

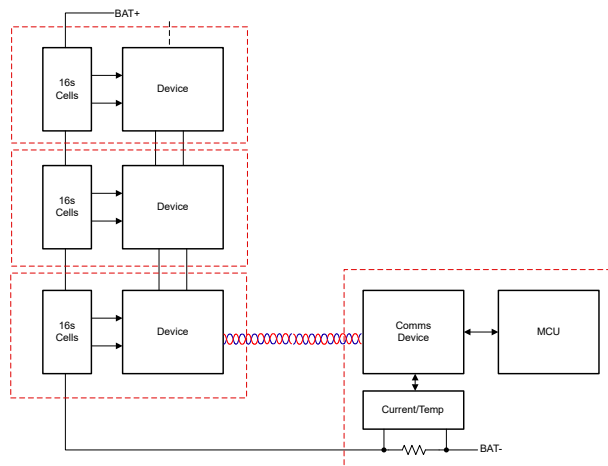
## BQ79616 16 节串联电池监测器、均衡器和集成硬件保护器

### 1 特性

- $\pm 1.5\text{mV}$  ADC 精度
- 用于电压、温度诊断的内置冗余路径
- 在  $128\mu\text{s}$  内对所有电芯通道进行高精度的电芯电压测量
- 集成式后 ADC 可配置数字低通滤波器
- 汇流条连接和测量支持
- 内置由主机控制的硬件复位，用于仿真类似 POR 的器件复位
- 支持内部电芯均衡
  - $240\text{mA}$  的均衡电流
  - 内置平衡热管理，具有自动暂停和恢复控制功能
- 隔离式差分菊花链通信，采用可选的环形架构
- 通过通信线路传输嵌入式故障信号和心跳信号
- UART/SPI 主机接口/通信桥接器件 BQ79600
- 内置 SPI 控制器

### 2 应用

- [混合动力和电动动力总成系统中的电池管理系统 \(BMS\)](#)
- [带有电池管理系统的储能电池组](#)



简化版系统图

### 3 说明

BQ79616 器件可对 HEV/EV 的高压电池管理系统中 16 节串联电池模块提供高精度的电芯电压测量，且仅需不到  $200\mu\text{s}$  的时间就能完成。该监测器在同一封装类型中提供不同的通道选项，同时提供引脚对引脚兼容性，并支持在任何平台上高度重复使用既有的软件和硬件。集成式前端滤波器使系统能够在电池输入通道上采用简单、低额定电压的差分 RC 滤波器。集成式后 ADC 低通滤波器可实现经滤波的类直流电压测量，从而更准确地计算荷电状态 (SOC)。此器件支持自主内部电池平衡，并通过监测温度来自动暂停和恢复平衡，以免出现过热条件。

#### 封装信息

器件型号	封装 <sup>(1)</sup>	封装尺寸 (标称值)
BQ79616	HTQFP (64 引脚)	10.00mm × 10.00mm

- (1) 如需了解所有可用封装，请参阅数据表末尾的可订购产品附录。



## 内容

<b>1 特性</b> .....	<b>1</b>	<b>6.5 寄存器映射</b> .....	<b>105</b>
<b>2 应用</b> .....	<b>1</b>	<b>7 应用和实施</b> .....	<b>182</b>
<b>3 说明</b> .....	<b>1</b>	7.1 应用信息.....	182
<b>4 引脚配置和功能</b> .....	<b>3</b>	7.2 典型应用.....	182
<b>5 规格</b> .....	<b>6</b>	<b>8 电源相关建议</b> .....	<b>192</b>
5.1 绝对最大额定值.....	6	<b>9 布局</b> .....	<b>193</b>
5.2 ESD 等级.....	6	9.1 布局指南.....	193
5.3 建议运行条件.....	7	9.2 布局示例.....	195
5.4 热性能信息.....	7	<b>10 器件和文档支持</b> .....	<b>199</b>
5.5 电气特性.....	7	10.1 器件支持.....	199
5.6 时序要求.....	13	10.2 接收文档更新通知.....	199
5.7 典型特性.....	16	10.3 支持资源.....	199
<b>6 详细说明</b> .....	<b>18</b>	10.4 商标.....	199
6.1 概述.....	18	10.5 静电放电警告.....	199
6.2 功能方框图.....	19	10.6 术语表.....	199
6.3 特性说明.....	20	<b>11 修订历史记录</b> .....	<b>199</b>
6.4 器件功能模式.....	96	<b>12 机械、封装和可订购信息</b> .....	<b>199</b>

## 4 引脚配置和功能

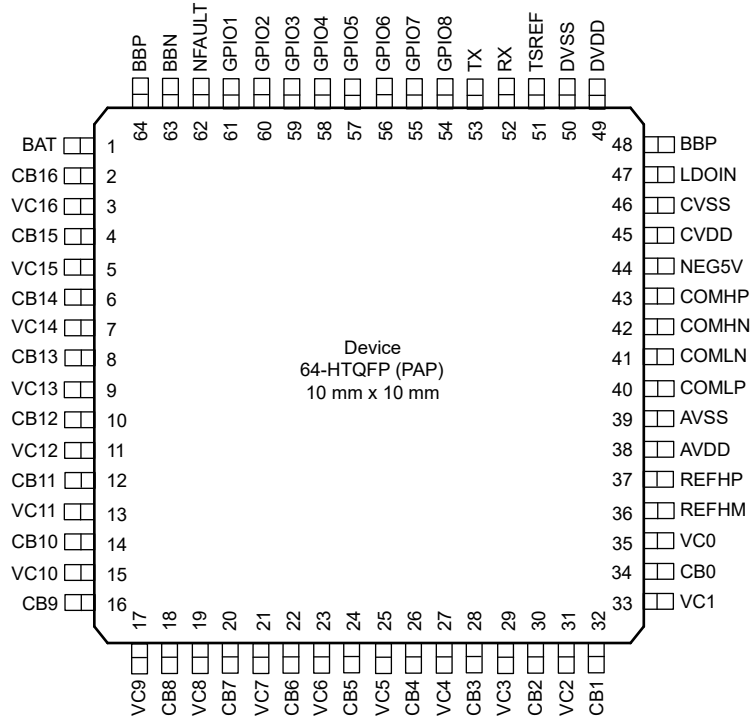


表 4-1. 引脚功能

引脚		类型	说明
名称	编号		
BQ79616			
BAT	1	P	电源输入和模块顶部测量输入。连接到电池模块的顶部电芯。
NPN	48	P	连接到外部 NPN 晶体管的基极。
LDOIN	47	P	6V 预稳压模拟电源输入/检测引脚。连接到外部 NPN 晶体管的发射极，并将 0.1μF 去耦电容器连接到 CVSS。
AVDD	38	P	5V 稳压输出。AVDD 为内部模拟电路供电。使用一个电容器将 AVDD 旁路至 AVSS。
AVSS	39	GND	模拟地。内部模拟电路的接地连接。在外部连接 DVSS、CVSS、REFHM 和 AVSS。
NEG5V	44	P	用于菊花链和主 ADC 的负 5V 电荷泵。使用电容器连接到 CVSS。
DVDD	49	P	1.8V 稳压输出。DVDD 为内部数字电路供电。使用一个电容器将 DVDD 旁路至 DVSS。
DVSS	50	GND	数字地。内部数字逻辑的接地连接。在外部连接 DVSS、CVSS、REFHM 和 AVSS。
CVDD	45	P	5V 菊花链通信和 I/O 电源。CVDD 为堆栈菊花链通信收发器电路和 I/O 引脚供电。该电源还在、活动和睡眠状态下支持额外的 10mA 外部负载。
CVSS	46	GND	菊花链通信接地。内部菊花链收发器的接地连接。在外部连接 DVSS、CVSS、REFHM 和 AVSS。
TSREF	51	P	NTC 热敏电阻的 5V 偏置电压。将 TSREF 连接到 NTC 电阻分压器网络的顶部，并在 GPIO 配置为进行 NTC 温度监测时连接到 GPIO。使用一个电容器将 TSREF 旁路至 CVSS。
REFHP	37	P	精密基准输出引脚。使用一个电容器旁路至 REFHM。
REFHM	36	GND	精密基准接地。用于内部精密基准的接地连接。在外部连接 DVSS、CVSS、REFHM 和 AVSS。
VC16	3	I	电芯电压检测输入。连接到电芯 16 的正极端子。将差分 RC 滤波器连接到 VC15。按照电芯连接中的说明，将未使用的 NC 引脚连接到 BAT 引脚。
VC15	5	I	电芯电压检测输入。连接到电芯 15 的正极端子。将差分 RC 滤波器连接到 VC14。按照电芯连接中的说明，将未使用的 NC 引脚连接到 BAT 引脚。
VC14	7	I	电芯电压检测输入。连接到电芯 14 的正极端子。将差分 RC 滤波器连接到 VC13。按照电芯连接中的说明，将未使用的 NC 引脚连接到 BAT 引脚。

表 4-1. 引脚功能 (续)

引脚		类型	说明
名称	编号		
BQ79616			
VC13	9	I	电芯电压检测输入。连接到电芯 13 的正极端子。将差分 RC 滤波器连接到 VC12。按照电芯连接中的说明, 将未使用的 NC 引脚连接到 BAT 引脚。
VC12	11	I	电芯电压检测输入。连接到电芯 12 的正极端子。将差分 RC 滤波器连接到 VC11。
VC11	13	I	电芯电压检测输入。连接到电芯 11 的正极端子。将差分 RC 滤波器连接到 VC10。
VC10	15	I	电芯电压检测输入。连接到电芯 10 的正极端子。将差分 RC 滤波器连接到 VC9。
VC9	17	I	电芯电压检测输入。连接到电芯 9 的正极端子。将差分 RC 滤波器连接到 VC8。
VC8	19	I	电芯电压检测输入。连接到电芯 8 的正极端子。将差分 RC 滤波器连接到 VC7。
VC7	21	I	电芯电压检测输入。连接到电芯 7 的正极端子。将差分 RC 滤波器连接到 VC6。
VC6	23	I	电芯电压检测输入。连接到电芯 6 的正极端子。将差分 RC 滤波器连接到 VC5。
VC5	25	I	电芯电压检测输入。连接到电芯 5 的正极端子。将差分 RC 滤波器连接到 VC4。
VC4	27	I	电芯电压检测输入。连接到电芯 4 的正极端子。将差分 RC 滤波器连接到 VC3。
VC3	29	I	电芯电压检测输入。连接到电芯 3 的正极端子。将差分 RC 滤波器连接到 VC2。
VC2	31	I	电芯电压检测输入。连接到电芯 2 的正极端子。将差分 RC 滤波器连接到 VC1。
VC1	33	I	电芯电压检测输入。连接到电芯 1 的正极端子。将差分 RC 滤波器连接到 VC0。
VC0	35	I	电芯电压检测输入。连接到电芯 1 的负极端子。将差分 RC 滤波器连接到 AVSS。
CB16	2	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB15 的差分 RC 滤波器将该引脚连接到电芯 16 的正极端子。滤波电阻器还设置内部均衡电流。通过 RC 将未使用的 CB16 引脚连接到 BAT 引脚, 并按照电芯连接中的说明, 将未使用的 NC 引脚连接到 BAT 引脚。
CB15	4	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB14 的差分 RC 滤波器将该引脚连接到电芯 15 的正极端子。滤波电阻器还设置内部均衡电流。按照电芯连接中的说明, 将未使用的 NC 引脚连接到 BAT 引脚。
CB14	6	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB13 的差分 RC 滤波器将该引脚连接到电芯 14 的正极端子。滤波电阻器还设置内部均衡电流。按照电芯连接中的说明, 将未使用的 NC 引脚连接到 BAT 引脚。
CB13	8	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB12 的差分 RC 滤波器将该引脚连接到电芯 13 的正极端子。滤波电阻器还设置内部均衡电流。按照电芯连接中的说明, 将未使用的 NC 引脚连接到 BAT 引脚。
CB12	10	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB11 的差分 RC 滤波器将该引脚连接到电芯 12 的正极端子。滤波电阻器还设置内部均衡电流。
CB11	12	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB10 的差分 RC 滤波器将该引脚连接到电芯 11 的正极端子。滤波电阻器还设置内部均衡电流。
CB10	14	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB9 的差分 RC 滤波器将该引脚连接到电芯 10 的正极端子。滤波电阻器还设置内部均衡电流。
CB9	16	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB8 的差分 RC 滤波器将该引脚连接到电芯 9 的正极端子。滤波电阻器还设置内部均衡电流。
CB8	18	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB7 的差分 RC 滤波器将该引脚连接到电芯 8 的正极端子。滤波电阻器还设置内部均衡电流。
CB7	20	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB6 的差分 RC 滤波器将该引脚连接到电芯 7 的正极端子。滤波电阻器还设置内部均衡电流。
CB6	22	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB5 的差分 RC 滤波器将该引脚连接到电芯 6 的正极端子。滤波电阻器还设置内部均衡电流。
CB5	24	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB4 的差分 RC 滤波器将该引脚连接到电芯 5 的正极端子。滤波电阻器还设置内部均衡电流。
CB4	26	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB3 的差分 RC 滤波器将该引脚连接到电芯 4 的正极端子。滤波电阻器还设置内部均衡电流。
CB3	28	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB2 的差分 RC 滤波器将该引脚连接到电芯 3 的正极端子。滤波电阻器还设置内部均衡电流。
CB2	30	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB1 的差分 RC 滤波器将该引脚连接到电芯 2 的正极端子。滤波电阻器还设置内部均衡电流。
CB1	32	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 CB0 的差分 RC 滤波器将该引脚连接到电芯 1 的正极端子。滤波电阻器还设置内部均衡电流。

**表 4-1. 引脚功能 (续)**

引脚		类型	说明
名称	编号		
<b>BQ79616</b>			
CB0	34	I/O	电芯均衡连接。该引脚连接到内部电芯均衡 FET。使用到 AVSS 的差分 RC 滤波器连接到电芯 1 的负极端子。滤波电阻器还设置内部均衡电流。
BBP	64	I	汇流条连接。通过将 BBP 和 BBN 连接到汇流条的每一端，该通道以 5 倍的增益为 ADC 测量提供差分输入。
BBN	63	I	汇流条连接。通过将 BBP 和 BBN 连接到汇流条的每一端，该通道以 5 倍的增益为 ADC 测量提供差分输入。
RX	52	I	UART 接收器输入。使用外部电阻上拉至 CVDD，并将器件 RX 连接到主机 MCU 的 TX 输出。如果未使用（例如，对于堆栈器件），请将 RX 连接到 CVDD。
TX	53	O	UART 发送器输出。将器件 TX 连接到主机 MCU 的 RX 输入端，并将从主机侧上拉。如果未使用（例如，对于堆栈器件），则将其保持悬空。
COMHP	43	I/O	适用于菊花链连接的垂直双向通信接口。高侧（北侧）差分 I/O。将连接到菊花链配置中的下部相邻器件的低侧（南侧）COMLP 和 COMLN。如果未使用，请使用 1kΩ 电阻器连接 COMHP 和 COMHN。
COMHN	42	I/O	
COMLP	40	I/O	适用于菊花链连接的垂直双向通信接口。低侧（南侧）差分 I/O。将连接到菊花链配置中的上部相邻器件的高侧（北侧）COMHP 和 COMHN。如果未使用，请使用 1kΩ 电阻器连接 COMLP 和 COMLN。
COMLN	41	I/O	
NFAULT	62	O	故障指示输出。低电平有效。如果在基底器件上使用，请使用上拉电阻将 NFAULT 上拉至 CVDD，并将 NFAULT 连接到主机 MCU GPIO。如果未使用，则将其保持未连接状态。
GPIO1	61	I/O	通用输入/输出、配置选项包括： <ul style="list-style-type: none"> <li>对于外部 NTC 热敏电阻连接，将 NTC 热敏电阻连接到该引脚并上拉至 TSREF。用作 ADC 以及 OT 和 UT 硬件比较器的输入</li> <li>对于外部直流电压测量，配置为 ADC 的输入</li> <li>双向数字输入/输出</li> <li>用作 SPI 控制器的 I/O。</li> </ul>
GPIO2	60	I/O	
GPIO3	59	I/O	
GPIO4	58	I/O	
GPIO5	57	I/O	
GPIO6	56	I/O	
GPIO7	55	I/O	
GPIO8	54	I/O	

## 5 规格

### 5.1 绝对最大额定值

在自然通风条件下的工作温度范围内测得 (除非另有说明) <sup>(1)</sup>

		最小值	最大值	单位
输入电压	BAT、VC* (VC0 除外)、CB* (CB0 除外)、NFAULT、BBP、BBN 至 AVSS <sup>(2) (3)</sup>	-0.3	100	V
	CB0、VC0 至 AVSS	-0.3	5.5	V
	VCn 至 VCn-1, n = 1 至 16 <sup>(2)</sup>	-80	80	V
	CBn 至 CBn-1, n = 1 至 16 <sup>(3)</sup>	-0.3	16	V
	BBP 至 BBN	-80	80	V
	LDOIN 至 AVSS	-0.3	9	V
	NPNB 至 AVSS	-0.3	10	V
	AVDD 至 AVSS	-0.3	5.5	V
	DVDD 至 DVSS	-0.3	1.98	V
	CVDD 至 CVSS	-0.3	6	V
	TSREF 至 AVSS	-0.3	5.5	V
	REFHP 至 REFHM	-0.3	5.5	V
	NEG5V 至 AVSS	-5.5	0	V
	TX、RX 至 AVSS	-0.3	6	V
	COMHP、COMHN、COMLP、COMLN 至 CVSS	-20	20	V
	COMHP 至 COMHN、COMLP 至 COMLN	-5.5	5.5	V
	GPIO* 至 AVSS	-0.3	5.5	V
CB* 电流	在 75°C 环境温度下最多可均衡 8 个电芯		240	mA
I/O 电流	GPIO*、RX、TX 电流		10	mA
T <sub>OTP_PROG</sub>	在高于该温度的条件下器件不会开始 OTP 编程		55	°C
T <sub>A</sub>	环境温度	-40	130	°C
T <sub>J</sub>	结温	-40	150	°C
T <sub>stg</sub>	贮存温度	-65	150	°C

- (1) 超出“绝对最大额定值”运行可能会对器件造成永久损坏。“绝对最大额定值”并不表示器件在这些条件下或在“建议运行条件”以外的任何其他条件下能够正常运行。如果超出“建议运行条件”但在“绝对最大额定值”范围内使用，器件可能不会完全正常运行，这可能影响器件的可靠性、功能和性能并缩短器件寿命。本免责声明自 2021 年第 1 季度起新增。所有新发布的数据表 m
- (2) VC 引脚电压必须同时满足 VCn 至 AVSS 以及 VCn 至 VCn-1 的标准。
- (3) CB 引脚电压必须同时满足 CBn 至 AVSS 以及 CBn 至 CBn-1 的标准。

### 5.2 ESD 等级

			值	单位
V <sub>(ESD)</sub>	静电放电	人体放电模型 (HBM) <sup>(1)</sup>	±2000	V
			±500	
		充电器件模型 (CDM) <sup>(2)</sup>	±750	
		所有引脚		
		其他引脚 (1、16、17、32、33、48、49、64)		

- (1) JEDEC 文档 JEP155 指出：500V HBM 时能够在标准 ESD 控制流程下安全生产。
- (2) JEDEC 文档 JEP157 指出：250V CDM 时能够在标准 ESD 控制流程下安全生产。

### 5.3 建议运行条件

在自然通风条件下的工作温度范围内测得 (除非另有说明)

		最小值	标称值	最大值	单位
V <sub>BAT_RANGE</sub>	总模块电压, 全功能, 无 OTP 编程	9		80	V
V <sub>BAT_OTP_RANGE</sub>	总模块电压, 全功能, 允许 OTP 编程	11		80	V
V <sub>CELL_RANGE</sub>	VC <sub>n</sub> - VC <sub>n-1</sub> , 其中 n = 2 至 16	-1		5	V
	VC1 - VC0	0		5	V
	VC0、CB0 至 AVSS	-0.3		5	V
	VC1、VC2、CB1、CB2 至 AVSS	-0.3		80	V
	VC <sub>n</sub> 、CB <sub>n</sub> 至 AVSS, 其中 n = 3 至 16	3		80	V
V <sub>BB_RANGE</sub>	V <sub>BBP</sub> - V <sub>BBN</sub>	-600		800	mV
V <sub>CB_RANGE</sub>	CB <sub>n</sub> - CB <sub>n-1</sub> , 其中 n = 1 至 16	0		5	V
V <sub>IO_RANGE</sub>	RX、TX、NFAULT	0		CVDD	V
V <sub>GPIO_RANGE</sub>	GPIO <sub>n</sub> 输入, 其中 n = 1 至 8	0.2		4.8	V
I <sub>IO</sub>	GPIO <sub>n</sub> 、RX、TX, 其中 n = 1 至 8			5	mA
T <sub>A</sub>	工作温度	-40		105	°C

### 5.4 热性能信息

热指标 <sup>(1)</sup>		BQ79616	单位
		PAP (HTQFP)	
		64 引脚	
R <sub>θJA</sub>	结至环境热阻	21.6	°C/W
R <sub>θJC(top)</sub>	结至外壳 (顶部) 热阻	8.7	°C/W
R <sub>θJB</sub>	结至电路板热阻	7.9	°C/W
ψ <sub>JT</sub>	结至顶部特征参数	0.1	°C/W
ψ <sub>JB</sub>	结至电路板特征参数	7.8	°C/W
R <sub>θJC(bot)</sub>	结至外壳 (底部) 热阻	2.1	°C/W

(1) 有关新旧热指标的更多信息, 请参阅应用手册《[半导体和 IC 封装热指标](#)》。

### 5.5 电气特性

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, V<sub>BAT</sub> = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
<b>热关断</b>						
T <sub>SHUT</sub>	热关断 (上升方向)		130	137	152	°C
T <sub>SHUT_FALL</sub>	热关断 (下降方向)		112		129	°C
T <sub>SHUT_HYS</sub>	热关断 (上升/下降方向)			20		°C
T <sub>WARN_RANGE</sub>	热警告阈值 (上升方向)		85		115	°C
T <sub>WARN_HYS</sub>	热警告迟滞 (下降方向)			10		°C
T <sub>WARN_ACC</sub>	热警告精度 (±)			5		°C
<b>电源电流</b>						
I <sub>SHDN</sub>	SHUTDOWN 模式下的电源电流	I <sub>BAT</sub> 与 I <sub>LDOIN</sub> 之和		16	23	μA

## 5.5 电气特性 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
I <sub>SLP(IDLE)</sub>	SLEEP 模式下的基准电源电流。 无故障, 无保护比较器, 无电芯 均衡	I <sub>BAT</sub> 与 I <sub>LDOIN</sub> 之和 T <sub>A</sub> = -20°C 至 65°C		120	160	μA
		I <sub>BAT</sub> 与 I <sub>LDOIN</sub> 之和 T <sub>A</sub> = -40°C 至 125°C			220	μA
I <sub>ACT(IDLE)</sub>	ACTIVE 模式下的基准电源电流	I <sub>BAT</sub> 与 I <sub>LDOIN</sub> 之和 无故障, 无通信, 无保护器比较 器, 无电芯均衡		10.4	11.6	mA
I <sub>CB_EN</sub>	电芯均衡开启时的额外电源电流	至少 1 个电芯均衡 FET 导通, OT <sub>CB</sub> 已启用。其他功能无效		1	1.5	mA
I <sub>PROTCOMP</sub>	保护器比较器开启时的额外电源 电流	任一 OV/UV/OT/UT 保护器已启 用。其他功能无效		20	60	μA
I <sub>ADC</sub>	启用 ADC 时的额外电源电流	一个 ADC 开启, 并且正在进行转 换。其他功能无效。		0.4	0.6	mA
		2 个 ADC 开启, 并且正在进行转 换。其他功能无效。		0.6	0.9	mA
I <sub>BAT</sub>	流入 BAT 引脚的电源电流	ACTIVE 模式		150		μA
		SLEEP 模式		25		μA
		SHUTDOWN 模式		5		μA
I <sub>COMT</sub>	菊花链广播读取 128 字节数据期 间的额外电源电流	针对菊花链接口使用变压器隔离。		10		mA
I <sub>COMC</sub>	菊花链广播读取 128 字节数据期 间的额外电源电流	针对菊花链接口使用电容器隔离, 或电容器与扼流圈隔离。		10		mA
I <sub>OW_SINK</sub>	用于开路测试的灌电流, 适用于 VC1 至 VC16、CB1 至 CB16		380	500	600	μA
I <sub>OW_SOURCE</sub>	用于开路测试的拉电流, 适用于 VC0 和 CB0		380	500	600	μA
I <sub>LEAK</sub>	VC、CB 引脚上的漏电流	ADC 关闭时的 VC、CB 引脚。			0.1	μA
<b>电源 (LDOIN)</b>						
V <sub>LDOIN</sub>	LDOIN 电压	无 OTP 编程	5.9	6	6.1	V
		OTP 编程	7.9	8	8.1	V
<b>电源 (CVDD)</b>						
V <sub>CVDD</sub>	CVDD 输出电压	ACTIVE 和 SLEEP 模式	4.9	5	5.1	V
		SHUTDOWN 模式, 无外部 I <sub>load</sub>	3.95		6	V
		SHUTDOWN 模式, 最大外部 I <sub>load</sub> = 5mA	3.4		5.5	V
V <sub>CVDD_LDRG</sub>	CVDD 负载调整率	ACTIVE/SLEEP 模式, 最大外部 I <sub>load</sub> = 10mA	-30		30	mV
V <sub>CVDD_OV</sub>	CVDD OV 阈值	ACTIVE/SLEEP 模式, 最大外部 I <sub>load</sub> = 10mA	5.3	5.5	5.7	V
V <sub>CVDD_OVHYS</sub>	CVDD OV 迟滞	ACTIVE/SLEEP 模式, 最大外部 I <sub>load</sub> = 10mA	130	150	170	mV
V <sub>CVDD_UV</sub>	CVDD UV 阈值	SHUTDOWN 模式		3.5		V
		ACTIVE/SLEEP 模式, 最大外部 I <sub>load</sub> = 10mA	4.3	4.45	4.65	V
V <sub>CVDD_UVHYS</sub>	CVDD UV 迟滞			260		mV
V <sub>CVDD_ILIMIT</sub>	CVDD 电流限制	ACTIVE、SLEEP	35	60	85	mA

## 5.5 电气特性 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
<b>电源 (AVDD)</b>						
V <sub>AVDD</sub>	AVDD 输出电压	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	4.85	5	5.21	V
V <sub>AVDD_OV</sub>	AVDD OV 阈值	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	5.25	5.5	5.7	V
V <sub>AVDD_OVHYS</sub>	AVDD OV 迟滞	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	135	155	165	mV
V <sub>AVDD_UV</sub>	AVDD UV 阈值	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	4.25	4.45	4.6	V
V <sub>AVDD_UVHYS</sub>	AVDD UV 迟滞	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	235	340	430	mV
V <sub>AVDD_ILIMIT</sub>	AVDD 电流限制	C <sub>SUPPLIES</sub> = 1μF	10	30	50	mA
<b>电源 (DVDD)</b>						
V <sub>DVDD</sub>	DVDD 输出电压	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	1.65	1.8	1.95	V
V <sub>DVDD_OV</sub>	DVDD OV 阈值	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	1.95	2.1	2.3	V
V <sub>DVDD_OVHYS</sub>	DVDD OV 迟滞	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	40	65	120	mV
V <sub>DVDD_UV</sub>	DVDD UV 阈值	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	1.623	1.65	1.71	V
V <sub>DVDD_UVHYS</sub>	DVDD UV 迟滞	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	15	50	73	mV
V <sub>DVDD_ILIMIT</sub>	DVDD 电流限制		13	30	53	mA
<b>电源 (TSREF)</b>						
V <sub>TSREF</sub>	TSREF 输出电压	C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	4.975	5	5.025	V
V <sub>TSREF_LDRG</sub>	TSREF 负载调整率	I <sub>load</sub> = 4mA, C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	-30		30	mV
V <sub>TSREF_OV</sub>	TSREF OV 阈值	I <sub>load</sub> = 4mA, C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	5.2	5.6	5.8	V
V <sub>TSREF_OVHYS</sub>	TSREF OV 迟滞	I <sub>load</sub> = 4mA, C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	98	110	120	mV
V <sub>TSREF_UV</sub>	TSREF UV 阈值	I <sub>load</sub> = 4mA, C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	4.0	4.2	4.4	V
V <sub>TSREF_UVHYS</sub>	TSREF UV 迟滞	I <sub>load</sub> = 4mA, C <sub>SUPPLIES</sub> = 1μF, ACTIVE 模式	300	350	400	mV
V <sub>TSREF_ILIMIT</sub>	TSREF 电流限制	器件处于 ACTIVE 模式	15	30	52	mA
<b>负电荷泵 (NEG5V)</b>						
V <sub>NEG5V</sub>	NEG5V 引脚电压	C <sub>NEG5V</sub> = 0.1μF	-5.3	-4.6	-4.0	V
V <sub>NEG5V_UV</sub>	NEG5V UV 阈值 (上升)	C <sub>NEG5V</sub> = 0.1μF	-4.1	-3.5	-3.0	V
V <sub>NEG5V_UVRECOV</sub>	NEG5V UV 恢复	C <sub>NEG5V</sub> = 0.1μF	-4.3	-3.8	-3.3	V
<b>电芯均衡</b>						
R <sub>DSON</sub>	内部电芯均衡 FET Rdson	V <sub>Cn</sub> > 2.8V, 其中 n = 1 至 16; -40°C < T <sub>A</sub> < 125°C	1.45		4.6	Ω
V <sub>CB_DONE</sub>	V <sub>CB_DONE</sub> 检测阈值设置范围 (非精度)	阶跃为 25mV	2.45		4	V
V <sub>MB_DONE</sub>	V <sub>MB_DONE</sub> 检测阈值设置范围 (非精度)	阶跃为 1V	18		65	V
T <sub>OTCB</sub>	OTCB 阈值设置范围 (非精度)	阶跃为 2%	10		24	%
T <sub>COOLOFF</sub>	COOLOFF 阈值设置范围 (非精度)	阶跃为 2%	4		14	%
T <sub>CB_WARN</sub>	CB T <sub>WARN</sub> 阈值			105		°C
T <sub>CB_WARN_HYS</sub>	CB T <sub>WARN</sub> 磁滞			10		°C
<b>ADC 分辨率</b>						

## 5.5 电气特性 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
ENOB <sub>MAIN</sub>	主 ADC 有效位数			16		位
ENOB <sub>AUX</sub>	AUX ADC 有效位数			14		位
V <sub>LSB_ADC</sub>	用于 VCELL 测量的主 ADC 和 AUX ADC 分辨率			190.73		μV/LSB
V <sub>LSB_BB</sub>	(BBP-BBN) 测量的主 ADC 和 AUX ADC 分辨率			30.52		μV/LSB
V <sub>LSB_MAIN_DIETEMP1</sub>	DieTemp1 分辨率 (主 ADC)	ADC 测量以 0x000 = 0°C 为中心		0.025		°C/LSB
V <sub>LSB_AUX_DIETEMP2</sub>	DieTemp2 分辨率 (AUX ADC)	ADC 测量以 0x000 = 0°C 为中心		0.025		°C/LSB
V <sub>LSB_AUX_BAT</sub>	BAT 分辨率 (AUX ADC)	适用于来自 AUX ADC 的 BAT 电压测量		3.05		mV/LSB
V <sub>LSB_GPIO</sub>	GPIO 分辨率 (主 ADC 和 AUX ADC)			152.59		μV/LSB
V <sub>LSB_TSREF</sub>	TSREF 分辨率 (主 ADC)			169.54		μV/LSB
V <sub>LSB_DIAG</sub>	诊断测量分辨率	REFL、VBG2、LPBG5、VCM、AVAO_REF、AVDD_REF, 以及所有硬件保护器 DAC		152.59		μV/LSB
<b>ADC 精度</b>						
I <sub>VC_DELTA</sub>	VCn 至 VCn-1 输入电流差值 (当主 ADC 开启时)	T <sub>A</sub> = -20°C 至 65°C			1.8	μA
		T <sub>A</sub> = -40°C 至 105°C			2	μA
I <sub>VC</sub>	VCn 输入电流 (当主 ADC 开启时)			8	12	μA
R <sub>CB_INPUT</sub>	CB 引脚输入阻抗 (当 AUX ADC 开启时)			16		mΩ
V <sub>ACC_MAIN_CELL</sub>	主 ADC VCELL 测量的总通道精度, LPF_VCELL[2:0] = 0x03 设置;	2V < V <sub>CELL</sub> < 4.5V; T <sub>A</sub> = 25°C	-2.2		1.5	mV
		2V < V <sub>CELL</sub> < 4.5V; -20°C < T <sub>A</sub> < 65°C	-3.0		2.4	mV
		2V < V <sub>CELL</sub> < 4.5V; -40°C < T <sub>A</sub> < 105°C	-3.5		2.6	mV
		2V < V <sub>CELL</sub> < 4.5V; -40°C < T <sub>A</sub> < 125°C	-3.5		2.6	mV
		1V < V <sub>CELL</sub> < 5V; -40°C < T <sub>A</sub> < 125°C	-3.7		2.8	mV
		-2V < V <sub>CELL</sub> < 5V; -40°C < T <sub>A</sub> < 125°C	-4.5		3.2	mV
V <sub>ACC_AUX_CELL</sub>	AUX ADC 测量的总通道精度 (不包括 BAT 和 GPIO 精度);	2V < V <sub>CELL</sub> < 4.5V; T <sub>A</sub> = 25°C	-7.5		5.4	mV
		2V < V <sub>CELL</sub> < 4.5V; -20°C < T <sub>A</sub> < 65°C	-8.0		6.3	mV
		2V < V <sub>CELL</sub> < 4.5V; -40°C < T <sub>A</sub> < 105°C	-9.0		6.3	mV
		2V < V <sub>CELL</sub> < 4.5V; -40°C < T <sub>A</sub> < 125°C	-9.0		6.5	mV
		1V < V <sub>CELL</sub> < 5V; -40°C < T <sub>A</sub> < 125°C	-9.0		6.6	mV
		0V < V <sub>CELL</sub> < 5V; -40°C < T <sub>A</sub> < 125°C	-9.0		6.6	mV

## 5.5 电气特性 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
V <sub>(MAIN-AUX)</sub>	V <sub>CELL</sub> 和 OVDAC 基准诊断期间的主 -AUX 测量。在相同的 T <sub>A</sub> 下为两个 ADC 提供相同的输入电压	2V < V <sub>CELL</sub> < 4.5V ; T <sub>A</sub> = 25°C	-7.1		6.1	mV
		2V < V <sub>CELL</sub> < 4.5V ; -20°C < T <sub>A</sub> < 65°C	-7.8		6.6	mV
		2V < V <sub>CELL</sub> < 4.5V ; -40°C < T <sub>A</sub> < 105°C	-7.8		6.6	mV
		2V < V <sub>CELL</sub> < 4.5V ; -40°C < T <sub>A</sub> < 125°C	-7.8		6.7	mV
		1V < V <sub>CELL</sub> < 5V ; -40°C < T <sub>A</sub> < 125°C	-7.9		6.9	mV
		0V < V <sub>CELL</sub> < 5V ; -40°C < T <sub>A</sub> < 125°C	-7.9		6.9	mV
V <sub>ACC_MAIN_GPIO_RATIO</sub>	来自主 ADC 的测量 GPIO/来自主 ADC 的测量 TSREF	0.08V < V <sub>IN</sub> < 0.2V , 85°C < T <sub>A</sub> < 125°C	-0.20		0.20	%
		0.2V < V <sub>IN</sub> < 4.6V , -40°C < T <sub>A</sub> < 105°C	-0.20		0.20	%
		4.6V < V <sub>IN</sub> < 4.8V , -40°C < T <sub>A</sub> < -20°C	-0.30		0.30	%
V <sub>ACC_AUX_GPIO_RATIO</sub>	来自 AUX ADC 的测量 GPIO/来自 AUX ADC 的测量 TSREF	0.08V < V <sub>IN</sub> < 0.2V , 85°C < T <sub>A</sub> < 125°C	-0.20		0.20	%
		0.2V < V <sub>IN</sub> < 4.6V , -40°C < T <sub>A</sub> < 105°C	-0.20		0.20	%
		4.6V < V <sub>IN</sub> < 4.8V , -40°C < T <sub>A</sub> < -20°C	-0.30		0.30	%
V <sub>ACC_MAIN_GPIO_ABS</sub>	GPIO 测量的总通道精度 (主 ADC)	0.08V < V <sub>IN</sub> < 0.2V , 85°C < T <sub>A</sub> < 125°C	-4.00		4.00	mV
		0.2V < V <sub>IN</sub> < 4.6V , -40°C < T <sub>A</sub> < 105°C	-5.00		3.00	mV
		4.6V < V <sub>IN</sub> < 4.8V , -40°C < T <sub>A</sub> < -20°C	-4.00		4.00	mV
V <sub>ACC_AUX_GPIO_ABS</sub>	GPIO 上来自 AUX ADC 的精度	0.08V < V <sub>IN</sub> < 0.2V , 85°C < T <sub>A</sub> < 125°C	-6.00		6.00	mV
		0.2V < V <sub>IN</sub> < 4.6V , -40°C < T <sub>A</sub> < 105°C	-6.00		6.00	mV
		4.6V < V <sub>IN</sub> < 4.8V , -40°C < T <sub>A</sub> < -20°C	-6.00		6.00	mV
V <sub>ACC_MAIN_BB</sub>	来自主 ADC 的 (BBP-BBN) 的总通道精度	LPF_BB[2:0] = 0x00	-1.1		1.1	mV
V <sub>ACC_AUX_BB</sub>	来自 AUX ADC 的 (BBP-BBN) 的总通道精度		-4		4	mV
V <sub>ACC_AUX_BAT</sub>	BAT 引脚的 AUX ADC 测量精度	电池包 Vbat 范围 : 32V 至 72V , T <sub>A</sub> = -40°C 至 125°C	-225		135	mV
V <sub>ACC_AUX_REFL</sub>	AUX ADC 测量结果		1.092	1.1	1.106	V
V <sub>ACC_AUX_VBG2</sub>	AUX ADC 测量结果		1.092	1.1	1.106	V
V <sub>ACC_AUX_VCM</sub>	AUX ADC 测量结果		2.400	2.5	2.550	V
V <sub>ACC_AUX_AVAO_REF</sub>	AUX ADC 测量结果		2.400	2.47	2.550	V
V <sub>ACC_AUX_AVDD_REF</sub>	AUX ADC 测量结果		2.400	2.47	2.550	V
V <sub>ACC_AUX_OVDAC</sub>	AUX ADC 测量结果	设置为 4.475V ; T <sub>A</sub> = -20°C 至 65°C	4.450		4.500	V

## 5.5 电气特性 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
V <sub>ACC_AUX_OVDAC</sub>	AUX ADC 测量结果	设置为 4.475V; T <sub>A</sub> = -40°C 至 105°C	4.445		4.500	V
V <sub>ACC_AUX_OVDAC</sub>	AUX ADC 测量结果	设置为 4.475V; T <sub>A</sub> = -40°C 至 125°C	4.445		4.500	V
V <sub>ACC_AUX_OVDAC</sub>	AUX ADC 测量结果	设置为 3.8V	3.770		3.825	V
V <sub>ACC_AUX_OVDAC</sub>	AUX ADC 测量结果	设置为 3V	2.970		3.030	V
V <sub>ACC_AUX_UVDAC</sub>	AUX ADC 测量结果	设置为 3.1V	3.095	3.1	3.150	V
V <sub>ACC_AUX_VCBONEDAC</sub>	AUX ADC 测量结果	设置为 4V	3.950	4	4.050	V
V <sub>ACC_AUX_OTDAC</sub>	AUX ADC 测量结果	设置为 39%	1.900	1.95	2.000	V
V <sub>ACC_AUX_UTDAC</sub>	AUX ADC 测量结果	设置为 80%	3.950	4	4.050	V
V <sub>ACC_MAIN_TSREF</sub>	主 ADC 测量结果		4.975	5	5.025	V
V <sub>ACC_MAIN_DIETEMP</sub>	芯片温度 1 测量的总通道精度 (±)			3		°C
V <sub>ACC_AUX_DIETEMP</sub>	芯片温度 2 测量的总通道精度 (±)			6		°C
<b>基准电压</b>						
V <sub>REFH</sub>	REFHP 至 REFHM 电压		4.975	5	5.025	V
<b>硬件电压比较器/保护器 (电芯 OV/UV)</b>						
V <sub>OV_COMP_RANGE</sub>	OV 比较器检测阈值设置范围 (非精度)	阶跃为 25mV	2700		3000	mV
		阶跃为 25mV	3600		3800	mV
		阶跃为 25mV	4175		4500	mV
V <sub>OV_COMP_HYS</sub>	检测后的 OV 比较器迟滞			50		mV
V <sub>OV_COMP_ACC</sub>	OV 比较器精度	T <sub>A</sub> = -20°C 至 65°C	-24		24	mV
		T <sub>A</sub> = -40°C 至 105°C	-28		28	mV
V <sub>UV_COMP_RANGE</sub>	UV 比较器检测阈值设置范围 (非精度)	阶跃为 50mV	1200		3100	mV
V <sub>UV_COMP_HYS</sub>	检测后的 UV 比较器迟滞			50		mV
V <sub>UV_COMP_ACC</sub>	UV 比较器精度	T <sub>A</sub> = -20°C 至 65°C	-35		35	mV
		T <sub>A</sub> = -40°C 至 105°C	-50		50	mV
<b>硬件温度比较器/保护器 (NTC OT/UT)</b>						
V <sub>OT_COMP_RANGE</sub>	OT 比较器检测阈值设置范围 (非精度)	阶跃为 1%, 与 TSREF 成比例	10		39	%
V <sub>OT_COMP_HYS</sub>	检测后的 OT 比较器迟滞			2		%
V <sub>OT_COMP_ACC</sub>	OT 比较器精度		-0.5		0.5	%
V <sub>UT_COMP_RANGE</sub>	UT 比较器检测阈值范围	阶跃为 2%, 与 TSREF 成比例	66		80	%
V <sub>UT_COMP_HYS</sub>	检测后的 UT 比较器迟滞			2		%
V <sub>UT_COMP_ACC</sub>	UT 比较器精度		-0.5		0.5	%
<b>数字 I/O (TX、RX、GPIO、SPI 控制器)</b>						
V <sub>OH</sub>	输出为逻辑高电平 (TX、GPIO 作为输出)	GPIO 配置为输出。I <sub>OUT</sub> = 1mA		V <sub>CVDD</sub> - 0.3		V
V <sub>OL</sub>	输出为逻辑低电平 (TX、NFAULT、GPIO 作为输出)	GPIO 配置为输出。I <sub>OUT</sub> = 1mA			0.3	V
V <sub>IH</sub>	输入为逻辑高电平 (RX、GPIO 作为故障输入)	GPIO 配置为输入。I <sub>OUT</sub> = 1mA		0.75 × V <sub>CVDD</sub>		V

## 5.5 电气特性 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
V <sub>IL</sub>	输入为逻辑低电平 (RX、GPIO 作为故障输入)	GPIO 配置为输入。I <sub>OUT</sub> = 1mA			0.25 × V <sub>CVDD</sub>	V
R <sub>WK_PU</sub>	GPIO 弱上拉电阻		20	37	60	KΩ
R <sub>WK_PD</sub>	GPIO 弱下拉电阻		20	40	60	KΩ
<b>COML 和 COMH</b>						
R <sub>DCTX</sub>	发送器输出阻抗 (COML 和 COMH)			18		Ω
R <sub>DCCM</sub>	共模阻抗 (COML 和 COMH)			45		kΩ
V <sub>DCCM</sub>	共模电压 (COML 和 COMH)		2.21	2.5	2.76	V
V <sub>COMM_DATA1</sub>	通信接收器阈值范围 (V <sub>COMP</sub> - V <sub>COML</sub> )	CODE:0	0.4		1.2	V
V <sub>COMM_TONE1</sub>	Tone 接收器阈值范围 (V <sub>COMP</sub> - V <sub>COML</sub> )	CODE:0	0.4		1.2	V

## 5.6 时序要求

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	标称值	最大值	单位
<b>功耗状态时序</b>						
t <sub>SU(WAKE_SHUT)</sub>	从 SHUTDOWN 启动到 ACTIVE 模式	基底器件: 从 WAKE ping 结束到转发 WAKE 音调开始		6	10	ms
		堆栈器件: 从接收到的 WAKE 音调结束到转发 WAKE 音调开始		6	10	ms
t <sub>SU(SLP2ACT)</sub>	从 SLEEP 启动到 ACTIVE 模式 (使用 SLEEP2ACTIVE ping/音调)	基底器件: 从 SLEEP2ACTIVE ping 结束到转发 SLEEP2ACTIVE 音调开始			230	μs
		堆栈器件: 从 SLEEP2ACTIVE 音调结束到转发 SLEEP2ACTIVE 音调开始			230	μs
t <sub>SU(WAKE_SLP)</sub>	从 SLEEP 启动到 ACTIVE 模式 (使用 WAKE ping/音调)	基底器件: 从 WAKE ping 结束到转发 WAKE 音调开始			1	ms
		堆栈器件: 从接收到的 WAKE 音调结束到转发 WAKE 音调开始			1	ms
t <sub>SLP</sub>	从 ACTIVE 到 SLEEP 模式	从接收到 SLEEP 进入条件到进入 SLEEP 模式			100	μs
t <sub>SHTDN</sub>	从 ACTIVE 到 SHUTDOWN 模式	从接收到 SHUTDOWN 进入条件到进入 SHUTDOWN 模式 (所有 LDO 都降至其标称值的 10%)		20		ms
t <sub>RST</sub>	ACTIVE 模式期间的复位时间	数字复位完成时发送 CONTROL1[SOFT_RST] = 1			1	ms
t <sub>HWRST</sub>	发出硬件复位 ping/音调后器件将处于硬件复位状态的时间				75	ms
<b>电源时序</b>						
t <sub>TSREF_ON</sub>	TSREF 上升时间 (10% - 90%)	C <sub>TSREF</sub> = 1μF		6		ms
t <sub>TSREF_OFF</sub>	TSREF 斜降时间 (90% - 10%)	C <sub>TSREF</sub> = 1μF			8	ms
<b>ping 信号时序</b>						
t <sub>HLD_WAKE</sub>	RX 引脚上的 WAKE ping 低电平时间; CVDD 上无外部负载		2		2.5	ms

## 5.6 时序要求 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	标称值	最大值	单位
t <sub>HLD_SD</sub>	RX 引脚上的 SHUTDOWN ping 低电平时间; CVDD 上无外部负载		7		10	ms
t <sub>UART(SIA)</sub>	RX 引脚上的 SLEEPTOACTIVE ping 低电平时间		250		300	μs
t <sub>HLD_HWRST</sub>	RX 引脚上的 HW_RESET ping 低电平时间		36			ms
<b>COML 和 COMH (脉冲和音调时序)</b>						
t <sub>PW_DC</sub>	COMM: 通信的数据脉冲宽度 (半位时间)			250		ns
t <sub>RECLK_DC</sub>	COMM: 每个器件从 COMH 到 COML (或反向) 的数据重定时延迟			4	5	μs
t <sub>COMTONE</sub>	通信音调脉冲之间的时间 (基于 HFO)。通信音调包括 WAKE、SLEEPTOACTIVE、SHUTDOWN、HWRST 音调			11	15	μs
t <sub>COMMTONE_HI</sub>	每个通信脉冲的高电平时间 (基于 HFO)		0.92	1	1.08	μs
t <sub>COMMTONE_LO</sub>	每个通信脉冲的低电平时间 (基于 HFO)		0.92	1	1.08	μs
t <sub>FLTONE</sub>	故障音调脉冲之间的时间 (基于 LFO)。适用于故障音调和检测信号			11.5		μs
t <sub>FLTONE_HI</sub>	音调对每个脉冲的高电平时间			1		μs
t <sub>FLTONE_LO</sub>	音调对每个脉冲的低电平时间			1		μs
Π <sub>WAKEDET</sub>	检测为 WAKE 音调的脉冲数			60		个脉冲
Π <sub>WAKE</sub>	针对一个 WAKE 音调传输的脉冲数			90		个脉冲
Π <sub>SHDNDET</sub>	检测为 SHUTDOWN 音调的脉冲数			180		个脉冲
Π <sub>SHDN</sub>	针对一个 SHUTDOWN 音调传输的脉冲数			270		个脉冲
Π <sub>SLPtoACTDET</sub>	检测为 SLEEPTOACTIVE 音调的脉冲数			20		个脉冲
Π <sub>SLPtoACT</sub>	针对一个 SLEEPTOACTIVE 音调传输的脉冲数			30		个脉冲
Π <sub>HWRSTDET</sub>	检测为 HW_RESET 音调的脉冲数			540		个脉冲
Π <sub>HWRST</sub>	针对一个 HW_RESET 音调传输的脉冲数			810		个脉冲
Π <sub>HBDET</sub>	检测信号: 检测为有效音调的脉冲数			20		个脉冲
Π <sub>HB</sub>	检测信号: 针对一个音调传输的脉冲数			30		个脉冲
t <sub>HB_PERIOD</sub>	检测信号: 心跳信号突发之间的时长 (从一个心跳信号开始到下一个心跳信号开始)		360	400	440	ms
t <sub>HB_TIMEOUT</sub>	检测信号: 被视为未接收到心跳信号的超时		0.9	1	1.1	s
t <sub>HB_FAST</sub>	检测信号: 如果在该时间内接收到心跳信号, 则认为接收心跳信号过快			200		ms
Π <sub>FTONEDET</sub>	故障音调: 检测为有效音调的脉冲数			60		个脉冲
Π <sub>FTONE</sub>	故障音调: 针对一个音调传输的脉冲数			90		个脉冲

## 5.6 时序要求 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	标称值	最大值	单位
t <sub>FTONE_PERIOD</sub>	故障音调: 故障音调突发之间的时长 (从一个故障音调开始到下一个故障音调开始)			50		ms
t <sub>F<sub>TS</sub>_LATENCY</sub>	堆栈器件中的故障音调延迟	从器件接收到音调到同一器件检测到并生成其故障音调		48		µs
t <sub>F<sub>TB</sub>_LATENCY</sub>	基底器件中的故障音调延迟	从器件接收到音调到同一器件检测到并使 NFAULT 生效		24		µs
<b>主 ADC 和 AUX ADC 时序</b>						
t <sub>SAR_CONV</sub>	单次转换时间 (主 ADC 和 AUX ADC)			8		µs
t <sub>MAIN_ADC_CYCLE</sub>	单个轮询周期 (主 ADC)			192		µs
t <sub>AUX_ADC_CYCLE</sub>	单个轮询周期 (AUX ADC)			192		µs
t <sub>AFE_SETTLE</sub>	器件从 SLEEP 或 SHUTDOWN 模式进入 ACTIVE 模式时的模拟前端 (电平移位器) 趋稳时间			4		ms
t <sub>ADC_ACC</sub>	这包括多路复用器轮询、ADC 转换和数字滤波器。		-1.5		1.5	%
<b>均衡时序</b>						
t <sub>BAL_ACC</sub>	均衡计时器精度		-5		5	%
<b>硬件比较器/保护器时序</b>						
t <sub>OV_CYCLE</sub>	OV 轮询周期			8		ms
t <sub>UV_CYCLE</sub>	UV 轮询周期			8		ms
t <sub>OVUV_BIST_CYCLE</sub>	OV 和 UV BIST 周期		21.8	23	24.2	ms
t <sub>OT_CYCLE</sub>	OT 轮询周期			4		ms
t <sub>UT_CYCLE</sub>	UT 轮询周期			4		ms
t <sub>PWR_BIST_CYCLE</sub>	在执行电源 BIST GO 命令之后电源 BIST 完成所需的时间		10.9	11.5	12.1	ms
t <sub>OTUT_BIST_CYCLE</sub>	OT 和 UT BIST 周期		19	20	21	ms
t <sub>HW_COMP_ACC</sub>	OV、UV、OT、UT 比较器计时精度		-5		5	%
<b>I/O 时序 (TX、RX、GPIO、NFAULT)</b>						
t <sub>RISE</sub>	上升时间	V <sub>CVDD</sub> > 最小 V <sub>CVDD</sub> , C <sub>LOAD</sub> = 150pF, GPIO 处于输出模式		12		ns
t <sub>FALL</sub>	下降时间 (不包括 NFAULT)	V <sub>CVDD</sub> > 最小 V <sub>CVDD</sub> , C <sub>LOAD</sub> = 150pF, GPIO 处于输出模式		7		ns
t <sub>FALL_NFAULT</sub>	NFAULT 上的下降时间	V <sub>CVDD</sub> > 最小 V <sub>CVDD</sub> , C <sub>LOAD</sub> = 150pF, R <sub>PULLUP</sub> = 10kΩ		100		ns
<b>UART 时序</b>						
UART <sub>BAUD</sub>	UART TX/RX 波特率			1		Mbps
UART <sub>ERR_BAUD(RX)</sub>	UART RX 波特率错误 - 对外部主机的要求		-1		1	%
UART <sub>ERR_BAUD(TX)</sub>	UART TX 波特率错误		-1.5		1.5	%
t <sub>UART(CLR)</sub>	UART 清除低电平时间		15		20	位周期
t <sub>UART(RX_HIGH)</sub>	在 COMM CLEAR 之后, 在发送新帧之前等待该时间		1			位周期
<b>OTP NVM 时序</b>						

## 5.6 时序要求 (续)

在 -40°C 至 125°C 自然通风条件下的工作温度范围内, VBAT = 9V 至 80V (除非另有说明)

参数		测试条件	最小值	标称值	最大值	单位
$t_{\text{CRC\_CUST}}$	在客户 OTP 空间上完成单个 CRC 检查周期的时间			175		$\mu\text{s}$
$t_{\text{CRC\_FACT}}$	在出厂 OTP 空间上完成单个 CRC 检查周期的时间			1.6		ms
<b>SPI 控制器时序</b>						
$f_{\text{SCLK}}$	SCL 频率		450	500	550	kHz
$t_{\text{HIGH}}, t_{\text{LOW}}$	SCL 占空比			50		%
$t_{\text{SS(HIGH)}}$	SS 高电平延迟时间。从寄存器写入高电平到 SS 引脚高电平的时间			4		$\mu\text{s}$
$t_{\text{SS(LOW)}}$	SS 低电平延迟时间。从寄存器写入低电平到 SS 引脚低电平的时间			4		$\mu\text{s}$
$t_{\text{SU(MISO)}}$	目标器件的 MISO 输入数据设置时间要求	MISO 在 SCLK 转换之前保持稳定	100			ns
$t_{\text{HD(MISO)}}$	MISO 输入数据保持时间	MISO 在 SCLK 转换之后保持稳定		0		ns
<b>振荡器</b>						
$f_{\text{HFO}}$	高频振荡器		31.52	32	32.48	MHz
$f_{\text{LFO}}$	低频振荡器		248.9	262	275.1	kHz

## 5.7 典型特性

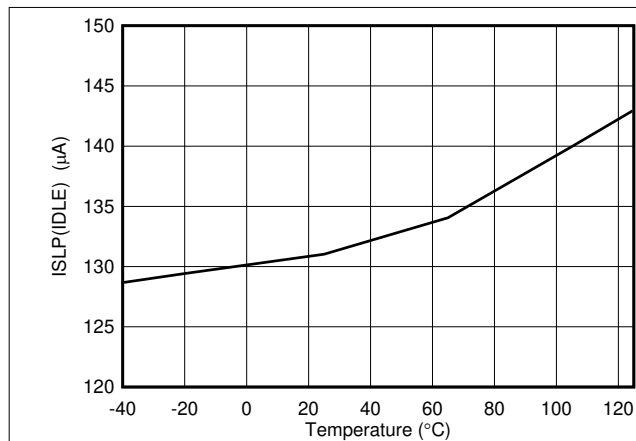


图 5-1. ISLP (IDLE) 与温度之间的关系

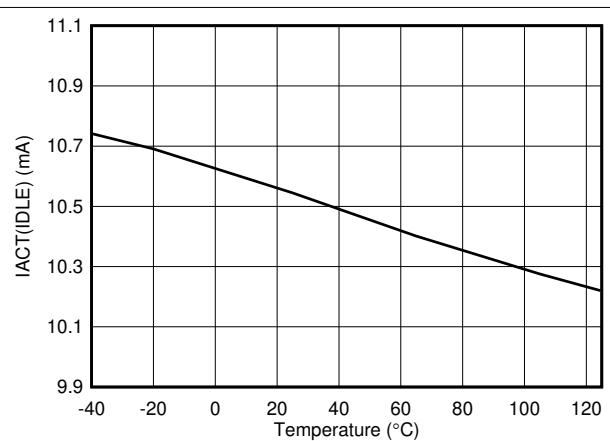


图 5-2. IACT (IDLE) 与温度之间的关系

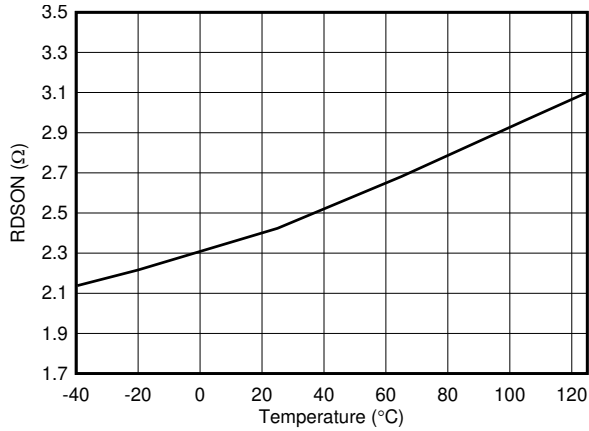


图 5-3. CBFET RDSON 与温度之间的关系

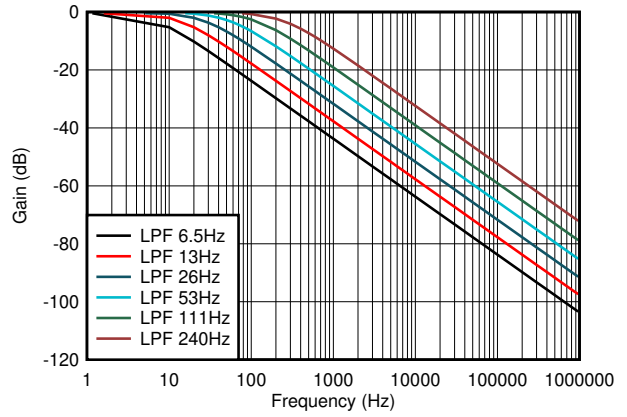


图 5-4. 数字低通滤波器

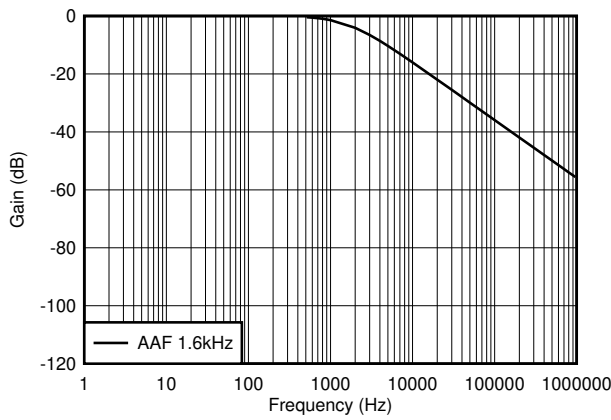


图 5-5. 主 ADC 滤波器

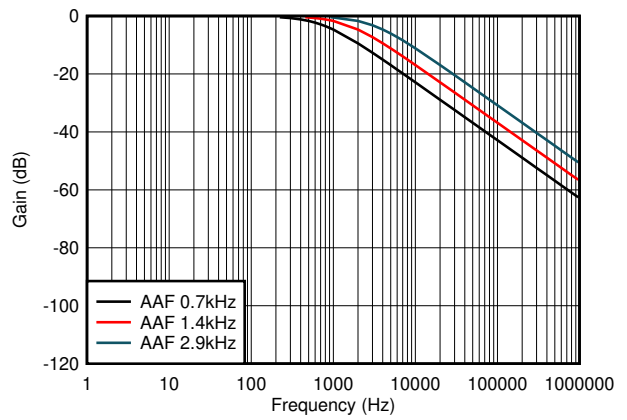


图 5-6. AUX ADC 滤波器

## 6 详细说明

### 6.1 概述

BQ79616 器件是一款可测量电芯电压和温度的可堆叠电池监测器。该器件支持 6 至 16 节串联 (6S 至 16S) 电池电芯。该器件允许使用电芯检测输入通道或专用汇流条通道进行多达三个汇流条连接和测量，从而最大限度地提高期间的灵活性，以支持各种电池模块尺寸。

多个器件可以通过菊花链方式连接在一起。每个器件都有一对高侧 (北侧) 和低侧 (南侧) 垂直差分通信端口，只需要一根双绞线电缆。该器件支持仅电容、电容和扼流圈或变压器隔离。每个菊花链器件上的通信都会重新计时，从而确保长距离通信的完整性。该设计支持可选的环形连接，可在电缆发生故障时改变菊花链通信方向。每个器件都包含一个通过 GPIO 配置的 SPI 控制器。

菊花链器件中的 ADC 可配置为对齐电芯电压测量的开始，并且可在 128  $\mu$ s 内测量所有电芯电压。每个电芯检测通道都包含一个后 ADC 数字低通滤波器 (LPF)，用于降噪，并提供移动平均测量结果。该器件具有八个 GPIO，所有这些 GPIO 均可配置用于 NTC 热敏电阻连接或用作通用 I/O。全部八个 GPIO 均可在 1.6ms 内进行测量。

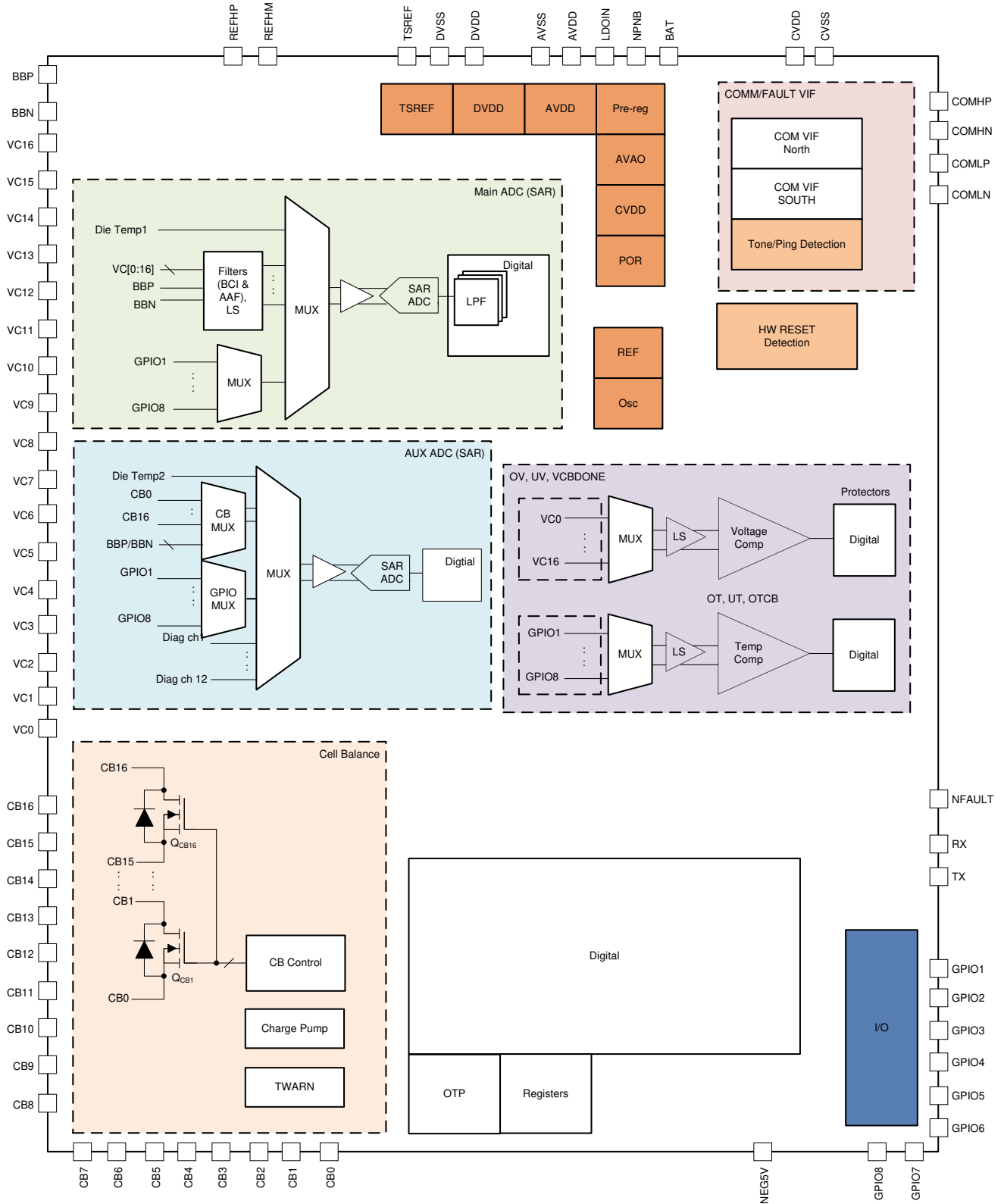
该器件通过用于每个电芯的内部电芯均衡 MOSFET (CBFET) 支持被动均衡。均衡功能自主运行，无需微控制器 (MCU) 交互。BQ79616 包含一个选项，用于根据外部热敏电阻检测到的可编程阈值或在内核温度过高 (高于 105°C) 时暂停均衡，然后再恢复均衡。均衡开始后，器件会跟踪每个电芯的均衡时间。MCU 可以随时读取剩余的均衡时间。

该器件包含一个硬件 OVUV 比较器和一个具有用户可配置阈值的 OTUT 比较器。这些比较器可用作独立于 ADC 测量的电芯过压和欠压以及热敏电阻过热和欠温检测的二级保护器。

该器件提供了将故障状态信息嵌入到通信帧中的选项。当菊花链中的器件检测到故障情况时，该信息将被嵌入并沿着通信响应帧传输到基底器件，该基底器件可以配置为触发 NFAULT 引脚作为系统的中断信号。这提供了一种减少通信开销的方法，无需添加额外的双绞线电缆和隔离，从而实现更快的故障检测。

该器件具有 SLEEP 和 SHUTDOWN 模式，用于降低功耗。所有功能都可以在 ACTIVE 模式下正常运行，均衡以及 OVUV 和 OTUT 硬件比较器也可以在 SLEEP 模式下正常运行。处于 SHUTDOWN 模式时，所有有效功能均关闭。提供了硬件复位功能，可以由主机 MCU 激活该功能。硬件复位为器件提供类似 POR 的事件，而无需实际移除电池。这为提高整体系统稳健性提供了一种可靠、低成本且可恢复的方式。

## 6.2 功能方框图



## 6.3 特性说明

### 6.3.1 电源

该器件直接从电池组生成其运行所需的所有电源。以下各节概述了每个内部电源块。有关建议的元件连接，请参阅节 7。有关电源块上的诊断控制和故障检测，请参阅节 6.3.6.4。

#### 6.3.1.1 AVAO\_REF 和 AVDD\_REF

AVAO\_REF 块（模拟电压始终开启）由 BAT 引脚供电。该引脚为所有功耗模式所需的常开低电流电路供电。该块还生成一个预稳压基准 AVAO\_REF。AVAO\_REF 电压会通过由 SHUTDOWN 模式控制的负载开关。该负载开关之后的基准电压为 AVDD\_REF。

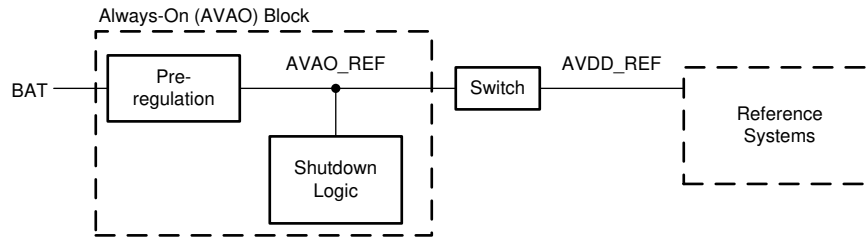


图 6-1. AVAO 块

#### 6.3.1.2 LDOIN

该器件由电池模块供电，该电池模块中每个电芯的电流消耗相同。该器件从电池模块的顶部通过内部线性稳压器和外部 NPN 晶体管在 LDOIN 引脚上生成 6V 稳定的电压（标称值）。NPNB 引脚控制稳压器的外部 NPN 晶体管。LDOIN 输出是其余内部低压降稳压器（LDO）的预稳压输入。在 OTP（一次性可编程）存储器编程期间，LDOIN 引脚将被稳压至 8V（标称值），以在内部为 OTP 编程提供编程电压。仅在硬件复位或 POR 事件期间关闭 LDOIN。

#### 6.3.1.3 AVDD

AVDD LDO 是模拟电路的电源。它从 LDOIN 获取输入电压并生成标称 5V 电压。它不会用于为任何外部电路供电。该 LDO 在 SHUTDOWN 模式、硬件复位或 POR 事件期间断电。

#### 6.3.1.4 DVDD

DVDD LDO 是数字电路的电源。它从 LDOIN 获取输入电压并生成标称 1.8V 电压。它不会用于为任何外部电路供电。该 LDO 在 SHUTDOWN 模式、硬件复位或 POR 事件期间断电。

#### 6.3.1.5 CVDD 和 NEG5V

CVDD LDO 是菊花链接口（或垂直接口 VIF）和 I/O 引脚（RX、TX、NFAULT 和 GPIO）的电源。它从 LDOIN 获取输入电压并生成标称 5V 电压。除了为内部使用提供电源外，该 LDO 还可以在 ACTIVE 和 SLEEP 模式下支持额外的 10mA 外部负载，而在 SHUTDOWN 模式下支持额外的 5mA 外部负载。

有一个 -5V 电荷泵用于菊花链接口（或垂直接口 VIF）和主 ADC 块。NEG5V 引脚具有 -4.6V 输出（标称值）。当器件处于 SLEEP 或 SHUTDOWN 模式时，它将处于低功耗突发模式。

#### 6.3.1.6 TSREF

TSREF 是一个 5V 缓冲基准，可以偏置外部热敏电阻电路，从而允许 ADC 测量温度并允许 OTUT 保护器检测温度故障。该基准可通过主 ADC 进行测量。主 ADC 测量的 TSREF 和 GPIO 采用比例式测量，以实现最佳温度测量。

TSREF 能够提供高达  $I_{TSREF\_ILIMIT}$  的电流，不用于为除热敏电阻偏置之外的任何外部电路供电。TSREF 默认处于关闭状态，可通过 `CONTROL2[TSREF_EN]` 位启用或禁用。TSREF 的启动时间由外部电容决定。MCU 可确保在进行任何 GPIO 测量或 OTUT 保护器检测之前 TSREF 保持稳定。启用 TSREF LDO 后，用户应等待 1.35ms，然后再发送下一条命令。

### 6.3.2 测量系统

器件中有两个 SAR ADC，即一个 16 位主 ADC 和一个 14 位 AUX ADC；两者都使用精密基准 (REFH) 进行高精度测量。每个 ADC 都有自己的独立控制功能，可以单独启用或禁用。主 ADC 通过连接到 GPIO 的热敏电阻对电芯电压 (VCELL) 和温度进行主要测量。它还提供 TSREF 和内核温度测量。AUX ADC 主要用于诊断过程，例如对 OVUV 和 OTUT 比较器的内部基准电压或 DAC 输出进行测量。它用于通过 GPIO 对电芯电压输入和热敏电阻温度输入进行冗余测量。

以下各节概述了主 ADC 和 AUX ADC 测量路径。有关建议的外部元件连接，请参阅节 7。有关该块的诊断控制功能和状态，请参阅节 6.3.6.4。

#### 6.3.2.1 主 ADC

一共有 24 个输入 (时隙) 多路复用到主 ADC (图 6-2)。所有输入都以轮询方式进行测量 (图 6-3)。测量每个输入需要 8 μs (标称值)，单个轮询周期在 192 μs (标称值) 内完成。主 ADC 的输入为：

- 内核温度 1
- TSREF
- 通过差分 VC<sub>n-1</sub> 至 VC<sub>n</sub> 实现的 Cell1 至 Cell16 电压，其中 n = 1 至 16
- 通过差分 BBP - BBN 引脚实现的汇流条输入
- 多路复用 GPIO1 至 GPIO8
- 备件 (RSVD)

