

概述

本手册使用范围：

本手册适用范围	
SPC1125 系列	SPC1125, SPC1128
SPC1168 系列	SPC1155, SPC1156, SPC1158, SPC1168, SPD1148, SPD1178, SPD1188, SPD1163, SPM1173
SPC2168 系列	SPC2168, SPC2166, SPC2165, SPC1198
SPC1169 系列	SPC1169, SPD1179, SPD1176
SPC2188 系列	SPC2188, SPC1185

目录

1	SPC1125 系列	8
1.1	特色	8
1.2	功能描述	8
1.3	功能实例	9
1.3.1	实例 1: ADC 单端采样	9
1.3.2	实例 2: ADC 双端采样	10
1.3.3	实例 3: ADC 开路检测	11
1.3.4	实例 4: ADC 短路检测	13
2	SPC1168 系列/SPC2168 系列	15
2.1	特色	15
2.2	功能描述	15
2.3	功能实例	16
2.3.1	实例 1: ADC 单端采样	16
2.3.2	实例 2: ADC 双端采样	17
2.3.3	实例 3: ADC 后处理单元	18
2.3.4	实例 4: ADC 开路检测	19
2.3.5	实例 5: ADC 短路检测	22
3	SPC1169 系列	23
3.1	特色	23
3.2	功能描述	23
3.3	功能实例	24
3.3.1	实例 1: ADC 单端采样	24
3.3.2	实例 2: ADC 双端采样	25
3.3.3	实例 3: ADC 后处理单元	26
3.3.4	实例 4: ADC 开路检测	27
3.3.5	实例 5: ADC 短路检测	29
4	SPC2188 系列	31
4.1	特色	31
4.2	功能描述	32
4.3	功能实例	33
4.3.1	实例 1: ADC 单端采样	33
4.3.2	实例 2: ADC 双端采样	34
4.3.3	实例 3: ADC 后处理单元	35

4.3.4	实例 4: ADC 开路检测	36
4.3.5	实例 5: ADC 短路检测	38
5	常见问题 QA	39
5.1	ADC 转换结果为负数	39

SPIN TROL

图片列表

图 1-1: ADC 功能分区框图	9
图 1-2: 单端采样信号流	10
图 1-3: 双端采样信号流	11
图 1-4: 检测 ADC 输入端引脚是否浮空 (预放电)	12
图 1-5: 检测 ADC 输入端引脚是否浮空 (预充电)	13
图 2-1: ADC 功能分区框图	16
图 2-2: 单端采样信号流	17
图 2-3: 双端采样信号流	18
图 2-4: PPU 结构图.....	19
图 2-5: 检测 ADC 输入端引脚是否浮空 (预放电)	20
图 2-6: 检测 ADC 输入端引脚是否浮空 (预充电)	21
图 3-1: ADC 功能分区框图	24
图 3-2: 单端采样信号流	25
图 3-3: 双端采样信号流	26
图 3-4: PPU 结构图.....	27
图 3-5: 检测 ADC 输入端引脚是否浮空 (预放电)	28
图 3-6: 检测 ADC 输入端引脚是否浮空 (预充电)	29
图 4-1: ADC 功能分区框图	32
图 4-2: 单端采样信号流	33
图 4-3: 双端采样信号流	34
图 4-4: PPU 结构图.....	35
图 4-5: 检测 ADC 输入端引脚是否浮空 (预放电)	36
图 4-6: 检测 ADC 输入端引脚是否浮空 (预充电)	37

表格列表

表 1-1: ADC 数据信息	8
表 1-2: 实例 1 代码路径	10
表 1-3: 实例 2 代码路径	11
表 1-4: 实例 3 代码路径	13
表 1-5: 实例 4 代码路径	14
表 2-1: ADC 配置	15
表 2-2: 实例 1 代码路径	17
表 2-3: 实例 2 代码路径	18
表 2-4: 实例 3 代码路径	19
表 2-5: 实例 4 代码路径	21
表 2-6: 实例 5 代码路径	22
表 3-1: ADC 配置	23
表 3-2: 实例 1 代码路径	25
表 3-3: 实例 2 代码路径	26
表 3-4: 实例 3 代码路径	27
表 3-5: 实例 4 代码路径	29
表 3-6: 实例 5 代码路径	30
表 4-1: ADC 配置	32
表 4-2: 实例 1 代码路径	33
表 4-3: 实例 2 代码路径	34
表 4-4: 实例 3 代码路径	35
表 4-5: 实例 4 代码路径	37
表 4-6: 实例 5 代码路径	38

版本历史

版本	日期	作者	状态	变更
A/0	2023-11-20	X.He	Outdated	1. 首次发布。
C/0	2024-08-09	LemengZhou	Released	1. 修改为全系列通用文档。

SPIN TROL

术语或缩写

术语或缩写	描述
共模电压	差分电路两个输入端电压的平均值，也称共模信号
SAR ADC	逐次逼近型 ADC

SPIN
TROL

1 SPC1125 系列

1.1 特色

ADC 模块主要包含以下功能：

- 13 位分辨率；
- 最小 250 ns 转换时间；
- 2 个差分采样保持电路；
- 模拟信号输入范围：0~3.339V；
- 内置 1.2V 参考电压；
- 输入可来自于外部 IO 输入、温度传感器、内部电源、可编程增益放大器输出、D2SBUF 的输出；
- 支持外部输入开路 and 短路检测；
- 数字逻辑的时钟使能控制独立可配，最高频率同 CPU 频率；
- 8 个转换控制通道 CH，触发源、输入信号、采样和转换时间、自动平均样本数均可独立配置，转换完成后产生独立的 EOC 事件；
- 支持转换优先级控制；
- 支持如下事件触发采样转换请求
 - 软件触发；
 - EOC 事件触发；
 - PWM REQ 信号；
 - TIMER REQ 信号；
 - 外部引脚请求。

1.2 功能描述

在电力电子系统中，通常使用 ADC 进行电压采样，将模拟电压转换为数字电压。ADC 的精度、参考电压、采样时间、转换时间和保持器数量，如表 1-1 示：

表 1-1: ADC 数据信息

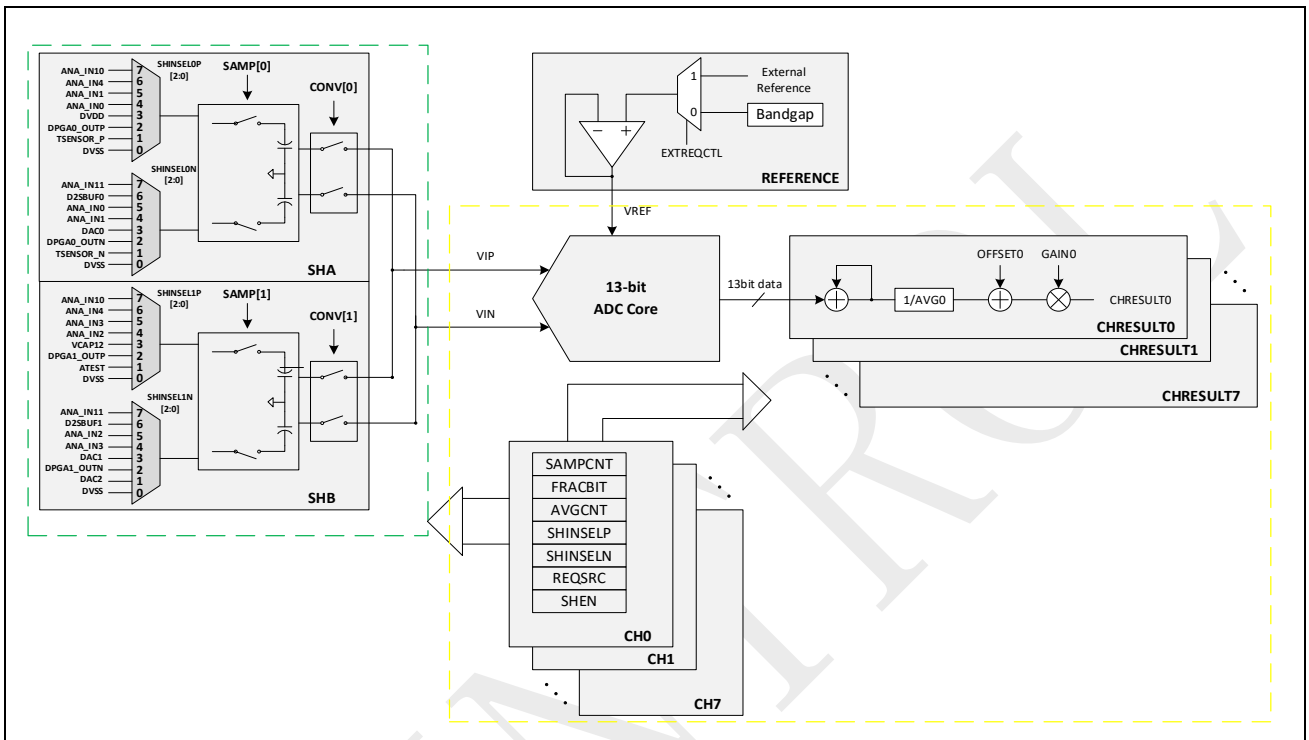
ADC 位数	参考电压值	采样经典时间	转换时间	保持器数量
13	3.339V	200ns	250ns	2

在实际使用 ADC 过程中，可以按照功能区块来帮助理解怎样使用 ADC，ADC 大体上可以按照图 1-1 中的不同颜色虚线分成 3 个功能区块，在使用过程中按照如下步骤进行配置：

- 首先绿色虚线框内只需要使能采样及转换开关，这个步骤可以等其它配置都设置完成之后再行；
- 然后设置黄色虚线框部分，在这部分中，有 8 路独立的 Channel 采样转换配置可以预先设置，每个 Channel 均可根据不同的目的而配置不同的参数，例如：采样时间（具体需

要设置多长的采样时间，请参考《ADC 建立时间计算方法使用指南》，ADC core 转换时间，是否需要多次采样且取平均值，ADC 输入的正负端，触发源选择等；在设置完这些信息之后，对应 Channel 的设置产生的 ADC 转换结果将存储在对应的 CHRESULTSx (x=0, 1, ..., 7) 中。

图 1-1: ADC 功能分区框图



1.3 功能实例

1.3.1 实例 1: ADC 单端采样

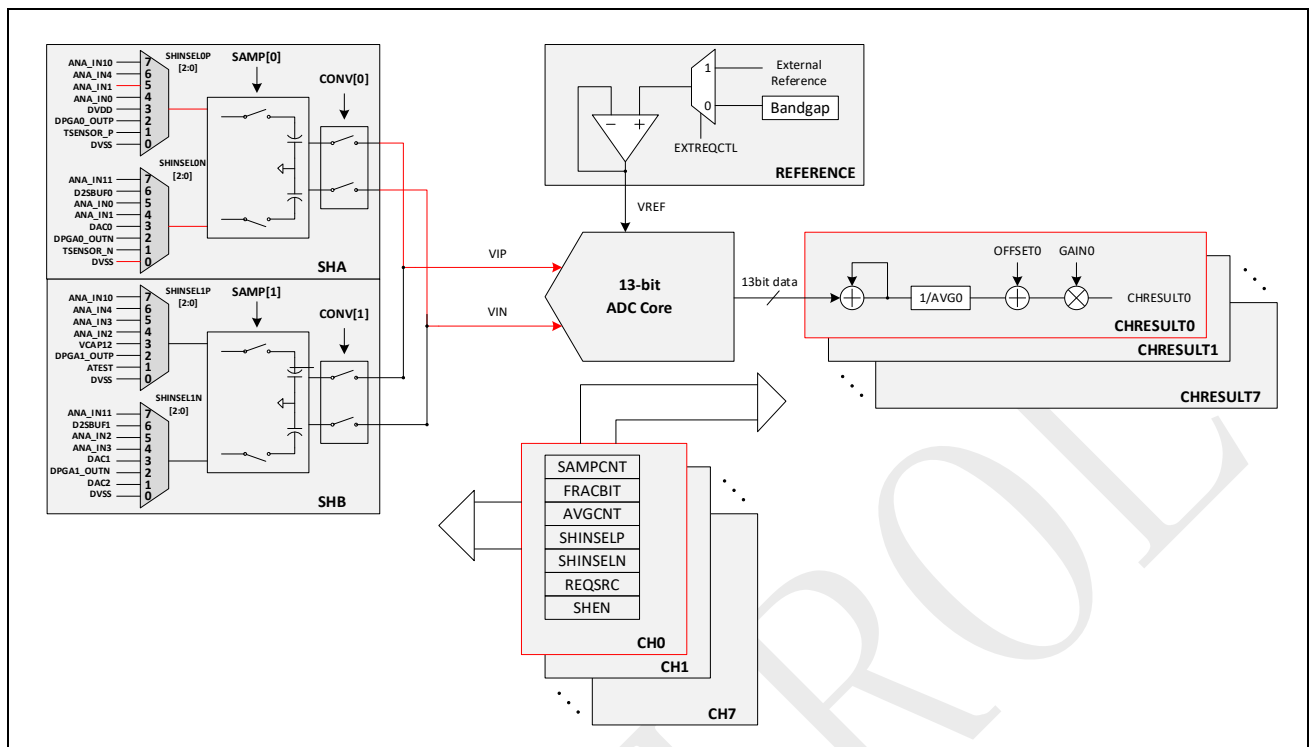
1.3.1.1 功能需求

利用 ADC 进行单端（也即有一端接 GND）采样，其中 SHINSELP 正端输入选择和 SHINSELN 负端输入其中一端接 GND，另一端接非 GND。

1.3.1.2 功能实现

1. 确定单端采样信号流如图 1-2，一端连接 ANA_IN1，另一端接地。

图 1-2: 单端采样信号流



2. 初始化 ADC 时钟。
3. ADC 通道选择, 输入信号的选择, 触发源, 采样时间, 转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo, [表 1-2](#):

