|  |  |  |  |
| --- | --- | --- | --- |
| 文档类型 | 功能设计文档 | 机密等级 | 内部公开 |
| 部门 | 西安分公司 | 版本 | V3.1. |

# 小区管理方案设计

|  |  |  |  |
| --- | --- | --- | --- |
| 作者 |  | 日期 | 2025/4/2 |
| 评审人 |  | 日期 |  |
| 批准人 |  | 日期 |  |

## 修订记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 修改描述 | 修改原因 | 修改人 | 日期 | 备注 |
| V3.1 |  |  |  |  |  |
|  |  |  |  |  |  |

## 术语及缩略语解释

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 中文名称 | 英文名称 | 解释说明 | 备注 |
|  |  |  |  |  |

目录

[小区管理方案设计 1](#_Toc195476722)

[修订记录 2](#_Toc195476723)

[术语及缩略语解释 2](#_Toc195476724)

[1. 概述 4](#_Toc195476725)

[2. 模块设计 4](#_Toc195476726)

[2.1 模块介绍 5](#_Toc195476727)

[2.2 数据流图 6](#_Toc195476728)

[3. 操作符管理 6](#_Toc195476729)

[3.1 小区运行状态 6](#_Toc195476730)

[3.2 操作符接纳 7](#_Toc195476731)

[3.3 轮询机制 7](#_Toc195476732)

[3.4 接口设计 9](#_Toc195476733)

[4. 告警管理 9](#_Toc195476734)

[4.1 小区类告警 9](#_Toc195476735)

[4.2 告警策略 9](#_Toc195476736)

[5. 状态接纳 (选做) 11](#_Toc195476737)

[6. RRU小区管理 13](#_Toc195476738)

[7. PHY小区管理 14](#_Toc195476739)

[8. 业务流程设计 17](#_Toc195476740)

[8.1 状态机介绍 17](#_Toc195476741)

[8.2 小区建立 22](#_Toc195476742)

[8.3 小区删除(选做) 27](#_Toc195476743)

[8.4 小区复位 28](#_Toc195476744)

[8.5 小区重配(选做) 35](#_Toc195476745)

[8.6 异常流程 35](#_Toc195476746)

[9. 信号流图 39](#_Toc195476747)

[10. 附录 42](#_Toc195476748)

## 概述

本文档为小区管理方案设计，详细介绍了小区管理所涉及的基本内容。该文档的目的在于给开发和测试人员提供小区管理方面系统性的理论指导，可帮助其对业务现状进行查漏补缺。该文档章节主要分为以下几部分：

1、模块设计：主要介绍参与小区管理流程各模块基本情况介绍。

2、操作符管理：主要针对复杂的小区交织操作场景给出应对的措施。

3、告警管理：本章节介绍了小区告警上报、恢复机制以及应遵循的基本策略。

4、状态接纳：介绍小区建立的基本先决条件。

5、RRU小区管理：介绍了RRU小区运作机制及基本接口设计。

6、PHY小区管理：围绕PHY状态机展开介绍PHY小区运作机制。

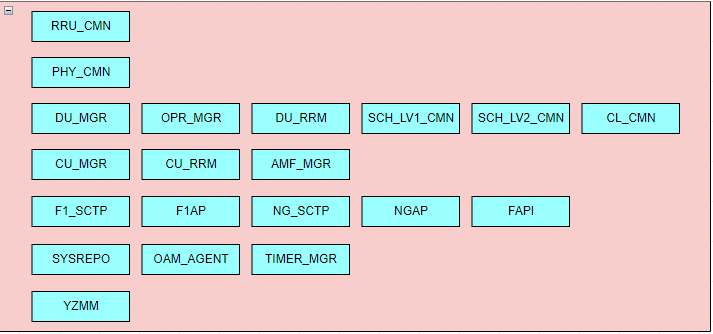
7、业务流程设计：详细介绍了业务小区管控基本流程。

8、信号流图：通过图例展示小区运作流程的几个基本典型场景。

## 模块设计

小区管理主要内容包括网管数据管控、平台数据管控、链路管控、CU小区管控、DU小区管控、PHY小区管控及射频小区管控等，参与的模块众多。

网管数据管控模块YZMM；平台数据管控模块包括SYSREPO、OAM\_AGENT、TIMER\_MGR等模块；链路管控模块包括F1\_SCTP、F1AP、NG\_SCTP、NGAP、FAPI等模块；CU小区管控模块主要包括CU\_MGR、CU\_RRM、AMF\_MGR等模块；DU小区管控模块主要包括DU\_MGR、OPR\_MGR、DU\_RRM、SCH\_LV1\_CMN、SCH\_LV2\_CMN、CL\_CMN等模块；PHY小区管控模块为PHY\_CMN模块；射频小区管控模块为RRU\_CMN模块。



**图2-1 小区管理关联模块**

### 2.1 模块介绍

1、网管数据管控模块

YZMM提供用户操作界面，主要负责小区配置参数修改/下发，以及小区状态的维护等。

2、平台数据管控模块

(1) SYSREPO数据存储库以YANG模型配置数据为基准，提供数据修改触发机制、数据访问机制及数据同步机制。

(2) OAM\_AGENT模块，主要负责数据存储/下发/上报；告警上报/恢复和KPI上报的通道。总之，该模块是业务和网管间业务数据交互的桥梁。

(3) TIMER\_MGR模块，主要负责业务流程消息握手保护、流程段保护、亦可作为部分功能触发激励(如链路断后在指定时间内释放基站所有UE、从小区操作符队列定时取操作符等)。

3、链路管控模块

(1) F1\_SCTP模块主要负责F1链路配置、通信机制实现。

(2) F1AP模块主要负责F1口消息定义、填写、编解码。

(3) NG\_SCTP模块主要负责NG链路配置、通信机制的实现。

(4) NGAP模块主要负责NG口消息定义、填写、编解码。

(5) FAPI链路主要承接CL\_CMN和PHY\_CMN模块间、DU\_MGR和RRU\_CMN模块间的数据交互工作。

4、CU小区管控模块

(1) CU\_MGR模块主要负责OAM\_AGENT模块转发数据的维护；DU实例、AMF实例、SCTP实例的创建；小区运行流程的控制。

(2) CU\_RRM模块主要负责小区配置参数维护，核心网配置信息存储。

(3) AMF\_MGR模块主要负责NG链路状态维护、NGAP建立/复位/删除流程控制、核心网配置信息存储。

5、DU小区管控模块

(1) DU\_MGR模块主要负责OAM\_AGENT模块转发数据的维护；SCTP实例的创建、F1AP建立状态维护；TRANS实例创建/删除；小区运行流程的控制。

(2) OPR\_MGR模块主要负责小区当前运行状态、操作记录维护、操作判决。

(3) DU\_RRM模块主要负责小区无限资源分配、管理。

(4) SCH\_LVL1\_CMN主要负责调度资源存储、分配以及下发。

(5) SCH\_LVL2\_CMN主要负责调度资源存储、分配以及下发。

(6) CL\_CMN主要负责物理层小区信息配置、运行流程控制，是物理层小区代理管控模块。

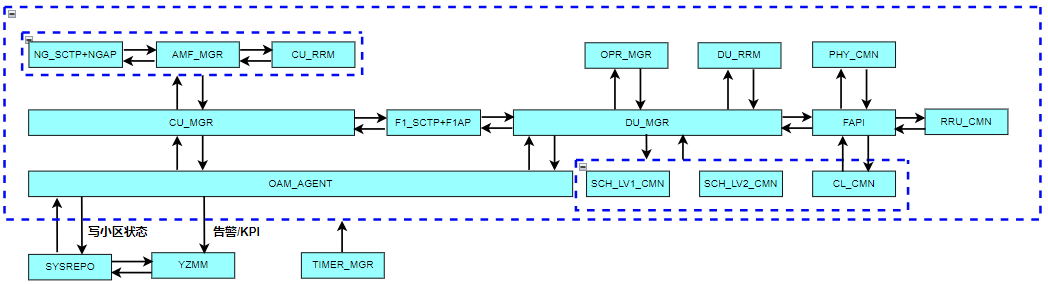
6、PHY小区管控模块

PHY\_CMN主要负责物理层运行状态维护、小区配置信息维护、数据转发以及小区运行流程的控制。

7、射频小区管控模块

RRU\_MGR主要负责射频配置数据存储、转发以及小区运行流程的控制。

### 2.2 数据流图



**图2-2 模块间数据流图**

## 操作符管理

小区操作符是对小区当前即将发生动作的描述，该动作可由用户操作触发、业务流程触发或者外部环境触发。操作符按照操作类型分为OPR\_ADD、OPR\_DEL、OPR\_RECFG，其触发场景如下所述。

OPR\_ADD触发场景：网管新增小区、网管小区解关断、部分复位参数修改新建小区、基站上电新建小区、小区建立异常(响应失败/超时)尝试重建、SCTP链路(F1/NG)异常尝试重建、时钟状态失锁尝试重建、基带卡/RRU运行状态异常尝试重建等。

OPR\_DEL触发场景：网管删除小区、网管小区关断、参数复位参数修改删除小区、小区建立异常(响应失败/超时)删除小区、SCTP链路(F1/NG)异常删除小区、时钟状态失锁删除小区、基带卡/ RRU运行状态异常删除小区等。

OPR\_RECFG触发场景：部分重配参数修改进行小区重配。

|  |  |
| --- | --- |
| 操作符 | 原因值 |
| OPR\_ADD | DBS\_ADD\_CELL、INIT\_ADD\_CELL、DEL\_ADD\_CELL |
| OPR\_DEL | DBS\_DEL\_CELL、PARA\_MOD\_DEL\_CELL、F1\_ASSOC\_DOWN\_DEL\_CELL、  NG\_ASSOC\_DOWN\_DEL\_CELL、CFG\_FAIL\_DEL\_CELL、CLOCK\_UNLOCK\_DEL\_CELL、  BORAD\_ABNORMAL\_DEL\_CELL、RRU\_ABNORMAL\_DEL\_CELL |
| OPR\_RECFG | PARA\_MOD\_RECFG\_CELL |

**表3-1 操作符原因值**

### 3.1 小区运行状态

小区运行状态是对小区已执行操作的过程或结果的描述，运行状态和已执行操作对应关系如表3-1所示。

|  |  |  |
| --- | --- | --- |
| 运行状态 | 含义 | 已执行操作 |
| CELL\_RUN\_IDLE | 运行初始状态 | 初始上电(结果)、OPR\_DEL(结果) |
| CELL\_WAITTING\_SETUP | 小区正在建立 | OPR\_ADD(过程) |
| CELL\_SETUP\_COMPLETE | 小区建立完成 | OPR\_ADD(结果)、OPR\_RECFG(结果) |
| CELL\_WAITTING\_DELETE | 小区正在删除 | OPR\_DEL(过程) |
| CELL\_WAITTING\_RECFG | 小区正在重配 | OPR\_RECFG(过程) |

**表3-2 小区运行状态和已执行操作关系表**

特别强调：OPR\_DEL操作符执行完成小区运行状态恢复至CELL\_RUN\_IDLE态；OPR\_RECFG操作符执行完成小区运行状态恢复至CELL\_SETUP\_COMPLETE。

### 3.2 操作符接纳

小区操作符是否执行取决于当前小区的运行状态，操作符接纳的基本规则如下表所示。其中处于CELL\_WAITTING\_DELETE运行状态的小区不接纳任何操作。

|  |  |  |
| --- | --- | --- |
| 运行状态 | 允许接纳操作符 | 备注 |
| CELL\_RUN\_IDLE | OPR\_ADD | 允许建立小区 |
| CELL\_WAITTING\_SETUP | OPR\_DEL | 允许删除小区 |
| CELL\_SETUP\_COMPLETE | OPR\_DEL、OPR\_RECFG | 允许删除/重配小区 |
| CELL\_WAITTING\_DELETE | 无 | 不允许任何操作 |
| CELL\_WAITTING\_RECFG | OPR\_DEL、OPR\_RECFG | 允许删除/重配小区 |

### 3.3 轮询机制

小区操作符队列用来存储基站所有小区的预操作符，业务侧周期性轮询从该队列中取出将要处理的操作符去执行，其过程示意图如图3-2所示。

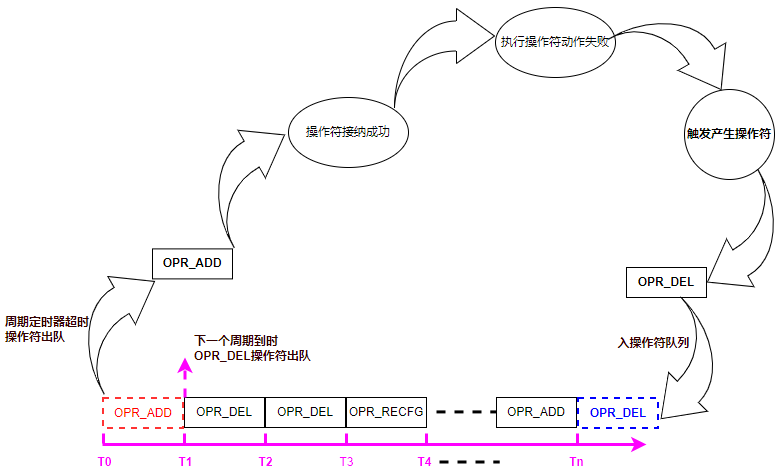
采用周期性轮询机制的前提是基于业务对小区级别的操作时延性要求并不高，轮询机的优点在于：

1 离散化操作符处理，能够确保处理顺序和操作符产生顺序的一致性。

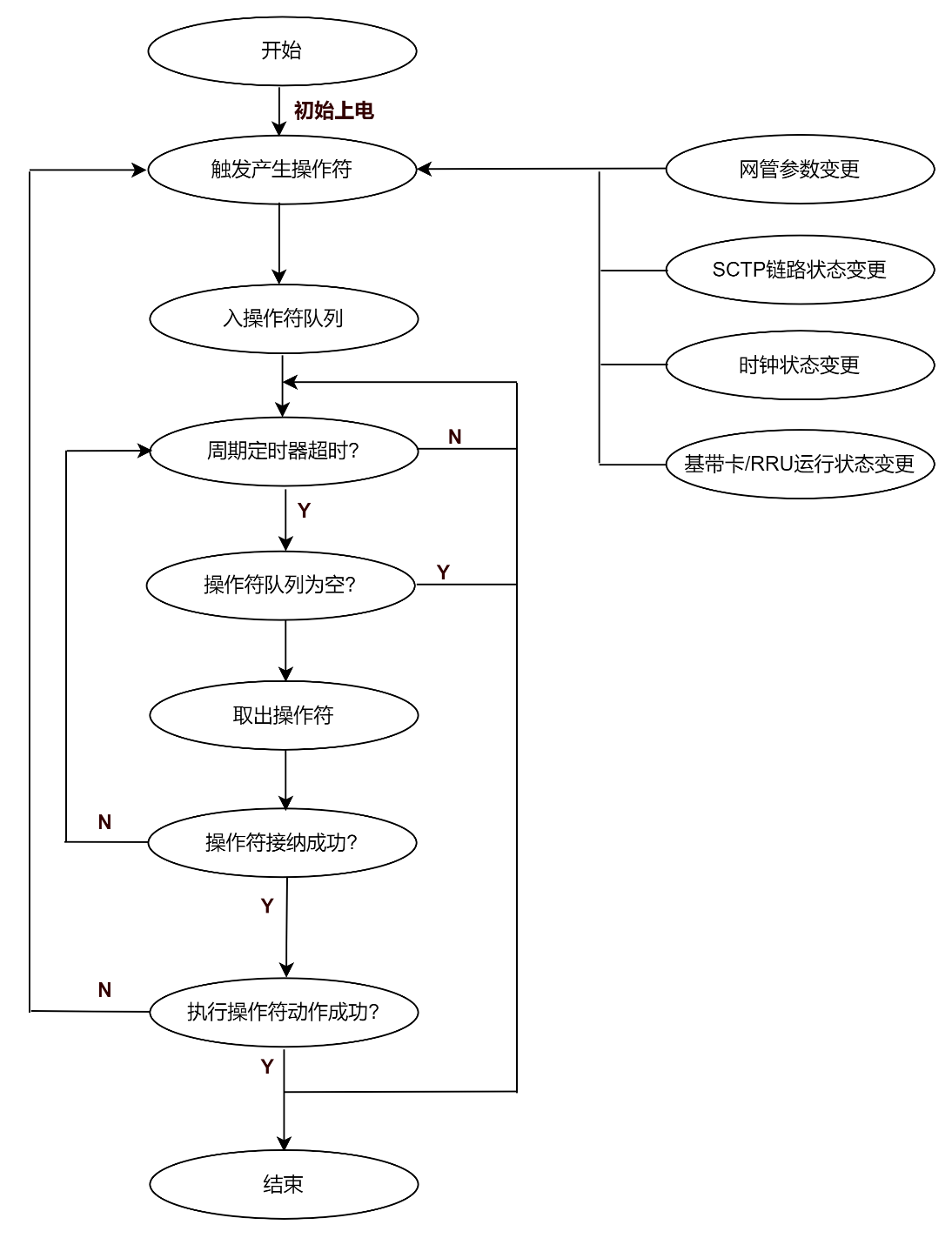
2离散化操作符的处理，大概率避免交织流程的发生。轮询周期一般在1s左右，在此时间窗内上一个小区操作符动作基本已经处理完成。