所有测量值均以 16 位十六进制二进制补码形式报告。结果会报告给相应的 \*\_HI (高字节) 和 \*\_LO (低字节) 寄存器。首先将十六进制结果转换为十进制值。按照表 6-1 中的公式将结果转换为 μV 或 °C。

启用主 ADC 后，表 6-1 中显示的所有主 ADC 相关结果寄存器都将重置为默认值 0x8000。当主 ADC 随着轮询周期进行转换时，测量结果将填充到结果寄存器中。当 MCU 读取 \*\_HI 寄存器时，器件将暂停到对相关 \*\_LO 寄存器的数据刷新，直到读取该 \*\_LO 寄存器。

表 6-1. 主 ADC 测量转换公式

主 ADC 输入	结果寄存器	转换公式
内核温度 1	DIETEMP1_HI/LO	以 °C 表示的结果 = 以十进制表示的 V <sub>LSB_MAIN_DIETEMP1</sub> * 结果 0x0000h 以 0°C 为中心。
TSREF	TSREF_HI/LO	以 μV 表示的结果 = 以十进制表示的 V <sub>LSB_TSREF</sub> * 结果
Cell1 至 Cell16	VCELL*_HI/LO, 其中 * = 1 至 16	以 μV 表示的结果 = 以十进制表示的 V <sub>LSB_ADC</sub> * 结果
汇流条	BUSBAR_HI/LO	以 μV 表示的结果 = 以十进制表示的 V <sub>LSB_BB</sub> * Result
GPIO1 至 GPIO8	GPIO*_HI/LO, 其中 * = 1 至 8	以 μV 表示的结果 = 以十进制表示的 V <sub>LSB_GPIO</sub> * 结果

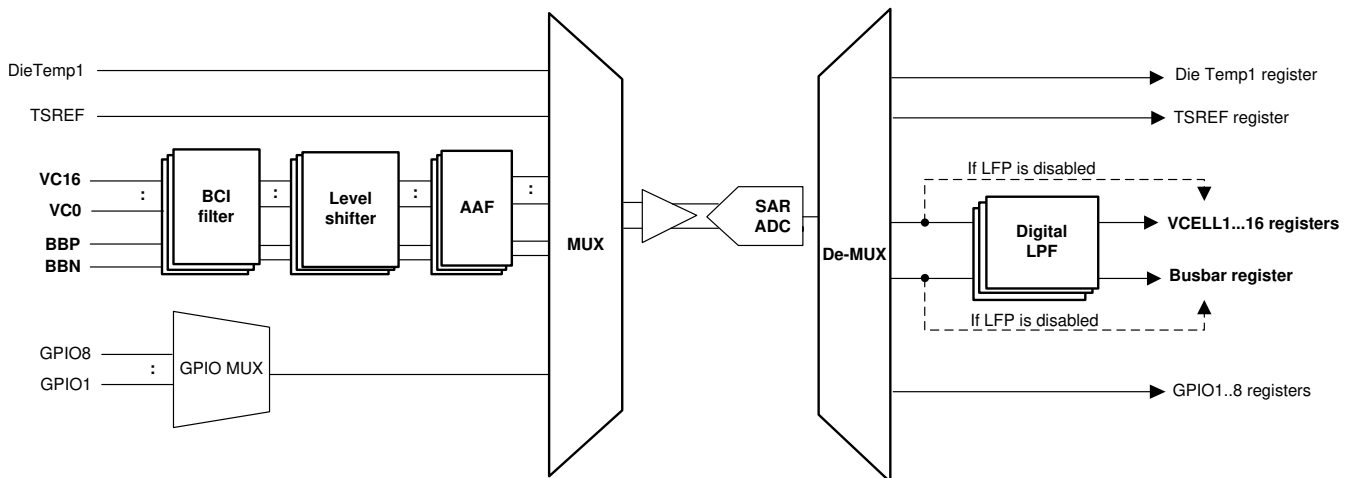


图 6-2. 主 ADC 测量路径

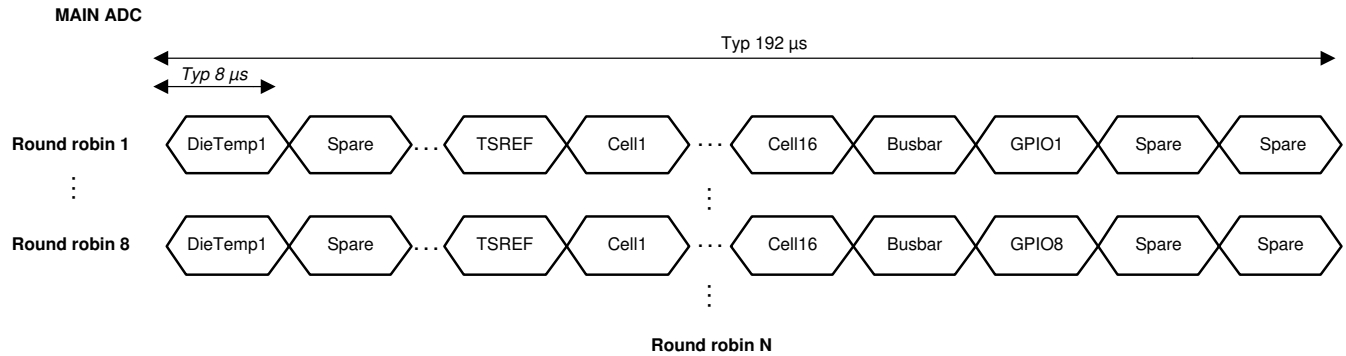


图 6-3. 主 ADC 轮询测量

### 6.3.2.1.1 电池电压测量

#### 6.3.2.1.1.1 模拟前端

主 ADC 的电芯电压测量通过 VC0 至 VC16 引脚进行。该器件允许测量最少 6 个电芯，最多 16 个电芯。VC0 至 VC16 引脚连接到模拟前端，模拟前端由每个 VC 输入通道上的 BCI 滤波器、电平转换器和抗混叠滤波器 (AAF) 组成。BCI 滤波器的截止频率 ( $f_{\text{cutoff}}$ ) 为 100kHz，AAF 的  $f_{\text{cutoff}}$  为 1.6kHz。这会在进入高压多路复用器并由主 ADC 测量之前滤除 VC 输入上的高频噪声。在 SLEEP 和 SHUTDOWN 模式下，电平转换器块会关闭以省电。

#### 6.3.2.1.1.2 VC 通道测量

VC 引脚是主 ADC 的电芯电压测量的输入通道，在轮询的 Cell1 至 Cell16 时隙中进行测量。即使连接至器件的电芯少于 16 个，轮询时序也始终相同 (图 6-4)。也就是说，对于无效 (或未使用) 的 VC 通道，器件会忽略相应的电芯时隙，但不会从轮询周期中删除该时隙。无论电芯数量配置如何，这都可以保持一致的测量时序。它还还为后 ADC 数字 LPF 输入提供一致的采样时间。

为了确定用于 ADC 测量的有效 VCELL 通道数，`ACTIVE_CELL[NUM_CELL3:0]` 参数设置最高的有效通道数。器件假设低于该设置的任何 VC 通道也有效。例如，当 14S 连接到器件时，MCU 将 `[NUM_CELL3:0]` 设置为 14S，主 ADC 忽略通道 15 和通道 16 测量，在通道 1 至 14 上进行测量。

测量结果在相应的 `VCELL*_HI` (高字节) 和 `VCELL*_LO` (低字节) 寄存器中报告，其中 \* = 1 至 16。如果禁用数字 LPFS，则结果寄存器将报告单个 ADC 转换值；否则，结果寄存器将报告经滤波的测量值。对于无效的 VC 通道，相应的 `_HI` 和 `_LO` 寄存器保留默认值 0x8000。

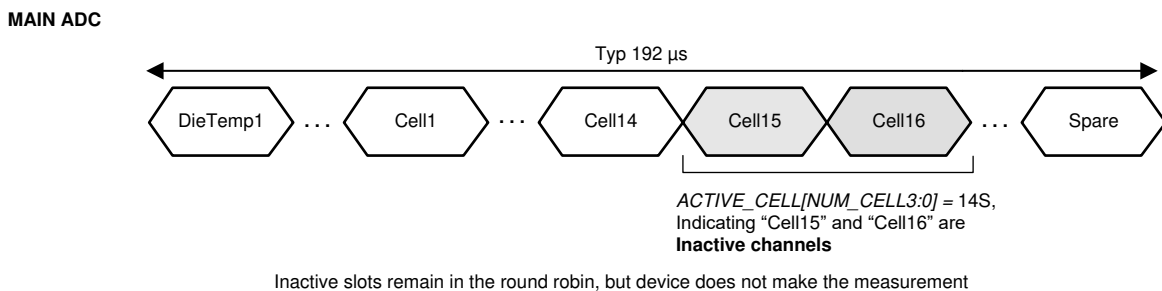


图 6-4. 6S 至 16S 具有相同的轮询时序

#### 6.3.2.1.1.3 后 ADC 数字 LPF

每个差分 VC 通道测量都配备了后 ADC LPF。LPF 的截止频率 ( $f_{\text{cutoff}}$ ) 要低得多。有 7 个  $f_{\text{cutoff}}$  选项：6.5Hz、13Hz、26Hz、53Hz、111Hz、240Hz 和 600Hz，可通过 `ADC_CONF1[LPF_VCELL2:0]` 设置进行配置。选择  $f_{\text{cutoff}}$  值并通过设置 `ADC_CTRL1[LPF_VCELL_EN] = 1` 启用 LPF 后，相同的  $f_{\text{cutoff}}$  设置适用于所有 VC 通道测量。

数字 LPF 作为单极滤波器实现，其响应与模拟 RC 电路非常相似。这意味着主 ADC 将以连续模式运行，以便数字 LPF 产生有效的滤波结果。

当输入直流电压电平出现阶跃变化时，MCU 应考虑数字滤波器稳定时间。下面的公式给出了达到 VC 电压阶跃稳定精度阈值的数字滤波器稳定时间典型估算方法。

数字滤波器稳定时间  $\sim [ ( \log_{10} ( \text{稳定精度阈值 [mV]} / \text{输入电压中的电压阶跃 [mV]} ) ) / \log_{10} ( 1 - \text{滤波器系数} ) ] \times 0.192\text{ms}$

Fcutoff (Hz)	600	240	111	53	26	13	6.5
滤波器系数	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.007813

例如：如果 VC 阶跃为 15mV，则对于 26Hz LPF 设置，用户必须实现约 27ms 的稳定时间才能处于输入阶跃的 1LSB 以内。

当 LPF 启动时，从禁用状态变为启用状态，它会跳转到其第一个输入值并从该点开始滤波。与从 0V 或某个中间电平电压开始相比，该实现可以在主 ADC 和 LFP 刚启动时提供很短的稳定时间。

#### 6.3.2.1.1.4 BBP 和 BBN 测量

BBP 和 BBN 引脚是来自主 ADC 的汇流条测量的输入。BB 通道的目的是使系统能够与一个电芯共享连接到单个 VC 通道的汇流条，如图 6-5 中的示例所示。因此，与 VC 输入类似，BBP/N 输入的前端也具有 BCI、电平转换器和 AAF 滤波器。差分 BB 通道测量还具有通过后 ADC 数字 LPF 的选项。通过使用不同的配置和使能控制 `ADC_CONF1[LPF_BB2:0]` 和 `ADC_CTRL1[LPF_BB_EN]`，具有与 VC 通道相同的  $f_{\text{cutoff}}$  选项。

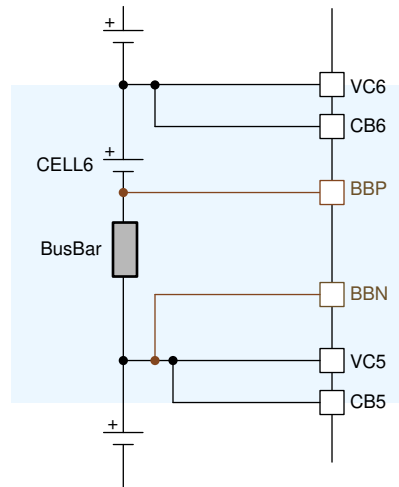


图 6-5. 简化的 BBP 和 BBN 连接

在 `BUSBAR_HI` (高字节) 和 `BUSBAR_LO` (低字节) 寄存器中报告 BB 通道测量值。如果禁用数字 LPF，则使用单个 ADC 转换值报告结果寄存器；否则，在滤波后的测量值中报告结果。在图 6-5 中，为了获得实际的 Cell6 测量值，MCU 会获取 (`VCELL6_HI/LO` 测量值 - `BUSBAR_HI/LO` 测量值) 差值。如果未使用 BBP 和 BBN 引脚 (悬空)，则 `BUSBAR_HI/LO` 寄存器值无意义。MCU 将忽略这些寄存器值。

#### 6.3.2.1.2 温度测量

##### 6.3.2.1.2.1 DieTemp1 测量

有 2 个内核温度传感器：DieTemp1 和 DieTemp2。DieTemp1 路由至主 ADC，并且还用于内部主 ADC 增益和失调电压校正。测量结果在 `DIETEMP1_HI` (高字节) 和 `DIETEMP1_LO` (低字节) 寄存器中报告。0°C 测量值以十六进制值 0x0000h 为中心，因此正值表示正温度，负值表示负温度。测量限值也为 +200°C 和 -100°C。

### 6.3.2.1.2.2 GPIO 和 TSREF 测量

有八个 GPIO。所有 GPIO 输入均可用于热敏电阻连接以进行温度测量，并可用作简单的单端电压输入测量。

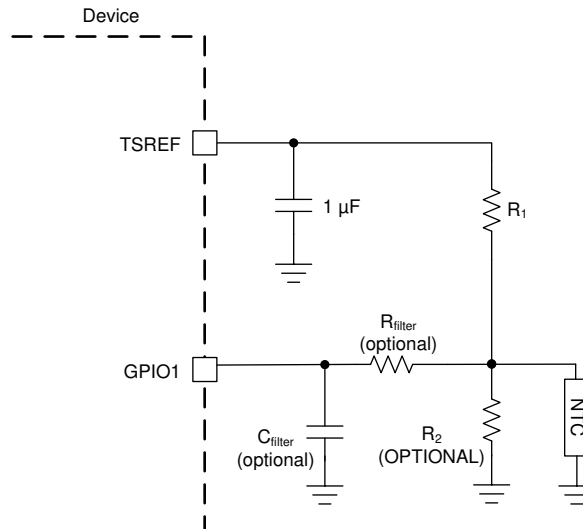


图 6-6. 热敏电阻连接

图 6-6 展示了启用 GPIO 进行热敏电阻测量时的热敏电阻电路。MCU 通过设置  $CONTROL2[TSREF\_EN] = 1$  确保 TSREF 已启用，并在获取测量值之前稳定。

GPIO 被多路复用至其中一个主 ADC 多路复用器输入。也就是说，在单个轮询周期中，仅测量一个 GPIO。要完成全部八个 GPIO 测量，需要八个轮询周期。

要启用 GPIO 进行 ADC 测量，相应的  $GPIO\_CONF_n[GPIO*2:0]$  (其中  $n = 1$  至  $4$ ， $*$  =  $1$  至  $8$ ，表示对应的 GPIO) 寄存器应配置为 ADC 输入或 ADC 和 OTUT 输入。例如，要启用 GPIO1 仅用于 ADC 测量，请将  $GPIO\_CONF_1[GPIO12:0]$  设置为 ADC 输入。有关更多详细信息，请参阅节 6.3.5。如果没有为任何 ADC 测量配置 GPIO，器件将忽略相应的 GPIO 时隙，但不会从轮询周期中删除该时隙。有关 GPIO2 配置为非 ADC 测量时的示例，请参阅图 6-7。

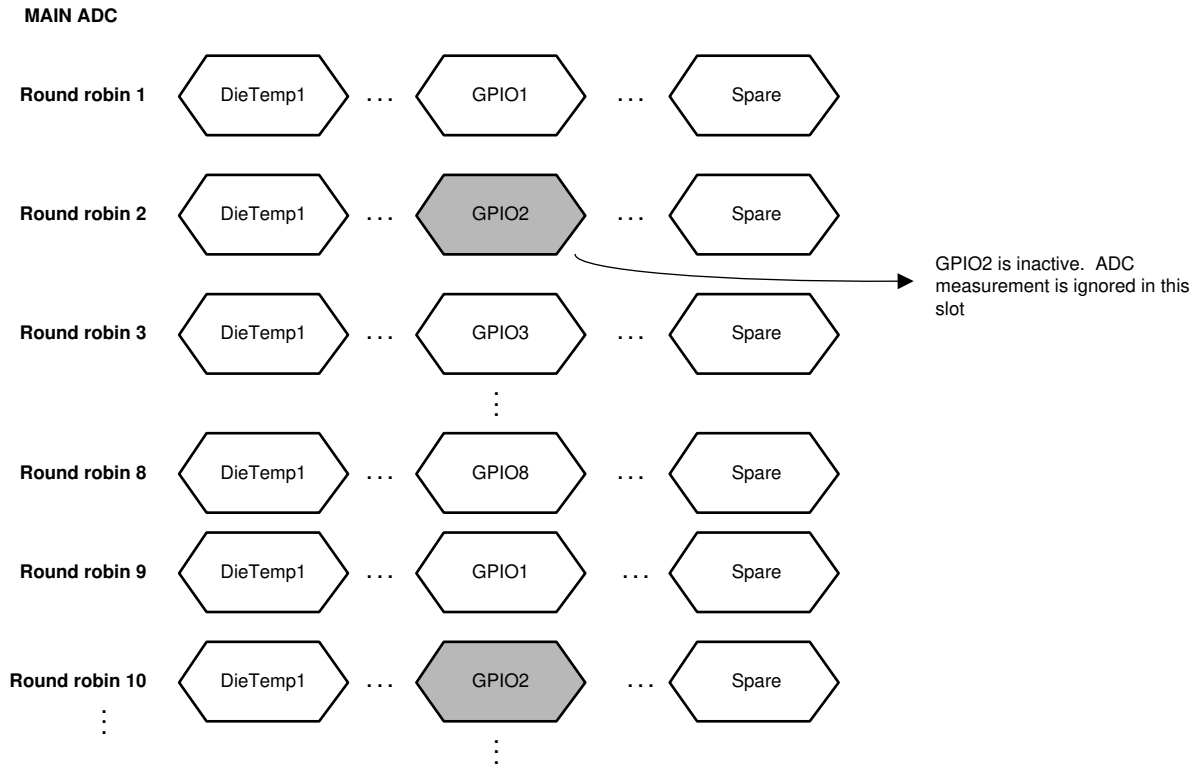


图 6-7. GPIO2 未配置用于 ADC 测量

测量结果在相应的  $GPIO*_HI$  (高字节) 和  $GPIO*_LO$  (低字节) 寄存器中报告, 其中  $* = 1$  至  $8$ 。测量结果以  $\mu V$  为单位。为了获得更高的温度精度, MCU 可以通过使用 TSREF 和 GPIO 测量来使用比例式测量, 公式如下:  $(GPIO\_ADC/TSREF\_ADC) = RNTC/(RNTC + R1)$ , 其中

- $GPIO\_ADC$  = GPIO 上的 ADC 测量值
- $TSREF\_ADC$  = TSREF 上的 ADC 测量值
- $RNTC$  = NTC 热敏电阻的阻值
- $ACTIVE\_CELL$  寄存器: 确定无效的 VC 通道并将结果寄存器保持为默认值  $0x8000$ 。
- $R1$  是如图 6-6 所示的上拉电阻器, 同时假设未使用  $R2$ 。

对于无效的 GPIO 通道, 相应的  $_HI$  和  $_LO$  寄存器保留默认值  $0x8000$ 。

### 6.3.2.1.3 主 ADC 运行控制

#### 6.3.2.1.3.1 运行模式和状态

为了启动主 ADC, 主机 MCU 设置  $ADC\_CTRL1[MAIN\_GO] = 1$ 。当器件接收到 GO 命令时, 它首先对以下设置进行采样, 以确定主 ADC 配置, 然后相应地运行主 ADC。要对以下设置进行任何更改, 都需要 MCU 重新发送另一条 GO 命令以实现新设置。

- $ADC\_CTRL1[MAIN\_MODE1:0]$ : 三种运行模式。有关详细信息, 请参阅表 6-2。
- $ADC\_CTRL1[LFP\_VCELL\_EN]$ : VC 通道的 LPF。如果启用, 则设置为  $ADC\_CONF1[LFP\_VCELL2:0] f_{cutoff}$ 。
- $ADC\_CTRL1[LFP\_BB\_EN]$ : BB 通道的 LPF。如果启用, 则设置为  $ADC\_CONF1[LFP\_BB2:0] f_{cutoff}$ 。
- $ADC\_CONF2[ADC\_DLY5:0]$ : 延迟主 ADC 的启动。用于在菊花链器件之间对齐 ADC 启动时间。
- $ACTIVE\_CELL$  寄存器: 确定无效的 VC 通道并将结果寄存器保持为默认值  $0x8000$ 。
- $GPIO\_CONF1$  至  $GPIO\_CONF4$ : 确定无效的 GPIO 通道并将结果寄存器保持为默认值  $0x8000$ 。
- $MAIN\_ADC\_CAL1$ 、 $MAIN\_ADC\_CAL2$ 、 $CS\_ADC\_CAL1$ 、 $CS\_ADC\_CAL2$ 、 $ADC\_CTRL1[CS\_DR]$  寄存器

### 备注

当在启用 LPF 滤波器的情况下使用 MAIN ADC 且需要 ADC 复位时，务必在再次运行 MAIN ADC 之前将 LPF\_VCELL\_EN 位、LPF\_BB\_EN 位和 MAIN\_GO 位设置为 0，然后再次设置为 1，因为需要重新初始化内部 LPF 缓冲区。如果省略该过程，则在接下来激活 MAIN ADC 时可能会出现 LPF\_FAIL 状态位。

有两个用于指示主 ADC 状态的状态位：

- **DEV\_STAT[MAIN\_RUN]**：指示主 ADC 是否正在运行。
- **ADC\_STAT1[DRDY\_MAIN\_ADC]**：在至少完成八个轮询周期时设置，指示所有活动 GPIO 通道和所有其他主 ADC 输入至少完成一次测量。

表 6-2. 主 ADC 运行模式总结

[MAIN_MODE1:0]	运行模式	说明
0b00	停止主 ADC	停止主 ADC
0b01	8 RR 运行（八个轮询周期）	主 ADC 运行八个轮询周期，然后停止。这样可以对所有电芯电压和系统的所有 GPIO 输入进行单次测量。在运行模式下滤波测量无效。例如，当 MCU 在系统空闲状态期间定期唤醒时用作快速突发读取。
0b10	连续运行	主 ADC 在连续模式下运行，如果 [MAIN_MODE1:0] = 0b00 且发送了 GO 命令，则会停止。例如，如果启用了 LPF，则必须使用该模式。也用于诊断操作。

当器件进入 ACTIVE 模式时，会为 **ACTIVE\_CELL[NUM\_CELL3:0]** 中指定数量的通道启用电平转换器。每当器件进入 ACTIVE 模式或更改 [NUM\_CELL3:0] 设置时，MCU 应首先等待  $t_{AFE\_SETTLE}$  时间，然后再启动主 ADC。

主 ADC 仅在 ACTIVE 模式下运行。如果在器件进入 SLEEP 模式时 ADC 正在运行，主 ADC 将被“冻结”（即 ADC 停止，但器件仍记得运行状态）。当器件在没有任何数字复位事件的情况下返回 ACTIVE 模式时，主 ADC 将重新启动并从其“冻结前”状态继续运行。在这种情况下，电芯电压测量在  $t_{AFE\_SETTLE}$  时间内关闭，因为 ADC 的输入电压尚未稳定。MCU 可以忽略这些测量，或者在  $t_{AFE\_SETTLE}$  之后发送新的 GO 命令来重新启动主 ADC。

#### 6.3.2.2 AUX ADC

一共有 24 个输入（时隙）多路复用到 AUX ADC（图 6-8）。所有输入都以轮询方式进行测量（图 6-9）。测量每个输入需要  $8\ \mu\text{s}$ （标称值），单个轮询周期在  $192\ \mu\text{s}$ （标称值）内完成。AUX ADC 的输入为：

- 内核温度 2
- 多路复用差分  $CB_{n-1}$  至  $CB_n$ （AUXCELL1 至 AUXCELL16），其中  $n = 1$  至 16，差分汇流条输入通过 BBP 至 BBN 引脚。
- 其他测量：
  - BAT 引脚
  - REFL，内部基准
  - VBG2，内部带隙
  - VCM，主 ADC 上的共模电压
  - AVAO\_REF，常开模块基准
  - AVDD\_REF
  - 来自 OV 保护器的 OV DAC
  - 来自 UV 保护器的 UV DAC
  - 来自 UV 保护器的 VCBDONE DAC
  - 来自 OT 保护器的 OT 或 OTCB DAC
  - 来自 UT 保护器的 UT DAC
- 多路复用 GPIO1 至 GPIO8
- 备件 (RSVD)

所有测量值均以 16 位十六进制二进制补码形式报告。结果会报告给相应的 \*\_HI (高字节) 和 \*\_LO (低字节) 寄存器。它首先将十六进制结果转换为十进制值。按照表 6-3 中的公式将结果转换为  $\mu\text{V}$  或  $^{\circ}\text{C}$ 。

启用 AUX ADC 后，表 6-3 中显示的所有 AUX ADC 相关结果寄存器都将重置为默认值 0x8000。当 AUX ADC 随着轮询周期进行转换时，测量结果将填充到结果寄存器中。当 MCU 读取 \*\_HI 寄存器时，器件将暂停到对相关 \*\_LO 寄存器的数据刷新，直到读取该 \*\_LO 寄存器。

表 6-3. AUX ADC 测量转换公式

AUX ADC 输入	结果寄存器	转换公式
内核温度 2	DIETEMP2_HI/LO	以 $^{\circ}\text{C}$ 表示的结果 = 以十进制表示的 $V_{\text{LSB\_AUX\_DIETEMP2}}$ * 结果 注意：0x0000h 以 $0^{\circ}\text{C}$ 为中心。
多路复用 AUXCELL1 至 AUXCELL16 和 BB 通道	AUX_CELL_HI/LO, 当 CB 多路复用器锁定到单个通道时	以 $\mu\text{V}$ 表示的结果 = 以十进制表示的 $V_{\text{LSB\_ADC}}$ * 结果
BAT	AUX_BAT_HI/LO	以 $\mu\text{V}$ 表示的结果 = 以十进制表示的 $V_{\text{LSB\_AUX\_BAT}}$ * 结果
REFL	AUX_REFL_HI/LO	以 $\mu\text{V}$ 表示的结果 = 以十进制表示的 $V_{\text{LSB\_AUX\_DIAG}}$ * 结果
VBG2	AUX_VBG2_HI/LO	
VCM	AUX_VCM_HI/LO	
AVAO_REF	AUX_AVAO_REF_HI/LO	
AVDD_REF	AUX_AVDD_REF_HI/LO	
OV DAC	AUX_OV_DAC_HI/LO	
UV_DAC	AUX_UV_DAC_HI/LO	
VCBDONE DAC	AUX_VCBDONE_DAC_HI/LO	
OT 或 OTCD DAC	AUX_OT_OTCD_DAC_HI/LO	
UT DAC	AUX_UT_DAC_HI/LO	
多路复用 GPIO1 至 GPIO8	AUX_GPIO_HI/LO	以 $\mu\text{V}$ 表示的结果 = 以十进制表示的 $V_{\text{LSB\_GPIO}}$ * 结果

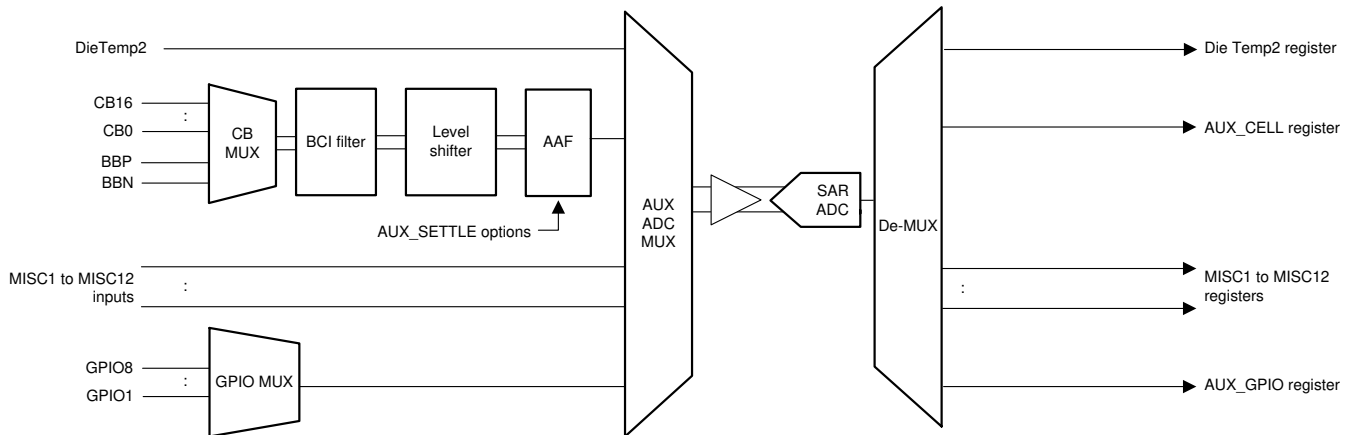


图 6-8. AUX ADC 测量路径

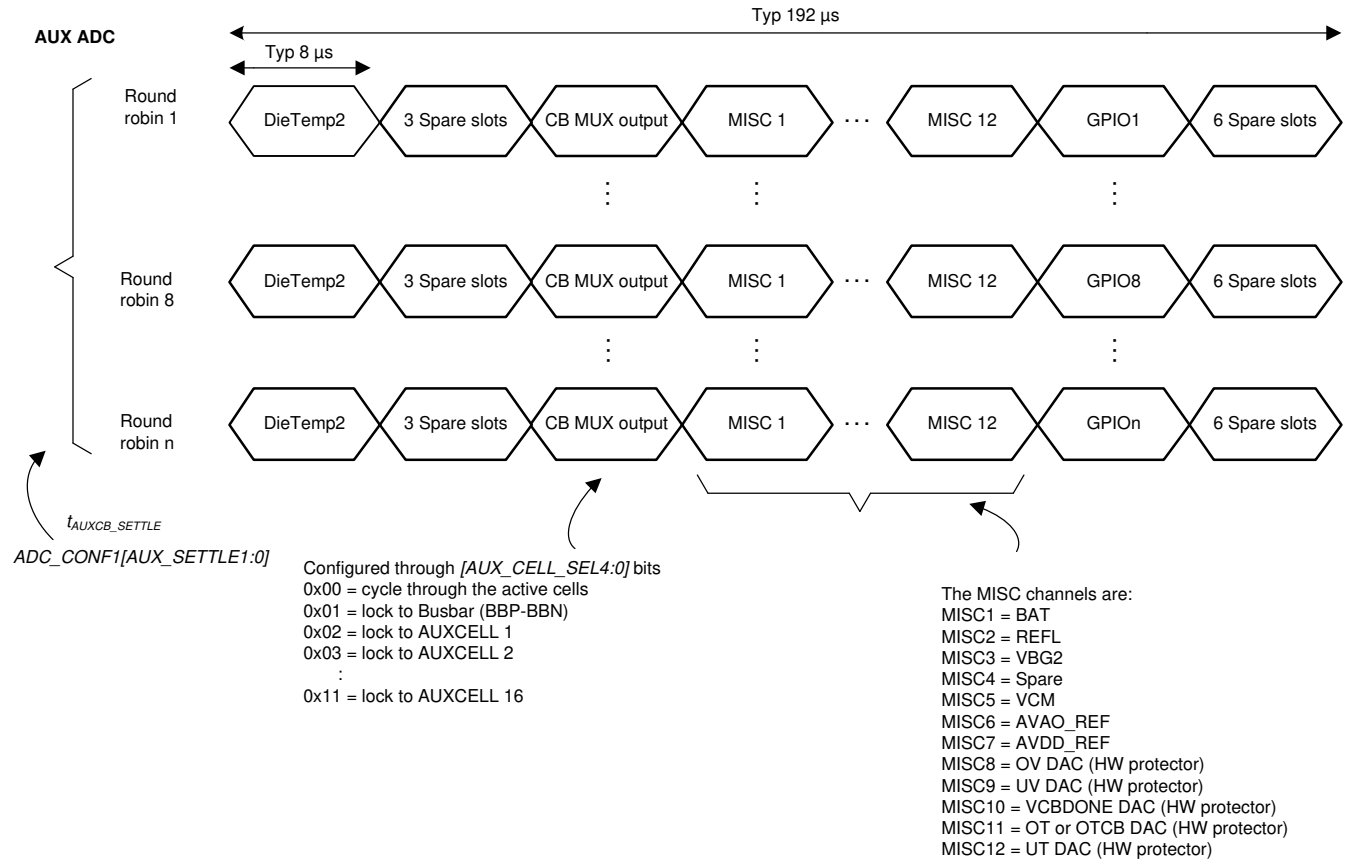


图 6-9. AUX ADC 轮询测量

### 6.3.2.2.1 AUX 电芯电压测量

#### 6.3.2.2.1.1 AUX 模拟前端

AUX ADC 路径充当有关电芯电压测量和汇流条测量的主 ADC 测量冗余路径。它还在 AUX ADC 路径中具有 BCI 滤波器和 AAF 滤波器等前端滤波器。AUX 路径中的 AUXCELL 通道和差分 BB 通道 (来自 BBP 和 BBN 引脚) 进行多路复用 (在图 6-8 中显示为 CB 多路复用器), 以共享单个 BCI 滤波器和 AAF 滤波器。前端滤波器之后的 CB 多路复用器输出进入 AUX ADC 多路复用器之一并进入 AUX ADC 进行测量。

由于前端滤波器是共享的, 因此器件必须等待 AAF 滤波器稳定, 然后才能进行任何有效的 CB 通道 (AUXCELL) 或 BBP 和 BBN 测量。与主 ADC 路径中一样, 默认 AAF  $f_{\text{cutoff}}$  为 1.6kHz, 这相当于完成单次 CB 或 BB 通道测量需要额外的 4.3ms 稳定时间。该器件提供 3 个 AAF 稳定时间选项, 即 4.3ms (默认值)、2.3ms 和 1.3ms, 由 ADC\_CONF1[AUX\_SETTLE1:0] 位配置。与主 ADC 路径中一样, BCI 滤波器  $f_{\text{cutoff}}$  为 100kHz。

#### 备注

为了通过 AUX ADC 实现最佳测量精度, 建议每次通过 AUX\_CELL\_SEL 位锁定新的 CB 通道时重置 ADC。这将确保重新运行共模误差校准例程, 从而对测量结果进行共模误差补偿。

#### 6.3.2.2.1.2 CB 和 BB 通道测量

一个时隙 (CB MUX 输出时隙) 在 AUX ADC 轮询周期中分配给 CB 通道 (差分  $CB_{n-1} - CB_n$ , 其中  $n = 1$  至 16) 和 BB (差分 BBP - BBN) 通道测量, 因为这些通道被多路复用为 AUX ADC 多路复用器的单个输入。对于单个 CB 或 BB 通道测量, 由于该器件还必须等待 AAF 稳定时间, 因此该器件需要多个轮询周期。

由于需要等待 AAF 稳定, AUX ADC 只会测量由 MCU 选择的活跃 CB 和 BB 通道; 非活跃或未选择的通道会被跳过。

活动 CB 通道由  $ACTIVE\_CELL[NUM\_CELL3:0]$  设置决定。这些位设置最高的活动通道编号。例如，当 14S 连接到器件时，MCU 将  $ACTIVE\_CELL[NUM\_CELL3:0]$  设置为 14S，器件假设 CB 通道 1 至 14 处于活动状态；CB 通道 15 和 16 处于非活动状态，将被 AUX ADC 跳过。

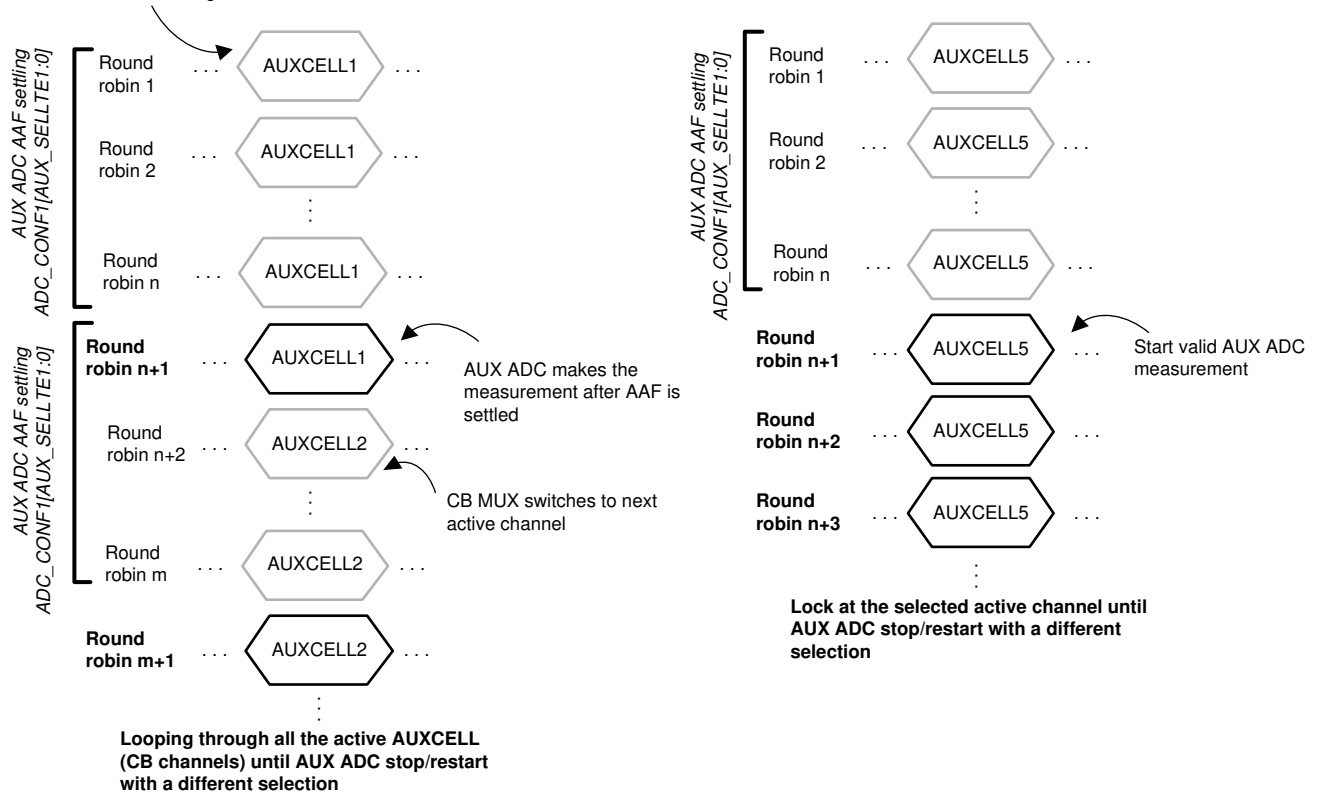
MCU 可以控制通过 AUX ADC 测量哪些 CB 和 BB 通道。 $ADC\_CTRL2[AUX\_CELL\_SEL4:0]$  提供了轮询所有活动 CB 通道和 BB 通道或锁定至单个 CB 通道或锁定至 BB 通道的选项。图 6-10 展示了如何使用不同的  $[AUX\_CELL\_SEL4:0]$  设置实现 AUXCELL 时隙的示例。

建议在连续模式下运行 AUX ADC，使用 AUX ADC 对所有活动 CB 通道进行一次测量。这使得器件能够减少 AUX ADC 测量中的共模误差。MCU 应在运行与 ADC 比较相关的诊断或锁定到单个 CB 或 BB 通道测量之前执行该过程。

AUX ADC 路径中没有后 ADC LPF。如果在诊断期间使用 AUX ADC 测量，则会将 AUX CELL (CB 通道) 测量值与主 ADC 预滤波测量值进行比较。当器件在内部执行 VCELL (来自主 ADC) 和 AUX CELL (来自 AUX ADC) 测量比较时，AUX BB 比较由主机执行。有关更多详细信息，请参阅节 6.3.6.4。

当  $[AUX\_CELL\_SEL4:0]$  位被设置为锁定在单个 CB (必须处于活动状态) 或 BB 通道上时，器件使 CB 或 BB 通道测量进入只读状态。测量结果在  $AUX\_CELL\_HI$  (高字节) 和  $AUX\_CELL\_LO$  (低字节) 寄存器中报告。结果寄存器将在 AAF 稳定时间之后更新。对于任何其他情况 (包括锁定到非活动 CB 通道)，结果寄存器将保留默认值 0x8000。

CB MUX stays at the selected channel for the AUX ADC AAF settling time, but the measurement during this time is discarded



(a)  $[AUX\_CELL\_SEL4:0] = \text{loop through all active CB channels}$

(b)  $[AUX\_CELL\_SEL4:0] = \text{Lock to CB channel 5 (AUXCELL5)}$

图 6-10. 具有不同  $[AUX\_CELL\_SEL4:0]$  设置的 CB 多路复用器输出时隙

### 6.3.2.2.2 AUX 温度测量

#### 6.3.2.2.2.1 DieTemp2 测量

有 2 个内核温度传感器：DieTemp1 和 DieTemp2。DieTemp2 路由至 AUX ADC，并且还用于内部 AUX ADC 增益和失调电压校正。测量结果在 *DIETEMP2\_HI*（高字节）和 *DIETEMP2\_LO*（低字节）寄存器中报告。0°C 测量值以十六进制值 0x00 为中心，因此正值表示正温度，负值表示负温度。测量限值也为 +200°C 和 -100°C。

#### 6.3.2.2.2.2 AUX GPIO 测量

AUX GPIO 路径与主 GPIO 路径相同。全部八个 GPIO 均多路复用至单个 AUX ADC 多路复用器输入。AUX ADC 轮询周期中只有一个 GPIO 时隙。也就是说，在单个 AUX ADC 轮询周期中，只会测量一个 GPIO。要完成全部八个 GPIO 测量，需要八个轮询周期。如果 GPIO 连接到热敏电阻网络，则 MCU 通过设置 *CONTROL2[TSREF\_EN]* = 1 来启用 TSREF，并确保在开始 AUX ADC 测量之前 TSREF 是稳定的。

启用 AUX ADC 后，第一个轮询周期中的 GPIO 时隙为 GPIO1，第二个轮询周期中的 GPIO 时隙为 GPIO3，依此类推。为了使 AUX ADC 对 GPIO 进行测量，必须在相应的 *GPIO\_CONF<sub>n</sub>[GPIO\*2:0]* 位中将 GPIO 配置为 ADC 输入或 ADC 和 OTUT 输入，其中对于相应的 GPIO 通道，n = 1 至 4，\* = 1 至 8。有关更多详细信息，请参阅节 6.3.5。如果 GPIO 对于 ADC 测量处于非活动状态，则器件会忽略相应的 GPIO 时隙，但不会从 AUX ADC 轮询周期中删除该时隙。

默认情况下，AUX ADC 循环遍历所有 GPIO 通道，并且测量结果不会向结果寄存器报告。但是，如果 MCU 锁定到单个 GPIO 通道，则锁定的 GPIO 测量结果将报告至 *AUX\_GPIO\*\_HI*（高字节）和 *AUX\_GPIO\*\_LO*（低字节）寄存器。可以通过 *ADC\_CTRL3[AUX\_GPIO\_SEL3:0]* 位设置该通道锁定。如果 *[AUX\_GPIO\_SEL3:0]* 锁定至单个 GPIO 通道，结果寄存器将报告 GPIO 测量，任何其他条件将显示默认值 0x8000。

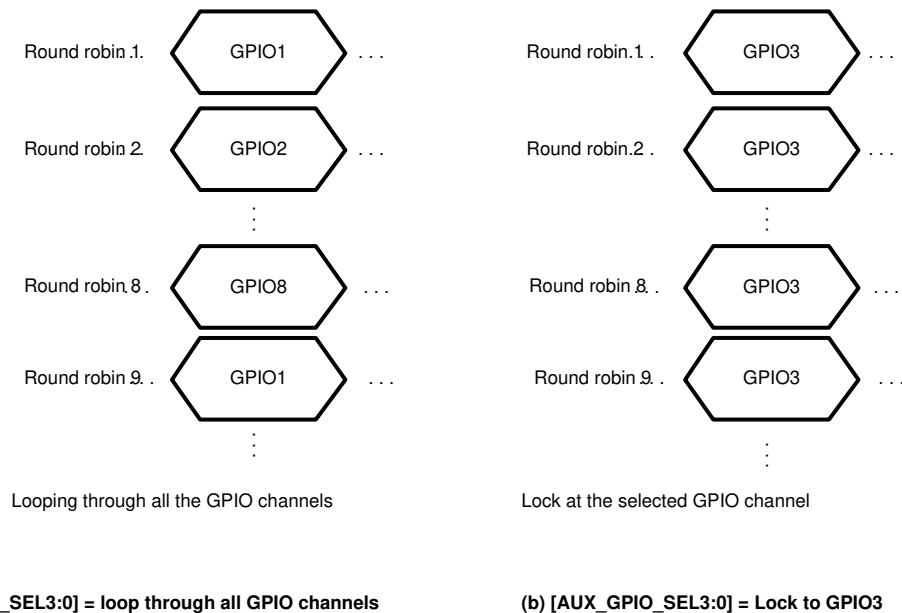


图 6-11. 具有不同 *[AUX\_GPIO\_SEL3:0]* 设置的 GPIO 时隙

#### 6.3.2.2.3 其他测量

AUX ADC 部分的开头列出了 12 个其他测量。当 AUX ADC 启用时，会在每个轮询周期中测量这些输入。表 6-3 展示了相应的结果寄存器。

OVUV 和 OTUT 保护器的 DAC 输入反映了器件的实时 DAC 值，该值显示了保护器中当前使用的 OVUV 和 OTUT 检测或恢复阈值。如果存在未使用的通道或者任何电芯或 GPIO 通道检测到故障，则观察到 DAC 测量值变化是正常的。有关保护器架构的说明，请参阅节 6.3.4，有关保护器 DAC 测量配置，请参阅节 6.3.6.4。

#### 6.3.2.2.4 AUX ADC 操作控制

为了启动 AUX ADC，主机 MCU 设置  $ADC\_CTRL3[AUX\_GO] = 1$ 。当器件收到 GO 命令时，首先对以下设置进行采样来确定 AUX ADC 配置，然后相应地操作 AUX ADC。无论对以下设置进行哪些更改，都需要 MCU 发送另一个 GO 命令来实现新设置。

- $ADC\_CTRL3[AUX\_MODE1:0]$ ：四种运行模式。有关详细信息，请参阅表 6-4。
- $ADC\_CTRL2[AUX\_CELL\_SEL4:0]$ ：选择由 AUX ADC 测量哪些 CB 通道。
- $ADC\_CONF1[AUX\_SETTLE1:0]$ ：配置 AUX ADC AAF 稳定时间。
- $ADC\_CTRL3[AUX\_GPIO\_SEL3:0]$ ：选择由 AUX ADC 测量哪些 GPIO 通道。
- $ACTIVE\_CELL$  寄存器：确定非活动的 CB 通道。
- $GPIO\_CONF1$  至  $GPIO\_CONF4$ ：确定非活动的 GPIO 通道。

有四个用于指示 AUX ADC 状态的状态位：

- $DEV\_STAT[AUX\_RUN]$ ：指示 AUX ADC 是否正在运行。
- $ADC\_STAT1[DRDY\_AUX\_MISC]$ ：当所有 MISC 输入至少被测量一次时设置。
- $ADC\_STAT1[DRDY\_AUX\_CELL]$ ：当  $[AUX\_CELL\_SEL4:0]$  选择的 CB 或 BB 通道至少被测量一次时设置。
- $ADC\_STAT1[DRDY\_AUX\_GPIO]$ ：当所有 GPIO 通道（活动或非活动）都被测量一次时设置。器件将忽略非活动通道测量。

表 6-4. AUX ADC 运行模式汇总

[AUX_MODE1:0]	运行模式	说明
0b00	停止 AUX ADC	停止 AUX ADC
0b01	单次运行（1 个轮询周期）	AUX ADC 运行一个轮询周期，然后停止。这将对所有 MISC 输入进行单次测量。例如，仅用作 MISC 输入的快速突发读取，无需向 AUX ADC 发出停止命令。
0b10	连续运行	AUX ADC 在连续模式下运行，如果 $[AUX\_MODE1:0] = 0b00$ 且发送了 GO 命令，则会停止。例如，当使用 ADC 诊断比较操作时必须使用该模式。有关详细信息，请参阅节 6.3.6.4。
0b11	8 RR 运行（八个轮询周期）	AUX ADC 运行八个轮询周期，然后停止。这将对所有活动的 GPIO 输入进行单次测量。

AUX ADC 仅在 ACTIVE 模式下运行。如果在器件进入 SLEEP 模式时 ADC 正在运行，AUX ADC 将被“冻结”；即 ADC 停止，但器件仍记得运行状态。当器件在没有任何数字复位事件的情况下返回 ACTIVE 模式时，AUX ADC 将重新启动并从中“冻结前”状态继续运行。

#### 6.3.2.3 主 ADC 测量与 AUX ADC 测量之间的同步

当  $ADC\_CTRL2$  寄存器中的  $AUX\_CELL\_ALIGN = 0x0$  时，器件会将 AUX 电芯测量（CB 多路复用器 - 时隙 5）与主电芯上的目标 VC 通道时隙对齐。DieTemp2 启动时没有任何延迟，AUX 电芯 CB 多路复用器时隙 5 动态移动以匹配选定的主电芯，其余 AUX ADC 时隙会相应地进行调整。这可确保主 VC 和 AUX CB ADC 采样之间没有时间偏差。

当  $AUX\_CELL\_ALIGN = 0x1$  时，动态对齐被禁用，AUX 电芯测量（CB 多路复用器时隙 5）始终与主 ADC 电芯 8 测量（主 ADC 时隙 12）对齐。



图 6-12. 主 ADC 和 AUX ADC 采样之间的同步

### 6.3.3 Cell Balancing

该器件在每个 CB 通道上集成了内部电芯均衡 MOSFET (CBFET)，以实现被动电芯均衡。均衡电流由电芯电压、与 CB 引脚串联的外部电阻器以及内部 CBFET  $R_{dson}$ 、 $R_{DSON}$  参数决定。以下公式用于计算相邻 CBFET 导通或不导通时的有效均衡电流。电芯均衡可以在 ACTIVE 或 SLEEP 模式下运行。

- 无连续 CBFET 导通时的均衡 (图 6-13 (a)) :  $I_{CB} = V_{Cell} / ((2 \times R_{CB}) + R_{dson_{QCB}})$
- 两个连续 CBFET 导通时的均衡 (图 6-13 (b)) :  $I_{CB} = (两个 V_{CELL} 之和) / ((2 \times R_{CB}) + R_{dson_{QCBn}} + R_{dson_{(QCBn-1)}})$

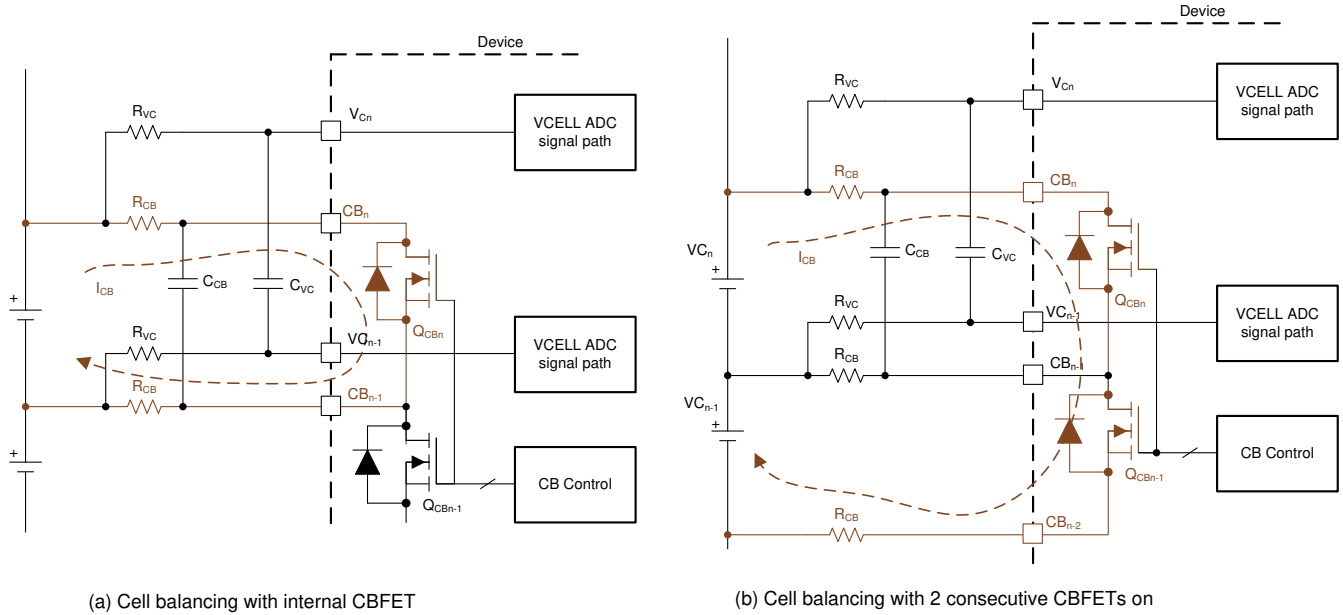


图 6-13. 内部电芯均衡和均衡电流的流动

### 6.3.3.1 设置电芯均衡

设置电芯均衡需要执行三个步骤。以下各节详细介绍了每个步骤。在开始电芯均衡之前，主机 MCU 按照以下步骤配置均衡控制。通过设置  $BAL\_CTRL2[BAL\_GO] = 1$  开始均衡。 $BAL\_STAT[CB\_RUN] = 1$  指示电芯均衡正在有效运行。请注意，在电芯均衡期间会绕过未由  $ACTIVE\_CELL[NUM\_CELL3:0]$  选择的通道。

1. 确定要启用哪个通道来实现电芯均衡。
2. 选择电芯均衡控制方法，即自动或手动均衡控制。
3. 确定附加控制配置：
  - a. 是否会启用基于热敏电阻测量的热管理？
  - b. 电芯均衡是否根据电芯电压停止？
  - c. 如果检测到任何未屏蔽的故障，电芯均衡是否会终止？

#### 6.3.3.1.1 步骤 1：确定均衡通道

该器件为每个通道提供单独的均衡计时器。均衡计时器是启动和停止通道上电芯均衡的主要控制设置。均衡计时器由  $CB\_CELL*\_CTRL$  寄存器配置，其中  $*$  = 1 至 16，对应于 CBFET 1 (CB 通道 1) 至 CBFET 16 (CB 通道 16)。这些寄存器中的非零值会设置相应的通道以进行均衡，但直到 MCU 发出  $BAL\_CTRL2[BAL\_GO] = 1$  后 CBFET 才会开启。当通道均衡计时器到期时，该通道上的电芯均衡会停止。电芯均衡还会在其他条件下停止，例如电芯电压低于特定的阈值、检测到未屏蔽的故障或主机强制停止。节 6.3.3.3 总结了电芯均衡停止条件。

#### 6.3.3.1.2 第 2 步：选择均衡控制方法

电芯均衡经过配置后会自动运行。可以使用  $BAL\_CTRL2[AUTO\_BAL]$  位通过两种方法来配置电芯均衡控制。

- 自动均衡控制 ( $[AUTO\_BAL] = 1$ )：通过该方法，主机 MCU 可以在任何通道上启用均衡。主机发送  $[BAL\_GO] = 1$  后，均衡启动，器件将自动以奇数和偶数方式设置所有启用的 CBFET 的占空比。占空比由  $BAL\_CTRL1[DUTY2:0]$  位配置。
  - 示例 1：MCU 设置全部 16 个通道以进行电芯均衡。

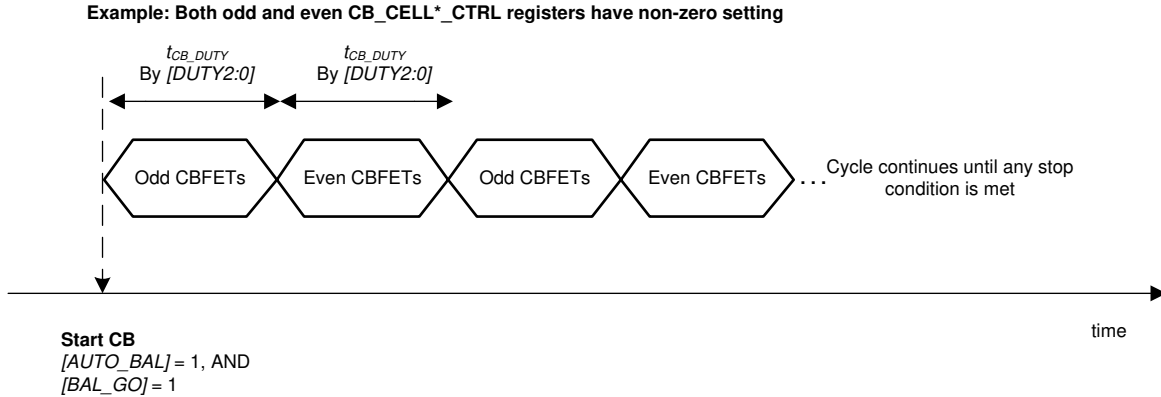


图 6-14. 自动均衡控制，示例 1

- 示例 2：MCU 仅设置奇数或偶数通道进行电芯均衡。 $BAL\_CTRL1[DUTY2:0]$  位设置无效，因为器件不会在奇数或偶数通道之间切换。

**Example: Odd CB\_CELL\*\_CTRL registers have non-zero value  
Even CB\_CELL\*\_CTRL registers are all zero**

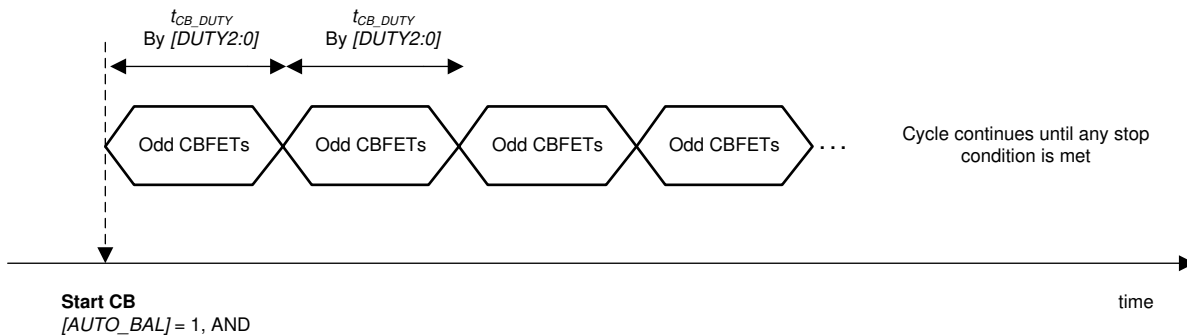


图 6-15. 自动均衡控制，示例 2

- 手动均衡控制 ( $[AUTO\_BAL]=0$ )：通过该方法，器件将在接收到  $[BAL\_GO]=1$  后开启具有非零均衡计时器设置的 CBFET。在电芯均衡期间不会进行奇数和偶数通道切换，并且在该控制下  $BAL\_CTRL1[DUTY2:0]$  设置不适用。主机 MCU 可以通过该方法启用两个连续的 CBFET，并且最多可以启用八个 CBFET。当两个连续的 CBFET 启用并且两个通道都连接到电池电芯时，与没有相邻的 CBFET 开启相比，均衡电流显著不同（图 6-13）。提供  $DEV\_CONF[NO\_ADJ\_CB]$  位是为了在不打算开启相邻通道以进行均衡的系统中避免无意启用相邻的 CBFET。在该控制方法中，器件依靠 MCU 来启用适当的通道。如果 MCU 发送  $[BAL\_GO]=1$ ，但 CBFET 在无效条件下启用，则器件不会开始均衡并且会设置  $BAL\_STAT[INVALID\_CBCONF]=1$ 。无效配置为以下任一配置：
  - 启用了八个以上的通道进行均衡（即超过 8 个  $CB\_CELL*_CTRL$  寄存器具有非零设置），
  - $DEV\_CONF[NO\_ADJ\_CB]=1$ ，但启用了相邻的通道进行均衡，
  - $DEV\_CONF[NO\_ADJ\_CB]=0$ ，但启用了两个以上的连续通道进行均衡：
    - 示例：启用 CBFET 1、2、4、5、7、10、12 和 14 有效。
    - 示例：启用 CBFET 1、2 和 3 无效。

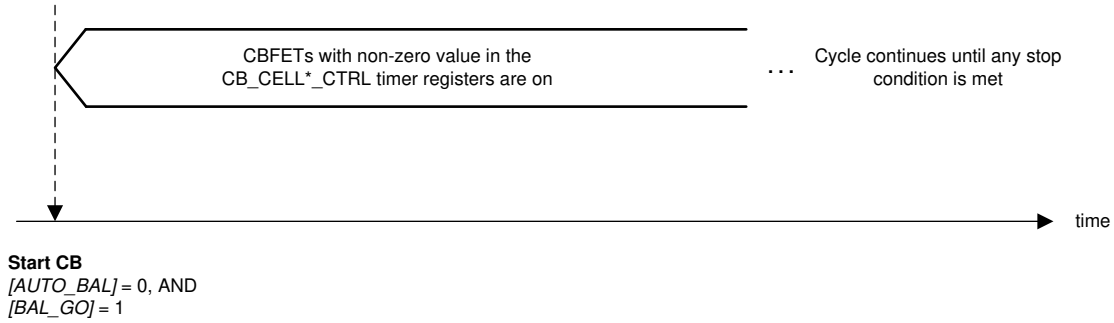


图 6-16. 手动均衡控制

### 6.3.3.1.3 步骤 3a : 均衡热管理

在进行被动均衡时，内部 CBFET 和外部均衡电阻器会产生热量。这会在 PCB、器件和均衡电阻器区域中产生两个热点。该器件可在 75°C 环境温度下支持高达 240mA 的电流。在较低的环境温度下可以支持更高的均衡电流。

该器件提供了两项热管理功能来避免芯片过热以及管理 PCB 温度。这两项功能都会监控温度（核心温度或热敏电阻温度），以便在温度超过暂停阈值时自动暂停均衡。当温度低于恢复阈值时，均衡会自动恢复。在电芯均衡暂停状态下，所有均衡计时器和均衡设置均为“冻结”，当器件退出暂停状态时，均衡将以相同的配置恢复。

- **CB TWARN 均衡暂停：**内部 CBFET 附近内置了内核温度传感器。发送  $[BAL\_GO] = 1$  后，这些温度传感器启用。如果任何传感器检测到内核温度高于  $T_{CB\_TWARN}$  阈值（标称值 105°C），则所有通道上的均衡都会暂停。器件会设置  $BAL\_STAT[CB\_INPAUSE] = 1$  和  $BAL\_STAT[OT\_PAUSE\_DET] = 1$ 。当所有传感器检测到内核温度低于  $(T_{CB\_TWARN} - T_{CB\_HYS})$  时，电芯均衡将在启用了均衡的通道上恢复。
- **热敏电阻 OTCB 均衡暂停：**为了管理由于外部均衡电阻器导致的热量增加，如果连接到 GPIO 的任何有源热敏电阻检测到温度高于  $OTCB\_THRESH[OTCB\_THR3:0]$  设置的阈值，器件可以选择暂停所有通道上的电芯均衡。触发 OTCB 检测后， $BAL\_STAT[CB\_INPAUSE] = 1$  且  $BAL\_STAT[OT\_PAUSE\_DET] = 1$ 。一旦所有有源热敏电阻检测到温度低于  $(OTCB\_THRESH[OTCB\_THR3:0] + OTCB\_THRESH[COOLOFF2:0])$  设置的恢复阈值，所有启用的通道上的均衡将恢复。OTCB 检测通过集成式 OT 保护器来执行。在电芯均衡开始之前，保护器必须开启并以轮询模式运行。有关保护器控制详细信息，请参阅节 6.3.4。为了使用 OTCB 功能，MCU 遵循以下设置序列状态：

- 启用 OT 保护器之前：
  - 用于该功能的 GPIO 将配置为 ADC 和 OTUT 输入。
  - 已配置  $[OTCB\_THR3:0]$  和  $[COOLOFF2:0]$ 。
- 在轮询模式下启用 OT 保护器。
- 将  $[OTCB\_EN]$  和  $[BAL\_GO]$  设置为 1。

如果未能进行该设置，则可能会导致 OTCB 无法暂停或在错误的温度下暂停。如果需要不同的 OTCB 或 COOLOFF 阈值，MCU 将配置新的阈值，然后重新启动 OT 保护器以锁存新的设置。不需要重新发送  $[BAL\_GO] = 1$ 。

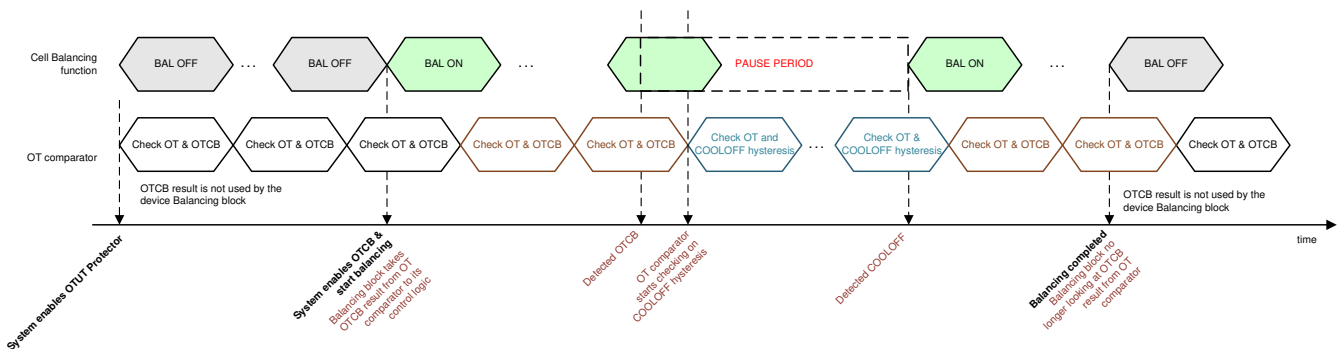


图 6-17. 通过 OTCB 检测暂停和恢复电芯均衡

#### 6.3.3.1.4 步骤 3b：在达到电芯电压阈值时停止的选项

除了均衡计时器之外，如果通道电压小于 `VCB_DONE_THRESH` 寄存器设置的非零阈值，电芯均衡也会停止。该停止电压阈值适用于所有通道。使用该停止选项时，通道将在其均衡计时器到期或其电压电平小于 `VCB_DONE_THRESH` 设置时停止均衡。

`VCB_DONE_THRESH` 设置检测由集成 UV 保护器执行。在电芯均衡开始之前，保护器必须开启并以轮询模式运行。有关保护器控制详细信息，请参阅节 6.3.4。

在使用 `VCB_DONE` 检测功能时，MCU 遵循以下设置序列状态：

- 配置 `VCB_DONE_THRESH` 寄存器
- 在轮询模式下启用 UV 保护器
- 发送 `[BAL_GO] = 1`

如果不这么做，可能会导致无法进行 `VCB_DONE` 检测或电芯均衡在错误的通道电压下停止。

如果需要不同的 `VCB_DONE` 阈值，MCU 会配置新的阈值，然后重新启动 UV 保护器以锁存新的设置。不需要重新发送 `[BAL_GO] = 1`。

#### 6.3.3.1.5 步骤 3c：在发生故障时停止的选项

该器件提供了一个选项，用于在检测到未屏蔽的故障时中止电芯均衡。为了启用该选项，MCU 在开始电芯均衡之前设置 `BAL_CTRL2[FLTSTOP_EN] = 1`。如果在这种情况下中止电芯均衡，则 `BAL_STAT[ABORTFLT] = 1`。

#### 6.3.3.2 SLEEP 模式下的电芯均衡

电芯均衡功能可以在 ACTIVE 和 SLEEP 模式下运行。要在 SLEEP 模式下运行电芯均衡，只需首先在 ACTIVE 模式下配置并启动电芯均衡。电芯均衡运行后，将器件置于 SLEEP 模式。电芯均衡将在 SLEEP 模式下自动继续运行。有关将器件置于 SLEEP 模式的说明，请参阅节 6.4。

当电芯均衡完成且 `BAL_STAT[CB_DONE] = 1` 时，可以选择使用 `BAL_CTRL2[BAL_ACT1:0]` 将器件置于不同的功耗模式。例如，将 `[BAL_ACT1:0]` 设置为 0b10 (SHUTDOWN 模式) 并启动电芯均衡，当所有启用均衡的通道都完成电芯均衡后，器件将自动进入无 MCU 交互的 SHUTDOWN 模式。当以菊花链结构连接多个器件时，一个器件可能完成均衡，但另一个器件可能无法完成均衡。使用该选项可能会导致器件在一段时间内处于不同的功耗状态。有关 `BAL_STAT[CB_DONE]` 位设置条件的详细信息，请参阅节 6.3.3.3。

#### 6.3.3.3 暂停和停止电芯均衡

##### 6.3.3.3.1 电芯均衡暂停

可以通过以下三种方法之一暂停电芯均衡：

- 如果均衡期间的内核温度  $> T_{CB\_TWARN}$ 。
- 如果任何热敏电阻检测到温度大于 `OTCB_THR` 时 `[OTCB_EN] = 1`。
- MCU 设置 `BAL_CTRL2[CB_PAUSE] = 1`。

节 6.3.3.1.3 介绍了前两个条件。第三个暂停条件是 MCU 控制的暂停操作，通常在涉及 CB 路径的诊断检查期间使用。均衡开始后，MCU 可以在任何给定的时间通过 `[CB_PAUSE]` 位暂停电芯均衡。

当由于任何暂停方法而暂停电芯均衡时，暂停活动是相同的：

- 关闭所有通道上的 CBFET。
- 所有均衡计时器都处于保持或“冻结”状态。
- `BAL_STAT[CB_INPAUSE] = 1`。
- 在暂停状态期间检测到的任何未屏蔽故障都不会终止电芯均衡。这是因为暂停事件可以在诊断期间使用，并且故障插入可以是诊断的一部分。

器件退出电芯均衡暂停状态后，电芯均衡就会恢复。电芯均衡计时器将继续递减计数。计时器中具有非零值的 CB 通道将继续进行均衡。

### 6.3.3.3.2 电芯均衡停止

在表 6-5 中总结的三种情况之一下，电芯均衡会停止。

表 6-5. 电芯均衡停止条件

停止条件	适用于单独的通道？	设置 $BAL\_STAT[CB\_DONE] = 1$ ?
电芯均衡计时器到期	是，针对每个通道监测该停止条件	是，当所有通道都满足停止条件 1 或停止条件 2 时。
CB 通道电压 < $VCB\_DONE\_THRESH$ 寄存器值	是，针对每个通道监测该停止条件	
$[FLTSTOP\_EN] = 1$ 且检测到未屏蔽的故障	否，这会停止所有通道上的电芯均衡	否，而是设置 $BAL\_STAT[ABORTFLT] = 1$

此外，MCU 还可以通过以下任一方式强制停止任何特定通道或所有通道上的电芯均衡：

- 将均衡计时器设置归零并发出  $[BAL\_GO] = 1$ 。
- 在  $VCB\_DONE\_THRESH$  寄存器中设置大于 CB 通道电压的电压并发出  $[BAL\_GO] = 1$ 。

由于电芯均衡计时器是启动电芯均衡的主要控制功能，因此如果 MCU 将所有均衡计时器重置为 0 并且  $[BAL\_GO] = 1$ ，则器件不会启动均衡并且  $BAL\_STAT[CB\_DONE]$  保持为 0。

另一方面，如果任何电芯均衡计时器不为零，但  $VCB\_DONE\_THRESH$  寄存器被设置为大于所有 CB 通道电压的阈值且  $[BAL\_GO] = 1$ ，则由于均衡计时器上的非零值，器件开始电芯均衡，但由于  $VCB\_DONE\_THRESH$  停止条件而立即停止。在这种情况下， $BAL\_STAT[CB\_DONE]$  被设置为 1。

### 6.3.3.3.3 剩余 CB 时间

每个通道都有一个均衡计时器，当均衡开始时，这些计时器从  $CB\_CELLn\_CTRL$  (其中  $n = 1$  至 16) 寄存器配置的均衡时间开始递减计数。当均衡暂停时，这些计时器会暂停。