表 1-2: 实例 1 代码路径

MCU 产品型号	代码路径
SPC1125 系列	SDK 目录\0_Examples\ADC_Single_End_Abs_Result

1.3.2 实例 2: ADC 双端采样

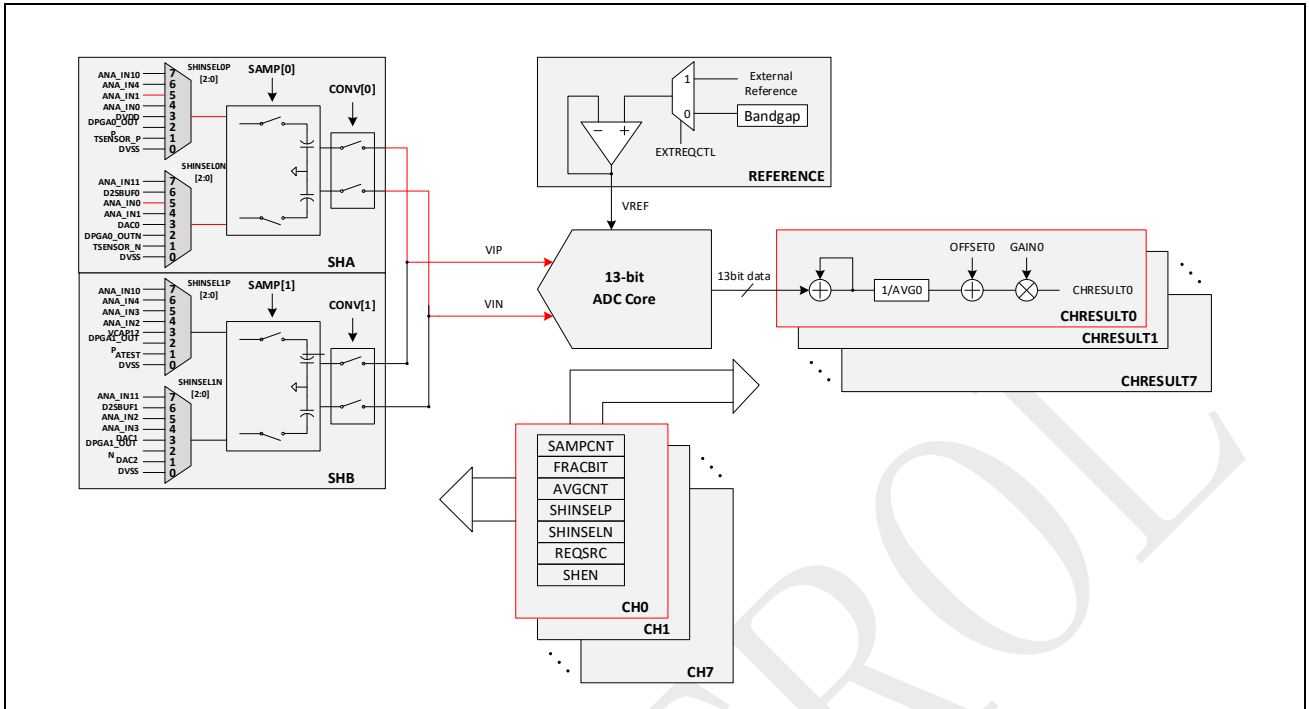
1.3.2.1 功能需求

利用 ADC 进行双端采样, 其中 SHINSELP 正端输入选择和 SHINSELN 负端输入选择两个非 GND。

1.3.2.2 功能实现

1. 确定单端采样信号流如图 1-3: 双端采样信号流, 一端连接 ANA_IN1, 另一端接 ANA_IN0。

图 1-3: 双端采样信号流



2. 初始化 ADC 时钟。
3. ADC 通道选择, 输入信号的选择, 触发源, 采样时间, 转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo, 如表 1-3:

表 1-3: 实例 2 代码路径

MCU 产品型号	代码路径
SPC1125 系列	SDK 目录\0_Examples\ADC_Differential_Abs_Result

1.3.3 实例 3: ADC 开路检测

1.3.3.1 功能需求

使用 ADC 单元支持的两种开路检测方式进行开路检测:

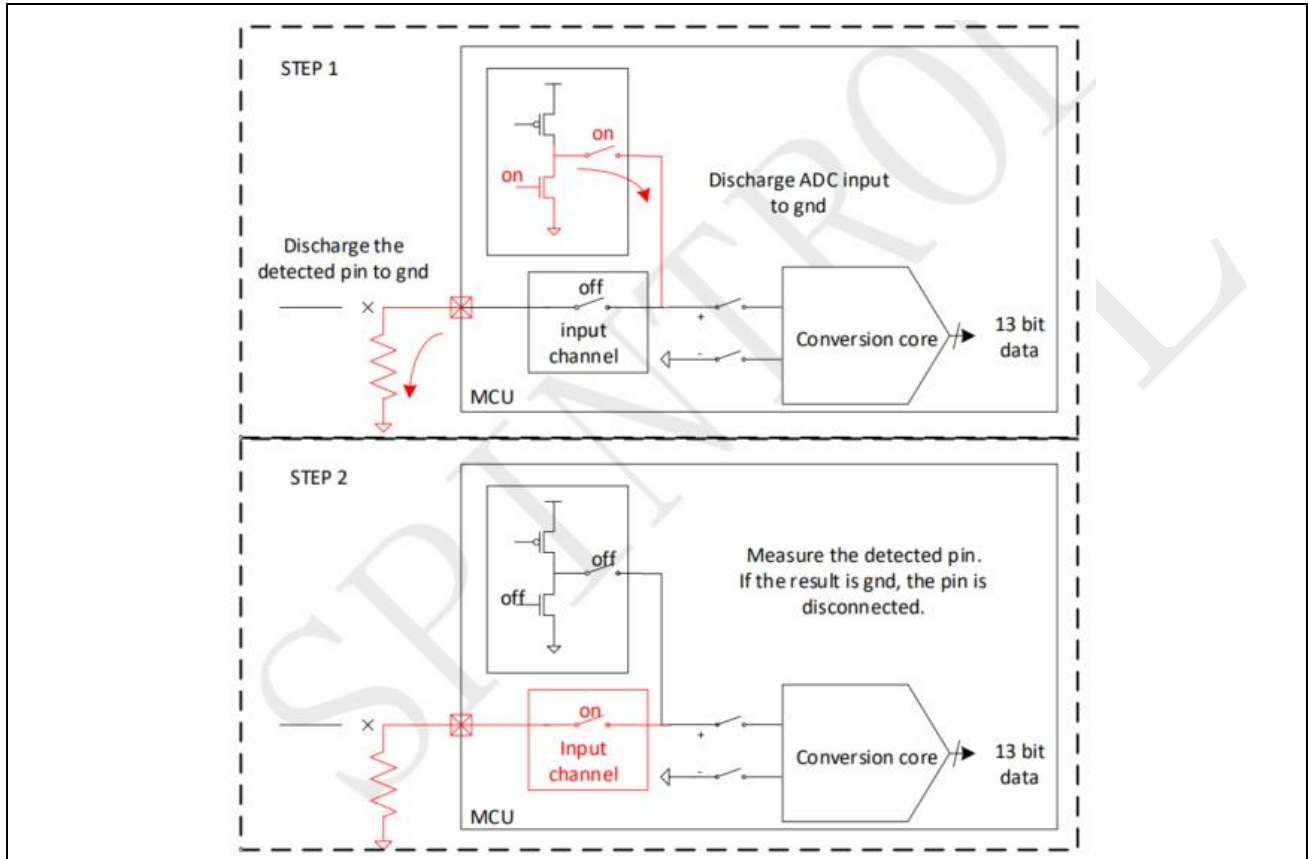
- 使用预放电故障注入法, 将节点放电到地的方式进行开路检测;
- 使用预充电故障注入法, 将节点充电到电源的方式进行开路检测;

1.3.3.2 功能实现

1. 预放电故障注入法检测如图 1-4, 即检测连接到采样器正输入端的引脚是否浮动。
 - a) 步骤 1: 将 PIN 脚切成 GPIO 功能, 设置为输入, 并使能 GPIO 下拉, 释放 IO 上残存的电量; 将 ADC 采样保持的正端强制接地, 释放通路上的残存电量;

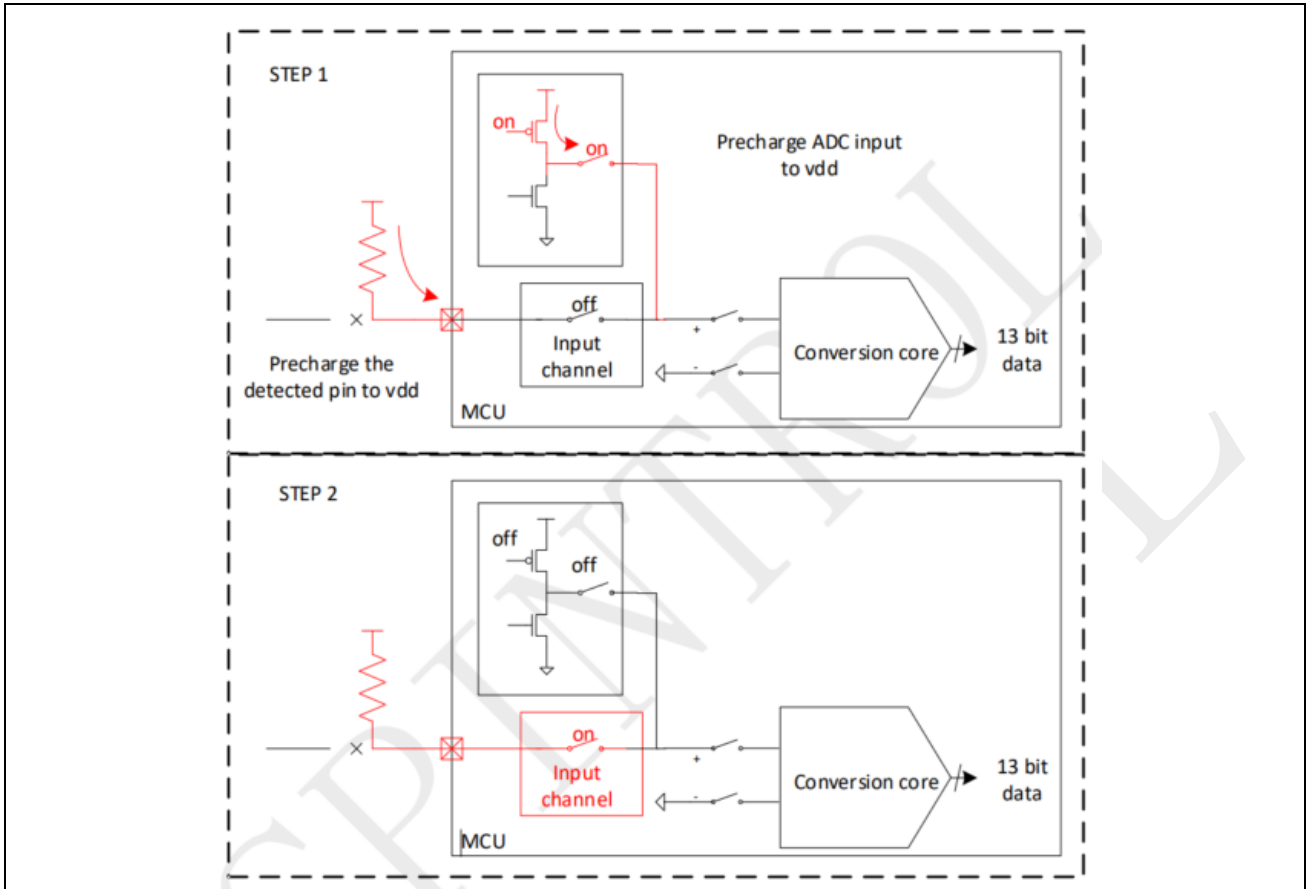
- b) 步骤 2: 将 ADC 采样保持的正端恢复断开状态, 将 PIN 脚切成 ADC 功能, 降低 ADC 模块的时钟频率, 设置较长采样时间, 以期获得更为准确的测试数据, 随后开始测量, 获得电压数值 Result1;

图 1-4: 检测 ADC 输入端引脚是否浮空 (预放电)



2. 预充电故障注入法检测图 1-5, 即检测连接到采样器正输入端的引脚是否悬空。
- 将 PIN 脚切成 GPIO 功能, 设置为输入, 并使能 GPIO 上拉, 给 IO 的通路进行预充电; 将 ADC 采样保持的正端强制拉高, 对采样保持器通路进行预充电;
 - 将 ADC 采样保持的正端恢复断开状态, 将 PIN 脚切成 ADC 功能, 降低 ADC 模块的时钟频率, 设置较长采样时间, 以期获得更为准确的测试数据, 随后开始测量, 获得电压数值 Result2;

图 1-5: 检测 ADC 输入端引脚是否浮空 (预充电)



3. 结果判断，Result1 在 0V 左右，且 Result2 在 3.3V 左右，则表明采样器的输入未链接外部输入，即正输入端处于漂浮状态。类似的步骤也可以用来检测采样器的负输入引脚是否悬空。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 1-4:

表 1-4: 实例 3 代码路径

MCU 产品型号	代码路径
SPC1125 系列	SDK 目录\0_Examples\ ADC_Open_Detect

1.3.4 实例 4: ADC 短路检测

1.3.4.1 功能需求

模拟输入可能存在短路情况，可以使用预充电或放电电路进行故障注入来检测。

1.3.4.2 功能实现

1. 输入短路检测功能可以用 ADCCTL 寄存器设置。检测可以分为下面几步：
 - a) ADC 测量输入 PIN 脚电压，设为结果 DATA1。

- b) 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱下拉 GND，ADC 测量输入 PIN 脚电压，设为结果 DATA2。
 - c) 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱上拉 VDD，ADC 测量输入 PIN 脚电压，设为结果 DATA3。
 - d) 如果 $DATA1 \sim = DATA2 \sim = DATA3$ ，那么输入 PIN 脚短路。
2. 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 1-5：