3 离散化操作符的处理，有利于操作符合并处理。当一个小区频繁的触发产生大量操作符时，可能会导致操作符队列中存在连续的几个触发原因相同的操作符，而这些连续相同的操作符实际上可做合并处理。

**特别强调:** 同一小区同触发类型的参数修改时，数据库尽可能做合并触发，以减少对高层的消息冲击。



**图3-3 操作符出入队示意图**

****

**图3-4 操作符管理流程图**

### 3.4 接口设计

## 告警管理

### 小区类告警

小区类告警主要有小区退服告警、小区关断告警以及基站退服告警，告警触发场景、恢复场景如表4-1所示。

|  |  |  |
| --- | --- | --- |
| 告警类型 | 触发场景 | 恢复场景 |
| 小区退服告警 | 1)SCTP(F1/NG)断链  2)时钟状态失锁  3)基带卡/RRU运行状态异常  4)小区配置失败 | 1)SCTP(F1/NG)偶联通  2)时钟状态锁定  3)基带卡/RRU运行状态正常  4)小区配置成功 |
| 小区关断告警 | 网管操作关断小区 | 网管操作解关断小区 |
| 基站退服告警 | 所有小区均不在服 | 至少存在一个小区在服 |

**表4-1 告警类型**

### 告警策略

#### 4.2.1 告警上报

小区告警上报是在小区建立过程中发生的，属于流程式上报。小区建立流程检测到告警触发场景满足时就会上报相应的告警，如建立流程中检测到NG链路不通上报NG链路断引起的小区退服告警。告警上报采用”AlarmCode+SubAlarmCode(告警码+告警子码)”的组合上报模式，其中AlarmCode表示告警基本类型，SubAlarmCode表示该类型告警上报的原因，具体如表4-2所示。

|  |  |  |
| --- | --- | --- |
| 告警类型 | AlarmCode | SubAlarmCode |
| 小区退服告警 | CELL\_OUT\_OF\_SERVICE | 1)F1断链:F1\_SHUTDOWN  2)NG断链:NG\_SHUTDOWN  3)时钟失锁:CLOCK\_OUT\_OF\_SYN  4)基带卡运行状态异常:BASE\_BOARD\_ABNORMAL  5)RRU运行状态异常:RRU\_ABNORMAL  6)小区配置失败:CELL\_CFG\_FAIL |
| 小区关断告警 | CELL\_SHUTDOWN | CELL\_SHUTDOWN |
| 基站退服告警 | GNB\_OUTOF\_SERVICE | 1)同小区退服告警类SubAlarmCode 1)~4）  2)NO\_AVAILABLE\_CELL |

**表4-2 告警码+告警子码表**

**需要注意事项:**

1)CELL\_CFG\_FAIL进一步原因可在告警附加文本信息中以ErrorCode的形式展现，ErrorCode需要参与小区建立的各个模块具体给出。

2)某一告警触发场景引起小区反复尝试建立时，为了避免告警的反复上报引起消息风暴，业务侧上报时应该做去重处理，即已经上报过该告警时就不要再次上报了。

#### 4.2.2 告警恢复

小区告警恢复亦是在小区建立过程中发生的，属于流程式恢复。小区建立流程检测到告警恢复场景满足时就会恢复相应的告警，如建立流程中检测到时钟状态锁定就恢复由于时钟失锁引起的小区退服告警。

需要注意事项:业务侧检测到某一告警恢复场景发生时会无条件恢复该告警，而有可能该告警并未上报过。为了减少反复恢复引起消息风暴，业务侧恢复时应该做校准处理，即没有上报过该告警时就不要尝试恢复了。

#### 4.2.3 告警优先级

同一告警类型的告警子码是有优先级的，高优先级子码告警上报时应屏蔽掉低优先级子码告警，直至高优先级子码告警恢复后再显示低优先级子码告警。

告警子码(不区分告警类型)优先级如下所示:

CLOCK\_OUT\_OF\_SYN>F1\_SHUTDOWN>NG\_SHUTDOWN>BASE\_BOARD\_ABNORMAL> RRU\_ABNORMAL>NO\_AVAILABLE\_CELL>CELL\_SHUTDOWN >CELL\_CFG\_FAIL

#### 4.2.4 告警时延

高优先级告警一般需要立即上报，此时可能整个基站已不能提供服务。部分低优先级告警可延时上报，主要考虑在延时期间内有恢复的可能。

|  |  |  |
| --- | --- | --- |
| SubAlarmCode | 是否延时上报 | 延时时长/s |
| CLOCK\_OUT\_OF\_SYN | 否 | 0 |
| F1\_SHUTDOWN | 否 | 0 |
| NG\_SHUTDOWN | 否 | 0 |
| BASE\_BOARD\_ABNORMAL | 否 | 0 |
| RRU\_ABNORMAL | 是 | 1 |
| NO\_AVAILABLE\_CELL | 是 | 1 |
| CELL\_SHUTDOWN | 是 | 2 |
| CELL\_CFG\_FAIL | 是 | 2 |

**表4-3 告警时延表**

#### 4.2.5 告警防抖

SCTP链路告警的上报/恢复需做防抖机制，因为链路的不稳定性可能导致告警的频繁上报/恢复而产生大量消息交互，影响系统整体性能，

防抖机制: SCTP断链触发的上报告警，每隔一定周期(3s)平台需检测一次是否告警待恢复队列中存在该告警，如果存在时进行告警恢复即可。

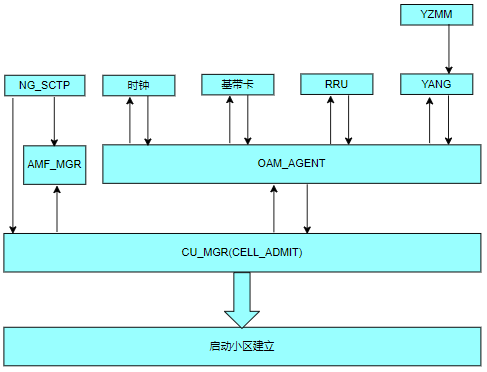
#### 4.2.6 接口设计

## 状态接纳 (选做)

状态接纳是小区建立的先决条件，接纳的内容(按接纳顺序给出)主要包括时钟状态监测、NG链路状态监测、基带卡运行状态监测、RRU运行状态监测、小区人工操作状态监测等。小区建立初期，某一项状态接纳失败时需上报相应的子码原因告警，接纳成功则恢复相应的子码原因告警，只有所有状态接纳成功后才允许发起小区建立流程。

#### 5.1.1 数据流图

状态接纳各模块间数据流图如下图所示，图中上行箭头代表主动获取状态，比如系统初始上电；下行箭头代表状态变更通知。



**图5-1各模块间状态接纳数据流图**

#### 5.1.2 信号流图

以NG链路闪断为例，状态接纳信号流图如图5-2所示。

1、NG\_SCTP链路断通知AMF模块存储当前断链状态。

2、NG\_SCTP链路断通知CU\_MGR发起删除动作，DEL\_OPR操作符入队列。

3、周期定时器到时取出DEL\_OPR操作符执行删除动作。

4、通知CU/DU/L1删除各模块小区实例。

5、删除完成后小区去激活状态入数据库。

6、删除原因为由于NG\_SCTP断导致，且数据库小区实例存在，再次尝试发起小区建立，ADD\_OPR操作符入队列。

7、小区去激活状态同步至网管。

8、周期定时器到时取出ADD\_OPR操作符执行建立动作。

9、小区建立前ADMIT\_MGR模块先做NG\_SCTP链路状态接纳。

10、NG\_SCTP链路不通，接纳失败。

11、各模块小区实例尚未创建，无需置删除操作，且数据库小区实例存在，再次尝试发起小区建立，ADD\_OPR操作符入队列。

12、NG\_SCTP链路接纳失败，上报由NG\_SCTP断链触发的小区退服告警。

13、告警上报同步至网管。

14、NG\_SCTP断链触发的小区不在服时长KPI指标开始统计。

15、NG\_SCTP链路通通知AMF\_MGR模块存储当前通链状态。

16、NG\_SCTP通链状态同步至ADMIT\_MGR模块存储。

17、周期定时器到时取出ADD\_OPR操作符执行建立动作。

18、小区建立前ADMIT\_MGR模块先做NG\_SCTP链路状态接纳。

19、状态接纳成功，通知CU/DU/L1建立各模块小区实例。

20、建立成功后小区激活状态入数据库。

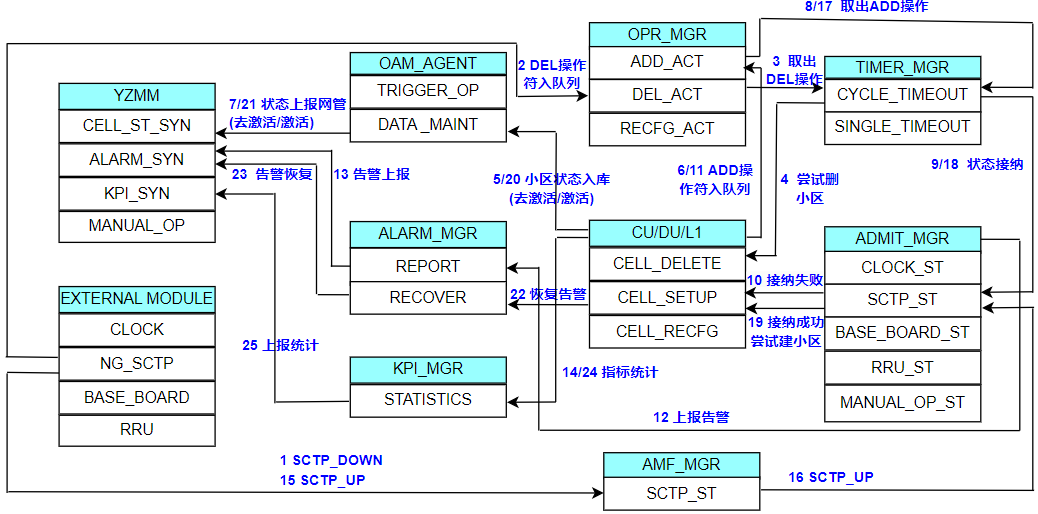
21、小区激活状态同步至网管。

22、小区建立成功恢复NG\_SCTP断链引起的小区退服告警。

23、告警恢复同步至网管。

24、NG\_SCTP断链触发的小区不在服时长KPI指标停止统计。

25、NG\_SCTP断链触发的小区不在服时长KPI指标同步至网管。



**图5-2 NG链路闪断状态接纳信号流图**

#### 5.1.3 接口设计

## RRU小区管理

上层通过调用RFIC API 函数来建立或关闭小区，其中建立小区需要重启和初始化整个RFIC，该过程大约会耗时25s左右。用户调整中心频点、发射功率、通道增益等参数时只需RFIC处于工作状态即可，不需要重启整个RFIC。其中中心频点参数调整后生效时长大约5s左右，发射功率、通道增益等参数调整后即时生效；其余射频参数调整需要关闭RRU小区后再以最新配置重新建立RRU小区。

#### 6.1.1 RFIC API接口设计



2、射频参数生效策略

#### 6.1.2 射频参数生效策略

由RFIC API接口设计文档可以看出，RRU射频参数生效方式和生效时长不尽相同。下表整理出了小区射频参数生效策略，以供参考。



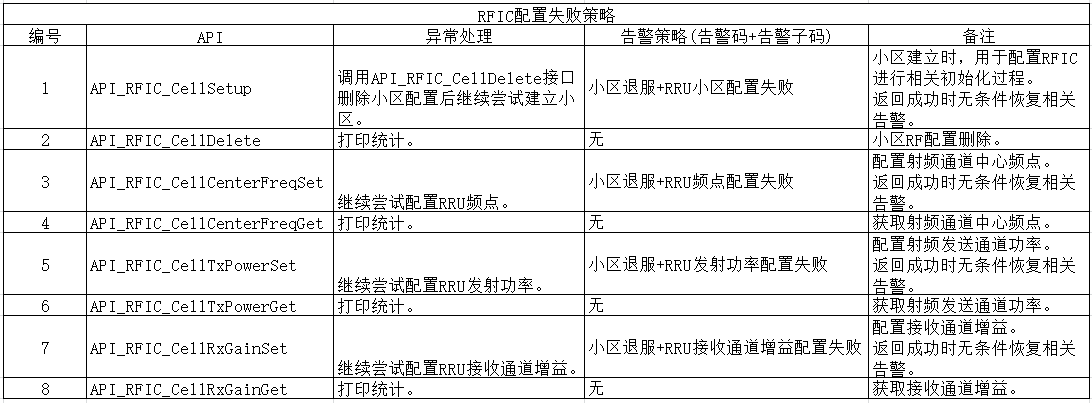
**表6-1 小区射频参数生效策略**

#### 6.1.3 RFIC配置失败策略

小区射频参数配置是小区管理建立流程中的首要环节，配置失败时：

1、上报由于配置失败导致的小区退服告警。

2、从数据库获取射频配置参数继续尝试配置，直至用户修改为正确配置后小区建立流程继续，小区建立完成后恢复相应告警。



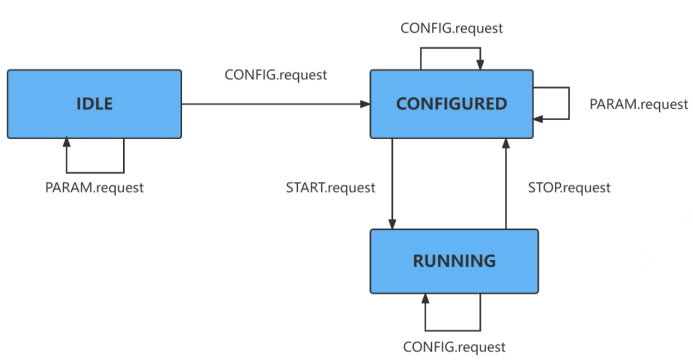
**表6-2 RFIC配置失败策略**

## PHY小区管理

小区管理过程中，DU与PHY之间的通过FAPI接口消息交互小区相关数据，最终完成PHY小区建立或小区停服。PHY对于DU小区相关配置消息可在不同状态机下处理，下面做一简单介绍。

#### 7.1.1 状态机设计

物理层状态机分三种状态，分别为空闲状态(Idle)、已配置状态(Configured)和运行状态(Running)，状态间的转换关系如图7-1所示，不同状态下支持的配置请求消息见表7-2所述。



**图7-1 物理层状态机转换**



**表7-2 不同状态下支持的配置请求消息**

#### 7.1.2 配置消息

1、PARAM消息交互流程: MAC发送PARAM.request消息至PHY，通过握手消息来识别FAPI链路是否通路。

**特别强调:** 思朗方案中提到PARAM.response消息会指示PHY配置及当前状态信息，实际上建小区时该响应消息只反馈配置成功与否。

2、CONFIG消息交互流程: MAC发送CONFIG.request消息携带小区配置参数，PHY进行小区实例创建。

3、START流程由MAC发送START.request消息来发起，指示PHY激活已完成配置的小区，状态机由已配置状态(Configured)跳转为运行状态(Running)。

4、STOP流程由MAC发送STOP.request消息来发起，指示PHY停止上下行业务，状态机由运行状态(Running)跳转为已配置状态(Configured)。

**特别注意:** 不同状态下支持的配置请求消息严格按照表7-2执行，否则反馈消息给出Error code=MSG\_INVALID\_STATE指示。

#### 7.1.3 交互流程

1、小区建立

(1) CL\_CMN模块通过FAPI接口发送PHY\_PARAM\_REQUEST消息至PHY\_CMN模块，识别FAPI链路是否通路。

(2) FAPI链路通，CL\_CMN模块发送PHY\_CONFIG\_REQUEST消息至PHY\_CMN模块，携带小区配置相关参数，通知PHY\_CMN创建小区实例。