为了读取剩余 CB 时间，MCU 将  $[BAL\_TIME\_SEL3:0]$  设置为选择单个通道，然后发出  $[BAL\_TIME\_GO] = 1$ ，从而将所选通道的剩余 CB 时间锁存到  $BAL\_TIME$  寄存器中。重复这些步骤可以读取其他通道上的其他剩余 CB 时间。仅当 CB 正在运行、处于暂停状态或处于有效的 CB 停止条件时该计时器信息才有效。

如果  $BAL\_TIME$  寄存器报告无效的 0x7F 或 0xFF 值，则表明均衡配置使均衡保持在停止状态，例如  $[BAL\_GO] = 1$  并且所有均衡计时器设置为 0，或者 MCU 从不发出  $[BAL\_GO] = 1$ 。

表 6-6.  $BAL\_TIME$  寄存器状态

CB 停止条件	$BAL\_TIME$ 寄存器
电芯均衡计时器到期	所选的 CB 通道报告 0s
CB 通道电压 < $VCB\_DONE\_THRESH$ 寄存器值	所选的 CB 通道报告剩余 CB 时间
$[FLTSTOP\_EN] = 1$ 且检测到未屏蔽的故障	

### 6.3.3.4 模块均衡

可以通过 GPIO 灌入较小的电流，以用作一种均衡模块电压的方法。主机可以在 GPIO 上连接一个负载电阻器，并将 GPIO 配置为数字输出高电平，进而通过由模块堆栈调节的 CVDD 加载电流。此类控制可以通过主机手动开启或关闭。

或者，器件可以通过称为模块均衡 (MB) 的功能来控制该负载路径。该概念与通过连接到 GPIO 的负载电阻器耗尽模块电压相同 (MB 接管 GPIO3 以实现此功能)。然而，主机可以设置模块均衡计时器和停止阈值，以通过模块均衡功能自动关闭负载路径。

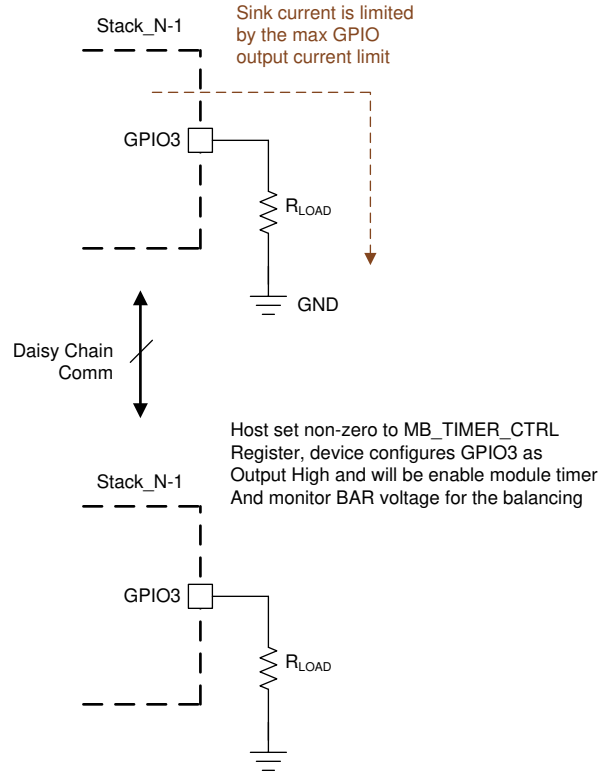


图 6-18. 模块均衡

#### 6.3.3.4.1 启动模块均衡

1. 将一个非零值设置到 `MB_TIMER_CTRL` 寄存器中。
2. 在 `VMB_DONE_THRESH` 寄存器中配置停止 MB 电压阈值。
3. 可以通过 AUX ADC 来执行堆栈模块监控。因此主机以连续模式启动 AUX ADC。
4. 发送 `BAL_CTRL2[BAL_GO] = 1`。
5. GPIO3 将被接管以执行该功能，被设置为数字输出端口，并开始通过负载电阻器灌入电流。  
`BAL_STAT[MB_RUN] = 1`

#### 6.3.3.4.2 停止模块均衡

启动后，如果发生以下情况之一，MB 将停止：

- 均衡计时器达到 `MB_TIMER_CTRL` 设置，器件将设置 `BAL_STAT[MB_DONE] = 1`。
- BAT 电压小于 `VMB_DONE_THRESH` 设置 (AUX ADC 必须开启)，设备将设置 `BAL_STAT[MB_DONE] = 1`。
- 主机通过设置 `MB_TIMER_CTRL = 0` 来停止均衡并发送 `BAL_CTRL2[BAL_GO] = 1`。 `BAL_STAT[MB_DONE] = 0`。
- 如果 `BAL_CTRL[FLTSTOP_EN] = 1` 并且检测到未屏蔽的故障，则 `BAL_STAT[MB_DONE] = 0`

#### 备注

- 模块均衡没有暂停控制 (手动暂停或热暂停)。
- 如果 AUX ADC 在模块均衡期间停止 (被主机停止或器件进入 SLEEP 模式)，器件将不会根据 BAT 电压停止模块均衡。模块均衡将因 MB 计时器到期条件而停止。
- `BAL_CTRL2[BAL_ACT1:0]` 设置也适用于模块均衡功能。当 `[MB_DONE] = 1` 且 `[CB_DONE] = 1` (如果启用了 CB) 时，器件可以进入由 `[BAL_ACT1:0]` 设置配置的功耗模式。

### 6.3.4 集成硬件保护器

该器件集成了电芯 OV 和 UV 保护器以及热敏电阻 OT 和 UT 保护器，这些保护器具有独立于 ADC 功能或 ADC 测量路径的可编程阈值。OVUV 和 OTUT 保护器可以在 ACTIVE 或 SLEEP 模式下运行。以下各节对保护器进行了概述。有关该块的诊断控制功能和状态，请参阅节 6.3.6.4。

#### 6.3.4.1 OVUV 保护器

设置的窗口比较器为所有 VC 通道提供电芯电压监测。该比较器功能完全独立于 ADC 功能，因此，即使 ADC 功能失败，模拟比较器仍会标记超过过压 (OV) 和欠压 (UV) 比较器阈值的情况。编程的阈值通过 DAC 转换至比较器。

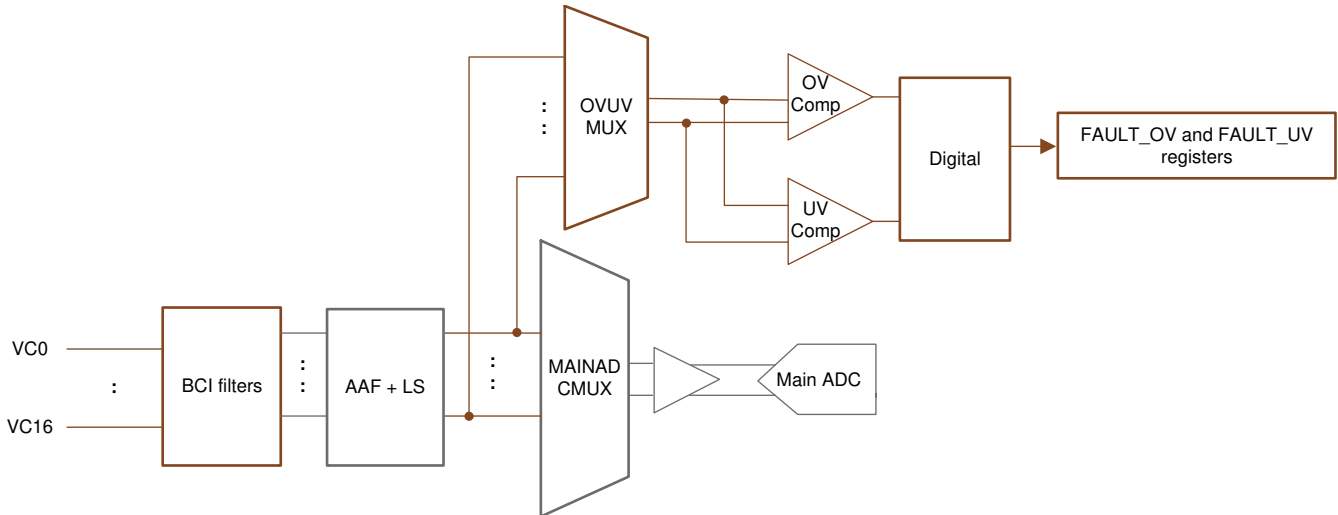


图 6-19. OV 和 UV 保护器

由 `OV_THRESH` 和 `UV_THRESH` 寄存器设置的 OV 和 UV 阈值对于所有 VC 通道都相同。有效通道由 `ACTIVE_CELL[NUM_CELL3:0]` 位定义。这些位设置最高的有效通道编号，并且器件假定任何较低的通道也有效。

`UV_DISABLE1` 和 `UV_DISABLE2` 寄存器设置禁用任何单独通道的 UV 检测，例如通道连接至汇流条。

否则，如果 VC 通道电压大于 `OV_THRESH` 设置，则 OV 保护器会检测到特定通道上的 OV 故障。如果 VC 通道电压低于 `UV_THRESH` 设置，则 UV 保护器会检测到特定通道上的 UV 故障。

##### 6.3.4.1.1 OVUV 运行模式

OV 和 UV 保护器具有多个由 `OVUV_CTRL[OVUV_MODE1:0]` 控制的运行模式，表 6-7 汇总了这些运行模式。为了启动 OVUV 保护器，MCU 设置 `OVUV_CTRL[OVUV_GO] = 1`。

表 6-7. OVUV 保护器运行模式

[OVUV_MOD1:0]	工作模式	说明
0b00	停止 OV 和 UV 保护器	停止 OV 和 UV 保护器
0b01	轮询运行	OV 和 UV 保护器遍历所有 VC 输入。会根据 OV 和 UV 阈值检查有效的通道 (图 6-19)。无论有效通道的数量有多少，轮询周期时序始终相同。对于无效的 VC 通道，数字逻辑会简单地忽略检测结果。UV 保护器检测 <code>UV_THRESH</code> 和 <code>VCB_DONE_THRESH</code> 。
0b10	OV 和 UV BIST 运行 (诊断用途，有关详细信息，请参阅节 6.3.6.4)	OV 和 UV 比较器以及检测路径上的 BIST (内置自检) 周期。在该运行期间，来自主 ADC 的 VCELL (VC 通道) ADC 测量以及通过 OVUV 保护器进行的 OV 和 UV 检测不可用。执行 OVUV BIST 时，MCU 应停止 ADC 测量。

表 6-7. OVUV 保护器运行模式 (续)

[OVUV_MOD1:0]	工作模式	说明
0b11	单通道运行 (诊断用途, 有关详细信息, 请参阅节 6.3.6.4)	用于检查 OV 和 UV DAC。在该模式下, OV 和 UV 比较器被锁定到单个 VC 输入通道。通道被 OVUV_CTRL[OVUV_LOCK3:0] 锁定。

如果 OVUV BIST 运行正在进行中, 但 MCU 启动 ADC, 则 ADC 结果寄存器将保持为 0x8000。OVUV BIST 完成且经过  $t_{AFE\_SETTLE}$  时间后, ADC 测量将恢复。

如果 ADC 正在运行, 但 MCU 启动 OVUV BIST, 则 ADC 结果寄存器将保持其最后一次测量值。OVUV BIST 完成且经过  $t_{AFE\_SETTLE}$  时间后, ADC 测量更新将恢复

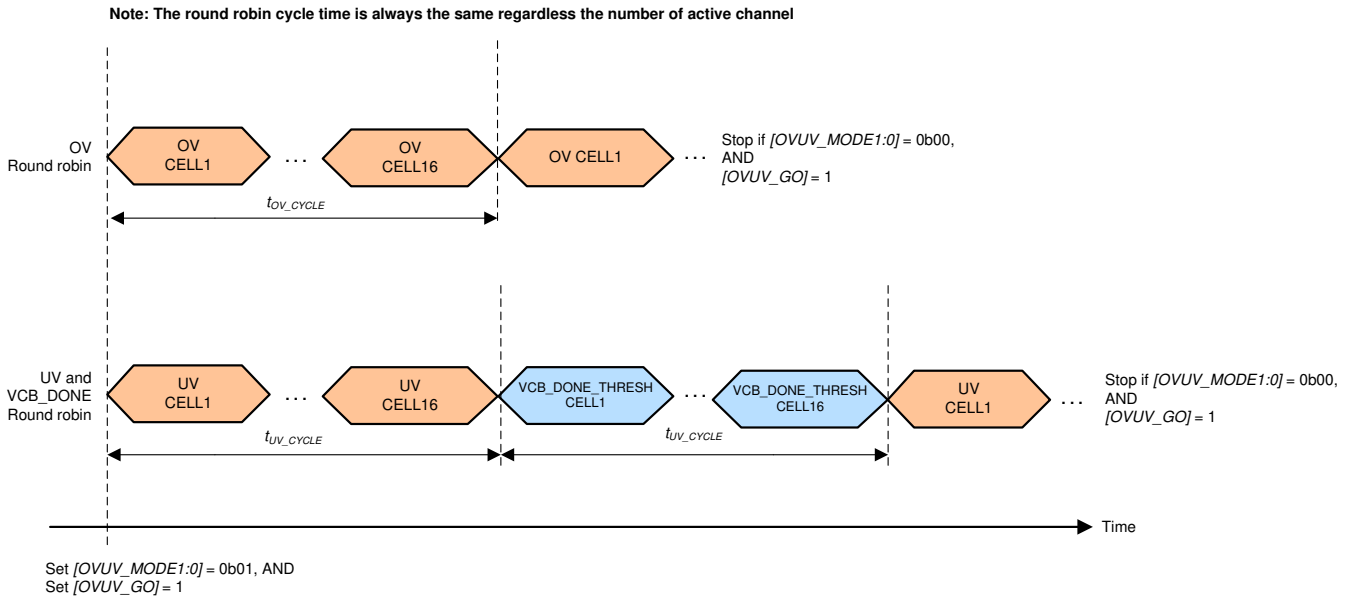


图 6-20. OV 和 UV 轮询模式

### 6.3.4.1.2 OVUV 控制和状态

#### 6.3.4.1.2.1 OVUV 控制

为了启动 OV 和 UV 保护器, MCU 设置  $OVUV\_CTRL[OVUV\_GO] = 1$ 。当器件接收到 GO 命令时, 它会对以下寄存器设置进行采样, 然后相应地启动 OVUV 保护器。要对以下设置进行任何更改, 都需要 MCU 重新发送另一条 GO 命令以实现新设置。

- **OV\_THRESH** 寄存器: 设置所有 VC 通道的 OV 阈值
- **UV\_THRESH** 寄存器: 设置所有 VC 通道的 UV 阈值
- **VCB\_DONE\_THRESH** 寄存器: 设置电芯均衡停止条件的 VCB\_DONE 阈值 (如果启用)
- **OVUV\_CTRL[OVUV\_MODE1:0]**: OVUV 运行模式选择
- **ACTIVE\_CELL** 寄存器: 确定无效 VC 通道并相应地忽略检测结果
- **UV\_DISABLE1** 和 **UV\_DISABLE2** 寄存器: 确定无效 VC 通道并相应地忽略检测结果。

OVUV 保护器还可以在 SLEEP 模式下运行。MCU 首先在 ACTIVE 模式下启动保护器, 然后将器件置于 SLEEP 模式。OVUV 保护器将继续运行, 直到 MCU 命令其停止或器件关闭。

#### 6.3.4.1.2.2 OVUV 状态

$DEV\_STAT[OVUV\_RUN] = 1$  指示 OVUV 保护器正在运行。OV 检测结果反映在 **FAULT\_OV1** 和 **FAULT\_OV2** 寄存器中; UV 检测结果反映在 **FAULT\_UV1** 和 **FAULT\_UV2** 寄存器中。

VCB\_DONE 检测不是故障，而是电芯均衡停止情况。结果反映在停止电芯均衡的特定通道中。有关详细信息，请参阅节 6.3.3。

### 6.3.4.2 OTUT 保护器

设置的窗口比较器可通过上拉至 TSREF 的外部热敏电阻网络来对所有 GPIO 输入进行温度监测。该比较器功能完全独立于 ADC 功能，因此，即使 ADC 功能失败，模拟比较器仍会标记超过过热 (OT) 和欠温 (UT) 比较器阈值的情况。编程的阈值通过 DAC 转换至比较器。

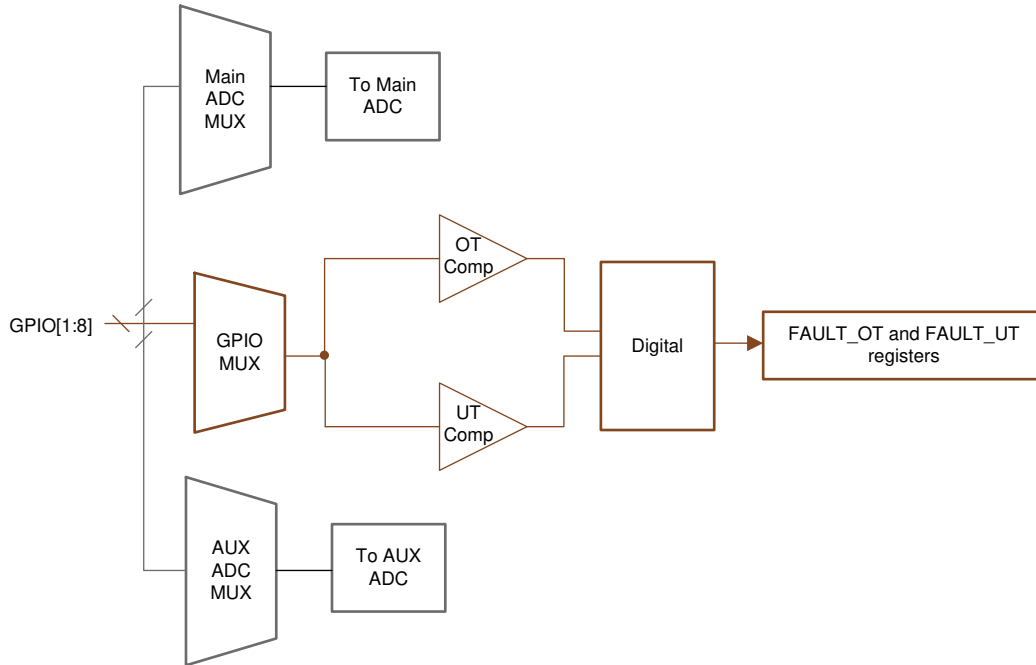


图 6-21. OT 和 UT 保护器

由  $OTUT\_THRESH[OT\_THR4:0]$  和  $OTUT\_THRESH[UT\_THR2:0]$  位设置的 OT 和 UT 阈值对于所有有效 GPIO 输入都是相同的。有效 GPIO 输入由  $GPIO\_CONF_n[GPIO*2:0]$  定义 (其中  $n = 1$  至  $4$ ,  $*$  =  $1$  至  $8$ , 表示对应的 GPIO 输入) GPIO 必须配置为 ADC 和 OTUT 输入才能被视为 OTUT 保护器的有效 GPIO 输入。

OTUT 比较器使用 TSREF 作为基准，因此检测是比例式的。如果 (GPIO 电压/TSREF) 小于  $OTUT\_THRESH[OT\_THR4:0]$  设置，则 OT 保护器会检测到特定 GPIO 上的 OT 故障。如果 (GPIO 电压/TSREF) 大于  $OTUT\_THRESH[UT\_THR2:0]$  设置，则 UT 保护器会检测到特定 GPIO 上的 UT 故障。OTUT 保护器假设使用 NTC 热敏电阻进行温度监测。

MCU 确保在启动 OTUT 保护器之前启用 TSREF。否则，OTUT 保护器会将所有 GPIO 输入上的所有 OT 和 UT 故障标记为异常检测指示。

#### 6.3.4.2.1 OTUT 运行模式

OT 和 UT 保护器具有多个由  $OTUT\_CTRL[OTUT\_MODE1:0]$  控制的运行模式，表 6-8 汇总了这些运行模式。为了启动 OTUT 保护器，MCU 设置  $OTUT\_CTRL[OTUT\_GO] = 1$ 。

表 6-8. OTUT 保护器运行模式

[OTUT_MOD1:0]	工作模式	说明
0b00	停止 OT 和 UT 保护器	停止 OT 和 UT 保护器

表 6-8. OTUT 保护器运行模式 (续)

[OTUT_MOD1:0]	工作模式	说明
0b01	轮询运行	OT 和 UT 保护器遍历所有 GPIO 输入。会根据 OT 和 UT 阈值检查有效的 GPIO 输入 (图 6-22)。无论有效 GPIO 的数量有多少, 轮询周期时序始终相同。对于无效的 GPIO 输入, 数字逻辑会简单地忽略检测结果。OT 保护器检测 OT 阈值和 OTCB 阈值。
0b10	OT 和 UT BIST 运行 (诊断用途, 有关详细信息, 请参阅节 6.3.6.4)	OT 和 UT 比较器以及检测路径上的 BIST (内置自检) 周期。在该运行期间, 来自自主 ADC 或 AUX ADC 的温度 (GPIO 通道) ADC 测量以及通过 OTUT 保护器进行的 OT 和 UT 检测不可用。
0b11	单通道运行 (诊断用途, 有关详细信息, 请参阅节 6.3.6.4)	用于检查 OT 和 UT DAC。在该模式下, OT 和 UT 比较器被锁定到单个 GPIO 输入通道。通道被 OTUT_CTRL[OTUT_LOCK2:0] 锁定。

Note: The round robin cycle time is always the same regardless of the number of active GPIO inputs

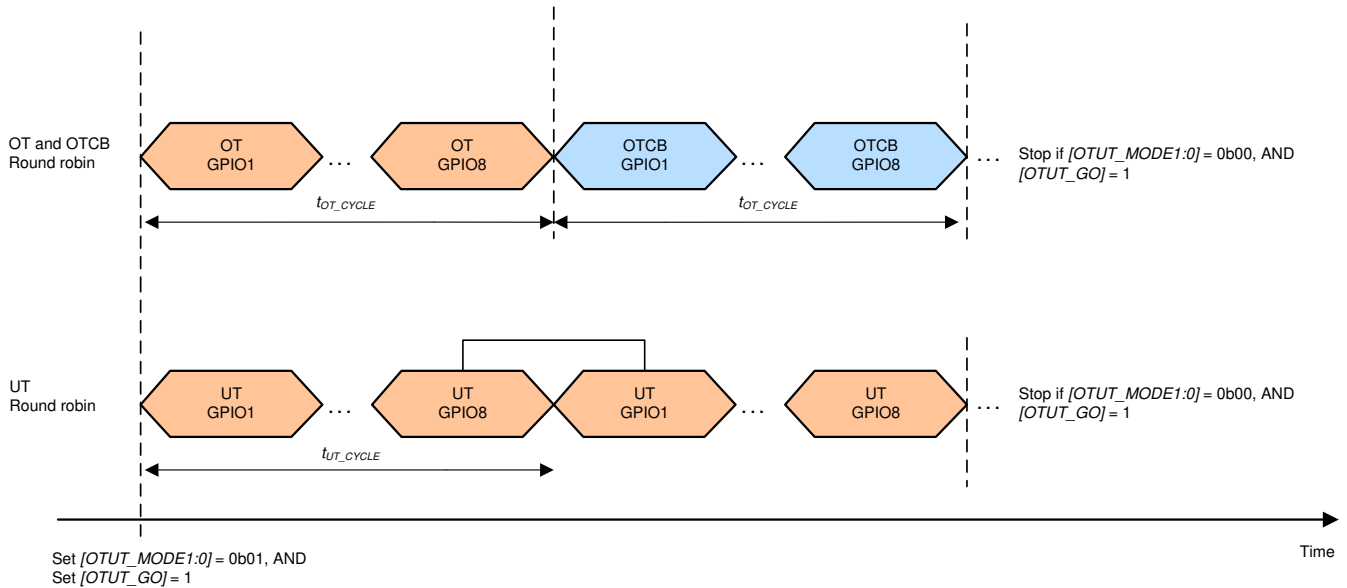


图 6-22. OT 和 UT 轮询模式

### 6.3.4.2.2 OTUT 控制和状态

#### 6.3.4.2.2.1 OTUT 控制

确保 TSREF 已启用。为了启动 OT 和 UT 保护器, 主机 MCU 设置  $OTUT\_CTRL[OTUT\_GO] = 1$ 。当器件接收到 GO 命令时, 它会对以下寄存器设置进行采样, 然后相应地启动 OTUT 保护器。要对以下设置进行任何更改, 都需要 MCU 发送另一条 GO 命令以实现新设置。

- $OTUT\_THRESH[OT\_THR4:0]$ : 设置所有有效 GPIO 输入的 OT 阈值
- $OTUT\_THRESH[UT\_THR2:0]$ : 设置所有有效 GPIO 输入的 UT 阈值
- $OTCB\_THRESH$  寄存器: 设置 OTCB 阈值和 COOLOFF 迟滞 (如果启用)
- $OTUT\_CTRL[OTUT\_MODE1:0]$ : OTUT 运行模式选择
- $GPIO\_CONF1$  至  $GPIO\_CONF4$ : 确定无效的 GPIO 通道并忽略检测结果。

OTUT 保护器还可以在 SLEEP 模式下运行。MCU 首先在 ACTIVE 模式下启动保护器, 然后将器件置于 SLEEP 模式。OTUT 保护器将继续运行, 直到 MCU 命令其停止或器件关闭。

#### 6.3.4.2.2.2 OTUT 状态

$DEV\_STAT[OTUT\_RUN] = 1$  指示 OTUT 保护器正在运行。OT 检测结果反映在  $FAULT\_OT$  寄存器中; UT 检测结果反映在  $FAULT\_UT$  寄存器中。

OTCB 检测不是故障，而是电芯均衡暂停情况。结果反映在暂停电芯均衡的特定通道中。有关详细信息，请参阅节 6.3.3。

### 6.3.5 GPIO 配置

该器件具有八个 GPIO。可以通过 *GPIO\_CONF1* 至 *GPIO\_CONF4* 寄存器将每个 GPIO 编程为以下配置之一。

GPIO	DISABLE	输入			输出		弱上拉/下拉		特殊		
	高阻态	数字	ADC 和 OTUT	仅 ADC	高	低	ADC 和弱上拉	ADC 和弱下拉	模块均衡 <i>MB_TIMER_CTRL</i> 不是 0x00	SPI 控制器 <i>[SPI_EN] = 1</i>	故障输入 <i>[FAULT_IN_EN] = 1</i>
GPIO1	✓	✓	✓	✓	✓	✓	✓	✓			
GPIO2	✓	✓	✓	✓	✓	✓	✓	✓			
GPIO3	✓	✓	✓	✓	✓	✓	✓	✓	✓ (输出, 高电平)		
GPIO4	✓	✓	✓	✓	✓	✓	✓	✓		✓ (SS)	
GPIO5	✓	✓	✓	✓	✓	✓	✓	✓		✓ (MISO)	
GPIO6	✓	✓	✓	✓	✓	✓	✓	✓		✓ (MOSI)	
GPIO7	✓	✓	✓	✓	✓	✓	✓	✓		✓ (SCLK)	
GPIO8	✓	✓	✓	✓	✓	✓	✓	✓			✓ (输入, 低电平有效)

GPIO 配置		说明
DISABLE	高阻态	如果未对 OTP 进行编程，则这是复位时的默认 GPIO 配置
输入	数字	当 GPIO 配置为数字输入时，器件会检测输入电压电平以确定相对于其 $V_{IL}$ 和 $V_{IH}$ 是 1 还是 0。结果显示在 <i>GPIO_STAT</i> 寄存器中。
	ADC 和 OTUT	GPIO 配置为可由 ADC (主 ADC 和 AUX ADC) 测量，也可作为 OTUT 保护器的输入。示例：针对用于热敏电阻连接的 GPIO 使用该选项。
	仅 ADC	GPIO 配置为只能由 ADC (主 ADC 和 AUX ADC) 进行测量。示例：使用该选项来测量 GPIO 上的电压。
输出	高	GPIO 配置为数字输出高电平 (在内部上拉至 CVDD)。逻辑状态也显示在 <i>GPIO_STAT</i> 寄存器中。
	低	GPIO 配置为数字输出低电平。逻辑状态也显示在 <i>GPIO_STAT</i> 寄存器中。
弱上拉/下拉	ADC 和弱上拉	GPIO 在内部上拉，配置为由 ADC (主 ADC 和 AUX ADC) 进行测量。
	ADC 和弱下拉	GPIO 在内部下拉，配置为由 ADC (主 ADC 和 AUX ADC) 进行测量。
特殊	模块均衡	当 <i>MB_TIMER_CTRL</i> 寄存器不为零时，GPIO3 将被接管以用于模块均衡控制。该配置的优先级高于 GPIO3 上的任何输入/输出配置。
	SPI 控制器	当 <i>GPIO_CONFIG1[SPI_EN] = 1</i> 时，GPIO4 至 GPIO7 被接管以用作 SPI 控制器通信线路。该配置的优先级高于 GPIO4 至 GPIO7 上的任何输入/输出配置。
	故障输入	当 <i>GPIO_CONFIG1[FAULT_IN_EN] = 1</i> 时，GPIO8 被接管以用作输入，如果 GPIO 生效 (低电平有效)，这将设置 <i>FAULT_SYS[GPIO] = 1</i> 并使 NFAULT 生效 (如果启用)。

### 6.3.6 通信、OTP、诊断控制

#### 6.3.6.1 通信

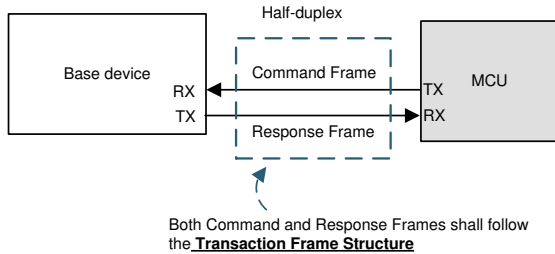
该器件可以在多点配置 (*DEV\_CONF[MULTIDROP\_EN] = 1*) 中作为独立器件运行，也可以在菊花链配置 (*DEV\_CONF[MULTIDROP\_EN] = 0*) 中作为基底/堆栈器件运行。在多点配置中，菊花链通信被禁用，主机仅通过 UART 接口与单个器件通信。本文档将重点介绍菊花链通信。

在菊花链配置中，每个器件由一个 6 位器件地址标识；因此，菊花链中最多可以连接 64 个器件。在该配置中，器件要么定义为基底器件 (通过 UART 与主机相连接)，要么定义为堆栈器件 (通过菊花链端口 COMH/COML 与

基底器件相连接)。本文档中的基底器件说明假设使用 **BQ79616** 作为基底器件。如果使用通信扩展器 (也称为桥接器件) 作为基底器件, 则用户必须参阅桥接器件的数据表以了解详细信息。

### 6.3.6.1.1 串行接口

该器件具有一个使用 **UART** 协议作为物理层在基底器件和主机之间进行通信的串行接口。该通信采用专有帧结构。



#### Transaction Frame Structure (to/from system MCU to the base device):

A transaction frame consists of 5 types of information as shown above. Data are all sent in byte, and each byte is sent through UART protocol.



Each frame is sent through the **UART Protocol**

UART Protocol:

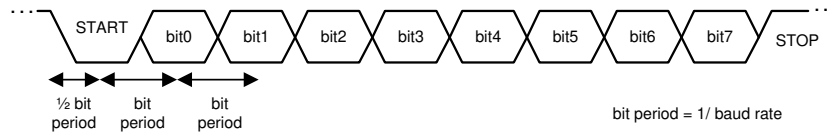


图 6-23. 与主机的 UART 通信

#### 6.3.6.1.1.1 UART 物理层

UART 接口遵循 **8-N-1** 标准串行协议, 发送的信息包含一个 **START** 位, 后跟八个数据位, 然后是一个 **STOP** 位。STOP 位指示字节结束。如果接收到的字节没有设置 **STOP** 位, 则会设置 **FAULT\_COMM1[STOP\_DET]** 位, 指示主机和器件之间可能存在波特率问题。该器件支持 **1Mbps** 波特率。此外, 在开发过程中, 需要使用较慢的波特率来调试通信, 可以在通信调试模式下启用可选的 **250kbps** 波特率。

UART 通过 **TX** 引脚发送数据, 通过 **RX** 引脚接收数据。空闲时, **TX** 和 **RX** 引脚为高电平。UART 接口要求 **RX** 通过基底器件上的电阻器上拉至 **CVDD**。**RX** 在器件侧上拉。请勿断开 **RX** 的连接。对于堆栈器件, 确保 **RX** 直接连接到 **CVDD**。

**TX** 引脚在堆栈器件中被禁用, 但必须通过基底器件上主机侧的电阻器拉高, 以防止在未连接通信电缆时或在断电或 **SHUTDOWN** 状态期间 (此时 **TX** 具有高阻抗) 触发无效的通信帧。在 **ACTIVE** 或 **SLEEP** 模式下, 无论启用还是禁用, **TX** 始终在内部拉至 **CVDD**。如果未在堆栈器件中使用, 请将 **TX** 保持未连接状态。

UART 接口严格来说是半双工接口。在发送时, 任何尝试在 **RX** 上进行的通信都将被忽略。唯一的例外是 **RX** 引脚上的 **COMM CLEAR** 信号, 该信号会立即终止通信。有关详细信息, 请参阅节 **6.3.6.1.1.1.3**。

在 UART 中使用两个 **STOP** 位:

可以为器件设置两个停止位 ( $DEV\_CONF[TWO\_STOP\_EN] = 1$ )，从器件传输到主机的 UART 响应帧始终返回两个 STOP 位，如下所示。主机无需向器件发送带有两个 STOP 位的命令帧。无论是否启用该功能，器件都能够接收一个或多个停止位。

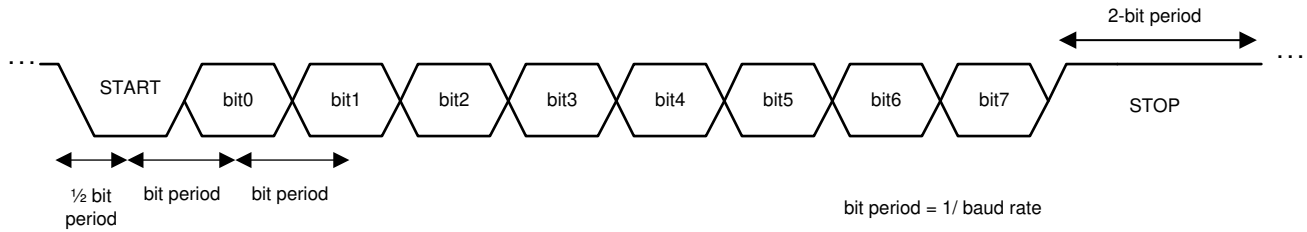


图 6-24. 具有两个 STOP 位的 UART 响应帧

两个停止位的潜在用途可能是：

- 主机在接收下一个数据帧之前获得额外的时间来处理数据。
- 器件和主机之间的时钟容差可能会导致数据检测不同步。具有两个 STOP 位允许重新同步通信，因此提高了通信的稳健性。

虽然 UART 仅由基底器件使用，但如果  $[TWO\_STOP\_EN] = 1$ ，那么即使堆栈中未使用 UART，堆栈器件也会设置  $[TWO\_STOP\_EN] = 1$ 。这是因为堆栈器件将使用位设置来确定在两个通信帧之间应用的合适间隙。

#### 6.3.6.1.1.1.1 UART 发送器

发送器配置为在使用  $TX\_HOLD\_OFF$  寄存器开始发送之前接收到最后一个位后等待指定数量的位周期。这为主机提供了在传输结束时切换总线方向的时间。堆栈器件中默认禁用 UART 发送器。

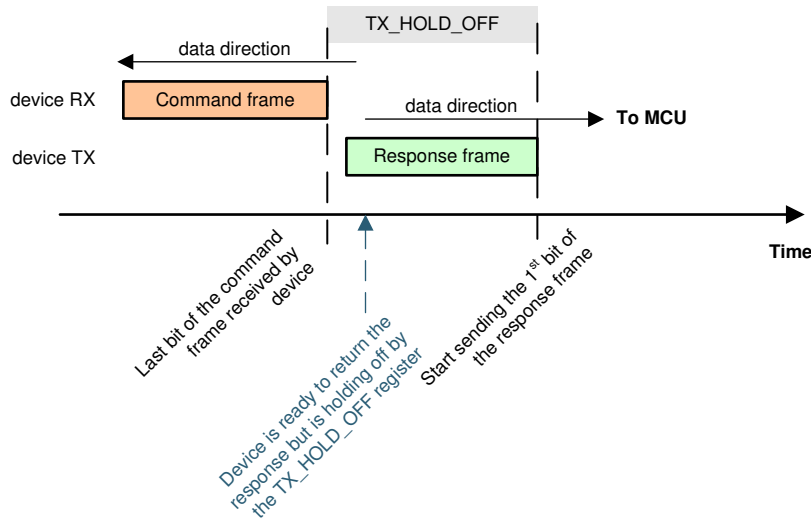


图 6-25. UART TX\_HOLD\_OFF

#### 6.3.6.1.1.1.2 UART 接收器

当器件在 TX 上传输数据时，除非接收到 COMM CLEAR，否则 RX 将被忽略。为了避免在菊花链接口上进行数据传输期间发生冲突，主机必须等待器件接收到通信传输的所有字节后再尝试与器件进行其他通信。如果主机在没有等待接收前一个事务的响应的情况下开始传输，则该通信被认为是不可靠的，主机必须发送 COMM CLEAR 来恢复与基底器件的正常通信。

### 6.3.6.1.1.3 COMM CLEAR

COMM CLEAR 在基底器件的 RX 引脚上发送。它不会发送到堆栈器件。无论 TX 状态如何，都无法禁用 RX，并且可以随时发送 COMM CLEAR。确保 COMM CLEAR 不超过  $t_{\text{UART(CLR)}}$  位周期的最大值，因为这可能会导致识别其他通信 ping。

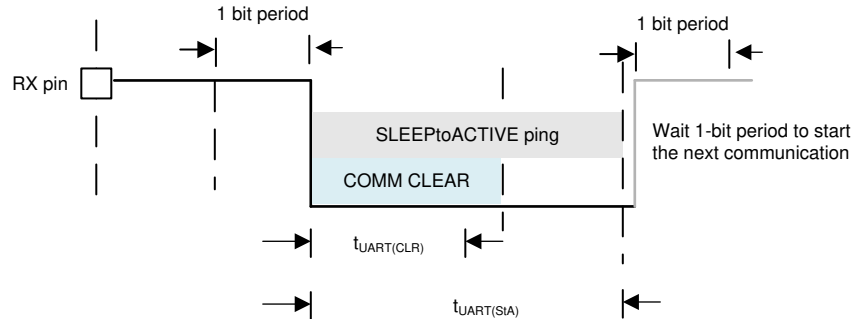


图 6-26. UART COMM CLEAR

使用 COMM CLEAR 命令清除接收器并指示 UART 引擎寻找新的帧开始。COMM CLEAR 之后的下一个字节始终被视为帧开始字节。当检测到时，COMM CLEAR 会设置  $\text{FAULT\_COMM1[COMMCLR\_DET]}$  标志。主机必须在 COMM CLEAR 之后至少等待  $t_{\text{UART(RXMIN)}}$  才能开始发送新帧。需要注意的是，除了  $[\text{COMMCLR\_DET}]$  标志之外，还会设置  $\text{FAULT\_COMM1[STOP\_DET]}$  标志，因为 COMM CLEAR 时序违反了典型的字节时序，STOP 位被视为 0。

SLEEPtoACTIVE ping/音也会清除 UART 接收器。在从 SLEEP 模式转换为 ACTIVE 模式时，该 ping/音会设置  $[\text{COMMCLR\_DET}]$  标志。如果在 ACTIVE 模式期间发送了该 ping/音，则会设置  $[\text{COMMCLR\_DET}]$  和  $[\text{STOP\_DET}]$  标志。

#### 在菊花链通信期间发送 COMM CLEAR :

当发送读取命令，但响应尚未完全返回到主机时，如果在这种情况下在基底器件中接收到 COMM CLEAR，则器件响应会被丢弃。此外，堆栈器件看不到 COMM CLEAR 并继续发送转发到主机的响应，从而导致主机接收到意外的响应帧。因此，主机应通过等待从堆栈接收到所有响应后再发送 COMM CLEAR 来避免这种情况。

如果发生上述情况，则可能会设置基底器件低级通信调试寄存器  $\text{DEBUG\_UART\_RR\_TR[TR\_WAIT]}$  ( 指示器件正在等待发送响应 ) 或  $\text{DEBUG\_UART\_RR\_TR[TR\_SOF]}$  ( 指示器件正在发送数据时接收到 COMM CLEAR ) 位，具体取决于接收 COMM CLEAR 信号的时序。

在使用多点配置时，必须在每个帧之前使用 COMM CLEAR 信号以确保一致的通信。

#### 6.3.6.1.1.2 命令和响应协议

主机发起主机和器件之间的每个事务。在没有首先从主机接收到命令帧的情况下，器件绝不会传输数据。命令帧是主机向器件发送的通信帧；响应帧是从器件到主机 ( 对读取命令 ) 的响应。命令帧发送后，主机必须等待所有预期响应返回 ( 或在发生错误时超时 )，然后才能启动新的命令帧。表 6-9 列出了器件支持的命令。

表 6-9. 命令

命令	说明
单器件读取	从单个器件 ( 基底或堆栈 ) 读取寄存器
单器件写入	将寄存器写入单个器件 ( 基底或堆栈 )
堆栈读取	仅从堆栈器件读取寄存器。器件必须配置为堆栈器件 ( $\text{COMM\_CTRL[STACK\_DEV]} = 1$ ) 才能响应堆栈读取命令。
堆栈写入	仅针对堆栈器件写入寄存器。器件必须配置为堆栈器件 ( $\text{COMM\_CTRL[STACK\_DEV]} = 1$ ) 才能响应堆栈写入命令。
广播读取	针对菊花链中的所有器件 ( 包括基底器件 ) 读取寄存器。
广播写入	针对菊花链中的所有器件 ( 包括基底器件 ) 写入寄存器。

表 6-9. 命令 (续)

命令	说明
广播写入反向	以 <i>CONTROL1[DIR_SEL]</i> 位设置的相反方向发送广播写入。该命令用于切换与 RING 接口的通信方向。

#### 6.3.6.1.1.2.1 事务帧结构

协议层由事务帧组成。事务帧有两种基本类型：命令帧（来自主机的事务）和响应帧（来自器件的事务）。事务帧由以下五个字段类型组成：

- 帧初始化 (INIT, 1 字节)
- 器件地址 (DEV ADR, 1 字节)
- 寄存器地址 (REG ADR, 2 字节)
- 数据 (DATA, 各种字节长度)
- 循环冗余校验 (CRC, 2 字节)

##### 6.3.6.1.1.2.1.1 帧初始化字节

帧初始化字节用于命令帧和响应帧。它始终是帧的第一个字节。帧初始化字节执行两项功能。首先，它将帧定义为命令帧（主机）或响应帧（器件）。其次，它定义帧初始化字节之后的帧长度。这为接收器提供了完整命令或响应所需的确切字节数。

表 6-10. 命令帧初始化字节定义

		命令帧		响应帧	
	位	位名字	说明	位名字	说明
INIT	7	FRAME_TYPE	1 = 定义命令帧	FRAME_TYPE	0 = 定义响应帧
	6	REQ_TYPE	000 = 单器件读取	RESPONSE_BYTE	数据字节数 0x00 = 1 个字节 0x01 = 2 个字节 : 0x7f = 128 个字节
	5		001 = 单器件写入		
	4		010 = 堆叠读取		
			011 = 堆叠写入		
	3	RSVD	100 = 广播读取	RSVD	
			101 = 广播写入		
2	DATA_SIZE	110 = 广播写入反向			
1		111 = RSVD <sup>(1)</sup>			
0		保留。该位被忽略			
			命令帧的数据字节数，不包括器件地址、寄存器地址或 CRC		
			000 = 1 个字节		
			001 = 2 个字节		
			:		
			111 = 8 个字节		

(1) 不过，该选项没有任何作用，选择该设置会设置 *DEBUG\_COMMH[RC\_IERR]* 或 *DEBUG\_COMMH[RC\_IERR]* 标志，具体取决于哪个菊花链接口接收命令帧。

##### 6.3.6.1.1.2.1.2 器件地址字节

器件地址字节标识单个器件读取/写入命令的目标器件。对于广播、堆栈和广播反向命令帧，会省略该字节。所有响应帧都包含器件地址字节。在单器件读取/写入命令中，在 *DIR0\_ADDR*（用于 *CONTROL1[DIR\_SEL] = 0* 时的通信方向）或 *DIR1\_ADDR*（用于 *CONTROL1[DIR\_SEL] = 1* 时的通信方向）中包含匹配值的器件会响应命令。如果多个器件具有匹配的值，则所有这些器件都会响应，从而导致冲突。

表 6-11. 器件地址字节定义

	命令帧			响应帧	
	位	位名字	说明	位名字	说明
DEV_ADR	7	RSVD	应始终写入 0	RSVD	应始终写入 0
	6	RSVD	应始终写入 0	RSVD	应始终写入 0
	5 至 0	器件地址	设置器件地址，其范围为 0x00 至 0x3F	器件地址	设置器件地址，其范围为 0x00 至 0x3F

## 6.3.6.1.1.2.1.3 寄存器地址字节

寄存器地址的长度是两个字节。任何对无效寄存器地址的写入命令都会被忽略。对无效寄存器的任何读取都会返回 0x00 响应。对于发送到单个具有无效地址的寄存器或者作为命令的一部分发送到多个具有无效地址的寄存器的命令帧，情况都是如此。当对只具有一些无效地址的一组寄存器进行读取/写入寻址时，有效地址将正常响应，而无效地址将如前所述进行响应。

表 6-12. 寄存器地址字节定义

	命令帧			响应帧	
	位	位名字	说明	位名字	说明
REG_ADR	7 至 0	寄存器地址 (MSB)	寄存器地址的目标或开头	寄存器地址 (MSB)	寄存器地址的目标或开头
	7 至 0	寄存器地址 (LSB)	寄存器地址的目标或开头	寄存器地址 (LSB)	寄存器地址的目标或开头

## 6.3.6.1.1.2.1.4 数据字节

数据字节的数量及其传输的相关信息由发送的命令帧类型和该命令帧中指定的目标寄存器决定。当数据字节是命令帧的一部分时，数据字节包含要写入寄存器的值。当数据字节是响应帧的一部分时，数据字节包含从寄存器返回的值。

表 6-13. 数据字节定义

	命令帧			响应帧	
	位	位名字	说明	位名字	说明
数据	7	数据字节 [0]	对于写入命令： 在 REG_ADR 帧中指定要写入寄存器的数据值 对于读取命令： 指定读取命令需要返回的字节数。 0x00 = 1 个字节 0x01 = 2 个字节 ： 0x7f = 128 个字节	数据字节 [0]	在 REG_ADR 帧中指定从寄存器返回的数据值
	6				
	5				
	4				
	3				
	2				
	1				
	0				
	...	.....	.....	.....	.....
	7	数据字节 [n]	对于写入命令： 在 REG_ADR 帧中指定要写入寄存器的数据值	数据字节 [n]	在 REG_ADR 帧中指定从寄存器返回的数据值
	6				
	5				
	4				
	3				
2					
1					
0					

### 6.3.6.1.1.2.1.5 CRC 字节

该器件使用 CRC (循环冗余校验) 来保护传输过程中的数据完整性。CRC 表示类似于多项式长除法的过程的余数, 其中正在检查的帧除以发生器。附加到帧中的 CRC 是余数。根据此过程, 在器件接收一帧时, 接收方通过整个帧 (包括发送的 CRC) 计算的 CRC 将为零, 表示传输和接收正确。结果非零表示出现通信错误。具体来说, 该器件使用 CRC-16-IBM 多项式 ( $x^{16} + x^{15} + x^2 + 1$ ), 初始化为 0xFFFF。

接收到通信帧后, 第一步将检查 CRC 值。如果 CRC 不正确, 将丢弃整个帧而不进行处理。不会检查任何其他帧错误, 并且不会指示除 CRC 错误之外的任何错误。字节仍然在堆栈中向上或向下传输, 因此处理该帧的每个器件都将指示 CRC 错误。这会导致多个器件指示同一通信帧上的 CRC 故障。

### 6.3.6.1.1.2.1.6 计算帧 CRC 值

发送器在整个传输帧 (CRC 除外) 中按位流顺序进行 CRC 计算。确定执行 CRC 算法的位流顺序时务必注意: 协议字节连续发送, 并且首先发送最低有效位。图 6-27 说明了位流顺序概念。

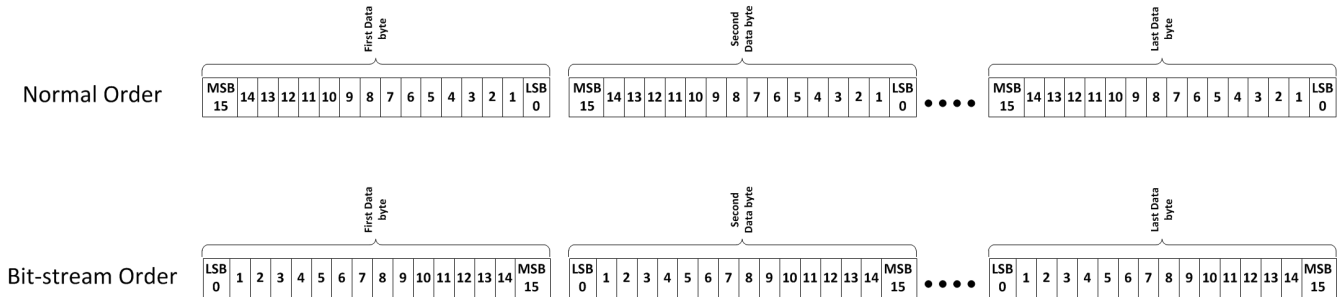


图 6-27. 位流顺序说明

CRC (0x0000) 附加在位流的末尾。然后通过 0xFFFF 进行异或运算来初始化该位流, 以捕获任何前导 0 错误。然后将该新位流除以多项式 (0xC002), 直到只剩下 2 字节 CRC。在此过程中, 位流的最高有效 17 位与多项式进行异或运算。结果的前导零被删除, 并且该结果再次与多项式进行异或运算。该过程会重复, 直到只剩下 2 字节 CRC。例如:

示例 1: 使用多项式除法进行 CRC 计算

```
Command Frame = 0x80 00 02 0F 0B (0b1000 0000 0000 0000 0000 0010 0000 1111 0000 1011)
Command Frame in bit stream order = 0x01 00 40 F0 D0 (0b0000 0001 0000 0000 0100 0000 1111 0000
1101 0000)
After Initialization (XOR with 0xFFFF) = 0b1111 1110 1111 1111 0100 0000 1111 0000 1101 0000
1111 1110 1111 1111 0100 0000 1111 0000 1101 0000 0000 0000 0000 0000 0000 0000 #append 0x0000 for CRC
1100 0000 0000 0010 1 #XOR with polynomial
0011 1110 1111 1101 1100 0000 1111 0000 1101 0000 0000 0000 0000 0000
11 1110 1111 1101 1100 0000 1111 0000 1101 0000 0000 0000 0000 0000 #delete leading zeros from
previous result
11 0000 0000 0000 101 #XOR with polynomial
00 1110 1111 1101 0110 0000 1111 0000 1101 0000
.....
.....
.....
1100 0110 0000 0001 0000 0000
1100 0000 0000 0010 1 #XOR with polynomial
0000 0110 0000 0011 1000 0000
110 0000 0011 1000 0000
110 0000 0000 0001 01 #XOR with polynomial
000 0000 0011 1001 0100
0000 0011 1001 0100 #CRC result in bit stream order
1100 0000 0010 1001 #final CRC result in normal order
CRC final 0xC029
```

### 6.3.6.1.1.2.1.7 验证帧 CRC

有多种方法可用于检查帧的 CRC。一种方法是使用前一节中所述的方法简单地计算除最后两个字节 (CRC 字节) 之外的传输命令的 CRC, 然后将该结果与传输的 CRC 字节进行比较。一种更简单的选项是通过 CRC 算法

运行整个传输。如果 CRC 正确，则结果为 0000。在这种情况下，不需要用 16 个零对位流进行初始补零。使用之前的结果并运行该算法会产生以下结果：

示例 1：使用多项式除法进行 CRC 验证：

```

Command Frame = 0x80 00 02 0F 0B (0b1000 0000 0000 0000 0000 0010 0000 1111 0000 1011)
CRC to check = 0xC029
Command Frame w/ CRC in bit stream order = 0x80 00 02 0F 0B C0 29 (0b1000 0000 0000 0000 0000 0010
0000 1111 0000 1011 0000 0011 1001 0100)
After Initialization (XOR with 0xFFFF) = 0b0 1111 1110 1111 1111 0100 0000 1111 0000 1101 0000 0000
0011 1001 0100
1111 1110 1111 1111 0100 0000 1111 0000 1101 0000 0000 0011 1001 010 #delete leading zeros from
previous result
1100 0000 0000 0010 1 #XOR with polynomial
0011 1110 1111 1101 1100 0000 1111 0000 1101 0000 0000 0011 1001 0100
11 1110 1111 1101 1100 0000 1111 0000 1101 0000 0000 0011 1001 0100 #delete leading zeros from
previous result
11 0000 0000 0000 101 #XOR with polynomial
00 1110 1111 1101 0110 0000 1111 0000 1101 0000 0000 0011 1001 0100
.....
.....
.....
1100 0110 0000 0010 1001 0100
1100 0000 0000 0010 1 #XOR with polynomial
0000 0110 0000 0000 0001 0100
1 1000 0000 0000 0101 00
1 1000 0000 0000 0101 #XOR with polynomial
0 0000 0000 0000 0000 00
0x0000 #verfiy that CRC checks out valid

```

---

### 备注

CRC 的结果 “0b0000 0000 0000 0000” 表示校验成功。

---

#### 6.3.6.1.1.2.2 事务帧示例

可以使用前面几节中讨论的帧结构来创建事务帧。本节概述了命令和响应帧如何通过菊花链。示例中的 CRC 值是正确的，可用于验证客户 CRC 算法。器件通过每个接收到的命令帧来验证 CRC，并且除非 CRC 有效，否则不会执行该命令。

##### 6.3.6.1.1.2.2.1 单器件读取/写入

###### 单器件读取：

使用该命令前必须先设置器件地址。单器件读取会生成一个响应帧，其长度取决于所请求读取的寄存器字节数。主机发送的命令帧必须包含起始寄存器地址（地址字段）和要返回的字节数（要读取的寄存器数）。单器件读取命令的初始化字节中的 DATA\_SIZE 字段始终为 0b000。

命令帧会传输到菊花链中的所有器件，但只有与命令帧的器件地址字段相匹配的器件才会响应单器件读取命令。相应的器件将按照响应帧格式通过返回的单器件读取请求数据进行响应。

###### 单器件写入：

使用该命令前必须先设置器件地址。利用针对单个器件的写入命令，客户能够使用一条命令更新多达八个连续寄存器。单器件写入命令帧必须包含起始寄存器地址（地址字段）和写入寄存器的数据字节。单器件写入命令的初始化字节中的 DATA\_SIZE 字段是要更新的寄存器数量。

命令帧会传输到菊花链中的所有器件，但只有与命令帧的器件地址字段相匹配的器件才会执行单器件写入命令。

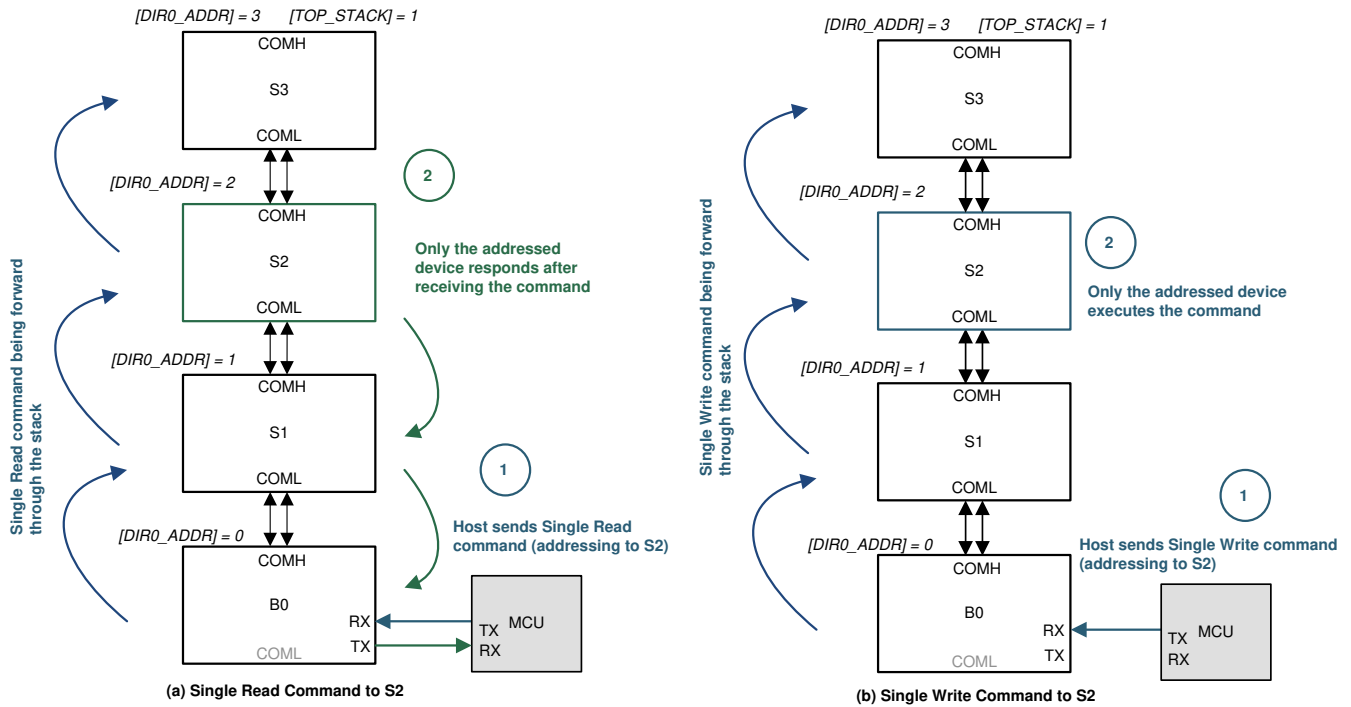


图 6-28. 单器件读取/写入

表 6-14. 单器件读取/写入

示例	主机发送的单个读取命令		主机发送的单个写入命令	
	数据	说明	数据	说明
初始化字节	0x80	始终为 0x80 FRAME_TYPE = 1 REQ_TYPE = 0b000 = 单个读取 DATA_SIZE = 0b000	0x93	0x90 用于 1 字节数据读取, 0x91 用于 2 字节数据读取, 0x92 用于 3 字节数据读取, 依此类推。 本示例中将 FRAME_TYPE = 1 REQ_TYPE = 0b001 = 单个写入 DATA_SIZE = 0b11 = 4 字节
器件地址	0x02	本例中的器件地址 0x02 (S2)	0x02	本例中的器件地址 0x02 (B0)
寄存器地址	0x0568	要读取的寄存器块的起始地址 (本例中为 VCELL16_HI 的地址)	0x0300	要写入的寄存器块的起始地址 (本例中为 OTP_PROG_UNLOCK1A 的地址)
数据	0x1F	指示目标器件返回 32 个字节的数据 (即从地址 0x0568 到 0x0587) (假设每个 VCELLn_HI = 0x80、VCELLn_LO = 0x00, 其中 n = 1 至 16)。	0x02B7 78BC	写入 OTP_PROG_UNLOCK1A 至 OTP_PROG_UNLOCK1D 的解锁值
CRC	0x5A6F		0xB8AE	

### 6.3.6.1.1.2.2.2 堆栈读取/写入

#### 堆栈读取：

在使用该命令之前，必须配置器件地址、*COMM\_CTRL[STACK\_DEV]* 位和 *[TOP\_STACK]* 位。堆栈读取命令会根据堆栈中器件的数量 (即 *COMM\_CTRL[STACK\_DEV]* = 1 的器件) 生成一定数量的响应帧，其长度取决于请求读取的寄存器字节数。堆栈读取命令帧必须包含起始寄存器地址 (地址字段) 和要返回的字节数 (要读取的寄存器数)。读取命令的初始化字节中的 *DATA\_SIZE* 字段始终为 0b000。

命令帧传输到菊花链中的所有器件，但只有  $COMM\_CTRL[STACK\_DEV] = 1$  的器件才会响应。在响应期间， $COMM\_CTRL[TOP\_STACK] = 1$  的器件会首先返回响应帧。堆栈中的每个器件（地址 N）都会等待，直到上面的器件（地址 N+1）响应后再附加其响应帧。在接收响应时验证 CRC。如果来自地址 N+1 的响应帧中出现 CRC 错误，则器件 N 不会附加其消息并生成无效 CRC 故障。

使用图 6-29，其中包含从 S1 至 S3 读取 16 个电芯电压的示例。对该命令的响应是 3 个独立的响应帧（每个器件一个响应帧），每个帧的总长度为 38 个字节（32 个数据字节 + 6 个协议字节）。尽管堆栈读取命令不包含器件地址字段，但每个响应帧会包含相应的器件地址字段，将数据与特定器件相关联。主机将首先接收来自 S3 的响应帧 (ToS)，然后接收来自 S2 的响应帧，最后接收来自 S1 的响应帧。

### 堆栈写入：

在使用该命令之前，必须配置  $COMM\_CTRL[STACK\_DEV]$ 。利用堆栈写入命令，主机能够通过一条命令更新堆栈器件（即  $COMM\_CTRL[STACK\_DEV] = 1$  的器件）的多达 8 个连续寄存器。命令帧必须包含起始寄存器地址（地址字段）和写入寄存器的数据字节。初始化帧中的  $DATA\_SIZE$  字段是要更新的寄存器的数量。

命令帧传输到菊花链中的所有器件，但只有  $COMM\_CTRL[STACK\_DEV] = 1$  的器件才会执行命令。

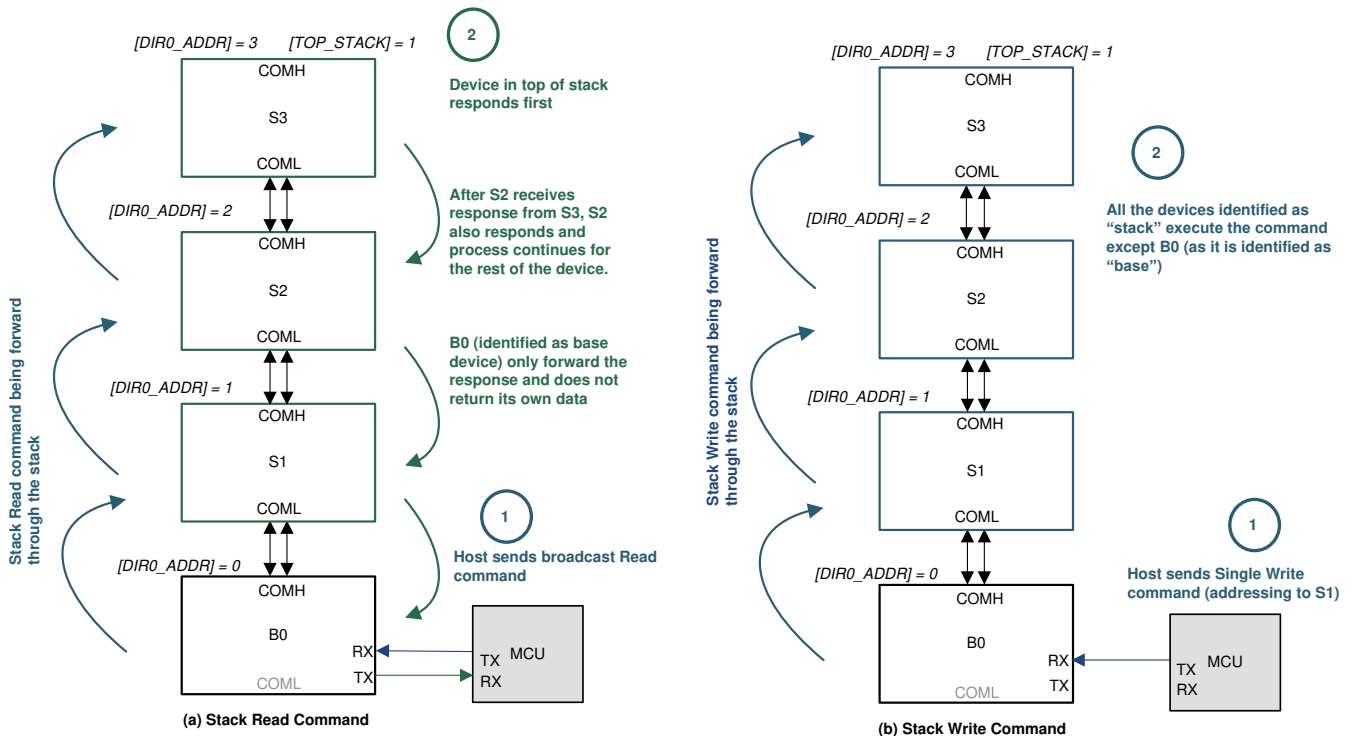


图 6-29. 堆栈读取/写入

表 6-15. 堆栈读取/写入

主机发送的堆栈读取命令			主机发送的堆栈写入命令	
示例	从 S1 至 S3 读取 16 个电芯电压		将 OTP 解锁代码写入 S1、S2 和 S3 的 OTP_PROG_UNLOCK1A 至 1D 寄存器	
帧字段	数据	说明	数据	说明
初始化字节	0xA0	始终为 0xA0 FRAME_TYPE = 1 REQ_TYPE = 0b010 = 堆栈读取 DATA_SIZE = 0b000	0xB3	0xB0 用于 1 字节数据读取, 0xB1 用于 2 字节数据读取, 0xB2 用于 3 字节数据读取, 依此类推。 本示例中将 FRAME_TYPE = 1 REQ_TYPE = 0b011 = 堆栈写入 DATA_SIZE = 0b011 = 4 字节
器件地址	不适用	无需在命令帧中添加器件地址字节	不适用	无需在命令帧中添加器件地址字节
寄存器地址	0x0568	要读取的寄存器块的起始地址 ( 本例中为 VCELL16_HI 的地址 )	0x0300	要写入的寄存器块的起始地址 ( 本例中为 OTP_PROG_UNLOCK1A 的地址 )
数据	0x1F	指示每个器件返回 32 个字节的数据 ( 即从地址 0x0568 到 0x0587 ) ( 假设每个 VCELLn_HI = 0x80、VCELLn_LO = 0x00, 其中 n = 1 至 16 ) 。	0x02B7 78BC	写入 OTP_PROG_UNLOCK1A 至 OTP_PROG_UNLOCK1D 的解锁值
CRC	0x5C2D		0x0BD7	

#### 6.3.6.1.1.2.2.3 广播读取/写入

##### 广播读取：

在使用该命令之前必须配置器件地址和 `[TOP_STACK]` 位。广播读取命令会根据菊花链中的器件数量 ( 堆栈器件和基底器件 ) 生成多个响应帧, 其长度取决于请求读取的寄存器字节数。广播读取命令帧必须包含起始寄存器地址 ( 地址字段 ) 和要返回的字节数 ( 要读取的寄存器数 )。读取命令的初始化字节中的 `DATA_SIZE` 字段始终为 0b000。

命令帧传输到菊花链中的所有器件, 每个器件都会做出响应。在响应期间, `COMM_CTRL[TOP_STACK] = 1` 的器件将首先返回响应帧, 堆栈中的每个器件 ( 地址 N ) 都会等待上面的器件 ( 地址 N+1 ) 响应后再附加其响应帧。在接收响应时验证 CRC。如果来自地址 N+1 的响应帧中出现 CRC 错误, 则器件 N 不会附加其消息并生成无效 CRC 故障。

使用表 6-16, 其中包含从 B0 至 S3 读取 16 个电芯电压的示例。对该命令的响应是 4 个独立的响应帧 ( 每个器件一个响应帧 ), 每个帧的总长度为 38 个字节 ( 32 个数据字节 + 6 个协议字节 )。尽管广播读取命令不包含器件地址字段, 但每个响应帧会包含相应的器件地址字段, 将数据与特定器件相关联。主机将首先接收来自 S3 的响应帧 (ToS), 然后接收来自 S2 的响应帧, 然后接收 S1 的响应帧, 最后接收来自 B0 的响应帧。

##### 广播写入：

可以在不采用自动寻址的情况下使用该命令。广播写入命令使主机能够通过一条命令更新菊花链中所有器件的最多八个连续寄存器。命令帧必须包含起始寄存器地址 ( 地址字段 ) 和写入寄存器的数据字节。初始化帧中的 `DATA_SIZE` 字段是要更新的寄存器的数量。