表 1-5：实例 4 代码路径

MCU 产品型号	代码路径
SPC1125 系列	SDK 目录\0_Examples\ ADC_Short_Detect



2 SPC1168 系列/SPC2168 系列

2.1 特色

ADC 模块主要包含以下功能：

- 14 位分辨率转换内核以及三个采样保持电路（Sample-and-hold circuit, SH）；
- 支持高达 4 兆次/秒采样率；
- 支持多达 20 个模拟输入通道的采样；
- 支持 3 个采样保持电路同时采样；
- 模拟满幅度输入范围：3.657V（内部基准源），或者和外部输入基准源成比例（外灌输入管脚为 GPIO12）；
- 16 套 SOC 配置，每套配置包含触发源，采样模式，通道配置，平均次数，采样时间；
- 16 个结果寄存器（独立寻址）用于储存转换结果；
- 多种触发源可供选择；
- 软件触发：
 - 通用定时器 Timer0/1/2 触发；
 - GPIO 外部触发；
 - PWMxSOCA、PWMxSOCB 和 PWMxSOCC（x= 0 ~ 7）触发，时序可配；
- 16 个独立的触发信号可以触发 16 个 NVIC 中断；

2.2 功能描述

在电力电子系统中，通常使用 ADC 进行电压采样，将模拟电压转换为数字电压。ADC 的精度、参考电压、采样时间、转换时间和保持器数量，如表 2-1 所示：

表 2-1: ADC 配置

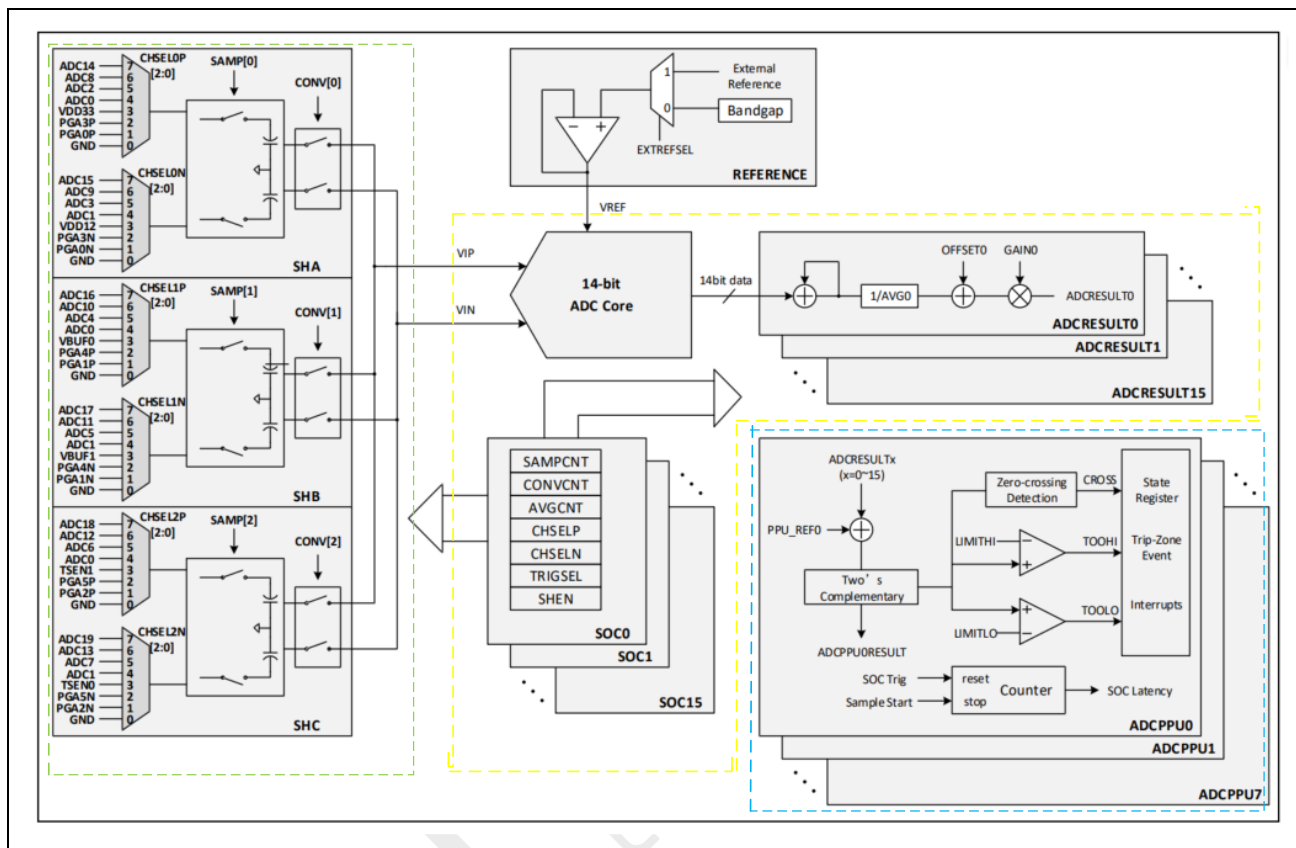
ADC 位数	参考电压值	采样经典时间	转换时间	保持器数量
14	3.65V	140ns	140ns	3

在实际使用 ADC 过程中，可以按照功能区块来帮助理解怎样使用 ADC，ADC 大体上可以按照图 2-1 中的不同颜色虚线分成 3 个功能区块，在使用过程中按照如下步骤进行配置：

- 首先绿色虚线框内只需要使能采样及转换开关，这个步骤可以等其它配置都设置完成之后再行进行；
- 然后设置黄色虚线框部分，在这部分中，有 16 路独立的 Channel 采样转换配置可以预先设置，每个 Channel 均可根据不同的目的而配置不同的参数，例如：采样时间（具体需要设置多长的采样时间，请参考《ADC 建立时间计算方法使用指南》），ADC core 转换时间，是否需要多次采样且取平均值，ADC 输入的正负端，触发源选择等；在设置完这些信息之后，对应 Channel 的设置产生的 ADC 转换结果将存储在对应的 ADCRESULTSx（x=0,1, ..., 15）中。

- 最后，如果有需求，可以继续设置蓝色虚线框部分，此部分即为 ADC PPU 单元。通常工程使用中，需要知道 ADC 转换的结果相对某个参考值而言是过高还是过低，这部分的工作就可以交给 PPU 单元处理，以减少 CPU 的计算压力。除此之外，PPU 单元还可以检测出从发起 ADC 转换请求至开始 ADC 转换时的时间计数。

图 2-1: ADC 功能分区框图



2.3 功能实例

2.3.1 实例 1: ADC 单端采样

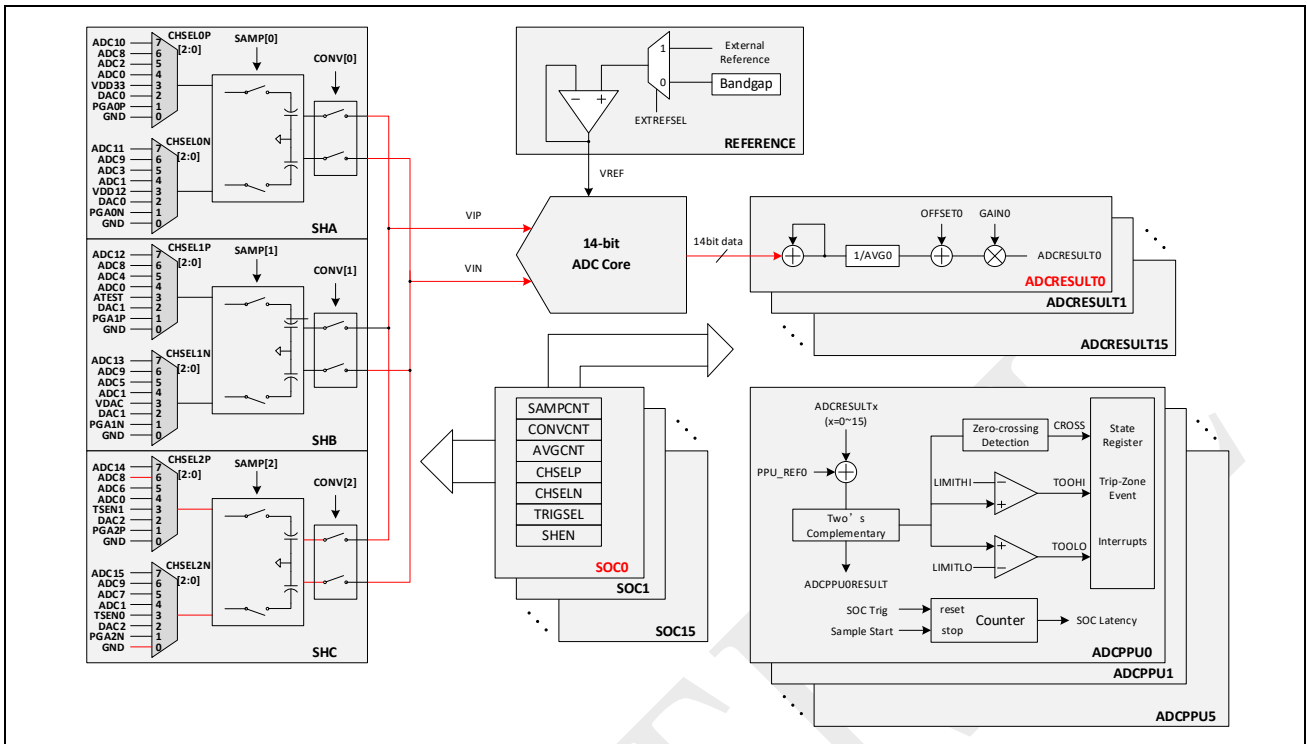
2.3.1.1 功能需求

利用 ADC 进行单端（也即有一端接 GND）采样，其中 SHINSELP 正端输入选择和 SHINSELN 负端输入其中一端接 GND，另一端接非 GND。

2.3.1.2 功能实现

1. 确定单端采样信号流如图 2-2，一端连接 ANA_IN1，另一端接地。

图 2-2: 单端采样信号流



1. 初始化 ADC 时钟。
2. ADC 通道选择, 输入信号的选择, 触发源, 采样时间, 转换时间的相关设置。
3. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo, 如表 2-2:

表 2-2: 实例 1 代码路径

MCU 产品型号	代码路径
SPC1168 系列、SPC2168 系列	SDK 目录\0_Examples\ ADC_Single_End_Result

2.3.2 实例 2: ADC 双端采样

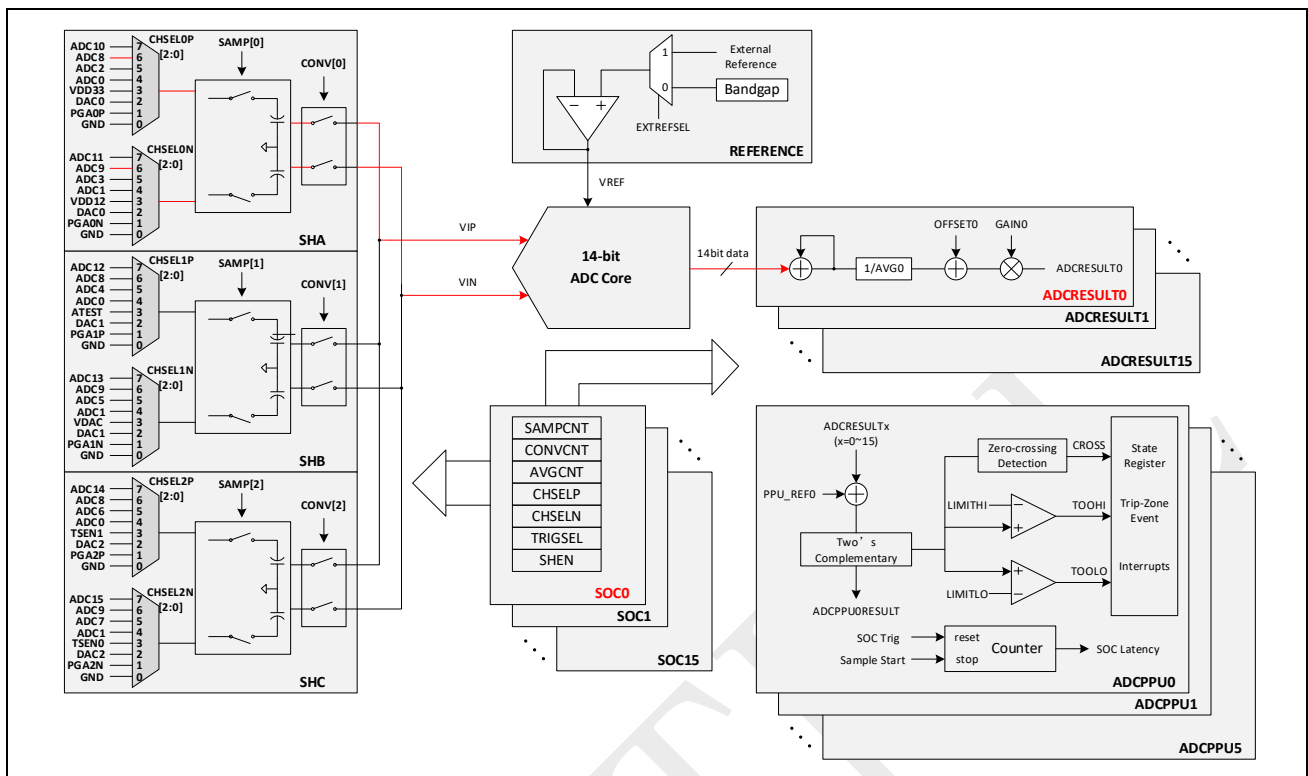
2.3.2.1 功能需求

利用 ADC 进行双端采样, 其中 SHINSELP 正端输入选择和 SHINSELN 负端输入选择两个非 GND。

2.3.2.2 功能实现

1. 确定单端采样信号流如图 2-3, 一端连接 ANA_IN1, 另一端接 ANA_IN0。

图 2-3: 双端采样信号流



2. 初始化 ADC 时钟。
3. ADC 通道选择, 输入信号的选择, 触发源, 采样时间, 转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo, 如表 2-3:

表 2-3: 实例 2 代码路径

MCU 产品型号	代码路径
SPC1168 系列、SPC2168 系列	SDK 目录\0_Examples\ADC_Differential_Trim_Result

2.3.3 实例 3: ADC 后处理单元

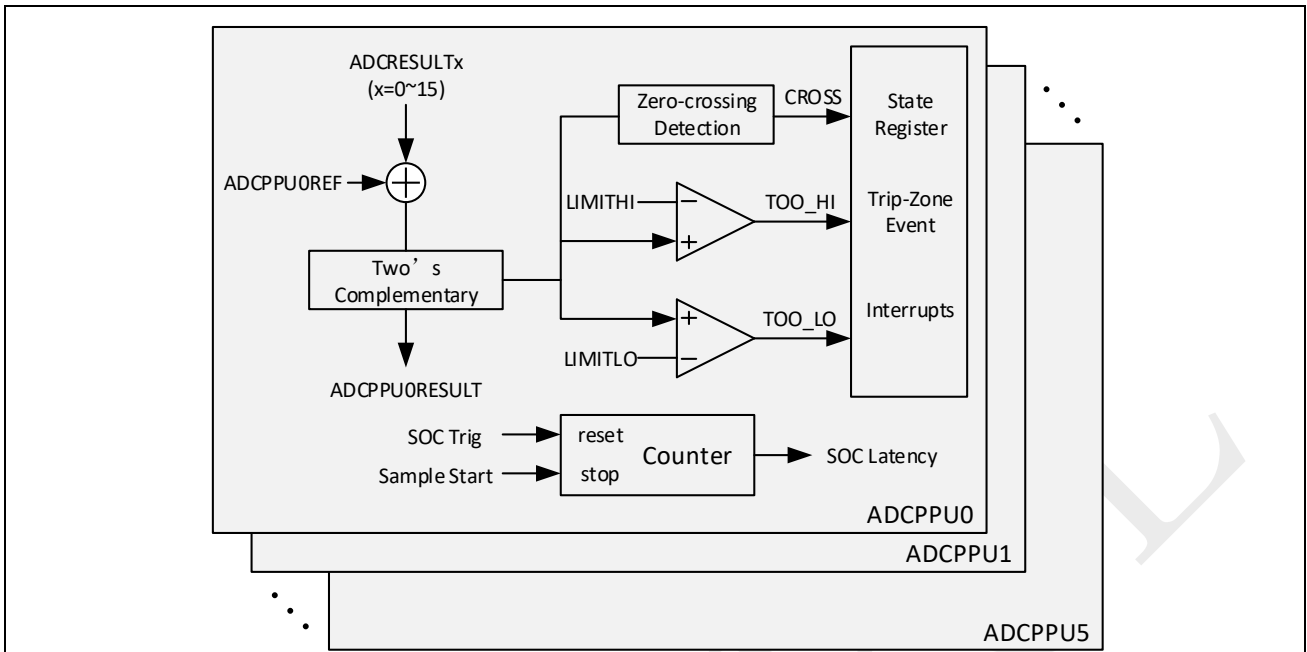
2.3.3.1 功能需求

使用 ADC 后处理单元, 检测 ADC 转换的结果相对某个参考值而言是过高还是过低。

2.3.3.2 功能实现

1. 确定 ADC 后处理单元 PPU 结构, 如图 2-4 所示:

图 2-4: PPU 结构图



2. 初始化 ADC 时钟。
 3. ADC 通道选择，输入信号的选择，触发源，采样时间，转换时间的相关设置。
 4. 将 GPIO 设置为模拟输入。
 5. 打开 PPU 过压、欠压、过参考电压的中断；
 6. 设置 PPU 过压以及欠压阈值；
 7. 选择 PPU 的输入，设置比较极性，设置参考电压值，随后使能 PPU；
- 以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 2-4：

表 2-4: 实例 3 代码路径

MCU 产品型号	代码路径
SPC1168 系列、SPC2168 系列	SDK 目录\0_Examples\ ADC_SHA_PPU

2.3.4 实例 4: ADC 开路检测

2.3.4.1 功能需求

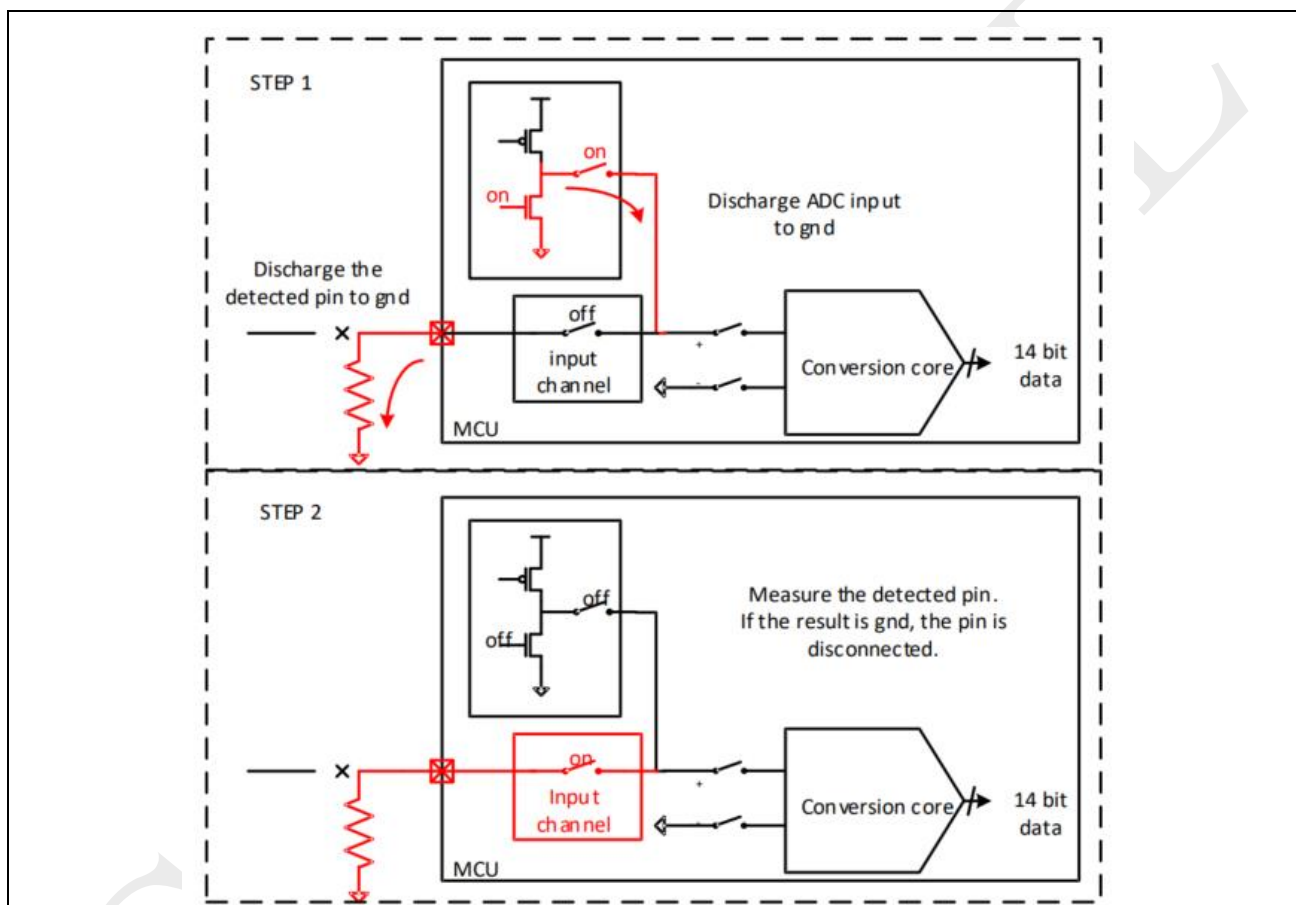
使用 ADC 单元支持的两种开路检测方式进行开路检测：

- 使用预放电故障注入法，将节点放电到地的方式进行开路检测；
- 使用预充电故障注入法，将节点充电到电源的方式进行开路检测；

2.3.4.2 功能实现

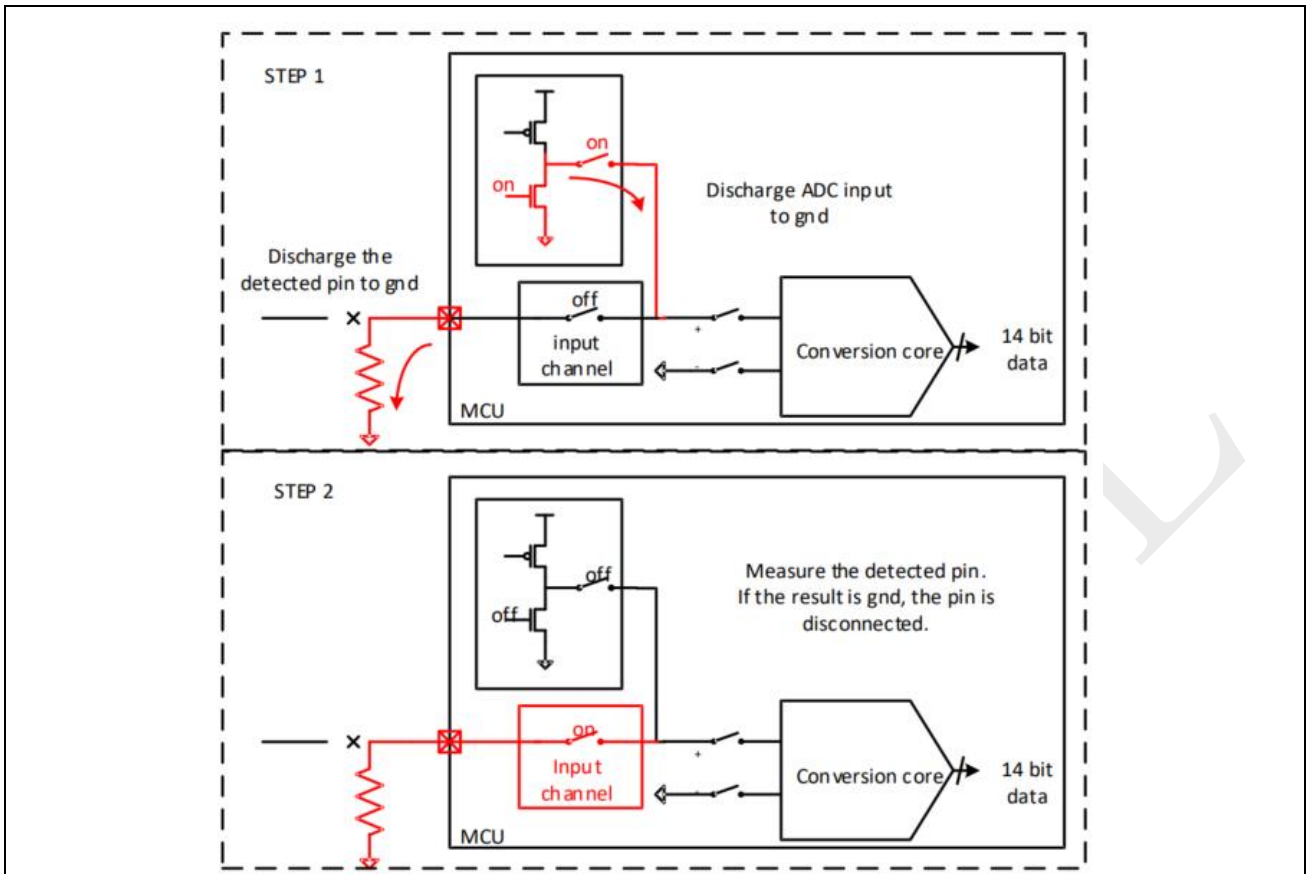
1. 预放电故障注入法检测如图 2-5，即检测连接到采样器正输入端的引脚是否浮动。
 - a) 步骤 1: 将 PIN 脚切成 GPIO 功能，设置为输入，并使能 GPIO 下拉，释放 IO 上残存的电量；将 ADC 采样保持的正端强制接地，释放通路上的残存电量；
 - b) 步骤 2: 将 ADC 采样保持的正端恢复断开状态，将 PIN 脚切成 ADC 功能，降低 ADC 模块的时钟频率，设置较长采样时间，以期获得更为准确的测试数据，随后开始测量，获得电压数值 Result1；

图 2-5: 检测 ADC 输入端引脚是否浮空（预放电）



2. 预充电故障注入法检测如图 2-6，即检测连接到采样器正输入端的引脚是否悬空。
 - a) 将 PIN 脚切成 GPIO 功能，设置为输入，并使能 GPIO 上拉，给 IO 的通路进行预充电；将 ADC 采样保持的正端强制拉高，对采样保持器通路进行预充电；
 - b) 将 ADC 采样保持的正端恢复断开状态，将 PIN 脚切成 ADC 功能，降低 ADC 模块的时钟频率，设置较长采样时间，以期获得更为准确的测试数据，随后开始测量，获得电压数值 Result2；

图 2-6: 检测 ADC 输入端引脚是否浮空 (预充电)



3. 结果判断，Result1 在 0V 左右，且 Result2 在 3.3V 左右，则表明采样器的输入未链接外部输入，即正输入端处于漂浮状态。类似的步骤也可以用来检测采样器的负输入引脚是否悬空。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 2-5:

表 2-5: 实例 4 代码路径

MCU 产品型号	代码路径
SPC1168 系列、SPC2168 系列	SDK 目录\0_Examples\ADC_SHA_Open_Detect

2.3.5 实例 5: ADC 短路检测

2.3.5.1 功能需求

模拟输入可能存在短路情况，可以使用预充电或放电电路进行故障注入来检测。

2.3.5.2 功能实现

1. 输入短路检测功能可以用 ADCCTL 寄存器设置。检测可以分为下面几步：
 - a) ADC 测量输入 PIN 脚电压，设为结果 DATA1。
 - b) 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱下拉 GND，ADC 测量输入 PIN 脚电压，设为结果 DATA2。
 - c) 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱上拉 VDD，ADC 测量输入 PIN 脚电压，设为结果 DATA3。
 - d) 如果 $DATA1 \neq DATA2 \neq DATA3$ ，那么输入 PIN 脚短路。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 2-6:

表 2-6: 实例 5 代码路径

MCU 产品型号	代码路径
SPC1168 系列、SPC2168 系列	SDK 目录\0_Examples\ADC_SHA_Short_Detect

3 SPC1169 系列

3.1 特色

ADC 模块的主要功能包括：

- 13 位分辨率的 ADC 核心，具有一个采样保持（S/H）电路：
 - 支持高达 2.5Msps 采样率；
- 模拟满量程输入为 3.657 V；
- 16 个独立通道，可独立配置：
 - 触发源；
 - 输入信号选择；
 - 采样转换时间和平均次数；
 - DMA 硬件握手接口（仅支持通道 0~7）；
 - 增益和偏移校准；
 - 记录结果的寄存器；
 - 中断处理；
- 支持如下事件触发 SOC 采样转换请求：
 - 软件触发请求；
 - 通用定时器 TIMER0、TIMER1、TIMER2 的触发请求；
 - 外部引脚的触发请求；
 - PWM 触发请求，包括：PWM0SOCA~PWM0SOCC，PWM1SOCA~PWM1SOCC，PWM2SOCA~PWM2SOCC，PWM3SOCA~PWM3SOCC；
 - 通道转换完成 EOC（End of conversion）事件；
- 3 个后处理单元：
 - 参考值进行比较，参考值大小和比较的极性可配；
 - 对比较结果进行上下限（阈值可配）卡测和过零检测，相应的事件可配置为触发 CPU 中断或者 PWM 封锁；

3.2 功能描述

在电力电子系统中，通常使用 ADC 进行电压采样，将模拟电压转换为数字电压。ADC 的精度、参考电压、采样时间、转换时间和保持器数量，如表 3-1 所示：

表 3-1: ADC 配置

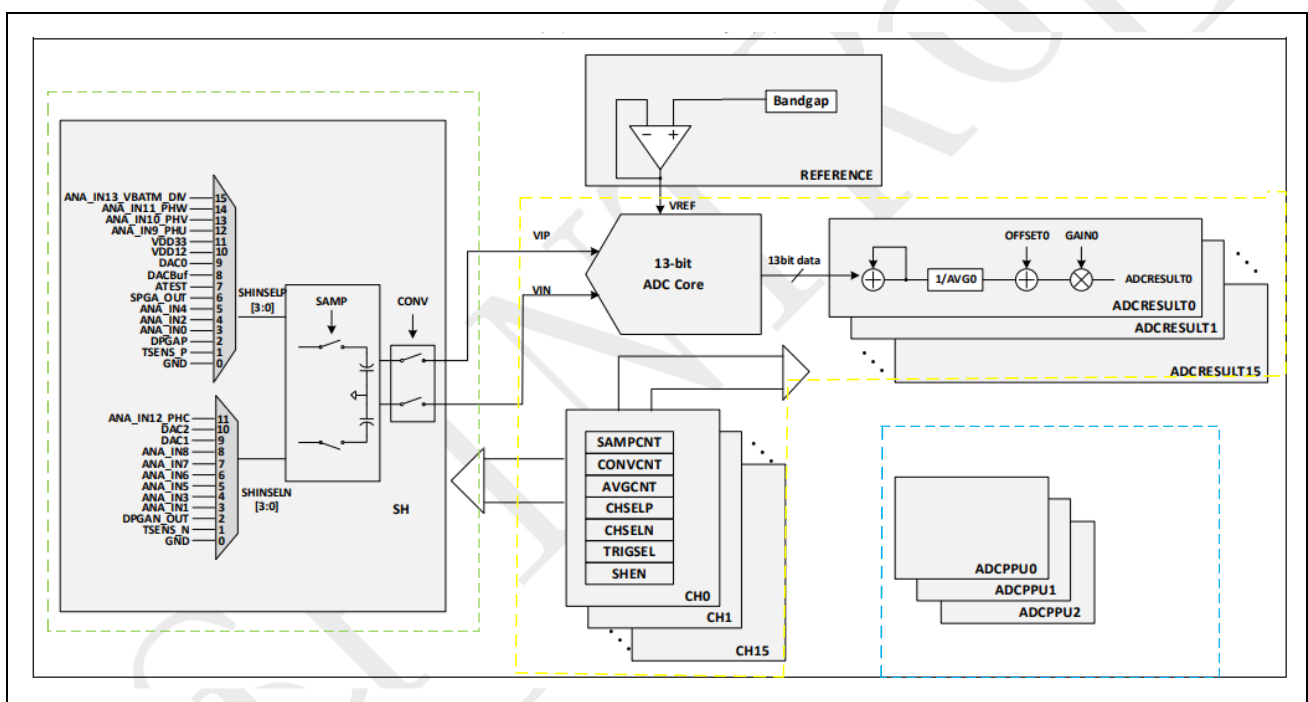
ADC 位数	参考电压值	采样经典时间	转换时间	保持器数量
13	3.65V	200ns	200ns	1

在实际使用 ADC 过程中，可以按照功能区块来帮助理解怎样使用 ADC，ADC 大体上可以按照图 3-1 的不同颜色虚线分成 3 个功能区块，在使用过程中按照如下步骤进行配置：

- 首先绿色虚线框内只需要使能采样及转换开关，这个步骤可以等其它配置都设置完成之后再行进行；

- 然后设置黄色虚线框部分，在这部分中，有 16 路独立的 Channel 采样转换配置可以预先设置，每个 Channel 均可根据不同的目的而配置不同的参数，例如：采样时间（具体需要设置多长的采样时间，请参考《ADC 建立时间计算方法使用指南》），ADC core 转换时间，是否需要多次采样且取平均值，ADC 输入的正负端，触发源选择等；在设置完这些信息之后，对应 Channel 的设置产生的 ADC 转换结果将存储在对应的 ADCRESULTSx (x=0,1, ..., 15) 中。
- 最后，如果有需求，可以继续设置蓝色虚线框部分，此部分即为 ADC PPU 单元。通常工程使用中，需要知道 ADC 转换的结果相对某个参考值而言是过高还是过低，这部分的工作就可以交给 PPU 单元处理，以减少 CPU 的计算压力。除此之外，PPU 单元还可以检测出从发起 ADC 转换请求至开始 ADC 转换时的时间计数。

图 3-1: ADC 功能分区框图



3.3 功能实例

3.3.1 实例 1: ADC 单端采样

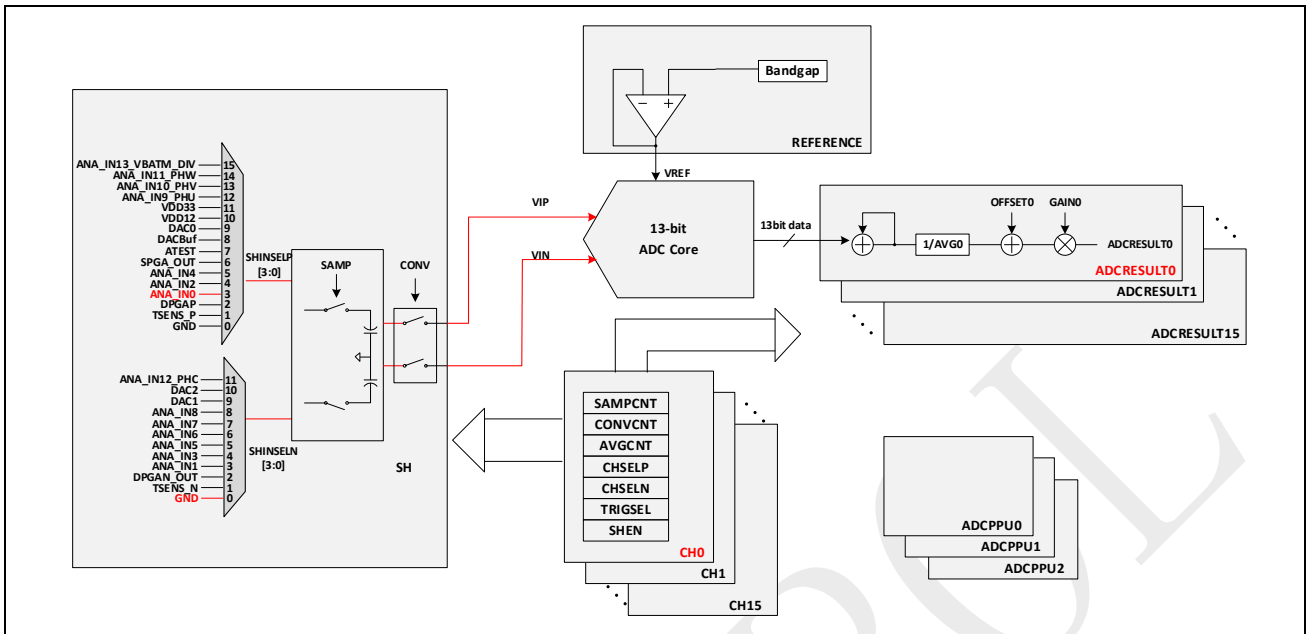
3.3.1.1 功能需求

利用 ADC 进行单端（也即有一端接 GND）采样，其中 SHINSELP 正端输入选择和 SHINSELN 负端输入其中一端接 GND，另一端接非 GND。

3.3.1.2 功能实现

1. 确定单端采样信号流如图 3-2，一端连接 ANA_IN1，另一端接地。

图 3-2: 单端采样信号流



2. 初始化 ADC 时钟。
3. ADC 通道选择，输入信号的选择，触发源，采样时间，转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-2:

表 3-2: 实例 1 代码路径

MCU 产品型号	代码路径
SPC1169 系列	SDK 目录\0_Examples\ADC_Single_Ended_Result

3.3.2 实例 2: ADC 双端采样

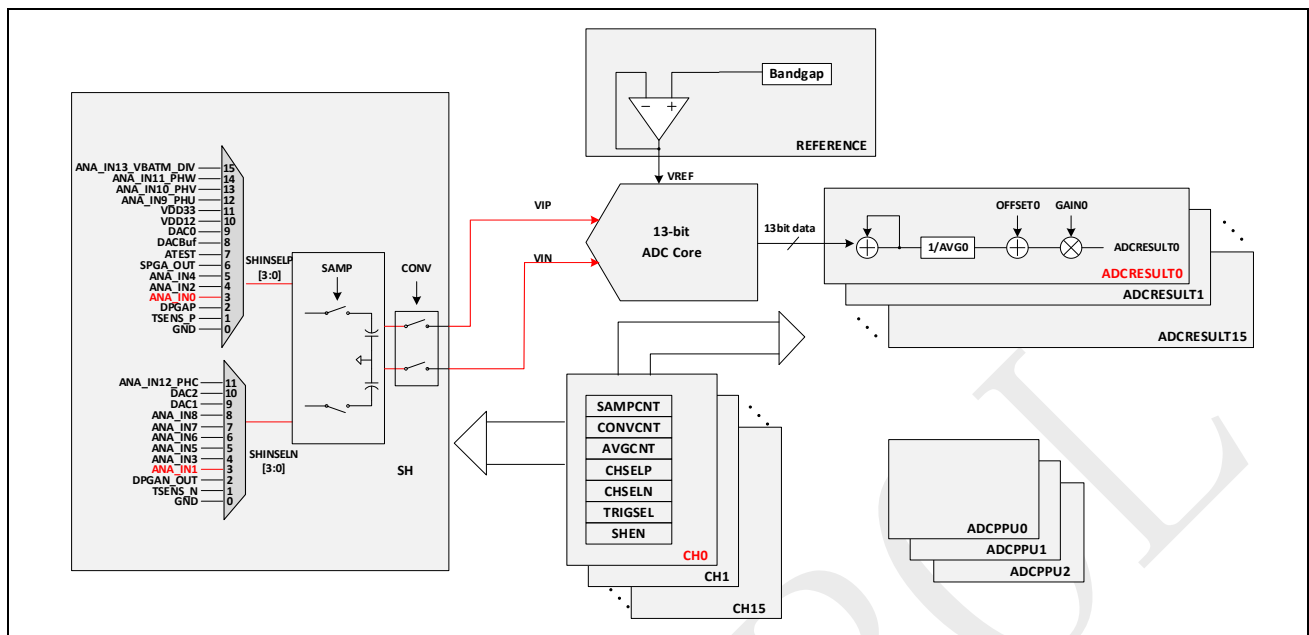
3.3.2.1 功能需求

利用 ADC 进行双端采样，其中 SHINSELN 正端输入选择和 SHINSELN 负端输入选择两个非 GND。

3.3.2.2 功能实现

1. 确定单端采样信号流如图 3-3 一端连接 ANA_IN1，另一端接 ANA_IN0。

图 3-3: 双端采样信号流



2. 初始化 ADC 时钟。
3. ADC 通道选择，输入信号的选择，触发源，采样时间，转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-3:

表 3-3: 实例 2 代码路径

MCU 产品型号	代码路径
SPC1169 系列	SDK 目录\0_Examples\ADC_Differential_Result

3.3.3 实例 3: ADC 后处理单元

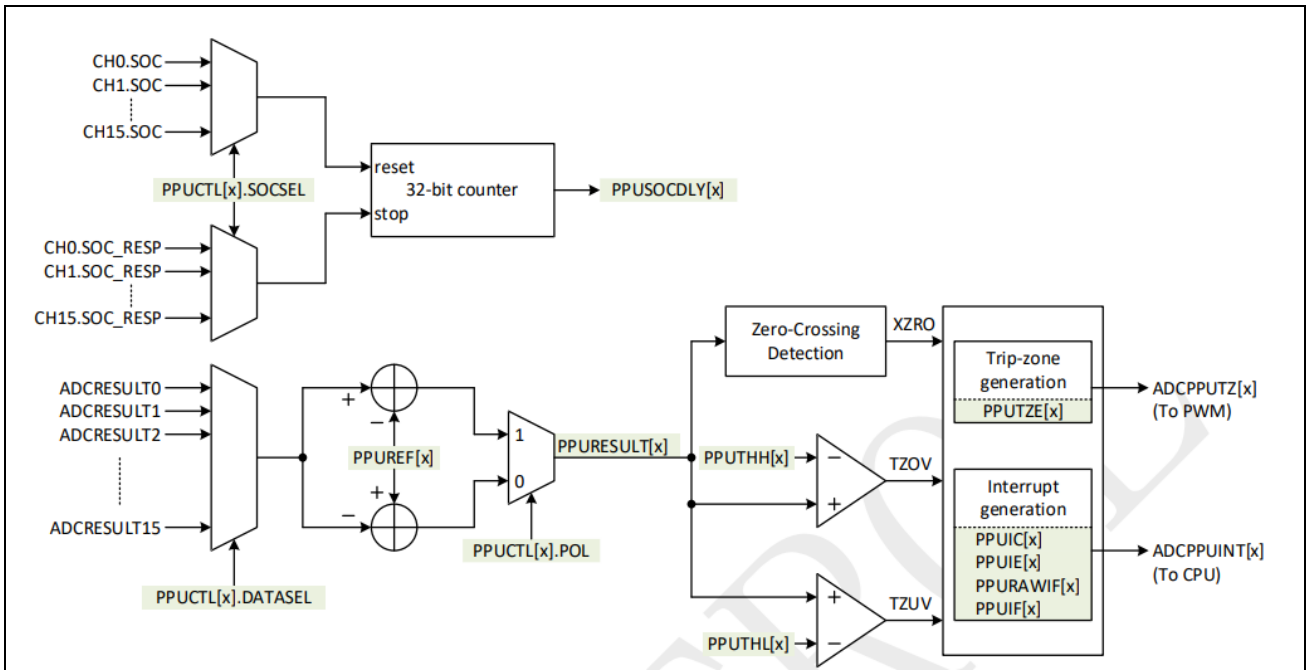
3.3.3.1 功能需求

使用 ADC 后处理单元，检测 ADC 转换的结果相对某个参考值而言是过高还是过低。

3.3.3.2 功能实现

1. 确定 ADC 后处理单元 PPU 结构，如图 3-4 所示:

图 3-4: PPU 结构图



2. 初始化 ADC 时钟。
3. ADC 通道选择，输入信号的选择，触发源，采样时间，转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。
5. 打开 PPU 过压、欠压、过参考电压的中断；
6. 设置 PPU 过压以及欠压阈值；
7. 选择 PPU 的输入，设置比较极性，设置参考电压值，随后使能 PPU；

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-4:

表 3-4: 实例 3 代码路径

MCU 产品型号	代码路径
SPC1169 系列	SDK 目录\0_Examples\ ADC_SHA_PPU

3.3.4 实例 4: ADC 开路检测

3.3.4.1 功能需求

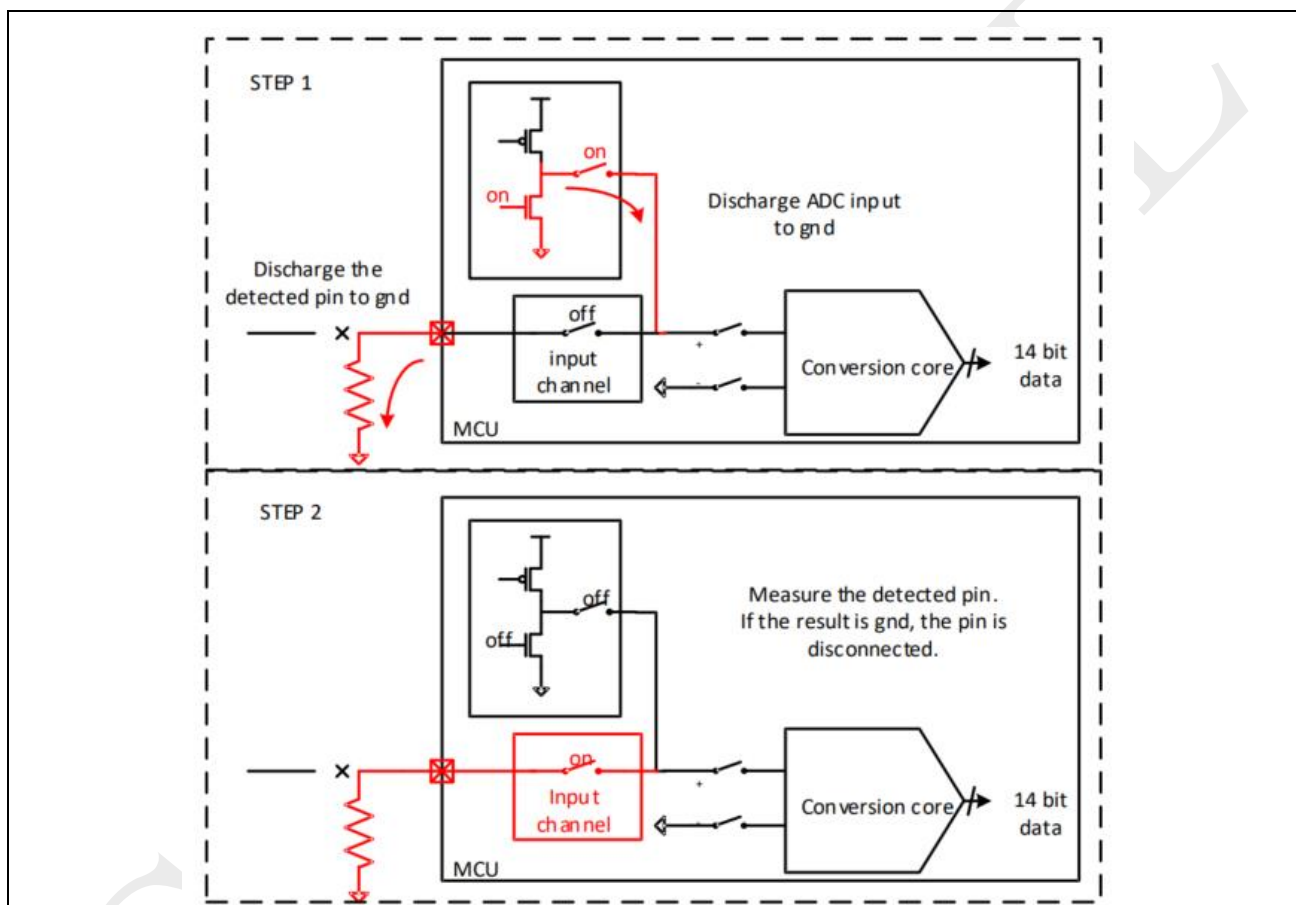
使用 ADC 单元支持的两种开路检测方式进行开路检测:

- 使用预放电故障注入法，将节点放电到地的方式进行开路检测；
- 使用预充电故障注入法，将节点充电到电源的方式进行开路检测；

3.3.4.2 功能实现

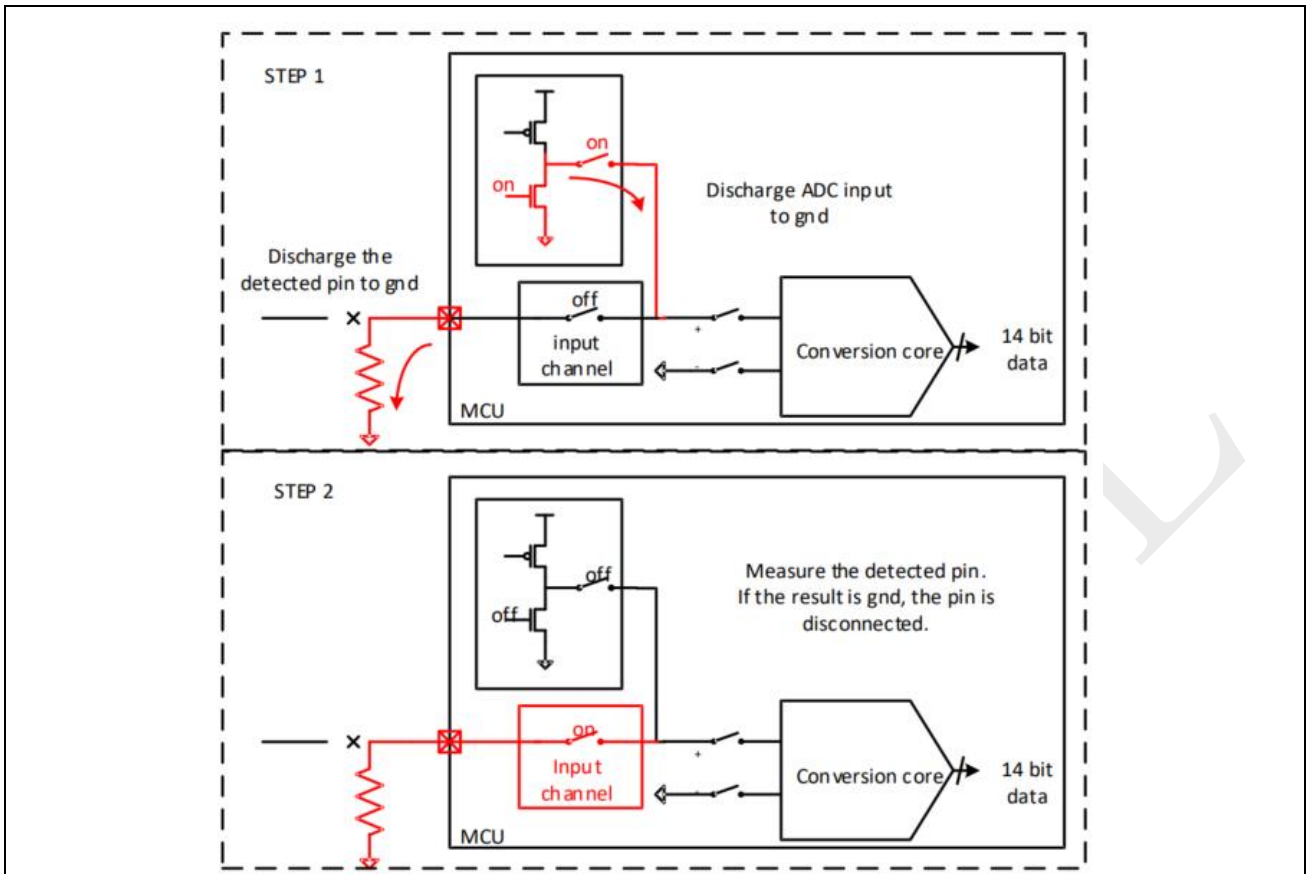
1. 预放电故障注入法检测如图 3-5，即检测连接到采样器正输入端的引脚是否浮动。
 - a) 步骤 1: 将 PIN 脚切成 GPIO 功能，设置为输入，并使能 GPIO 下拉，释放 IO 上残存的电量；将 ADC 采样保持的正端强制接地，释放通路上的残存电量；
 - b) 步骤 2: 将 ADC 采样保持的正端恢复断开状态，将 PIN 脚切成 ADC 功能，降低 ADC 模块的时钟频率，设置较长采样时间，以期获得更为准确的测试数据，随后开始测量，获得电压数值 Result1；

图 3-5: 检测 ADC 输入端引脚是否浮空（预放电）



2. 预充电故障注入法检测如图 3-6，即检测连接到采样器正输入端的引脚是否悬空。
 - a) 将 PIN 脚切成 GPIO 功能，设置为输入，并使能 GPIO 上拉，给 IO 的通路进行预充电；将 ADC 采样保持的正端强制拉高，对采样保持器通路进行预充电；
 - b) 将 ADC 采样保持的正端恢复断开状态，将 PIN 脚切成 ADC 功能，降低 ADC 模块的时钟频率，设置较长采样时间，以期获得更为准确的测试数据，随后开始测量，获得电压数值 Result2；

图 3-6: 检测 ADC 输入端引脚是否浮空 (预充电)



- 结果判断, Result1 在 0V 左右, 且 Result2 在 3.3V 左右, 则表明采样器的输入未链接外部输入, 即正输入端处于漂浮状态。类似的步骤也可以用来检测采样器的负输入引脚是否悬空。

以上实现步骤的示例代码可参考 SDK 提供的 Demo, 如表 3-5:

表 3-5: 实例 4 代码路径

MCU 产品型号	代码路径
SPC1169 系列	SDK 目录\0_Examples\ADC_SHA_Open_Detect

3.3.5 实例 5: ADC 短路检测

3.3.5.1 功能需求

模拟输入可能存在短路情况, 可以使用预充电或放电电路进行故障注入来检测。

3.3.5.2 功能实现

- 输入短路检测功能可以用 ADCCTL 寄存器设置。检测可以分为下面几步:
 - ADC 测量输入 PIN 脚电压, 设为结果 DATA1。
 - 将 PIN 脚切成 GPIO 功能, 设置为输入, 给输入 PIN 接弱下拉 GND, ADC 测量输入 PIN 脚电压, 设为结果 DATA2。

- c) 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱上拉 VDD，ADC 测量输入 PIN 脚电压，设为结果 DATA3。
- d) 如果 $DATA1 \sim = DATA2 \sim = DATA3$ ，那么输入 PIN 脚短路。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-6：