(3) CL\_CMN模块发送PHY\_START\_REQUEST消息至PHY\_CMN模块激活小区实例。

2、小区删除

CL\_CMN模块通过FAPI接口发送PHY\_STOP\_REQUEST消息至PHY\_CMN模块，通知PHY停止上下行业务。

**特别强调:** PHY小区实例一旦创建无法删除。

3、小区重配/复位

为实现简单起见，PHY参数修改触发小区重配或小区复位，可采用相同的一套流程。

(1) CL\_CMN模块通过FAPI接口发送PHY\_STOP\_REQUEST消息至PHY\_CMN模块，通知PHY停止上下行业务。

**特别强调:** PHY\_CMN不在Running态时，STOP响应可能会回复失败，不影响后续流程继续执行。

(2) CL\_CMN模块通过FAPI接口发送PHY\_PARAM\_REQUEST消息至PHY\_CMN模块，识别FAPI链路是否通路。

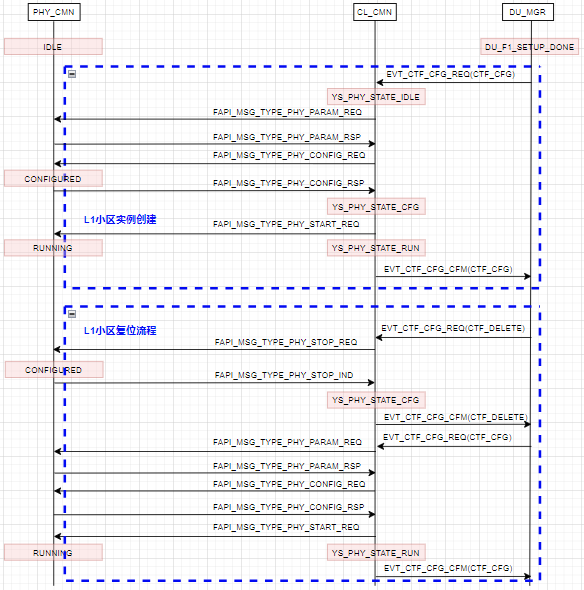
**特别强调:** PHY\_CMN在Running态时，PARAM响应可能会回复失败，不影响后续流程继续执行。

(3) FAPI链路通，CL\_CMN模块发送PHY\_CONFIG\_REQUEST消息至PHY\_CMN模块，携带最新小区配置相关参数，通知PHY\_CMN重配小区实例。

(4) CL\_CMN模块发送PHY\_START\_REQUEST消息至PHY\_CMN模块激活小区实例。

4、异常流程

由第3点可以推出，无论小区建立、删除(Stop)还是重配失败时，可采用统一处理策略，即删除时发送**StopReq**，再次尝试建立或重配时按照**ParamReqt**🡪**ConfigReq**🡪**StartReq**消息流处理。



**图7-3 小区建立/复位(含删除)消息交互时序**

#### 7.1.4 接口设计

## 业务流程设计

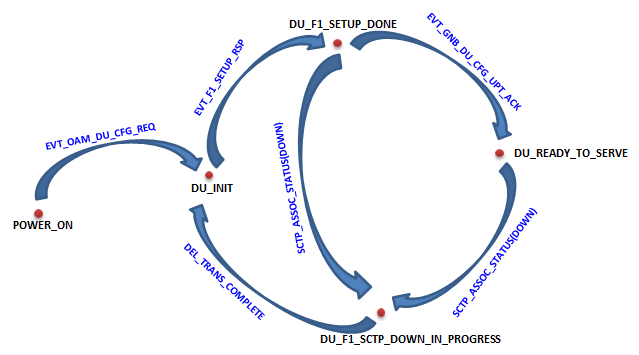
### 状态机介绍

小区管理流程依附于各主控模块状态机的正常运行，小区的建立、删除及重配等动作只在某些特定的状态才能触发。因此，状态机的设计一定要具备自洽性/完备性，即进入异常状态时有恢复到正常状态的可能，尽可能避免进入僵尸状态。

该小节介绍一下与小区管理息息相关的几个状态机，它们分别是DU\_MGR状态机、CU\_MGR状态机、DU\_INST状态机以及AMF\_INST状态机。

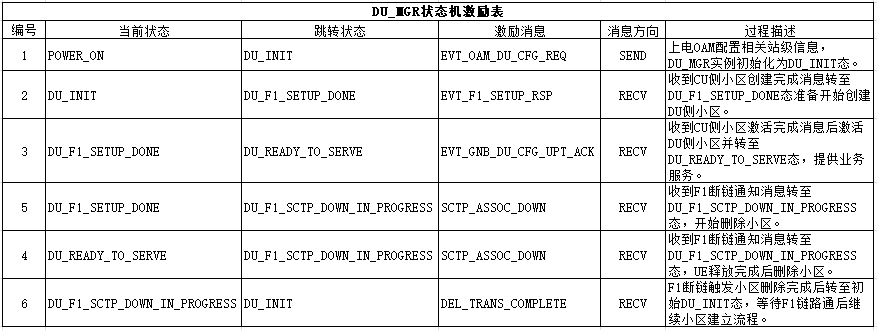
#### DU\_MGR状态机

DU\_MGR状态机是以基站F1AP维护过程为基准设计的，涉及的状态有DU\_INIT态、DU\_F1\_SETUP\_DONE态、DU\_READY\_TO\_SERVE态、DU\_F1\_SCTP\_DOWN\_IN\_PROGRESS态。



**图8-1 DU\_MGR状态机转换示意图**

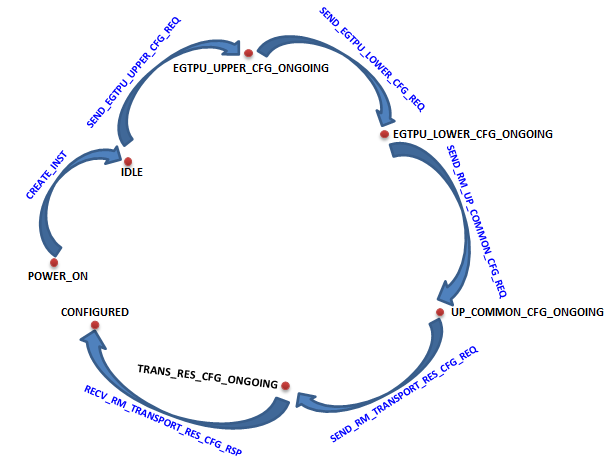
由图8-1可以看出，DU\_MGR状态机的设计具备自洽性/完备性。小区建立、删除、重配等管控流程几乎涉及所有状态(除DU\_INIT态)。不同状态的触发激励可参见表8-2。



**表8-2 DU\_MGR状态机激励表**

#### CU\_MGR状态机

CU\_MGR状态机设计较为简单，目前都是单向跳转，不支持状态回退。由于CU\_MGR状态机主要负责基站配置数据的传递工作，不存在异常状态，因此不需要考虑状态回退。CU\_MGR状态机涉及的状态有IDLE态、EGTPU\_UPPER\_CFG\_ONGOING态、EGTPU\_LOWER\_CFG\_ONGOING态、UP\_COMMON\_CFG\_ONGOING态、TRANS\_RES\_CFG\_ONGOING态、CONFIGURED态。



**图8-3 CU\_MGR状态机转换示意图**

需要强调的是，虽然CU\_MGR状态机中所有状态都不涉及小区管控流程，但是小区建立、删除、重配的前提条件是CU\_MGR状态机一定要处于CONFIGURED态。不同状态的触发激励可参见表8-4。

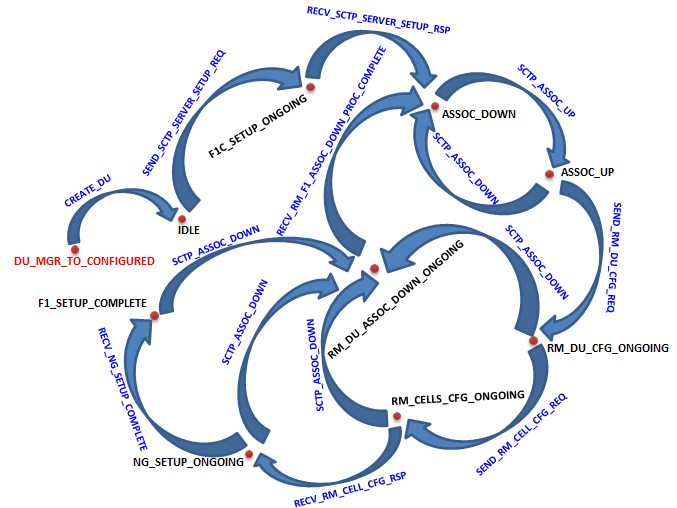


**表8-4 CU\_MGR状态机激励表**

#### DU\_INST状态机

DU\_INST状态机是以基站F1AP维护过程以及NGAP维护过程为基准设计的，涉及的状态有IDLE态、F1\_SETUP\_ONGOING态、ASSOC\_DOWN态、ASSOC\_UP态、

RM\_DU\_CFG\_ONGOING态、RM\_CELLS\_CFG\_ONGOING态、NG\_SETUP\_ONGOING态、F1\_SETUP\_COMPLETE态、RM\_DU\_ASSOC\_DOWN\_ONGOING态。



**图8-5 DU\_INST状态机转换示意图**

由图8-5可以看出，DU\_INST状态机的设计具备自洽性/完备性。小区建立、删除、重配等管控流程几乎涉及所有状态(除IDLE态)。不同状态的触发激励可参见表8-6。

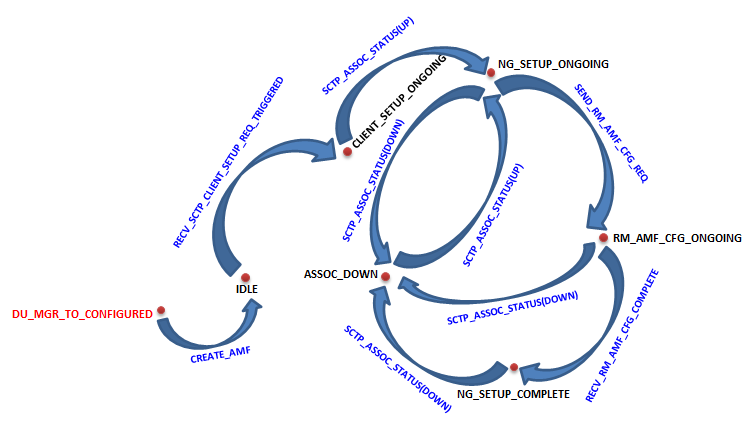


**表8-6 DU\_INST状态机激励表**

#### AMF\_INST状态机

AMF\_INST状态机是以基站NGAP维护过程为基准设计的，涉及的状态有IDLE态、CLIENT\_SETUP\_ONGOING态、NG\_SETUP\_ONGOING态、RM\_AMF\_CFG\_ONGOING态、