命令帧传输到菊花链中的所有器件, 菊花链中的每个器件都将执行该命令。

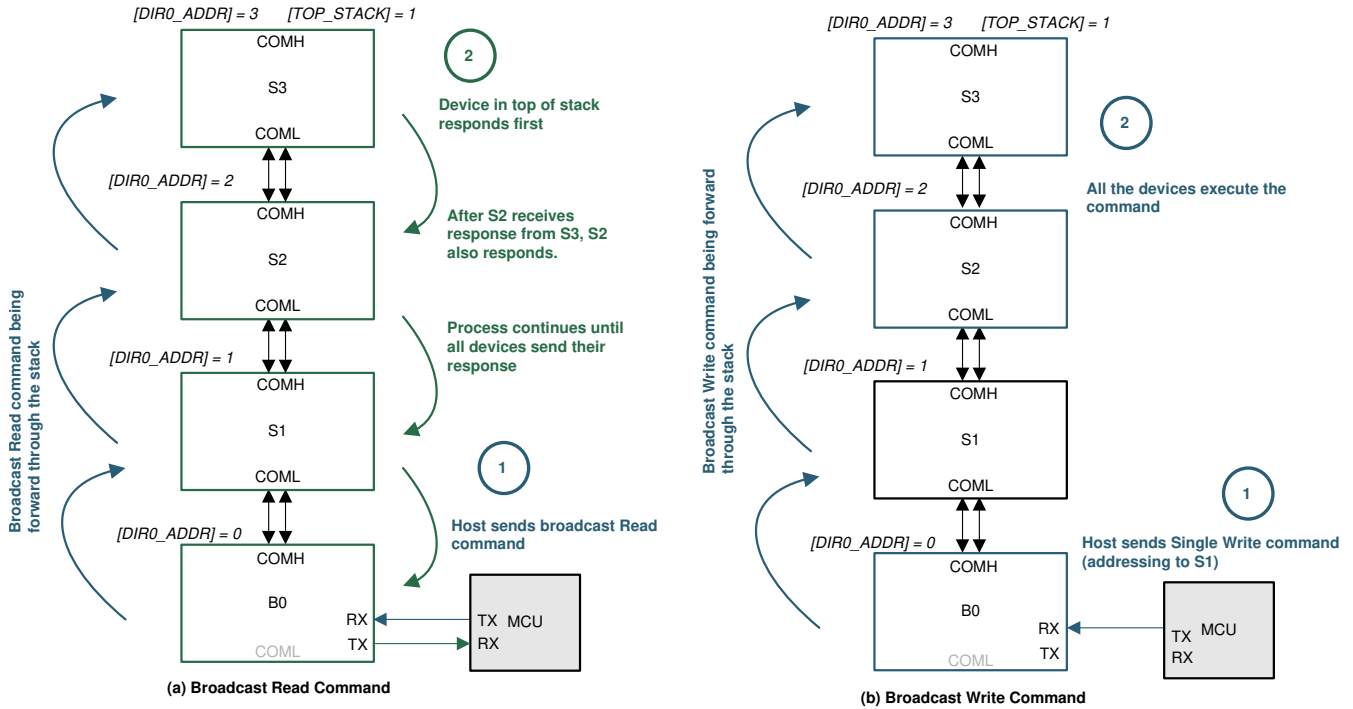


图 6-30. 广播读取/写入

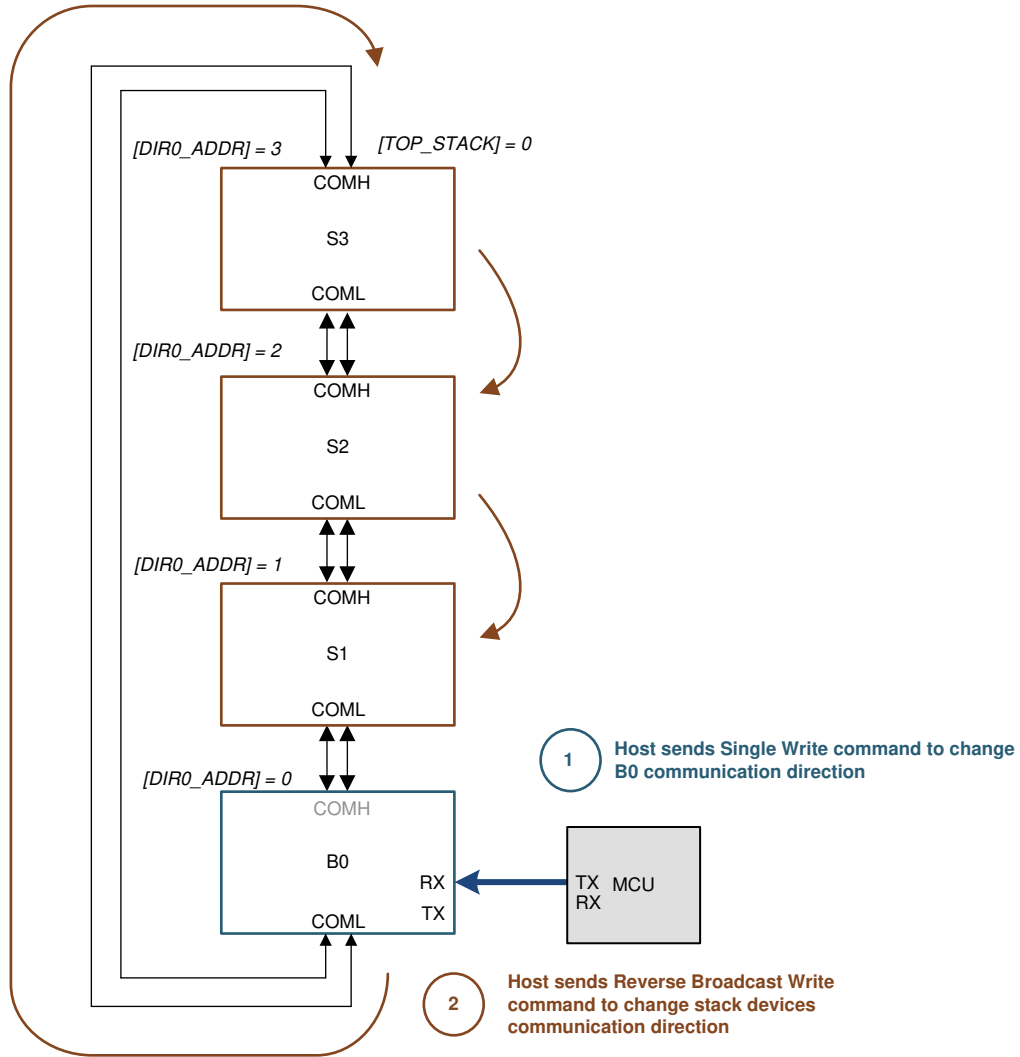
表 6-16. 广播读取/写入

主机发送的广播读取命令			主机发送的广播写入命令	
示例	从 B0 至 S3 读取 16 个电芯电压		将 OTP 解锁代码写入 B0、S1、S2 和 S3 的 OTP_PROG_UNLOCK1A 至 1D 寄存器	
帧字段	数据	说明	数据	说明
初始化字节	0xC0	始终为 0xC0 FRAME_TYPE = 1 REQ_TYPE = 0b100 = 广播读取 DATA_SIZE = 0b000	0xD3	0xD0 用于 1 字节数据读取, 0xD1 用于 2 字节数据读取, 0xD2 用于 3 字节数据读取, 依此类推。 本示例中将 FRAME_TYPE = 1 REQ_TYPE = 0b101 = 广播写入 DATA_SIZE = 0b011 = 4 字节
器件地址	不适用	无需在命令帧中添加器件地址字节	不适用	无需在命令帧中添加器件地址字节
寄存器地址	0x0568	要读取的寄存器块的起始地址 (本例中为 VCELL16_HI 的地址)	0x0300	要写入的寄存器块的起始地址 (本例中为 OTP_PROG_UNLOCK1A 的地址)
数据	0x1F	指示每个器件返回 32 个字节的数据 (即从地址 0x0568 到 0x0587) (假设每个 VCELLn_HI = 0x80、VCELLn_LO = 0x00, 其中 n = 1 至 16)。	0x02B7 78BC	写入 OTP_PROG_UNLOCK1A 至 OTP_PROG_UNLOCK1D 的解锁值
CRC	0x422D		0x6BD1	

6.3.6.1.1.2.2.4 广播写入反向

通常, 器件希望根据 [DIR\_SEL] 设置接收通信。如果器件接收到与 [DIR\_SEL] 设置相反的通信帧 (例如在 [DIR\_SEL] = 0 时从 COMH 接收命令帧), 则会将通信标记为错误。广播写入反向是用于在主机需要切换菊花链通信方向时翻转 [DIR\_SEL] 设置的命令。在反向通信方向过程中, 该命令应从与 [DIR\_SEL] 设置相反的方向接收。有关详细信息, 请参阅节 6.3.6.1.3.4。

虽然广播写入反向允许向器件写入任何寄存器值, 但不建议写入除 CONTROL1[DIR\_SEL] 之外的任何其他寄存器设置, 以避免通信冲突。不会检测通信冲突, 从而导致堆栈接口上的通信损坏。



Reverse Broadcast Write Command

图 6-31. 广播写入反向

表 6-17. 广播写入反向

主机发送的广播写入反向命令		
在菊花链中的所有器件上设置 [DIR_SEL] = 1		
帧字段	数据	说明
初始化字节	0xE0	始终为 0xE0 FRAME_TYPE = 1 REQ_TYPE = 0b110 = 广播写入反向 DATA_SIZE = 0b000
器件地址	不适用	无需在命令帧中添加器件地址字节
寄存器地址	0x0309	CONTROL1 寄存器的地址
数据	0x80	设置 CONTROL1[DIR_SEL] = 1
CRC	0xC014	

### 6.3.6.1.2 菊花链接口

菊花链通信是使用差分信号创建的，以最大限度地降低电磁敏感性 (EMS) 和增强大电流注入 (BCI) 抗扰度。差分通信分别在 COM\*P 和 COM\*N 引脚上传输原码数据和补码数据。在多器件堆栈中，存在一些配置，其中器件物理上位于同一块板上，或者位于通过双绞线连接的完全独立的电池组中。

该器件支持使用变压器或电容器来电气隔离堆栈中器件之间的信号。对于在同一 PCB 上有多个器件的应用，器件的 COMH/L 引脚之间连接了一个电平转换电容器。对于噪声极大的环境，可能需要额外的滤波。对于通过电缆分隔的器件，需要使用额外的隔离元件。有关选择元件的具体详细信息，请参阅节 7。

#### 6.3.6.1.2.1 菊花链发送器和接收器功能

菊花链是双向和半双工的，因此在 COMH 和 COML 接口上有一个发送器 (TX) 和一个接收器 (RX)。TX 和 RX 功能由硬件根据器件的基底/堆栈检测自动控制。当接收到 WAKE ping/音调时，通信方向由 CONTROL1[DIR\_SEL] 和 COMM\_CTRL[TOP\_STACK] 配置进行设置。有关详细信息，请参阅节 6.3.6.1.3。此外，用户可以使用 DEBUG\_CTRL\_UNLOCK、DEBUG\_COMM\_CTRL1 和 DEBUG\_COMM\_CTRL2 寄存器在通信调试模式下进行覆盖以接管 COMH 和 COML 的完全控制。有关详细信息，请参阅节 6.5.4.14。

#### 6.3.6.1.2.2 菊花链协议

差分菊花链 (垂直) 接口使用异步 13 位字节传输协议。数据首先传输 LSB，然后复制每一位 (采用补码形式) 以确保传输中没有直流成分。

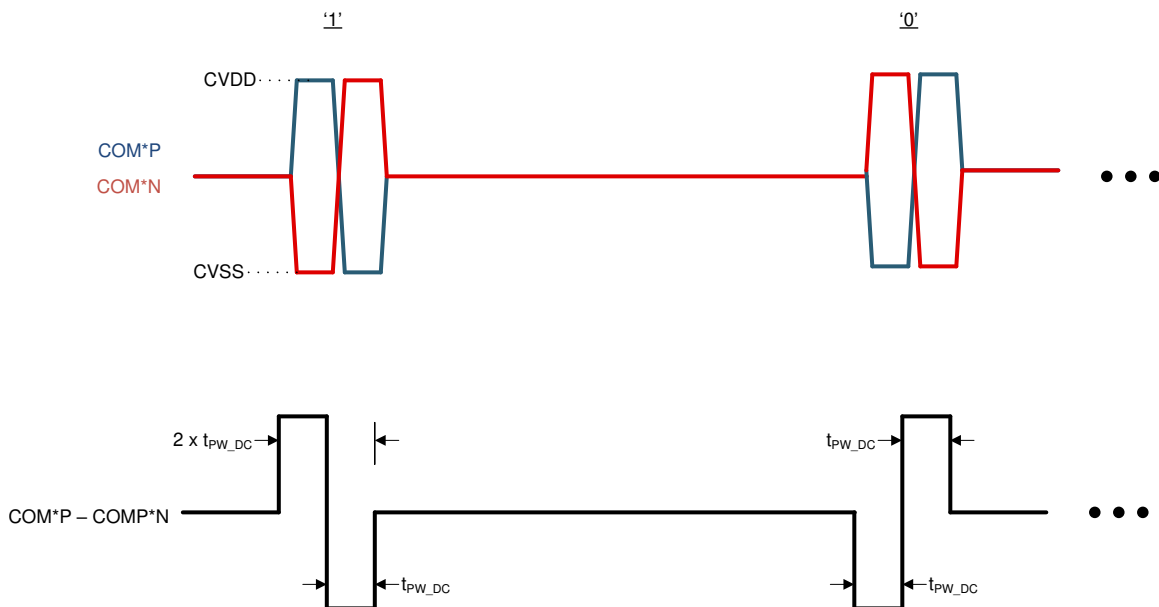


图 6-32. 菊花链位定义

一个字节以前导码开始，随后是两个 SYNC 位、一个帧起始位、从 LSB D0 到 MSB D7 的八个数据位 (D0 在帧状态之后传输，D7 最后传输，其后是错误和后同步码)。

器件使用前导码和 SYNC 位来提取时序信息，以解码字节中其余的位值。如果检测到以下任何错误，则不会处理该字节并设置寄存器错误位。

- 前导码和 SYNC 位是已知值，如果解码的值有错误，则  $DEBUG\_COMH/L\_BIT[SYNC1] = 1$  取决于哪个 COM 端口接收该数据。
- 如果从前导码和 SYNC 位提取的时序超出预期范围，则  $DEBUG\_COMH/L\_BIT[SYNC2] = 1$ 。

接收到两个有效的 SYNC 位后，附加位就会被解码并发送到命令处理器。器件继续检测该字节上的任何错误，如果检测到错误，则会在该字节中设置字节错误 (BERR) 位。  $DEBUG\_COMH/L\_BIT[PERR] = 1$  取决于哪个 COM 端口检测到错误。以下条件将设置字节中的 BERR 位。

- 没有足够的样本来指示某个位的逻辑电平。也就是说，某个位既没有被解码为强 1，也没有被解码为强 0。  
 $DEBUG\_COMH/L\_BIT[BIT] = 1$  取决于哪个 COM 端口检测到错误。

与此同时，每个位仍在重新传输到下一个器件。如果器件无法将该位解码为 1 或 0，它将以 0 重新传输，并在该字节中设置 BERR 位。当新器件检测到接收字节中 BERR 位被设置为 1 时，它将忽略有问题的字节并设置  $DEBUG\_COMH/L\_BIT[BERR\_TAG] = 1$ ，指示接收到的字节具有 BERR。被忽略的可疑字节可能会导致其他通信错误，并且还可能会触发在新器件中设置  $DEBUG\_COMH/L\_BIT[PERR] = 1$ 。可疑字节继续在设置 BERR 的情况下沿菊花链重新传输，该过程持续进行。

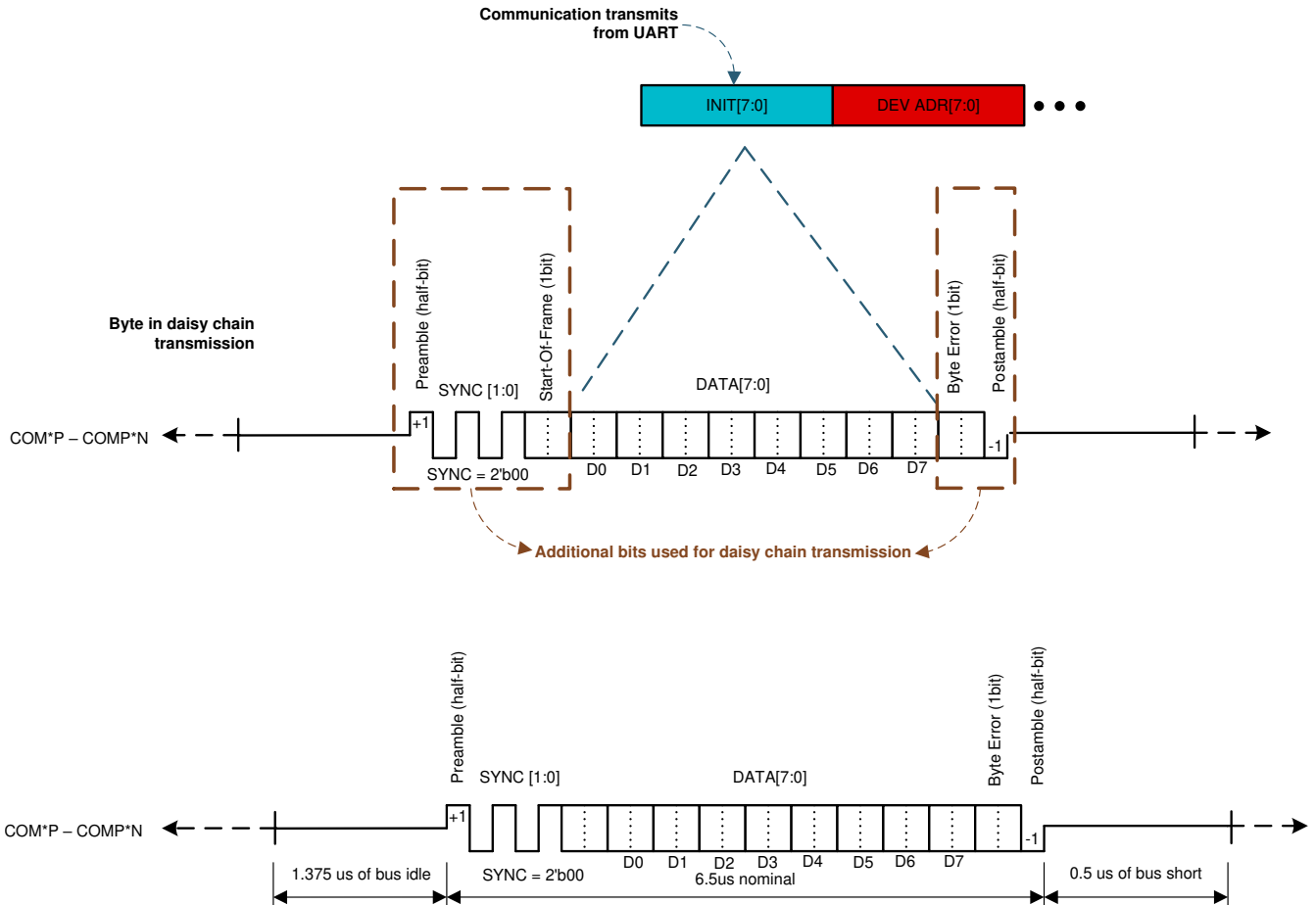


图 6-33. 菊花链字节定义

表 6-18. 菊花链字节定义

位域	说明
前导码 (半位)	表示事务开始，向接收器发出开始采样的信号。该半位和随后的两个 SYNC 位用于额外的时序信息。
SYNC[1:0]	始终为 0b00。SYNC 位用于数字电路评估字节上的时序和噪声水平，从而改进噪声环境中 1 和 0 的检测。
帧起始 (1 位)	帧起始 (SOF) 位指示后继数据字节是初始化学字节，即通信事务帧的开始。堆栈器件需要使用该信息来处理通信。对于命令帧事务，基底器件负责设置 SOF 位，因为它将 UART 通信转换为菊花链通信。初始化学字节包含数据大小信息。基底器件将根据数据大小信息来计算接收到的字节数并相应地设置下一个 SOF 位。UART COMM CLEAR 信号重置 UART 接收器，其中包括逻辑的帧处理。因此，COMM CLEAR 之后的下一个字节必须将 SOF 设置为 1，因为 COMM CLEAR 指示系统清除 UART 并重新启动通信。
Data[7:0]	通信事务帧的实际字节

表 6-18. 菊花链字节定义 (续)

位域	说明
字节错误 BERR (1 位)	指示在该字节中检测到错误。当器件接收到下部器件设置了 BERR 的字节时，它也会在设置 BERR = 1 的情况下重新传输该字节。 由于每个数据位都会从一个器件重新计时到下一个器件，因此下一个器件可能无法检测到通信错误。然而，[BERR] 位的标签会指示该通信帧在其先前的事务期间产生错误。
后同步码 (半位)	指示事务结束

每个字节以 2MHz (每个脉冲 250ns, 或每对 500ns) 的频率传输。两个字节之间的传输时间取决于 UART 波特率 (正常运行状态下为 1Mbps), 但字节传输时间始终相同。通信帧是用字节之间的空闲时间定义的。在某些极少数情况下, 通信信号可能无法完全终止, 从而在一个字节末尾留下振铃。在这种情况下, 增加字节间的间隙可以提高通信的稳健性。器件允许通过 `STACK_RESPONSE` 寄存器设置在响应帧中的字节之间插入额外的字节间隙。

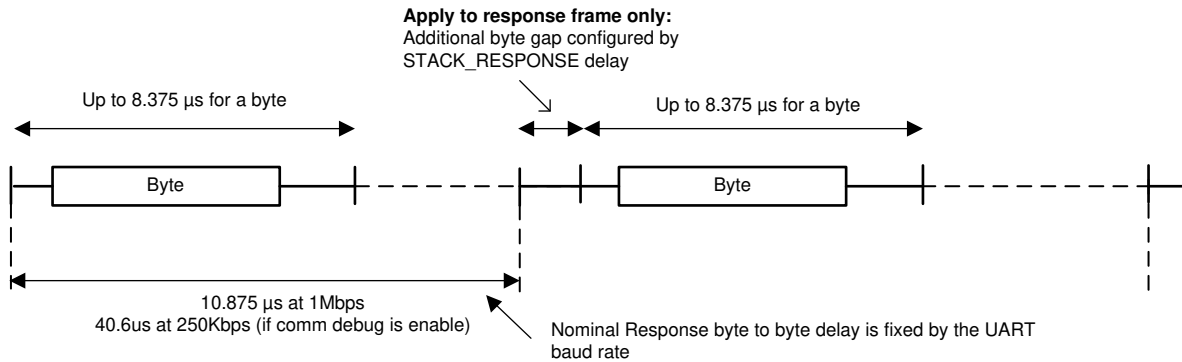


图 6-34. 菊花链字节传输

### 6.3.6.1.3 开始通信

从 SHUTDOWN 模式或器件复位后, 主机按照以下步骤启动器件进行通信。

- 主机发送 WAKE ping 以重置器件或使器件进入 ACTIVE 模式。在该过程中, 菊花链中的器件将根据其在菊花链中的位置 (基底器件或堆栈器件) 来配置其自己的 COMH 和 COML 端口。
  - 完成该步骤后, 即可支持广播写入。
- 主机执行自动寻址, 从而为每个器件分配器件地址
  - 完成该步骤后, 即可支持广播读取/写入和单器件读取/写入。
- 主机配置 `COMM_CTRL[STACK_DEV]` 和 `[TOP_STACK]` 位。栈顶 (ToS) 器件将禁用其 COMH (或 COML, 取决于通信方向) 的发送器
  - 完成该步骤后, 即可支持所有命令, 即广播读取/写入、单器件读取/写入、堆栈读取/写入。

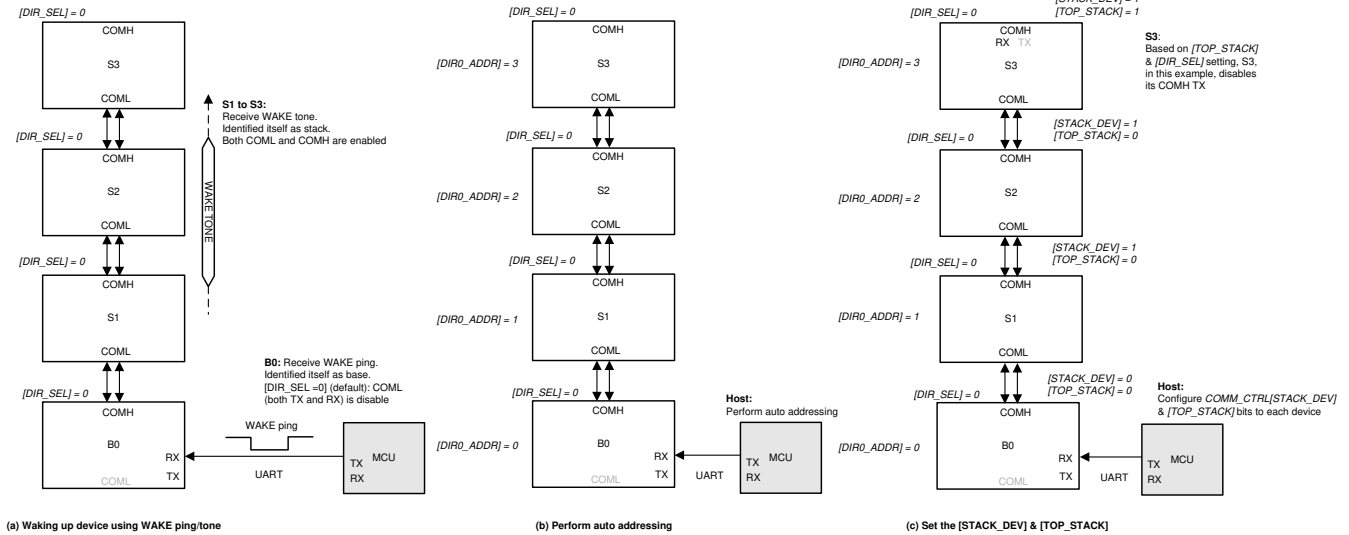


图 6-35. 配置器件进行通信

### 6.3.6.1.3.1 识别基底和堆栈

器件使用 WAKE ping/音调来识别其在菊花链中的位置。

- 基底器件：通过 UART 与主机连接的器件
- 堆栈器件：通过 COMH 和 COML 与基底器件连接的器件

基底器件会通过 RX 引脚由 WAKE ping 唤醒，而堆栈器件会通过 COMH/COML 端口由 WAKE 音调唤醒。因此，器件使用 WAKE ping 或 WAKE 音调来识别其自身为基底器件或堆栈器件。该信息存储在 AVAO\_REF 块中，该块在所有功耗模式下均可用，并且每当收到 WAKE ping/音调时都会刷新。

通过使用 CONTROL1[DIR\_SEL] 设置，基底器件将禁用未使用的菊花链端口（发送器和接收器）。如果主机更改 CONTROL1[DIR\_SEL] 设置，基底器件将重新配置其 COMH/COML。

#### 备注

主机在更改 [DIR\_SEL] 设置后至少 100μs 开始通信，以确保器件完成 COMH/COML 重新配置。

### 6.3.6.1.3.2 自动寻址

每个器件都必须具有唯一的器件地址才能使读取协议正常工作。如果出于任何原因两个器件分配了相同的器件地址，则广播和堆栈读取可能不起作用。此外，单个器件读取双地址会导致通信被破坏。

假设 OTP 中的器件地址未编程，则默认器件地址为 0x00。为了使主机与独立器件（仅包含一个器件的堆栈）通信，主机只需使用默认的 0x00 器件地址。否则，器件地址遵循以下规则：

- 器件基址可以从任何值开始，不一定是 0x00。
- 所有器件地址都必须按顺序的。也就是说，如果基址是 0x00，则下一个器件的地址必须是 0x01，下一个器件的地址必须是 0x02，依此类推。

在开始自动寻址过程之前，所有器件必须处于 ACTIVE 模式。在该状态下，器件只能处理广播写入命令，该命令将用于自动寻址过程。根据 CONTROL1[DIR\_SEL] 设置，自动寻址过程会将器件地址设置为 DIR0\_ADDR 寄存器（当 [DIR\_SEL] = 0 时）或 DIR1\_ADDR 寄存器（当 [DIR\_SEL] = 1 时）。

#### 6.3.6.1.3.2.1 设置器件地址

CONTROL1[ADDR\_WR] 位启用自动寻址模式。在该模式下，器件会关闭其 COMH/COML（取决于 [DIR\_SEL] 设置）发送器并持续一个通信帧的时长（遵循自动寻址过程，这将是其自身器件的地址），清除

$CONTROL1[ADDR\_WR] = 0$ 。当接收到下一个通信时（遵循自动寻址过程，这将是下一个器件的地址），该器件会将通信转发至下一个器件。

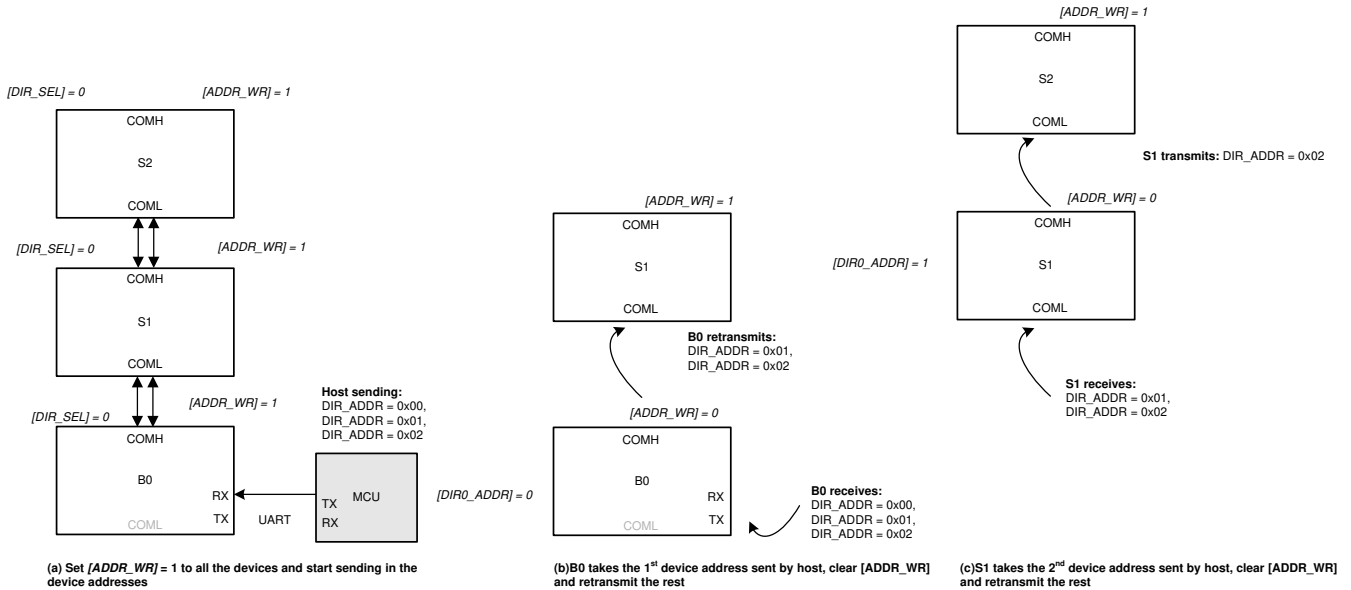


图 6-36. 自动寻址

6.3.6.1.3.2.2 设置  $COMM\_CTRL[STACK\_DEV]$  和  $[TOP\_STACK]$

自动寻址的最后一个步骤是配置  $COMM\_CTRL[STACK\_DEV]$  和  $[TOP\_STACK]$  设置。必须配置这些位才能使广播读取和堆栈读取/写入正常运行。

- 基底器件： $[STACK\_DEV] = 0$  且  $[TOP\_STACK] = 0$
- 堆栈器件（ToS 器件除外）： $[STACK\_DEV] = 1$  且  $[TOP\_STACK] = 0$
- ToS 器件： $[STACK\_DEV] = 1$  且  $[TOP\_STACK] = 1$

假设  $CONTROL1[DIR\_SEL] = 0$ （即每个器件设置为将主机发送的命令帧从其 COML 传输到 COMH），表 6-19 展示了自动寻址步骤。

表 6-19. 自动寻址

步骤	过程
1	如果在执行自动寻址过程之前发生了器件复位，则需要执行该步骤。 虚拟写入，以在写入方向上同步所有菊花链器件 DLL（延迟锁定环）斜坡。 主机发送广播写入以将 0x00 写入 $ECC\_DATA1$ 至 $ECC\_DATA8$ 寄存器。
2	启用自动寻址过程。 主机发送广播写入以设置 $CONTROL1[ADDR\_WR] = 1$ 。
3	发送器件地址。主机发送广播写入以将连续地址设置为 $DIR0\_ADDR[ADDRESS5:0]$ 。以下是菊花链中共包含三个器件的示例： 1. 使用数据 0x00 将广播写入发送到 $DIR0\_ADDR$ 寄存器。 2. 使用数据 0x01 将广播写入发送到 $DIR0\_ADDR$ 寄存器。 3. 使用数据 0x02 将广播写入发送到 $DIR0\_ADDR$ 寄存器。
4	为每个器件设置 $COMM\_CTRL[STACK\_DEV]$ 和 $[TOP\_STACK]$ 位。 选项 1：主机向每个器件发送单个器件写入，以设置正确的 $[STACK\_DEV]$ 和 $[TOP\_STACK]$ 值。 选项 2（较少的通信步骤）： 1. 主机发送广播写入以设置 $[STACK\_DEV] = 1$ 和 $[TOP\_STACK] = 0$ 。 2. 主机通过设置 $[STACK\_DEV] = 0$ ，向基底器件（本例中为器件地址 0x00）发送单个器件写入。 3. 主机通过设置 $[TOP\_STACK] = 1$ ，向 ToS 器件（本例中为地址 0x02）发送单个器件写入。

表 6-19. 自动寻址 (续)

步骤	过程
5	如果在执行自动寻址过程之前发生了器件复位,则需要执行该步骤。 虚拟读取,以同步读取方向上的所有菊花链器件 DLL 斜坡。 主机发送广播读取以读取 <i>ECC_DATA1</i> 至 <i>ECC_DATA8</i> 寄存器。主机可能不会接收所有数据,因为该步骤会同步 DLL。
7	建议采用这种方法。使用广播读取来读取 <i>DIR0_ADDR</i> 寄存器,以读回所有器件地址,从而确保所有器件都被正确寻址。
8	如果执行虚拟写入和虚拟读取步骤来同步 DLL,则触发通信故障是正常情况。如果发生这种情况,请清除故障寄存器。

#### 6.3.6.1.3.2.3 将器件地址存储到 OTP

器件使用 *DIR0\_ADDR* (在  $[DIR\_SEL] = 0$  时使用) 和 *DIR1\_ADDR* (在  $[DIR\_SEL] = 1$  时使用) 寄存器来保存其器件地址。在自动寻址过程中,器件地址被写入这些寄存器之一,新的器件地址立即生效。

主机可以选择将  $[DIR\_SEL] = 0$  和 1 方向的器件地址编程到 OTP 中,从而允许在器件复位时加载已编程的地址。为了将器件地址编程到 OTP 中,主机将所需的地址写入 OTP 影子寄存器 *DIR0\_ADDR\_OTP* (在  $[DIR\_SEL] = 0$  时使用) 和 *DIR1\_ADDR\_OTP* (在  $[DIR\_SEL] = 1$  时使用) 并执行 OTP 编程。这两个影子寄存器仅反映在 OTP 中编程的值,或供主机用于将所需的值编程到 OTP 中。这两个影子寄存器不是通信期间的器件地址设置。有关编程详细信息,请参阅节 6.3.6.3.2。

#### 6.3.6.1.3.3 同步菊花链 DLL

当器件复位或从 SLEEP 模式进入 ACTIVE 模式时,MCU 执行虚拟写入和读取以同步菊花链器件上的 DLL。

在器件复位情况下,如果未在 OTP 中对器件地址进行编程,则 MCU 必须执行自动寻址。DLL 同步是该步骤的一部分。如果在 OTP 中对器件地址进行了编程,则器件复位后无需自动寻址。然而,MCU 应执行表 6-19 所示的虚拟写入和虚拟读取步骤(步骤 1 和步骤 5)来同步 DLL。

当器件使用 SLEEPtoACTIVE 信号从 SLEEP 模式进入 ACTIVE 模式时,器件不会复位。但是,建议执行 1 数据字节的虚拟写入和读取,以确保稳健性。执行表 21 中类似的虚拟写入和读取步骤,但仅对 *OTP\_ECC\_DATAIN1* 进行写入和读取。

#### 6.3.6.1.3.4 环形通信

该器件的菊花链通信采用环形架构。在该架构中,两个器件之间的电缆断开不会像正常的非环形方案那样阻止与所有上游器件的通信。当主机检测到通信中断时,该器件允许主机切换通信方向,以便与中断点两侧的器件进行通信。这样可以确保安全运行,直到线路中的中断点被修复。

*CONTROL1[DIR\_SEL]* 控制通信方向。器件将根据  $[DIR\_SEL]$  和  $[TOP\_STACK]$  设置重新配置 COMH 和 COML 端口。需要执行自动寻址过程来重新寻址反向通信方向的器件地址。

下面是一个针对整个菊花链将通信方向更改为  $[DIR\_SEL] = 1$  的示例:

1. 主机发送单个器件写入以更改基底器件  $[DIR\_SEL] = 1$ 。基底器件将禁用其 COMH 并启用其 COML。
2. 主机发送广播写入反向以清除所有器件上的 *COMM\_CTRL* 寄存器设置。
3. 主机发送广播写入反向以更改其余器件的  $[DIR\_SEL] = 1$ 。在该步骤中,整个菊花链设置为在  $[DIR\_SEL] = 1$  方向上传输通信(即每个器件设置为将主机发送的命令帧从其 COMH 传输到其 COML)。
4. 主机执行自动寻址过程,以在 *DIR1\_ADDR* 寄存器中设置器件地址。除非器件已复位,否则主机可以跳过虚拟读取/写入步骤以在自动寻址过程中同步 DLL。
5. 主机设置新的栈顶器件,新的 ToS 设备将禁用其 COML 发送器。

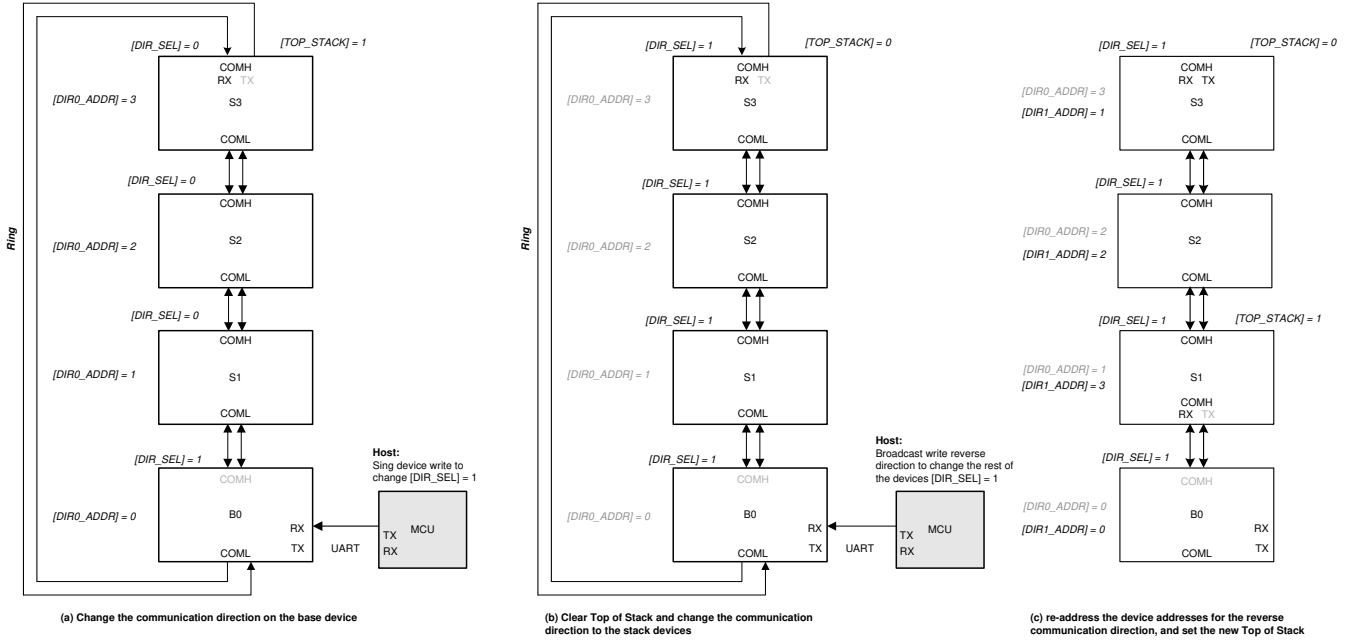


图 6-37. 更改菊花链中的通信方向的示例

设置两个通信方向的器件地址之后，主机可以在切换通信方向时跳过自动寻址步骤。

当电缆损坏时，主机遵循相同的过程来更改通信方向。要访问菊花链中的所有器件，主机必须在菊花链的某些器件上  $[DIR\_SEL] = 0$  以及其他器件上  $[DIR\_SEL] = 1$  的情况下通信。菊花链还将有两个 ToS 器件，每个通信方向一个。

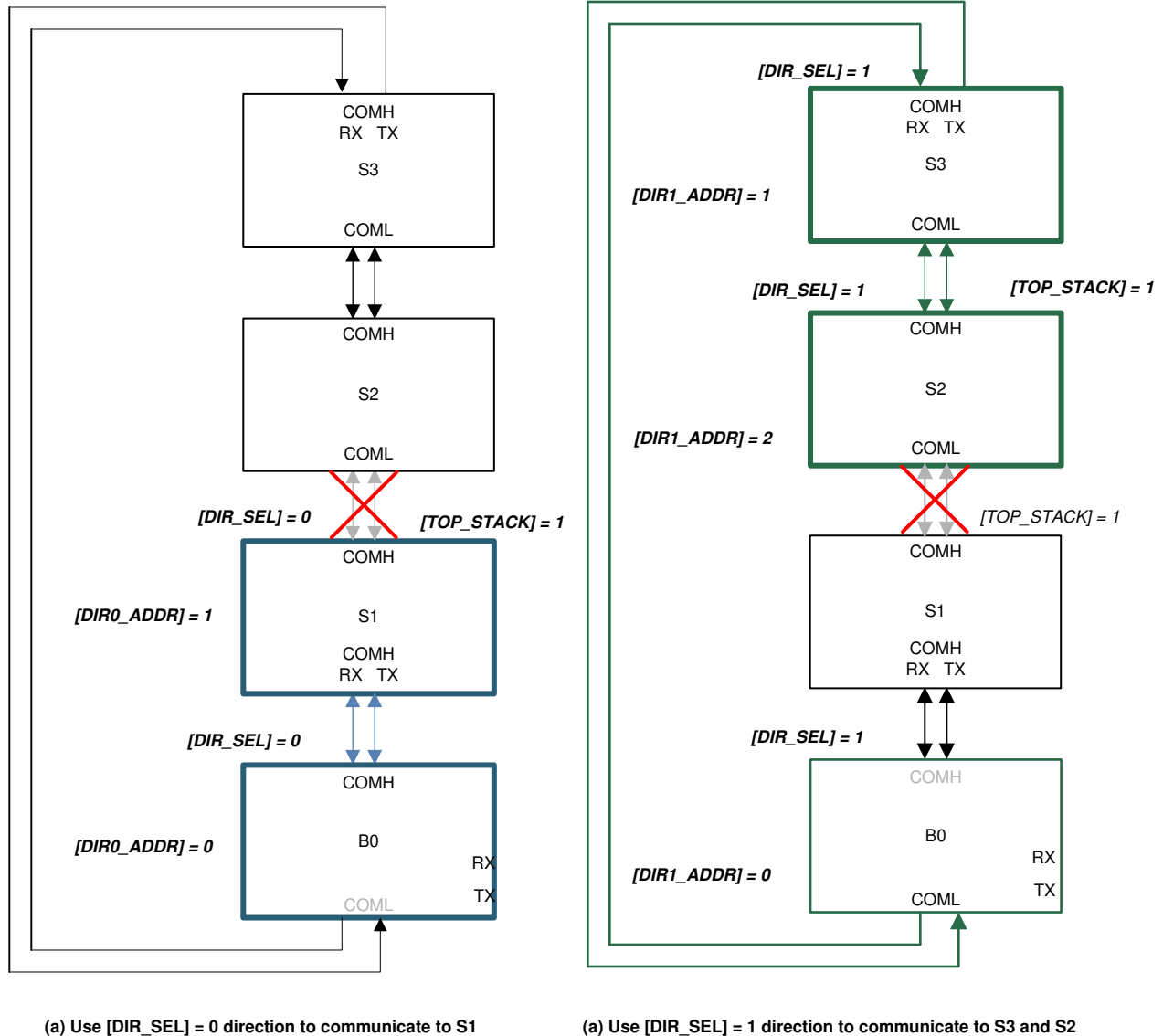


图 6-38. 使用环形架构在电缆断开的情况下访问所有器件

#### 6.3.6.1.4 通信超时

有两个可编程通信超时阈值，即 CTS 计时器和 CTL 计时器，它们用于监测 UART 或菊花链通信中是否存在有效帧。有效帧定义为任何不包含任何阻止帧被处理的错误的帧（响应或命令）。通信超时仅在 ACTIVE 模式下才会主动计数。计数器在 SHUTDOWN 模式期间被禁用和重置。在 SLEEP 模式下，最后的计数器值保持冻结。

##### 6.3.6.1.4.1 短通信超时

短通信超时在触发时用作向主机发出的警报。超时周期可通过 `COMM_TIMEOUT_CONF[CTS_TIME2:0]` 位进行编程。如果启用，则每次收到有效响应或命令帧时都会重置计时器。如果计时器到期，则会设置 `FAULT_SYS[CTS]` 位。

##### 6.3.6.1.4.2 长通信超时

长通信超时允许主机将器件置于 SLEEP 或 SHUTDOWN 模式以省电。超时周期可通过 `COMM_TIMEOUT_CONF[CTL_TIME2:0]` 位进行编程。如果启用，则每次收到有效响应或命令帧时都会重置计时器。如果计时器到期，则主机可以通过 `COMM_TIMEOUT_CONF[CTL_ACT]` 位选择以下操作之一。

- 设置 `FAULT_SYS[CTL] = 1` 并进入 SLEEP 模式。
- 进入 SHUTDOWN 模式。

#### 6.3.6.1.5 通信调试模式

该器件提供通信调试模式以简化初始开发阶段。为了进入该调试模式，主机将解锁代码 `0xA5` 写入寄存器 `DEBUG_CTRL_UNLOCK`。调试模式解锁后，`DEBUG_COMM_CTRL1` 和 `DEBUG_COMM_CTRL2` 中的设置生效。

要退出调试模式，将除 `0xA5` 之外的任何值写入 `DEBUG_CTRL_UNLOCK` 即可（例如写入 `0x00`）。无论 `DEBUG_COMM_CTRL1` 和 `DEBUG_COMM_CTRL2` 寄存器中的设置如何，COMH、COML 和 UART 都将返回至其正常运行状态。

进入通信调试模式后，主机将获得以下控制权：

表 6-20. 通信调试模式功能

控制功能	使能位	说明
完全 COMH/L 发送器和接收器控制	<code>[USER_DAISSY_EN]</code>	如果 <code>[USER_DAISSY_EN] = 1</code> ，则器件会根据 <code>DEBUG_COMM_CTRL2</code> 寄存器设置来启用或禁用其 COMH/L 发送器和接收器。 如果 <code>[USER_DAISSY_EN] = 0</code> ，即使在通信调试模式下，COMH/L 也会处于正常运行状态。
将菊花链中的数据镜像到 UART 中	<code>[USER_UART_EN]</code>	如果 <code>[USER_UART_EN] = 1</code> ，则主机可设置 <code>[UART_MIRROR_EN] = 1</code> 来指示器件将菊花链转换到 UART，从而使主机能够从 UART 接口读取菊花链中接收或转发的数据。数据将以 UART 通信帧格式呈现。 对于堆栈器件，默认情况下禁用 UART TX。为了使用该功能，主机还设置 <code>[UART_TX_EN] = 1</code> 。 如果 <code>[USER_UART_EN] = 0</code> ，则会禁用任何与 UART 相关的调试功能。无论 <code>[UART_MIRROR_EN]</code> 和 <code>[UART_TX_EN]</code> 设置如何，UART 都将处于正常运行状态。
将 UART 波特率降至 250kbps	<code>[USER_UART_EN]</code>	如果 <code>[USER_UART_EN] = 1</code> ，则主机可以设置 <code>[UART_BAUD] = 1</code> 以将 UART 波特率更改为 250kbps。这将导致菊花链上的吞吐量较慢。 如果 <code>[USER_UART_EN] = 0</code> ，则无论 <code>[UART_BAUD]</code> 设置如何，UART 波特率都将保持在 1Mbps。

`DEBUG_COMM_STAT` 寄存器具有指示 UART 和 COMH/L 是受用户还是硬件（器件）控制的状态位。该寄存器还指示 COMH/L 发送器和接收器的状态。该调试状态寄存器根据器件状态进行更新，并且在启用或不启用通信调试模式的情况下都可读。

事实上，在未启用通信调试模式的情况下，只读调试寄存器在 ACTIVE 模式下都是可读的。这其中的大多数寄存器是较低级别的通信故障状态寄存器，用于在通信故障事件中提供额外信息，例如 `DEBUG_UART*`、`DEBUG_COMH*` 和 `DEBUG_COML*` 寄存器。请参阅节 6.3.6.2 和节 6.5.4 了解更多详细信息。

#### 6.3.6.1.6 多点配置

多点配置是系统中多个器件通过 UART 与主机系统通信的配置。器件之间不存在菊花链通信。当 `[MULTIDROP] = 1` 时，器件 COMH 和 COML 端口被禁用。与菊花链配置（即 `[MULTIDROP] = 0`）中一样，仍然支持所有通信协议、单器件读取/写入、广播读取/写入、堆栈读取/写入和反向广播写入。然而，在多点配置中，不太可能使用堆栈和反向广播命令。如果使用了广播命令，则仍然需要设置器件地址连续的器件，并在器件地址最高的器件上设置 `[TOP_STACK]` 位。与菊花链通信中一样，`[TOP_STACK] = 1` 的器件在接收到广播读命令时启动数据返回，器件地址较低的器件接下来会响应。此外，在多点配置中，必须在每个帧之前使用 `COMM_CLR` 以确保一致的通信。

#### 6.3.6.1.7 SPI 控制器

当 `GPIO_CONF1[SPI_EN] = 1` 时，GPIO4 至 GPIO7 可配置为 SPI 控制器接口。SPI 控制器包含四个 I/O：

- SCLK：SPI 时钟，由器件生成，用于同步
- MOSI：控制器数据输出，由器件驱动向目标输出数据
- MISO：控制器数据输入，检测来自目标的数据

- SS：目标选择，在 SPI 通信期间由器件驱动。

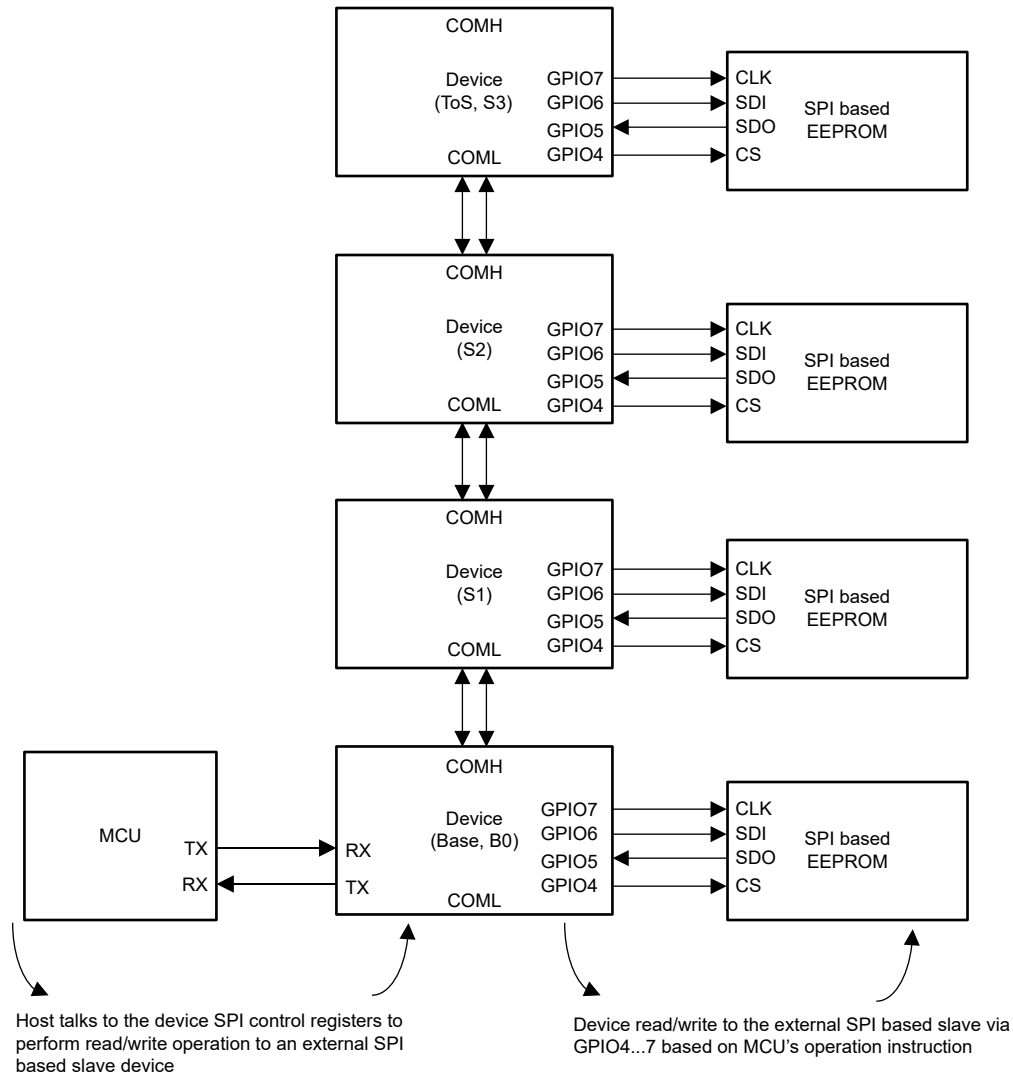


图 6-39. SPI 控制器堆栈配置

$SPI\_CONF[CPOL]$  (时钟极性) 和  $[CPHA]$  (时钟相位) 定义 SPI 时钟格式。在 SPI 时钟反相或同相时会定义  $[CPOL]$ 。如果在前 (第一个) 时钟沿或后 (第二个) 时钟沿对 MISO 和 MOSI 进行采样, 则定义  $[CPHA]$ , 无论该时钟沿是上升沿还是下降沿都是如此。 $SPI\_CONF[NUMBER4:0]$  定义该事务有多少位 (1 位至 24 位事务)。

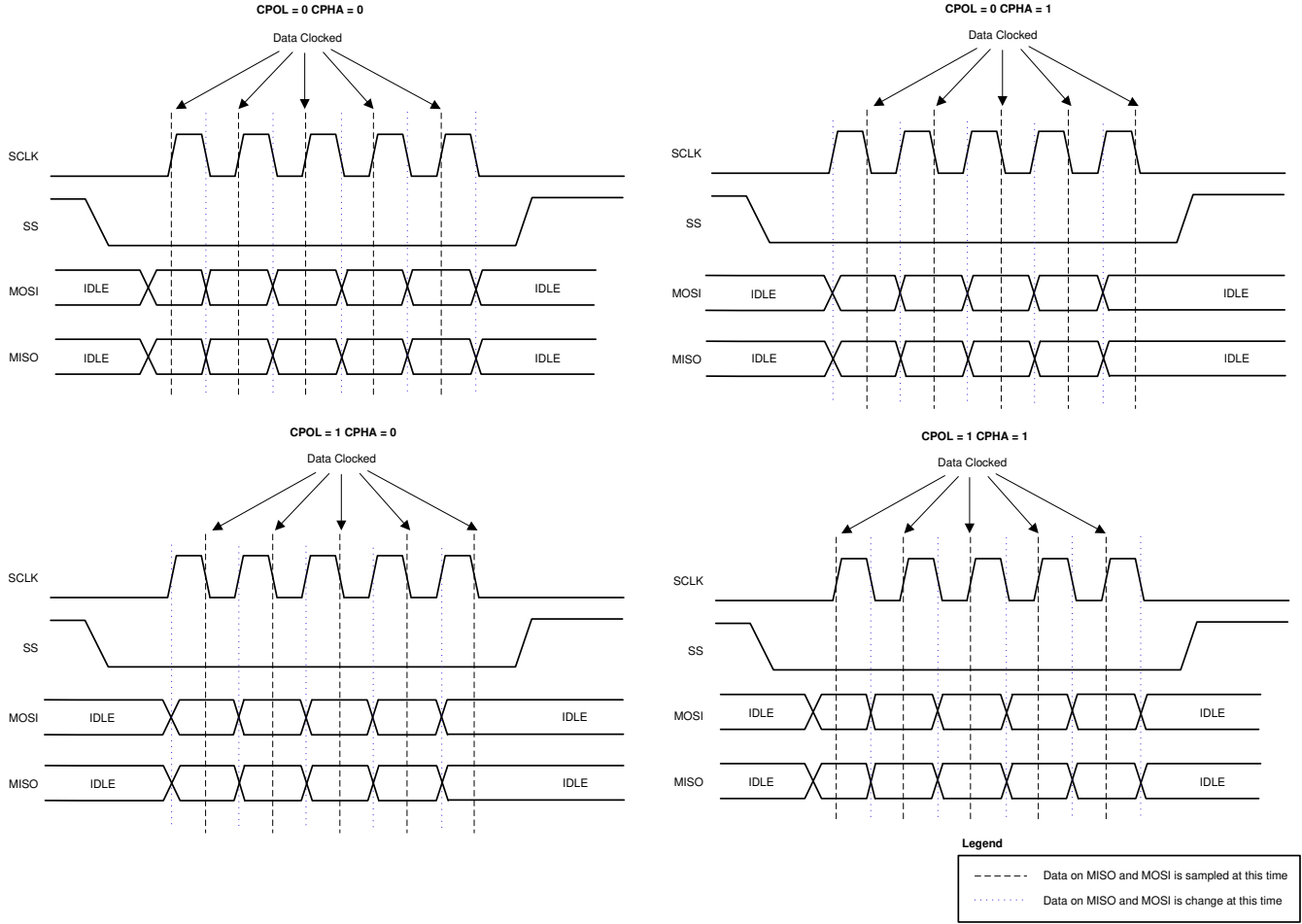


图 6-40. SPI 控制器 CPOL 和 CPHA

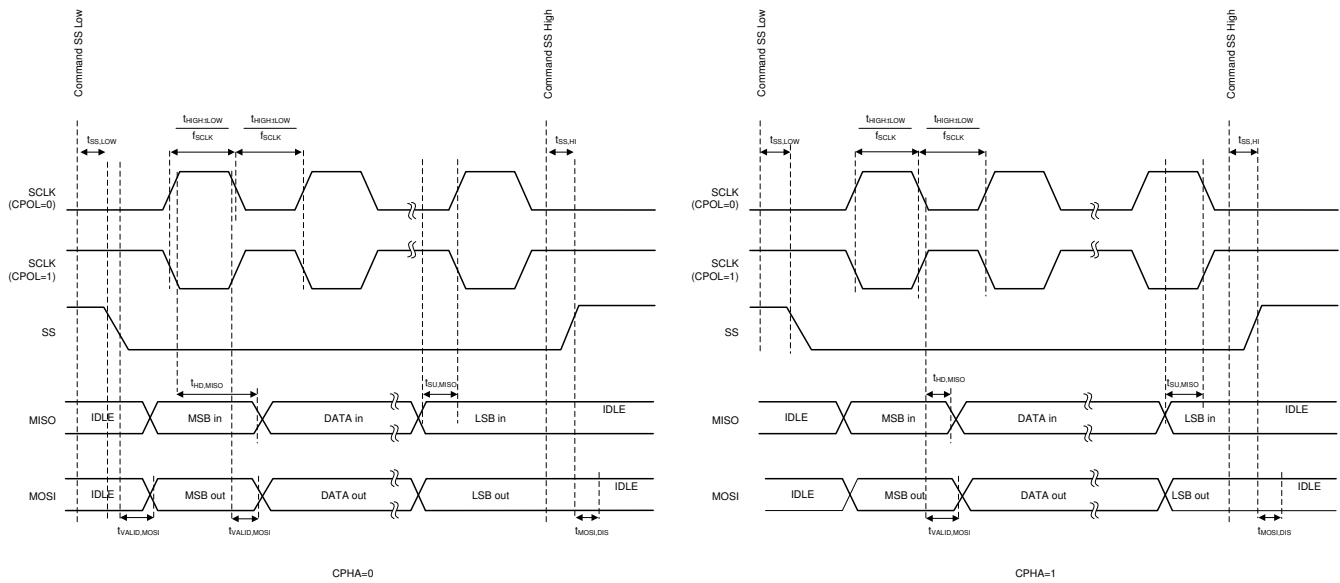


图 6-41. SPI 控制器时序图

**表 6-21. 对外部 SPI 目标进行写入**

步进	说明
1	配置 SPI 时钟极性、时钟相位、事务位数： a. 对 <i>SPI_CONF</i> 寄存器进行写入来配置 SPI 通信
2	写入数据（1 至 24 位，在 <i>SPI_CONF[NUMBIT4:0]</i> 设置中指定）： a. 将要发送到 SPI 目标的数据设置到 <i>SPI_TX1</i> 至 <i>SPI_TX3</i> 寄存器中 b. <i>SPI_TX1</i> 是 LSByte， <i>SPI_TX3</i> 是 MSByte
3	选择目标（假设低电平有效）并执行 SPI 写入操作： a. 发送 <i>SPI_EXE</i> 寄存器 = 0x01（即 [ <i>SS_CTRL</i> ] = 0 且 [ <i>SPI_GO</i> ] = 1）
4	等待 SPI 通信完成
5	取消选择 SS 端口（假设低电平有效，因此取消选择意味着将 SS 引脚拉为高电平） a. 发送 <i>SPI_EXE</i> 寄存器 = 0x02（即 [ <i>SS_CTRL</i> ] = 1 且 [ <i>SPI_GO</i> ] = 0）

**表 6-22. 从外部 SPI 目标进行读取**

步进	说明
1	配置 SPI 时钟极性、时钟相位、事务位数： a. 对 <i>SPI_CONF</i> 寄存器进行写入来配置 SPI 通信
2	选择目标并执行 SPI 通信： a. 发送 <i>SPI_EXE</i> 寄存器 = 0x01（即 [ <i>SS_CTRL</i> ] = 0 且 [ <i>SPI_GO</i> ] = 1）
3	等待数据事务完成
4	读取数据（1 至 24 位，在 <i>SPI_CONF[NUMBIT4:0]</i> 设置中指定）： a. 从 <i>SPI_RX1</i> 至 <i>SPI_RX3</i> 寄存器中读取 SPI 目标数据 b. <i>SPI_TX1</i> 是 LSByte， <i>SPI_TX3</i> 是 MSByte
5	取消选择 SS 端口（假设低电平有效，因此取消选择意味着将 SS 引脚拉为高电平） a. 发送 <i>SPI_EXE</i> 寄存器 = 0x02（即 [ <i>SS_CTRL</i> ] = 1 且 [ <i>SPI_GO</i> ] = 0）

### 6.3.6.1.8 SPI 环回

SPI 控制器具有可使用 *DIAG\_COMM\_CTRL[SPI\_LOOPBACK]* 位启用的环回功能。启用后，*SPI\_TX\** 寄存器中的字节直接向 SPI 控制器的 MISO 引脚提供时钟输入，以验证 SPI 控制器功能。这是在内部执行的，因此无需外部连接即可运行该测试。这可以验证 SPI 功能是否正常工作。会作为正常 SPI 事务对 *SPI\_CFG*、*SPI\_TX\** 和 *SPI\_EXE* 寄存器进行写入，但外部引脚在该模式下不会切换。也就是说，外部引脚在其最后状态下保持静态，在环回操作期间不改变状态。

测试的预期结果是 *SPI\_TX\** 寄存器中的字节被读入 *SPI\_RX\** 寄存器。SS 引脚被锁存至启用 LOOPBACK 模式时存在的 *SPI\_EXE[SS\_CTRL]* 中的设置。进入 LOOPBACK 模式之前必须设置 CPHA 和 CPOL 参数，以确保正常运行。在 LOOPBACK 模式下更改 CPOL 或 CPHA 参数可能会导致 SPI 输出上出现错误的脉冲，因此不建议这么做。

### 6.3.6.2 故障处理

#### 6.3.6.2.1 故障状态层次结构

该器件会监测多种类型的故障，例如：

- 通过硬件保护器进行电池电芯监测，例如电芯 OV/UV、电芯 OT/UT 等
- 系统操作驱动的故障，例如器件复位、通信超时、热警告等
- 与基于命令的诊断检查相关的故障，例如通过主 ADC 和 AUX ADC 进行的各种比较、BIST 运行等
- 在后台运行的自动诊断检查，例如内部电源、OTP CRC 等
- 通信故障。

*FAULT\_SUMMARY* 寄存器中的每个位表示一组故障，这些故障存储在一个或多个较低级别的故障寄存器中。*FAULT\_SUMMARY* 寄存器表示器件检测到的故障状态最高层次结构级别。主机系统可以定期轮询 *FAULT\_SUMMARY* 寄存器以检查故障状态，仅在需要时读取较低级别的故障寄存器（例如，如果

$FAULT\_SUMMARY[FAULT\_OVUV] = 1$ ，则主机可以读取  $FAULT\_OV1/2$  和  $FAULT\_UV1/2$  寄存器以确定哪个电芯通道触发了故障)。

表 6-23 展示了哪个较低级别的寄存器与  $FAULT\_SUMMARY$  寄存器位相对应。节 6.5 介绍了该寄存器。

表 6-23. 低级故障寄存器

FAULT_SUMMARY 位名称	FAULT_PROT	FAULT_COMP_ADC	FAULT_OTP	FAULT_COMM	FAULT_OTUT	FAULT_OVUV	FAULT_SYS	FAULT_PWR
较低级别的寄存器名称	FAULT_PROT <sub>1</sub>	FAULT_COMP_GPIO	FAULT_OTP <sup>(1)</sup>	FAULT_COMM1 <sup>(1)</sup>	FAULT_OT	FAULT_OV1	FAULT_SYS	FAULT_PWR1
	FAULT_PROT <sub>2</sub>	FAULT_COMP_VCCB1		FAULT_COMM2 <sup>(1)</sup>	FAULT_UT	FAULT_OV2		FAULT_PWR2
		FAULT_COMP_VCCB2		FAULT_COMM3		FAULT_UV1		FAULT_PWR3
		FAULT_COMP_VCOW1				FAULT_UV2		
		FAULT_COMP_VCOW2						
		FAULT_COMP_CBOW1						
		FAULT_COMP_CBOW2						
		FAULT_COMP_CBFET1						
		FAULT_COMP_CBFET2						
		FAULT_COMP_MISC						

(1)  $FAULT\_COMM1/2$  和  $FAULT\_OTP$  寄存器中的某些位的故障信息级别低于  $DEBUG\_COMM^*$  和  $DEBUG\_OTP$  寄存器中显示的故障信息级别。

#### 6.3.6.2.1.1 调试寄存器

$DEBUG\_COMM^*$  和  $DEBUG\_OTP$  寄存器是故障状态的一种形式，显示  $FAULT\_COMM1$ 、 $FAULT\_COMM2$  和  $FAULT\_OTP$  中某些位故障信息的较低层次结构级别。

表 6-24 展示了层次关系。有关寄存器说明的详细信息，请参阅节 6.5。

表 6-24. 调试寄存器

低级故障寄存器	低级寄存器位		关联的调试寄存器
$FAULT\_COMM1$	[UART_RC]	与来自 UART 的接收命令帧相关的故障	$DEBUG\_UART\_RC$
	[UART_RR] [UART_TR]	与来自 UART 的接收或发送响应帧相关的故障	$DEBUG\_UART\_RR\_TR$
	[COMH_BIT]	与 COMH 一个字节中的错误相关的故障	$DEBUG\_COMH\_BIT$
$FAULT\_COMM2$	[COMH_RC]	与来自 COMH 的接收命令帧相关的故障	$DEBUG\_COMH\_RC$
	[COMH_RR] [COMH_TR]	与来自 COMH 的接收或发送响应帧相关的故障	$DEBUG\_COMH\_RR\_TR$
	[COML_BIT]	与 COML 一个字节中的错误相关的故障	$DEBUG\_COML\_BIT$
	[COML_RC]	与来自 COML 的接收命令帧相关的故障	$DEBUG\_COML\_RC$
	[COML_RR] [COML_TR]	与来自 COML 的接收或发送响应帧相关的故障	$DEBUG\_COML\_RR\_TR$
	$FAULT\_OTP$	[SEC_DET]	OTP 中的单错校正
[DED_DET]		OTP 中的双错校正	$DEBUG\_OTP\_DED\_BLK$

#### 6.3.6.2.2 故障屏蔽和复位

##### 6.3.6.2.2.1 故障屏蔽

当器件检测到故障时，会设置相应的低级寄存器位（包括  $DEBUG\_*$  寄存器中的相关位）。根据故障层次关系，故障会反映到  $FAULT\_SUMMARY$  寄存器中。

可以屏蔽一组故障，仍会设置相关的低级寄存器标志，但故障不会反映到对应的  $FAULT\_SUMMARY$  寄存器中。可以通过  $FAULT\_MSK1$  和  $FAULT\_MSK2$  寄存器屏蔽故障。

例如，要屏蔽将设置的  $FAULT\_SUMMARY[FAULT\_OTUT]$ ，主机会设置  $FAULT\_MSK1[MSK\_OT] = 1$  和  $[MSK\_UT] = 1$ 。

当故障被屏蔽时，它还会阻止器件在发生屏蔽故障时使  $NFAULT$  引脚生效。有关  $NFAULT$  信号的详细信息，请参阅节 6.3.6.2.3。

表 6-25. 故障屏蔽

	屏蔽位名称	受影响的相关低级寄存器	将被屏蔽的 $FAULT\_SUMMARY$ 寄存器位
$FAULT\_MSK1$	$[MSK\_PROT]$	$FAULT\_PROT^*$	$[FAULT\_PROT]$
	$[MSK\_UT]$	$FAULT\_UT$	$[FAULT\_OTUT]$
	$[MSK\_OT]$	$FAULT\_OT$	
	$[MSK\_UV]$	$FAULT\_UV^*$	$[FAULT\_OVUV]$
	$[MSK\_OV]$	$FAULT\_OV^*$	
	$[MSK\_COMP]$	$FAULT\_COMP\_*$	$[FAULT\_COMP]$
	$[MSK\_SYS]$	$FAULT\_SYS$	$[FAULT\_SYS]$
	$[MSK\_PWR]$	$FAULT\_PWR^*$	$[FAULT\_PWR]$
$FAULT\_MSK2$	$[MSK\_OTP\_CRC]$	$FAULT\_OTP[CUST\_CRC][FACT\_CRC]$	$[FAULT\_OTP]$
	$[MSK\_OTP\_DATA]$	$FAULT\_OTP$ 、 $DEBUG\_OTP\_*$ 中的所有非 CRC 位	
	$[MSK\_COMM3\_FCOMM]$	$FAULT\_COMM3[FCOMM\_DET]$	$[FAULT\_COMM3]$
	$[MSK\_COMM3\_FTONE]$	$FAULT\_COMM3[FTONE\_DET]$	
	$[MSK\_COMM3\_HB]$	$FAULT\_COMM3[HB\_FAIL][HB\_FAST]$	
	$[MSK\_COMM2]$	$FAULT\_COMM2$ 、 $DEBUG\_COMH\_*$ 、 $DEBUG\_COML\_*$	$[FAULT\_COMM2]$
	$[MSK\_COMM1]$	$FAULT\_COMM1$ 、 $DEBUG\_UART\_*$	$[FAULT\_COMM1]$

#### 6.3.6.2.2.2 故障复位

一旦检测到故障，故障状态位就会被锁存，直到使用复位位将其清除。与故障屏蔽类似，当特定故障复位位被设置时，相关的低级故障寄存器（包括  $DEBUG\_*$  寄存器）会被清除。如果所有关联的低级寄存器都被清除， $FAULT\_SUMMARY$  寄存器中的相应位将被清除。如果故障条件持续存在且写入复位位，则故障状态位不会复位。在基础故障条件消除后，故障指示器才能复位。

故障通过  $FAULT\_RST1$  和  $FAULT\_RST2$  寄存器进行复位；故障复位位与故障屏蔽位采用相同的对应故障状态寄存器结构。

#### 6.3.6.2.3 故障信令

主机可以通过以下方法获取故障状态：

- 不断轮询菊花链中每个器件的  $FAULT\_SUMMARY$  状态。如果  $FAULT\_SUMMARY$  不为零，则读取低级故障状态寄存器以获得更多信息。
- 启用故障状态以将菊花链传递到基底器件。当菊花链中任何器件中的  $FAULT\_SUMMARY$  不为零时，使基底器件的  $NFAULT$  引脚生效。主机监测  $NFAULT$ 。当  $NFAULT$  被触发时，主机会在  $FAULT\_SUMMARY$  上进行广播读取以确定出现故障的器件。

当使用基底器件中的  $NFAULT$  引脚在故障检测下向主机发送信号时，堆栈器件必须将其故障状态信息传输到基底器件。信息通过  $COMH/L$  使用相同的通信电缆进行传输。在  $ACTIVE$  模式下，当转发响应帧时，每个器件都会将故障状态嵌入到通信中。在  $SLEEP$  模式下，或在  $SLEEP$  模式下使用检测信号和故障音调。

可以通过配置  $DEV\_CONF[NFAULT\_EN] = 0$  来屏蔽  $NFAULT$  引脚。当  $NFAULT$  被禁用时，器件将在  $FAULT\_SUMMARY$  寄存器中设置相应的标志，但不会使  $NFAULT$  生效。

### 6.3.6.2.3.1 ACTIVE 模式下的故障状态传输

在 ACTIVE 模式下，如果  $DEV\_CONF[FCOMM\_EN] = 1$ ，则堆栈器件可以在重新传输响应帧之前嵌入其故障状态。当  $[FCOMM\_EN] = 1$  时，堆栈器件将响应帧的器件地址字节、寄存器地址字节（高地址字节和低地址字节）中的 SOF 位重用为故障状态位。请参阅图 6-42。这将在本节的其余部分称为故障状态位。