表 3-6：实例 5 代码路径

MCU 产品型号	代码路径
SPC1169 系列	SDK 目录\0_Examples\ADC_SHA_Short_Detect

SPIN TROL

4 SPC2188 系列

4.1 特色

ADC 模块主要包含以下功能：

- 14 位分辨率转换内核以及一个差分采样保持电路：
 - 最小 140ns 转换时间，支持高达 4Msps 采样率；
 - 支持多达 24 个外部模拟输入通道的采样；
- 支持 3 个 ADC 同时采样。
- 满幅度输入：3.339V。
- 8 个控制通道，可独立配置：
 - 触发源；
 - 输入信号选择；
 - 采样转换时间和平均次数；
 - DMA 硬件握手接口；
 - 偏移和增益校准；
 - 记录结果的寄存器；
 - 中断处理；
- 支持如下事件触发 SOC 采样转换请求：
 - 软件请求；
 - 通道转换完成 EOC（End of Conversion）事件；
 - 来自 PWM 的请求；
 - 通用定时器 TIMERO、TIMER1、TIMER2 的请求；
 - 外部引脚输入的请求；
- 每个 ADC 提供 4 个数字后处理单元：
 - 测量从 SOC 触发事件产生到实际采样转换开始的延迟；
 - 与参考值进行比较，参考值大小和比较的极性可配；
 - 对比较结果进行上下限（阈值可配）卡测和过零检测，相应的事件可配置为触发 CPU 中断或者 PWM 封锁；

4.2 功能描述

在电力电子系统中，通常使用 ADC 进行电压采样，将模拟电压转换为数字电压。ADC 的精度、参考电压、采样时间、转换时间和保持器数量，如表 4-1 所示：

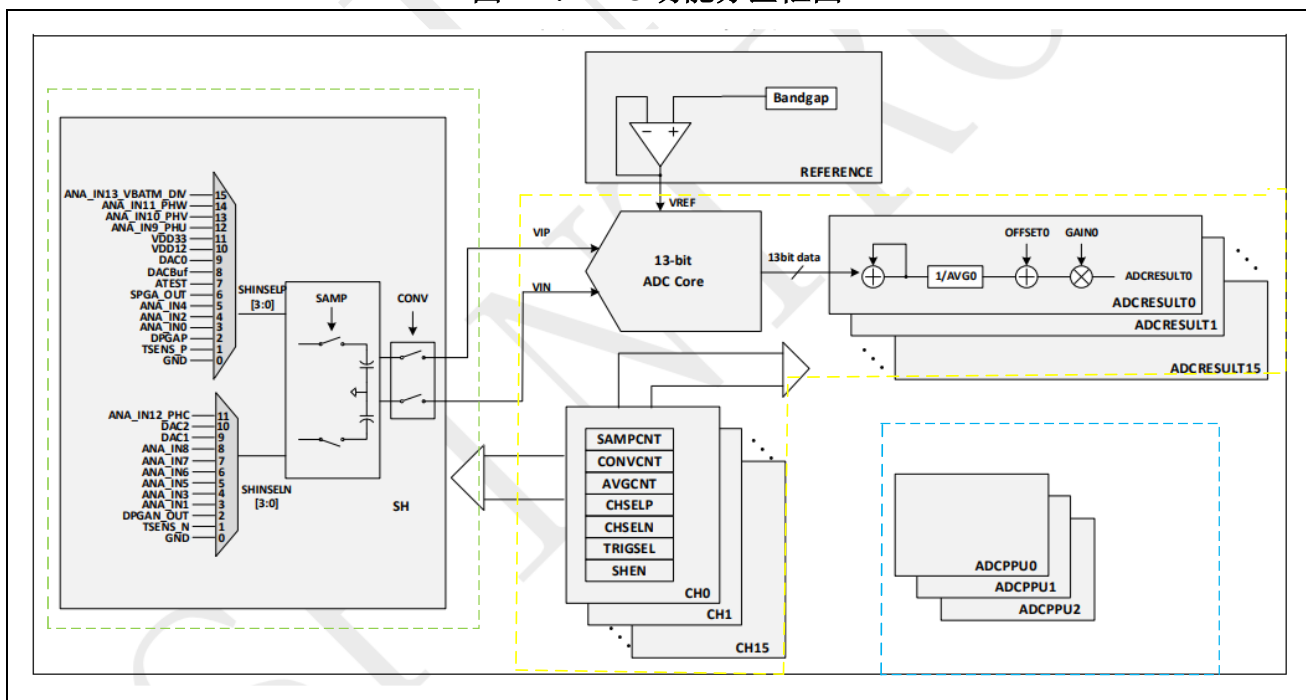
表 4-1: ADC 配置

ADC 位数	参考电压值	采样经典时间	转换时间	保持器数量
14	3.339V	140ns	140ns	1

在实际使用 ADC 过程中，可以按照功能区块来帮助理解怎样使用 ADC，ADC 大体上可以按照图 4-1 的不同颜色虚线分成 3 个功能区块，在使用过程中按照如下步骤进行配置：

- 首先绿色虚线框内只需要使能采样及转换开关，这个步骤可以等其它配置都设置完成之后再行进行；
- 然后设置黄色虚线框部分，在这部分中，有 16 路独立的 Channel 采样转换配置可以预先设置，每个 Channel 均可根据不同的目的而配置不同的参数，例如：采样时间（具体需要设置多长的采样时间，请参考《ADC 建立时间计算方法使用指南》），ADC core 转换时间，是否需要多次采样且取平均值，ADC 输入的正负端，触发源选择等；在设置完这些信息之后，对应 Channel 的设置产生的 ADC 转换结果将存储在对应的 ADCRESULTSx (x=0, 1, ..., 15) 中。
- 最后，如果有需求，可以继续设置蓝色虚线框部分，此部分即为 ADC PPU 单元。通常工程使用中，需要知道 ADC 转换的结果相对某个参考值而言是过高还是过低，这部分的工作就可以交给 PPU 单元处理，以减少 CPU 的计算压力。除此之外，PPU 单元还可以检测出从发起 ADC 转换请求至开始 ADC 转换时的时间计数。

图 4-1: ADC 功能分区框图



4.3 功能实例

4.3.1 实例 1: ADC 单端采样

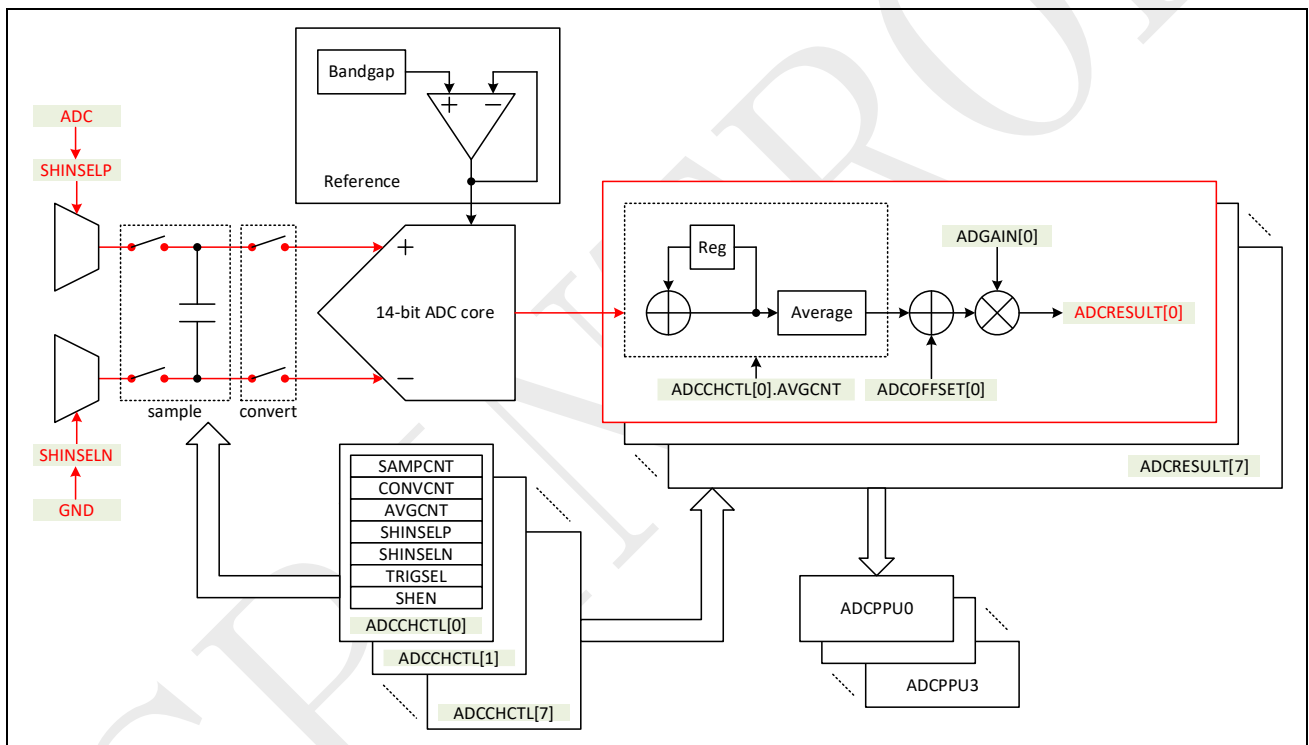
4.3.1.1 功能需求

利用 ADC 进行单端（也即有一端接 GND）采样，其中 SHINSELP 正端输入选择和 SHINSELN 负端输入其中一端接 GND，另一端接非 GND。

4.3.1.2 功能实现

1. 确定单端采样信号流如图 4-2，一端连接 ANA_IN1，另一端接地。

图 4-2: 单端采样信号流



2. 初始化 ADC 时钟。
3. ADC 通道选择，输入信号的选择，触发源，采样时间，转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 4-2:

表 4-2: 实例 1 代码路径

MCU 产品型号	代码路径
SPC2188 系列	SDK 目录\0_Examples\ ADC_Single_End_Abs_Result

4.3.2 实例 2: ADC 双端采样

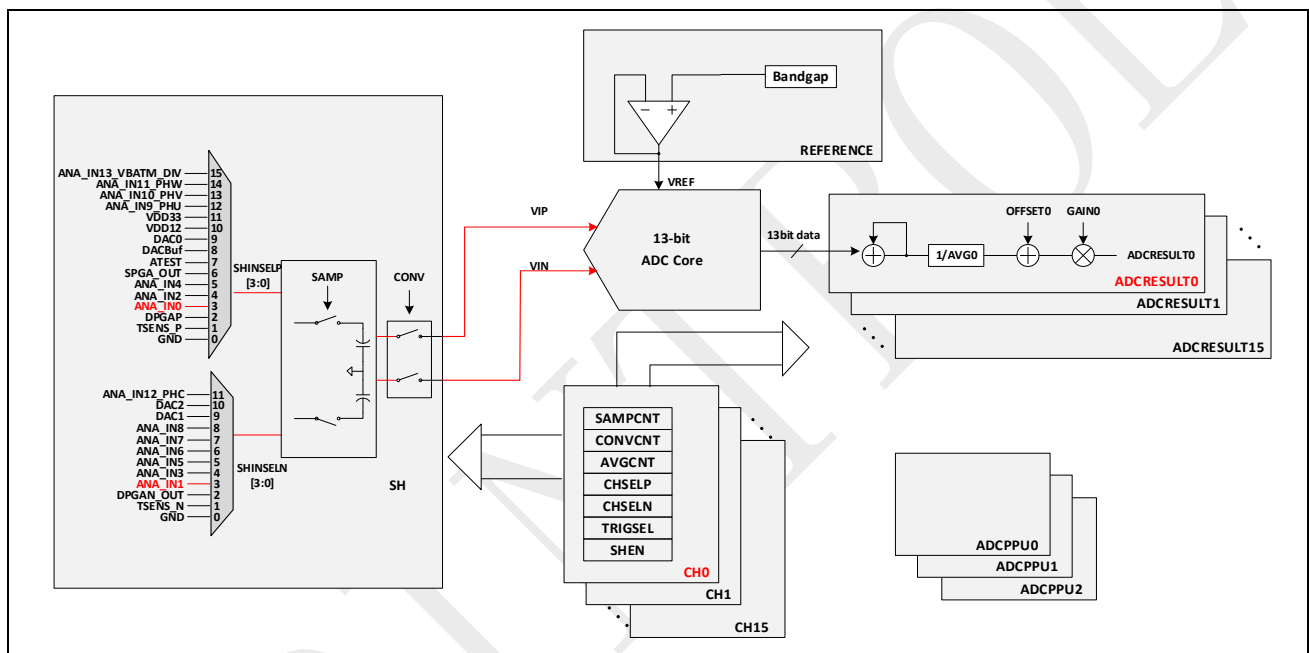
4.3.2.1 功能需求

利用 ADC 进行双端采样，其中 SHINSELP 正端输入选择和 SHINSELN 负端输入选择两个非 GND。

4.3.2.2 功能实现

1. 确定单端采样信号流如图 4-3 一端连接 ANA_IN1，另一端接 ANA_IN0。

图 4-3: 双端采样信号流



2. 初始化 ADC 时钟。
3. ADC 通道选择，输入信号的选择，触发源，采样时间，转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 3-3:

表 4-3: 实例 2 代码路径

MCU 产品型号	代码路径
SPC2188 系列系列	SDK 目录\0_Examples\ADC_Differential_Result

4.3.3 实例 3: ADC 后处理单元

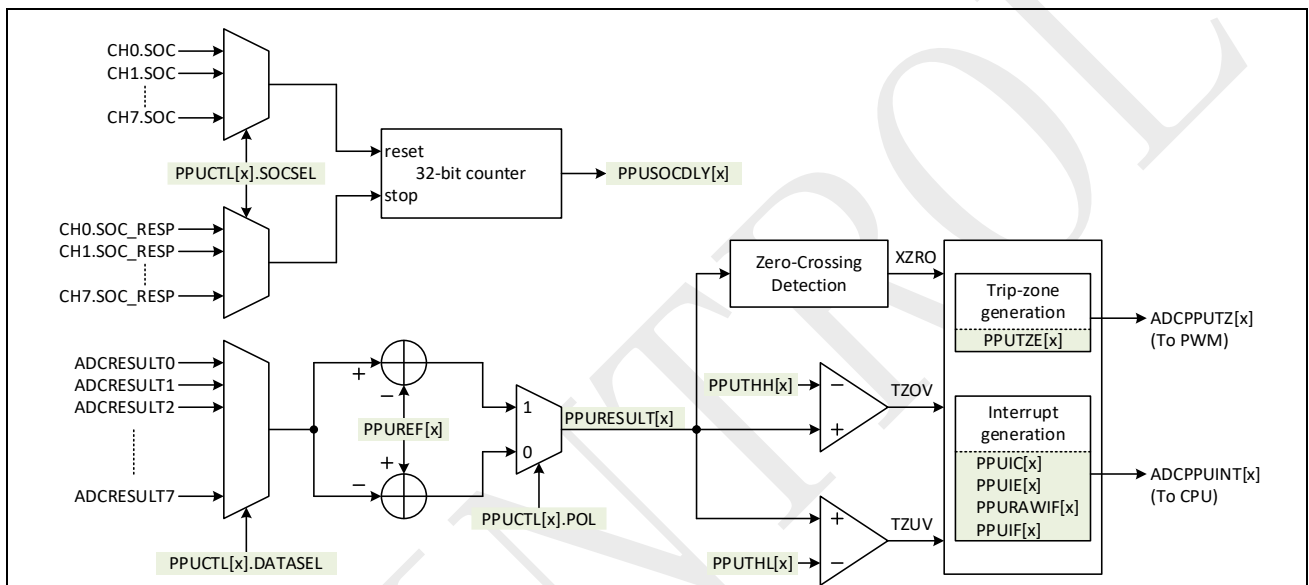
4.3.3.1 功能需求

使用 ADC 后处理单元，检测 ADC 转换的结果相对某个参考值而言是过高还是过低。

4.3.3.2 功能实现

1. 确定 ADC 后处理单元 PPU 结构，如图 4-4 所示：

图 4-4: PPU 结构图



2. 初始化 ADC 时钟。
3. ADC 通道选择，输入信号的选择，触发源，采样时间，转换时间的相关设置。
4. 将 GPIO 设置为模拟输入。
5. 打开 PPU 过压、欠压、过参考电压的中断；
6. 设置 PPU 过压以及欠压阈值；
7. 选择 PPU 的输入，设置比较极性，设置参考电压值，随后使能 PPU；

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 4-4:

表 4-4: 实例 3 代码路径

MCU 产品型号	代码路径
SPC2188 系列	SDK 目录\0_Examples\ADC_PPU

4.3.4 实例 4: ADC 开路检测

4.3.4.1 功能需求

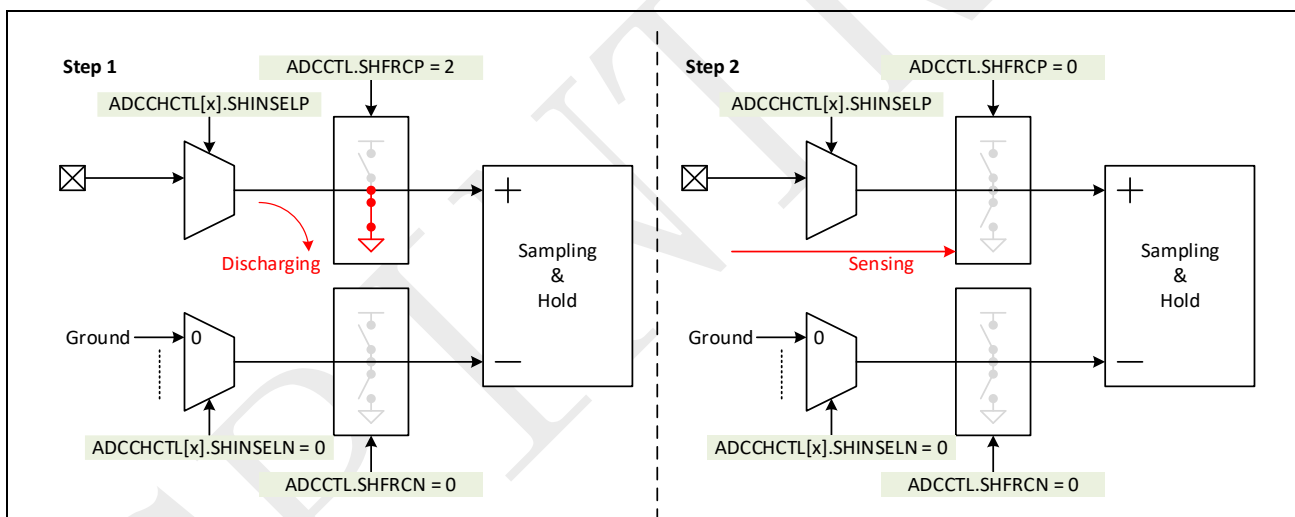
使用 ADC 单元支持的两种开路检测方式进行开路检测:

- 使用预放电故障注入法, 将节点放电到地的方式进行开路检测;
- 使用预充电故障注入法, 将节点充电到电源的方式进行开路检测;

4.3.4.2 功能实现

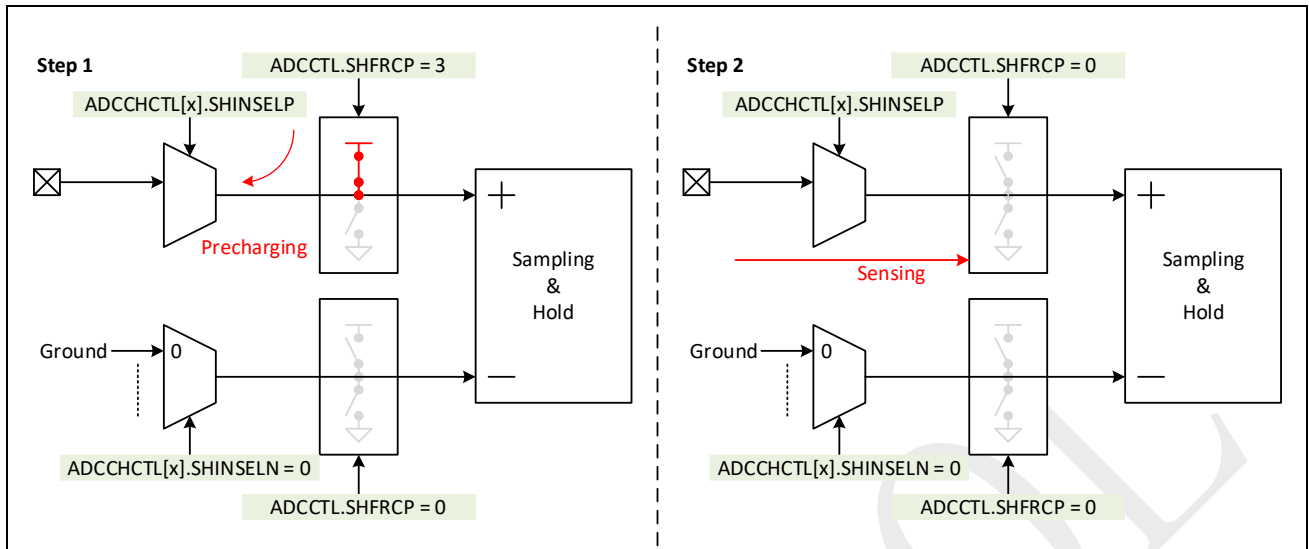
1. 预放电故障注入法检测如图 4-5, 即检测连接到采样器正输入端的引脚是否浮动。
 - a) 步骤 1: 将 PIN 脚切成 GPIO 功能, 设置为输入, 并使能 GPIO 下拉, 释放 IO 上残存的电量; 将 ADC 采样保持的正端强制接地, 释放通路上的残存电量;
 - b) 步骤 2: 将 ADC 采样保持的正端恢复断开状态, 将 PIN 脚切成 ADC 功能, 降低 ADC 模块的时钟频率, 设置较长采样时间, 以期获得更为准确的测试数据, 随后开始测量, 获得电压数值 Result1;

图 4-5: 检测 ADC 输入端引脚是否浮空 (预放电)



2. 预充电故障注入法检测图 4-6, 即检测连接到采样器正输入端的引脚是否悬空。
 - a) 将 PIN 脚切成 GPIO 功能, 设置为输入, 并使能 GPIO 上拉, 给 IO 的通路进行预充电; 将 ADC 采样保持的正端强制拉高, 对采样保持器通路进行预充电;
 - b) 将 ADC 采样保持的正端恢复断开状态, 将 PIN 脚切成 ADC 功能, 降低 ADC 模块的时钟频率, 设置较长采样时间, 以期获得更为准确的测试数据, 随后开始测量, 获得电压数值 Result2;

图 4-6: 检测 ADC 输入端引脚是否浮空 (预充电)



3. 结果判断，Result1 在 0V 左右，且 Result2 在 3.3V 左右，则表明采样器的输入未链接外部输入，即正输入端处于漂浮状态。类似的步骤也可以用来检测采样器的负输入引脚是否悬空。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 4-5:

表 4-5: 实例 4 代码路径

MCU 产品型号	代码路径
SPC2188 系列	SDK 目录\0_Examples\ADC_Open_Detect

4.3.5 实例 5：ADC 短路检测

4.3.5.1 功能需求

模拟输入可能存在短路情况，可以使用预充电或放电电路进行故障注入来检测。

4.3.5.2 功能实现

输入短路检测功能可以用 ADCCTL 寄存器设置。检测可以分为下面几步：

1. ADC 测量输入 PIN 脚电压，设为结果 DATA1。
2. 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱下拉 GND，ADC 测量输入 PIN 脚电压，设为结果 DATA2。
3. 将 PIN 脚切成 GPIO 功能，设置为输入，给输入 PIN 接弱上拉 VDD，ADC 测量输入 PIN 脚电压，设为结果 DATA3。
4. 如果 $DATA1 \sim = DATA2 \sim = DATA3$ ，那么输入 PIN 脚短路。

以上实现步骤的示例代码可参考 SDK 提供的 Demo，如表 4-6：

表 4-6：实例 5 代码路径

MCU 产品型号	代码路径
SPC2188 系列	SDK 目录\0_Examples\ADC_SHA_Short_Detect

5 常见问题 QA

5.1 ADC 转换结果为负数

如果 ADC 正端电压高于负端，则 ADC 转换结果会有负的码值，此时负码值，并非表示 ADC 输入端的某个端口电压为负电压，应该称之为相对负电压或者差分负电压。如手册第一章所述，ADC 端口的输入的绝对电压不允许为负电压，否则 ADC 将无法正常工作。以 S/HA 为例：

$$\text{Voltage ADC} = \text{AIP} - \text{AIN}$$

其中， $0 < \text{AIP} < +3.339$ ， $0 < \text{AIN} < +3.339$

则在 ADC 端电压范围为：

$$-3.339 < \text{Voltage ADC} < +3.339$$

此时转换出的码值就分为两种量程 $[0, 16383]$ ， $[-8192, 8191]$ ，本质只是表示方法不同，SDK 中提供了对应的两种接口可供调用。

Example Code

```
/******  
 * @brief      Get SH[x] result after EOC (End Of Conversion)  
 *  
 * @param[in]  eSHx : Sampling Holder x defined by ADC_SamplingHolderEnum  
 *              Following value is valid:  
 *              - \ref ADC_SH0  
 *  
 * @return     Signed value.  
 *              Range:  
 *              [ -8192  
 *                : 8191  
 *              ]  
 *  
*****/  
#define ADC_GetSHResult(ADCx,eSHx)  
\ ( (int32_t)READ_REG( (ADCx)->SHRAWCODE[eSHx] ) )  
  
/******  
 * @brief      Get SH[x] raw result after EOC (End Of Conversion)  
 *  
 * @param[in]  eSHx : Sampling Holder x defined by ADC_SamplingHolderEnum  
 *              Following value is valid:  
 *              - \ref ADC_SH0  
 *  
 * @return     Unsigned value.  
 *              Range: [ 0 : 16383 ]  
 *  
*****/  
#define ADC_GetSHRawResult(ADCx,eSHx)  
\ ( (uint32_t)( ADC_GetSHResult(ADCx, eSHx) + 8192 ) )  
  
/******  
 * @brief      Get CH[x] result value  
 *  
*****
```

Example Code

```

*
* @param[in]  eCHx:  Channel x defined by ADC_ChannelEnum
*              Following value is valid:
*              - \ref ADC_CH0 ~ ADC_CH7
*
* @return     Signed value.
*              Range:
*              [ -8192
*                : 8191
*              ]
*
*****/
#define ADC_GetChannelResult(ADCx,eCHx)
\
    ( (int32_t)READ_REG( (ADCx)->ADCRESULT[eCHx] ) )

/*****
* @brief      Get CH[x] raw result value
*
* @param[in]  eCHx:  Channel x defined by ADC_ChannelEnum
*              Following value is valid:
*              - \ref ADC_CH0 ~ ADC_CH7
*
* @return     Unsigned value.
*              Range: [ 0 : 16383 ]
*
*****/
#define ADC_GetChannelRawResult(ADCx,eCHx)
\
    ( (uint32_t)( ADC_GetChannelResult(ADCx, eCHx) + 8192 ) )

/*****
* @brief      For Single End Mode Only (One terminal is GND)
*              Get Trim result(code) from ADCx result register
*
* @param[in]  eCHx:  Channel x defined by ADC_ChannelEnum
*              Following value is valid:
*              - \ref ADC_CH0 ~ ADC_CH7
*
* @return     Signed value.
*              Range: [ 0 : 8192 ]
*
*****/
#define ADC_GetChannelAbsoluteResult(ADCx,eCHx)
\
    ( ABS( ADC_GetChannelResult(ADCx, eCHx) ) )
    
```

注意： 当使用 ADC 进行单端采样时，即使被测电压是绝对正电压，但若将此电压接在 ADC 负端，则测量出来的码值采用[-8192,8191]量程的话，此时最后的码值是负值，但并不代表输入电压是绝对负电压。