个子模块：子模块 1、子模块 2 和子模块 N。*

#### 8.2.1.1 子模块 1

##### 8.2.1.1.1 设计图

##### 8.2.1.1.2 功能描述

*简要描述子模块 1 的业务功能。*

##### 8.2.1.1.3 输入数据

*详细描述用户输入的数据(包括任何输入设备)以及这些数据的有效性检验规则。*

*详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。*

##### 8.2.1.1.4 输出数据

*详细描述子功能 1 所产生的数据以及这些数据的表现形式。*

##### 8.2.1.1.5 业务算法和流程

*从业务角度详细描述根据输入数据产生输出数据的业务算法和流程。*

##### 8.2.1.1.6 数据设计

*给出本程序中的局部数据结构说明，包括数据结构名称，功能说明，具体数据结构说明（定义、注释设计、取值）等。相关数据库表，数据存储设计（具体说明需要以文件方式保存的数据文件名、数据存储格式、数据项及属性等。）*

##### 8.2.1.1.7 源程序文件说明

*给出本程序的各源程序文件的说明，包括源程序文件名称及其所在目录，功能说明，包含的前导文件及函数名称等。*

##### 8.2.1.1.8 函数说明

*具体说明本程序中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。*

#### 8.2.1.1.9 限制条件

#### 8.2.1.1.10 其他说明

### 8.2.2 模块 2

*模块 1 主要分为以下几个子模块：子模块 1、子模块 2 和子模块 N。*

#### 8.2.2.1 子模块 1

##### 8.2.2.1.1 设计图

##### 8.2.2.1.2 功能描述

*简要描述子模块 1 的业务功能。*

##### 8.2.2.1.3 输入数据

*详细描述用户输入的数据(包括任何输入设备)以及这些数据的有效性检验规则。*

*详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。*

##### 8.2.2.1.4 输出数据

*详细描述子功能 1 所产生的数据以及这些数据的表现形式。*

##### 8.2.2.1.5 业务算法和流程

*从业务角度详细描述根据输入数据产生输出数据的业务算法和流程。*

##### 8.2.2.1.6 数据设计

*给出本程序中的局部数据结构说明，包括数据结构名称，功能说明，具体数据结构说明（定义、注释设计、取值）等。相关数据库表，数据存储设计（具体说明需要以文件方式保存的数据文件名、数据存储格式、数据项及属性等。）*

##### 8.2.2.1.7 源程序文件说明

*给出本程序的各源程序文件的说明，包括源程序文件名称及其所在目录，功能说明，包含的前导文件及函数名称等。*

##### 8.2.2.1.8 函数说明

*具体说明本程序中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。*

##### 8.2.2.1.9 限制条件

##### 8.2.2.1.10 其他说明

## 9. 接口设计

### 9.1 内部接口

### 9.2 外部接口

#### 9.2.1 接口说明

例如：XX 子系统通过 XX 从 XX 子系统取得 XX 等，相关标准，调用示例，可根据需要增加章节描述接口。

#### 9.2.2 调用方式

例如：内部接口调用：

例：

/\*\*

\*通过用户服务号码取得该客户认证密码等信息，如果该客户存在返回为 0，其他情况参考错误编码

\*/

```
public RUserInfo getUserInfo (String userNo);
```

## 10. 数据库设计

详见[xxx 数据库设计说明书]

如果数据库设计内容比较少，则直接在此处描述。

## 11. 软件安全保密

### 11.1 说明

对于运行在服务器上的组建具有天然的安全性优势,本身具有较高的安全性.软件中所使用的算法为如何有效的保护可执行程序 and dll 文件,对于运行在单机上面的软件进行强签名和混淆。设计

## 12. 系统性能设计

子通道分析计算方法是一个主要使用工程经验公式计算的算法,与计算流体力学 CFD 计算方法有本质上的区别,对硬件性能要求较低.MOSCA 程序服务器上运行的代码是天然并行的,可设置并发数目和可用核心数进行并行计算.对于单机版本,推荐也选择使用调用服务器远程计算的方式并行.