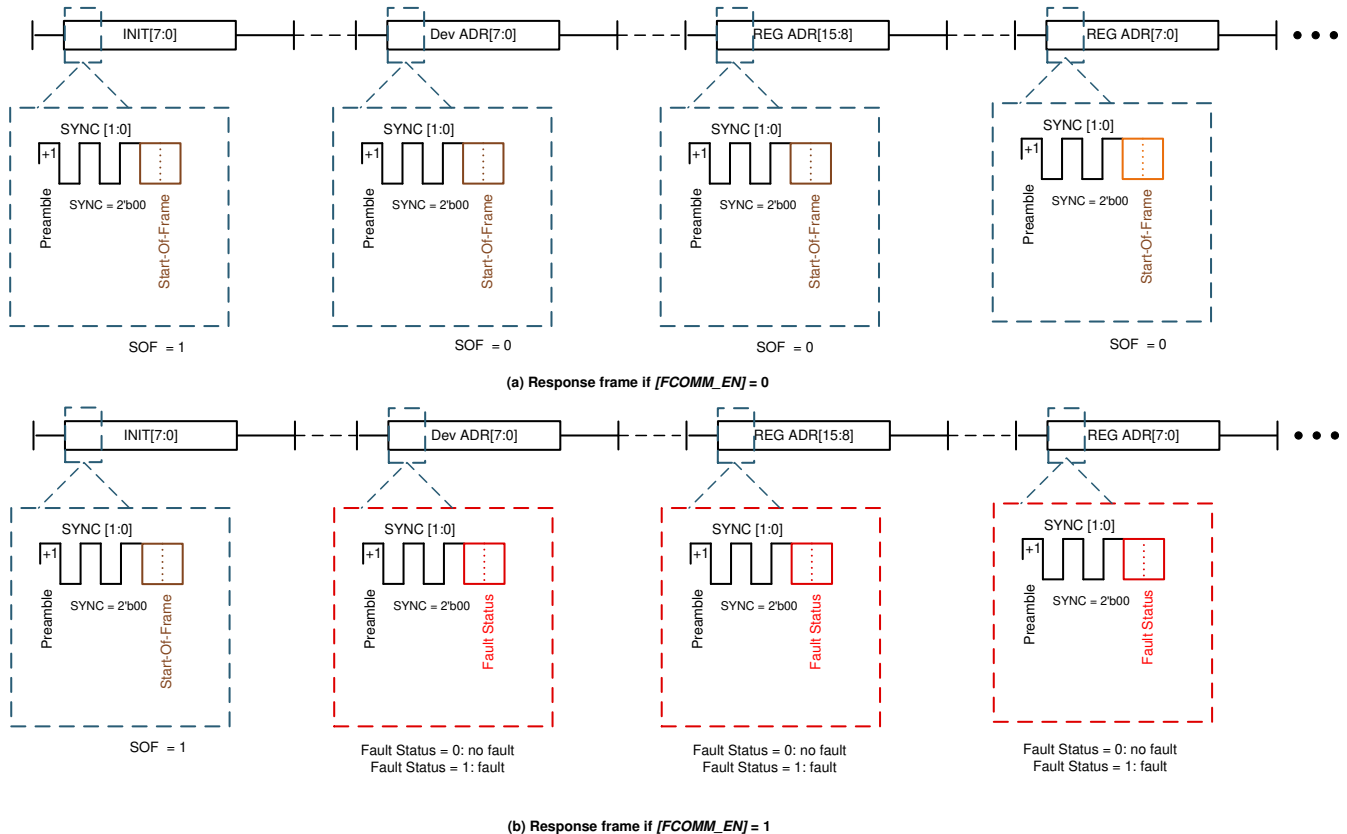


图 6-42. 在通信响应帧中嵌入故障状态

为了传递堆栈器件的故障状态，主机向 ToS 器件发送广播读取或发送单个器件读取。这两种类型的读取都会导致响应帧通过菊花链中的每个器件，从而使每个器件都有机会将其故障状态与响应帧中的故障状态位进行或运算。

图 6-43 展示了从单器件读取命令通过菊花链到达顶部器件的响应帧示例。

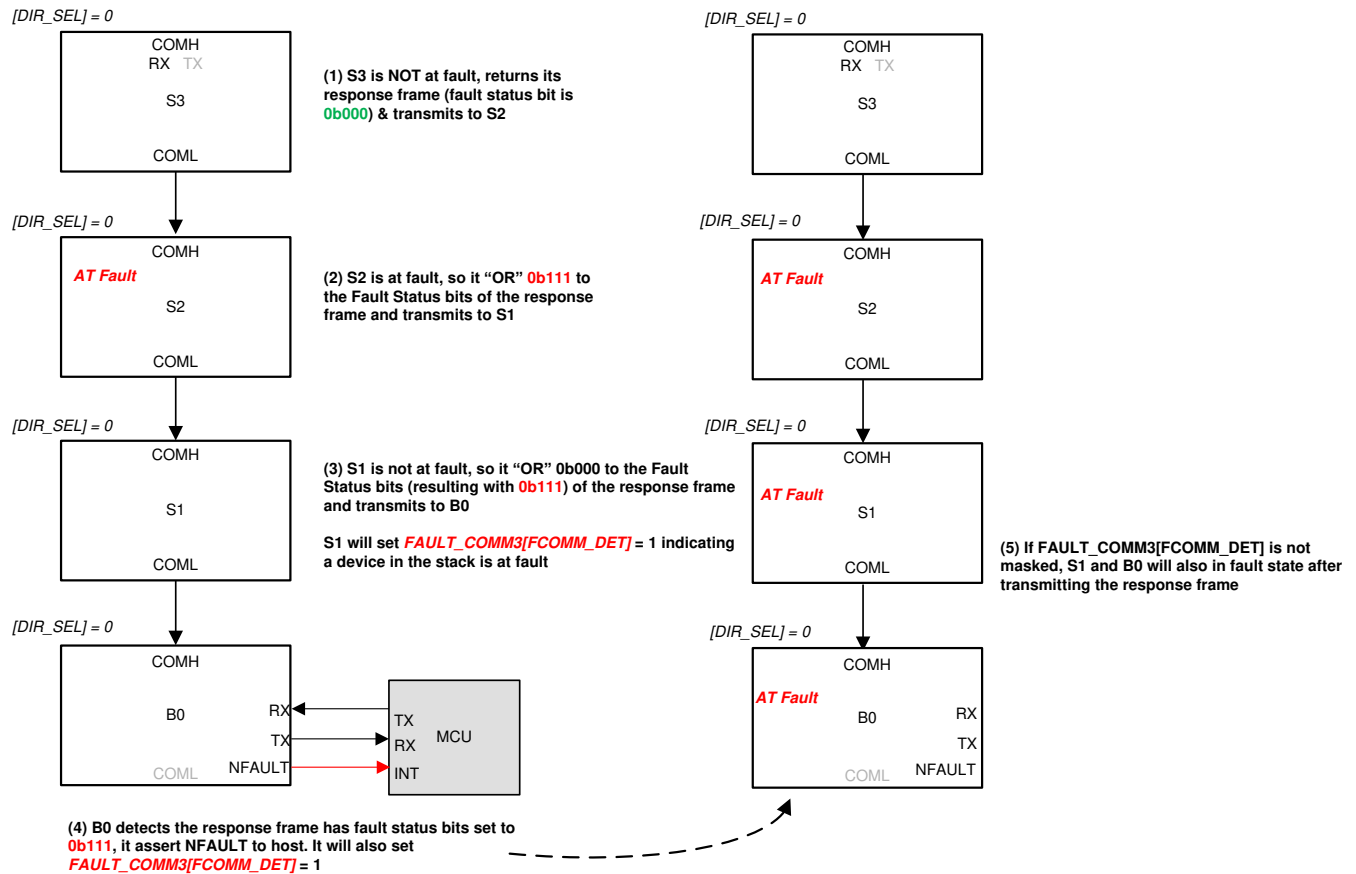


图 6-43. ACTIVE 模式下的传输故障状态 (响应单器件读取)

当器件没有故障时，它会将故障状态位与 0b000 进行或运算；否则，它会将故障状态位与 0b111 进行或运算。因此，如果菊花链中的任何器件存在故障，则故障状态位将为 0b111。为了基底器件能够使 `NFAULT` 引脚生效，需要至少两个故障状态位为 1。

此外，当器件检测到至少有两个故障状态位为 1 的响应帧时，器件也会设置 `FAULT_COMM3[FCOMM_DET] = 1`。如果该故障未被屏蔽，器件也将处于故障状态。下次传输响应帧时，器件会将故障状态位与 0b111 进行或运算。

主机执行广播读取来检测菊花链中的哪个器件出现故障以及故障为何种类型。

### 6.3.6.2.3.2 SLEEP 模式下的故障状态传输

在 SLEEP 模式下，以下故障检测仍有效：

- 客户和出厂 OTP 影子寄存器 CRC 校验
- 器件热警告
- 电源 OV、UV 和振荡检测
- 电芯 OV 和 UV 检测 (如果启用了 OVUV 保护器)。
- 热敏电阻 OT 和 UT 检测 (如果启用了 OTUT 保护器)。

由于在 SLEEP 模式下无法进行通信，因此该器件提供了一个可通过检测信号 (器件处于正常状态) 和故障 (器件处于故障状态) 音调传输故障状态的选项。这些音调与通信命令帧相同的方向 (基于 `CONTROL1[DIR_SEL]` 设置) 传输。为了使音调信号返回至基底器件 (从而可以在需要时触发 `NFAULT`)，必须使用一个环形架构来支持在 SLEEP 模式下传输故障状态。

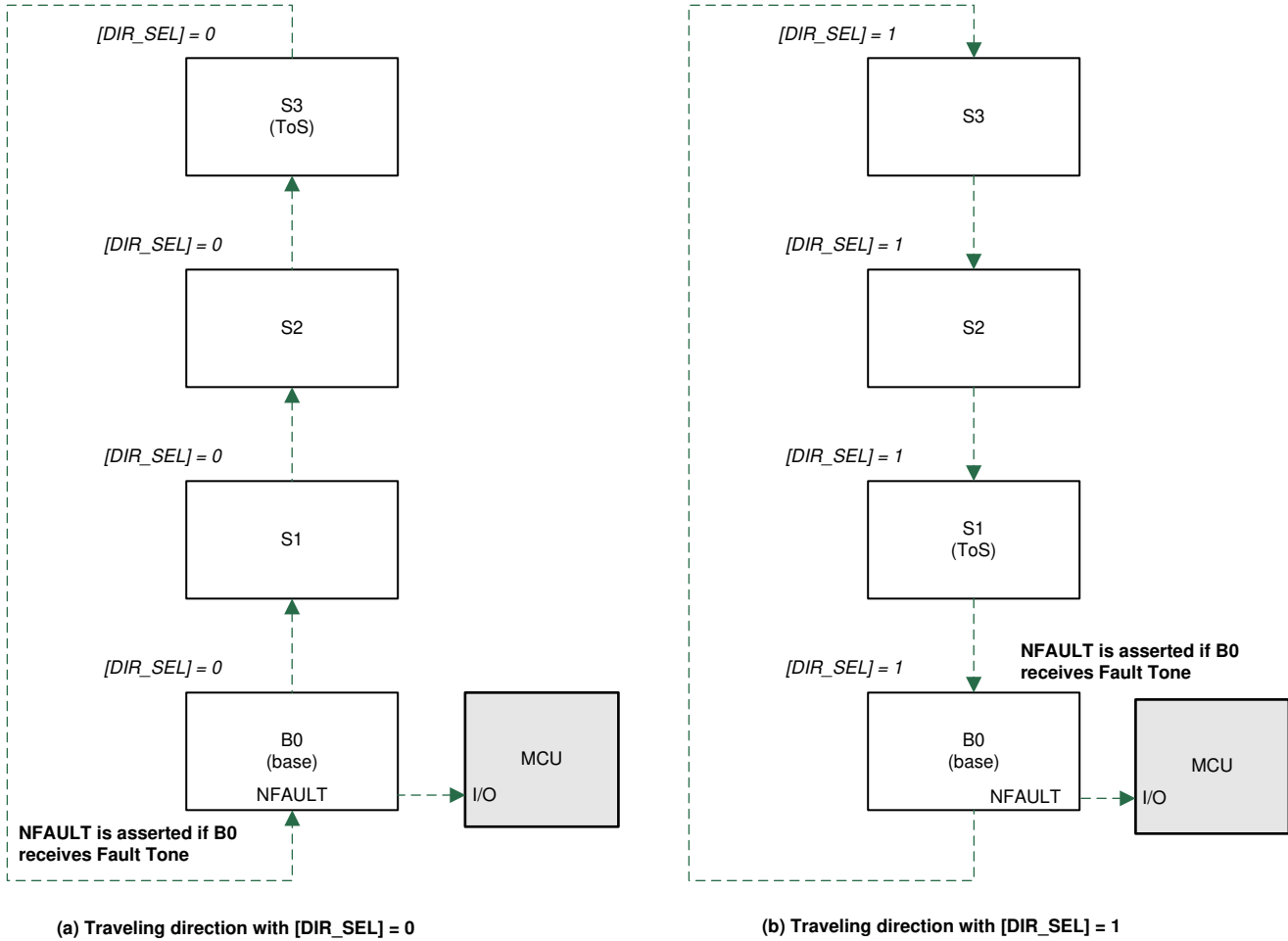


图 6-44. 检测信号或故障音调行进方向

检测信号和故障音调都是与通信音调相似的音调类型。一个主要区别是通信音调仅与单个对突发一起传输，而检测信号和故障音调则定期与对突发一起传输。有关详细信息，请参阅节 6.3.6.2.3.3。

#### 6.3.6.2.3.3 检测信号和故障音调

可以通过设置  $DEV\_CONF[HB\_EN] = 1$  和  $DEV\_CONF[FTONE\_EN] = 1$  来启用检测信号和故障音调发送器，从而启用这些音调。无论  $[HB\_EN]$  和  $[FTONE\_EN]$  设置如何，检测信号和故障音调接收器在 **SLEEP** 模式下始终开启。为了避免因检测信号或故障音调故障而检测到故障（使 **NFAULT** 或 **FAULT\_SUMMARY** 寄存器生效），应通过  $[MSK\_COMM3\_HB] = 1$  或  $[MSK\_COMM3\_FTONE] = 1$  屏蔽故障。

通过具有“-”极性的对来形成检测信号和故障音调。两者之间的区别在于对的数量。与通信音调不同，检测信号和故障音调都是定期传输的。音调之间的周期是突发周期。传输的对数始终大于检测所需的对数。

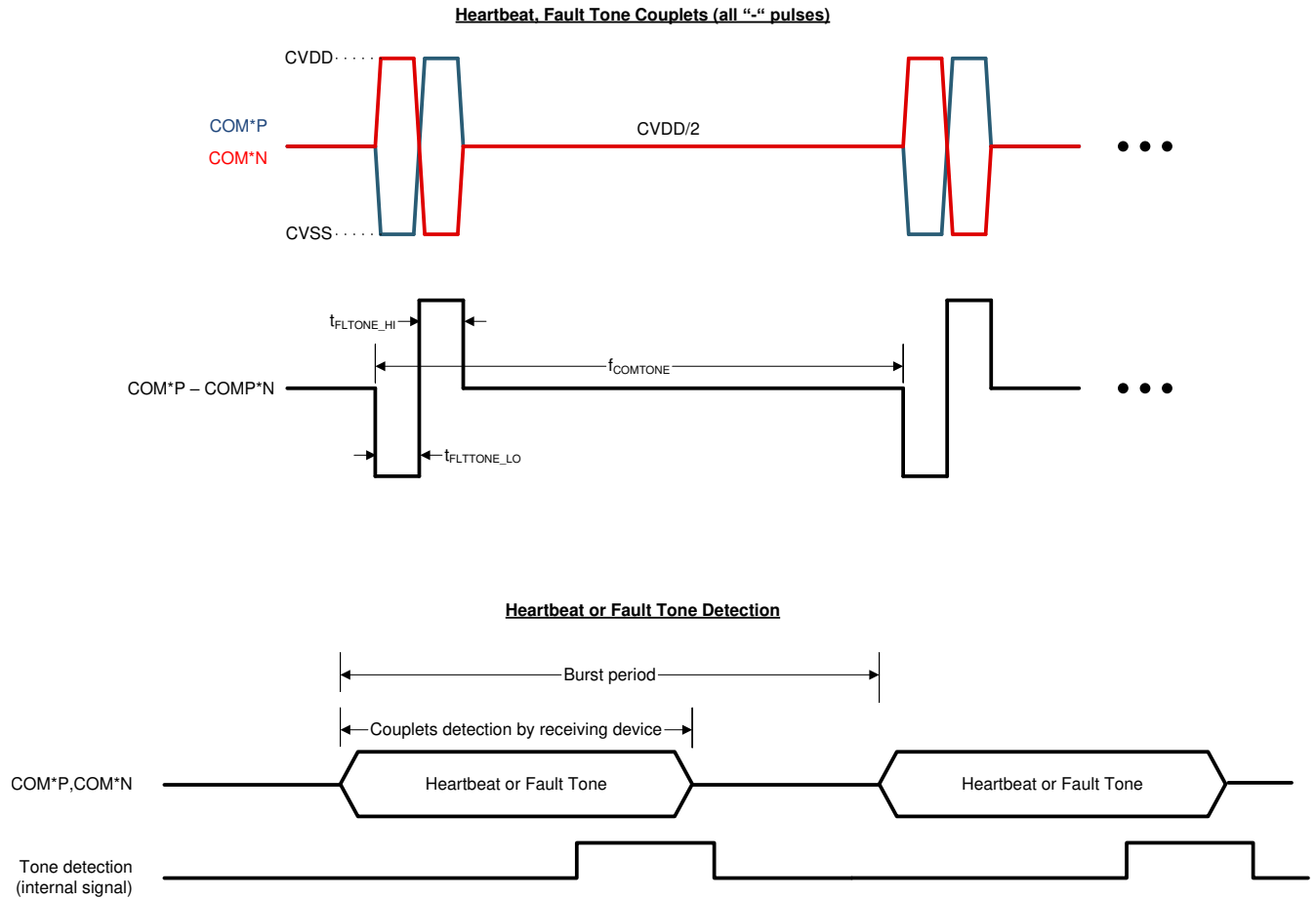


图 6-45. 检测信号和故障音调

### 6.3.6.3 非易失性内存

非易失性存储器 (NVM) 中的某些存储器位置可使用 OTP (一次性可编程) 进行编程。存储器空间分为两组：出厂空间和客户空间。出厂空间存储正常运行所必需的器件配置。主机无法访问该空间。客户空间包含主机系统可以针对其应用配置自定义的器件默认设置。该空间可由主机读取和编程。

发生器件复位时，出厂和客户 OTP 值会重新加载到其影子寄存器中。在出厂和客户空间 OTP 加载期间会执行错误检查和校正 (ECC)、单错校正 (SEC) 和双错检测 (DED)。如果检测到错误，则会设置相应的 `FAULT_OTP[SEC_DET]` 或 `FAULT_OTP[DED_DET]`。

出厂 OTP 空间的任何加载错误都会使用 `FAULT_OTP[FACTLDERR]` 指示故障。客户 OTP 空间的任何加载错误都会使用 `FAULT_OTP[CUSTLDERR]` 指示故障。此外，使用 CRC 来保护 OTP 空间 (出厂和客户) 免受数据完整性问题的影响。如果检测到 CRC 错误，则会设置相应的 `FAULT_OTP[FACT_CRC]` 和 `[CUST_CRC]` 位。

如果编程期间 OTP 页面空间 (出厂和客户) 中存在任何过压错误情况，则会设置 `OTP_FAULT[GBLOVERR]` 位。从存在此错误的器件收到的信息不得视为可靠的信息。

#### 6.3.6.3.1 OTP 页面状态

有两个未使用的 OTP 存储器页面可供客户编程。每个页面状态都保存在 `OTP_CUST1_STAT` 和 `OTP_CUST2_STAT` 寄存器中。这些寄存器提供有关页面当前状态的信息，例如：

- 加载状态 (已加载、加载出错或加载失败)
- 编程成功或可供编程
- 编程状态

发生复位时，器件会评估 OTP 页面状态并选择要加载的最新且有效的 OTP 页面。第 2 页的优先级比第 1 页的优先级高。如果两个页面均未写入，则会加载出厂 OTP 默认值。节 6.5.1 显示了所有客户可编程 OTP 参数。当客户 OTP 第 1 页和第 2 页未编程时，寄存器摘要还会显示默认值。

- 有效页面是指  $OTP\_CUST\_STAT[PROGOK] = 1$  的页面。
- 当选择该页面进行加载时， $OTP\_CUST\_STAT[LOADED] = 1$ 。
- 如果在页面加载过程中出现单个错误，则在更正该单个错误后加载页面，并且  $OTP\_CUST\_STAT[LOADWRN] = 1$ 。
  - 此外，会使用已纠错的块的位置来更新  $DEBUG\_OTP\_SEC\_BLK$  寄存器。
- 如果发生双错，则会终止加载该块，并加载该块的硬件默认值（如节 6.5.1 中所示）。
  - DED 不会终止整个页面加载过程，仅终止受影响的块。
  - 当发生 DED 时， $OTP\_CUST\_STAT[LOADERR] = 1$ 。此外，会使用发生双重错误的块来更新  $DEBUG\_OTP\_DED\_BLK$  寄存器。

### 6.3.6.3.2 OTP 编程

节 6.5.1 展示了所有可以编程到客户 OTP 页面中的参数。有两个 OTP 存储器页面可供客户使用。

在对 OTP 进行编程之前，主机确保：

- 所有 OTP 影子寄存器都具有正确的设置
- 客户 OTP 页面是有效的，可以进行编程。有效页面是  $OTP\_CUST\_STAT[TRY] = 0$  且  $OTP\_CUST\_STAT[FMERR] = 0$  的页面。

表 6-26. 对 OTP 进行编程

步进	过程
1	解锁 OTP 编程： <ol style="list-style-type: none"> <li>将以下数据写入 <math>OTP\_PROG\_UNLOCK1A</math> 至 <math>OTP\_PROG\_UNLOCK1D</math> 寄存器。               <ul style="list-style-type: none"> <li>• <math>OTP\_PROG\_UNLOCK1A \leq</math> 数据 0x02</li> <li>• <math>OTP\_PROG\_UNLOCK1B \leq</math> 数据 0xB7</li> <li>• <math>OTP\_PROG\_UNLOCK1C \leq</math> 数据 0x78</li> <li>• <math>OTP\_PROG\_UNLOCK1D \leq</math> 数据 0xBC</li> </ul> </li> <li>使用以下数据再次对 <math>OTP\_PROG\_UNLOCK2A</math> 至 <math>OTP\_PROG\_UNLOCK2D</math> 寄存器进行写入。               <ul style="list-style-type: none"> <li>• <math>OTP\_PROG\_UNLOCK2A \leq</math> 数据 0x7E</li> <li>• <math>OTP\_PROG\_UNLOCK2B \leq</math> 数据 0x12</li> <li>• <math>OTP\_PROG\_UNLOCK2C \leq</math> 数据 0x08</li> <li>• <math>OTP\_PROG\_UNLOCK2D \leq</math> 数据 0x6F</li> </ul> </li> </ol> 必须按顺序（即 A、B、C，然后是 D）对每个寄存器块进行写入，其间不得进行其他写入或读取。最佳做法是使用相同的写入命令进行更新。如果在不按顺序的情况下进行任何寄存器更新尝试，或者在写入之间写入或读取另一个寄存器，则必须重做整个序列。
2	检查以确认 OTP 解锁程序是否成功： <ol style="list-style-type: none"> <li>读取以确认 <math>OTP\_PROG\_STAT[UNLOCK] = 1</math> 在步骤 1 之后发出读取命令是正常的，但发出 <math>[PROG\_GO]</math> 必须是解锁程序之后的下一条写入命令。</li> </ol>
3	选择合适的 OTP 页面并开始 OTP 编程： <ol style="list-style-type: none"> <li>要对第 1 页进行编程，请设置 <math>OTP\_PROG\_CTRL[PAGESEL][PROG\_GO] = 0x01</math>，或者</li> <li>要对第 2 页进行编程，请设置 <math>OTP\_PROG\_CTRL[PAGESEL][PROG\_GO] = 0x03</math></li> </ol>
4	等待 $t_{PROG}$ 以完成 OTP 编程
5	检查以确保 OTP 编程期间没有错误。成功完成 OTP 编程后，以下位预计为 1： <ol style="list-style-type: none"> <li><math>OTP\_PROG\_STAT[DONE] = 1</math>，OTP 编程完成。不会设置该寄存器中的其他位。</li> <li>如果对第 1 页进行了编程，则 <math>OTP\_CUST1\_STAT[PROGOK]</math>、<math>[TRY]</math>、<math>[OVOK]</math> 和 <math>[UVOK]</math> 位为 1。其他位为 0。</li> <li>如果对第 2 页进行了编程，则 <math>OTP\_CUST2\_STAT[LOADED]</math>、<math>[PROGOK]</math>、<math>[TRY]</math>、<math>[OVOK]</math> 和 <math>[UVOK]</math> 位为 1。其他位为 0。</li> </ol>
6	发出数字复位以在寄存器中重新加载已更新的 OTP 值： <ol style="list-style-type: none"> <li><math>CONTROL1[SOFT\_RESET] = 1</math></li> </ol>

编程期间，如果发生编程电压 OV 或 UV 事件，则 `OTP_CUST*_STAT[UVOK]` 或 `OTP_CUST_STAT2[OVOK]` 位为 0，指示在编程尝试期间检测到编程电压欠压或过压情况。此外，`OTP_PROG_STAT` 寄存器中的 `[UVERR]`、`[OVERR]`、`[SUVERR]` 和 `[SOVERR]` 位指示在编程和稳定性测试期间是否存在编程电压错误。

#### 备注

- 在编程期间，器件在实际对 OTP 进行编程之前执行编程电压稳定性测试。如果编程电压未通过稳定性测试，则器件不会设置 `OTP_CUST*_STAT[TRY]` 位，从而使客户能够再次尝试对该页面进行编程。
- 如果主机错误地选择了用于编程的页面，则会设置 `OTP_PROG_STAT[PROGERR]` 位。这表明所选页面不可进行编程。选择正确的页面并重试编程。
- 在高于 55°C 的温度下器件不会开始 OTP 编程。
- 对于 0.1 μF 的 LDOIN 电容器，OTP 编程时间（从 `[PROG_GO] = 1` 到 `[DONE] = 1`）为 100ms。

### 6.3.6.4 诊断控制/状态

以下各节介绍可用作安全机制的一部分的诊断控制和故障状态。

可从德州仪器 (TI) 单独获取 BQ79616 安全手册和 BQ79606 FMEDA 文档。如需更多信息，请联系 TI 销售助理或应用工程师。

#### 6.3.6.4.1 电源检查

##### 6.3.6.4.1.1 电源诊断检查

内部电源电路具有过压、欠压、振荡检测和/或电流限制检查功能。当器件处于 ACTIVE 或 SLEEP 模式时，所有这些检测都会在后台持续运行。如果检测到故障，会设置 `FAULT_PWR*` 寄存器中的相应标志，或者在某些故障模式下，器件将复位。表 6-27 总结了适用于每个电源的诊断以及检测到故障时的相应操作。

表 6-27. 电源诊断检查

电源/接地引脚	OV 检查	UV 检查	OSC 检查	电流限值	引脚开路
LDOIN					
AVDD	如果该引脚失败，则设置 <code>FAULT_PWR1[AVDD_OV]</code>	如果该引脚失败，则禁用 DVDD 并触发数字复位。软复位后，器件设置 <code>[AVDDUV_DRST]</code> 以指示复位是由 AVDD UV 引起的。	如果失败，则设置 <code>FAULT_PWR1[AVDD_OSC]</code>	将电流限制为 EC 表电流限制规格	
DVDD	如果该引脚失败，则设置 <code>FAULT_PWR1[DVDD_OV]</code>	如果失败，则触发数字复位		将电流限制为 EC 表电流限制规格	
CVDD	如果该引脚失败，则设置 <code>FAULT_PWR1[CVDD_OV]</code>	如果该引脚失败，则设置 <code>FAULT_PWR1[CVDD_UV]</code>		将电流限制为 EC 表电流限制规格	
TSREF	如果失败，则将 <code>FAULT_PWR2[TSREF_OV]</code> 、 <code>FAULT_OT</code> 和 <code>FAULT_UT</code> 寄存器设置为全 1。	如果失败，则将 <code>FAULT_PWR2[TSREF_UV]</code> 、 <code>FAULT_OT</code> 和 <code>FAULT_UT</code> 寄存器设置为全 1。	如果失败，则将 <code>FAULT_PWR2[TSREF_OSC]</code> 、 <code>FAULT_OT</code> 和 <code>FAULT_UT</code> 寄存器设置为全 1。	将电流限制为 EC 表电流限制规格	
NEG5V		如果该引脚失败，则设置 <code>FAULT_PWR2[NEG5V_UV]</code>			

表 6-27. 电源诊断检查 (续)

电源/接地引脚	OV 检查	UV 检查	OSC 检查	电流限值	引脚开路
REFHP/REFHM			如果 REFHP 失败, 则设置 <code>FAULT_PWR2[REFH_OSC]</code>		如果 REFHM 开路, 则设置 <code>FAULT_PWR1[REFHM_OPEN]</code>
DVSS					如果该引脚开路, 则设置 <code>FAULT_PWR1[DVSS_OPEN]</code>
CVSS					如果该引脚开路, 则设置 <code>FAULT_PWR1[CVSS_OPEN]</code>

## 备注

由于所实现的检测逻辑, 当检测到 AVDD OV 或 UV 时, 也会触发 AVDD OSC 故障。同样, 当发生 TSREF OV 或 UV 时, 也可以触发 TSREF OSC 故障。

## 6.3.6.4.1.2 电源 BIST

该器件实现了电源 BIST (内置自检) 功能来测试主电源故障诊断路径, 其中涵盖以下检测:

- `FAULT_PWR1[AVDD_OV]`、`[AVDD_OSC]`、`[DVDD_OV]`、`[CVDD_OV]`、`[CVDD_UV]`、`[REFHM_OPEN]`、`[DVSS_OPEN]` 和 `[CVSS_OPEN]`
- `FAULT_PWR2[TSREF_OV]`、`[TSREF_UV]`、`[TSREF_OSC]`、`[NEG5V_UV]`、`[REFHM_OSC]` 和 `[PWRBIST_FAIL]`

电源 BIST 本质上是对校验器的检查, 它是由主机启动的基于命令的功能。

电源 BIST 启动后会强制在每个电源的故障检测路径上生成一个故障。以 AVDD OV 诊断路径为例, 当 BIST 引擎测试 AVDD OV 路径时, 会发生以下情况:

1. BIST 引擎强制 AVDD OV 比较器失效
2. 然后, BIST 引擎进行检查以确保触发 `FAULT` 寄存器的信号已生效, 并且触发 `NFAULT` 的信号也已生效
3. BIST 引擎重置 `FAULT` 寄存器和 `NFAULT` 信号 (即清除 `FAULT_PWR1/2/3` 寄存器并使 `NFAULT` 无效)
4. BIST 引擎在下一电源诊断路径检查 (例如 AVDD OSC) 时重复执行步骤 1 至步骤 3, 直到测试完 BIST 涵盖的所有预期诊断路径。

## 备注

- 在 BIST 运行期间, `NFAULT` 引脚将开启和关闭。主机忽略 `NFAULT` 引脚状态, 或者可以通过设置 `DEV_CONF[NFAULT_EN] = 0` 来禁用 `NFAULT` 引脚输出。
- 在所有内部电源中, `TSREF` 是可以由主机启用或禁用的电源。为了确保在 BIST 运行期间测试 `TSREF` 诊断路径, 主机在启动电源 BIST 之前启用 `TSREF`。否则, BIST 引擎将在 BIST 运行期间忽略 `TSREF` 诊断路径测试结果。
- 由于其他非电源相关故障也可能触发 `NFAULT`, 因此建议在电源 BIST 运行之前通过 `FAULT_MSK1/2` 寄存器屏蔽所有非电源相关故障。
- 在启动电源 BIST 运行之前, 主机还可确保没有电源故障。

通过发送 `DIAG_PWR_CTRL[PWR_BIST_GO] = 1`, 启动电源 BIST。即使在运行期间检测到故障, BIST 运行也不会中止。BIST 运行结束时, 结果由 `FAULT_PWR2[PWRBIST_FAIL]` 标志指示。

电源 BIST 强制发生故障并确保诊断路径相应地触发故障。BIST 运行失败表示在故障条件下无法触发诊断路径。要进一步检查哪条路径无法指示故障, 主机可以设置 `DIAG_PWR_CTRL[BIST_NO_RST] = 1`。该位在 BIST 运行

期间禁用复位步骤。启用该选项后重新启动电源 BIST。在 BIST 运行结束时，检查 *FAULT\_PWR1* 和 *FAULT\_PWR2* 寄存器。任何保持为 0 的寄存器标志都表示该标志无法标记故障。

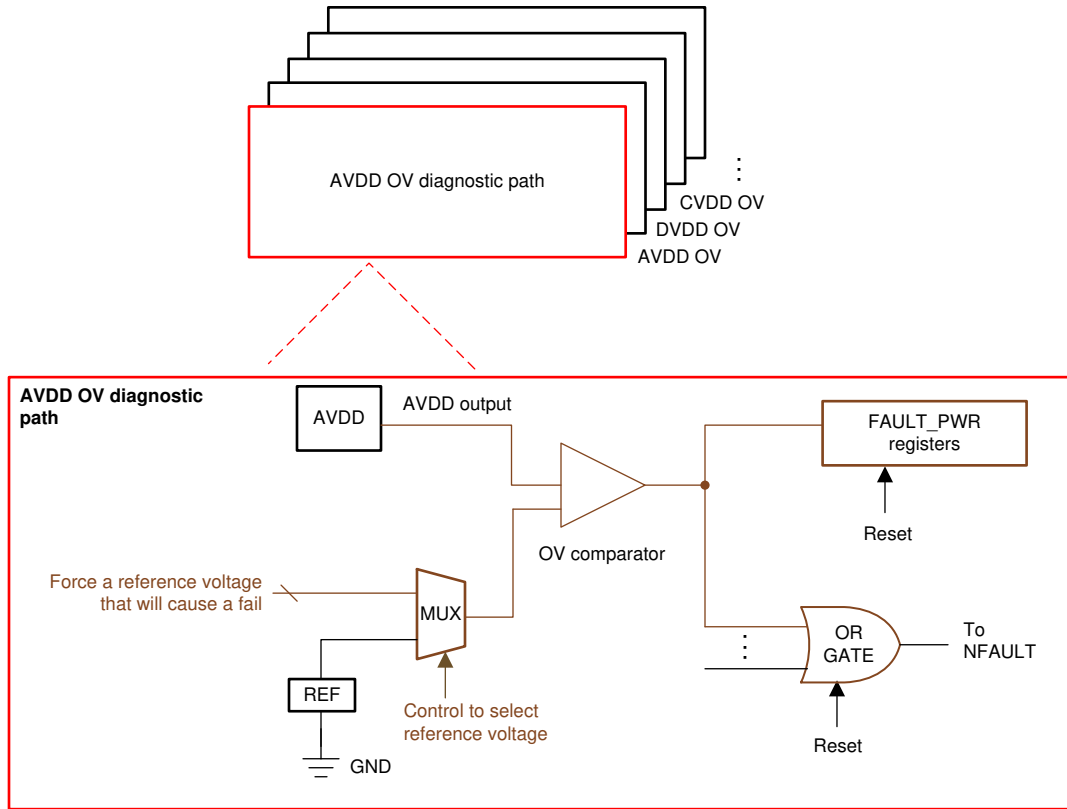


图 6-46. 电源 BIST

#### 6.3.6.4.2 热关断和警告检查

##### 6.3.6.4.2.1 热关断

当热关断传感器检测到器件的过热情况时，会发生热关断。该传感器在无交互的情况下运行，并且与 ADC 测量的内核传感器分离。热关断功能具有一个寄存器状态指示器标志 (*FAULT\_SYS[TSHUT]*)，该标志在关断事件期间保存，并且可以在器件唤醒备份后读取。当发生 TSHUT 故障时，器件立即进入 SHUTDOWN 模式。UART 或菊花链上的任何待处理事务都会被丢弃。当发生热关断事件时，不会执行任何故障信令，因为器件会立即关闭。

为了唤醒器件，主机确保环境温度低于  $T_{SHUT\_FALL}$  并向基底器件发送 WAKE ping。如果环境温度仍高于  $T_{SHUT\_FALL}$ ，主机将不会尝试唤醒器件。

唤醒后，会设置 *FAULT\_SYS[TSHUT]* 位。有关更多详细信息，请参阅节 6.4.1.1。如果系统故障未被屏蔽，则 *FAULT\_MSK1[MSK\_SYS] = 0*，热关断将反映为故障，并将在 *FAULT\_SUMMARY* 寄存器和 NFAULT 引脚生效中指示。

##### 6.3.6.4.2.2 热警告

为了警告主机即将发生热过载，该器件包含一个过热警告，当内核温度接近热关断时，该警告会指示故障。该器件通过 *TWARN* 传感器根据热警告阈值检测内核温度。有四个可以通过 *PWR\_TRANSIT\_CONF[TWARN\_THR1:0]* 设置配置的阈值选项。

当系统故障未被屏蔽且发生温度警告故障时，*FAULT\_SYS[TWARN] = 1*。主机可以执行相应的操作来避免热关断。

### 6.3.6.4.3 振荡器看门狗

振荡器由看门狗电路进行监控。该器件中有两个振荡器，即 HFO 和 LFO。如果这些振荡器不起作用，则器件将无法运行。如果 HFO 或 LFO 未在预期时间内进行转换，则看门狗电路会引发数字复位。

当发生该意外复位时，建议主机向有问题的器件发送 SHUTDOWN ping/音调，然后发送 WAKE ping 以重置菊花链。如果振荡器确实损坏，器件将无法重新启动，必须进行更换。

除看门狗之外，LFO 频率也会受到监控，以确保其保持在可接受的限值内。如果 LFO 频率超出预期范围，则会设置 `FAULT_SYS_FAULT[LFO]` 位。

### 6.3.6.4.4 OTP 错误检查

#### 6.3.6.4.4.1 OTP CRC 测试和故障

CRC 测试：

会对出厂寄存器和客户 OTP 影子寄存器执行 CRC 校验，该校验在后台连续运行。`CUST_CRC_RSLT_HI` 和 `CUST_CRC_RSLT_LO` 寄存器保存当前器件计算出的 CRC 值。该值与 CRC 寄存器 `CUST_CRC_HI` 和 `CUST_CRC_LO` 中的客户编程值进行比较。在更新 CRC 涵盖的任何客户 OTP 影子寄存器时，主机必须将新的 CRC 值更新至 `CUST_CRC_HI` 和 `CUST_CRC_LO` 寄存器。CRC 计算以与节 6.3.6.1.1.2.1.6 中所述相同的方式（包括位流排序）并使用相同的多项式执行。会定期对出厂空间和客户空间进行 CRC 校验和比较，并且在校验完成后设置 `DEV_STAT[CUST_CRC_DONE]` 和 `[FACT_CRC_DONE]` 位。如果该位已设置，则该位会保持设置状态，直到通过读取清除。

CRC 故障：

当 `CUST_CRC_HI/LO` 和 `CUST_CRC_RSLT_HI/LO` 不匹配时，会设置 `FAULT_OTP[CUST_CRC]` 标志，直到该情况得到纠正。对出厂 NVM 空间的连续监控以类似的方式与对客户空间的监控同时进行。当检测到出厂寄存器更改时，会设置 `FAULT_OTP[FACT_CRC]` 标志。发生该故障时，主机应重置故障标志以查看故障是否仍然存在。如果故障仍然存在，则主机必须对该器件执行复位操作。如果复位无法解决问题，则器件已损坏，不得使用。

#### 6.3.6.4.4.2 OTP 裕度读取

该器件提供 OTP 裕度读取测试模式，在该模式下主机可以设置为以裕度 1 或裕度 0 重新加载 OTP。为了启动裕度读取测试，主机通过 `DIAG_OTP_CTRL[MARGIN_MODE2:0]` 选择所需的测试模式并设置 `DIAG_OTP_CTRL[MARGIN_GO] = 1`。该器件将根据 `[MARGIN_MODE2:0]` 设置重新加载 OTP。任何与 OTP 相关的错误都将标记至 `FAULT_OTP` 寄存器。

#### 6.3.6.4.4.3 错误检查和纠正 (ECC) OTP

ECC：

所选寄存器的寄存器值 (0x0000 至 0x002F) 永久存储在 OTP 中。所有寄存器还作为易失性存储位置存在于相同的地址处，称为影子寄存器。易失性寄存器用于读取、写入及器件控制。有关 OTP 中包含的寄存器列表，请参阅节 6.5.1。

在唤醒期间，器件首先使用节 6.5.1 中列出的硬件默认值加载所有影子寄存器。然后，器件有条件地使用来自 OTP 错误检查和纠正 (ECC) 评估结果的 OTP 内容加载寄存器。OTP 以 64 位块的形式加载到影子寄存器中；每个块都存储有自己的错误检查和纠正 (ECC) 值。ECC 检测 OTP 存储数据中的 single-bit (单错校正) 或 double-bit (双错检测) 更改。每个块的 ECC 单独计算。

系统会纠正 single-bit 错误，但只检测 double-bit 错误，不会纠正。ECC 良好的块被加载。具有 single-bit 错误的块被纠正，并且会设置 `FAULT_OTP[SEC_DET]` 位以标记已纠正的错误事件。此外，会使用已纠错的块的位置来更新 `DEBUG_OTP_SEC_BLK` 寄存器。这使得主机能够跟踪可能损坏的存储器。该块在纠正 single-bit 错误后加载到影子寄存器中。由于评估是逐块进行的，因此多个块有可能具有单一可纠正错误，但仍能正确加载。只要每个块仅限于一个错误，就可以存在多位错误并被全部纠正。

具有糟糕 ECC 比较的块 ( 一个块中有两位错误 ) 不会被加载, 并且会设置 *FAULT\_OTP[DED\_DET]* 位以标记失败的位错误事件。此外, 会使用发生双重错误的块来更新 *DEBUG\_OTP\_DED\_BLK* 寄存器。硬件默认值保留在寄存器中。这允许正确加载某些块 ( 无失败或 **single-bit** 校正位 ), 而某些块则不加载。当 *FAULT\_OTP[SEC\_DET]* 或 *FAULT\_OTP[DED\_DET]* 位被设置且器件复位未清除该条件时, 器件已损坏且不得使用。

ECC 引擎使用行业标准 72,64 SEC DEC ECC 实现。OTP 由 (72, 64) 汉明码保护, 提供单错校正、双错检测 (SECDED)。对于 OTP 中存储的每 64 位数据, 还存储额外的 8 位奇偶校验信息。奇偶校验位被指定为 p0、p1、p2、p4、p8、p16、p32 和 p64。位 p0 覆盖整个编码的 72 位 ECC 块。其余七个奇偶校验位根据以下规则分配:

- 奇偶校验位 p1 覆盖奇数位位置, 即位位置的最低有效位等于 1 的位位置 ( 1、3、5 等 ), 包括 p1 位本身 ( 位 1 )。
- 奇偶校验位 p2 涵盖位位置的第二个最低有效位等于 1 的位位置 ( 2、3、6、7、10、11 等 ), 包括 p2 位本身 ( 位 2 )。
- p4、p8、p16、p32 和 p64 继续采用这种模式。表 6-28 指定了完整的编码。

表 6-28. (72, 64) 奇偶校验编码

位位置	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	
编码位	d63	d62	d61	d60	d59	d58	d57	p64	d56	d55	d54	d53	d52	d51	d50	d49	d48	d47	
奇偶校验位覆盖情况	p0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	p1	x		x		x		x		x		x		x		x			
	p2	x	x			x	x			x	x			x	x			x	x
	p4	x	x	x	x					x	x	x	x					x	x
	p8									x	x	x	x	x	x	x	x		
	p16									x	x	x	x	x	x	x	x	x	x
	p32									x	x	x	x	x	x	x	x	x	x
	p64	x	x	x	x	x	x	x	x										
位位置	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	
编码位	d46	d45	d44	d43	d42	d41	d40	d39	d38	d37	d36	d35	d34	d33	d32	d31	d30	d29	
奇偶校验位覆盖情况	p0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	p1	x		x		x		x		x		x		x		x			
	p2			x	x			x	x			x	x			x	x		
	p4	x	x					x	x	x	x					x	x	x	x
	p8							x	x	x	x	x	x	x					
	p16	x	x	x	x	x	x												
	p32	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	p64																		
位位置	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
编码位	d28	d27	d26	p32	d25	d24	d23	d22	d21	d20	d19	d18	d17	d16	d15	d14	d13	d12	
奇偶校验位覆盖情况	p0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	p1	x		x		x		x		x		x		x		x			
	p2	x	x			x	x			x	x			x	x			x	x
	p4					x	x	x	x					x	x	x	x		
	p8					x	x	x	x	x	x	x							
	p16					x	x	x	x	x	x	x	x	x	x	x	x	x	x
	p32	x	x	x	x														
	p64																		
位位置	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
编码位	d11	p16	d10	d9	d8	d7	d6	d5	d4	p8	d36	d2	d1	p4	d0	p2	p1	p0	
奇偶校验位覆盖情况	p0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
	p1	x		x		x		x		x		x		x		x			
	p2			x	x			x	x			x	x			x	x		
	p4			x	x	x	x					x	x	x	x				
	p8			x	x	x	x	x	x	x	x								
	p16	x	x																
	p32																		
	p64																		

表 6-29. 编码器和解码器数据输入和输出定位

编码器			
数据输入	编码位	数据输出	位位置
OTP_ECC_DATAIN 1	d0 至 d7	OTP_ECC_DATAOUT 1	0 至 7
OTP_ECC_DATAIN 2	d8 至 d15	OTP_ECC_DATAOUT 2	8 至 15
OTP_ECC_DATAIN 3	d16 至 d23	OTP_ECC_DATAOUT 3	16 至 23
OTP_ECC_DATAIN 4	d24 至 d31	OTP_ECC_DATAOUT 4	24 至 31
OTP_ECC_DATAIN 5	d32 至 d39	OTP_ECC_DATAOUT 5	32 至 39
OTP_ECC_DATAIN 6	d40 至 d47	OTP_ECC_DATAOUT 6	40 至 47
OTP_ECC_DATAIN 7	d48 至 d55	OTP_ECC_DATAOUT 7	48 至 55
OTP_ECC_DATAIN 8	d56 至 d63	OTP_ECC_DATAOUT 8	56 至 63
		OTP_ECC_DATAOUT 9	64 至 71
解码器			
数据输入	位位置	数据输入	编码位
OTP_ECC_DATAIN 1	0 至 7	OTP_ECC_DATAOUT 1	d0 至 d7
OTP_ECC_DATAIN 2	8 至 15	OTP_ECC_DATAOUT 2	d8 至 d15
OTP_ECC_DATAIN 3	16 至 23	OTP_ECC_DATAOUT 3	d16 至 d23
OTP_ECC_DATAIN 4	24 至 31	OTP_ECC_DATAOUT 4	d24 至 d31
OTP_ECC_DATAIN 5	32 至 39	OTP_ECC_DATAOUT 5	d32 至 d39
OTP_ECC_DATAIN 6	40 至 47	OTP_ECC_DATAOUT 6	d40 至 d47
OTP_ECC_DATAIN 7	48 至 55	OTP_ECC_DATAOUT 7	d48 至 d55
OTP_ECC_DATAIN 8	56 至 63	OTP_ECC_DATAOUT 8	d56 至 d63
OTP_ECC_DATAIN 9	64 至 71		

ECC 诊断测试：该器件提供了一个诊断工具来测试 ECC 功能。有两种模式可用于运行诊断。第一种是自动模式 ( $OTP\_ECC\_TEST[MANUAL\_AUTO] = 0$ )，该模式使用内部数据来运行测试。在自动模式下， $OTP\_ECC\_TEST[DED\_SEC]$  位选择要执行的测试类型， $OTP\_ECC\_TEST[ENC\_DEC]$  位决定要测试编码器还是解码器功能。ECC 测试结果在  $OTP\_ECC\_DATAOUT^*$  寄存器中提供，具有  $1\ \mu s$  的延迟。每个测试的测试步骤和预期结果如下所示。

自动解码步骤：

1. 将 ECC 测试设置为自动  $OTP\_ECC\_TEST[MANUAL\_AUTO] = 0$
2. 设置解码器设置  $OTP\_ECC\_TEST[ENC\_DEC] = 0$
3. 使用  $OTP\_ECC\_TEST[DED\_SEC]$  将解码器设置为单编码或双编码设置 (1 表示 DED, 0 表示 SEC)
4. 通过  $FAULT\_RST2[RST\_OTP\_DATA] = 1$  清除所有 SEC/DED 故障
5. 启用 ECC 测试  $OTP\_ECC\_TEST[ENABLE] = 1$
6. 读取  $FAULT\_OTP[SEC\_DET]$  标志 (对于 SEC) 或  $FAULT\_OTP[DED\_DET]$  标志 (对于 DED)
7. 对  $OTP\_ECC\_DATAOUT1$  至  $OTP\_ECC\_DATAOUT8$  进行块读取，以验证解码器测试结果，如表 6-30 所示
8. 禁用 ECC 测试  $OTP\_ECC\_TEST[ENABLE] = 0$

自动编码步骤：

1. 将 ECC 测试设置为自动  $OTP\_ECC\_TEST[MANUAL\_AUTO] = 0$
2. 使用  $OTP\_ECC\_TEST[ENC\_DEC] = 1$  设置编码器设置
3. 使用  $OTP\_ECC\_TEST[ENABLE] = 1$  启用 ECC 测试
4. 对  $OTP\_ECC\_DATAOUT1$  至  $OTP\_ECC\_DATAOUT9$  进行块读取，以验证编码器测试结果，如表 6-30 所示
5. 禁用 ECC 测试  $OTP\_ECC\_TEST[ENABLE] = 0$

表 6-30. 解码器和编码器测试验证

[DED_SEC]	[ENC_DEC]	[SEC_DET]	[DED_DET]	OTP_DATAOUT*
0 (SEC 测试)	0 (解码器测试)	1	0	0x18C3 FF8A 68A9 8069
0 (SEC 测试)	1 (编码器测试)	不适用	不适用	0xCD 3968 C140 2EA5 ED6D
1 (DED 测试)	0 (解码器测试)	0	1	0x0000 0000 0000 0000
1 (DED 测试)	1 (编码器测试)	不适用	不适用	0xCD 3968 C140 2EA5 ED6D

### 6.3.6.4.5 集成硬件保护器检查

#### 6.3.6.4.5.1 奇偶效验检查

启用 OVUV 和 OTUT 保护器后，与 OVUV 和 OTUT 配置相关的寄存器设置会锁存到保护器块中。该器件将在后台定期检查，以确保锁定的配置在整个保护器运行过程中保持不变。

奇偶校验涵盖以下锁存设置。如果检测到 OVUV 保护器中存在奇偶校验故障，则器件会设置  $FAULT\_PROT1[VPARITY\_FAIL] = 1$ 。如果检测到 OTUT 保护器中存在奇偶校验故障，则器件会设置  $FAULT\_PROT1[TPARITY\_FAIL] = 1$ 。

表 6-31. 保护器奇偶校验设置

OVUV 保护器	OTUT 保护器	注释
OV 阈值、UV 阈值	OT 阈值、UT 阈值	确保阈值设置在运行期间保持不变
OVUV_MODE 设置	OTUT_MODE 设置	确保保护器不会切换到其他运行模式
NUM_CELL 设置	GPIO_CONF1 至 GPIO_CONF4 设置	确保有效通道（针对 OVUV 的单元通道或针对 OTUT 的 GPIO 通道）在运行期间保持不变

#### 6.3.6.4.5.2 OVUV 和 OTUT DAC 检查

OV、UV、OT 和 UT DAC 值多路复用到 AUX ADC，作为保护器阈值设置诊断检查的一部分，主机可以从中读取这些值。

要测量保护器的 DAC 值，建议通过  $OVUV\_CTRL[OVUV\_LOCK3:0]$  将 OVUV 或 OTUT 保护器锁定到单个通道以进行 OV 和 UV DAC 测量；并通过  $OTUT\_CTRL[OTUT\_LOCK2:0]$  进行 OT 和 UT DAC 测量，然后重新启动 OVUV 保护器或 OTUT 保护器以单通道运行模式运行。主机确保锁定的电芯通道不处于 OV 或 UV 故障状态，或者锁定的 GPIO 通道不处于 OT 或 UT 故障状态。否则，DAC 测量将不会反映触发阈值。请注意，OV 和 UV DAC 值为 (0.8 x 阈值设置)。

#### 6.3.6.4.5.3 OVUV 保护器 BIST

该器件实现了 OVUV BIST (内置自测试) 功能来测试主 OVUV 保护器路径。主机可以通过设置  $[OVUV\_MODE1:0] = 0b10$  和  $[OVUV\_GO] = 1$  来启动 BIST 运行。BIST 运行涵盖：

- OV 和 UV 比较器阈值：
  - 检查电压值是否高于和低于设定的阈值，以确保比较器正确触发。
  - 如果检测到故障，则会设置相应的  $FAULT\_PROT2[OVCOMP\_FAIL]$  或  $[UVCOMP\_FAIL]$  位。
- 从 OVUV 多路复用器到 UV 故障状态位和 NFAULT 引脚的路径：
  - 对于每个 VC 通道，开关断开，以便 OVUV 多路复用器的输入断开，从而能够对被测通道进行 UV 检测
  - 然后，BIST 引擎检查逻辑以使相应的  $FAULT\_UV$  寄存器位生效，并且正确设置 NFAULT。
  - BIST 引擎重置相应的  $FAULT\_UV$  位并使 NFAULT 无效，然后切换到测试下一个通道并重复该过程，直到测试完所有有效通道。
  - 如果检测到故障，则会设置相应的  $[VPATH\_FAIL]$  位。
- OV 故障位和 NFAULT 路径
  - BIST 引擎强制将 1 写入  $FAULT\_OV^*$  寄存器，一次一位，以确保可以相应地设置每个  $FAULT\_OV^*$  寄存器，并且可以使 NFAULT 生效。

- b. 如果检测到故障，则会设置相应的 `[VPATH_FAIL]` 位。

如果启用了 `NFAULT`，则主机会在 `BIST` 运行期间观察 `NFAULT` 切换。`BIST` 运行结束后，`OVUV` 比较器将关闭。主机通过发送 `[OVUV_GO] = 1` 以及 `[OVUV_MODE1:0] = 0b01` ( 轮询模式 ) 来启动常规 `OVUV` 轮询模式。

#### 备注

- 如果在 `OVUV BIST` 运行期间发送了 `[OVUV_GO] = 1`，则器件会根据 `[OVUV_MODE1:0]` 设置执行新的 `GO` 命令。
- 在启动 `OVUV` 保护器 `BIST` 之前，主机屏蔽所有非 `OVUV` 相关故障，并确保任何电芯通道上没有 `OV` 和 `UV` 故障 ( 建议在 `BIST` 运行期间所有电芯电压与 `OV` 或 `UV` 阈值至少相差 `100mV` )。否则，`BIST` 结果无效。
- `BIST` 启动后，如果在开始步骤 2 之前检测到预先存在的故障，`BIST` 引擎将中止并且 `FAULT_PROT2[BIST_ABORT] = 1`。
- 提供了无复位选项 `DIAG_PROT_CTRL[PROT_BIST_NO_RST] = 1`，用于命令 `BIST` 引擎在测试每个通道后不重置故障状态和 `NFAULT` 引脚。如果 `BIST` 运行失败，主机可以选择该选项并重新运行 `BIST` 以检测哪个电芯通道路径无法反映故障寄存器中的故障情况。

#### 6.3.6.4.5.4 OTUT 保护器 BIST

该器件实现了 `OTUT BIST` 功能来测试主 `OTUT` 保护器路径。主机可以通过设置 `[OTUT_MODE1:0] = 0b10` 和 `[OTUT_GO] = 1` 来启动 `BIST` 运行。`BIST` 运行涵盖：

1. `OT` 和 `UT` 比较器阈值
  - a. 检查电压值是否高于和低于设定的阈值，以确保比较器正确触发。
  - b. 如果检测到故障，则会设置相应的 `FAULT_PROT2[OTCOMP_FAIL]` 或 `[UTCOMP_FAIL]` 位。
2. 从 `GPIO` 多路复用器到 `UT` 故障位的路径和 `NFAULT` 路径
  - a. 对于每个 `GPIO` 通道，`GPIO` 在内部上拉，因此 `OTUT` 多路复用器的输入为高电平，从而会引发对被测通道的 `UT` 检测。
  - b. 然后，`BIST` 周期检查逻辑以使相应的 `FAULT_UT` 寄存器位生效，并且正确设置 `NFAULT`。
  - c. `BIST` 引擎重置相应的 `FAULT_UT` 位并使 `NFAULT` 无效，然后切换到测试下一个通道。
  - d. 如果检测到故障，则会设置相应的 `[TPATH_FAIL]` 位。
3. `OV` 故障位和 `NFAULT` 路径
  - a. `BIST` 引擎强制将 `1` 写入 `FAULT_OT` 寄存器，一次一位，以确保可以相应地设置每个 `FAULT_OT` 寄存器，并且可以使 `NFAULT` 生效。
  - b. 如果检测到故障，则会设置相应的 `[TPATH_FAIL]` 位。

如果启用了 `NFAULT`，则主机会在 `BIST` 运行期间观察 `NFAULT` 切换。`BIST` 运行结束后，`OTUT` 比较器将关闭。主机通过发送 `[OTUT_GO] = 1` 以及 `[OTUT_MODE1:0] = 0b01` ( 轮询模式 ) 来启动常规 `OTUT` 轮询模式。

## 备注

- 如果在 OTUT BIST 运行期间发送了  $[OTUT\_GO] = 1$ ，则器件会根据  $[OVUV\_MODE1:0]$  设置执行新的 GO 命令。
- 在启动 OTUT 保护器 BIST 之前，主机会屏蔽所有非 OTUT 相关故障，并确保 BIST 运行期间任何 GPIO 上不存在 OT 和 UT 故障。否则，BIST 结果无效。
- BIST 启动后，如果在开始步骤 2 之前检测到预先存在的故障，BIST 引擎将中止并且  $FAULT\_PROT2[BIST\_ABORT] = 1$ 。
- 提供了无复位选项  $DIAG\_PROT\_CTRL[PROT\_BIST\_NO\_RST] = 1$ ，用于命令 BIST 引擎在测试每个通道后不重置故障状态和 NFAULT 引脚。如果 BIST 运行失败，主机可以选择该选项并重新运行 BIST 以检测哪个 GPIO 通道路径无法反映故障寄存器中的故障情况。

## 6.3.6.4.6 通过 ADC 比较进行诊断

## 6.3.6.4.6.1 电芯电压测量检查

电芯电压测量路径比较：

通过比较主 ADC 的预滤波测量结果与 AUX ADC 的测量结果来执行电芯电压测量检查。要读取主 ADC 和 AUX ADC 测量的比较值，MCU 必须使用  $[AUX\_CELL\_SEL]$  设置将该诊断检查设置为锁定在单个通道上，然后启动该诊断检查。在该配置中，主 ADC 和 AUX ADC 的比较值将分别报告给  $DIAG\_MAIN\_HI/LO$  寄存器和  $DIAG\_AUX\_HI/LO$  寄存器。

主 ADC 和 AUX ADC 具有相同的前端滤波器。该诊断时间主要用于等待 AUX ADC 路径上的 AAF 稳定。 $[AUX\_SETTLE]$  设置使 MCU 能够在诊断时间和噪声滤波水平之间进行权衡。此外，当 AUX ADC 启动时，默认情况下，AUXCELL 时隙始终与主 ADC Cell1 时隙对齐。 $[AUX\_CELL\_ALIGN]$  设置使 MCU 能够更改与主 ADC Cell8 时隙的对齐方式，从而减少较高通道上主 ADC 和 AUX ADC 之间的采样时间差值。

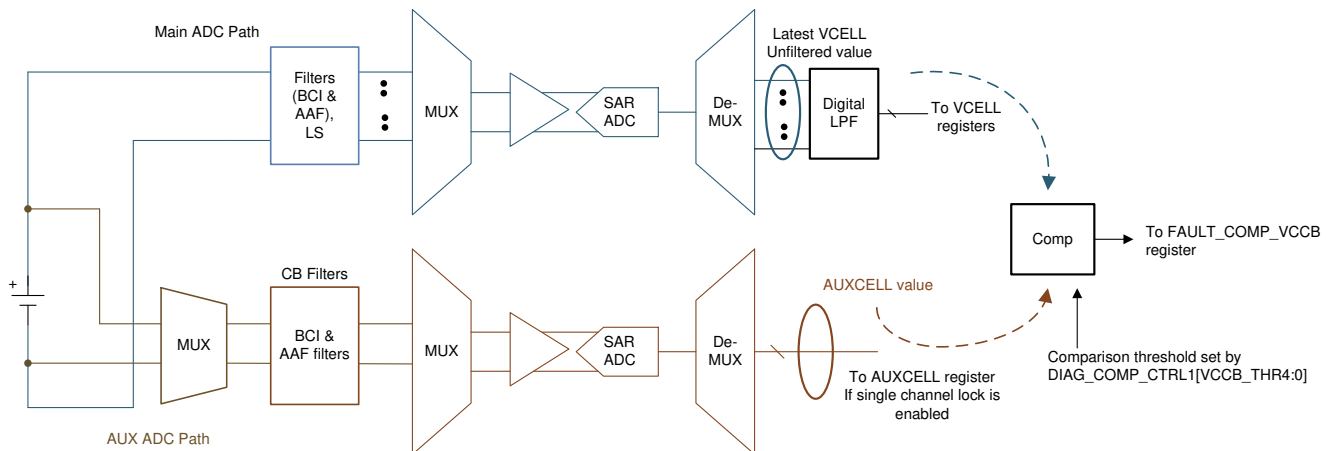


图 6-47. 电芯电压测量诊断

在开始电芯电压测量比较之前，主机确保：

- 在  $ADC\_CTRL2[AUX\_CELL\_SEL4:0]$  设置中配置所需的待测试 AUXCELL 通道，并且 AUX ADC 被启用并处于连续模式。
- 在开始该诊断检查之前，允许 AUX ADC 运行器件的所有 AUXCELL 通道，以补偿共模误差。
- 主 ADC 必须启用并处于连续模式。
- 通过  $DIAG\_COMP\_CTRL1[VCCB\_THR4:0]$  设置选择 (VCELL - AUXCELL) 比较阈值。
- 通过  $ADC\_CONF1[AUX\_SETTLE1:0]$  为 AUX CELL 通道选择所需的稳定时间。

要开始电芯电压测量比较，请执行以下操作：

1. 设置  $DIAG\_COMP\_CTRL3[COMP\_ADC\_SEL2:0] =$  电芯电压测量检查 ( 即 0b001 ) 并设置  $[COMP\_ADC\_GO] = 1$ 。
2. 对于  $[AUX\_CELL\_SEL4:0]$  启用的每个通道, 器件将比较  $abs[(VCELL - AUXCELL)] < [VCCB\_THR4:0]$ 。
3. 等待比较完成, 时间大致为  $[(通道数) * (AUXCELL 稳定时间 + 一轮循环时间)]$ 。
4. 当  $ADC\_STAT2[DRDY\_VCCB] = 1$  时, 电芯电压测量比较完成。

主机检查  $FAULT\_COMP\_VCCB1$  和  $FAULT\_COMP\_VCCB2$  寄存器以获得比较结果。

### ADC 比较中止条件：

在下面列出的无效条件下, 器件不会启动电芯电压测量比较。当比较中止时,  $FAULT\_COMP\_MISC[COMP\_ADC\_ABORT] = 1$ ,  $[DRDY\_AUX\_CEL] = 1$ ,  $[DRDY\_VCCB] = 1$ , 并且  $FAULT\_COMP\_VCCB1/2$  寄存器 = 0xFF。如果  $[AUX\_CELL\_SEL4:0]$  设置为锁定在单个通道上, 则在比较运行中止时  $AUX\_CELL\_HI/LO$  寄存器将被重置为默认值 0x8000。

会阻止开始电芯电压测量比较的无效条件或设置：

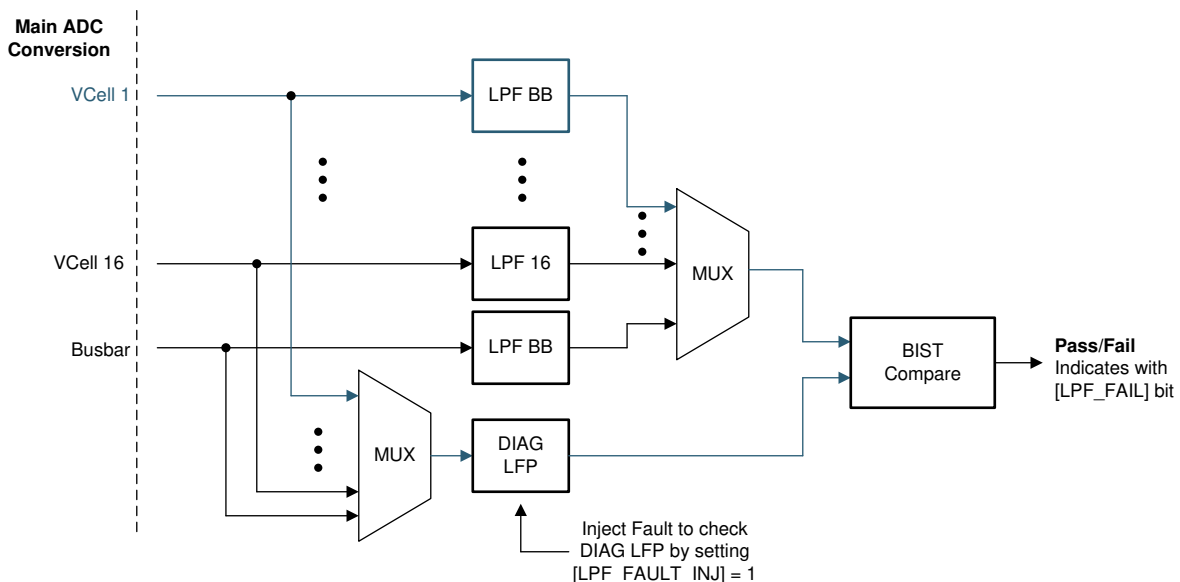
- 无效  $[AUX\_CELL\_SEL]$  设置：导致所选通道上没有 AUX ADC 测量。  $AUX\_CELL\_HI/LO$  寄存器保持默认值。
- 选择了高于 NUM\_CELL 配置的通道。
- 无效的 BBVC\_POSN 设置：
  - 在  $BBVC\_POSN1/2$  寄存器中启用了相邻通道。
  - 启用了  $BBVC\_POSN2[CELL1]$ 。
  - 在  $BBVC\_POSN1/2$  中选择了两个以上的通道。
  - $[AUX\_CELL\_SEL]$  被锁定到  $BBVC\_POSN1/2$  中所选的任何通道。
- 主 ADC 或 AUX ADC 关闭或未设置为连续模式。

### 后 ADC 数字 LPF 检查：

只要主 ADC 在运行, 就会连续检查数字 LPF。实施重复的诊断 LPF 来检查每个 VC 通道和 BBP/N 通道的每个 LPF。一次对一个 LPF 执行检查。

例如, 要测试电芯通道 1 的 LPF1, 输入 ( 即电芯 1 的 ADC 测量结果 ) 会馈送到 LPF1 和诊断 LPF 一段时间。LPF1 和诊断 LPF 的输出相互比较。将比较 LPF1 和诊断 LPF 的多个输出, 以确保 LPF1 的运行, 然后再检查下一个 LFP。如果任何 LPF 未能通过诊断检查, 则  $FAULT\_COMP\_MISC[LPF\_FAIL] = 1$ 。

当每个活动电芯通道的 LPF 被测试一次时,  $ADC\_STAT2[DRDY\_LPF] = 1$ 。只要主 ADC 在运行, LPF 的诊断检查就会在后台持续运行。



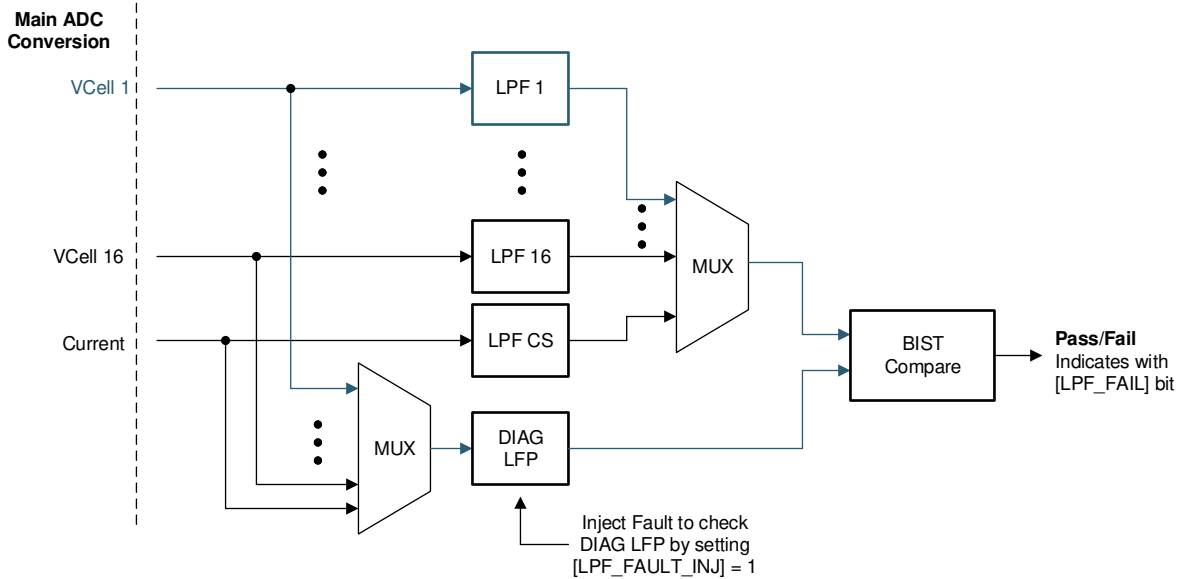


图 6-48. 后 ADC LPF 诊断 (蓝色路径是检查 LPF1 的示例)

此外，器件还执行检查以验证诊断 LPF 本身的功能。通过设置  $DIAG\_COMP\_CTRL4[LPF\_FAULT\_INJ] = 1$  并重新启动主 ADC，器件将向诊断 LPF 注入故障，从而在 LPF 诊断检查期间强制发生故障，然后设置  $[LPF\_FAIL] = 1$ 。测试完成后，只需设置  $[LPF\_FAULT\_INJ] = 0$ 。

#### 6.3.6.4.6.2 温度测量检查

与电芯电压测量检查类似，该器件通过将主 ADC 测量值与 AUX ADC 测量值进行比较来检查热敏电阻温度测量。为了读取主 ADC 和 AUX ADC 测量的比较值，MCU 使用  $[AUX\_GPIO\_SEL]$  设置锁定单个通道并启动该诊断检查。在该配置中，主 ADC 和 AUX ADC 的比较值将分别报告给  $DIAG\_MAIN\_HI/LO$  寄存器和  $DIAG\_AUX\_HI/LO$  寄存器。

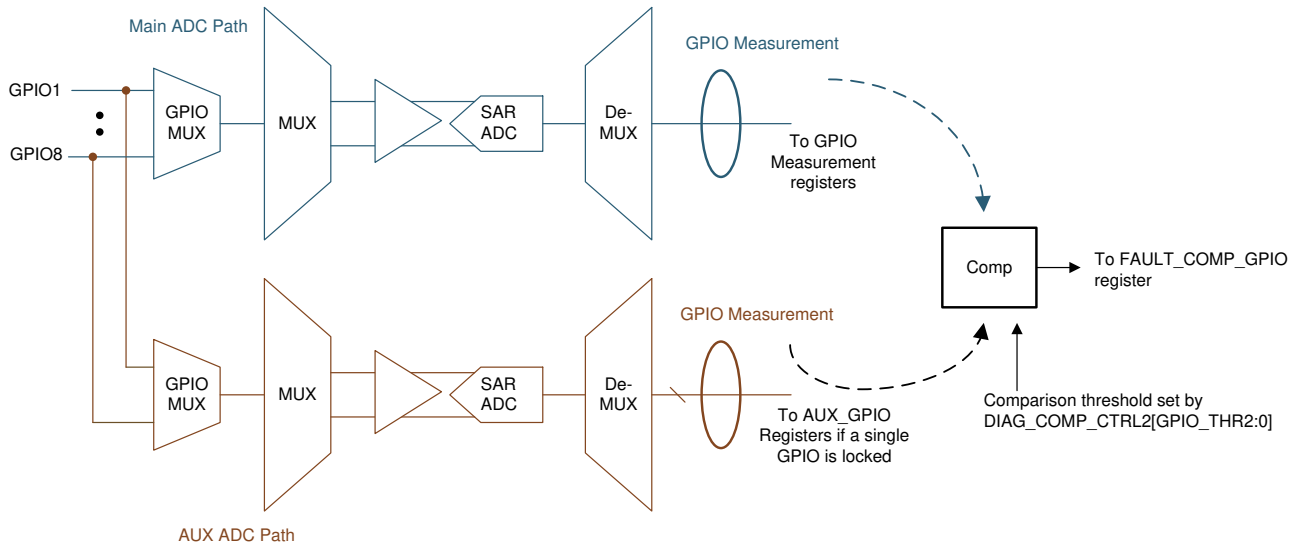


图 6-49. 热敏电阻温度 (GPIO) 测量诊断

在开始温度测量比较之前，主机确保：

- 主 ADC 必须启用并处于连续模式。

- 要测试的所需 GPIO 通道在  $ADC\_CTRL3[AUX\_GPIO\_SEL3:0]$  设置中进行配置，并且 AUX ADC 已启用并处于连续模式。
- 通过  $DIAG\_COMP\_CTRL2[GPIO\_THR2:0]$  设置选择比较阈值。