NG\_SETUP\_COMPLETE态、ASSOC\_DOWN态。



**图8-7 AMF\_INST状态机转换示意图**

由图8-7可以看出，AMF\_INST状态机的设计具备自洽性/完备性，NG偶联状态是小区建立、重配等动作的先决条件。不同状态的触发激励可参见表8-8。



**表8-8 AMF\_INST状态机激励表**

### 小区建立

小区建立基本流程按照以下几个步骤进行：

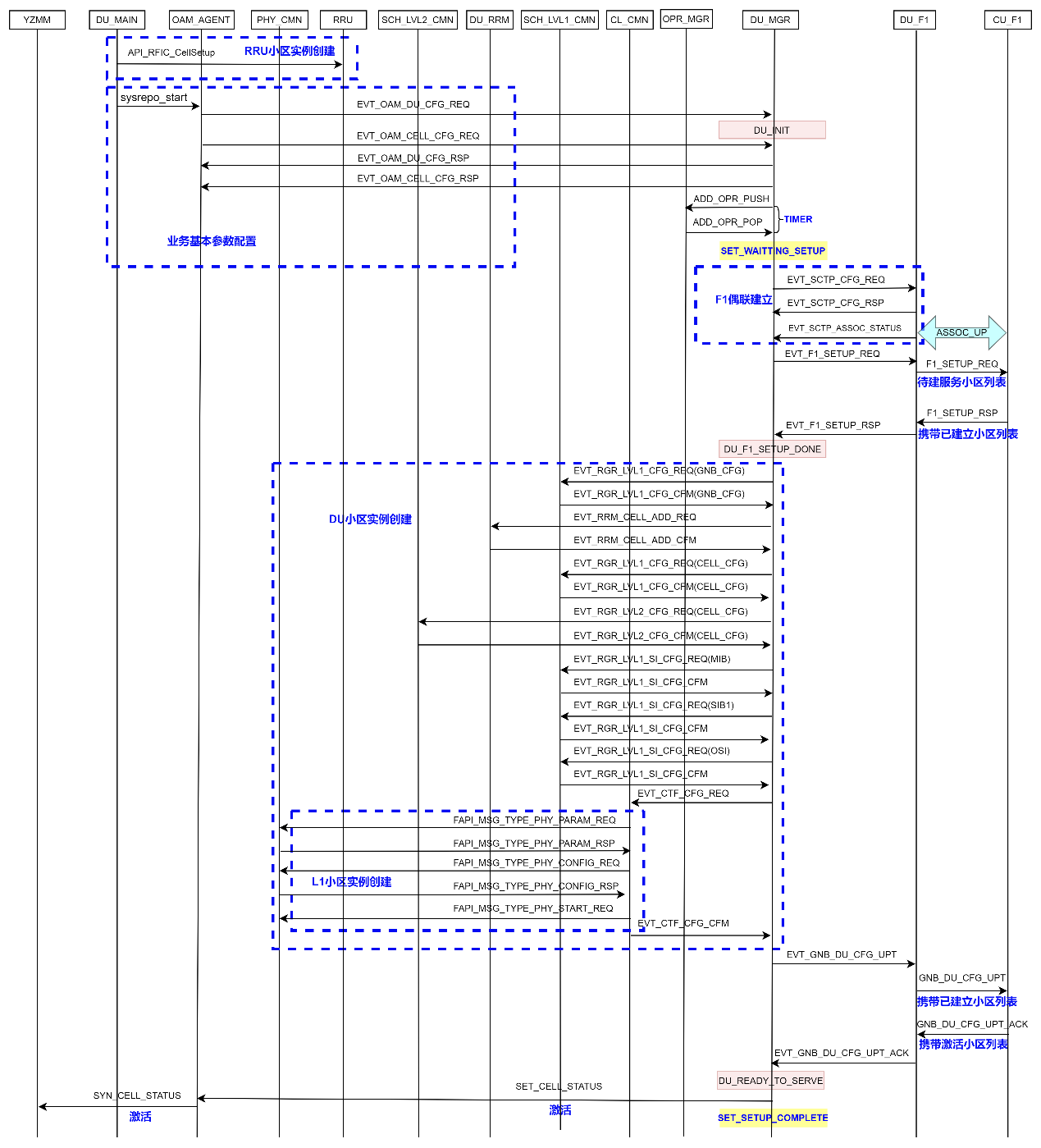
1. 小区数据同步

(1) 初始上电，OAM\_AGENT从YANG模型配置获取小区数据并同步至业务数据区，触发小区建立流程，建立原因为INIT\_ADD\_CELL。

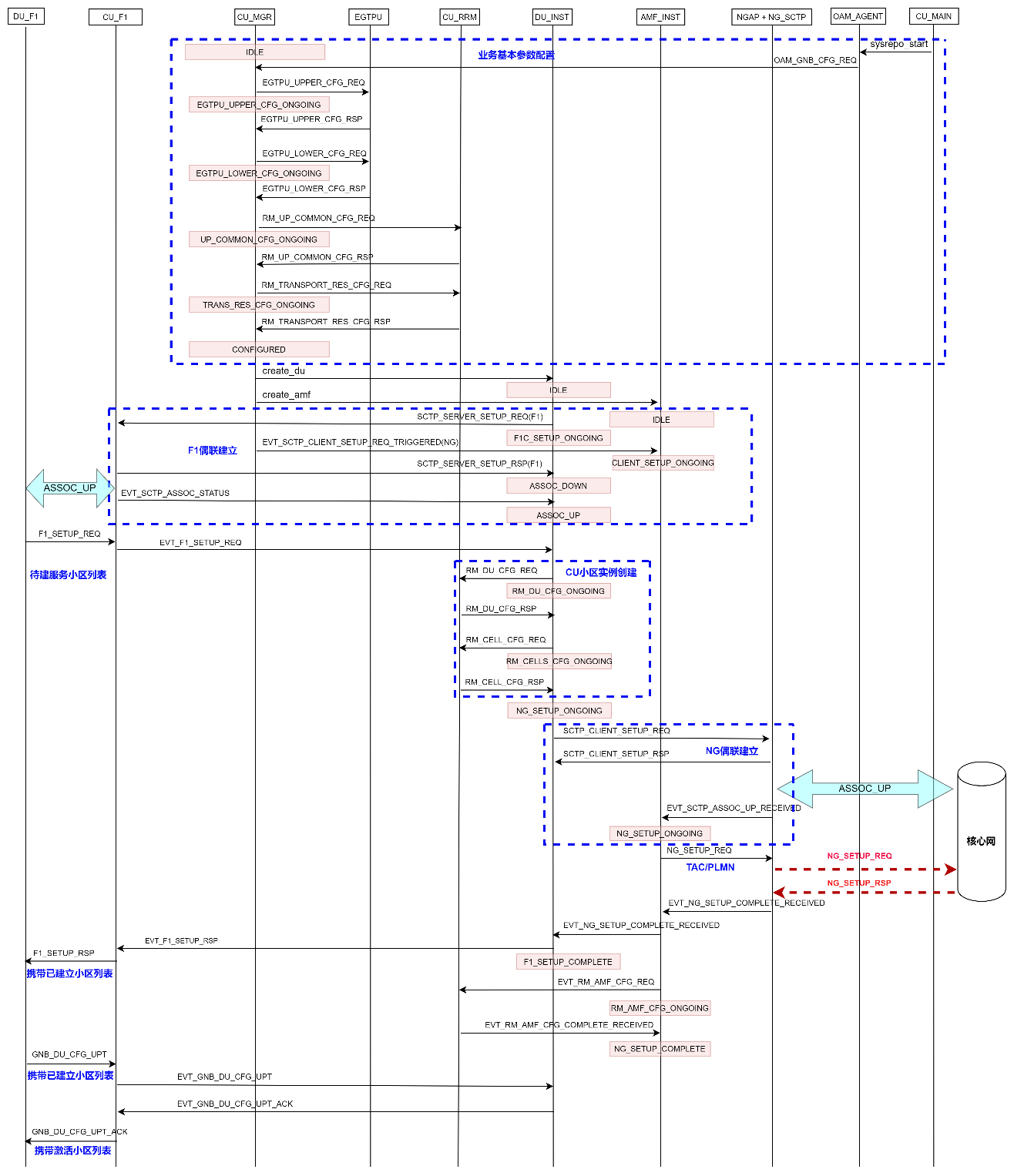
(2) 网管动态新增小区, 小区数据入库并同步至业务数据区，触发小区建立流程，建立原因为DBS\_ADD\_CELL。

1. RRU小区建立策略: 初始上电或网管动态新增小区，RRU创建小区实例。
2. OPR\_ADD操作符入队列，周期定时器超时取出接纳，接纳成功后执行建立动作。
3. F1偶联建立过程是CU&DU小区建立信息同步的基础，且后续的业务依赖于F1链路。
4. CU依据F1口携带的待建小区列表创建CU小区实例。
5. NG偶联建立是小区建立的先决条件，且后续的业务依赖于NG链路。
6. DU依据F1口携带的CU已建小区列表创建DU小区实例。
7. CL\_CMN模块发送FAPI接口消息通知PHY创建小区实例。
8. 小区激活，状态同步至网管。

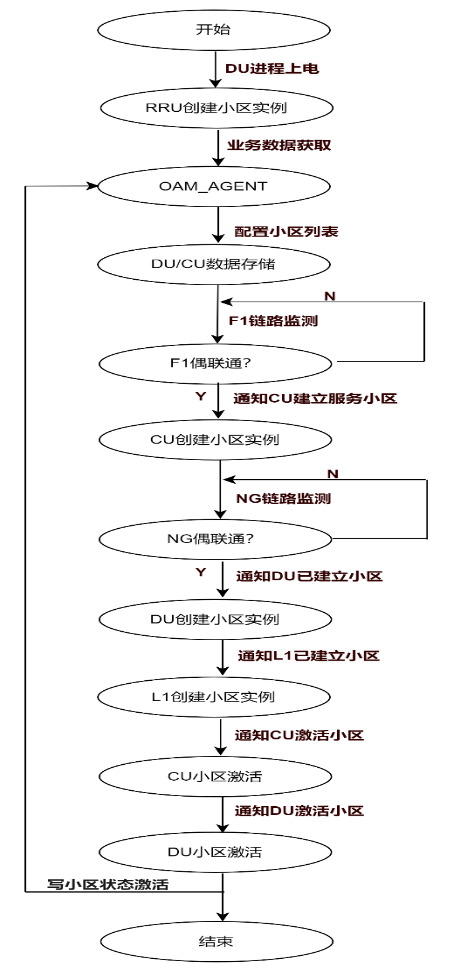
#### 8.2.1 初始上电



**图8-9 初始上电DU小区建立消息交互时序图**

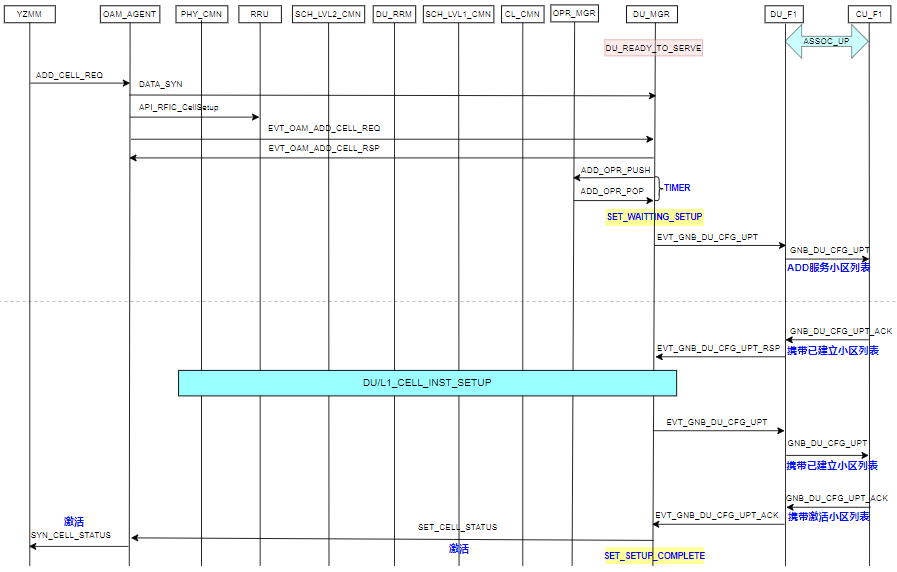


**图8-10 初始上电CU小区建立消息交互时序图**

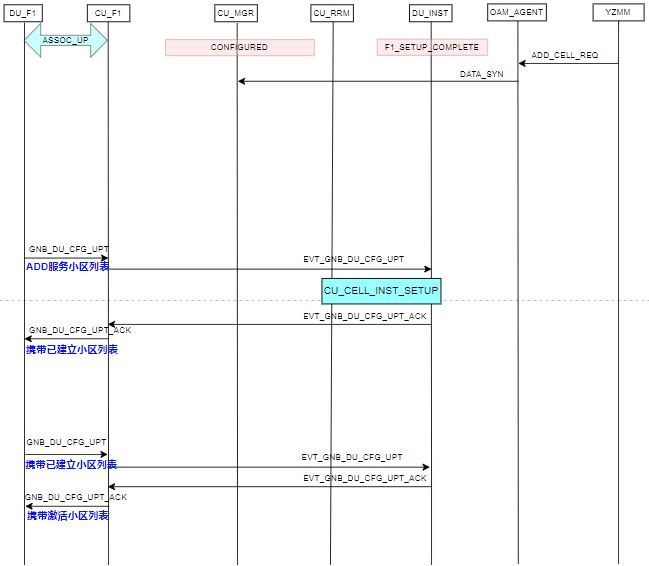


**图8-11 初始上电小区建立流程图**

#### 8.2.2 动态新增(选做)



**8-12 动态新增DU小区建立消息交互时序图**



**图8-13 动态新增CU小区建立消息交互时序图**

### 小区删除(选做)

小区删除基本流程按照以下几个步骤进行：

1、 网管动态删除小区, 清数据库小区实例，触发小区删除流程，删除原因为DBS\_DEL\_CELL。

2、RRU小区实例删除。

1. OPR\_DEL操作符入队列，周期定时器超时取出接纳，接纳成功后执行删除动作。
2. 小区删除前处于稳态(DU\_MGR状态机为DU\_READY\_TO\_SERVE态)时：

(1) CU小区实例删除采取进程复位策略，此时会导致F1偶联断。

(2) DU执行BARRED小区操作，禁止UE接入或切入。

(3) 释放在线UE。

(4) CL\_CMN模块发送FAPI接口消息通知PHY停止小区上下行业务。

(5) DU业务各模块删除小区实例。

1. 小区删除前DU\_MGR状态机为DU\_F1\_SETUP\_DOWN态时：

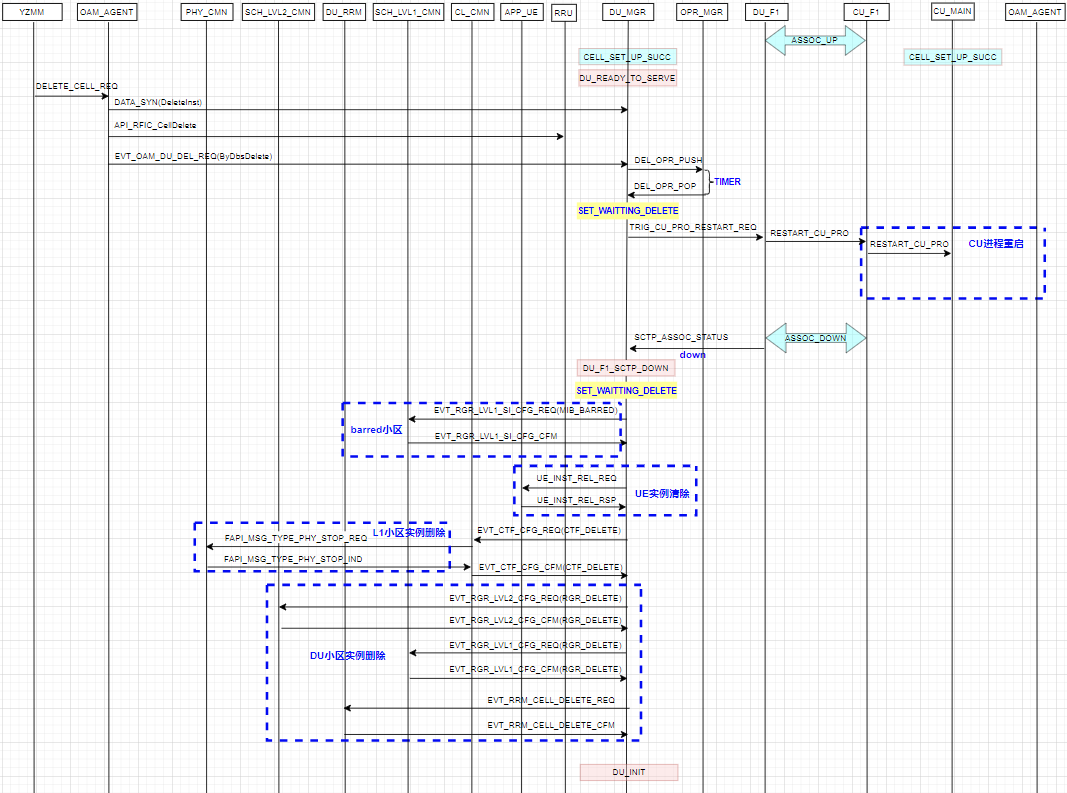
(1) CU小区实例删除采取进程复位策略，此时会导致F1偶联断。

(2) CL\_CMN模块发送FAPI接口消息通知PHY停止小区上下行业务。

(3) DU业务各模块删除小区实例。

6、小区删除前DU\_MGR状态机为DU\_INIT态时：CU进程软重启。

7、小区删除原因为DBS\_DEL\_CELL，清DU小区实例后流程结束。



**图8-14 动态删除小区(DU\_READY\_TO\_SERVE态)消息交互时序图**

### 小区复位

一、DU参数修改小区复位基本流程按照以下几个步骤进行：

1、网管修改小区复位参数，新参数入库并同步至业务数据区，同时触发小区删除流程，删除原因为PARA\_MOD\_DEL\_CELL。

2、RRU小区删除策略。

(1) 中心频点、发射功率、通道增益等射频参数修改，RRU小区做相应参数重配更新动作。

(2) 非(1)提及的其它射频参数修改，RRU小区实例先做删除操作后再次尝试建立。

3、OPR\_DEL操作符入队列，周期定时器超时取出接纳，接纳成功后执行删除动作。

4、小区删除前处于稳态(DU\_MGR状态机为DU\_READY\_TO\_SERVE态)时：

(1) CU小区实例删除采取进程复位策略，此时会导致F1偶联断。

(2) DU执行BARRED小区操作，禁止UE接入或切入。

(3) 释放在线UE。

(4) CL\_CMN模块发送FAPI接口消息通知PHY停止小区上下行业务。

(5) DU业务各模块删除小区实例。

5、小区删除前DU\_MGR状态机为DU\_F1\_SETUP\_DOWN态时：

(1) CU小区实例删除采取进程复位策略，此时会导致F1偶联断。

(2) CL\_CMN模块发送FAPI接口消息通知PHY停止小区上下行业务。

(3) DU业务各模块删除小区实例。

6、小区删除前DU\_MGR状态机为DU\_INIT态时不做任何动作。

7、小区去激活，状态同步至网管。

8、小区删除原因为PARA\_MOD\_DEL\_CELL，OPR\_ADD操作符入队列，建立原因为DEL\_ADD\_CELL，周期定时器超时取出接纳，接纳成功后执行建立动作。

9、F1偶联建立过程是CU&DU小区建立信息同步的基础，且后续的业务依赖于F1链路。

10、CU依据F1口携带的待建小区列表创建CU小区实例。

11、NG偶联建立是小区建立的先决条件，且后续的业务依赖于NG链路。

12、DU依据F1口携带的CU已建小区列表创建DU小区实例。

13、CL\_CMN模块发送FAPI接口消息通知PHY创建小区实例。

14、小区激活，状态同步至网管。

二、CU参数修改小区复位基本流程按照以下几个步骤进行：

1、网管修改小区复位参数，CU小区实例删除采取进程复位策略。

2、小区删除前处于稳态(DU\_MGR状态机为DU\_READY\_TO\_SERVE态)时收到F1偶联断通知，启动小区删除流程，删除原因为F1\_ASSOC\_DOWN\_DEL\_CELL：

(1) DU执行BARRED小区操作，禁止UE接入或切入。

(2) 释放在线UE。

(3) CL\_CMN模块发送FAPI接口消息通知PHY停止小区上下行业务。

(4) DU业务各模块删除小区实例。

3、小区删除前DU\_MGR状态机为DU\_F1\_SETUP\_DOWN态时收到F1偶联断通知，启动小区删除流程，删除原因为F1\_ASSOC\_DOWN\_DEL\_CELL：