# 13. 出错处理

例如：为了在用户输入有误或系统出现异常情况下给用户以明确的提示，程序通过输出消息的方式予以提示.消息标注以下类型：

- 1. Info:通知信息用以知晓
- 2. Warning:警告可能的错误
- 3. Fatal:致命的错误,程序必须中止
- 2. error:局部错误,程序可以继续运行

这些通知消息会在用户计算时显示在用户界面上.

# 14. 设计和开发规范

## 14.1 编码规范

- 1. 程序中具有物理含义的变量须尽量贴近物理公式中变量的表示方法,本软件物理参数\*必须\*优先使用下表中的表示方法.


- 2. 类和命名空间使用名词词组,并大驼峰方式命名,首字母必须大写,不要使用下划线字符（\_）
- 3. 方法名称首字母大写，必须写明其执行的功能注释.推荐使用<summary>注释
- 4. 功能极度近似的代码不能连续重复超过两次,考虑使用函数方法的方式重复执行.
- 5. 被继承的基类以 Base 结尾,接口以大写字母 I 开头.
- 6. If 和 While 等判断语句须使用 {},除非只有一条语句并和 if 语句同一行
- 7. 代码中推荐的变量命名规则排列顺序为:大驼峰,小驼峰,下划线.

例如推荐使用顺序 ParaName>paraName>para\_name. 一般的私有成员变量推荐使用小驼峰或者下划线开头,全局变量采用推荐采用大驼峰和下划线命名法,局部变量推荐使用小驼峰和下划线命名法.下划线推荐使用的方式：对于一个较长的方法或者变量名称，可以使用下划线加首字母大写方式命名。例如方法：CaculateChannelsSteady 推荐些为 Caculate\_Channels\_Steady，达到更好的可读性。

- 8. 使用 Stream 流时须用 using 语句调用，另外对于存在导致程序溢出的代码,必须使用 try...Catch 包裹

9. 一些常识性代码，不需要写注释，注释单起一行位于被注释代码上方。尽量避免代码后方注释。

```
#region 计算主程序
/// <summary>将输入数据和程序中变量做对应，需要存在输入文件后再调用，全局变量赋值，并简化表示
public void Recognize()
{
    ///////////////////////////////////////////////////
    //主要对输入数据的表示方法进行简化处理，被赋值的变量均为[全局变量]
    //同时可对输入数据进行[简单的]计算
    //
    InputModel InputData = MyIOManager.InputData;
    //子通道对象
    Channels = InputData.ChannelCollection.Channels;
    //燃料棒对象
    Rods = InputData.RodCollection.Rods;
    //燃料棒类型集合
    RodTypes = InputData.RodTypes;
    //子通道数
    Ni = Channels.Count;
    //轴向分段数
    Nj = InputData.RodCollection.Segment;
    //燃料棒个数
    Nk = Rods.Count;
    //功率因子
    PowerFactor = InputData.Options.PowerFactor.Multiplier;
    //燃料芯块功率份额
    PelletShare = InputData.Options.PowerFactor.PelletShare;
    //燃料包壳功率份额
    CladShare = InputData.Options.PowerFactor.CladShare;
    //流体中慢化功率份额
    FluidShare = InputData.Options.PowerFactor.FluidShare;
    //临界热流密度CHF计算公式选用
    chf_indicator = InputData.Options.DNBR_Formula;
    //冷却剂模型
    Coolent = InputData.MaterialCollection.Fluid;
    //气体间隙
    GasGap = InputData.MaterialCollection.GasGap;
    //固体材料集合
    Materials = InputData.MaterialCollection.Materials;
    //初始的流量数据模型
    MassFlow = InputData.MassFlow;
```