要开始电芯电压测量比较，请执行以下操作：

1. 设置  $DIAG\_COMP\_CTRL3[COMP\_ADC\_SEL2:0] = GPIO$  测量检查 (即 0b101) 并设置  $[COMP\_ADC\_GO] = 1$ 。
2. 对于  $[AUX\_GPIO\_SEL4:0]$  启用的每个通道，该器件将比较  $abs[(来自主 ADC 的 GPIO - 来自 AUX ADC 的 GPIO)] < [GPIO\_THR2:0]$ 。
3. 等待比较完成，这可能需要多达 64 个 ADC 轮询时间。
4. 当  $ADC\_STAT2[DRDY\_GPIO] = 1$  时，GPIO 测量比较完成。

主机检查  $FAULT\_COMP\_GPIO$  寄存器以获取比较结果。

#### ADC 比较中止条件：

在下面列出的无效条件下，器件不会启动温度测量比较。当比较中止时， $FAULT\_COMP\_MISC[COMP\_ADC\_ABORT] = 1$ 、 $[DRDY\_GPIO] = 1$  且  $FAULT\_COMP\_GPIO = 0xFF$ 。如果  $[AUX\_GPIO\_SEL3:0]$  设置为锁定在单个通道上，则在比较运行中止时  $AUX\_GPIO\_HI/LO$  寄存器将被重置为默认值 0x8000。

会阻止开始温度测量比较的无效条件或设置：

- $[AUX\_GPIO\_SEL]$  设置无效，所选的 GPIO 未配置为进行 ADC 测量。 $AUX\_GPIO\_HI/LO$  寄存器保持默认值。这也适用于为所有 GPIO 选择了  $[AUX\_GPIO\_SEL]$  但没有将任何 GPIO 配置为进行 ADC 测量的情况。
- 主 ADC 或 AUX ADC 关闭或未设置为连续模式。

#### 6.3.6.4.6.3 电芯均衡 FET 检查

可以通过开启均衡 FET 并比较 FET 两端的电压 (通过 AUX ADC 路径) 与电芯电压 (通过主 ADC 路径) 来执行电芯均衡 FET 检查。要读取用于检查的 AUXCELL 测量值，MCU 必须使用  $[AUX\_CELL\_SEL]$  设置将该诊断检查设置为锁定在单个通道上，然后启动该诊断检查。AUXCELL 比较值将报告给  $DIAG\_AUX\_HI/LO$  寄存器。

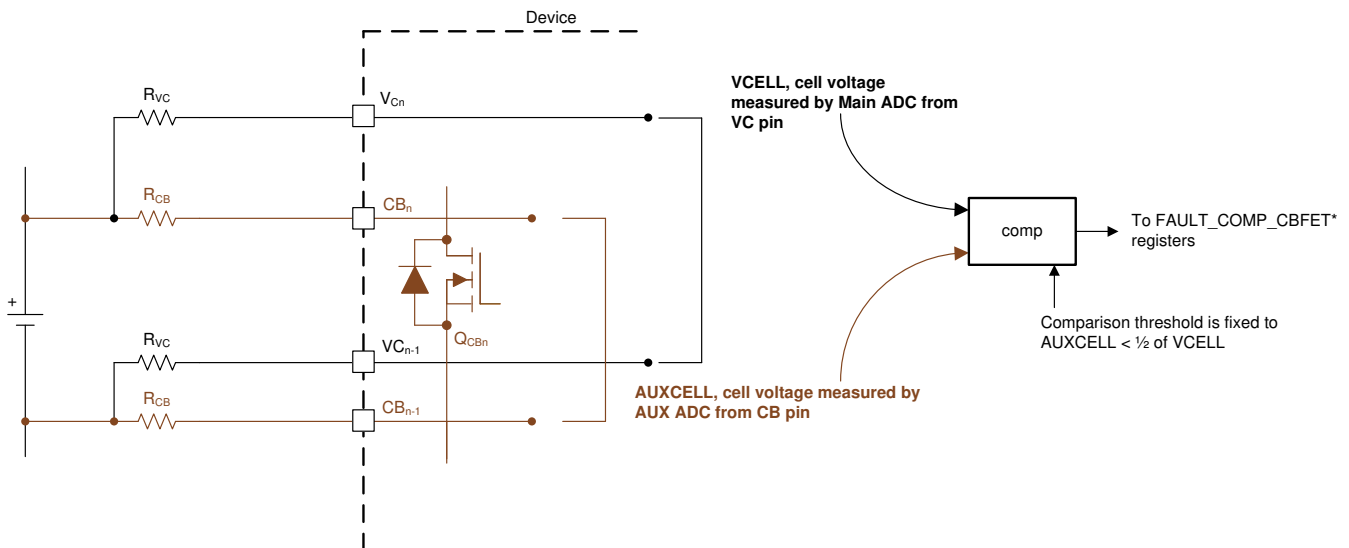


图 6-50. 电芯均衡 FET 诊断

在开始电芯均衡 FET 比较之前，主机确保：

- 主 ADC 以连续模式运行。

- 在 `ADC_CTRL2[AUX_CELL_SEL4:0]` 中进行配置，以选择测试 CB FET 的 AUXCELL 通道。
- 通过 `ADC_CONF1[AUX_SETTLE1:0]` 为 AUX CELL 通道选择所需的稳定时间。
- 如果均衡正在运行，则暂停 CB。
- 通过 `DIAG_CBFET_CTRL1` 和 `DIAG_CBFET_CTRL2` 寄存器配置要测试的 CBFET。
  - 最多开启 8 个 CBFET 以及开启不超过两个连续 CBFET 的规则仍然适用。
  - 建议以奇数和偶数方式进行测试。

要开始 CBFET 比较，请执行以下操作：

1. 以连续模式启动 AUX ADC。
2. 通过设置 `DIAG_COMP_CTRL3[CBFET_CTRL_GO] = 1` 开启选定的 CBFET 并等待适当的 dv/dt 时间。
3. 设置 `DIAG_COMP_CTRL3[COMP_ADC_SEL2:0] = CBFET 检查 (即 0b100)` 并设置 `[COMP_ADC_GO] = 1`。
4. 器件开启上一步中配置的 CBFET，并比较 AUXCELL 测量值 (通过 CB 通道) 是否小于 VCELL 测量值 (通过 VC 通道) 的一半。仅会检查已启用的 CBFET。
5. 当 `ADC_STAT2[DRDY_CBFET] = 1` 时，CBFET 比较完成。
6. 对其他 CBFET 测试设置重复该过程。为了关闭为该测试启用的 CBFET，MCU 清除 `DIAG_CBFET1` 和 `DIAG_CBFET2` 寄存器，然后设置 `[CBFET_CTRL_GO] = 1`。否则，通过发送 `[CB_PAUSE] = 0` 退出 CB 暂停状态将恢复常规均衡，从而关闭为该测试启用的 CBFET 并在为均衡设置的 CBFET 上恢复。

主机检查 `FAULT_COMP_CBFET1` 和 `FAULT_COMP_CBFET2` 寄存器以获得比较结果。重复这些步骤以比较剩余的 CBFET。

#### ADC 比较中止条件：

在下面列出的无效条件下，器件不会启动 CBFET 比较。当比较中止时，`FAULT_COMP_MISC[COMP_ADC_ABORT] = 1`，`[DRDY_AUX_CEL] = 1`，`[DRDY_CBFET] = 1`，并且 `FAULT_COMP_CBFET1/2 = 0xFF`。如果 `[AUX_CELL_SEL4:0]` 设置为锁定在单个通道上，则在比较运行中止时 `AUX_CELL_HI/LO` 寄存器将被重置为默认值 `0x8000`。

会阻止开始电芯电压测量比较的无效条件或设置：

- 无效 `[AUX_CELL_SEL]` 设置，导致所选通道上没有 AUX ADC 测量。`AUX_CELL_HI/LO` 寄存器保持默认值。
- 选择了高于 `NUM_CELL` 配置的通道。
- 无效的 `BBVC_POSN` 设置：
  - 在 `BBVC_POSN1/2` 寄存器中启用了相邻通道。
  - 启用了 `BBVC_POSN2[CELL1]`。
  - 在 `BBVC_POSN1/2` 中选择了两个以上的通道。
  - `[AUX_CELL_SEL]` 被锁定到 `BBVC_POSN1/2` 中所选的任何通道。
- 主 ADC 或 AUX ADC 关闭或未设置为连续模式。
- CB 正在运行且未处于暂停模式。
- 在 `DIAG_CBFET_CTRL1/2` 寄存器中启用 8 个以上的 CBFET，或启用两个以上的连续 CBFET。

#### 6.3.6.4.6.4 VC 和 CB 开路检查

该器件可以检测 VC 和 CB 引脚上的开路连接。每个 VC 和 CB 引脚均连接了一个电流阱，但 VC0 和 CB0 引脚除外，这些引脚连接了电流源。

当电流阱 (或电流源) 启用并且存在开路连接时，外部差分电容器将耗尽，电芯电压测量值将随着时间的推移下降到异常水平。类似的检测概念适用于带有电流源的 VC0 和 CB0 引脚。如果存在开路连接，VC0 或 CB0 将被电流源上拉，从而使电芯电压测量值随着时间的推移而降低。

启用诊断比较后，器件会将来自主 ADC 的电池电压测量值 (用于 VC 引脚开路检测) 与主机编程的阈值进行比较；或者将来自 AUX ADC 的 AUX 电芯测量值 (用于 CB 引脚开路检测) 与主机编程的阈值进行比较。

如果 MCU 在开始 CB 开路检查之前通过 `[AUX_CELL_SEL]` 锁定至单个 CB 通道，则器件会报告用于检查比较的 AUX 电芯测量值。该值在 `DIAG_AUX_HI/LO` 寄存器中报告。由于主 ADC 中没有单通道锁定机制，因此用于 VC 开路的 VC 通道测量值将不会在 `DIAG_MAIN_HI/LO` 寄存器中报告。

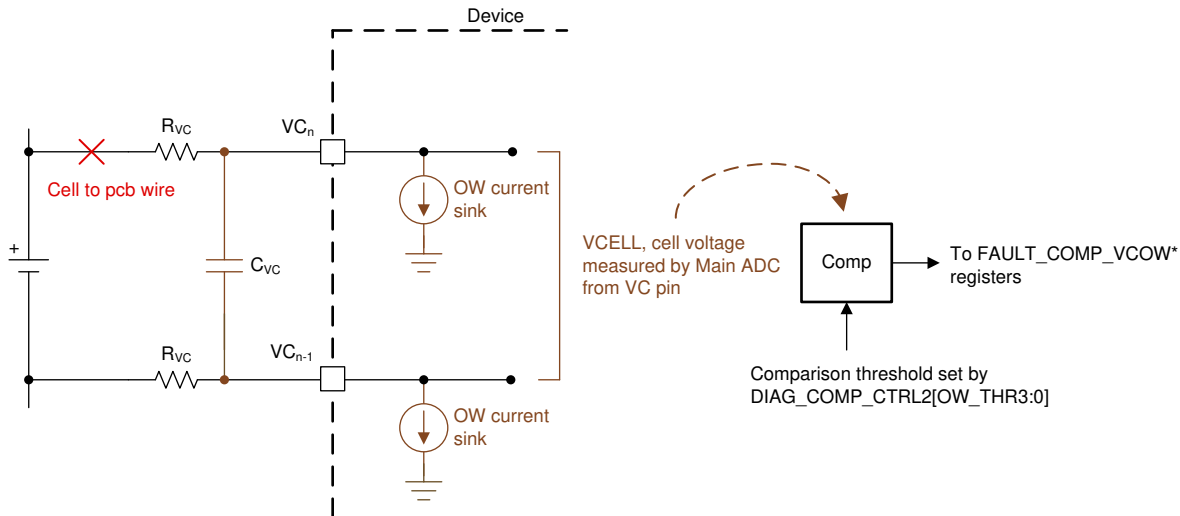


图 6-51. 开路检测

在开始开路比较之前，主机确保：

- 对于 VC 开路检测，主 ADC 以连续模式运行。
- 对于 CB 开路检测，AUX ADC 以连续模式运行
  - 在 `ADC_CTRL2[AUX_CELL_SEL4:0]` 中配置以选择 AUX 电芯通道
  - 通过 `ADC_CONF1[AUX_SETTLE1:0]` 为 AUX CELL 通道选择所需的稳定时间。
- 通过 `DIAG_COMP_CTRL2[OW_THR3:0]` 配置开路检测阈值。

要开始开路比较，请执行以下操作：

1. 通过 `DIAG_COMP_CTRL3[OW_SNK1:0]` 开启 VC 引脚（或 CB 引脚）电流阱或电流源。
2. 如果存在开路故障，则等待外部电容器的  $dV/dt$  时间消耗至检测阈值。
3. 对于 VC 开路检测，选择 `DIAG_COMP_CTRL3[COMP_ADC_SEL2:0] = OW VC 检查`（即 0b010）并设置 `[COMP_ADC_GO] = 1`。或者对于 CB 开路检测，`[COMP_ADC_SEL2:0] = OW CB 检查`（即 0b011）。
4. 该器件将所有有效 VCELL 测量值（对于 VC 开路）或 AUX 电芯测量值（对于 CB 开路）与 `[OW_THR3:0]` 阈值设置进行比较。
5. 比较完成后，`ADC_STAT2[DRDY_VCOW] = 1`（对于 VC 开路）或 `[DRDY_CBOW] = 1`（对于 CB 开路）。
6. 然后主机通过 `DIAG_COMP_CTRL3[OW_SNK1:0]` 关闭所有电流阱和电流源。

主机检查 `FAULT_COMP_VCOW1/2` 或 `FAULT_COMP_CBOW1/2` 寄存器以获取比较结果。

### 6.3.7 汇流条支持

该器件支持在两种类型的连接中进行汇流条测量：

- 汇流条通过 BBP 和 BBN 引脚连接到专用汇流条通道
- 汇流条连接到 VC 通道

单个器件总共可以连接三个汇流条，一个通过 BBP/N 引脚进行连接，两个通过 VC 通道进行连接。表 6-32 展示了这两种连接方法之间的差异。后续各节将对此进行详细介绍。

表 6-32. 汇流条连接方法

支持功能/限制	汇流条连接在 BBP/BBN 引脚上	汇流条单独连接在 VC 引脚上
每个器件可以连接的汇流条数量	1	2
连接通道	可以连接到除底部通道之外的任何通道	可以连接到除底部通道之外的任何通道
汇流条测量	是，结果输出到 <i>BUSBAR_HI/LO</i> 寄存器	是，结果输出到 <i>VCELLx_HI/LO</i> 寄存器，其中 x 是汇流条连接到的 VC 通道
集成滤波器用于汇流条测量	是，前端滤波器与常规电芯通道和后 ADC 数字 LPF 相同。BBP/N 通道具有 5 倍增益，输入范围为 $\pm 1V$ 。专用数字 LPF 设置与用于电芯电压测量的 LPF 设置分离。	是，前端滤波器与常规电芯通道和后 ADC 数字 LPF 相同。使用相同的电芯电压 LPF 设置进行汇流条测量。
主机需要调整电芯测量	是，使用该方法，电芯 + 汇流条共享单个 VC 通道。主机需要分离出汇流条测量以获得共享通道上的实际电芯测量值。	否，电芯和汇流条连接到各自的 VC 通道，并且单独报告测量结果
电芯均衡限制	无 CB 限制，但在均衡汇流条连接的通道上方的电芯时，主机会开启相邻的 CBFET	无 CB 限制，但需要将汇流条连接的通道上的上部 CB 引脚悬空

当汇流条连接到 VC 通道时，该器件支持在连接中测量汇流条。总共可以通过 VC 通道将两个汇流条连接到单个器件。表 6-33 展示了后续各节中介绍的详细信息。

表 6-33. 汇流条连接方法

支持功能/限制	汇流条单独连接在 VC 引脚上
每个器件可以连接的汇流条数量	2
连接通道	可以连接到除底部通道之外的任何通道
汇流条测量	是，结果输出到 <i>VCELLx_HI/LO</i> 寄存器，其中 x 是汇流条连接到的 VC 通道
集成滤波器用于汇流条测量	是，前端滤波器与常规电芯通道和后 ADC 数字 LPF 相同。使用相同的电芯电压 LPF 设置进行汇流条测量。
主机需要调整电芯测量	否，电芯和汇流条连接到各自的 VC 通道，并且单独报告测量结果
电芯均衡限制	无 CB 限制，但需要将汇流条连接的通道上的上部 CB 引脚悬空

### 6.3.7.1 BBP/BBN 引脚上的汇流条

该器件通过 BBP/BBN 引脚提供专用汇流条通道，用于汇流条连接和测量。该通道是一个悬空通道，允许汇流条连接至除模块底部电芯之外的任何电芯。使用汇流条通道可以最大限度地利用不同模块尺寸的器件中的电芯通道。

#### 6.3.7.1.1 典型连接

通过将汇流条连接到 BBP/BBN 引脚，允许单个电芯通道 (VC 通道) 与电芯 + 汇流条共享 (请参阅图 6-52 (a) 连接)。通常，这种连接会给系统的电芯测量带来额外的 IR 误差。该器件支持通过 BBP/BBN 引脚实现专用汇流条通道，从而允许主机测量汇流条电压，以获得实际的电芯测量值。

图 6-52 (a) 连接适用于连接到任何中间 VC 通道的汇流条。也就是说，在单个器件中，有一个电芯连接在 BBP/BBN 通道的上方和下方。为了支持汇流条通道上的热插拔，该器件仅需要在 BBP/N 引脚上各连接一个 400 $\Omega$  滤波电阻器，并在 BBP/N 引脚上各连接一个 0.47  $\mu$ F/16V 差分电容器。

如果连接到 BBP/N 的汇流条放置在模块顶部 (请参阅图 6-52 (b) 连接)，则此类连接是 BBP/N 情况下的例外情况，即不共享电芯通道。在该连接中，实际的电芯测量是通过 VC 通道进行的，主机不需要额外的计算。

图 6-52 (b) 连接适用于连接到模块顶部的汇流条，其中在单个器件中，汇流条上方未连接任何电芯。为了支持汇流条通道上的热插拔，除了 BBP/N 引脚上各自的 400 $\Omega$  滤波电阻器和 BBP/N 引脚上各自的 0.47  $\mu$ F/16V 差分电容器，需要在 BBN 和顶部 CB 引脚之间连接一个额外的 0.47  $\mu$ F/16V 差分电容器。该额外的电容器形成从模块中的所有电芯到汇流条的完整电容器梯，从而在热插拔期间允许在电容器梯上分配高尖峰电压。

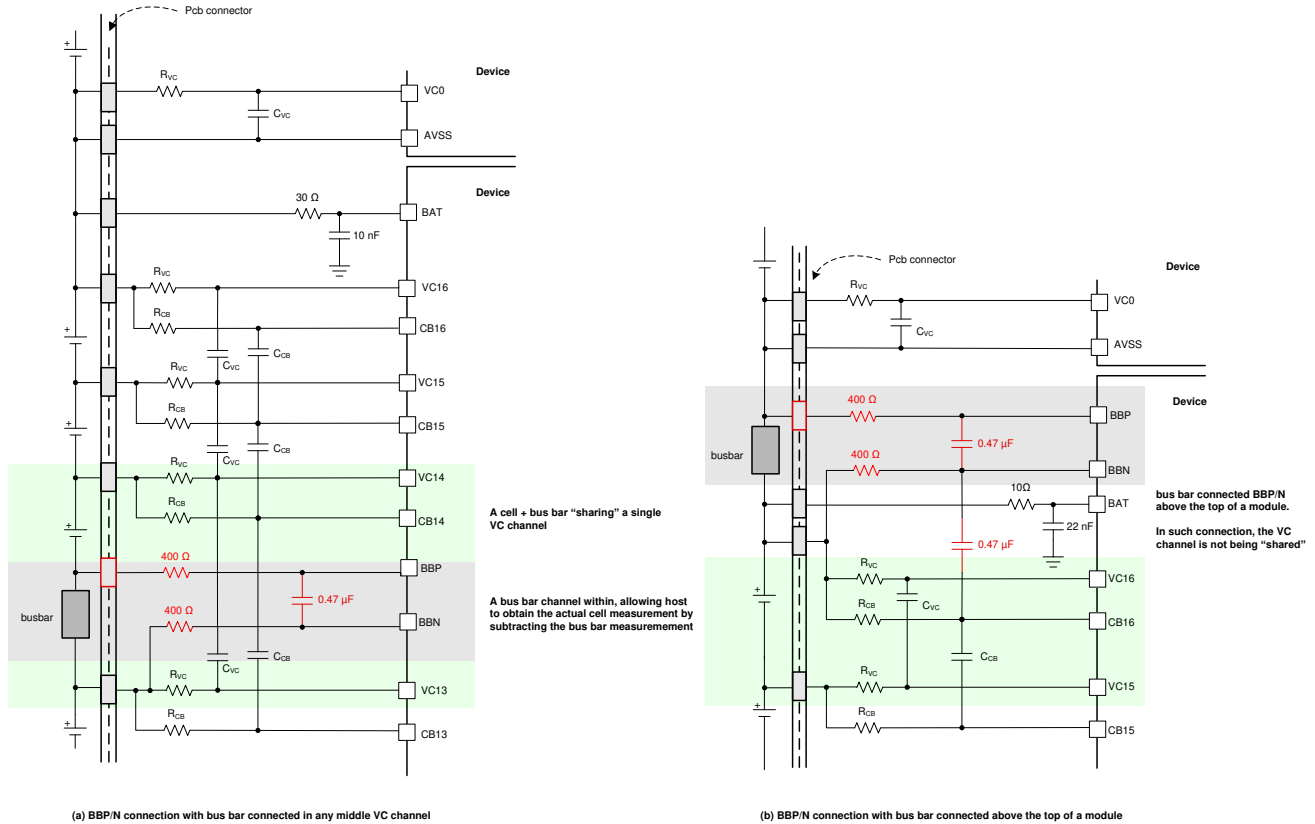


图 6-52. 连接在 BBP 和 BBN 引脚之间的汇流条

### 6.3.7.1.2 汇流条测量

(BBP-BBN) 上的差分测量由主 ADC 和 AUX ADC 执行。详细信息请见节 6.3.2.1.1 和节 6.3.2.2.1。可以使用 `BBP_LOC` 寄存器来指示与 `BBP/N` 连接共享的 VC 通道。利用该信息，器件可以更好地为最终的 ADC 测量提供共模校正。主机将知道共享 VC 通道中引入了额外的 IR 误差。如果启用 `OVUV` 保护器，则由于共享通道测量的额外 IR 增加（充电期间）或减少（放电期间），该共享通道可能会触发过早的 `OV` 或 `UV` 检测。

### 6.3.7.1.3 电芯均衡处理

由于汇流条与电芯到电芯通道共享，因此对电芯均衡控制没有特殊处理。主机将知道共享通道上的 `VCB_DONE` 检测（通过 VC 通道）中引入了额外的 IR 误差。

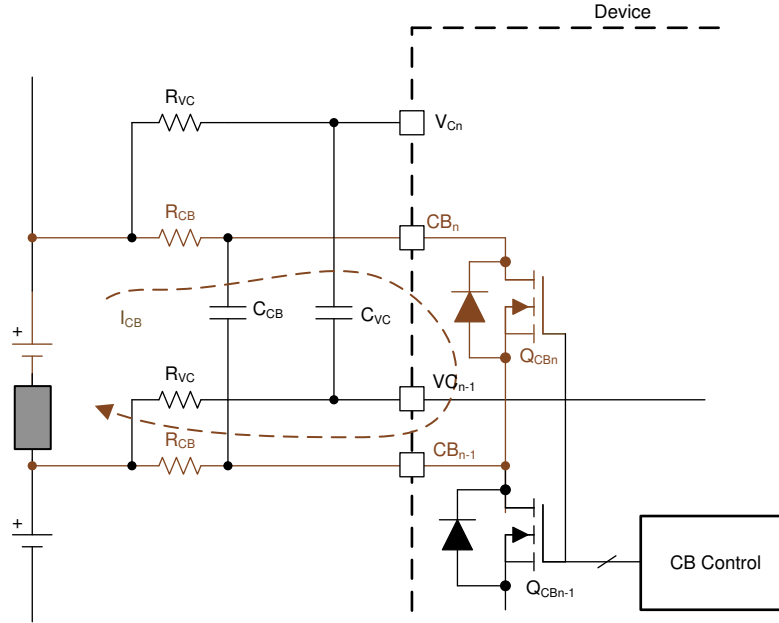


图 6-53. 通过共享电芯通道的汇流条 (通过 BBP/N) 进行电芯均衡

#### 6.3.7.1.4 电芯电压诊断控制

通过查看共享通道上的 (电芯 + 汇流条测量) 总和比较检查, 该器件仍然支持 VC 通道与 CB 通道。请参阅节 6.3.6.4.6.1。此外, 可以通过比较主 ADC 和 AUX ADC 的汇流条通道测量来检查汇流条测量。

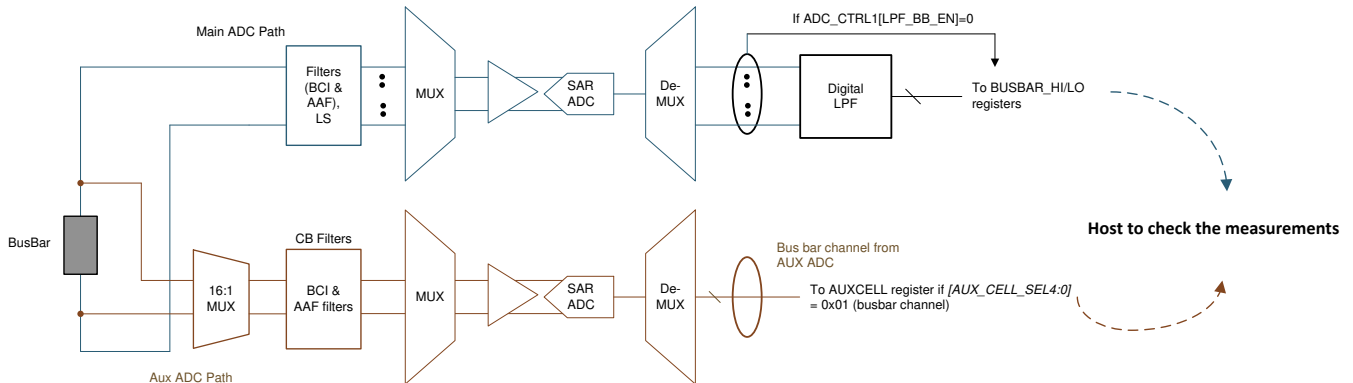


图 6-54. 通过 BBP/N 的汇流条测量检查

BBP/N 引脚具有用于开路检测的内置电流阱。该电流阱在  $DIAG\_COMP\_CTRL3[OW\_SNK1:0] = 0b11$  时开启。当有电流流过汇流条时, (BBP - BBN) 测量值不为零。如果 BBP 或 BBN 引脚上存在开路, 则电流阱会将 (BBP - BBN) 测量值更改为异常值。

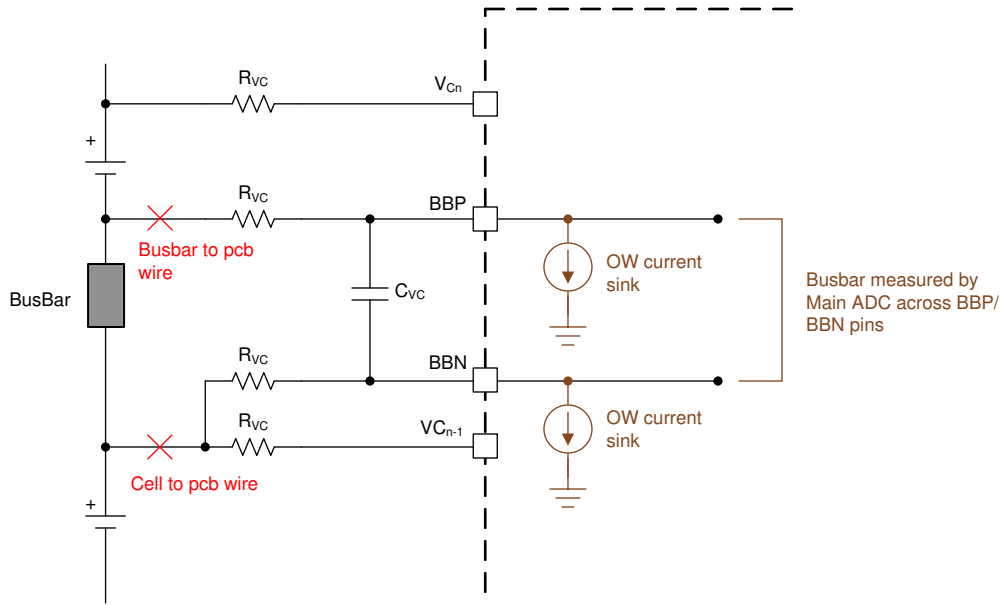


图 6-55. 用于 BBP/N 开路检测的电流阱

### 6.3.7.2 单独的 VC 通道上的汇流条

除了通过 BBP/N 连接汇流条之外，该器件还支持汇流条连接到单独的 VC 通道。除底部通道 (VC1-VC0) 之外，所有 VC 通道均支持 -2V 至 5V 测量。

该器件支持汇流条连接到单独的 VC 通道。除底部通道 (VC1-VC0) 之外，所有 VC 通道均支持 -2V 至 5V 测量。

当汇流条连接到单独的 VC 通道时，主机在 *BBVC\_POSN1* 和 *BBVC\_POSN2* 寄存器中指示汇流条位置。通过单独的 VC 通道连接汇流条时，不支持以下配置。对 *BBVC\_POSN1* 寄存器进行此类配置可能会导致均衡、OVUV 检测和电芯电压测量比较检查出现错误。

- 底部通道不支持汇流条连接。也就是说，*BBVC\_POSN1[CELL1]* 必须为 0。
- 通过该连接最多可以连接两个汇流条。也就是说，*BBVC\_POSN1* 寄存器中只有两个位被设置为 1。
- 汇流条无法连接到相邻的通道。

#### 6.3.7.2.1 典型连接

当汇流条单独连接到某个 VC 通道时，该通道上的上部 CB 引脚保持悬空，以避免内部 CBFET 正向偏置（请参阅图 6-56 (a) 连接）。该连接适用于单独连接到任何中间 VC 通道的汇流条。也就是说，在单个器件中，有一个电芯连接在 VC 通道的上方和下方，并连接了汇流条。为了确保热插拔性能，即使上部 CB 引脚悬空，连接汇流条的 CB 通道仍具有差分电容器。该电容器由所有电芯和连接到器件的汇流条形成完整的电容器梯，从而允许热插拔期间在电容器梯上分布高电压尖峰。

如果汇流条连接到模块顶部上方的单独 VC 通道（请参阅图 6-56 (b) 连接），则该通道上的上部 CB 引脚将保持悬空，但仍将连接 CB 差分电容器。此外，从顶部 CB 引脚到 BAT 引脚连接了一个额外的 RC 滤波器。该额外的 RC 滤波器（使用与 CB 引脚上的其他 RC 滤波器相同的 RC 值）用于确保为器件形成完整的电容器梯，以便在热插拔事件期间以与 CB 引脚复位相同的 RC 常数分配高电压尖峰。

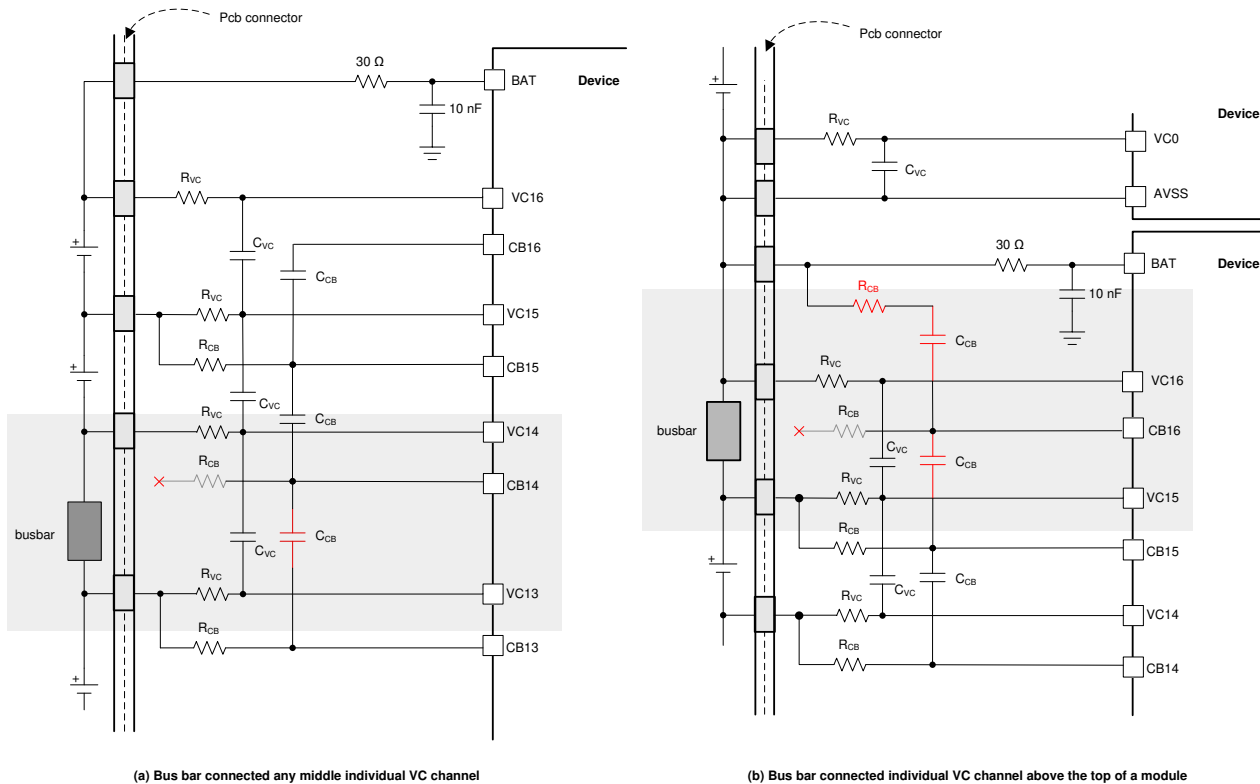


图 6-56. 连接到单个 VC 通道的汇流条

### 6.3.7.2.2 汇流条测量

汇流条测量通过作为 VC 通道之一的主 ADC 测量来执行。结果报告给 `VCELLx_HI/LO` 寄存器，其中 x 是与汇流条连接的通道。会启用数字 LPF 并将其作为 VC 通道测量配置的其余部分进行应用。

指示用于汇流条连接（通过 `BBVC_POSN1/2` 寄存器）的 VC 通道将在电芯均衡期间、OVUV 保护器启用时的 OV 和 UV 检测期间跳过 `VCB_DONE` 检查，并且会在电芯电压测量比较检查期间进行特殊处理。

### 6.3.7.2.3 电芯均衡处理

由于连接汇流条的通道上部 CB 引脚断开，因此为了均衡连接在汇流条上方的电芯，主机将开启相邻的 CBFET 并使用相同的计时器设置进行配置。

主机配置 `BBVC_POSN1/2` 寄存器以指示汇流条连接。该信息用于避免与汇流条连接的通道触发 `VCB_DONE` 检测并关闭其 CBFET，从而断开汇流条上方电芯的均衡路径。

汇流条上方电芯的均衡仍会根据计时器和电芯电压阈值终止，当满足停止条件之一时，其 CBFET 将关闭。即使汇流条连接的通道上的 CBFET 保持导通状态，均衡路径也会断开。

### 备注

汇流条连接的通道上的 CBFET 将导通，直到计时器到期。即使实际的电芯均衡已完成，这也可能导致 `[CB_DONE] = 1` 的延迟标记。

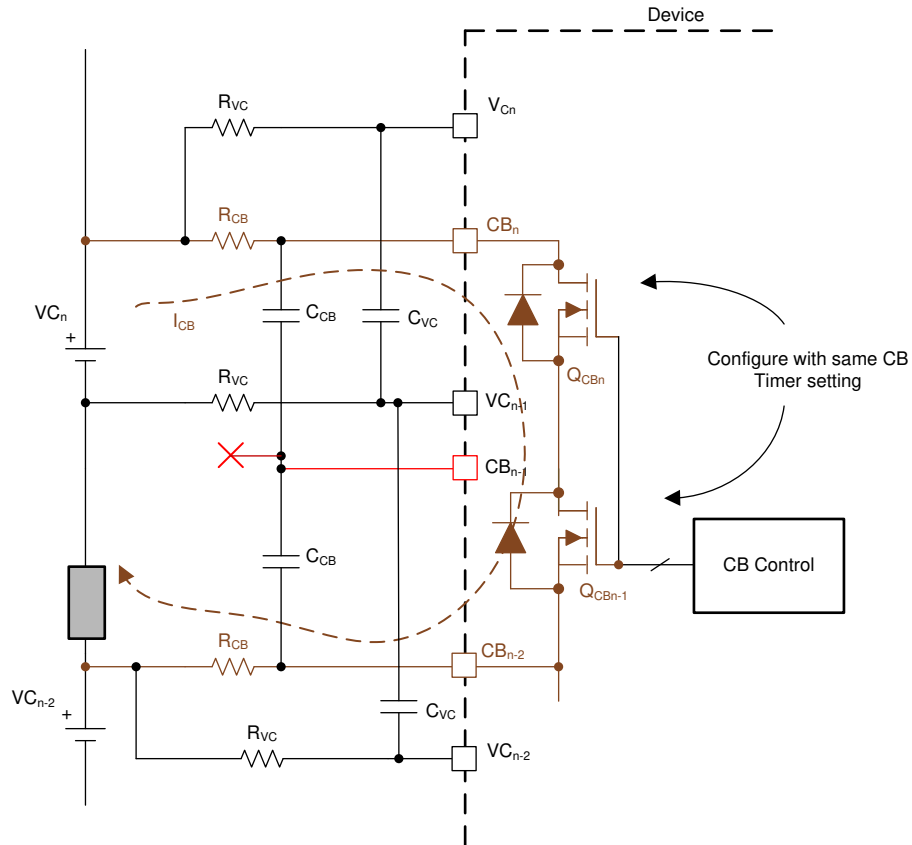


图 6-57. 通过连接到单独 VC 通道的汇流条进行电芯均衡

#### 6.3.7.2.4 电芯电压诊断控制

仍然会通过检查主 ADC 测量与 AUX ADC 测量来执行电芯电压比较检查。由于连接了汇流条的 CB 通道的上部 CB 引脚处于开路状态，因此器件通过比较来自主 ADC 的 ( 电芯 + 汇流条 ) 之和与来自 AUX ADC 的 ( 电芯 + 汇流条 ) 之和来处理比较检查。

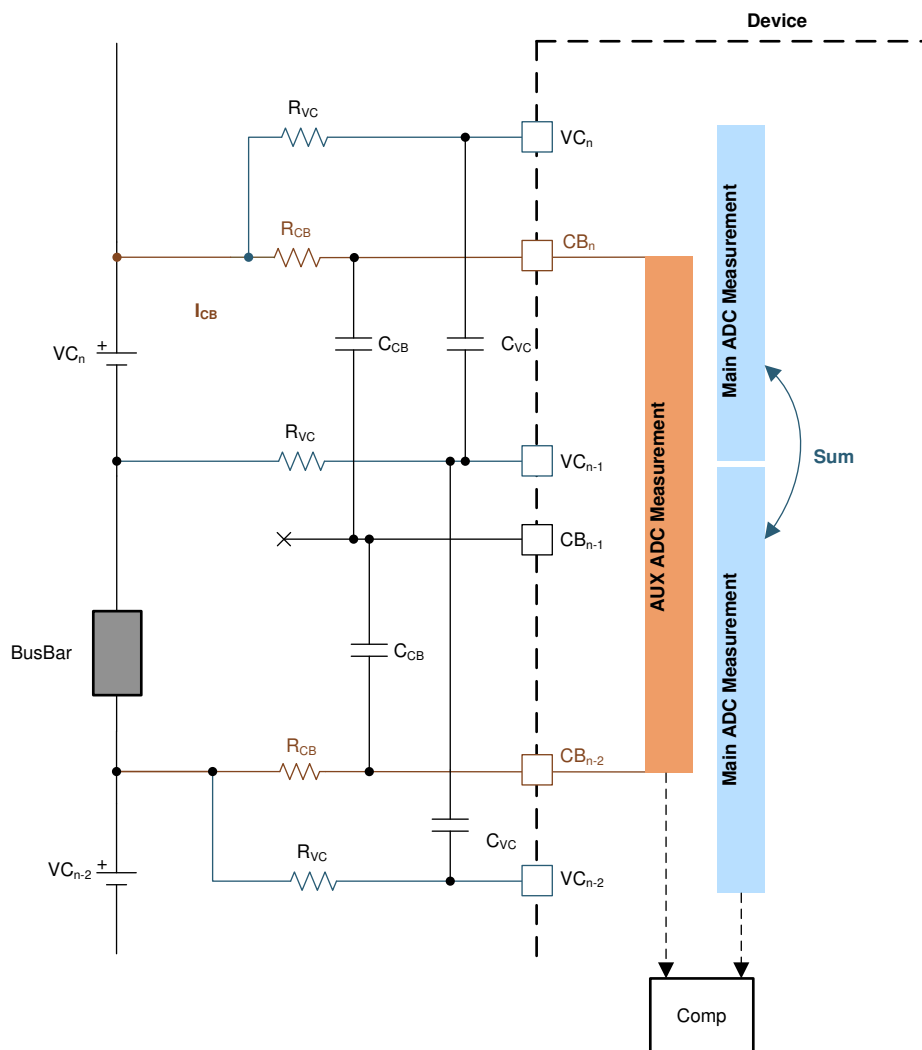


图 6-58. 汇流条连接到单独 VC 通道时的电芯测量检查

## 6.4 器件功能模式

该器件具有三种功耗模式和一个 POR 状态。

- **POR**：这不是功耗模式。在该情况下，BAT 引脚上的电压低于 VBAT 最小值，并且器件中包括 AVAO\_REF 块在内的所有电路都会断电。
- **SHUTDOWN**：这是最低功耗模式。AVDD、DVDD 和 CVDD 电源关闭。仅维持 LDOIN 引脚上的总调节。CVDD 引脚将通过内部电路具有与 LDOIN 引脚相似的电压，以支持 WAKE 检测。
- **SLEEP**：这是低功耗运行模式。仅提供有限的功能。
- **ACTIVE**：这是全功率运行模式。在该状态下支持所有功能。

表 6-34 总结了不同功耗模式下支持的各种功能，图 6-59 展示了功耗状态图。

**表 6-34. 有效功能总结**

功能模块	SHUTDOWN	SLEEP	ACTIVE	POR
主 ADC			√	这不是功耗状态。所有电路均关闭。VBAT 上足够的电压将对器件进行 POR 并将其置于 SHUTDOWN 模式
AUX ADC			√	
OV/UV 保护器		√ <sup>(1)</sup>	√	
OT/UT 保护器		√ <sup>(1)</sup>	√	
Cell Balancing		√ <sup>(1)</sup>	√	
OTCB 检测		√ <sup>(1)</sup>	√	
模块均衡 (通过 MB_TIMER_CTRL 控制)			√	
UART			√	
垂直通信			√	
故障状态和 NFAULT 通信		√	√	
通信超时			√	
SLEEP 超时		√		
热关断检测		√	√	
SPI 控制器			√	
OTP 编程			√	
用于检测器件 POR 的常开模块	√	√	√	

- (1) 要在 SLEEP 模式下启用电芯均衡、OV/UV 或 OT/UT 保护器，主机必须首先在 ACTIVE 模式下启用这些功能，然后将器件置于 SLEEP 模式。

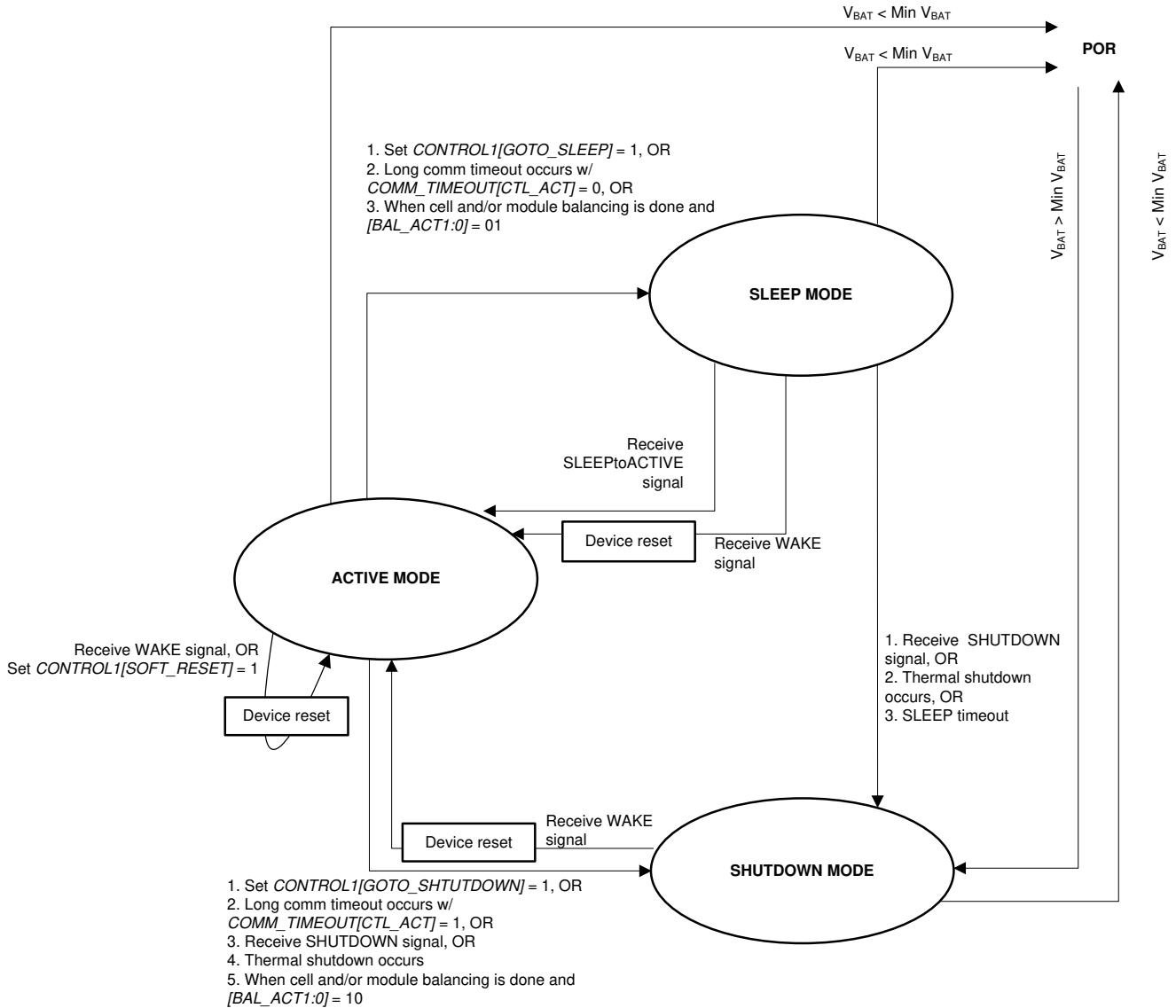


图 6-59. 功耗状态图

## 6.4.1 电源模式

### 6.4.1.1 SHUTDOWN 模式

这是最低功耗模式。在 SHUTDOWN 模式下，大部分功能都会关闭。器件保持空闲状态，仅监测 WAKE ping/音调（有关详细信息，请参阅节 6.4.3）以从该状态中唤醒。仅对 LDOIN 和 CVDD 引脚进行总体调节以用于 WAKE ping/音调检测。

#### 6.4.1.1.1 退出 SHUTDOWN 模式

SHUTDOWN 模式不支持通信，主机必须发送 WAKE ping 或 WAKE 音调才能进入 ACTIVE 模式。器件从 SHUTDOWN 模式转换为 ACTIVE 模式后，下表指示在发生此类转换时预期设置的故障位。

表 6-35. 器件从 SHUTDOWN 模式唤醒后的预期故障位

器件在菊花链中的位置	从 SHUTDOWN 模式唤醒后的预期故障位	
基底器件	$FAULT\_SYS[DRST] = 1$	通过唤醒 ping 进行数字复位
	$FAULT\_COMM3[FCOMM\_DET] = 1$	$[DRST] = 1$ 来自上部器件
	$FAULT\_COMM1[COMMCLR\_DET] = 1$	UART 引擎复位
堆栈器件 (栈顶除外)	$FAULT\_SYS[DRST] = 1$	通过唤醒音调进行数字复位
	$FAULT\_COMM3[FCOMM\_DET] = 1$	$[DRST] = 1$ 来自上部器件
栈顶器件	$FAULT\_SYS[DRST] = 1$	通过唤醒音调进行数字复位

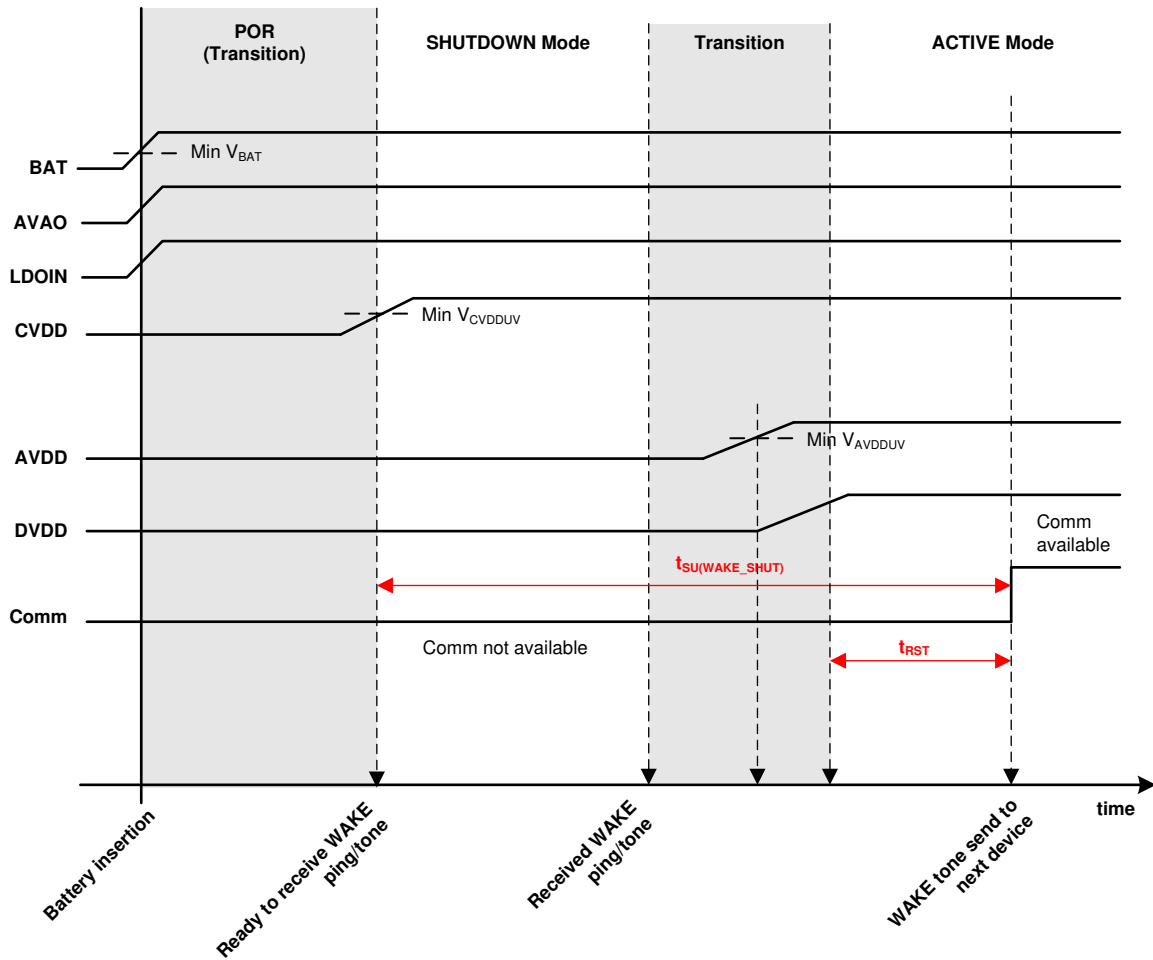


图 6-60. SHUTDOWN 至 ACTIVE 模式转换

#### 6.4.1.1.2 进入 SHUTDOWN 模式

在正常操作期间，主机通过发送  $CONTROL1[GOTO\_SHUTDOWN] = 1$  将器件置于 SHUTDOWN 模式。在菊花链配置中，使用广播写入发送此命令会将菊花链中的所有器件置于 SHUTDOWN 模式。

器件还可以通过以下条件之一进入 SHUTDOWN 模式：

- 通信超时：如果在配置的时间内没有有效通信，则自动从 ACTIVE 模式转换至 SHUTDOWN 模式。主机可以通过 `COMM_TIMEOUT_CONF` 寄存器启用该选项。
- SLEEP 模式超时：如果器件处于 SLEEP 模式达到配置的时间，则自动从 SLEEP 模式转换至 SHUTDOWN 模式。主机可以通过 `PWR_TRANSIT_CONF[SLP_TIME2:0]` 启用该选项。
- 均衡完成后：当器件的所有均衡完成后，自动进入 SHUTDOWN 模式。有关详细信息，请参阅节 6.3.3。此选项可能会导致菊花链配置下的器件在一段时间内处于不同的功耗模式。
- 热关断：当内部内核温度大于  $T_{SHUT}$  时关闭器件
- SHUTDOWN 或 HW\_RESET ping/音调：这些 ping/音调用作通信中断情况下的恢复尝试。SHUTDOWN ping/音调会在不使用通信的情况下将器件置于 SHUTDOWN 模式，从而强制关闭大部分电路。更积极的恢复尝试使用 HW\_RESET ping/音调，这会关闭除带隙之外的所有电路并在 SHUTDOWN 模式下重新启动器件。

#### 6.4.1.2 SLEEP 模式

这是低功耗运行模式。在 SLEEP 模式下，所有内部电源仍然开启，但功能仅限于电芯均衡、OVUV 和 OTUT 保护器、检测信号/故障音调/NFAULT 传输和检测。

##### 6.4.1.2.1 退出 SLEEP 模式

由于主机无法与器件通信，因此要退出 SLEEP 模式，主机必须发送 WAKE ping/音调或 SLEEPtoACTIVE ping/音调才能转换至 ACTIVE 模式。WAKE 唤醒并重置器件，主机需要重新配置器件设置；SLEEPtoACTIVE 仅唤醒器件。

##### 6.4.1.2.2 进入 SLEEP 模式

器件只能从 ACTIVE 模式进入 SLEEP 模式。在正常运行期间，主机通过通信（通过发送 `CONTROL1[GOTO_SLEEP] = 1`）将器件置于 SLEEP 模式。在菊花链配置中，使用广播写入发送该命令可使所有器件进入 SLEEP 模式。

在以下条件下，器件也可以进入 SLEEP 模式：

- 通信超时：如果在配置的时间内没有有效通信，则自动从 ACTIVE 模式转换至 SLEEP 模式。主机可以通过 `COMM_TIMEOUT_CONF` 寄存器启用该选项。

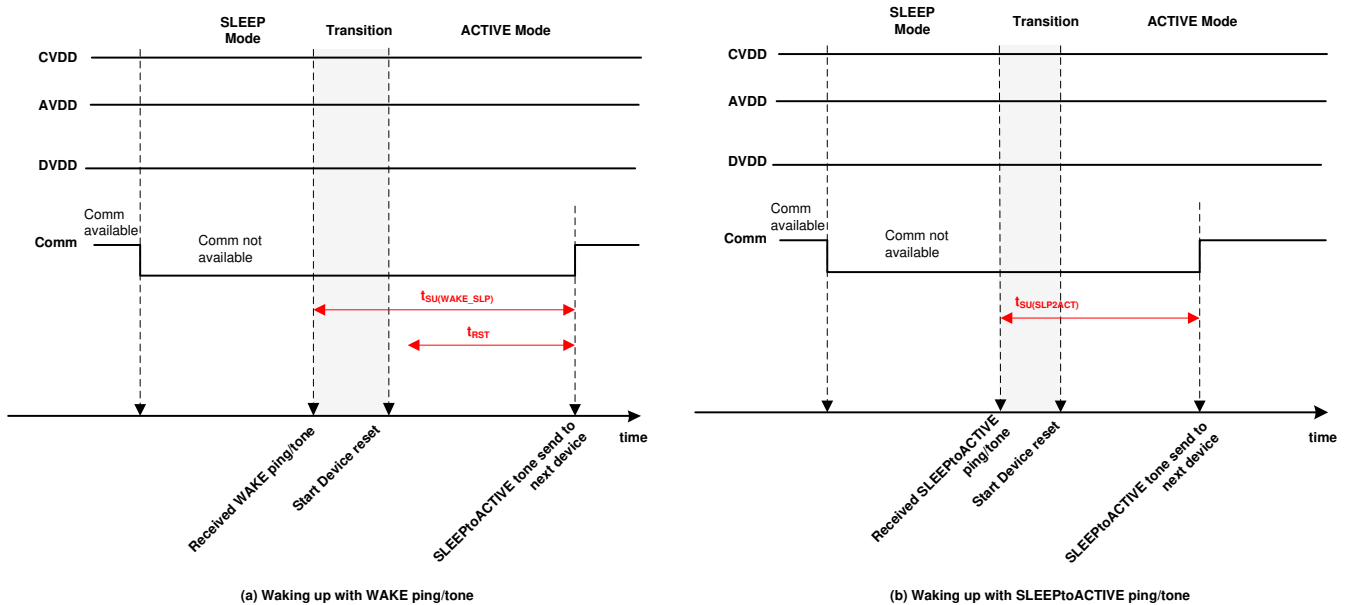


图 6-61. SLEEP 至 ACTIVE 模式转换

#### 6.4.1.3 ACTIVE 模式

这是具有全功能支持的运行模式。在该模式下，主机可以与器件进行通信，完全控制各种功能以及性能诊断。

#### 6.4.1.3.1 退出 ACTIVE 模式

器件可以通过命令、ping/音调、计时器或特定事件从 ACTIVE 模式进入 SLEEP 或 SHUTDOWN 模式。详细信息请见节 6.4.1.1 和节 6.4.1.2。

#### 6.4.1.3.2 从 SHUTDOWN 模式进入 ACTIVE 模式

器件只能通过 WAKE ping/音调从 SHUTDOWN 模式转换至 ACTIVE 模式。进入 ACTIVE 模式后，主机会清除一些与复位相关的故障，这些故障是预期的故障（有关详细信息，请参阅节 6.4.1.1），指示由于从 SHUTDOWN 模式转换至 ACTIVE 模式而在某些块上发生 POR。寄存器重置为默认值；OTP 影子寄存器会重新加载 OTP 编程值。

#### 6.4.1.3.3 从 SLEEP 模式进入 ACTIVE 模式

WAKE 或 SLEEPtoACTIVE ping/音调可以将器件从 SLEEP 模式置于 ACTIVE 模式。WAKE ping/音调 可向器件生成数字复位。由于 LDO 电源在 SLEEP 模式期间保持开启状态，因此只会设置  $FAULT\_SYS[DRST] = 1$ ，只是发生了数字复位。会设置与复位相关的某些预期故障。有关详细信息，请参阅 SHUTDOWN 模式。寄存器重置为默认值；OTP 影子寄存器会重新加载 OTP 编程值。

如果使用 SLEEPtoACTIVE ping/音调将器件从 SLEEP 模式唤醒至 ACTIVE 模式，则器件会在不进行数字复位的情况下进入 ACTIVE 模式，但 UART 引擎会复位；因此，在基底器件中， $FAULT\_COMM1[COMMCLR\_DET] = 1$ ，the 器件会在进入 ACTIVE 模式后将其清除。

### 6.4.2 器件复位

设备将经历以下几个状态：数字复位、将寄存器置于默认设置以及重新加载 OTP。

- 发送 WAKE ping/音调以从 SHUTDOWN 模式或 SLEEP 模式转换为 ACTIVE 模式。
- 在 ACTIVE 模式下接收 WAKE ping/音调。
- 在 ACTIVE 模式下发送  $CONTROL1[SOFT\_RESET] = 1$  命令。
- 在任何功耗模式下发送 HW\_RESET ping/音调。这会针对器件生成类似 POR 的事件。当检测到 HW\_RESET ping/音调时，器件将在  $t_{HWRST}$  持续时间关闭除带隙之外的所有内部块。之后，器件将以 SHUTDOWN 模式重新启动。
- 内部电源故障。有关详细信息，请参阅节 6.3.6.4。
  - 检测到 AVDD UV、DVDD UV。
- HFO 或 LFP 看门狗故障会重置数字电路。

除了完全复位情况外，以下情况只会使 UART 引擎复位。这些情况主要影响基底器件，因为 UART 用于与主机 MCU 通信。在基底器件中，会设置  $FAULT\_COMM1[COMMCLR\_DET] = 1$ 。这些情况不会影响堆栈器件，因为 UART 在这些器件中处于非活动状态。

- 会发送 SLEEPtoACTIVE ping 从 SLEEP 模式转换为 ACTIVE 模式。
- 以下情况不仅会清除 UART 引擎并设置  $[COMMCLR\_DET] = 1$ ，还会设置  $FAULT\_COMM1[STOP\_DET] = 1$ ，以指示检测到意外的 UART 停止。
  - 在 ACTIVE 模式下会发送 SLEEPtoACTIVE ping。
  - 会发送 COMM CLEAR 信号。这是一个专用信号，用于清除 UART 引擎并指示引擎寻找新的通信开始帧。有关更多详细信息，请参阅节 6.3.6.1.1.1。

### 6.4.3 ping 和音调

在不可通信的情况（例如处于 SHUTDOWN 或 SLEEP 模式）或通信中断的情况下，当主机发送复位或断电指令作为通信恢复尝试时，ping 或音调是针对特定操作与器件进行通信的一种形式。

表 6-36. 不同功耗模式下支持的 ping 和音调

ping/音调检测	检测到的引脚	SHUTDOWN	SLEEP	ACTIVE
SHUTDOWN ping	RX		✓	✓
SLEEPtoACTIVE ping	RX		✓	✓
WAKE ping	RX	✓	✓	✓

表 6-36. 不同功耗模式下支持的 ping 和音调 (续)

ping/音调检测	检测到的引脚	SHUTDOWN	SLEEP	ACTIVE
HW_RESET ping	RX		✓	✓
SHUTDOWN 音调	COMH/L		✓	✓
SLEEPtoACTIVE 音调	COMH/L		✓	✓
WAKE 音调	COMH/L	✓	✓	✓
HW_RESET 音调	COMH/L		✓	✓
故障音调	COMH/L		✓	仅故障音调检测可用。
检测信号	COMH/L		✓	

### 6.4.3.1 Ping

ping 是发送到器件 RX 引脚的特定高-低-高信号。ping 在基底器件上使用，因为只有基底器件连接到可访问 UART RX 的主机。器件通过检测 ping 信号的不同低电平时间来区分不同的 ping 信号。

通信 ping 是指 WAKE ping、SLEEPtoACTIVE ping、SHUTDOWN ping 和 HW\_RESET ping。当正常通信不可用时，这些 ping 指示器件进入特定的功耗模式。根据定义，RX 引脚上的 COMM CLEAR 信号是 ping 的一种形式。由于 COMM\_CLR 用于清除 UART 引擎，因此节 6.3.6.1.1.1 介绍了该信号。

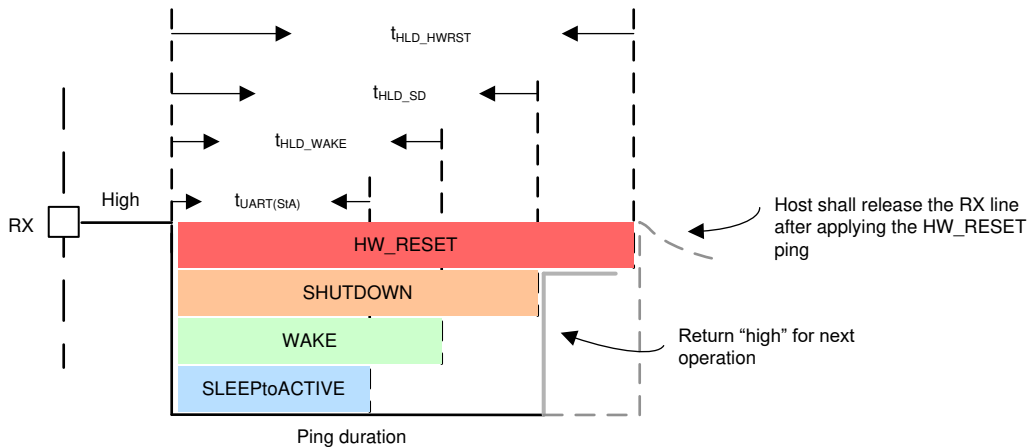


图 6-62. 通信 ping

### 6.4.3.2 音调

音调是通过差分垂直接口 COMH 和 COML 端口发送的固定数量具有指定极性 (全“+”或全“-”) 的对 (脉冲)。音调在堆栈器件上使用，因为只有 COMH/L 端口可访问。用于传输的对数始终大于检测所需的对数。

有四种通信音调对应于四种通信 ping。这些音调是 WAKE 音调、SLEEPtoACTIVE 音调、SHUTDOWN 音调和 HW\_RESET 音调。除了通信音调之外，还有两种与器件故障状态相关的额外音调：检测信号音调和故障音调。这两种故障状态音调仅在 SLEEP 模式下可用。有关详细信息，请参阅节 6.3.6.2.3.3。

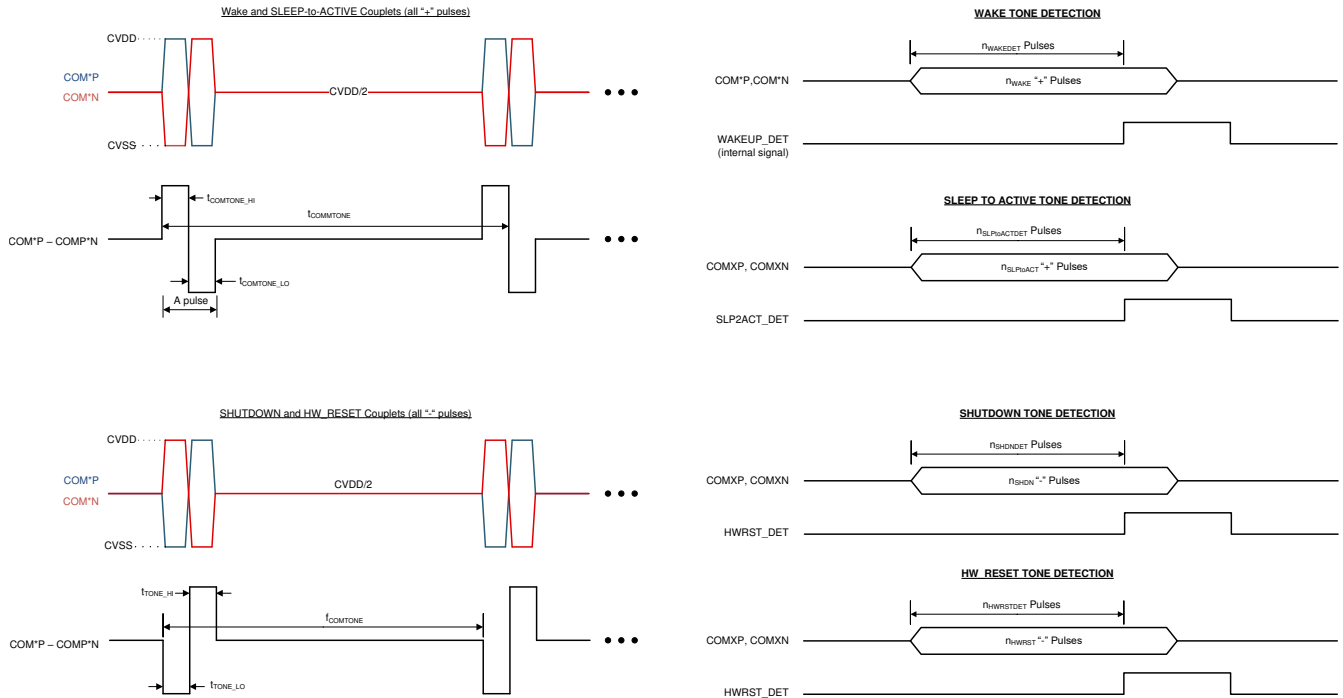


图 6-63. 通信音调

#### 6.4.3.3 ping 和音调传播

##### 传播：

WAKE 和 SLEEPtoACTIVE ping/音调是唤醒器件的正常操作的一部分；因此，这两个 ping/音调可以传播到菊花链配置中的下一个器件。也就是说，当器件接收到 WAKE ping/音调时，它会生成 WAKE 音调并将其转发到下一个器件。类似的操作适用于 SLEEPtoACTIVE ping/音调。

音调转发的方向遵循通信方向，该方向由 `CONTROL1[DIR_SEL]` 位设置。有关更多详细信息，请参阅节 6.3.6.1。无论 `[DIR_SEL]` 设置如何，堆栈器件上的 COMH 和 COML 端口都支持音调检测。这不适用于基底器件，因为基底器件会检测 ping。

在正常运行期间，主机只需向基底器件发送 WAKE 或 SLEEPtoACTIVE ping，就会向其余堆栈器件生成相应的音调。在系统开发过程中，如果需要仅向菊花链中的部分器件发送 WAKE 或 SLEEPtoACTIVE，则主机可以使用 `CONTROL1[SEND_WAKE]` 或 `CONTROL1[SEND_SLPTOACT]` 位。接收到该命令的器件将向菊花链中的下一个器件发送相应的音调。由于 WAKE 和 SLEEPtoACTIVE 音调会传播，因此上面连接的菊花链其余部分也会收到相应的音调。

##### 不传播：

SHUTDOWN 和 HW\_RESET ping/音调主要用作通信恢复尝试。因此，这些 ping/音调不会传播。也就是说，当器件接收到 SHUTDOWN ping/音调时，它会启动关闭过程，但该器件不会向下一个器件生成另一个 SHUTDOWN 音调。类似的操作适用于 HW\_RESET ping/音调。

对于基底器件，由于 RX 引脚连接到主机，因此可以在基底器件上使用 SHUTDOWN 或 HW\_RESET ping。对于堆栈器件，要求至少有一个堆栈器件可与问题器件通信。主机必须与相邻器件通信并设置 `CONTROL1[SEND_SHUTDOWN] = 1` 或 `CONTROL2[SEND_HW_RESET] = 1`，以指示相邻器件向问题器件发出相应的音调。无论 `[DIR_SEL]` 设置如何，堆栈器件上的 COMH 和 COML 端口都支持音调检测。这不适用于基底器件，因为基底器件会检测 ping。

表 6-37. ping 和音调传播总结

ping/音调	可传播	不可传播
WAKE	接收器件会向下一个器件发出 WAKE 音调	
SLEEPTOACTIVE	接收器件会向下一个器件发出 SLEEPTOACTIVE 音调	
SHUTDOWN		接收器件将初始化关闭过程
HW_RESET		接收器件将初始化硬件复位过程

## 6.5 寄存器映射

本节包含三个寄存器映射汇总表，其中按寄存器地址的顺序列出了寄存器：

- **NVM (OTP) 影子寄存器。**这些读取/写入影子寄存器通过在客户 OTP 空间中编程的 OTP 值进行重置。为了对自定义 OTP 空间进行编程，主机将所需的值写入这些 OTP 影子寄存器并按照编程过程进行操作。OTP CRC 校验涵盖这些寄存器。如果未对客户 OTP 空间进行编程，则影子寄存器中会加载出厂配置默认值。如果器件复位后 OTP ( 出厂配置默认值或客户 OTP 空间中编程的值 ) 无法加载，则影子寄存器将改为加载硬件复位默认值。对于大多数 OTP 影子寄存器，硬件复位默认值和出厂配置默认值相同。与出厂默认值相比，仅 *DIR0\_ADDR\_OTP*、*DIR1\_ADDR\_OTP*、*PWR\_TRANSIT\_CONF*、*CUST\_CRC\_HI/LO* 寄存器具有复位值，这在节 6.5.1 及其寄存器字段说明中进行了指定。
- **读取/写入寄存器。**这些是主机在运行时可以读取/写入的寄存器。器件复位会将这些寄存器恢复到其复位值。
- **读取寄存器。**这些是主机只能进行读取访问的寄存器。器件复位会将这些寄存器恢复到其复位值。