(1) CL\_CMN模块发送FAPI接口消息通知PHY停止小区上下行业务。

(2) DU业务各模块删除小区实例。

4、小区删除前DU\_MGR状态机为DU\_INIT态时不做任何动作。

5、小区去激活，状态同步至网管。

6、OPR\_ADD操作符入队列，建立原因为DEL\_ADD\_CELL，周期定时器超时取出接纳，接纳成功后执行建立动作。

7、F1偶联建立过程是CU&DU小区建立信息同步的基础，且后续的业务依赖于F1链路。

8、CU依据F1口携带的待建小区列表创建CU小区实例。

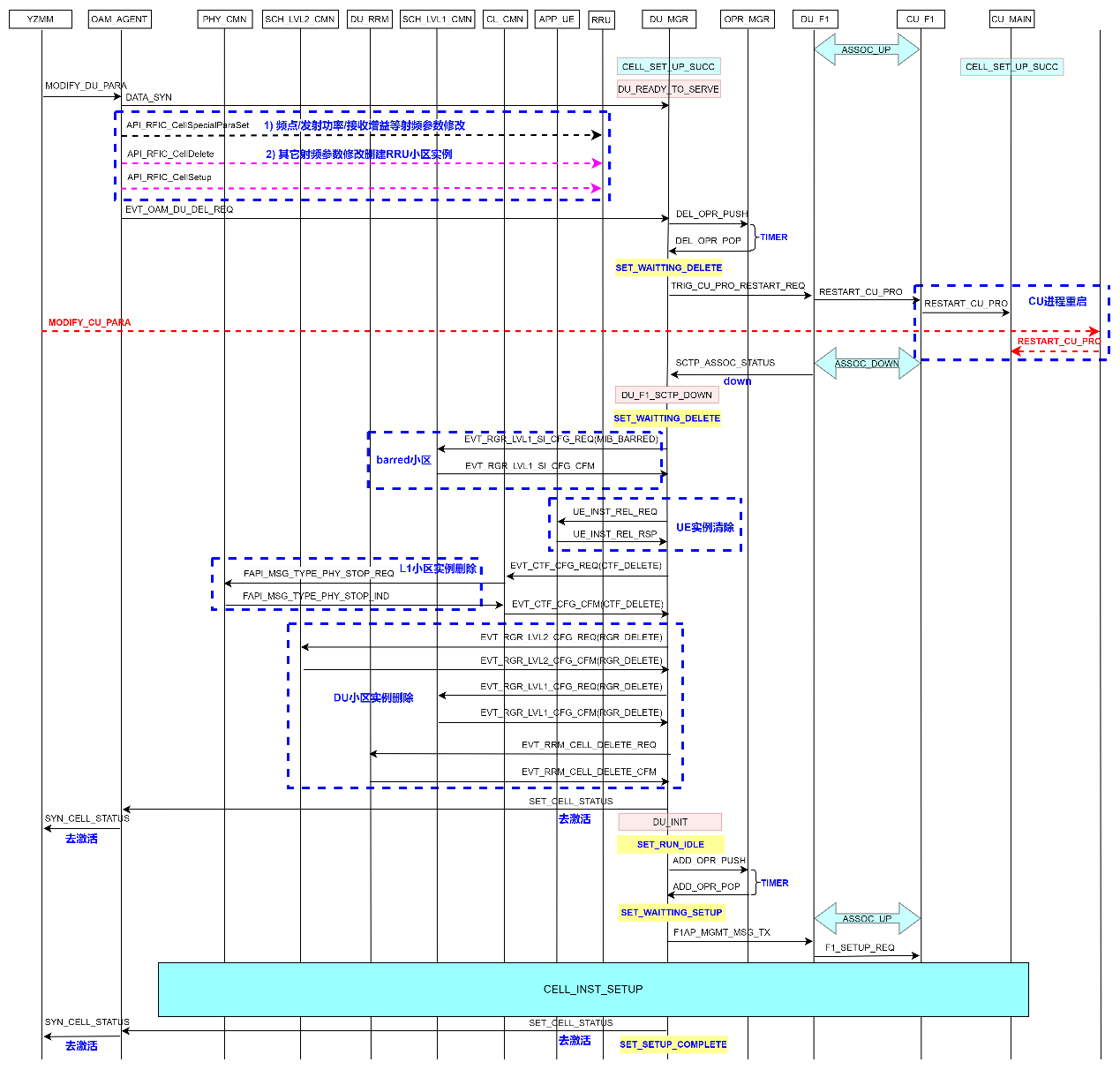
9、NG偶联建立是小区建立的先决条件，且后续的业务依赖于NG链路。

10、DU依据F1口携带的CU已建小区列表创建DU小区实例。

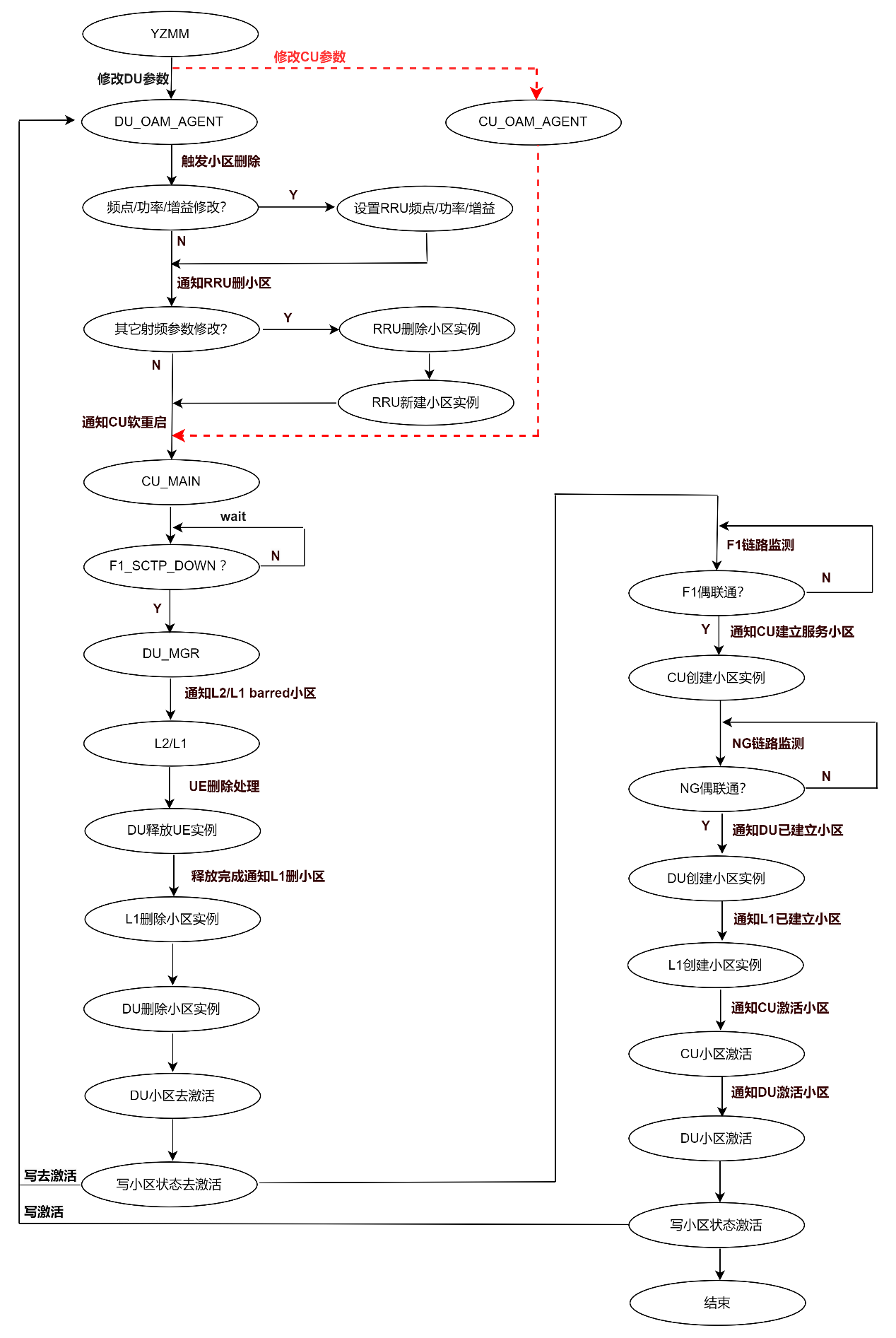
11、CL\_CMN模块发送FAPI接口消息通知PHY创建小区实例。

12、小区激活，状态同步至网管。

#### 8.4.1 小区稳态复位

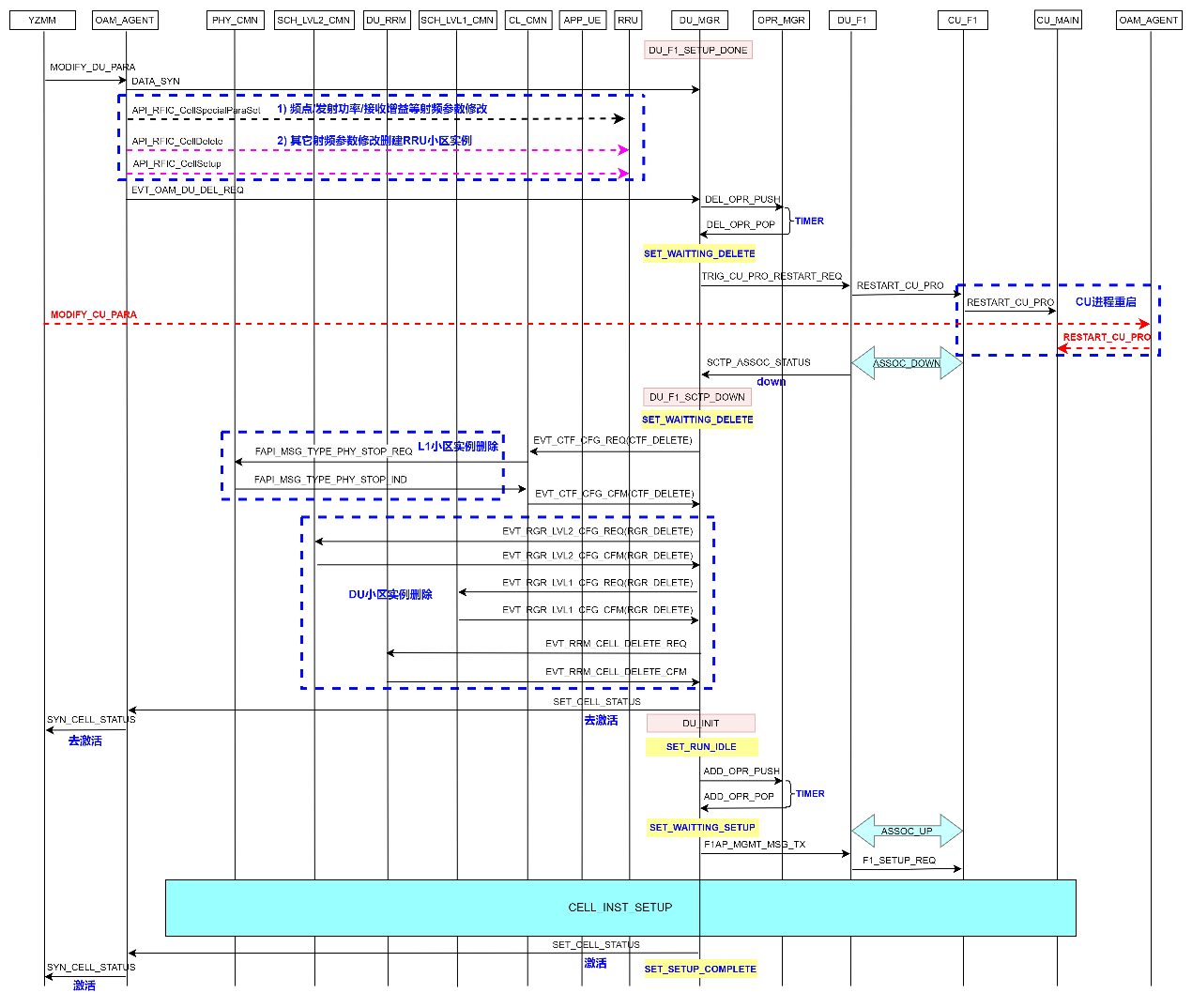


**图8\_15 小区稳态复位消息交互时序图**

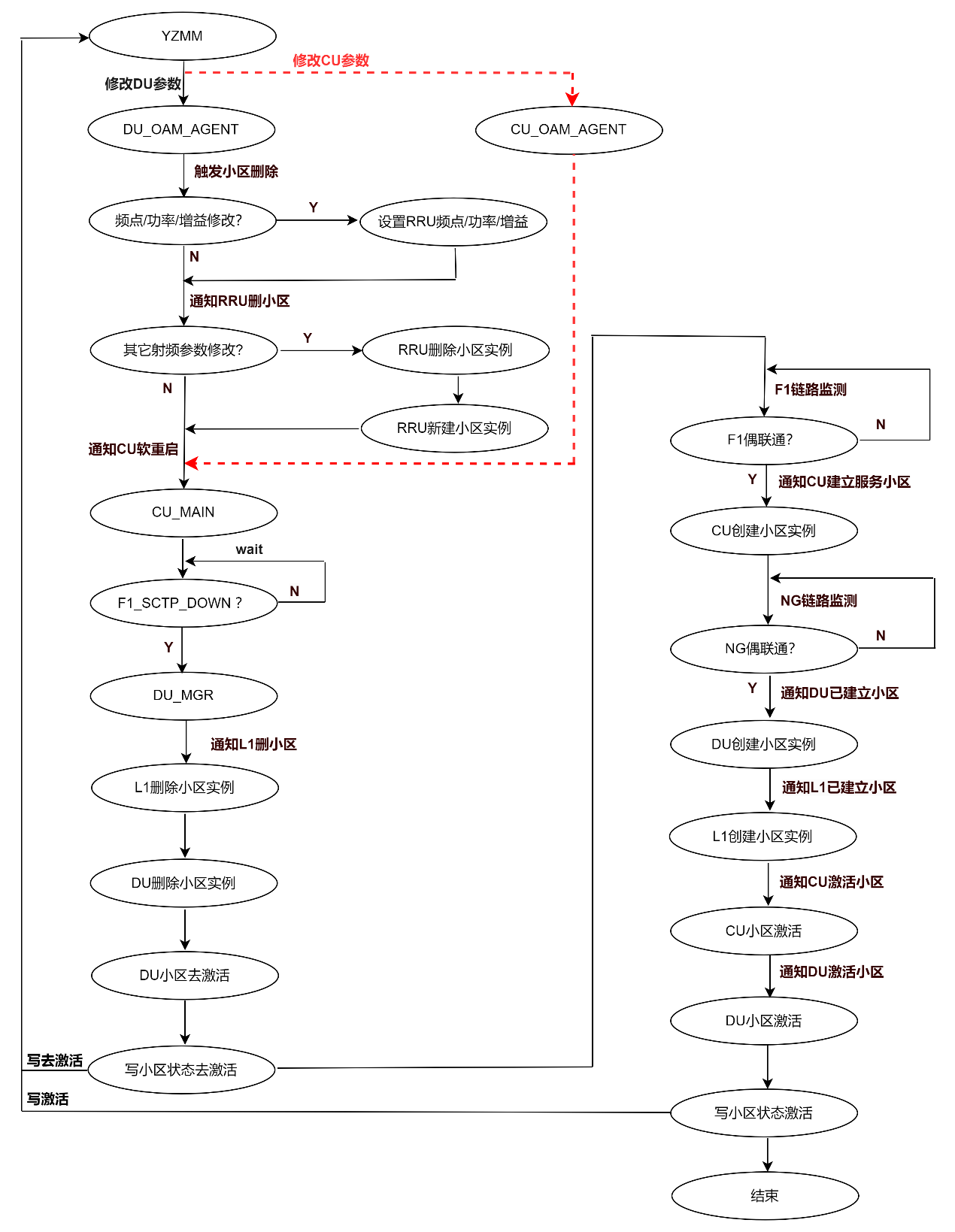


**图8\_16小区稳态复位流程图**

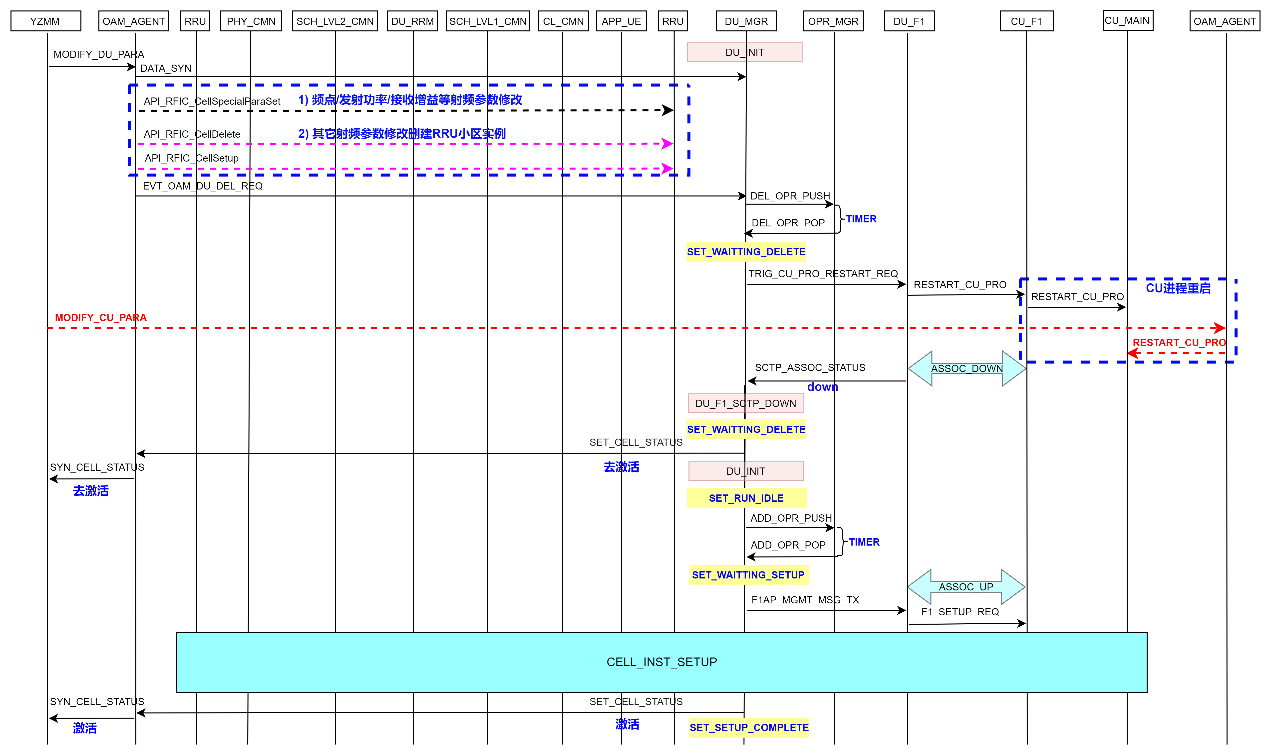
#### 8.4.2 小区非稳态复位



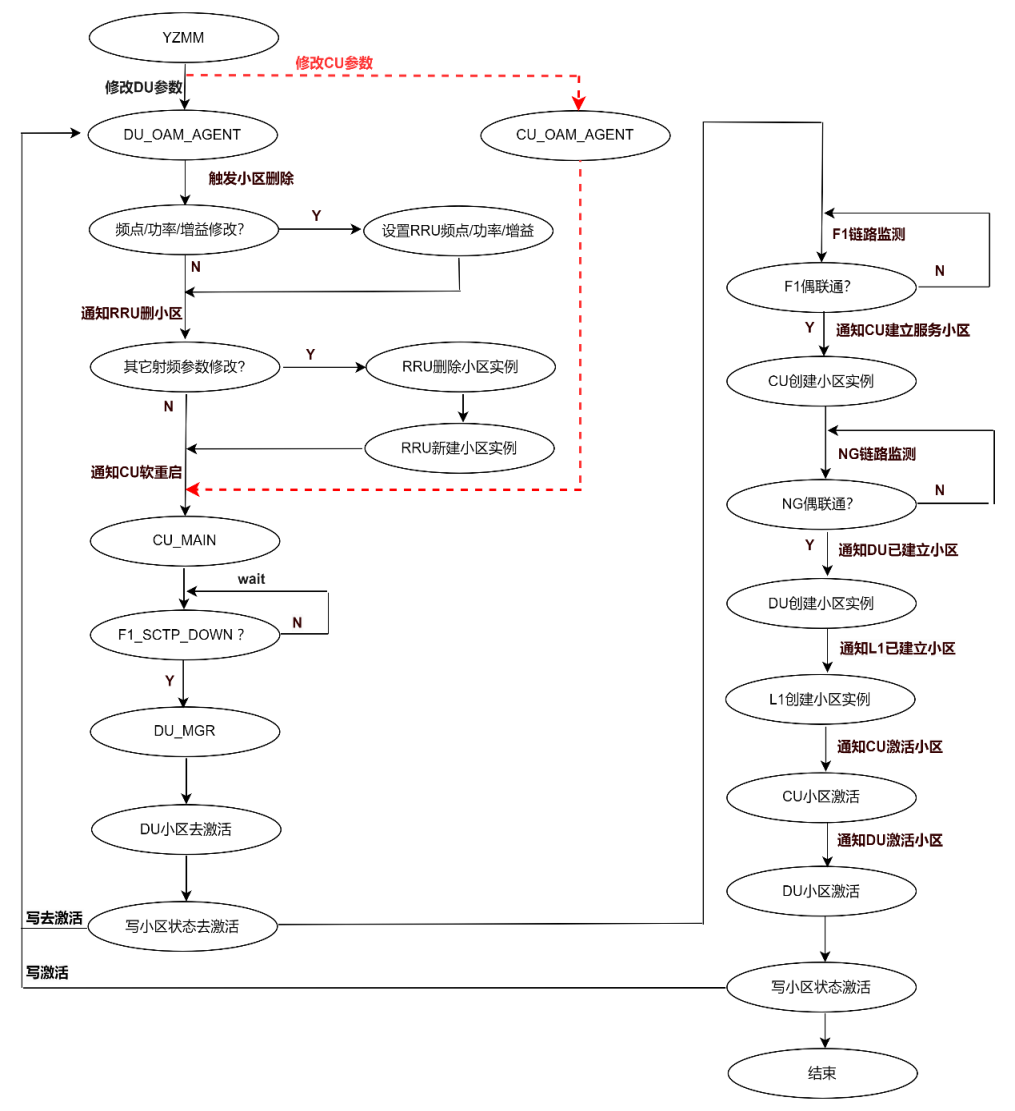
**图8-17 小区非稳态复位(DU\_F1\_SETUP\_DOWN)消息交互时序图**



**图8-18小区非稳态复位(DU\_F1\_SETUP\_DOWN)流程图**



**图8-19 小区非稳态复位(DU\_INIT)消息交互时序图**



**图8-20小区非稳态复位(DU\_INIT)流程图**

### 小区重配(选做)

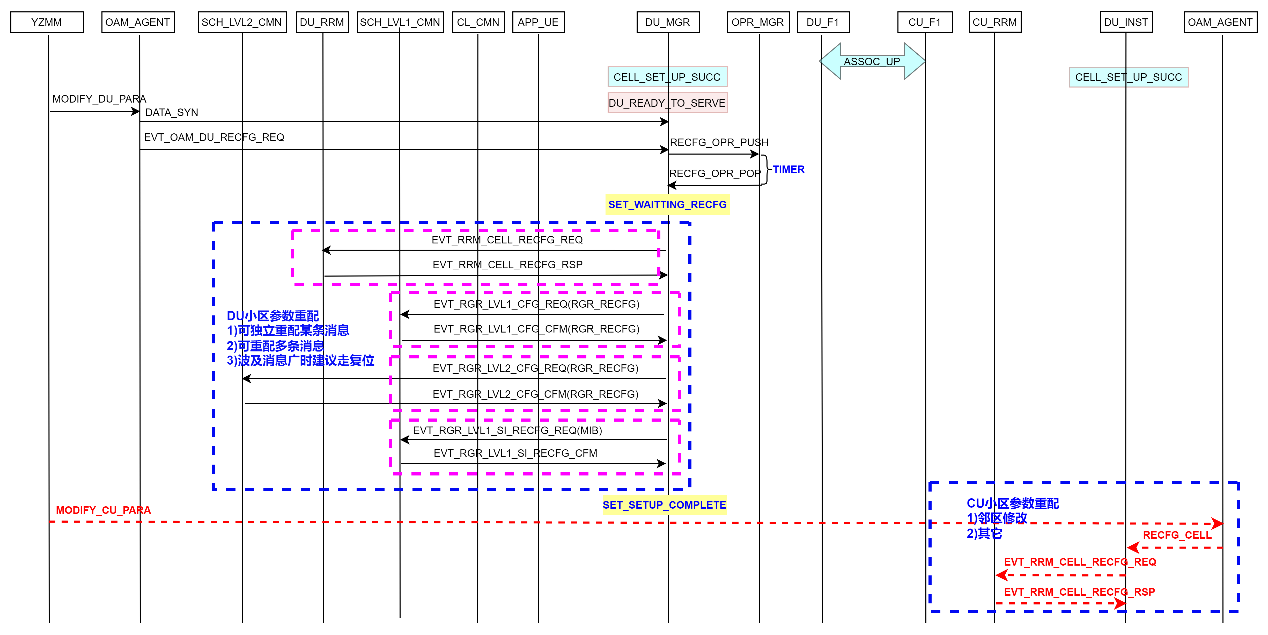