10. 只要合适，在变量名的末尾追加计算限定符（Avg、Sum、Min、Max、Index）
11. 每个新的代码文件头部应该包含代码的主要信息,如

```

//Main.cs
//主计算流程
//创建:2017-8-15 宋仕钊
//修改:2018-4-13 宋仕钊
using System;
using System.Collections.Generic;
using MoscaCore.Helper;
using MoscaCore.Models;
using MathNet.Numerics.LinearAlgebra;
using System.Diagnostics;

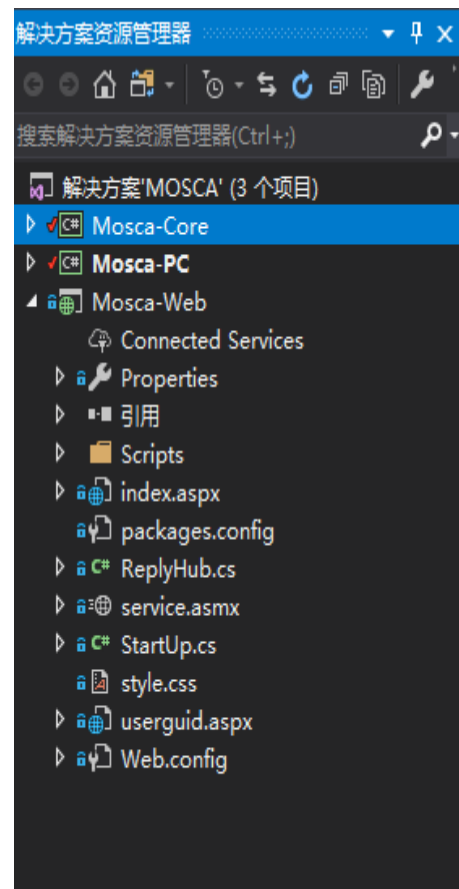
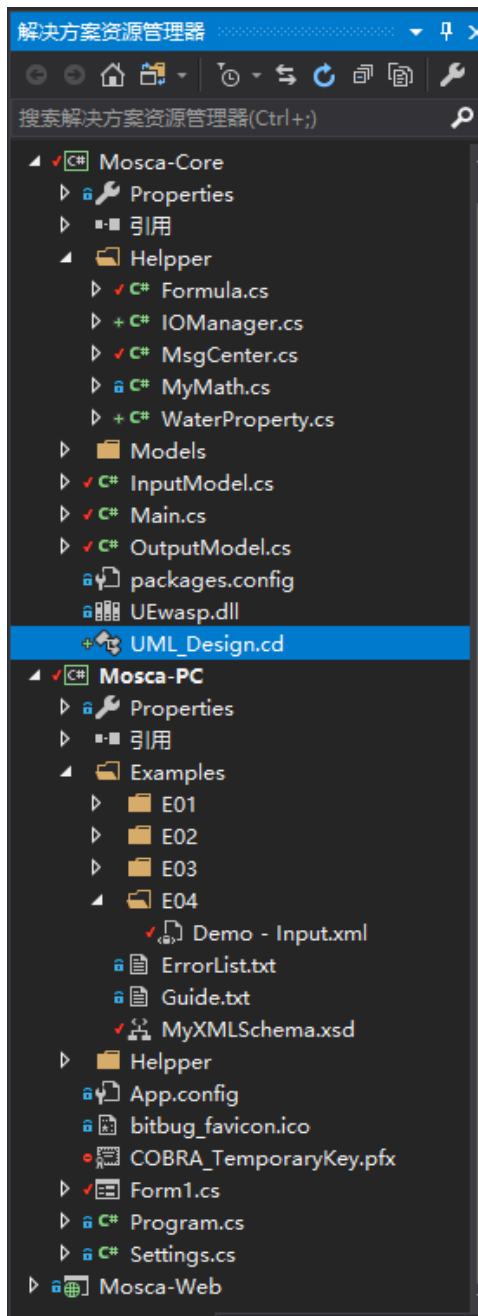
namespace MoscaCore

```

## 14.2 代码目录结构

系统架构目录结构设计图,如下:





结构说明：

作用范围	包名	描述
	Images	存放图片
	Scripts	Jquery 以及样式表等
	Scripts/jquery-easyui-1.2.6	jquery-easyui 包数字为版本

整个程序 Content	Scripts/jquery-ui-1.8.20	jquery-ui 包数字为版本
	Scripts/jthok-ui	自定义的 js 格式文件
	Scripts/themes	Icon 格式图片以及样式
Controllers	../Controller	控制器处理来自浏览器的输入，并返回相应页面给用户。
Data	../Data	放自定义的 json 文件
Models	../Models	存放模型
Scripts	../Scripts	引入的各种外部 js 文件
Views	../Views	存放各种类型的试图
Global.asax	../Global.asax	注意定义路由规则等
Web.config	../Web.config	配置文件