寄存器汇总表使用以下关键字：

- **Addr** = 寄存器地址
- **Hex** = 十六进制值
- **NVM** = 非易失性存储器 (OTP) 影子寄存器
- **RSVD** = 保留。保留的寄存器地址或位未在器件中实现。对这些位的任何写入都会被忽略。读取这些位始终返回 0。
- **OTP\_SPARE**：这些是器件中实现的备用 OTP 和影子寄存器位。这些备用位作为 CRC 计算的一部分而存在。这些位可以正常读取/写入，但不执行任何功能或影响任何器件行为。
- **OTP\_RSVDn** = 已实现但保留供器件内部使用的 OTP 和影子寄存器，其中 n 指寄存器地址。MCU 必须将这些寄存器保持为其默认值。
- **硬件复位默认值是数字复位 ( 类似 POR 事件 ) 时加载的值，而出厂配置默认值是加载到 OTP 单元中的默认值 ( 如果客户自己未对其进行编程 )。客户无法读取硬件复位值。**

节 6.5.4 介绍了这些寄存器中每个位的定义。该节中的寄存器按功能块分组。

### 6.5.1 OTP 影子寄存器汇总

寄存器名称	地址十六进制值	RW 类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
<i>DIR0_ADDR_OTP</i>	0	NVM	硬件复位默认值 = 0x00 出厂配置默认值 = 0x01	SPARE[1:0]		ADDRESS[5:0]					
<i>DIR1_ADDR_OTP</i>	1	NVM	硬件复位默认值 = 0x00 出厂配置默认值 = 0x01	SPARE[1:0]		ADDRESS[5:0]					
<i>DEV_CONF</i>	2	NVM	0x54	被保留	NO_ADJ_CB	MULTI_DROP_EN	FCOMM_EN	TWO_STOP_EN	NFAULT_EN	FTONE_EN	HB_EN
<i>ACTIVE_CELL</i>	3	NVM	硬件复位默认值 = 0x00 出厂配置默认值 = 0x0A	SPARE[3:0]				NUM_CELL[3:0]			
<i>OTP_SPARE</i> 15	4	NVM	0x00	SPARE[7:0]							

**BQ79616**

ZHCSR7A - JUNE 2023 - REVISED JUNE 2026

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
BBVC_POS N1	5	NVM	0x00	CELL16	CELL15	CELL14	CELL13	CELL12	CELL11	CELL10	CELL9
BBVC_POS N2	6	NVM	0x00	CELL8	CELL7	CELL6	CELL5	CELL4	CELL3	CELL2	CELL1
ADC_CONF 1	7	NVM	0x00	AUX_SETTLE[1:0]		LPF_BB[2:0]			LPF_VCELL[2:0]		
ADC_CONF 2	8	NVM	0x00	SPARE[1:0]			ADC_DLY[5:0]				
OV_THRES H	9	NVM	0x3F	SPARE	SPARE	OV_THR[5:0]					
UV_THRES H	A	NVM	0x00	SPARE	SPARE	UV_THR[5:0]					
OTUT_THR ESH	B	NVM	0xE0	UT_THR[2:0]			OT_THR[4:0]				
UV_DISABLE1	C	NVM	0x00	CELL16	CELL15	CELL14	CELL13	CELL12	CELL11	CELL10	CELL9
UV_DISABLE2	D	NVM	0x00	CELL8	CELL7	CELL6	CELL5	CELL4	CELL3	CELL2	CELL1
GPIO_CONF 1	E	NVM	0x00	FAULT_IN_EN	SPI_EN	GPIO2[2:0]			GPIO1[2:0]		
GPIO_CONF 2	F	NVM	0x00	SPARE	SPARE	GPIO4[2:0]			GPIO3[2:0]		
GPIO_CONF 3	10	NVM	0x00	SPARE[1:0]		GPIO6[2:0]			GPIO5[2:0]		
GPIO_CONF 4	11	NVM	0x00	SPARE[1:0]		GPIO8[2:0]			GPIO7[2:0]		
OTP_SPARE 14	12	NVM	0x00	SPARE[7:0]							
OTP_SPARE 13	13	NVM	0x00	SPARE[7:0]							
OTP_SPARE 12	14	NVM	0x00	SPARE[7:0]							
OTP_SPARE 11	15	NVM	0x00	SPARE[7:0]							
FAULT_MSK 1	16	NVM	0x00	MSK_PROT	MSK_UT	MSK_OT	MSK_UV	MSK_OV	MSK_COMP	MSK_SYS	MSK_PWR
FAULT_MSK 2	17	NVM	0x00	SPARE[1]	MSK_OTP_CRC	MSK_OTP_DATA	MSK_COMM3_FCOMM	MSK_COMM3_FTONE	MSK_COMM3_HB	MSK_COMM2	MSK_COMM1
PWR_TRANSIT_CONF	18	NVM	硬件复位默认值 = 0x18 出厂配置默认值 = 0x10	SPARE[2:0]			TWARN_THR[1:0]		SLP_TIME[2:0]		
COMM_TIME_OUT_CONF	19	NVM	0x00	SPARE	CTS_TIME[2:0]			CTL_ACT	CTL_TIME[2:0]		
TX_HOLD_OFF	1A	NVM	0x00	DLY[7:0]							
MAIN_ADC_CAL1	1B	NVM	0x00	GAINL[7:0]							

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
MAIN_ADC_CAL2	1C	NVM	0x00	GAINH	OFFSET[6:0]						
AUX_ADC_CAL1	1D	NVM	0x00	GAINL[7:0]							
AUX_ADC_CAL2	1E	NVM	0x00	GAINH	OFFSET[6:0]						
OTP_RSVD 1F	1F	NVM	0x00	内部使用。请勿对该地址进行写入							
OTP_RSVD 20	20	NVM	0x00	内部使用。请勿对该地址进行写入							
CUST_MISC 1 至 CUST_MISC 8	21	NVM	0x00	DATA[7:0]							
	22	NVM	0x00	DATA[7:0]							
	23	NVM	0x00	DATA[7:0]							
	24	NVM	0x00	DATA[7:0]							
	25	NVM	0x00	DATA[7:0]							
	26	NVM	0x00	DATA[7:0]							
	27	NVM	0x00	DATA[7:0]							
	28	NVM	0x00	DATA[7:0]							
STACK_RESPONSE	29	NVM	0x00	SPARE[1:0]	DELAY[5:0]						
BBP_LOC	2A	NVM	0x00	SPARE[2:0]			LOC[4:0]				
OTP_RSVD 2B	2B	NVM	0x00	内部使用。请勿对该地址进行写入							
OTP_SPARE 10	2C	NVM	0x00	SPARE[7:0]							
OTP_SPARE 9	2D	NVM	0x00	SPARE[7:0]							
OTP_SPARE 8	2E	NVM	0x00	SPARE[7:0]							
OTP_SPARE 7	2F	NVM	0x00	SPARE[7:0]							
OTP_SPARE 6	30	NVM	0x00	SPARE[7:0]							
OTP_SPARE 5	31	NVM	0x00	SPARE[7:0]							
OTP_SPARE 4	32	NVM	0x00	SPARE[7:0]							
OTP_SPARE 3	33	NVM	0x00	SPARE[7:0]							
OTP_SPARE 2	34	NVM	0x00	SPARE[7:0]							
OTP_SPARE 1	35	NVM	0x00	SPARE[7:0]							
CUST_CRC_HI	36	NVM	硬件复位默认值 = 0x57 出厂配置默认值 = 0x31	CRC[7:0]							

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CUST_CRC_LO	37	NVM	硬件复位默认值 = 0x89 出厂配置默认值 = 0xF3	CRC[7:0]							

6.5.2 读取/写入寄存器汇总

寄存器名称	地址十六进制值	RW类型	复位值	数据								
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	
OTP_PROG_UNL OOCK1A 至 OTP_PROG_UNL OOCK1D	300	RW	0x00	CODE[7:0]								
	301	RW	0x00	CODE[7:0]								
	302	RW	0x00	CODE[7:0]								
	303	RW	0x00	CODE[7:0]								
DIR0_ADDR	306	RW	0x00	被保留			ADDRESS[5:0]					
DIR1_ADDR	307	RW	0x00	被保留			ADDRESS[5:0]					
COMM_CTRL	308	RW	0x00	被保留							STACK_DEV	TOP_STACK
CONTROL1	309	RW	0x00	DIR_SEL	SEND_SHUT_DOWN	SEND_WAKE	SEND_SLPTO_ACT	GOTO_SHUT_DOWN	GOTO_SLEEP	SOFT_RESET	ADDR_WR	
CONTROL2	30A	RW	0x00	被保留							SEND_HW_RESET	TSREF_EN
OTP_PROG_CTRL	30B	RW	0x00	被保留							PAGE_SEL	PROG_GO
ADC_CTRL1	30D	RW	0x00	被保留	RSVD		LPF_BB_EN	LPF_VCELL_EN	MAIN_GO	MAIN_MODE[1:0]		
ADC_CTRL2	30E	RW	0x00	被保留		AUX_CELL_ALIGN	AUX_CELL_SEL[4:0]					
ADC_CTRL3	30F	RW	0x00	被保留	AUX_GPIO_SEL[3:0]				AUX_GO	AUX_MODE[1:0]		
REG_INT_RSVD	310	RW	0x00	内部使用。请勿对该地址进行写入								
CB_CELL16_CTRL L 至 CB_CELL1_CTRL	318	RW	0x00	被保留			TIME[4:0]					
	319	RW	0x00	被保留			TIME[4:0]					
	31A	RW	0x00	被保留			TIME[4:0]					
	31B	RW	0x00	被保留			TIME[4:0]					
	31C	RW	0x00	被保留			TIME[4:0]					
	31D	RW	0x00	被保留			TIME[4:0] :					
	31E	RW	0x00	被保留			TIME[4:0]					
	31F	RW	0x00	被保留			TIME[4:0]					
	320	RW	0x00	被保留			TIME[4:0]					
	321	RW	0x00	被保留			TIME[4:0]					
	322	RW	0x00	被保留			TIME[4:0]					
	323	RW	0x00	被保留			TIME[4:0]					
	324	RW	0x00	被保留			TIME[4:0]					
	325	RW	0x00	被保留			TIME[4:0]					
	326	RW	0x00	被保留			TIME[4:0]					
327	RW	0x00	被保留			TIME[4:0]						
VMB_DONE_THRESH	328	RW	0x3F	被保留			MB_THR[5:0]					
MB_TIMER_CTRL	329	RW	0x00	被保留			TIME[4:0]					
VCB_DONE_THRESH	32A	RW	0x00	被保留			CB_THR[5:0]					
OTCB_THRESH	32B	RW	0x0F	被保留	COOLOFF[2:0]				OTCB_THR[3:0]			

**BQ79616**

ZHCSR7A - JUNE 2023 - REVISED JUNE 2026

寄存器名称	地址十六进制值	RW类型	复位值	数据								
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	
OVUV_CTRL	32C	RW	0x00	VCB_DONE_THR_LOCK	OVUV_LOCK[3:0]				OVUV_GO	OVUV_MODE[1:0]		
OTUT_CTRL	32D	RW	0x00	被保留	OTCB_THR_LOCK	OTUT_LOCK[2:0]		OTUT_GO	OTUT_MODE[1:0]			
BAL_CTRL1	32E	RW	0x00	被保留				DUTY[2:0]				
BAL_CTRL2	32F	RW	0x00	被保留	CB_PAUSE	FLTSTOP_EN	OTCB_EN	BAL_ACT[1:0]		BAL_GO	AUTO_BAL	
BAL_CTRL3	330	RW	0x00	被保留			BAL_TIME_SEL[3:0]				BAL_TIME_GO	
FAULT_RST1	331	RW	0x00	RST_PROT	RST_UT	RST_OT	RST_UV	RST_OV	RST_COMP	RST_SYS	RST_PWR	
FAULT_RST2	332	RW	0x00	被保留	RST_OTP_CRC	RST_OTP_DATA	RST_COMM3_FCOMM	RST_COMM3_FTONE	RST_COMM3_HB	RST_COMM2	RST_COMM1	
DIAG_OTP_CTRL	335	RW	0x00	被保留			FLIP_FACT_CRC	MARGIN_MODE[2:0]			MARGIN_GO	
DIAG_COMM_CTRL	336	RW	0x00	被保留						SPI_LOOP_BACK	FLIP_TR_CRC	
DIAG_PWR_CTRL	337	RW	0x00	被保留						BIST_NO_RST	PWR_BIST_GO	
DIAG_CBFET_CTRL1	338	RW	0x00	CBFET16	CBFET15	CBFET14	CBFET13	CBFET12	CBFET11	CBFET10	CBFET9	
DIAG_CBFET_CTRL2	339	RW	0x00	CBFET8	CBFET7	CBFET6	CBFET5	CBFET4	CBFET3	CBFET2	CBFET1	
DIAG_COMP_CTRL1	33A	RW	0x00	VCCB_THR[4:0]					BB_THR[2:0]			
DIAG_COMP_CTRL2	33B	RW	0x00	被保留	GPIO_THR[2:0]			OW_THR[3:0]				
DIAG_COMP_CTRL3	33C	RW	0x00	被保留	CBFET_CTRL_GO	OW_SNK[1:0]		COMP_ADC_SEL[2:0]			COMP_ADC_GO	
DIAG_COMP_CTRL4	33D	RW	0x00	被保留						COMP_FAULT_INJ	LPF_FAULT_INJ	
DIAG_PROT_CTRL	33E	RW	0x00	被保留							PROT_BIST_NO_RST	
OTP_ECC_DATA1 N1 至 OTP_ECC_DATA1 N9	343	RW	0x00	DATA[7:0]								
	344	RW	0x00	DATA[7:0]								
	345	RW	0x00	DATA[7:0]								
	346	RW	0x00	DATA[7:0]								
	347	RW	0x00	DATA[7:0]								
	348	RW	0x00	DATA[7:0]								
	349	RW	0x00	DATA[7:0]								
	34A	RW	0x00	DATA[7:0]								
34B	RW	0x00	DATA[7:0]									

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
OTP_ECC_TEST	34C	RW	0x00	被保留				DED_SEC	MANUAL_AUTO	ENC_DEC	ENABLE
SPI_CONF	34D	RW	0x00	被保留	CPOL	CPHA	NUMBIT[4:0]				
SPI_TX3、 SPI_TX2 和 SPI_TX1	34E	RW	0x00	DATA[7:0]							
	34F	RW	0x00	DATA[7:0]							
	350	RW	0x00	DATA[7:0]							
SPI_EXE	351	RW	0x02	被保留					SS_CTRL	SPI_GO	
OTP_PROG_UNL OCK2A 至 OTP_PROG_UNL OCK2D	352	RW	0x00	CODE[7:0]							
	353	RW	0x00	CODE[7:0]							
	354	RW	0x00	CODE[7:0]							
	355	RW	0x00	CODE[7:0]							
DEBUG_CTRL_UNLOCK	700	RW	0x00	CODE[7:0]							
DEBUG_COMM_CTRL1	701	RW	0x04	被保留			UART_BAUD	UART_MIRROR_EN	UART_TX_EN	USER_UART_EN	USER_DAISSY_EN
DEBUG_COMM_CTRL2	702	RW	0x0F	被保留			COML_TX_EN	COML_RX_EN	COMH_TX_EN	COMH_RX_EN	

### 6.5.3 只读寄存器汇总

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
PARTID	500	R	0x00	REV[7:0]							
DEV_REVID	E00	R	0x00	DEV_REVID[7:0]							
DIE_ID1 至 DIE_ID9	501	R	0x00	ID[7:0]							
	502	R	0x00	ID[7:0]							
	503	R	0x00	ID[7:0]							
	504	R	0x00	ID[7:0]							
	505	R	0x00	ID[7:0]							
	506	R	0x00	ID[7:0]							
	507	R	0x00	ID[7:0]							
	508	R	0x00	ID[7:0]							
	509	R	0x00	ID[7:0]							
CUST_CRC_RSLT_HI	50C	R	0x31	CRC[7:0]							
CUST_CRC_RSLT_LO	50D	R	0xF3	CRC[7:0]							

寄存器名称	地址十六进制值	RW类型	复位值	数据								
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	
OTP_ECC_DATA OUT1 至 OTP_ECC_DATA OUT9	510	R	0x00	DATA[7:0]								
	511	R	0x00	DATA[7:0]								
	512	R	0x00	DATA[7:0]								
	513	R	0x00	DATA[7:0]								
	514	R	0x00	DATA[7:0]								
	515	R	0x00	DATA[7:0]								
	516	R	0x00	DATA[7:0]								
	517	R	0x00	DATA[7:0]								
518	R	0x00	DATA[7:0]									
OTP_PROG_STAT	519	R	0x00	UNLOCK	OTERR	UVERR	OVERR	SUVERR	SOVERR	PROGERR	DONE	
OTP_CUST1_STAT	51A	R	0x00	LOADED	LOADWRN	LOADERR	FMTERR	PROGOK	UVOK	OVOK	TRY	
OTP_CUST2_STAT	51B	R	0x00	LOADED	LOADWRN	LOADERR	FMTERR	PROGOK	UVOK	OVOK	TRY	
SPI_RX3、SPI_RX2 和 SPI_RX1	520	R	0x00	DATA[7:0]								
	521	R	0x00	DATA[7:0]								
	522	R	0x00	DATA[7:0]								
DIAG_STAT	526	R	0x00	被保留			DRDY_OTUT	DRDY_OVUV	DRDY_BIST_OTUT	DRDY_BIST_OVUV	DRDY_BIST_PWR	
ADC_STAT1	527	R	0x00	被保留			RSVD	DRDY_AUX_GPIO	DRDY_AUX_CELL	DRDY_AUX_MISC	DRDY_MAIN_ADC	
ADC_STAT2	528	R	0x00	被保留		DRDY_LPF	DRDY_GPIO	DRDY_VCOW	DRDY_CBOW	DRDY_CBFET	DRDY_VCCB	
GPIO_STAT	52A	R	0x00	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	
BAL_STAT	52B	R	0x00	INVALID_CBCONF	OT_PAUSE_DET	CB_INPAUSE	MB_RUN	CB_RUN	ABORTFLT	MB_DONE	CB_DONE	
DEV_STAT	52C	R	0x00	被保留	FACT_CRC_DONE	CUST_CRC_DONE	OTUT_RUN	OVUV_RUN	RSVD	AUX_RUN	MAIN_RUN	
FAULT_SUMMARY	52D	R	0x00	FAULT_PROT	FAULT_COMP_ADC	FAULT_OTP	FAULT_COMM	FAULT_OTUT	FAULT_OVUV	FAULT_SYS	FAULT_PWR	
FAULT_COMM1	530	R	0x00	被保留			UART_TR	UART_RR	UART_RC	COMM_CLR_DET	STOP_DET	
FAULT_COMM2	531	R	0x00	COML_TR	COML_RR	COML_RC	COML_BIT	COMH_TR	COMH_RR	COMH_RC	COMH_BIT	
FAULT_COMM3	532	R	0x00	被保留				FComm_DET	FTONE_DET	HB_FAIL	HB_FAST	
FAULT_OTP	535	R	0x00	被保留	DED_DET	SEC_DET	CUST_CRC	FACT_CRC	CUSTLD_ERR	FACTLD_ERR	GBLOV_ERR	
FAULT_SYS	536	R	0x00	LFO	被保留	GPIO	DRST	CTL	CTS	TSHUT	TWARN	
FAULT_PROT1	53A	R	0x00	被保留							TPARITY_FAIL	VPARITY_FAIL
FAULT_PROT2	53B	R	0x00	被保留	BIST_ABORT	TPATH_FAIL	VPATH_FAIL	UTCOMP_FAIL	OTCOMP_FAIL	OVCOMP_FAIL	UVCOMP_FAIL	

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
FAULT_OV1	53C	R	0x00	OV16_DET	OV15_DET	OV14_DET	OV13_DET	OV12_DET	OV11_DET	OV10_DET	OV9_DET
FAULT_OV2	53D	R	0x00	OV8_DET	OV7_DET	OV6_DET	OV5_DET	OV4_DET	OV3_DET	OV2_DET	OV1_DET
FAULT_UV1	53E	R	0x00	UV16_DET	UV15_DET	UV14_DET	UV13_DET	UV12_DET	UV11_DET	UV10_DET	UV9_DET
FAULT_UV2	53F	R	0x00	UV8_DET	UV7_DET	UV6_DET	UV5_DET	UV4_DET	UV3_DET	UV2_DET	UV1_DET
FAULT_OT	540	R	0x00	OT8_DET	OT7_DET	OT6_DET	OT5_DET	OT4_DET	OT3_DET	OT2_DET	OT1_DET
FAULT_UT	541	R	0x00	UT8_DET	UT7_DET	UT6_DET	UT5_DET	UT4_DET	UT3_DET	UT2_DET	UT1_DET
FAULT_COMP_GPIO	543	R	0x00	GPIO8_FAIL	GPIO7_FAIL	GPIO6_FAIL	GPIO5_FAIL	GPIO4_FAIL	GPIO3_FAIL	GPIO2_FAIL	GPIO1_FAIL
FAULT_COMP_VCCB1	545	R	0x00	CELL16_FAIL	CELL15_FAIL	CELL14_FAIL	CELL13_FAIL	CELL12_FAIL	CELL11_FAIL	CELL10_FAIL	CELL9_FAIL
FAULT_COMP_VCCB2	546	R	0x00	CELL8_FAIL	CELL7_FAIL	CELL6_FAIL	CELL5_FAIL	CELL4_FAIL	CELL3_FAIL	CELL2_FAIL	CELL1_FAIL
FAULT_COMP_VCOW1	548	R	0x00	VCOW16_FAIL	VCOW15_FAIL	VCOW14_FAIL	VCOW13_FAIL	VCOW12_FAIL	VCOW11_FAIL	VCOW10_FAIL	VCOW9_FAIL
FAULT_COMP_VCOW2	549	R	0x00	VCOW8_FAIL	VCOW7_FAIL	VCOW6_FAIL	VCOW5_FAIL	VCOW4_FAIL	VCOW3_FAIL	VCOW2_FAIL	VCOW1_FAIL
FAULT_COMP_CBOW1	54B	R	0x00	CBOW16_FAIL	CBOW15_FAIL	CBOW14_FAIL	CBOW13_FAIL	CBOW12_FAIL	CBOW11_FAIL	CBOW10_FAIL	CBOW9_FAIL
FAULT_COMP_CBOW2	54C	R	0x00	CBOW8_FAIL	CBOW7_FAIL	CBOW6_FAIL	CBOW5_FAIL	CBOW4_FAIL	CBOW3_FAIL	CBOW2_FAIL	CBOW1_FAIL
FAULT_COMP_CBFET1	54E	R	0x00	CBFET16_FAIL	CBFET15_FAIL	CBFET14_FAIL	CBFET13_FAIL	CBFET12_FAIL	CBFET11_FAIL	CBFET10_FAIL	CBFET9_FAIL
FAULT_COMP_CBFET2	54F	R	0x00	CBFET8_FAIL	CBFET7_FAIL	CBFET6_FAIL	CBFET5_FAIL	CBFET4_FAIL	CBFET3_FAIL	CBFET2_FAIL	CBFET1_FAIL
FAULT_COMP_MISC	550	R	0x00	被保留						COMP_ADC_ABORT	LPF_FAIL
FAULT_PWR1	552	R	0x00	CVSS_OPEN	DVSS_OPEN	REFHM_OPEN	CVDD_UV	CVDD_OV	DVDD_OV	AVDD_OSC	AVDD_OV
FAULT_PWR2	553	R	0x00	被保留	PWRBIST_FAIL	被保留	REFH_OSC	NEG5V_UV	TSREF_OSC	TSREF_UV	TSREF_OV
FAULT_PWR3	554	R	RSVD	RSVD					RSVD	RSVD	AVDDUV_DRST
CB_COMPLETE1	556	R	0x00	CELL16_DONE	CELL15_DONE	CELL14_DONE	CELL13_DONE	CELL12_DONE	CELL11_DONE	CELL10_DONE	CELL9_DONE
CB_COMPLETE2	557	R	0x00	CELL8_DONE	CELL7_DONE	CELL6_DONE	CELL5_DONE	CELL4_DONE	CELL3_DONE	CELL2_DONE	CELL1_DONE
BAL_TIME	558	R	0x00	TIME_UNIT	TIME[6:0]						
VCELL16_HI/LO	568	R	0x80	RESULT[7:0]							
	569	R	0x00	RESULT[7:0]							
VCELL15_HI/LO	56A	R	0x80	RESULT[7:0]							
	56B	R	0x00	RESULT[7:0]							
VCELL14_HI/LO	56C	R	0x80	RESULT[7:0]							
	56D	R	0x00	RESULT[7:0]							
VCELL13_HI/LO	56E	R	0x80	RESULT[7:0]							
	56F	R	0x00	RESULT[7:0]							

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
VCELL12_HI/LO	570	R	0x80	RESULT[7:0]							
	571	R	0x00	RESULT[7:0]							
VCELL11_HI/LO	572	R	0x80	RESULT[7:0]							
	573	R	0x00	RESULT[7:0]							
VCELL10_HI/LO	574	R	0x80	RESULT[7:0]							
	575	R	0x00	RESULT[7:0]							
VCELL9_HI/LO	576	R	0x80	RESULT[7:0]							
	577	R	0x00	RESULT[7:0]							
VCELL8_HI/LO	578	R	0x80	RESULT[7:0]							
	579	R	0x00	RESULT[7:0]							
VCELL7_HI/LO	57A	R	0x80	RESULT[7:0]							
	57B	R	0x00	RESULT[7:0]							
VCELL6_HI/LO	57C	R	0x80	RESULT[7:0]							
	57D	R	0x00	RESULT[7:0]							
VCELL5_HI/LO	57E	R	0x80	RESULT[7:0]							
	57F	R	0x00	RESULT[7:0]							
VCELL4_HI/LO	580	R	0x80	RESULT[7:0]							
	581	R	0x00	RESULT[7:0]							
VCELL3_HI/LO	582	R	0x80	RESULT[7:0]							
	583	R	0x00	RESULT[7:0]							
VCELL2_HI/LO	584	R	0x80	RESULT[7:0]							
	585	R	0x00	RESULT[7:0]							
VCELL1_HI/LO	586	R	0x80	RESULT[7:0]							
	587	R	0x00	RESULT[7:0]							
BUSBAR_HI/LO	588	R	0x80	RESULT[7:0]							
	589	R	0x00	RESULT[7:0]							
TSREF_HI/LO	58C	R	0x80	RESULT[7:0]							
	58D	R	0x00	RESULT[7:0]							
GPIO1_HI/LO	58E	R	0x80	RESULT[7:0]							
	58F	R	0x00	RESULT[7:0]							
GPIO2_HI/LO	590	R	0x80	RESULT[7:0]							
	591	R	0x00	RESULT[7:0]							
GPIO3_HI/LO	592	R	0x80	RESULT[7:0]							
	593	R	0x00	RESULT[7:0]							
GPIO4_HI/LO	594	R	0x80	RESULT[7:0]							
	595	R	0x00	RESULT[7:0]							
GPIO5_HI/LO	596	R	0x80	RESULT[7:0]							
	597	R	0x00	RESULT[7:0]							
GPIO6_HI/LO	598	R	0x80	RESULT[7:0]							
	599	R	0x00	RESULT[7:0]							
GPIO7_HI/LO	59A	R	0x80	RESULT[7:0]							
	59B	R	0x00	RESULT[7:0]							
GPIO8_HI/LO	59C	R	0x80	RESULT[7:0]							
	59D	R	0x00	RESULT[7:0]							

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DIETEMP1_HI/LO	5AE	R	0x80	RESULT[7:0]							
	5AF	R	0x00	RESULT[7:0]							
DIETEMP2_HI/LO	5B0	R	0x80	RESULT[7:0]							
	5B1	R	0x00	RESULT[7:0]							
AUX_CELL_HI/LO	5B2	R	0x80	RESULT[7:0]							
	5B3	R	0x00	RESULT[7:0]							
AUX_GPIO_HI/LO	5B4	R	0x80	RESULT[7:0]							
	5B5	R	0x00	RESULT[7:0]							
AUX_BAT_HI/LO	5B6	R	0x80	RESULT[7:0]							
	5B7	R	0x00	RESULT[7:0]							
AUX_REFL_HI/LO	5B8	R	0x80	RESULT[7:0]							
	5B9	R	0x00	RESULT[7:0]							
AUX_VBG2_HI/LO	5BA	R	0x80	RESULT[7:0]							
	5BB	R	0x00	RESULT[7:0]							
AUX_AVAO_REF_HI/LO	5BE	R	0x80	RESULT[7:0]							
	5BF	R	0x00	RESULT[7:0]							
AUX_AVDD_REF_HI/LO	5C0	R	0x80	RESULT[7:0]							
	5C1	R	0x00	RESULT[7:0]							
AUX_OV_DAC_HI/LO	5C2	R	0x80	RESULT[7:0]							
	5C3	R	0x00	RESULT[7:0]							
AUX_UV_DAC_HI/LO	5C4	R	0x80	RESULT[7:0]							
	5C5	R	0x00	RESULT[7:0]							
AUX_OT_OTCB_DAC_HI/LO	5C6	R	0x80	RESULT[7:0]							
	5C7	R	0x00	RESULT[7:0]							
AUX_UT_DAC_HI/LO	5C8	R	0x80	RESULT[7:0]							
	5C9	R	0x00	RESULT[7:0]							
AUX_VCBDONE_DAC_HI/LO	5CA	R	0x80	RESULT[7:0]							
	5CB	R	0x00	RESULT[7:0]							
AUX_VCM_HI/LO	5CC	R	0x80	RESULT[7:0]							
	5CD	R	0x00	RESULT[7:0]							
REFOVDAC_HI/LO	5D0	R	0x00	RESULT[7:0]							
	5D1	R	0x00	RESULT[7:0]							
DIAG_MAIN_HI/LO	5D2	R	0x00	RESULT[7:0]							
	5D3	R	0x00	RESULT[7:0]							
DIAG_AUX_HI/LO	5D4	R	0x00	RESULT[7:0]							
	5D5	R	0x00	RESULT[7:0]							
DEBUG_COMM_STAT	780	R	针对堆栈的基址 0x3F，为 0x33	被保留	HW_UART_DRV	HW_DAISSY_DRV	COML_TX_ON	COML_RX_ON	COMH_TX_ON	COMH_RX_ON	
DEBUG_UART_RC	781	R	0x00	被保留	RC_IERR	RC_TXDIS	RC_SOF	RC_BYTE_ERR	RC_UNEXP	RC_CRC	

**BQ79616**

ZHCSR7A - JUNE 2023 - REVISED JUNE 2026

寄存器名称	地址十六进制值	RW类型	复位值	数据							
				位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DEBUG_UART_RR_TR	782	R	0x00	被保留			TR_SOF	TR_WAIT	RR_SOF	RR_BYTE_ERR	RR_CRC
T	783	R	0x00	被保留			PERR	BERR_TAG	Sync2	Sync1	位
DEBUG_COMH_RC	784	R	0x00	被保留		RC_IERR	RC_TXDIS	RC_SOF	RC_BYTE_ERR	RC_UNEXP	RC_CRC
DEBUG_COMH_RR_TR	785	R	0x00	被保留		TR_WAIT	RR_TXDIS	RR_SOF	RR_BYTE_ERR	RR_UNEXP	RR_CRC
DEBUG_COML_BIT	786	R	0x00	被保留			PERR	BERR_TAG	Sync2	Sync1	位
DEBUG_COML_RC	787	R	0x00	被保留		RC_IERR	RC_TXDIS	RC_SOF	RC_BYTE_ERR	RC_UNEXP	RC_CRC
DEBUG_COML_RR_TR	788	R	0x00	被保留		TR_WAIT	RR_TXDIS	RR_SOF	RR_BYTE_ERR	RR_UNEXP	RR_CRC
DEBUG_UART_DISCARD	789	R	0x00	COUNT[7:0]							
DEBUG_COMH_DISCARD	78A	R	0x00	COUNT[7:0]							
DEBUG_COML_DISCARD	78B	R	0x00	COUNT[7:0]							
DEBUG_UART_VALID_HI/LO	78C	R	0x00	COUNT[7:0]							
	78D	R	0x00	COUNT[7:0]							
DEBUG_COMH_VALID_HI/LO	78E	R	0x00	COUNT[7:0]							
	78F	R	0x00	COUNT[7:0]							
DEBUG_COML_VALID_HI/LO	790	R	0x00	COUNT[7:0]							
	791	R	0x00	COUNT[7:0]							
DEBUG_OTP_SECC_BLK	7A0	R	0x00	BLOCK[7:0]							
DEBUG_OTP_DECD_BLK	7A1	R	0x00	BLOCK[7:0]							

## 6.5.4 寄存器字段说明

### 6.5.4.1 器件寻址设置

#### 6.5.4.1.1 DIR0\_ADDR\_OTP

地址	0x0000							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[1:0]			ADDRESS[5:0]				
复位	0	0	0	0	0	0	0	0
SPARE[1:0] = 备用								
ADDRESS[5:0] = 该寄存器显示当 $[DIR\_SEL] = 0$ 时使用并在 OTP 中编程的默认器件地址。对该寄存器进行写入不会更改正在使用的器件地址。 该寄存器用于系统将器件地址编程到 OTP 中，OTP 将在 POR 时加载到 DIR0_ADDR 寄存器。对于编程，请遵循 OTP 编程过程。								

#### 6.5.4.1.2 DIR1\_ADDR\_OTP

地址	0x0001							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[1:0]			ADDRESS[5:0]				
复位	0	0	0	0	0	0	0	0
SPARE[1:0] = 备用								
ADDRESS[5:0] = 该寄存器显示当 $[DIR\_SEL] = 1$ 时使用并在 OTP 中编程的默认器件地址。对该寄存器进行写入不会更改正在使用的器件地址。 该寄存器用于系统将器件地址编程到 OTP 中，OTP 将在 POR 时加载到 DIR1_ADDR 寄存器。对于编程，请遵循 OTP 编程过程。								

#### 6.5.4.1.3 CUST\_MISC1 至 CUST\_MISC8

地址	0x0021 至 0x0028							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	DATA[7:0]							
复位	0	0	0	0	0	0	0	0
DATA[7:0] = 客户暂存区								

#### 6.5.4.1.4 DIR0\_ADDR

地址	0x0306							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD			ADDRESS[5:0]				
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
ADDRESS[5:0] = 当 $[DIR\_SEL] = 0$ 时，始终显示器件使用的当前器件地址。在 POR 时，从 OTP 中的器件地址值加载此寄存器（加载到 DIR0_ADDR_OTP 寄存器中的相同 OTP 器件地址）。主机可以通过将不同的器件地址写入此寄存器来对器件重新寻址，并且该器件将立即获取新地址。 注意：需要 $CONTROL1[ADDR\_WR] = 1$ 才能写入此寄存器。详细信息请参阅 节 6.5.4.3.11。								

#### 6.5.4.1.5 DIR1\_ADDR

地址	0x0307							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留			ADDRESS[5:0]				
复位	0	0	0	0	0	0	0	0

RSVD = 被保留
ADDRESS[5:0] = 当 $[DIR\_SEL] = 1$ 时, 始终显示器件使用的当前器件地址。在 POR 时, 从 OTP 中的器件地址值加载此寄存器 (加载到 $DIR1\_ADDR\_OTP$ 寄存器中的相同 OTP 器件地址)。主机可以通过将不同的器件地址写入此寄存器来对器件重新寻址, 并且该器件将立即获取新地址。 注意: 需要 $CONTROL1[ADDR\_WR] = 1$ 才能写入此寄存器。详细信息请参阅 节 6.5.4.3.11。

### 6.5.4.2 器件 ID 和暂存区

#### 6.5.4.2.1 PARTID

地址	0x0500							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	PARTID[7:0]							
复位	0	0	0	0	0	0	0	0
PARTID[7:0] 器件标识: = 0x21 = BQ79616 所有其他代码 = 保留								

#### 6.5.4.2.2 DEV\_REVID

地址	0xE00							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
复位	0	0	0	0	0	0	0	0
值 0x00 指示器件处于正常工作模式。如果故障激活出厂测试模式检测, 则该值将不为零。有关 SM426 的详细信息, 请参阅安全手册: 出厂测试模式检测。								

#### 6.5.4.2.3 DIE\_ID1 至 DIE\_ID9

地址	0x0501							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	ID[7:0]							
复位	0	0	0	0	0	0	0	0
ID[7:0] = 器件修订版本 0x10 = 修订版本 A0 0x11 = 修订版本 A1 0x20 = 修订版本 B0 0x21 = 修订版本 B1 0x22 = 修订版本 B2 所有其他代码 = 保留								

地址	0x0502 至 0x0509							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	ID[7:0]							
复位	0	0	0	0	0	0	0	0
ID[7:0] = 供 TI 工厂使用的芯片 ID								

### 6.5.4.3 通用配置和控制

#### 6.5.4.3.1 DEV\_CONF

地址	0x0002							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD	NO_ADJ_CB	MULTIDROP_EN	FCOMM_EN	TWO_STOP_EN	NFAULT_EN	FTONE_EN	HB_EN

复位	0	1	0	1	0	1	0	0
RSVD = 被保留								
NO_ADJ_CB = 指示器件不允许通过手动 CB 控制开启相邻的 CB FET。如果 MCU 已启用相邻的 CB FET，那么即使主机发送 $[BAL\_GO] = 1$ ，器件也不会启动 CB。 0 = 器件将允许启用两个相邻的 CB FET。 1 = 器件不允许启用相邻的 CB FET。								
MULTIDROP_EN = 定义器件是用于多点配置还是用于菊花链配置。将根据配置来启用或禁用 COML 和 COMH 的 TX 和 RX。 0 = 基底器件的菊花链 1 = 多点								
FCOMM_EN = 通过 ACTIVE 模式下的通信启用故障状态检测。 0 = 禁用 1 = 启用								
TWO_STOP_EN = 在主机和器件中出现严重振荡器错误的情况下为 UART 启用两个停止位。 0 = 一个 STOP 位 1 = 两个 STOP 位								
NFAULT_EN = 启用 NFAULT 功能。 0 = 始终上拉 NFAULT 1 = 下拉 NFAULT 以指示检测到未屏蔽的故障。 注意：该位设置不影响 FAULT_SUMMARY 寄存器。								
FTONE_EN = 当器件处于 SLEEP 模式时启用故障音调发送器。 0 = 禁用 1 = 启用								
HB_EN = 当器件处于 SLEEP 模式时启用 HEARTBEAT 发送器。 0 = 禁用 1 = 启用								

#### 6.5.4.3.2 ACTIVE\_CELL

地址	0x0003							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[3:0]				NUM_CELL[3:0]			
复位	0	0	0	0	0	0	0	0
出厂 OTP 复位	0	0	0	0	1	0	1	0
SPARE[3:0] = 备用								
NUM_CELL[3:0] = 配置串联电芯数。 0x0 = 6S 0x1 = 7S 0x2 = 8S : 0xA = 16S 未使用的代码默认为 CHIP_TYPE[MAX_CH1:0] 设置（采用出厂调整）。如果 NUM_CELL 设置的通道数超过器件提供的通道数，则上限为由器件提供的最大通道数。								

#### 6.5.4.3.3 BBVC\_POSN1

地址	0x0005							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL16	CELL15	CELL14	CELL13	CELL12	CELL11	CELL10	CELL9
复位	0	0	0	0	0	0	0	0
CELL9 至 CELL16 = 该寄存器指定与汇流条连接的通道。电芯电压测量诊断比较将以不同方式处理这些通道 0 = 无特殊处理 1 = 特殊处理启用的通道。有关详细信息，请参阅节 6.3.7。								

#### 6.5.4.3.4 BBVC\_POSN2

地址	0x0006							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL8	CELL7	CELL6	CELL5	CELL4	CELL3	CELL2	CELL1
复位	0	0	0	0	0	0	0	0
CELL1 至 CELL8 = 在 <i>ACTIVE_CELL</i> 寄存器指定的活动电芯中，该寄存器指示哪个活动通道被排除在 OV、UV 和 VCB_DONE 监控之外。该寄存器信息还用于电芯电压测量诊断比较。 0 = 不对上述功能进行特殊处理。 1 = 对上述功能进行特殊处理。有关详细信息，请参阅节 6.3.7。								

#### 6.5.4.3.5 PWR\_TRANSIT\_CONF

地址	0x0018							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[2:0]			TWARN_THR[1:0]		SLP_TIME[2:0]		
复位	0	0	0	1	1	0	0	0
出厂配置默认值	0	0	0	1	0	0	0	0
SPARE[2:0] = 备用								
TWARN_THR[1:0] = 设置 TWARN 阈值。 00 = 85°C 01 = 95°C 10 = 105°C (默认值) 11 = 115°C								
SLP_TIME[2:0] = SLEEP 模式下的超时。当器件进入 SLEEP 模式时，该计时器开始计数。当计时器到期时，器件进入 SHUTDOWN 模式。如果器件唤醒至 ACTIVE 模式，则计时器会复位。 000 = 无超时。器件保持在 SLEEP 模式 (复位时的默认设置) 001 = 5s 010 = 10s 011 = 1 分钟 100 = 10 分钟 101 = 30 分钟 110 = 1 小时 111 = 2 小时								

#### 6.5.4.3.6 COMM\_TIMEOUT\_CONF

地址	0x0019							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE	CTS_TIME[2:0]			CTL_ACT	CTL_TIME[2:0]		
复位	0	0	0	0	0	0	0	0
SPARE = 备用								
CTS_TIME[2:0] = 设置较短的通信超时。当该计时器到期时，器件会设置 <i>FAULT_SYS[CTL]</i> 位。这可用作向系统发出的警报，以防止较长的通信超时。 000 = 禁用较短的通信超时 (复位时的默认值) 001 = 100ms 010 = 2s 011 = 10s 100 = 1 分钟 101 = 10 分钟 110 = 30 分钟 111 = 1 小时								
CTL_ACT = 配置当较长的通信超时时器到期时的器件操作。 0 = 设置 <i>FAULT_SYS[CTL]</i> 并将器件置于睡眠模式 1 = 将器件置于关断状态。将不会设置 <i>FAULT_SYS[CTL]</i> 位。(复位时的默认设置)								

CTL\_TIME[2:0] = 设置较长的通信超时。当此计时器到期时，器件将执行由 [CTL\_ACT] 位配置的操作。

- 000 = 禁用较长的通信超时
- 001 = 100ms
- 010 = 2s
- 011 = 10s
- 100 = 1 分钟 (复位时的默认值)
- 101 = 10 分钟
- 110 = 30 分钟
- 111 = 1 小时

#### 6.5.4.3.7 TX\_HOLD\_OFF

地址	0x001A							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	DLY[7:0]							
复位	0	0	0	0	0	0	0	0
DLY[7:0] = 设置在接收到命令帧的 STOP 位之后、发送响应帧的第 1 位之前延迟的位周期数，范围为 0 至 255。								

#### 6.5.4.3.8 STACK\_RESPONSE

地址	0x0029							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[1:0]			DELAY[5:0]				
复位	0	0	0	0	0	0	0	0
DELAY[5:0] 在菊花链数据响应帧中添加额外的字节延迟间隙 = 0x00 = 0μs 0x01 至 0x3F = 0.25μs 至 15.75μs，阶跃为 0.25us								

#### 6.5.4.3.9 BBP\_LOC

地址	0x002A							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[3:0]			LOC[4:0]				
复位	0	0	0	0	0	0	0	0
LOC[4:0] = 指示 BBP 引脚位置并显示 BB 通道与哪个 VC 通道共享。该信息用于校正 BB 通道 ADC 测量期间的共模误差 0x00 = 未使用 BBP/N 0x01 = BBP 至 Cell2 的负极侧，BBN 至 VC1 0x02 = BBP 至 Cell3 的负极侧，BBN 至 VC2 0x03 = BBP 至 Cell4 的负极侧，BBN 至 VC3 : 0x0F = BBP 至 Cell16 的负极侧，BBn 至 VC15 0x10 = BBP 至 VC16，BBN 至 Cell16 的正极侧								

#### 6.5.4.3.10 COMM\_CTRL

地址	0x0308							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留						STACK_DEV	TOP_STACK
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
STACK_DEV = 将器件定义为菊花链配置中的基底器件或堆栈器件。 0 = 基底器件 1 = 堆栈器件								

TOP\_STACK = 将器件定义为堆栈中地址最高的器件。  
0 = 不是 ToS 器件  
1 = 是 ToS 器件

#### 6.5.4.3.11 CONTROL1

地址	0x0309							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	DIR_SEL	SEND_SHUTDOWN	SEND_WAKE	SEND_SLPTOACT	GOTO_SHUTDOWN	GOTO_SLEEP	SOFT_RESET	ADDR_WR
复位	0	0	0	0	0	0	0	0
DIR_SEL = 选择菊花链通信方向。 0 = 在两个器件以菊花链方式连接的情况下，命令帧从下部器件的 COMH 传输到下一个器件的 COML。 1 = 在两个器件以菊花链方式连接的情况下，命令帧从下部器件的 COML 传输到下一个器件的 COMH。								
SEND_SHUTDOWN = 沿堆栈向上，向下一个器件发送 SHUTDOWN 音。接收此位设置的器件不受影响。读取时该位被清除。 0 = 就绪 1 = 沿堆栈向上发送关断音								
SEND_WAKE = 沿堆栈向上，向下一个器件发送唤醒音。读取时该位被清除。 0 = 就绪 1 = 沿堆栈向上，向下一个器件发送唤醒音。								
SEND_SLPTOACT = 沿堆栈向上，发送 SLEEPTOACTIVE 音。读取时该位被清除。 0 = 就绪 1 = 沿堆栈向上，发送 SLEEPTOACTIVE 音								
GOTO_SHUTDOWN = 将器件转换为 SHUTDOWN 模式。读取时该位被清除。 0 = 就绪 1 = 进入 SHUTDOWN 模式								
GOTO_SLEEP = 将器件转换为 SLEEP 模式。读取时该位被清除。 0 = 就绪 1 = 进入 SLEEP 模式								
SOFT_RESET = 将数字复位为 OTP 默认值。读取时该位被清除。设置该位将使器件向上层堆栈器件生成唤醒音。 0 = 就绪 1 = 复位器件								
ADDR_WR = 使器件启动自动寻址。设置该位时，器件不会转发它接收到的第一个转换，从而允许将器件地址写入单个器件。有关详细信息，请参阅节 6.3.6.1.3.2。 0 = 不执行自动寻址。器件正常转发通信事务。 1 = 正在对器件自动寻址；将不会转发它收到的第一个通信事务。								

#### 6.5.4.3.12 CONTROL2

地址	0x030A							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留						SEND_HW_RESET	TSREF_EN
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
SEND_HW_RESET = 沿堆栈向上发送 HW_RESET 音。读取时该位被清除。 0 = 就绪 1 = 沿堆栈向上，向下一个器件发送 HW_RESET 音								
TSREF_EN = 启用 TSREF LDO 输出。用于偏置 NTC 热敏电阻。 0 = 禁用 1 = 启用								

#### 6.5.4.3.13 CUST\_CRC\_HI

地址	0x0036							

NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CRC[7:0]							
复位	0	1	0	1	0	1	1	1
出厂配置 复位	0	0	1	1	0	0	0	1
CRC[7:0] = 客户 OTP 空间的主机计算的 CRC 的高字节。								

#### 6.5.4.3.14 CUST\_CRC\_LO

地址	0x0037							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CRC[7:0]							
复位	1	0	0	0	1	0	0	1
出厂配置 复位	1	1	1	1	0	0	1	1
CRC[7:0] = 客户 OTP 空间的主机计算的 CRC 的低字节。								

#### 6.5.4.3.15 CUST\_CRC\_RSLT\_HI

地址	0x050C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CRC[7:0]							
复位	0	0	1	1	0	0	0	1
CRC[7:0] = 客户 OTP 空间的器件计算的 CRC 的高字节。								

#### 6.5.4.3.16 CUST\_CRC\_RSLT\_LO

地址	0x050D							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CRC[7:0]							
复位	1	1	1	1	0	0	1	1
CRC[7:0] = 客户 OTP 空间的器件计算的 CRC 的低字节。								

### 6.5.4.4 运行状态

#### 6.5.4.4.1 DIAG\_STAT

地址	0x0526							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留			DRDY_OTUT	DRDY_OVUV	DRDY_BIST_OTUT	DRDY_BIST_OVUV	DRDY_BIST_PWR
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
<p>DRDY_OTUT = 指示 OTUT 轮循至少运行过一次。当 [OTUT_GO] = 1 且 [OTUT_MODE1:0] = 01 (启动 OTUT 轮循运行) 时清除此位, 并在至少 1 个周期的轮循完成时设置该位。 0 = OTUT 尚未开始或第一个轮循尚未完成。 1 = 至少完成了 1 个轮循周期。</p>								
<p>DRDY_OVUV = 指示 OVUV 轮循至少运行过一次。当 [OVUV_GO] = 1 且 [OVUV_MODE1:0] = 01 (启动 OVUV 轮循运行) 时清除此位, 并在至少 1 个周期的轮循完成时设置该位。 0 = OVUV 尚未开始或第一个轮循尚未完成。 1 = 至少完成了 1 个轮循周期。</p>								

<p>DRDY_BIST_OTUT = 指示 OTUT 保护器诊断的状态。当 [OTUT_GO] = 1 以及 [OTUT_MODE1:0] = 10 ( 启动 BIST 运行 ) 时清除该位，并在完成 BIST 周期时设置该位。 0 = 未启动或仍在运行。 1 = BIST 周期完成。</p>
<p>DRDY_BIST_OVUV = 指示 OVUV 保护器诊断的状态。当 [OVUV_GO] = 1 以及 [OVUV_MODE1:0] = 10 ( 启动 BIST 运行 ) 时清除该位，并在完成 BIST 周期时设置该位。 0 = 未启动或仍在运行。 1 = BIST 周期完成。</p>
<p>DRDY_BIST_PWR = 指示电源诊断的状态。当 [PWR_BIST_GO] = 1 ( 开始 BIST 运行 ) 时清除该位，并在 BIST 周期完成时设置该位。 0 = 未启动或仍在运行。 1 = BIST 周期完成。</p>

#### 6.5.4.4.2 ADC\_STAT1

地址	0x0527							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留			RSVD	DRDY_AUX_GPIO	DRDY_AUX_CELL	DRDY_AUX_MISC	DRDY_MAIN_ADC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
DRDY_AUX_GPIO = AUX ADC 已在所有配置为 ADC 测量的活动 GPIO 通道上完成至少一次测量。当 [AUX_GO] 从 0 更改为 1 时，该位被清除。 0 = 未就绪 1 = 所有 GPIO 输入已通过 AUX ADC 完成至少一次测量								
DRDY_AUX_CELL = 器件已在 [AUX_CELL_SEL4:0] 设置的所有 AUXCELL 通道上完成至少一次测量。当 [AUX_GO] 从 0 更改为 1 时，该位被清除。 0 = 未就绪 1 = 所有 [AUX_CELL_SEL4:0] 配置的通道已完成至少一次测量								
DRDY_AUX_MISC = 器件已在所有 AUX ADC MISC 输入通道上完成至少一次测量 ( 即完成单次轮询运行 )。当 [AUX_GO] 从 0 更改为 1 时，该位被清除。 0 = 未就绪 1 = 所有 AUX ADC MISC 输入已完成至少一次测量								
DRDY_MAIN_ADC = 器件已在所有主 ADC 输入通道 ( 包括所有 GPIO ) 上完成至少一次测量 ( 即完成了单次轮询运行 )。当 [MAIN_GO] 从 0 更改为 1 时，该位被清除。 0 = 未就绪 1 = 所有主 ADC 输入已完成至少一次测量								

#### 6.5.4.4.3 ADC\_STAT2

地址	0x0528							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留		DRDY_LPF	DRDY_GPIO	DRDY_VCOW	DRDY_CBOW	DRDY_CBFET	DRDY_VCCB
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
DRDY_LPF = 器件已在所有活动电芯通道上完成至少 1 轮 LPF 检查。只要主 ADC 在运行，比较就会在后台继续进行。当 [MAIN_GO] = 1 时，该位被清除。 当使用 [LPF_FLT_INJ] 位注入故障以测试 DIAG_LPF 引擎时，也会使用该数据就绪位。当 [LPF_FLT_INJ] = 1 时，该位清零，器件将从一开始就使用故障注入 [DIAG_LPF] 重新启动 VC 和 BB 和 BB 通道 LPF 检查。检查所有通道 LPFS 之后，[DRDY_LPF] = 1。 0 = 未就绪 1 = 诊断比较完成								
DRDY_GPIO = 器件已在所有活动通道上完成 GPIO 主 ADC 和 AUX ADC 诊断比较，比较停止。当 [COMP_ADC_GO] = 1 时，该位被清除。 0 = 未就绪 1 = 诊断比较完成								

DRDY_VCOW = 器件已在所有活动通道上完成 VC OW 诊断比较，比较停止。当 [COMP_ADC_GO] = 1 时，该位被清除。 0 = 未就绪 1 = 诊断比较完成
DRDY_CBOW = 器件已在所有活动通道上完成 CB OW 诊断比较，比较停止。当 [COMP_ADC_GO] = 1 时，该位被清除。 0 = 未就绪 1 = 诊断比较完成
DRDY_CBFET = 器件已在所有活动通道上完成 CB FET 诊断比较，比较停止。当 [COMP_ADC_GO] = 1 时，该位被清除。 0 = 未就绪 1 = 诊断比较完成
DRDY_VCCB = 器件已在所有活动通道上完成 VCELL 与 AUXCELL 的诊断比较。当 [COMP_ADC_GO] = 1 时，该位被清除。 0 = 未就绪 1 = 诊断比较完成

#### 6.5.4.4.4 GPIO\_STAT

地址	0x052A							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1
复位	0	0	0	0	0	0	0	0
GPIO1 至 GPIO8 = 当 GPIO 配置为数字输入或输出时，该寄存器显示 GPIO 状态。 0 = 低电平 1 = 高电平								

#### 6.5.4.4.5 BAL\_STAT

地址	0x052B							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	INVALID_CBCONF	OT_PAUSE_DET	CB_INPAUSE	MB_RUN	CB_RUN	ABORTFLT	MB_DONE	CB_DONE
复位	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
INVALID_CBCONF = 指示由于 CB 控制设置不当，CB 无法启动（在 [BAL_GO] = 1 之后）。不正确的设置包括： <ul style="list-style-type: none"> <li>为 CB 启用了八个以上的电芯。</li> <li>在 DEVICE_CONF[NO_ADJ_CB] = 0 的情况下为 CB 启用了两个以上的相邻电芯。</li> <li>在 DEVICE_CONF[NO_ADJ_CB] = 1 的情况下为 CB 启用了任何相邻的电芯。</li> </ul> 每次 [BAL_GO] = 1 时该位都会更新。 0 = 有效的 CB 设置 1 = 无效的 CB 设置								
OT_PAUSE_DET = 如果 [OTCB_EN] = 1，则指示检测到 OTCB。如果检测到 CB TWARN，也会设置该位，这也会暂停 CB。仅在 [BAL_GO] = 1 后有效 0 = 未检测到 OTCB 或 CB TWARN 1 = 任何 NTC 热敏电阻测量值大于 OTCB_THR[3:0] 设置，或内核 (CBFET) 温度高于 CB TWARN								
CB_INPAUSE = 指示电芯均衡暂停状态。 0 = CB 正在运行或未启动 1 = 暂停（可能由 OTCB 检测或主机设置 [CB_PAUSE] = 1 引起）								
MB_RUN = 指示由器件控制的模块均衡正在运行。仅当 MB_TIMER_CTRL 不为 0x00 且在 [BAL_GO] = 1 之后有效。不指示电芯均衡状态。 0 = 已完成或未启动 1 = 由器件控制的模块均衡正在运行								
CB_RUN = 指示电芯均衡正在运行。仅在 [BAL_GO] = 1 后有效。不指示模块均衡状态。即使 CB 处于暂停状态，该位仍保持为 1。 0 = 已完成或未启动 1 = 至少 1 个电芯处于活动电芯均衡状态								

<p>ABORTFLT = 指示由于检测到未屏蔽的故障而中止电芯均衡。当 <math>BAL\_CTRL1[BAL\_GO] = 1</math> 时清除。如果 CB 处于暂停状态 (<math>CB\_INPAUSE = 1</math>)，即使检测到未屏蔽的故障，也不会触发 CB 中止。如果 CB 不再处于暂停状态，则发生故障时中止功能将恢复。</p> <p>0 = 未中止或电芯均衡未运行 1 = 已中止</p>
<p>MB_DONE = 指示模块均衡已完成。当 <math>BAL\_CTRL1[BAL\_GO] = 1</math> 时清除。</p> <p>0 = 模块均衡仍在运行或尚未启动 1 = 模块均衡已完成</p>
<p>CB_DONE = 指示所有电芯均衡已完成。当 <math>BAL\_CTRL1[BAL\_GO] = 1</math> 时清除。</p> <p>0 = 电芯均衡仍在运行或尚未启动 1 = 所有电芯均衡已完成</p>

#### 6.5.4.4.6 DEV\_STAT

地址	0x052C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD	FACT_CRC_DONE	CUST_CRC_DONE	OTUT_RUN	OVUV_RUN	RSVD	AUX_RUN	MAIN_RUN
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
FACT_CRC_DONE = 指示出厂 CRC 状态机的状态。当在内部至少计算和验证一次出厂 CRC 时，会设置该位。读取该寄存器将清除此位。 0 = 未完成 1 = 已完成 (读取时清除)								
CUST_CRC_DONE = 指示客户 CRC 状态机的状态。当至少计算 CRC 一次并将其与 CUST_CRC* 寄存器进行比较时，会设置该位。读取该寄存器将清除此位。 0 = 未完成 1 = 已完成 (读取时清除)								
OTUT_RUN = 显示 OTUT 保护器比较器的状态。当 OTUT BIST 启动时会设置该位。当 BIST 完成或中止时，器件将自动关闭 OT 和 UT 比较器，然后该位将被清除。 0 = 关闭 (即 OTUT 未启动或当 $[OTUT\_GO] = 1$ 且 $[OTUT\_MODE1:0] = 0$ 时) 1 = 开启 (即当 $[OTUT\_GO] = 1$ 且 $[OTUT\_MODE1:0]$ 不为零时)								
OVUV_RUN = 显示 OVUV 保护器比较器的状态。当 OVUV BIST 启动时会设置该位。当 BIST 完成或中止时，器件将自动关闭 OV 和 UV 比较器，然后该位将被清除。 0 = 关闭 (即 OVUV 未启动或当 $[OVUV\_GO] = 1$ 且 $[OVUV\_MODE1:0] = 0$ 时) 1 = 开启 (即当 $[OVUV\_GO] = 1$ 且 $[OVUV\_MODE1:0]$ 不为零时)								
AUX_RUN = 显示 AUX ADC 的状态。 0 = 关闭 1 = 开启								
MAIN_RUN = 显示主 ADC 的状态。 0 = 关闭 1 = 开启								

#### 6.5.4.5 ADC 配置和控制

##### 6.5.4.5.1 ADC\_CONF1

地址	0x0007							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	AUX_SETTLE[1:0]		LPF_BB[2:0]			LPF_VCELL[2:0]		
复位	0	0	0	0	0	0	0	0
AUX_SETTLE[1:0] = AUXCELL 配置 AUX CELL 稳定时间。每个 AUXCELL 都必须等待抗混叠滤波器 (AAF) 稳定时间，才能被视为有效的测量值。这些位提供了使用不同的 AAF 或绕过 AAF 以换取快速测量的选项。 00 = 4.3ms 01 = 2.3ms 10 = 1.3ms 11 = 保留								

LPF\_BB[2:0] = 配置用于 BBP/N 测量的后主 SAR ADC 低通滤波器截止频率。与 LPF\_VCELL[2:0] 相同的选项。

LPF\_VCELL[2:0] = 配置用于 VCELL 测量的后 ADC 低通滤波器截止频率。  
 0x0 = 6.5Hz ( 154ms 平均值 )  
 0x1 = 13Hz ( 77ms 平均值 )  
 0x2 = 26Hz ( 38ms 平均值 )  
 0x3 = 53Hz ( 19ms 平均值 )  
 0x4 = 111Hz ( 9ms 平均值 )  
 0x5 = 240Hz ( 4ms 平均值 )  
 0x6 = 600Hz ( 1.6ms 平均值 )  
 0x7 = 240Hz

#### 6.5.4.5.2 ADC\_CONF2

地址	0x0008							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[1:0]			ADC_DLY[5:0]				
复位	0	0	0	0	0	0	0	0
SPARE[1:0] = 备用								
ADC_DLY[5:0] = 如果 [MAIN_GO] 位被写为 1, 则主 ADC 会在延迟该设置时间后开始转换。该设置在整个菊花链堆栈中同步主 ADC 的启动。 选项范围为 0µs ( 无延迟 ) 至 200µs, 阶跃为 5µs。 未定义的代码 = 0µs ( 无延迟 )								

#### 6.5.4.5.3 MAIN\_ADC\_CAL1

地址	0x001B							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	GAINL[7:0]							
复位	0	0	0	0	0	0	0	0
GAINL[7:0] = 主 ADC 25°C 增益校准结果 ( 低 8 位 )。如果客户在生产流程期间执行增益校准, 则增益结果可以编程到 OTP 中, 并将在器件复位时发送到该增益寄存器。在 ADC 校正步骤期间, 器件会自动应用此数据。 范围为 -0.78125% 至 0.7782%, 阶跃为 0.0031%。								

#### 6.5.4.5.4 MAIN\_ADC\_CAL2

地址	0x001C							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	GAINH	OFFSET[6:0]						
复位	0	0	0	0	0	0	0	0
GAINH 主 ADC 25°C 增益校准结果 ( MS 位 )。如果客户在生产流程期间执行增益校准, 则增益结果可以编程到 OTP 中, 并将在器件复位时发送到该增益寄存器。在 ADC 校正步骤期间, 器件会自动应用此数据。 范围为 -0.78125% 至 0.7782%, 阶跃为 0.0031%。								
OFFSET[6:0] = 主 ADC 25°C 失调电压校准结果。如果客户在生产流程期间执行失调电压校准, 则失调电压结果可以编程到 OTP 中, 并将在器件复位时发送到该失调电压寄存器。在 ADC 校正步骤期间, 器件会自动应用此数据。 范围为 -12.20703mV 至 12.01630mV, 阶跃为 0.19073mV								

#### 6.5.4.5.5 AUX\_ADC\_CAL1

地址	0x001D							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	GAINL[7:0]							
复位	0	0	0	0	0	0	0	0

GAINL[7:0] = AUX ADC 25°C 增益校准结果 (低 8 位)。如果客户在生产流程期间执行增益校准, 则增益结果可以编程到 OTP 中, 并将在器件复位时发送到该增益寄存器。在 ADC 校正步骤期间, 器件会自动应用此数据。范围为 -0.78125% 至 0.7782%, 阶跃为 0.0031%。

#### 6.5.4.5.6 AUX\_ADC\_CAL2

地址	0x001E							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	GAINH	OFFSET[6:0]						
复位	0	0	0	0	0	0	0	0
GAINH AUX ADC 25°C 增益校准结果 (MS 位)。如果客户在生产流程期间执行增益校准, 则增益结果可以编程到 OTP 中, 并将在器件复位时发送到该增益寄存器。在 ADC 校正步骤期间, 器件会自动应用此数据。范围为 -0.78125% 至 0.7782%, 阶跃为 0.0031%。								
OFFSET[6:0] = AUX ADC 25°C 失调电压校准结果。如果客户在生产流程期间执行失调电压校准, 则失调电压结果可以编程到 OTP 中, 并将在器件复位时发送到该失调电压寄存器。在 ADC 校正步骤期间, 器件会自动应用此数据。范围为 -12.20703mV 至 12.01630mV, 阶跃为 0.19073mV								

#### 6.5.4.5.7 ADC\_CTRL1

地址	0x030D							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	RSVD		LPF_BB_EN	LPF_VCELL_EN	MAIN_GO	MAIN_MODE[1:0]	
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
LPF_BB_EN = 启用数字低通滤波器后 ADC 转换。LPF 仅适用于 BBP/N 测量。截止频率由 ADC_CONFIG1[LPF_BB[2:0]] 进行配置。								
LPF_VCELL_EN = 启用数字低通滤波器后 ADC 转换。LPF 仅适用于 VCELL 测量。截止频率由 ADC_CONFIG1[LPF_VCELL[2:0]] 进行配置。								
MAIN_GO = 启动主 ADC 转换。向该位写入 1 时, 会对所有主 ADC 输入进行采样。主 ADC 启动后, 在再次向该位写入 1 之前, 对主 ADC 控制设置进行的任何更改都不会生效。该位在读取时清零。 0 = 就绪。写入 0 无效 1 = 启动主 ADC								
MAIN_MODE[1:0] = 设置主 ADC 运行模式。在连续运行中, 如果用户要停止 ADC, 用户必须读取所有 ADC 转换结果, 然后将其停止。在下次重新启用 ADC 之前, ADC 结果无效。 00 = 主 ADC 未运行 01 = 单次运行。运行主 ADC 轮询 8 次, 然后停止 10 = 连续运行。连续运行主 ADC 轮询, 直到主机发送停止命令 11 = 保留								

#### 6.5.4.5.8 ADC\_CTRL2

地址	0x030E							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	RSVD	AUX_CELL_ALIGN	AUX_CELL_SEL[4:0]				
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
AUX_CELL_ALIGN = 将 AUX ADC AUXCELL 测量与主 ADC CELL1 或 CELL8 对齐 0 = 动态对齐 1 = 与主 ADC CELL8 对齐								

AUX\_CELL\_SEL[4:0] = 选择将通过 AUX ADC 多路复用哪个 ( 哪些 ) AUXCELL 通道。  
 0x00 = 运行由 ACTIVE\_CELL\_CONF 寄存器设置的所有活动的电芯通道  
 0x01 = 锁定至 AUX 汇流条 (BBP-BBN)  
 0x02 = 锁定至 AUXCELL1  
 0x03 = 锁定至 AUXCELL2  
 0x04 = 锁定至 AUXCELL3  
 :  
 0x11 = 锁定至 AUXCELL16  
 0x12 至 0x1F = RSVD  
 注意: 如果选择非活动通道或 RSVD 代码, 器件将不会在 AUXCELL 时隙上执行 AUX ADC 转换, 并且 AUX\_CELL\_HI/LO 寄存器将保持复位值。

#### 6.5.4.5.9 ADC\_CTRL3

地址	0x030F							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	AUX_GPIO_SEL[3:0]				AUX_GO	AUX_MODE[1:0]	
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
AUX_GPIO_SEL[3:0] = 选择将通过 AUX ADC 多路复用哪个 ( ) GPIO 通道以用于温度测量诊断。如果该选择未设置为 0x00, 那么 AUX ADC 将锁定在单个 GPIO 通道上, 并且测量结果将输出到 AUX_GPIO_HI/LO 寄存器。 0x00 = AUX ADC 循环使用所有配置为仅 ADC 或 ADC 和 OTUT 的 GPIO 通道。 0x01 = 锁定至 GPIO1 0x02 = 锁定至 GPIO2 : 0x08 = 锁定至 GPIO8 所有其他代码均为 RSVD。 注: 如果 GPIO 未配置为 ADC 测量或选择了 RSVD 代码, 则器件不会在 GPIO 时隙上执行 AUX ADC 转换, 并且 AUX_GPIO_HI/LO 寄存器将保持复位值。								
AUX_GO = 启动 AUX ADC 转换。向该位写入 1 时, 会对所有 AUX ADC 输入进行采样。AUX ADC 启动后, 在再次向该位写入 1 之前, 对 AUX ADC 控制设置进行的任何更改都不会生效。该位在读取时清零。 0 = 就绪。写入 0 无效。 1 = 启动 AUX ADC								
AUX_MODE[1:0] = 设置主 ADC 运行模式。在连续运行中, 如果用户要停止 ADC, 用户必须读取所有 ADC 转换结果, 然后将其停止。在下次重新启用 ADC 之前, ADC 结果无效。 00 = AUX ADC 未运行 01 = 单次运行。运行 AUX ADC 轮询一次, 然后停止。 10 = 连续运行。持续运行 AUX ADC 轮询, 直到主机发送停止命令。 11 = 8 轮询运行, 用于测量全部八个 GPIO 一次。								

### 6.5.4.6 ADC 测量结果

#### 6.5.4.6.1 VCELL16\_HI/LO

##### VCELL16\_HI

地址	0x0568							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell16 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell16 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL16\_LO

地址	0x0569							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell16 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.2 VCELL15\_HI/LO

##### VCELL15\_HI

地址	0x056A							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell15 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell15 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL15\_LO

地址	0x056B							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell15 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.3 VCELL14\_HI/LO

##### VCELL14\_HI

地址	0x056C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell14 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell14 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL14\_LO

地址	0x056D							
----	--------	--	--	--	--	--	--	--

只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell14 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.4 VCELL13\_HI/LO

##### VCELL13\_HI

地址	0x056E							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell13 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell13 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL13\_LO

地址	0x056F							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell13 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.5 VCELL12\_HI/LO

##### VCELL12\_HI

地址	0x0570							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell12 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell12 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL12\_LO

地址	0x0571							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell12 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.6 VCELL11\_HI/LO

##### VCELL11\_HI

地址	0x0572							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0

RESULT[7:0] = Cell11 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell11 电压低字节以防止更新，直到读取高字节和低字节寄存器。

### VCELL11\_LO

地址	0x0573							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell11 电压的低字节的 ADC 测量结果，以二进制补码表示。								

### 6.5.4.6.7 VCELL10\_HI/LO

#### VCELL10\_HI

地址	0x0574							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell10 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell10 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

### VCELL10\_LO

地址	0x0575							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell10 电压的低字节的 ADC 测量结果，以二进制补码表示。								

### 6.5.4.6.8 VCELL9\_HI/LO

#### VCELL9\_HI

地址	0x0576							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell9 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell9 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

### VCELL9\_LO

地址	0x0577							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell9 电压的低字节的 ADC 测量结果，以二进制补码表示。								

### 6.5.4.6.9 VCELL8\_HI/LO

#### VCELL8\_HI

地址	0x0578							
----	--------	--	--	--	--	--	--	--

只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell8 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell8 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

### VCELL8\_LO

地址	0x0579							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell8 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.10 VCELL7\_HI/LO

### VCELL7\_HI

地址	0x057A							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell7 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell7 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

### VCELL7\_LO

地址	0x057B							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell7 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.11 VCELL6\_HI/LO

### VCELL6\_HI

地址	0x057C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell6 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell6 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

### VCELL6\_LO

地址	0x057D							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell6 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.12 VCELL5\_HI/LO

##### VCELL5\_HI

地址	0x057E							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell5 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell5 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL5\_LO

地址	0x057F							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell5 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.13 VCELL4\_HI/LO

##### VCELL4\_HI

地址	0x0580							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell4 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell4 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL4\_LO

地址	0x0581							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell4 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.14 VCELL3\_HI/LO

##### VCELL3\_HI

地址	0x0582							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell3 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell3 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL3\_LO

地址	0x0583							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0

名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell3 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.15 VCELL2\_HI/LO

##### VCELL2\_HI

地址	0x0584							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell2 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell2 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL2\_LO

地址	0x0585							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell2 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.16 VCELL1\_HI/LO

##### VCELL1\_HI

地址	0x0586							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = Cell1 电压的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定 Cell1 电压低字节以防止更新，直到读取高字节和低字节寄存器。								

##### VCELL1\_LO

地址	0x0587							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = Cell1 电压的低字节的 ADC 测量结果，以二进制补码表示。								

#### 6.5.4.6.17 BUSBAR\_HI/LO

##### BUSBAR\_HI

地址	0x0588							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 差分汇流条引脚 (BBP - BBN) 的高字节的 ADC 测量结果，以二进制补码表示。当主机读取该寄存器时，器件会锁定低字节以防止更新，直到读取高字节和低字节寄存器。								

**BUSBAR\_LO**

<b>地址</b>	<b>0x0589</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 差分汇流条引脚 (BBP - BBN) 的低字节的 ADC 测量结果，以二进制补码表示。								