小区重配基本流程按照以下几个步骤进行：

1、网管修改小区重配参数，新参数入库并同步至业务数据区，同时触发小区重配流程，重配消息指定重配原因(重配模块+功能对象)。

2、OPR\_RECFG操作符入队列，周期定时器超时取出接纳，接纳成功后执行重配动作。

3、依据重配原因，进行相应模块和功能参数的配置更新。其中DU可重配模块为RRM、SCH\_LVL1、SCH\_LVL2、LVL1\_SI等模块，各模块功能参数需要梳理归类给出；CU可重配模块为RRM模块，主要为邻区配置参数。

特别强调：重配流程是基于小区稳态前提下才可能发生的流程。



**图8-21 小区重配消息交互时序图**

### 异常流程

#### 小区建立异常

一、DU小区建立异常基本流程按照以下几个步骤进行：

1、初始上电，OAM\_AGENT从YANG模型配置获取小区数据并同步至业务数据区，触发小区建立流程，建立原因为INIT\_ADD\_CELL。

2、RRU创建小区实例。

3、OPR\_ADD操作符入队列，周期定时器超时取出接纳，接纳成功后执行建立动作。

4、F1偶联建立过程是CU&DU小区建立信息同步的基础，且后续的业务依赖于F1链路。

5、CU依据F1口携带的待建小区列表创建CU小区实例。

6、NG偶联建立是小区建立的先决条件，且后续的业务依赖于NG链路。

7、DU依据F1口携带的CU已建小区列表创建DU小区实例时建立失败，上报告警。

8、OPR\_DEL操作符入队列，删除原因为CFG\_FAIL\_DEL\_CELL，周期定时器超时取出接纳，接纳成功后执行删除动作。

9、小区删除前DU\_MGR状态机为DU\_F1\_SETUP\_DOWN态：

(1) CU小区实例删除采取进程复位策略，此时会导致F1偶联断。

(2) CL\_CMN模块发送FAPI接口消息通知PHY停止小区上下行业务。

(3) DU业务各模块删除小区实例。

10、小区去激活，状态同步至网管。

11、小区删除原因为CFG\_FAIL\_DEL\_CELL，OPR\_ADD操作符入队列，建立原因为DEL\_ADD\_CELL，周期定时器超时取出接纳，接纳成功后执行建立动作。

12、F1偶联建立过程是CU&DU小区建立信息同步的基础，且后续的业务依赖于F1链路。

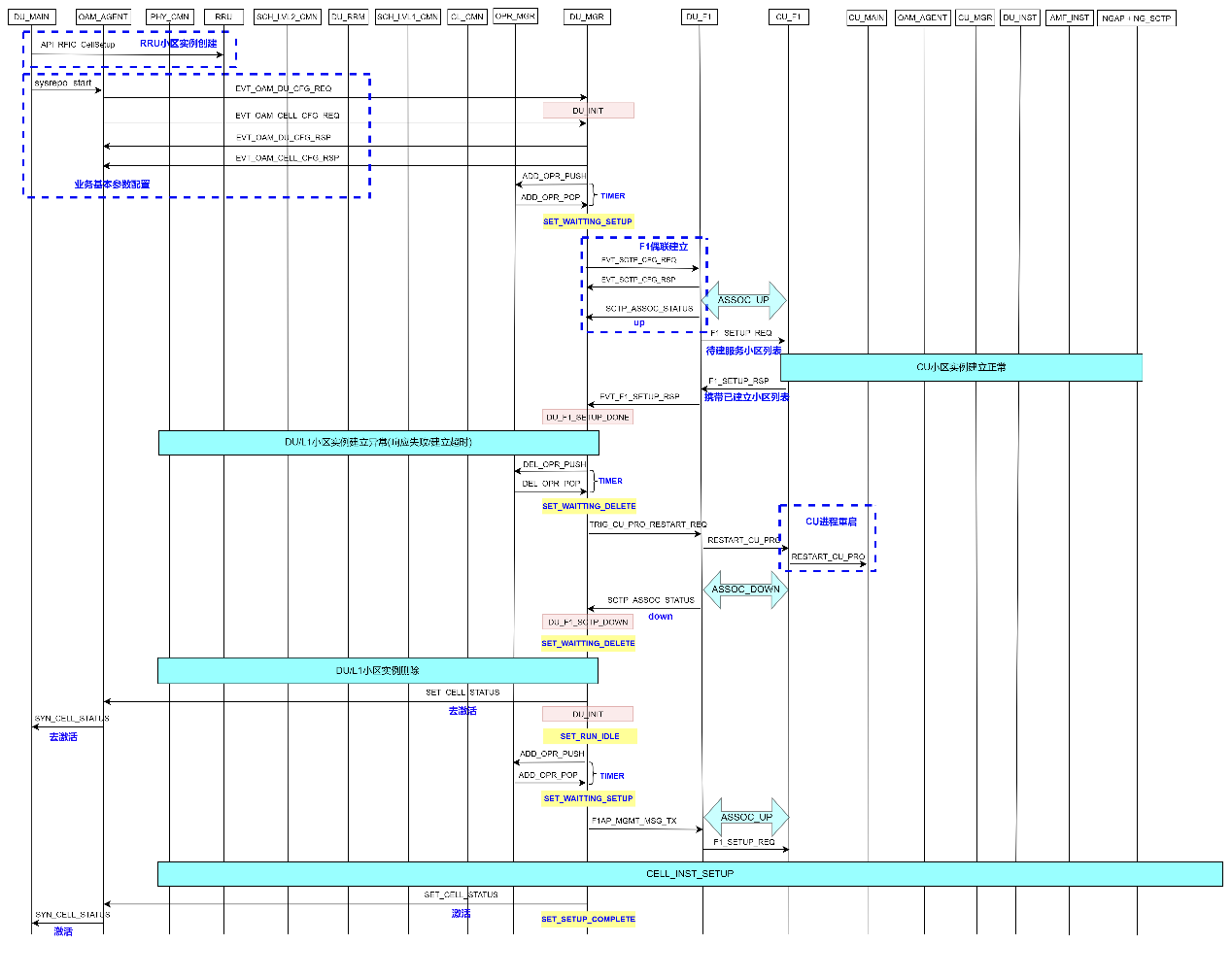
13、CU依据F1口携带的待建小区列表创建CU小区实例。

14、NG偶联建立是小区建立的先决条件，且后续的业务依赖于NG链路。

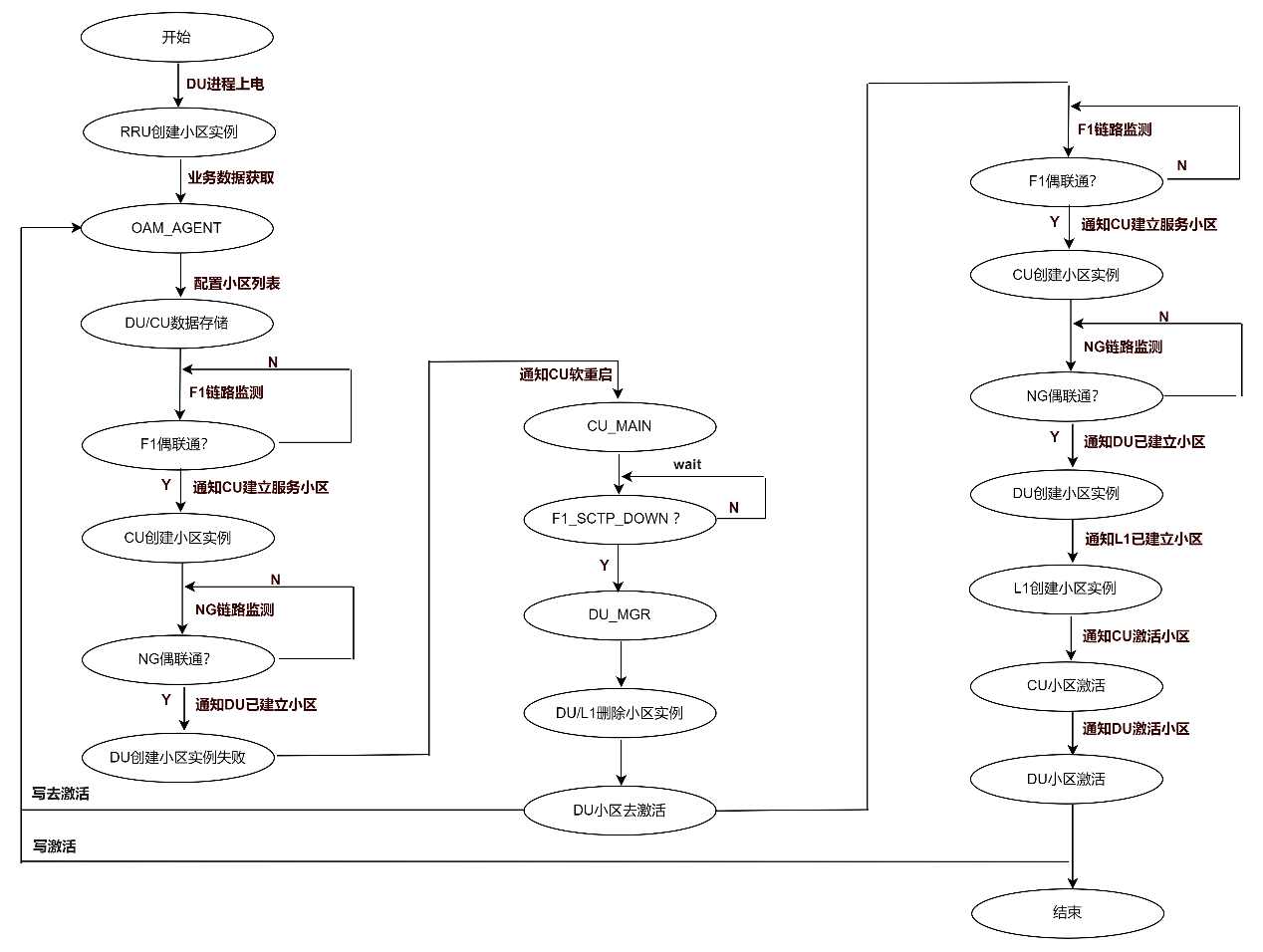
15、DU依据F1口携带的CU已建小区列表创建DU小区实例。

16、CL\_CMN模块发送FAPI接口消息通知PHY创建小区实例。

17、小区激活，状态同步至网管，恢复告警。



**图8-22 DU小区建立异常消息交互时序图**



**图8-23 DU小区建立异常流程图**

二、CU小区建立异常基本流程按照以下几个步骤进行：

1、初始上电，OAM\_AGENT从YANG模型配置获取小区数据并同步至业务数据区，触发小区建立流程，建立原因为INIT\_ADD\_CELL。

2、RRU创建小区实例。

3、OPR\_ADD操作符入队列，周期定时器超时取出接纳，接纳成功后执行建立动作。

4、F1偶联建立过程是CU&DU小区建立信息同步的基础，且后续的业务依赖于F1链路。

5、CU依据F1口携带的待建小区列表创建CU小区实例时建立失败，上报告警。

6、CU小区实例删除采取进程复位策略，此时会导致F1偶联断。

7、小区去激活，状态同步至网管。

8、OPR\_ADD操作符入队列，建立原因为INIT\_ADD\_CELL，周期定时器超时取出接纳，接纳成功后执行建立动作。

9、F1偶联建立过程是CU&DU小区建立信息同步的基础，且后续的业务依赖于F1链路。

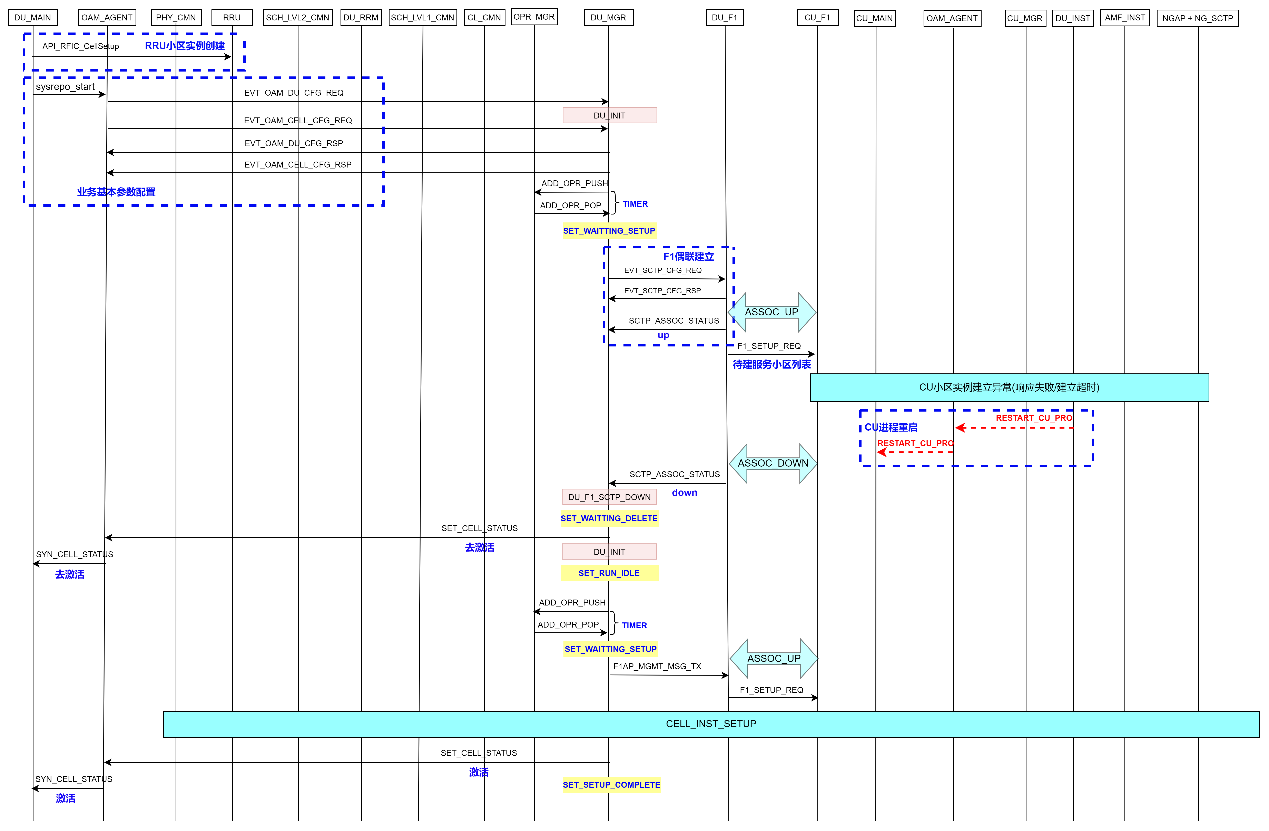
10、CU依据F1口携带的待建小区列表创建CU小区实例。

11、NG偶联建立是小区建立的先决条件，且后续的业务依赖于NG链路。

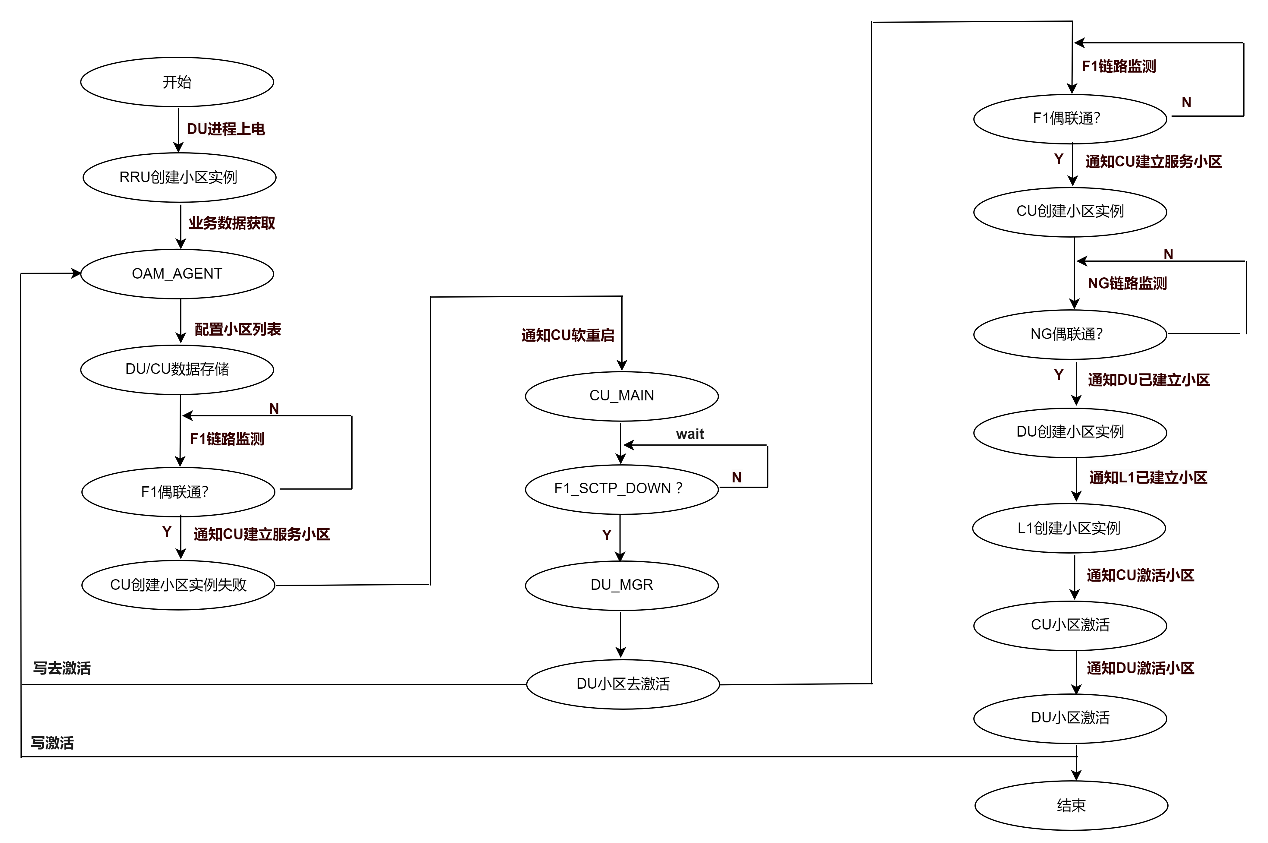
12、DU依据F1口携带的CU已建小区列表创建DU小区实例。

13、CL\_CMN模块发送FAPI接口消息通知PHY创建小区实例。

14、小区激活，状态同步至网管，恢复告警。



**图8-24 CU小区建立异常消息交互时序图**



**图8-25 CU小区建立异常流程图**

#### 小区删除异常

小区删除异常基本原则：删除某个模块小区实例失败时，加统计记录失败信息后继续删除下一个模块小区实例。总之，删除时各模块必须给出反馈，无论反馈结果成功与否，高层都将继续执行删除流程。

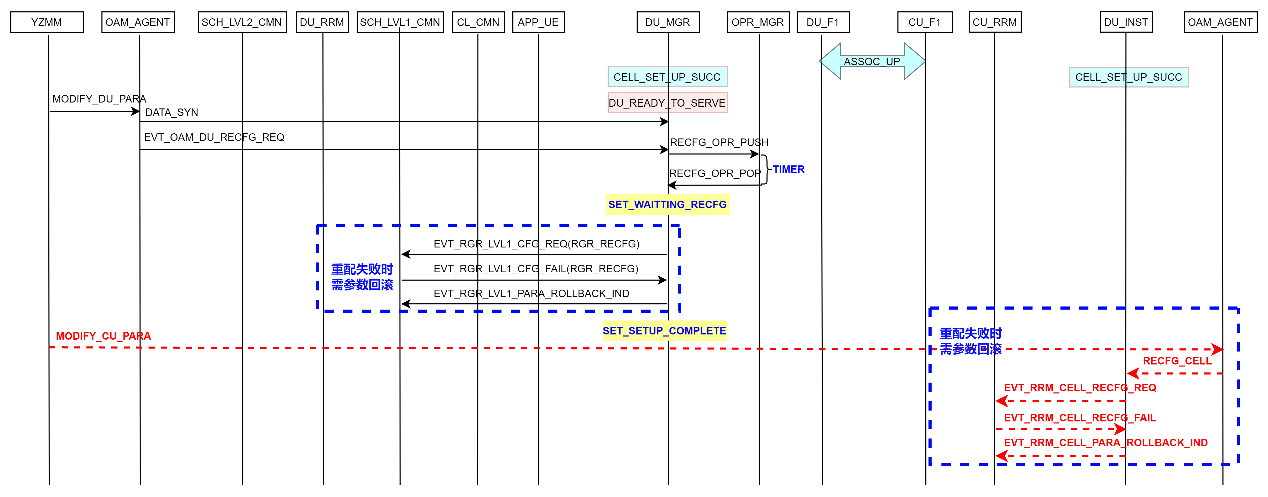
#### 小区重配异常(选做)

小区重配基本流程按照以下几个步骤进行：

1、网管修改小区重配参数，新参数入库并同步至业务数据区，同时触发小区重配流程，重配消息指定重配原因(重配模块+功能对象)。

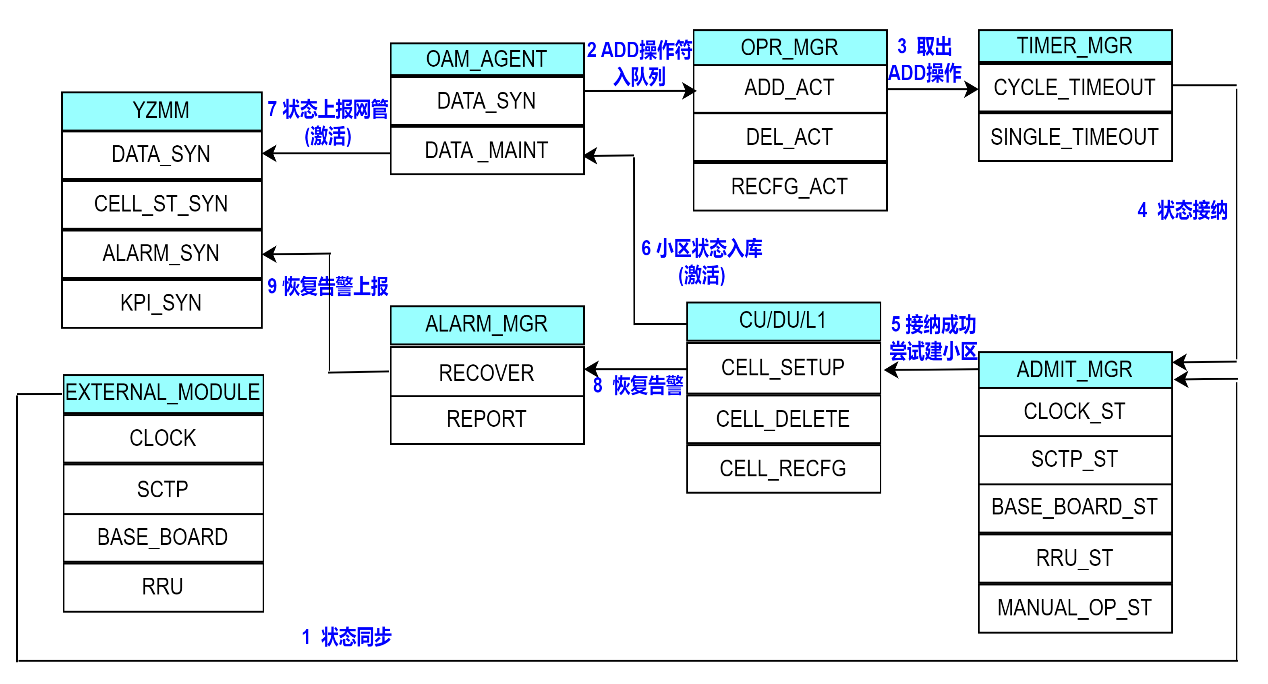
2、OPR\_RECFG操作符入队列，周期定时器超时取出接纳，接纳成功后执行重配动作。

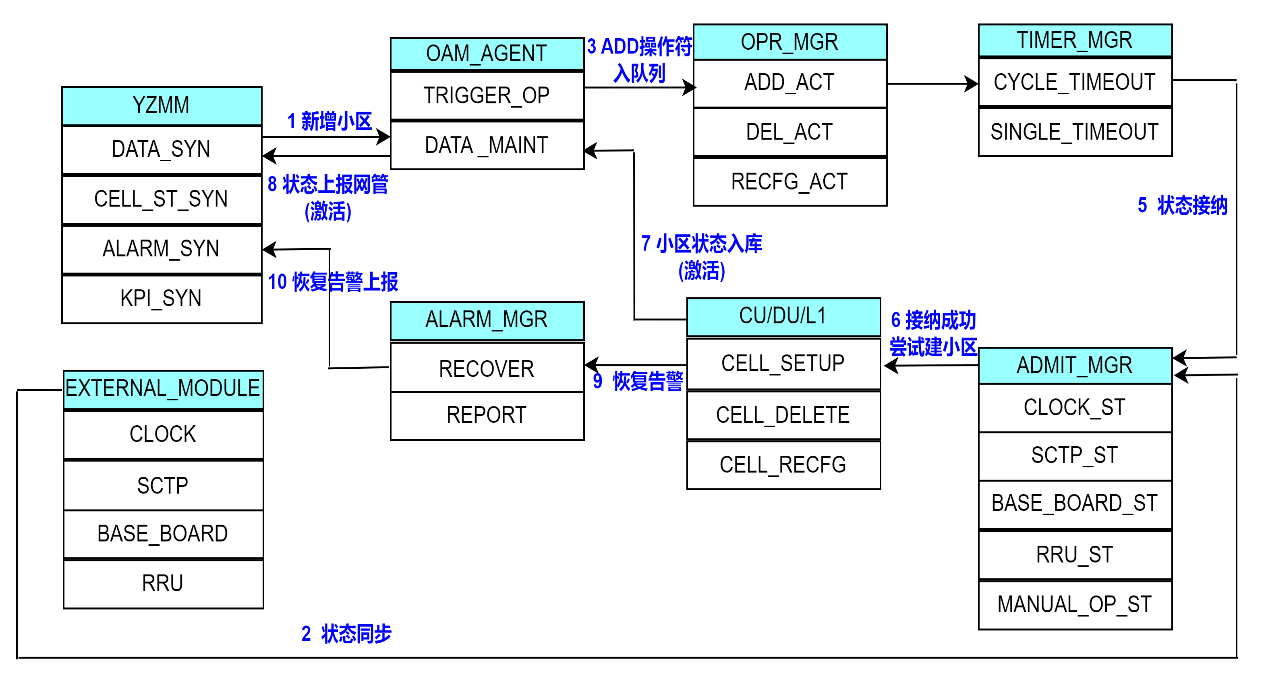
3、依据重配原因，进行相应模块和功能参数的配置更新。如果配置更新失败，需要通知相应的模块进行参数回滚，依旧生效旧值。



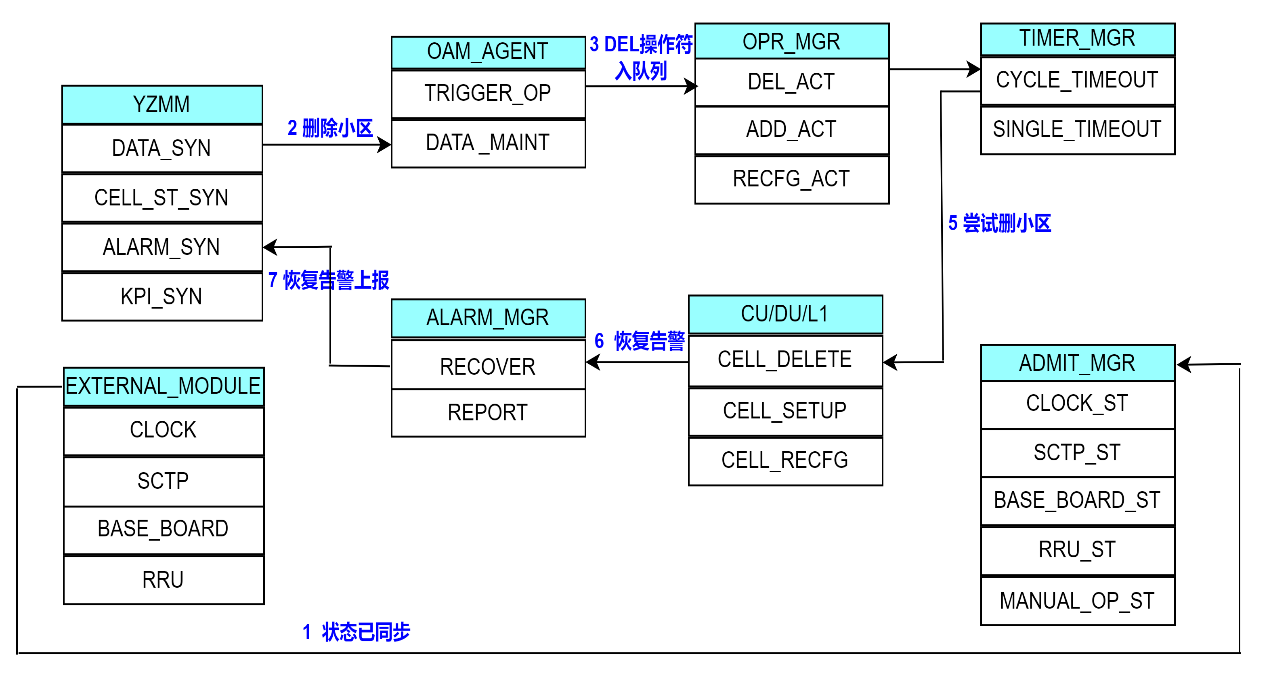
**图8-26小区重配异常流程图**

## 信号流图

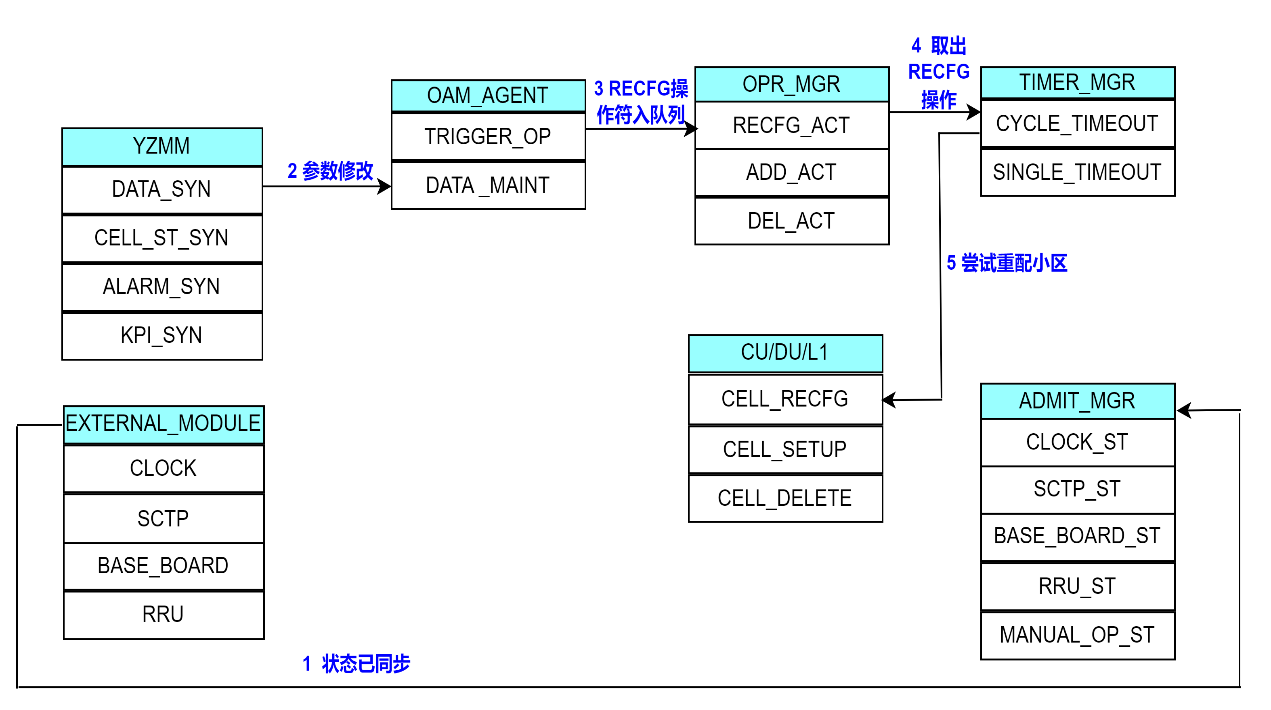
**图9-1 初始上电小区建立信号流图**



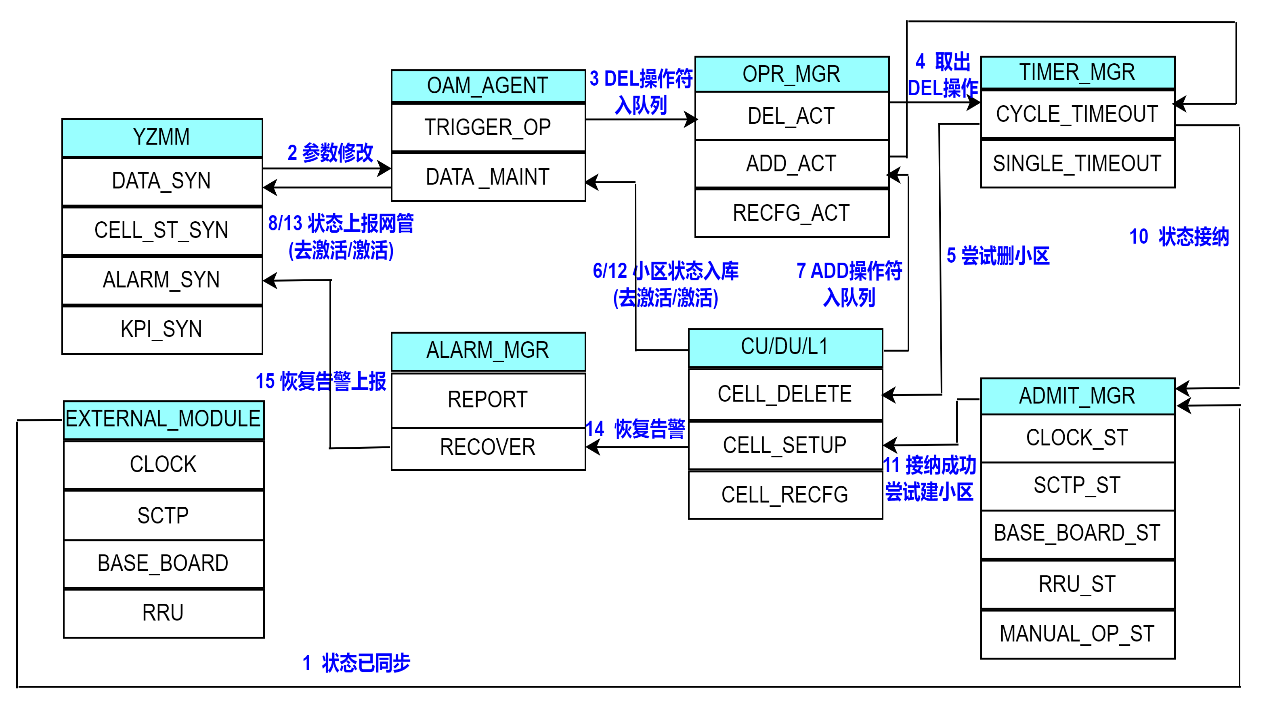
**图9-2 小区新增信号流图**



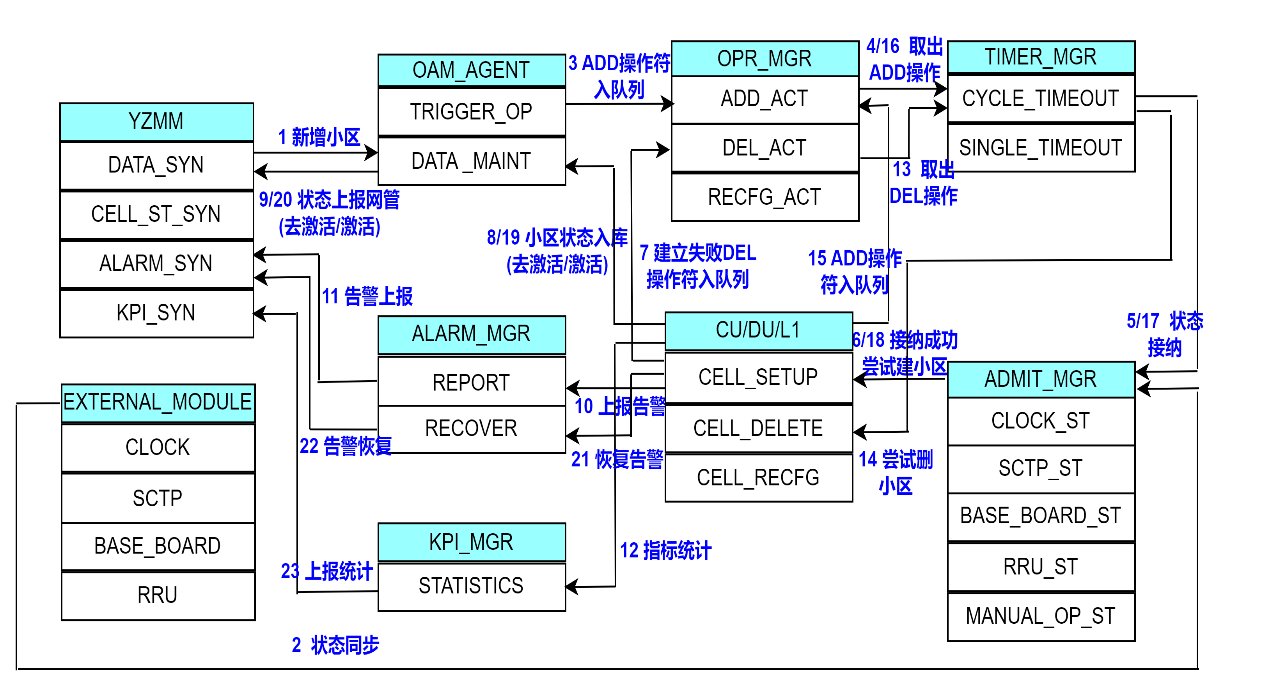
**图9-3 小区删除信号流图**



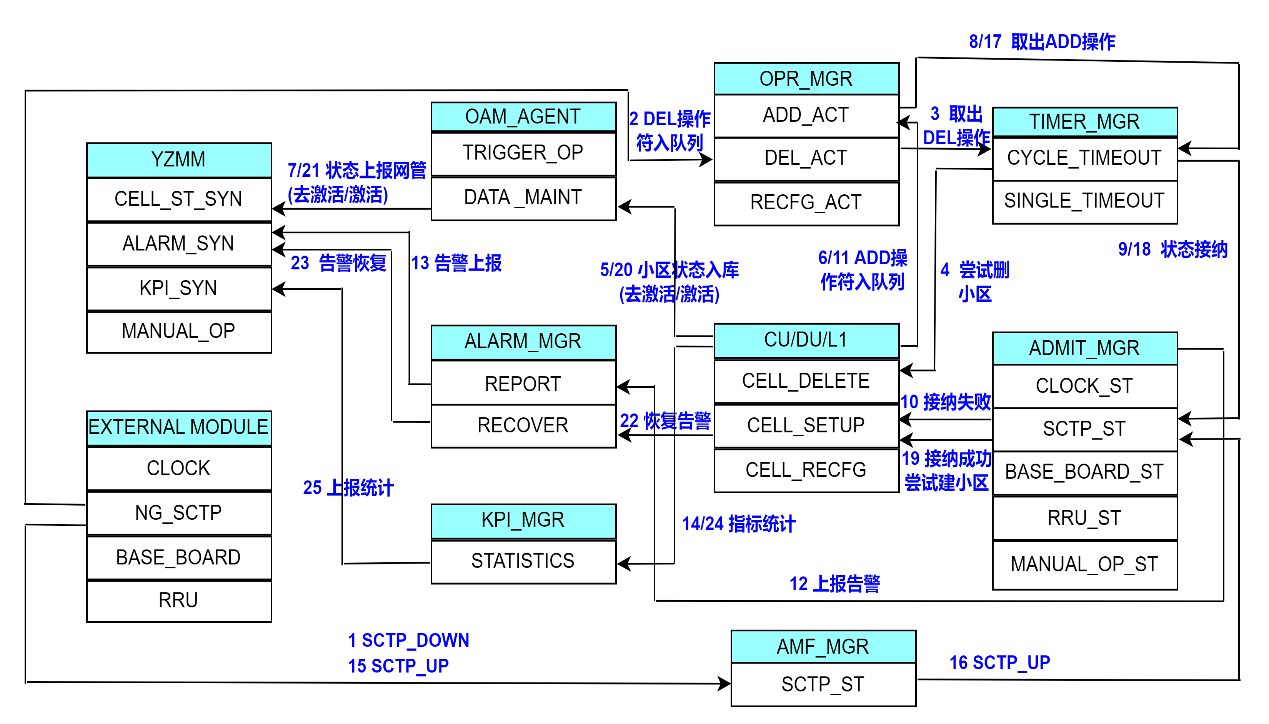
**图9-4 小区重配信号流图**



**图9-5 小区复位信号流图**

****

**图9-6 小区建立异常信号流图**



**图9-7 NG链路状态接纳信号流图**

## 附录