**6.5.4.6.18 TSREF\_HI/LO****TSREF\_HI**

<b>地址</b>	<b>0x058C</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自主 ADC 的 TSREF 高字节结果。当主机读取该寄存器时，器件会锁定 TSREF 低字节以防止更新，直到读取高字节和低字节寄存器。								

**TSREF\_LO**

<b>地址</b>	<b>0x058D</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自主 ADC 的 TSREF 低字节结果								

**6.5.4.6.19 GPIO1\_HI/LO****GPIO1\_HI**

<b>地址</b>	<b>0x058E</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO1 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO1 低字节以防止更新，直到读取高字节和低字节寄存器。								

**GPIO1\_LO**

<b>地址</b>	<b>0x058F</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO1 的 ADC 测量低字节结果。								

**6.5.4.6.20 GPIO2\_HI/LO****GPIO2\_HI**

<b>地址</b>	<b>0x0590</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							

复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO2 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO2 低字节以防止更新，直到读取高字节和低字节寄存器。								

### GPIO2\_LO

地址	0x0591							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO2 的 ADC 测量低字节结果。								

#### 6.5.4.6.21 GPIO3\_HI/LO

### GPIO3\_HI

地址	0x0592							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO3 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO3 低字节以防止更新，直到读取高字节和低字节寄存器。								

### GPIO3\_LO

地址	0x0593							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO3 的 ADC 测量低字节结果。								

#### 6.5.4.6.22 GPIO4\_HI/LO

### GPIO4\_HI

地址	0x0594							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO4 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO4 低字节以防止更新，直到读取高字节和低字节寄存器。								

### GPIO4\_LO

地址	0x0595							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO4 的 ADC 测量低字节结果。								

#### 6.5.4.6.23 GPIO5\_HI/LO

### GPIO5\_HI

<b>地址</b>	<b>0x0596</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO5 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO5 低字节以防止更新，直到读取高字节和低字节寄存器。								

## GPIO5\_LO

<b>地址</b>	<b>0x0597</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO5 的 ADC 测量低字节结果。								

### 6.5.4.6.24 GPIO6\_HI/LO

## GPIO6\_HI

<b>地址</b>	<b>0x0598</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO6 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO6 低字节以防止更新，直到读取高字节和低字节寄存器。								

## GPIO6\_LO

<b>地址</b>	<b>0x0599</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO6 的 ADC 测量低字节结果。								

### 6.5.4.6.25 GPIO7\_HI/LO

## GPIO7\_HI

<b>地址</b>	<b>0x059A</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO7 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO7 低字节以防止更新，直到读取高字节和低字节寄存器。								

## GPIO7\_LO

<b>地址</b>	<b>0x059B</b>							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0

RESULT[7:0] = GPIO7 的 ADC 测量低字节结果。

#### 6.5.4.6.26 GPIO8\_HI/LO

##### GPIO8\_HI

地址	0x059C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO8 的 ADC 测量高字节结果。当主机读取该寄存器时，器件会锁定 GPIO8 低字节以防止更新，直到读取高字节和低字节寄存器。								

##### GPIO8\_LO

地址	0x059D							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO8 的 ADC 测量低字节结果。								

#### 6.5.4.6.27 DIETEMP1\_HI/LO

##### DIETEMP1\_HI

地址	0x05AE							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自主 ADC 的 DieTemp1 高字节结果。当主机读取该寄存器时，器件会锁定 DIETEMP1 低字节以防止更新，直到读取高字节和低字节寄存器。								

##### DIETEMP1\_LO

地址	0x05AF							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自主 ADC 的 DieTemp1 低字节 (用于 ADC 校正的温度)。								

#### 6.5.4.6.28 DIETEMP2\_HI/LO

##### DIETEMP2\_HI

地址	0x05B0							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 DieTemp2 高字节结果。当主机读取该寄存器时，器件会锁定 DIETEMP2 低字节以防止更新，直到读取高字节和低字节寄存器。								

##### DIETEMP2\_LO

地址	0x05B1							
----	--------	--	--	--	--	--	--	--

只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 DieTemp2 低字节 (用于 ADC 校正的温度) 结果								

#### 6.5.4.6.29 AUX\_CELL\_HI/LO

##### AUX\_CELL\_HI

地址	0x05B2							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = AUXCELL 电压的高字节的 ADC 测量结果, 以二进制补码表示。如果主机将 [AUX_CELL_SEL4:0] 配置为锁定到单个 AUXCELL 通道, 则这些 AUX_CELL_HI/LO 寄存器将仅报告 AUXCELL 电压测量值。 当主机读取该寄存器时, 器件会锁定 AUXCELL 电压低字节以防止更新, 直到读取高字节和低字节寄存器。								

##### AUX\_CELL\_LO

地址	0x05B3							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = AUX 电芯电压的低字节的 ADC 测量结果, 以二进制补码表示。如果主机将 [AUX_CELL_SEL4:0] 配置为锁定到单个 AUXCELL 通道, 则这些 AUX_CELL_HI/LO 寄存器将仅报告 AUXCELL 电压测量值。								

#### 6.5.4.6.30 AUX\_GPIO\_HI/LO

##### AUX\_GPIO\_HI

地址	0x05B4							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = GPIO 的由 [AUXGPIO_SEL3:0] 位锁定的 AUX ADC 测量高字节结果。当主机读取该寄存器时, 器件会锁定 AUX_GPIO 低字节以防止更新, 直到读取高字节和低字节寄存器。								

##### AUX\_GPIO\_LO

地址	0x05B5							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = GPIO 的由 [AUXGPIO_SEL3:0] 位锁定的 AUX ADC 测量低字节结果。								

#### 6.5.4.6.31 AUX\_BAT\_HI/LO

##### AUX\_BAT\_HI

地址	0x05B6							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0

RESULT[7:0] = 来自 AUX ADC 的 BAT 引脚测量的高字节结果。当主机读取该寄存器时，器件会锁定 AUX\_BAT 低字节以防止更新，直到读取高字节和低字节寄存器。

### AUX\_BAT\_LO

地址	0x05B7							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 BAT 引脚测量的低字节结果。								

### 6.5.4.6.32 AUX\_REFL\_HI/LO

#### AUX\_REFL\_HI

地址	0x05B8							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 内部基准 REFL 的高字节结果，来自 AUX ADC 的测量结果。当主机读取该寄存器时，器件会锁定 AUX_REL 低字节以防止更新，直到读取高字节和低字节寄存器。								

#### AUX\_REFL\_LO

地址	0x05B9							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 内部基准 REFL 的低字节结果，来自 AUX ADC 的测量结果。								

### 6.5.4.6.33 AUX\_VBG2\_HI/LO

#### AUX\_VBG2\_HI

地址	0x05BA							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 内部基准 VBG2 的高字节结果，来自 AUX ADC 的测量结果。当主机读取该寄存器时，器件会锁定 AUX_VBG2 低字节以防止更新，直到读取高字节和低字节寄存器。								

#### AUX\_VBG2\_LO

地址	0x05BB							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 内部基准 VBG2 的低字节结果，来自 AUX ADC 的测量结果。								

### 6.5.4.6.34 AUX\_AVAO\_REF\_HI/LO

#### AUX\_AVAO\_REF\_HI

地址	0x05BE							
----	--------	--	--	--	--	--	--	--

只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 AVAO_REF 测量的高字节结果。当主机读取该寄存器时，器件会锁定 AUX_AVAO_REF 低字节以防止更新，直到读取高字节和低字节寄存器。								

### AUX\_AVAO\_REF\_LO

地址	0x05BF							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 AVAO_REF 测量的低字节结果。								

#### 6.5.4.6.35 AUX\_AVDD\_REF\_HI/LO

### AUX\_AVDD\_REF\_HI

地址	0x05C0							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 AVDD_REF 测量的高字节结果。当主机读取该寄存器时，器件会锁定 AUX_AVDD_REF 低字节以防止更新，直到读取高字节和低字节寄存器。								

### AUX\_AVDD\_REF\_LO

地址	0x05C1							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 AVDD_REF 测量的低字节结果。								

#### 6.5.4.6.36 AUX\_OV\_DAC\_HI/LO

### AUX\_OV\_DAC\_HI

地址	0x05C2							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 OV 比较器 DAC 测量的高字节结果 (0.8 x OV 阈值)。当主机读取该寄存器时，器件会锁定 AUX_OV_DAC 低字节以防止更新，直到读取高字节和低字节寄存器。								

### AUX\_OV\_DAC\_LO

地址	0x05C3							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 OV 比较器 DAC 测量的低字节结果 (0.8 x OV 阈值)。								

#### 6.5.4.6.37 AUX\_UV\_DAC\_HI/LO

##### AUX\_UV\_DAC\_HI

地址	0x05C4							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 UV 比较器 DAC 测量的高字节结果 ( 0.8 x UV 阈值 )。当主机读取该寄存器时, 器件会锁定 AUX_UV_DAC 低字节以防止更新, 直到读取高字节和低字节寄存器。								

##### AUX\_UV\_DAC\_LO

地址	0x05C5							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 UV 比较器 DAC 测量的低字节结果 ( 0.8 x UV 阈值 )。								

#### 6.5.4.6.38 AUX\_OT\_OTCB\_DAC\_HI/LO

##### AUX\_OT\_OTCB\_DAC\_HI

地址	0x05C6							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 OT 比较器 ( 基于 [OTCB_THR_LOCK] 设置的 OT 或 OTCB 阈值 ) DAC 测量的高字节结果。当主机读取该寄存器时, 器件会锁定 AUX_OT_OTCB_DAC 低字节以防止更新, 直到读取高字节和低字节寄存器。								

##### AUX\_OT\_OTCB\_DAC\_LO

地址	0x05C7							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 OT 比较器 ( 基于 [OTCB_THR_LOCK] 设置的 OT 或 OTCB 阈值 ) DAC 测量的低字节结果。								

#### 6.5.4.6.39 AUX\_UT\_DAC\_HI/LO

##### AUX\_UT\_DAC\_HI

地址	0x05C8							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 UT 比较器 DAC 测量的高字节结果。当主机读取该寄存器时, 器件会锁定 AUX_UT_DAC 低字节以防止更新, 直到读取高字节和低字节寄存器。								

##### AUX\_UT\_DAC\_LO

地址	0x05C9							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0

名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 UT 比较器 DAC 测量的低字节结果。								

#### 6.5.4.6.40 AUX\_VCBDONE\_DAC\_HI/LO

##### AUX\_VCBDONE\_DAC\_HI

地址	0x05CA							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 UV 比较器 (VCBDONE 阈值) DAC 测量的高字节结果。当主机读取该寄存器时, 器件会锁定 AUX_VCBDONE_DAC 低字节以防止更新, 直到读取高字节和低字节寄存器。								

##### AUX\_VCBDONE\_DAC\_LO

地址	0x05CB							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 UV 比较器 (VCBDONE 阈值) DAC 测量的低字节结果。								

#### 6.5.4.6.41 AUX\_VCM\_HI/LO

##### AUX\_VCM\_HI

地址	0x05CC	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 VCM (主 ADC 上的共模电压) 测量的高字节结果。当主机读取该寄存器时, 器件会锁定 AUX_VCM 低字节以防止更新, 直到读取高字节和低字节寄存器。								

##### AUX\_VCM\_LO

地址	0x05CD							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 来自 AUX ADC 的 VCM (主 ADC 上的共模电压) 测量的低字节结果。								

#### 6.5.4.6.42 REFOVDAC\_HI/LO

##### REFOVDAC\_HI

地址	0x05D0							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 出厂修整的记录 OVDAC 基准电压的高字节结果。								

##### REFOVDAC\_LO

地址	0x05D1							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 出厂修整的记录 OVDAC 基准电压的低字节结果。								

#### 6.5.4.6.43 DIAG\_MAIN\_HI/LO

##### DIAG\_MAIN\_HI

地址	0x05D2							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 用于诊断 ADC 比较的报告主 ADC 比较值的高字节结果。如果在单个通道锁定时运行诊断 ADC 比较，则有效								

##### DIAG\_MAIN\_LO

地址	0x05D3							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 用于诊断 ADC 比较的报告主 ADC 比较值的低字节结果。如果在单个通道锁定时运行诊断 ADC 比较，则有效								

#### 6.5.4.6.44 DIAG\_AUX\_HI/LO

##### DIAG\_AUX\_HI

地址	0x05D4							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	1	0	0	0	0	0	0	0
RESULT[7:0] = 用于诊断 ADC 比较的报告 AUX ADC 比较值的高字节结果。如果在单个通道锁定时运行诊断 ADC 比较，则有效								

##### DIAG\_AUX\_LO

地址	0x05D5							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RESULT[7:0]							
复位	0	0	0	0	0	0	0	0
RESULT[7:0] = 用于诊断 ADC 比较的报告 AUX ADC 比较值的低字节结果。如果在单个通道锁定时运行诊断 ADC 比较，则有效								

#### 6.5.4.7 均衡配置、控制和状态

##### 6.5.4.7.1 CB\_CELL16\_CTRL 至 CB\_CELL1\_CTRL

地址	0x0318 至 0x0327							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留			TIME[4:0]				
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								

TIME[4:0] = 设置 cell\* 均衡的计时器。只要主机 MCU 设置 [BAL\_GO] = 1, 就会对选择进行采样。  
 0x00 = 0s = 停止均衡  
 0x01 = 10s  
 0x02 = 30s  
 0x03 = 60s  
 0x04 = 300s  
 0x05 至 0x10 = 范围为 10 分钟至 120 分钟, 阶跃为 10 分钟  
 0x11 至 0x1F = 范围为 150 分钟至 540 分钟 (阶跃为 30 分钟) 以及 600 分钟

#### 6.5.4.7.2 VMB\_DONE\_THRESH

地址	0x0328							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留		MB_THR[5:0]					
复位	0	0	1	1	1	1	1	1
RSVD = 被保留								
MB_THR[5:0] = 如果 MB_TIMER_CTRL 不为 0x00 且 BAT 电压小于该阈值, 则通过 GPIO3 进行的模块均衡停止。只要主机 MCU 设置 [AUX_GO] = 1, 就会对选择进行采样。 注意: 为了使用该选项, MCU 在发送 [BAL_GO] = 1 之前先启用 AUX ADC。如果 MCU 重新发送 [AUX_GO] = 1, 则新的阈值设置将生效。无需重新发送 [BAL_GO] = 1, 因为均衡已在运行。 范围为 18V 至 65V, 阶跃为 1V。未使用的代码默认为 65V。								

#### 6.5.4.7.3 MB\_TIMER\_CTRL

地址	0x0329							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留		TIME[4:0]					
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
TIME[4:0] = 设置用于模块均衡的计时器。只要主机 MCU 设置 [BAL_GO] = 1, 就会对选择进行采样。 0x00 = 0s = 停止均衡 0x01 = 10s 0x02 = 30s 0x03 = 60s 0x04 = 300s 0x05 至 0x10 = 范围为 10 分钟至 120 分钟, 阶跃为 10 分钟 0x11 至 0x1F = 范围为 150 分钟至 540 分钟 (阶跃为 30 分钟) 以及 600 分钟								

#### 6.5.4.7.4 VCB\_DONE\_THRESH

地址	0x032A							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留		CB_THR[5:0]					
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
CB_THR[5:0] = 如果电芯电压低于该阈值, 则该电芯上的电芯均衡会停止。该阈值设置适用于所有电芯。只要主机 MCU 设置 [OVUV_GO] = 1, 就会对选择进行采样。 注意: 为了使用 VCB_DONE 检测功能, 主机设置该阈值, 然后在启动 CB 之前发出 [OVUV_GO] = 1 (即发送 [BAL_GO] = 1)。 要更改 VCB_DONE 阈值检测, 请设置新阈值, 然后重新发出 [OVUV_GO] = 1 以使新阈值生效。在这种情况下, 无需重新发出 [BAL_GO] = 1 来重新启动均衡。 范围为 2.45V 至 4V, 阶跃为 25mV, 其中 0x00 = 根据 CB_DONE 比较禁用电压 0x01 = 2.45V 的阈值 0x3F = 4V 的阈值								

#### 6.5.4.7.5 OTCB\_THRESH

地址	0x032B							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	COOLOFF[2:0]			OTCB_THR[3:0]			
复位	0	0	0	0	1	1	1	1
RSVD = 被保留								
COOLOFF[2:0] = 设置 COOLOFF 迟滞 ( 恢复温度 = OTCB_THR - COOLOFF 迟滞 ) , 以便在 BAL_CTRL1[OTCB_EN] = 1 且检测到 OTCB 时恢复 CB。MCU 将相应的 GPIO 配置为 ADC 和 OTUT 选项。 范围为 4% 至 14% , 阶跃为 2%。 未使用的代码设置为 14%。								
OTCB_THR[3:0] = 当 BAL_CTRL1[OTCB_EN] = 1 时设置 OTCB 阈值。MCU 将相应的 GPIO 配置为 ADC 和 OTUT 选项。 范围为 10% 至 24% , 阶跃为 2%。 未使用的代码设置为 24%。								

#### 6.5.4.7.6 BAL\_CTRL1

地址	0x032E							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留					DUTY[2:0]		
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
DUTY[2:0] = 只要主机 MCU 设置 [BAL_GO] = 1 , 就会对选择进行采样。 0x0 = 5s 0x1 = 10s 0x2 = 30s 0x3 = 60s 0x4 = 5 分钟 0x5 = 10 分钟 0x6 = 20 分钟 0x7 = 30 分钟								

#### 6.5.4.7.7 BAL\_CTRL2

地址	0x032F							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	CB_PAUSE	FLTSTOP_EN	OTCB_EN	BAL_ACT[1:0]		BAL_GO	AUTO_BAL
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
CB_PAUSE = 暂停所有电芯的电芯均衡以允许运行诊断。 0 = 正常电芯均衡操作 1 = 暂停所有电芯均衡								
FLTSTOP_EN = 如果发生未屏蔽的故障, 则停止电芯或模块均衡。只要主机 MCU 设置 [BAL_GO] = 1 , 就会对选择进行采样。 0 = 无论故障情况如何 ( 热关断除外 ) , 均衡都持续进行 1 = 当发生任何未屏蔽的故障时, 所有 CB 均衡都停止								
OTCB_EN = 在电芯均衡期间启用 OTCB 检测。只要主机 MCU 设置 [BAL_GO] = 1 , 就会对选择进行采样。 0 = 禁用 OTCB 检测 1 = 启用 OTCB 检测								
BAL_ACT[1:0] = 控制 MB 和 CB 完成时的器件操作。只要主机 MCU 设置 [BAL_GO] = 1 , 就会对这些位进行采样。操作有效。 00 = 无操作 01 = 进入 SLEEP 模式 10 = 进入 SHUTDOWN 模式 11 = 保留								

**BAL\_GO** = 开始电芯或模块均衡。当写入 1 时，将对所有均衡配置寄存器进行采样。在再次向该位写入 1 之前，对配置寄存器的任何更改都不会生效。该位会自行清零。  
0 = 就绪  
1 = 开始均衡

**AUTO\_BAL** = 在自动或手动电芯均衡控制之间进行选择。只要主机 MCU 设置  $[BAL\_GO] = 1$ ，就会对选择进行采样。  
0 = 手动电芯均衡  
1 = 自动电芯均衡

#### 6.5.4.7.8 BAL\_CTRL3

地址	0x0330							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留			BAL_TIME_SEL[3:0]				BAL_TIME_GO
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
BAL_TIME_GO 指示器件向 BAL_TIME 寄存器报告选定的 CB 通道 (由 [BAL_TIME_SEL3:0] 设置) 剩余均衡时间								
BAL_TIME_SEL[3:0] = 选择单个 CB 通道以报告其剩余均衡时间 0x0 = CB 通道 1 0x1 = CB 通道 2 : 0xF = CB 通道 16								

#### 6.5.4.7.9 CB\_COMPLETE1

地址	0x0556							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL16_DONE	CELL15_DONE	CELL14_DONE	CELL13_DONE	CELL12_DONE	CELL11_DONE	CELL10_DONE	CELL9_DONE
复位	0	0	0	0	0	0	0	0
CELL9_DONE 至 cell9 至 cell16 的电芯均衡完成。当 MCU 设置 $[BAL\_GO] = 1$ 时，该寄存器被清除。 CELL16_DONE = 0 = 特定电芯上的均衡仍在运行或尚未开始 1 = 特定电芯上的均衡已完成								

#### 6.5.4.7.10 CB\_COMPLETE2

地址	0x0557							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL8_DONE	CELL7_DONE	CELL6_DONE	CELL5_DONE	CELL4_DONE	CELL3_DONE	CELL2_DONE	CELL1_DONE
复位	0	0	0	0	0	0	0	0
CELL1_DONE 至 cell1 至 cell8 的电芯均衡完成。当 MCU 设置 $[BAL\_GO] = 1$ 时，该寄存器被清除。 CELL8_DONE = 0 = 特定电芯上的均衡仍在运行或尚未开始 1 = 特定电芯上的均衡已完成								

#### 6.5.4.7.11 BAL\_TIME

地址	0x0558							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	TIME_UNIT	TIME[6:0]						
复位	0	0	0	0	0	0	0	0
TIME_UNIT = 指示 [TIME6:0] 报告的单位 0 = 秒 1 = 分钟								

TIME[6:0] = 报告选定的 CB 通道剩余均衡时间  
 如果 [TIME\_UNIT] = 0, 则报告的时间以秒为单位, 阶跃为 5 秒  
 如果 [TIME\_UNIT] = 1, 则报告的时间以分钟为单位, 阶跃为 5 分钟

### 6.5.4.8 保护器配置和控制

#### 6.5.4.8.1 OV\_THRESH

地址	0x0009							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE	SPARE	OV_THR[5:0]					
复位	0	0	1	1	1	1	1	1
SPARE = 备用								
OV_THR[5:0] = 设置 OV 比较器的过压阈值。对这些位的更改要求主机发送另一条 [OVUV_GO] = 1 命令。 所有设置的阶跃均为 25mV。 0x02 至 0x0E : 范围为 2700mV 至 3000mV 0x12 至 0x1E : 范围为 3500mV 至 3800mV 0x22 至 0x2E : 范围为 4175mV 至 4475mV 所有其他设置将默认为 2700mV。								

#### 6.5.4.8.2 UV\_THRESH

地址	0x000A							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE	SPARE	UV_THR[5:0]					
复位	0	0	0	0	0	0	0	0
SPARE = 备用								
UV_THR[5:0] = 设置 UV 比较器的欠压阈值。对这些位的更改要求主机发送另一条 [OVUV_GO] = 1 命令。 所有设置的阶跃均为 50mV。 0x00 至 0x26 : 范围为 1200mV 至 3100mV 所有其他设置将默认为 3100mV。								

#### 6.5.4.8.3 UV\_DISABLE1

地址	0x000C							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL16	CELL15	CELL14	CELL13	CELL12	CELL11	CELL10	CELL9
复位	0	0	0	0	0	0	0	0
CELL9 至 指示应从 UV 和 VCB_DONE 检测中排除哪些通道 CELL16 = 0 = UV 和 VCB_DONE 监测适用于通道 1 = 从通道中排除 UV 和 VCB_DOME 监测								

#### 6.5.4.8.4 UV\_DISABLE2

地址	0x000D							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL8	CELL7	CELL6	CELL5	CELL4	CELL3	CELL2	CELL1
复位	0	0	0	0	0	0	0	0
CELL8 至 指示应从 UV 和 VCB_DONE 检测中排除哪些通道 CELL1 = 0 = UV 和 VCB_DONE 监测适用于通道 1 = 从通道中排除 UV 和 VCB_DOME 监测								

#### 6.5.4.8.5 OTUT\_THRESH

地址	0x000B							

NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	UT_THR[2:0]			OT_THR[4:0]				
复位	1	1	1	0	0	0	0	0
UT_THR[2:0] = 设置 UT 比较器的 UT 阈值。对这些位的更改要求主机发送另一条 [OTUT_GO] = 1 命令。MCU 将相应的 GPIO 配置为 ADC 和 OTUT 输入。 范围为 66% 至 80%，阶跃为 2%								
OT_THR[4:0] = 设置 OT 比较器的 OT 阈值。对这些位的更改要求主机发送另一条 [OTUT_GO] = 1 命令。MCU 将相应的 GPIO 配置为 ADC 和 OTUT 输入。 范围位 10% 至 39%，阶跃为 1% 未使用的代码默认为 39%。								

#### 6.5.4.8.6 OVUV\_CTRL

地址	0x032C							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	VCBDONE_THR_LOCK	OVUV_LOCK[3:0]				OVUV_GO	OVUV_MODE[1:0]	
复位	0	0	0	0	0	0	0	0
VCBDONE_THR_LOCK = 由于 UV 比较器在 UV 阈值和 VCBDONE 阈值之间切换以测量 UV DAC 或 VCBDONE DAC 阈值结果，从而进行诊断，因此 UV 比较器必须在开始 AUX ADC 测量之前仅锁定一个阈值。该位选择将哪个阈值锁定到 UV 比较器。 当 OVUV_MODE[1:0] 为 0b11 时对该位进行采样，锁定至单通道模式。 0 = 选择 UV 阈值 1 = 选择 VCBDONE 阈值								
OVUV_LOCK[3:0] = 当 [OVUV_MOD1:0] = 0b11 时，将特定的单个通道配置为 OV 和 UV 比较器输入。对这些位的更改要求主机发送另一条 [OVUV_GO] = 1 命令。 0x0 = 锁定到 Cell1 0x1 = 锁定到 Cell2 0x2 = 锁定到 Cell3 : 0xF = 锁定到 Cell16								
OVUV_GO = 启动 OV 和 UV 比较器。当写入 1 时，会对所有 OVUV 配置设置进行采样。该位会自行清除。 0 = 就绪 1 = 启动 OV 和 UV 比较器								
OVUV_MODE[1:0] = 当 [OVUV_GO] = 1 时设置 OV 和 UV 比较器运行模式。对这些位的更改要求主机发送另一条 [OVUV_GO] = 1 命令。 00 = 不运行 OV 和 UV 比较器 01 = 针对所有有效电芯运行 OV 和 UV 轮询 10 = 运行 OV 和 UV BIST 周期。 11 = 将 OV 和 UV 比较器锁定到由 [OVUV_LOCK3:0] 配置的单个通道 注意：有效电芯由 ACTIVE_CELL[NUM_CELL3:0] 寄存器定义。								

#### 6.5.4.8.7 OTUT\_CTRL

地址	0x032D							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	OTCB_THR_LOCK	OTUT_LOCK[2:0]			OTUT_GO	OTUT_MODE[1:0]	
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
OTCB_THR_LOCK = 由于 OT 比较器在 OT 阈值和 OTCB 阈值之间切换以测量 OT 或 OTCB DAC 阈值结果，从而进行诊断，因此 OT 比较器必须在开始 AUX ADC 测量之前仅锁定一个阈值。该位选择将哪个阈值锁定到 OT 比较器。当 OTUT_MODE[1:0] = 0b11 时对该位进行采样，锁定至单通道模式。 0 = 选择 OT 阈值 1 = 选择 OTCB 阈值								

OTUT\_LOCK[2:0] = 当 [OTUT\_MOD1:0] = 0b11 时，将特定的单个通道配置为 OT 和 UT 比较器输入。对这些位的更改要求主机发送另一条 [OTUT\_GO] = 1 命令。  
 0x0 = 锁定至 GPIO1A  
 0x1 = 锁定至 GPIO2A  
 :  
 0x7 = 锁定至 GPIO8A

OTUT\_GO = 启动 OT 和 UT 比较器。当写入 1 时，会对所有 OTUT 配置设置进行采样。该位会自行清除。  
 0 = 就绪  
 1 = 启动 OT 和 UT 比较器

OTUT\_MODE[1:0] = 当 [OTUT\_GO] = 1 时设置 OT 和 UT 比较器运行模式。对这些位的更改要求主机发送另一条 [OTUT\_GO] = 1 命令。  
 00 = 不运行 OT 和 UT 比较器  
 01 = 针对所有有效电芯运行 OT 和 UT 轮询  
 10 = 运行 OT 和 UT BIST 周期。  
 11 = 将 OT 和 UT 比较器锁定到由 [OTUT\_LOCK3:0] 配置的单个通道

### 6.5.4.9 GPIO 配置

#### 6.5.4.9.1 GPIO\_CONF1

地址	0x000E							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	FAULT_IN_EN	SPI_EN	GPIO2[2:0]			GPIO1[2:0]		
复位	0	0	0	0	0	0	0	0
FAULT_IN_EN = 启用 GPIO8 作为低电平有效输入，从而在输入信号为低电平时触发 NFAULT 引脚。 0 = 无故障输入功能。GPIO8 根据 [GPIO8_CONF2:0] 设置进行配置。 1 = GPIO8 设置为低电平有效输入以触发 NFAULT 引脚，忽略了 [GPIO8_CONF2:0] 设置。								
SPI_EN = 在 GPIO4、GPIO5 和 GPIO6、GPIO7 上启用 SPI 控制器。 0 = 禁用了 SPI 控制器。 1 = 启用了 SPI 控制器。覆盖 [GPIO4_CONF2:0]、[GPIO5_CONF2:0]、[GPIO6_CONF2:0] 和 [GPIO7_CONF2:0] 设置。								
GPIO2[2:0] = 配置 GPIO2。 000 = 已禁用，高阻态 001 = 作为 ADC 和 OTUT 输入 010 = 作为仅 ADC 输入 011 = 作为数字输入 100 = 作为输出高电平 101 = 作为输出低电平 110 = 作为 ADC 输入并启用弱上拉 111 = 作为 ADC 输入并启用弱下拉								
GPIO1[2:0] = 配置 GPIO1。 000 = 已禁用，高阻态 001 = 作为 ADC 和 OTUT 输入 010 = 作为仅 ADC 输入 011 = 作为数字输入 100 = 作为输出高电平 101 = 作为输出低电平 110 = 作为 ADC 输入并启用弱上拉 111 = 作为 ADC 输入并启用弱下拉								

#### 6.5.4.9.2 GPIO\_CONF2

地址	0x000F							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE	RSVD	GPIO4[2:0]			GPIO3[2:0]		
复位	0	0	0	0	0	0	0	0
SPARE = 备用								

GPIO4[2:0] = 配置 GPIO4。如果  $[SPI\_EN] = 1$ ，这些配置位将被忽略，并且该引脚用作 SPI 控制器的 SS。有关详细信息，请参阅节 6.3.6.1.7。

- 000 = 已禁用，高阻态
- 001 = 作为 ADC 和 OTUT 输入
- 010 = 作为仅 ADC 输入
- 011 = 作为数字输入
- 100 = 作为输出高电平
- 101 = 作为输出低电平
- 110 = 作为 ADC 输入并启用弱上拉
- 111 = 作为 ADC 输入并启用弱下拉

GPIO3[2:0] = 配置 GPIO3。如果  $MB\_TIMER\_CTRL$  不是 0x00，则忽略此配置，并将引脚配置为实现模块均衡。

- 000 = 已禁用，高阻态
- 001 = 作为 ADC 和 OTUT 输入
- 010 = 作为仅 ADC 输入
- 011 = 作为数字输入
- 100 = 作为输出高电平
- 101 = 作为输出低电平
- 110 = 作为 ADC 输入并启用弱上拉
- 111 = 作为 ADC 输入并启用弱下拉

#### 6.5.4.9.3 GPIO\_CONF3

地址	0x0010							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[1:0]		GPIO6[2:0]			GPIO5[2:0]		
复位	0	0	0	0	0	0	0	0
SPARE[1:0] = 备用								
GPIO6[2:0] = 配置 GPIO6。如果 $[SPI\_EN] = 1$ ，这些配置位将被忽略，并且该引脚用作 SPI 控制器的 MOSI。有关详细信息，请参阅节 6.3.6.1.7。 <ul style="list-style-type: none"> <li>000 = 已禁用，高阻态</li> <li>001 = 作为 ADC 和 OTUT 输入</li> <li>010 = 作为仅 ADC 输入</li> <li>011 = 作为数字输入</li> <li>100 = 作为输出高电平</li> <li>101 = 作为输出低电平</li> <li>110 = 作为 ADC 输入并启用弱上拉</li> <li>111 = 作为 ADC 输入并启用弱下拉</li> </ul>								
GPIO5[2:0] = 配置 GPIO5。如果 $[SPI\_EN] = 1$ ，这些配置位将被忽略，并且该引脚用作 SPI 控制器的 MISO。有关详细信息，请参阅节 6.3.6.1.7。 <ul style="list-style-type: none"> <li>000 = 已禁用，高阻态</li> <li>001 = 作为 ADC 和 OTUT 输入</li> <li>010 = 作为仅 ADC 输入</li> <li>011 = 作为数字输入</li> <li>100 = 作为输出高电平</li> <li>101 = 作为输出低电平</li> <li>110 = 作为 ADC 输入并启用弱上拉</li> <li>111 = 作为 ADC 输入并启用弱下拉</li> </ul>								

#### 6.5.4.9.4 GPIO\_CONF4

地址	0x0011							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[1:0]		GPIO8[2:0]			GPIO7[2:0]		
复位	0	0	0	0	0	0	0	0
SPARE[1:0] = 备用								

<p>GPIO8[2:0] = 配置 GPIO8。如果 [FAULT_IN_EN] = 1，则会忽略这些配置位，并且该引脚将用作输入，以便低电平有效将触发 NFAULT。</p> <p>000 = 已禁用，高阻态          001 = 作为 ADC 和 OTUT 输入          010 = 作为仅 ADC 输入          011 = 作为数字输入          100 = 作为输出高电平          101 = 作为输出低电平          110 = 作为 ADC 输入并启用弱上拉          111 = 作为 ADC 输入并启用弱下拉</p>
<p>GPIO7[2:0] = 配置 GPIO7。如果 [SPI_EN] = 1，这些配置位将被忽略，并且该引脚用作 SPI 控制器的 SCLK。有关详细信息，请参阅节 6.3.6.1.7。</p> <p>000 = 已禁用，高阻态          001 = 作为 ADC 和 OTUT 输入          010 = 作为仅 ADC 输入          011 = 作为数字输入          100 = 作为输出高电平          101 = 作为输出低电平          110 = 作为 ADC 输入并启用弱上拉          111 = 作为 ADC 输入并启用弱下拉</p>

### 6.5.4.10 SPI 控制器

#### 6.5.4.10.1 SPI\_CONF

地址	0x034D								
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	
名称	被保留	CPOL	CPHA	NUMBIT[4:0]					
复位	0	0	0	0	0	0	0	0	
RSVD = 被保留									
CPOL = 设置 SCLK 极性。 0 = 低电平时空闲，高电平时计时 1 = 高电平时空闲，低电平时计时									
CPHA = 设置在 MISO 上对数据进行采样的 SCLK 边沿。 0 = 前沿时钟转换 1 = 后沿时钟转换									
NUMBIT[4:0] = SPI 事务长度。设置要读取/写入的 SPI 位数。 00000 = 24 位 00001 = 1 位 00010 = 2 位 : 10111 = 23 位 所有其他值 = 23 位									

#### 6.5.4.10.2 SPI\_EXE

地址	0x0351							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留						SS_CTRL	SPI_GO
复位	0	0	0	0	0	0	1	0
RSVD = 被保留								
SS_CTRL = 对 SS 的状态进行编程。 0 = 输出低电平 1 = 输出高电平								

SPI\_GO = 执行 SPI 事务。该位会自行清除。  
0 = 空闲  
1 = 执行 SPI

#### 6.5.4.10.3 SPI\_TX3、SPI\_TX2 和 SPI\_TX1

地址	0x034E 至 0x0350							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	DATA[7:0]							
复位	0	0	0	0	0	0	0	0
DATA[7:0] = 要用于对 SPI 目标器件进行写入的数据。这些位通过使用 SPI_CONF[NUMBER4:0] 进行编程，从 LSB SPI_TX1 开始 -> LSB SPI_TX2 -> LSB SPI_TX3 按时钟从 MOSI 输出。								

#### 6.5.4.10.4 SPI\_RX3、SPI\_RX2 和 SPI\_RX1

地址	0x0520 至 0x0522							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	DATA[7:0]							
复位	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
DATA[7:0] = SPI 事务期间从读取返回的数据。已更新，从 LSB SPI_RX1 开始 -> LSB SPI_RX2 -> LSB SPI_RX3，位数由通过 MISO 按时钟输入的 SPI_CONF[NUMBER4:0] 设置。								

### 6.5.4.11 诊断控制

#### 6.5.4.11.1 DIAG\_OTP\_CTRL

地址	0x0335							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留			FLIP_FACT_ CRC	MARGIN_MODE[2:0]			MARGIN_GO
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
FLIP_FACT_CRC = 用于翻转出厂 CRC 值的使能位。这用于出厂 CRC 诊断。 0 = 正常运行。不修改出厂 CRC 1 = 翻转 CRC 值。这会导致出厂 CRC 故障 FAULT_OTP[FACT_CRC]。								
MARGIN_MODE[2:0] = 配置 OTP 裕度读取模式： 0b000 = 正常读取 0b001 = 保留 0b010 = 裕度 1 读取 0b011 至 0b111 = 保留								
MARGIN_GO = 开始由 [MARGIN_MOD] 位设置的 OTP 裕度测试。该位会自行清除并且始终读为 0。 0 = 就绪 1 = 开始测试								

#### 6.5.4.11.2 DIAG\_COMM\_CTRL

地址	0x0336							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD						SPI_ LOOPBACK	FLIP_TR_ CRC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								

SPI\_LOOPBACK = 启用 SPI 环回功能以验证 SPI 功能。有关更多详细信息，请参阅节 6.3.6.1.7。  
0 = 禁用  
1 = 启用

FLIP\_TR\_CRC = 通过反转所有计算出的 CRC 位，故意发送错误的通信（在传输响应期间）CRC。  
0 = 发送计算出的 CRC  
1 = 发送反转的 CRC

#### 6.5.4.11.3 DIAG\_PWR\_CTRL

地址	0x0337							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留						BIST_NO_RST	PWR_BIST_GO
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
BIST_NO_RST = 用于在电源 BIST 检测到故障时进行进一步诊断。当该位设置为 1 时，然后使用 [PWR_BIST_GO] 运行 BIST 周期，器件将不会清除 FAULT_PWR1 和 FAULT_PWR2 寄存器，并且不会在 BIST 周期结束时取消置位 NFAULT 信号。 0 = 在 LDO 比较器上循环执行 BIST。FAULT_PWR* 寄存器复位为 0，且 NFAULT 在每次 LDO BIST 运行结束时取消置位。 1 = 在 LDO 比较器上循环执行 BIST。FAULT_PWR* 寄存器不复位为 0，且 NFAULT 在每次 LDO BIST 运行结束时保持置位状态。								
PWR_BIST_GO = 当写入 1 时，电源 BIST 诊断将启动。[BIST_NO_RST] 中的任何更改均没有效果，直到该位再次写入 1。该位会自行清除。 0 = 就绪 1 = 启动电源 BIST 诊断。								

#### 6.5.4.11.4 DIAG\_CBFET\_CTRL1

地址	0x0338							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CBFET16 至 CBFET9							
复位	0	0	0	0	0	0	0	0
CBFET16 至 CBFET9 = 启用 CBFET 以进行 CBFET 诊断。仅当 [COMP_ADC_SEL2:0] = 0b100 时，才对该寄存器进行采样。 0 = CBFET 关闭 1 = CBFET 开启								

#### 6.5.4.11.5 DIAG\_CBFET\_CTRL2

地址	0x0339							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CBFET8 至 CBFET1							
复位	0	0	0	0	0	0	0	0
CBFET8 至 CBFET1 = 启用 CBFET 以进行 CBFET 诊断。仅当 [COMP_ADC_SEL2:0] = 0b100 时，才对该寄存器进行采样。 0 = CBFET 关闭 1 = CBFET 开启								

#### 6.5.4.11.6 DIAG\_COMP\_CTRL1

地址	0x033A							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	VCCB_THR[4:0]							
复位	0	0	0	0	0	0	0	0

VCCB\_THR[4:0] = 配置 VCELL 与 AUXCELL 差值。如果测量的差值小于此阈值，则 VCELL 与 AUXCELL 检查被视为通过。该阈值也适用于从主 ADC 到 AUX ADC 的汇流条比较。  
范围为 6mV 至 99mV，阶跃为 3mV

BB\_THR[2:0] = RSVD = 添加到 VCCB\_THR 设置的附加差值，在比较连接在汇流条上方的电芯（汇流条单独连接到 VC 通道）时，在 VCELL 与 AUXCELL 比较期间使用。  
范围为 5mV 至 40mV，阶跃为 5mV。保留

#### 6.5.4.11.7 DIAG\_COMP\_CTRL2

地址	0x033B							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	GPIO_THR[2:0]			OW_THR[3:0]			
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
GPIO_THR[2:0] = 配置主 ADC 测量和 AUX ADC 测量之间的 GPIO 比较差值阈值。 范围为 4mV 至 32mV，阶跃为 4mV								
OW_THR[3:0] = 配置用于诊断比较的 OW 检测阈值。该阈值适用于 CB OW 和 VC OW 诊断。 范围为 500mV 至 5V，阶跃为 300mV。								

#### 6.5.4.11.8 DIAG\_COMP\_CTRL3

地址	0x033C							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	CBFET_CTRL_GO	OW_SNK[1:0]		COMP_ADC_SEL[2:0]			COMP_ADC_GO
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
CBFET_CTRL_GO = 当该 GO 位 = 1 时，器件开启所配置的 CBFET，并关闭 DIAG_CBFET_CTRL1/2 寄存器中清除的任何 CBFET。仅当 CB 未运行或处于暂停状态时才会执行此 GO 操作，否则，CBFET 由常规 CB 控制进行控制。如果 CBFET 通过该 GO 位开启，一旦 CB 启动或恢复，CBFET 控制将返回到常规 CB 控制（而不是通过该 GO 位操作）								
OW_SNK[1:0] = 开启 VC 引脚、CB 引脚或 BBP/N 引脚上的电流阱。对这些位所做的更改将立即生效。主机 MCU 负责在执行开路 (OW) 测试之前开启正确的灌电流，并在 OW 测试完成后关闭灌电流。 00 = 所有 VC，BBP/N，CB 引脚灌电流关闭。 01 = 开启所有 VC 引脚上的灌电流 10 = 开启所有 CB 引脚上的灌电流 11 = 开启 BBP/N 引脚上的灌电流								

COMP\_ADC\_SEL[2:0] = 支持通过 ADC 测量进行器件诊断比较。在启用此诊断之前，主机以连续模式启用相应的 ADC。当 [COMP\_ADC\_GO] = 1 时，将对这些位进行采样。

000 = 不执行 ADC 比较  
001 = 电芯电压测量检查。  
器件将根据以下标准比较 [AUX\_CELL\_SEL4:0] 指定的电芯通道：  
VCELL (来自自主 ADC) 与 AUXCELL (来自 AUX ADC) 的差值小于 [VCCB\_THR4:0]。  
当比较完成时，[DRDY\_VCCB] = 1。  
010 = VC 引脚上的开路 (OW) 检查。  
在启用此比较之前，MCU 通过 [OW\_SNK1:0] 启用所有 VC 引脚上的电流阱。器件将根据以下标准比较由 ACTIVE\_CELL 寄存器指定的相应 VC 引脚：VCELL (来自自主 ADC) 小于 DIAG\_COMP\_CTRL2 [OW\_THR3:0]。  
比较完成时，[DRDY\_VC\_OW] = 1。  
011 = CB 引脚上的开路 (OW) 检查  
在启用此比较之前，MCU 通过 [OW\_SNK1:0] 启用所有 VC 引脚上的电流阱。器件将根据以下标准比较由 [AUX\_CELL\_SEL4:0] 指定的相应 CB 引脚：AUXCELL (来自 AUX ADC) 小于 DIAG\_COMP\_CTRL2 [OW\_THR3:0]。当比较完成时，[DRDY\_CBOW] = 1。  
100 = CBFET 检查。  
在开始此检查之前，MCU 预先配置以下内容：

- 如果启用了均衡，则暂停电芯均衡。
- 启用由 DIAG\_CBFET\_CTRL1/2 寄存器配置的 CBFET。
- 配置 [EXTD\_CBFET]，以便决定是所有 CBFET 都返回到暂停状态 (即关闭所有 CBFET)，还是保持由 DIAG\_CBFET\_CTRL1/2 寄存器指定的状态。

当该测试开始时，器件将开启由 DIAG\_CBFET\_CTRL1/2 寄存器指定的 CBFET，然后，根据以下标准比较由 [AUX\_CELL\_SEL4:0] 指定的通道：  
AUXCELL (来自 AUX ADC) < VCELL 的 1/3 (来自自主 ADC)。比较完成时，[DRDY\_CBFET] = 1。  
101 = GPIO 测量检查 (适用于配置为 ADC 和 OTUT 输入或仅 ADC 输入的 GPIO)。  
器件将主 GPIO 测量值与 AUX GPIO 测量值进行比较，差值小于 [GPIO\_THR2:0]。当比较完成时，[DRDY\_GPIO] = 1。  
其他代码：不进行 ADC 比较

COMP\_ADC\_GO = 器件启动由 [COMP\_ADC\_SEL2:0] 设置指定的诊断测试。当该位写入 1 时，将对所选的 [COMP\_ADC\_SEL2:0] 进行采样。除非再次将该 GO 位写入 1，否则更改 [COMP\_ADC\_SEL2:0] 设置不会产生任何影响。  
该位在读取时清零。  
0 = 就绪。写入 0 无效  
1 = [COMP\_ADC\_SEL2:0] 选择的星型诊断

#### 6.5.4.11.9 DIAG\_COMP\_CTRL4

地址	0x033D							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD						COMP_FAULT_INJ	LPF_FAULT_INJ
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
COMP_FAULT_INJ = 向 ADC 比较逻辑注入故障。如果在设置了该位的情况下运行任何 ADC 比较诊断，则比较结果预计会失败。 0 = 禁用 1 = 启用								
LPF_FAULT_INJ = 在 LPF 诊断期间向诊断 LPF 注入故障条件。预计会设置 FAULT_COMP_MISC[LPF_FAIL]。 0 = 禁用 1 = 启用								

#### 6.5.4.11.10 DIAG\_PROT\_CTRL

地址	0x033E							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留							PROT_BIST_NO_RST
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								

PROT\_BIST\_NO\_RST = 用于在保护器 BIST 检测到故障时进行进一步诊断。当该位设置为 1 时，器件将不会清除 *FAULT\_OV1/2*、*FAULT\_UV1/2*、*FAULT\_OT* 和 *FAULT\_UT* 寄存器。一旦 NFAULT 信号生效，该信号就会被锁存。  
 注意：主机应确保在该位设置为 0 的情况下启动 BIST 运行之前没有故障。  
 0 = 在 BIST 运行期间，器件会设置一个故障以检查保护器比较器和多路复用器，并为 NFAULT 引脚设置正确的 OV、UV、OT 和 UT 故障位。如果该位为 0，器件将清除故障并使 NFAULT 无效，然后切换到下一个通道。  
 1 = 在 BIST 运行期间，在切换到下一个电芯或 GPIO 通道之前，不会清除在测试期间产生的故障。一旦 NFAULT 引脚生效，该引脚就会被锁存。

#### 6.5.4.12 故障配置和复位

##### 6.5.4.12.1 FAULT\_MSK1

地址	0x0016							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	MSK_PROT	MSK_UT	MSK_OT	MSK_UV	MSK_OV	MSK_COMP	MSK_SYS	MSK_PWR
复位	0	0	0	0	0	0	0	0
MSK_PROT = 屏蔽 <i>FAULT_PROT*</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_PROT*</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_PROT*</i> 位状态如何，都不执行 NFAULT 操作。								
MSK_UT = 屏蔽 <i>FAULT_UT*</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_UT*</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_UT*</i> 位状态如何，都不执行 NFAULT 操作。								
MSK_OT = 屏蔽 <i>FAULT_OT*</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_OT*</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_OT*</i> 位状态如何，都不执行 NFAULT 操作。								
MSK_UV = 屏蔽 <i>FAULT_UV*</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_UV*</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_UV*</i> 位状态如何，都不执行 NFAULT 操作。								
MSK_OV = 屏蔽 <i>FAULT_OV*</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_OV*</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_OV*</i> 位状态如何，都不执行 NFAULT 操作。								
MSK_COMP = 屏蔽 <i>FAULT_COMP_*</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_COMP_*</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_COM_*</i> 位状态如何，都不执行 NFAULT 操作。								
MSK_SYS = 屏蔽 <i>FAULT_SYS</i> 寄存器的所有位，使其不会导致 NFAULT 生效。 0 = 如果 <i>FAULT_SYS</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_SYS</i> 位状态如何，都不执行 NFAULT 操作。								
MSK_PWR = 屏蔽 <i>FAULT_PWR1</i> 至 <i>FAULT_PWR3</i> 寄存器的所有位，使其不会导致 NFAULT 生效。 0 = 如果 <i>FAULT_PWR1</i> 至 <i>FAULT_PWR3</i> 中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_PWR1</i> 至 <i>FAULT_PWR3</i> 位状态如何，都不执行 NFAULT 操作。								

##### 6.5.4.12.2 FAULT\_MSK2

地址	0x0017							
NVM	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	SPARE[1]	MSK_OTP_CRC	MSK_OTP_DATA	MSK_COMM3_FCOMM	MSK_COMM3_FTONE	MSK_COMM3_HB	MSK_COMM2	MSK_COMM1
复位	0	0	0	0	0	0	0	0
SPARE[1] = 备用								
MSK_OTP_CRC = 屏蔽 <i>FAULT_OTP</i> 寄存器 ( 仅限 [ <i>CUST_CRC</i> ] 和 [ <i>FACT_CRC</i> ] )，使其不触发 NFAULT。 0 = 如果上述任何位设置为 1，则使 NFAULT 生效。 1 = 无论上述位的状态如何，都不执行 NFAULT 操作。								
MSK_OTP_DATA = 屏蔽 <i>FAULT_OTP</i> 寄存器 ( 除 [ <i>CUST_CRC</i> ] 和 [ <i>FACT_CRC</i> ] 以外的所有位 )，使其不触发 NFAULT。 0 = 如果上述任何位设置为 1，则使 NFAULT 生效。 1 = 无论上述位的状态如何，都不执行 NFAULT 操作。								

MSK_COMM3_FCOMM	屏蔽 <i>FAULT_COMM3[FCOMM_DET]</i> 故障，使其不触发 NFAULT。 = 0 = 如果 <i>FAULT_COMM3[FCOMM_DET]</i> 设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_COMM3[FCOMM_DET]</i> 状态如何，都不执行 NFAULT 操作。
MSK_COMM3_FTONE	屏蔽 <i>FAULT_COMM3[FTONE_DET]</i> 故障，使其不触发 NFAULT。 = 0 = 如果 <i>FAULT_COMM3[FTONE_DET]</i> 设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_COMM3[FTONE_DET]</i> 状态如何，都不执行 NFAULT 操作。
MSK_COMM3_HB	屏蔽 <i>FAULT_COMM3[HB_FAST]</i> 或 <i>[HB_FAIL]</i> 故障，使其不触发 NFAULT。 0 = 如果 <i>FAULT_COMM3[HB_FAST]</i> 或 <i>[HB_FAIL]</i> 设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_COMM3[HB_FAST]</i> 或 <i>[HB_FAIL]</i> 状态如何，都不执行 NFAULT 操作。
MSK_COMM2	屏蔽 <i>FAULT_COMM2</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_COMM2</i> 寄存器中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_COMM2</i> 寄存器位状态如何，都不执行 NFAULT 操作。
MSK_COMM1	屏蔽 <i>FAULT_COMM1</i> 寄存器，使其不触发 NFAULT。 0 = 如果 <i>FAULT_COMM1</i> 寄存器中的任何位设置为 1，则使 NFAULT 生效。 1 = 无论 <i>FAULT_COMM1</i> 寄存器位状态如何，都不执行 NFAULT 操作。

### 6.5.4.12.3 FAULT\_RST1

地址	0x0331							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RST_PROT	RST_UT	RST_OT	RST_UV	RST_OV	RST_COMP	RST_SYS	RST_PWR
复位	0	0	0	0	0	0	0	0
RST_PROT = 将 <i>FAULT_PROT1</i> 和 <i>FAULT_PROT2</i> 寄存器复位为 0x00。 0 = 无复位 1 = 将寄存器复位为 0x00								
RST_UT = 将所有 <i>FAULT_UT</i> 寄存器复位为 0x00。 0 = 无复位 1 = 将寄存器复位为 0x00								
RST_OT = 将所有 <i>FAULT_OT</i> 寄存器复位为 0x00。 0 = 无复位 1 = 将寄存器复位为 0x00								
RST_UV = 将所有 <i>FAULT_UV*</i> 寄存器复位为 0x00。 0 = 无复位 1 = 将寄存器复位为 0x00								
RST_OV = 将所有 <i>FAULT_OV*</i> 寄存器复位为 0x00。 0 = 无复位 1 = 将寄存器复位为 0x00								
RST_COMP = 将所有 <i>FAULT_COMP_*</i> 寄存器复位为 0x00。 0 = 无复位 1 = 将寄存器复位为 0x00								
RST_SYS = 将 <i>FAULT_SYS</i> 寄存器复位为 0x00。写入 1 后该位会自清零。 0 = 不复位 1 = 复位为 0x00								
RST_PWR = 将 <i>FAULT_PWR1</i> 至 <i>FAULT_PWR3</i> 寄存器复位为 0x00。写入 1 后该位会自清零。 0 = 不复位 1 = 复位为 0x00								

### 6.5.4.12.4 FAULT\_RST2

地址	0x0332							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	RST_OTP_CRC	RST_OTP_DATA	RST_COMM3_FCOMM	RST_COMM3_FTONE	RST_COMM3_HB	RST_COMM2	RST_COMM1
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								

<b>RST_OTP_CRC</b> = 复位 <i>FAULT_OTP</i> 寄存器 ( 仅限 [ <i>CUST_CRC</i> ] 和 [ <i>FACT_CRC</i> ] )。 0 = 无复位 1 = 将寄存器复位为 0x00
<b>RST_OTP_DATA</b> = 复位 <i>FAULT_OTP</i> 寄存器 ( 仅限 [ <i>SEC_DETECT</i> ] 和 [ <i>DED_DETECT</i> ] )。 0 = 无复位 1 = 将寄存器复位为 0x00
<b>RST_COMM3_FCOMM</b> 复位 <i>FAULT_COMM3[FCOMM_DET]</i> 。 = 0 = 无复位 1 = 将相关位复位为 0
<b>RST_COMM3_FTONE</b> 复位 <i>FAULT_COMM3[FTONE_DET]</i> 。 = 0 = 无复位 1 = 将相关位复位为 0
<b>RST_COMM3_HB</b> = 复位 <i>FAULT_COMM3[HB_FAST]</i> 和 [ <i>HB_FAIL</i> ] 位。 0 = 无复位 1 = 将相关位复位为 0
<b>RST_COMM2</b> = 复位 <i>FAULT_COMM2</i> 、 <i>DEBUG_COML*</i> 和 <i>DEBUG_COMM_COMH*</i> 寄存器。 0 = 无复位 1 = 将寄存器复位为 0x00
<b>RST_COMM1</b> = 复位 <i>FAULT_COMM1</i> 和 <i>DEBUG_COMM_UART*</i> 寄存器。 0 = 无复位 1 = 将寄存器复位为 0x00

### 6.5.4.13 故障状态

#### 6.5.4.13.1 FAULT\_SUMMARY

该寄存器是 NFAULT 的软版本。

地址	0x052D							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	FAULT_PROT	FAULT_COMP_ADC	FAULT_OTP	FAULT_COMM	FAULT_OTUT	FAULT_OVUV	FAULT_SYS	FAULT_PWR
复位	0	0	0	0	0	0	0	0
<b>FAULT_PROT</b> = 如果 [ <i>MSK_PROT</i> ] = 0 并且设置了任何 <i>FAULT_PROT1</i> 或 <i>FAULT_PROT2</i> 寄存器位, 则会设置该位。 0 = 无保护器 ( OVUV、OTUT 比较器 ) 故障。 1 = 检测到保护器故障								
<b>FAULT_COMP_ADC</b> = 如果 [ <i>MSK_COMP</i> ] = 0 并且设置了以下任何寄存器, 则会设置该位 : <ul style="list-style-type: none"> <li>• <i>FAULT_COMP_VCCB1/2</i></li> <li>• <i>FAULT_COMP_VCOW1/2</i></li> <li>• <i>FAULT_COMP_CBOW1/2</i></li> <li>• <i>FAULT_COMP_CBFET1/2</i></li> <li>• <i>FAULT_COMP_GPIO</i></li> <li>• <i>FAULT_COMP_MISC</i></li> </ul> 0 = 无 ADC 比较故障 ( 即没有设置任何 <i>FAULT_COMP_*</i> 寄存器 )。 1 = 检测到 ADC 比较故障。								
<b>FAULT_OTP</b> = 如果 [ <i>MSK_OTP</i> ] = 0 并且设置了任何 <i>FAULT_OTP</i> 寄存器位, 则会设置该位。 0 = 未检测到 OTP 相关故障或 OTP 故障被屏蔽。 1 = 检测到 OTP 相关故障。								

<p> <b>FAULT_COMM</b> = 如果满足以下任一条件，则会设置该位：           <ul style="list-style-type: none"> <li>• <math>[MSK\_COMM1] = 0</math> 并且设置了任何 <b>FAULT_COMM1</b> 寄存器位。</li> <li>• <math>[MSK\_COMM2] = 0</math> 并且设置了任何 <b>FAULT_COMM2</b> 寄存器位。</li> <li>• <math>[MSK\_COMM3\_HB] = 0</math> 并且设置了 <b>FAULT_COMM3[HB\_FAST]</b> 位或 <b>[HB\_FAIL]</b> 位。</li> <li>• <math>[MSK\_COMM3\_FTONE] = 0</math> 并且设置了 <b>FAULT_COMM3[FTONE\_DET]</b>。</li> <li>• <math>[MSK\_COMM3\_FCOMM] = 0</math> 并且设置了 <b>FAULT_COMM3[FCOMM\_DET]</b>。</li> </ul>           0 = 未检测到 UART、VIF 或 FTONE 故障，或者 UART、VIF 和 FTONE 故障被屏蔽。            1 = 检测到 UART、VIF 或 UT 故障。         </p>
<p> <b>FAULT_OTUT</b> = 如果满足以下任一条件，则会设置该位：           <ul style="list-style-type: none"> <li>• <math>[MSK\_OT] = 0</math> 并且设置了任一 <b>FAULT_OT1</b> 或 <b>FAULT_OT2</b> 位。</li> <li>• <math>[MSK\_UT] = 0</math> 并且设置了任一 <b>FAULT_UT1</b> 或 <b>FAULT_UT2</b> 位。</li> </ul>           0 = 未检测到 OT 或 UT 故障，或者 OT 和 UT 故障被屏蔽。            1 = 检测到 OT 或 UT 故障         </p>
<p> <b>FAULT_OVUV</b> = 如果满足以下任一条件，则会设置该位：           <ul style="list-style-type: none"> <li>• <math>[MSK\_OV] = 0</math> 并且设置了任一 <b>FAULT_OV1</b> 或 <b>FAULT_OV2</b> 位。</li> <li>• <math>[MSK\_UV] = 0</math> 并且设置了任一 <b>FAULT_UV1</b> 或 <b>FAULT_UV2</b> 位。</li> </ul>           0 = 未检测到 OV 或 UV 故障，或者 OV 和 UV 故障被屏蔽。            1 = 检测到 OV 或 UV 故障。         </p>
<p> <b>FAULT_SYS</b> = 如果 <math>[MSK\_SYS] = 0</math> 并且设置了任何 <b>FAULT_SYS</b> 寄存器位，则会设置该位。            0 = 未检测到系统相关故障或系统故障被屏蔽。            1 = 检测到系统相关故障。         </p>
<p> <b>FAULT_PWR</b> = 如果 <math>[MSK\_PWR] = 0</math> 并且设置了任何 <b>FAULT_PWR1</b> 至 <b>FAULT_PWR3</b> 寄存器位，则会设置该位。            0 = 未检测到电源轨相关故障或电源轨故障被屏蔽。            1 = 检测到电源轨相关故障。         </p>

#### 6.5.4.13.2 FAULT\_COMM1

地址	0x0530							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD			UART_TR	UART_RR	UART_RC	COMMCLR_DET	STOP_DET
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
<p> <b>UART_TR</b> = 指示在传输响应帧时检测到 UART 故障。有关故障信息的更多详细信息，请参阅 <b>DEBUG_UART_RR_TR</b> 寄存器。            0 = 无故障            1 = 故障         </p>								
<p> <b>UART_RR</b> = 指示在接收响应帧时检测到 UART 故障。有关故障信息的更多详细信息，请参阅 <b>DEBUG_UART_RR_TR</b> 寄存器。            0 = 无故障            1 = 故障         </p>								
<p> <b>UART_RC</b> = 指示在接收命令帧期间检测到 UART 故障。有关故障信息的更多详细信息，请参阅 <b>DEBUG_UART_RC</b> 寄存器。            0 = 无故障            1 = 故障         </p>								
<p> <b>COMMCLR_DET</b> = 检测到 UART 通信清除信号。在 ACTIVE 或 SLEEP 模式下检测到 SLEEPtoACTIVE ping 或在 ACTIVE 模式下检测到 WAKE 引脚也会设置该位。            0 = 无 UART 清除            1 = 检测到 UART 清除         </p>								
<p> <b>STOP_DET</b> = 指示接收到意外的停止条件。在 ACTIVE 模式下检测到 SLEEPtoACTIVE 信号也会设置该位。            0 = 无故障            1 = 故障         </p>								

### 6.5.4.13.3 FAULT\_COMM2

地址	0x0531							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COML_TR	COML_RR	COML_RC	COML_BIT	COMH_TR	COMH_RR	COMH_RC	COMH_BIT
复位	0	0	0	0	0	0	0	0
COML_TR = 指示发送响应帧时检测到 COML 字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COML_RR_TR</i> 寄存器。 0 = 无故障 1 = 故障								
COML_RR = 指示接收响应帧时检测到 COML 字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COML_RR_TR</i> 寄存器。 0 = 无故障 1 = 故障								
COML_RC = 指示接收命令帧时检测到 COML 字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COML_RR_RC</i> 寄存器。 0 = 无故障 1 = 故障								
COML_BIT = 指示检测到 COML 位级故障，这将导致至少一个字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COML_BIT</i> 寄存器。 0 = 无故障 1 = 故障								
COMH_TR = 指示发送响应帧时检测到 COMH 字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COMH_RR_TR</i> 寄存器。 0 = 无故障 1 = 故障								
COMH_RR = 指示接收响应帧时检测到 COMH 字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COMH_RR_TR</i> 寄存器。 0 = 无故障 1 = 故障								
COMH_RC = 指示接收命令帧时检测到 COMH 字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COMH_RR_RC</i> 寄存器。 0 = 无故障 1 = 故障								
COMH_BIT = 指示检测到 COMH 位级故障，这将导致至少一个字节级故障。有关故障信息的更多详细信息，请参阅 <i>DEBUG_COMH_BIT</i> 寄存器。 0 = 无故障 1 = 故障								

### 6.5.4.13.4 FAULT\_COMM3

地址	0x0532							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD				FCOMM_DET	FTONE_DET	HB_FAIL	HB_FAST
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
FCOMM_DET = 已接收到由任何上部堆栈器件设置故障状态位的通信事务。 0 = 已清除故障状态，指示未从任何上部堆栈器件中检测到故障。 1 = 已通过接收通信事务设置故障状态。								
FTONE_DET = 指示接收到故障音调。如果 <i>[DIR_SEL] = 0</i> ，则检测在监视 COML 侧，反之亦然。 0 = 未检测到故障音调 1 = 检测到故障音调								
HB_FAIL = 指示在预期时间内未接收到检测信号。如果 <i>[DIR_SEL] = 0</i> ，则检测在监视 COML 侧，反之亦然。 0 = 无故障 1 = 故障								

HB\_FAST = 指示过于频繁地接收到检测信号。如果 [DIR\_SEL] = 0，则检测在监视 COML 侧，反之亦然。当 [FTONE\_DET] = 1 时，也可能设置该位，具体取决于多久之后从前一个检测信号检测到故障音调  
0 = 无故障  
1 = 故障

#### 6.5.4.13.5 FAULT\_OTP

地址	0x0535							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD	DED_DET	SEC_DET	CUST_CRC	FACT_CRC	CUSTLDERR	FACTLDERR	GBLOVERR
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
DED_DET = 指示在 OTP 加载期间发生 DED 错误。(编码期间未知) 0 = 无故障 1 = 故障								
SEC_DET = 指示在 OTP 加载期间发生 SEC 错误。(编码期间未知) 0 = 无故障 1 = 故障								
CUST_CRC = 指示在客户寄存器空间中发生 CRC 错误。 0 = 无故障 1 = 故障								
FACT_CRC = 指示在出厂寄存器空间中发生 CRC 错误。 0 = 无故障 1 = 故障								
CUSTLDERR = 指示客户空间 OTP 加载过程中的错误。读取 OTP_CUST1_STAT 和 OTP_CUST2_STAT 寄存器以获知具体的错误情况。如果满足以下任一条件，则会设置该错误位： <ul style="list-style-type: none"> <li>• 未对客户 OTP 页面进行编程。</li> <li>• 最高客户 OTP 页面具有一个 [FMterr]。</li> <li>• 最高客户 OTP 页面具有 [TRY] = 1，不是 [PROGOK]。</li> <li>• 在选定的客户 OTP 页面上发生 LOADERR。</li> </ul> 从存在此错误的器件收到的信息不得视为可靠的信息。 写入 [RST_OTP_DATA] = 1 不会重置该位。要重新检查该错误，需要执行器件复位或 HW_RESET。 0 = 无故障 1 = 故障								
FACTLDERR = 指示出厂空间 OTP 加载过程中的错误。如果满足以下任一条件，则会设置该错误位： <ul style="list-style-type: none"> <li>• 未对出厂 OTP 页面进行编程。</li> <li>• 最高出厂 OTP 页面具有一个 [FMterr]。</li> <li>• 最高出厂 OTP 页面具有 [TRY] = 1，不是 [PROGOK]。</li> <li>• 在选定的出厂 OTP 页面上发生 LOADERR。</li> </ul> 从存在此错误的器件收到的信息不得视为可靠的信息。 写入 [RST_OTP_DATA] = 1 不会重置该位。要重新检查该错误，需要执行器件复位或 HW_RESET。 0 = 无故障 1 = 故障								
GBLOVERR = 指示在一个 OTP 页面上检测到一个过压错误。读取 OTP_CUST1_STAT 和 OTP_CUST2_STAT 寄存器以确定具体的页面。从存在此错误的器件收到的信息不得视为可靠的信息。 写入 [RST_OTP_DATA] = 1 不会重置该位。要清除该位，需要执行器件复位或 HW_RESET。在另一个页面 (如果可用) 上重复编程过程会强制器件重新评估条件。 0 = 无故障 1 = 故障								

#### 6.5.4.13.6 FAULT\_SYS

地址	0x0536							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0

名称	LFO	RSVD	GPIO	DRST	CTL	CTS	TSHUT	TWARN
复位	0	0	0	0	0	0	0	0
LFO = 指示 LFO 频率超出预期范围 0 = 未检测到故障 1 = 检测到故障								
RSVD = 被保留								
GPIO = 当 $GPIO\_CONF1[FAULT\_IN\_EN] = 1$ 时指示 GPIO8 检测到故障输入。 0 = 未检测到故障 1 = 检测到故障输入								
DRST = 指示发生了数字复位。 0 = 无数字复位 1 = 发生数字复位								
CTL = 指示发生了长通信超时。器件操作由 $[CTL\_ACT]$ 进行配置。如果操作被设置为器件关断，则无法观察到该位。 0 = 无故障 1 = 发生长通信超时。在长超时操作被设置为 SLEEP 时可以观察到。								
CTS = 指示发生了短通信超时。器件无操作。可以在达到长通信超时之前针对该情况向系统发出警报。 0 = 无故障 1 = 发生短通信超时								
TSHUT = 指示上次关断是热关断，其中内核温度（内核温度 2）高于热关断阈值。 0 = 内核温度低于热关断阈值 1 = 上次关断是热关断								
TWARN = 指示内核温度（内核温度 2）高于 $TWARN\_THR[1:0]$ 设置。器件目前未执行任何操作。这用作指示内核温度接近热关断的警告信号。 0 = 内核温度低于 $TWARN\_THR[1]$ 1 = 内核温度高于 $TWARN\_THR[1:0]$								

#### 6.5.4.13.7 FAULT\_PROT1

地址	0x053A							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD						TPARITY_	VPARITY_
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
TPARITY_FAIL = 指示在以下任何 OTUT 相关配置上检测到奇偶校验故障： <ul style="list-style-type: none"> <li>OT 或 UT 阈值设置</li> <li><math>[OTUT\_MODE1:0]</math> 设置</li> <li><math>GPIO\_CONF1...4</math> 设置</li> </ul> 0 = 无故障 1 = 故障								
VPARITY_FAIL = 指示在以下任何 OVUV 相关配置上检测到奇偶校验故障： <ul style="list-style-type: none"> <li>OV 或 UV 阈值设置</li> <li><math>[OVUV\_MODE1:0]</math> 设置</li> <li><math>[NUM\_CELL3:0]</math> 设置</li> </ul> 0 = 无故障 1 = 故障								

#### 6.5.4.13.8 FAULT\_PROT2

地址	0x053B							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD	BIST_ABORT	TPATH_FAIL	VPATH_FAIL	UTCOMP_	OTCOMP_	OVCOMP_	UVCOMP_
					FAIL	FAIL	FAIL	FAIL

复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
BIST_ABORT = 指示 OVUV 或 OTUT BIST 运行中止。 0 = BIST 运行至完成 1 = BIST 中止								
TPATH_FAIL = 指示在 BIST 测试期间沿 OTUT 信号路径检测到故障。 0 = 无故障 1 = 故障								
VPATH_FAIL = 指示在 BIST 测试期间沿 OVUV 信号路径检测到故障。 0 = 无故障 1 = 故障								
UTCOMP_FAIL = 指示在 BIST 测试期间 UT 比较器发生故障。 0 = 无故障 1 = 故障								
OTCOMP_FAIL = 指示在 BIST 测试期间 OT 比较器发生故障。 0 = 无故障 1 = 故障								
OVCOMP_FAIL = 指示在 BIST 测试期间 OV 比较器发生故障。 0 = 无故障 1 = 故障								
UVCOMP_FAIL = 指示在 BIST 测试期间 UV 比较器发生故障。 0 = 无故障 1 = 故障								

#### 6.5.4.13.9 FAULT\_OV1

地址	0x053C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	OV16_DET	OV15_DET	OV14_DET	OV13_DET	OV12_DET	OV11_DET	OV10_DET	OV9_DET
复位	0	0	0	0	0	0	0	0
OV9_DET 至 OV16_DET = Cell9 至 Cell16 的 OV 故障状态，结果来自 OV 比较器检测。								

#### 6.5.4.13.10 FAULT\_OV2

地址	0x053D							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	OV8_DET	OV7_DET	OV6_DET	OV5_DET	OV4_DET	OV3_DET	OV2_DET	OV1_DET
复位	0	0	0	0	0	0	0	0
OV1_DET 至 OV8_DET = Cell1 至 Cell8 的 OV 故障状态，结果来自 OV 比较器检测。								

#### 6.5.4.13.11 FAULT\_UV1

地址	0x053E							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	UV16_DET	UV15_DET	UV14_DET	UV13_DET	UV12_DET	UV11_DET	UV10_DET	UV9_DET
复位	0	0	0	0	0	0	0	0
UV9_DET 至 UV16_DET = Cell9 至 Cell16 的 UV 故障状态，结果来自 UV 比较器检测。								

#### 6.5.4.13.12 FAULT\_UV2

地址	0x053F							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0

名称	UV8_DET	UV7_DET	UV6_DET	UV5_DET	UV4_DET	UV3_DET	UV2_DET	UV1_DET
复位	0	0	0	0	0	0	0	0

UV1\_DET 至 UV8\_DET = Cell1 至 Cell8 的 UV 故障状态，结果来自 UV 比较器检测。

#### 6.5.4.13.13 FAULT\_OT

地址	0x0540							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	OT8_DET	OT7_DET	OT6_DET	OT5_DET	OT4_DET	OT3_DET	OT2_DET	OT1_DET
复位	0	0	0	0	0	0	0	0

OT1\_DET 至 OT8\_DET = GPIO1 至 GPIO8 的 OT 故障状态，结果来自 OT 比较器检测。

#### 6.5.4.13.14 FAULT\_UT

地址	0x0541							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	UT8_DET	UT7_DET	UT6_DET	UT5_DET	UT4_DET	UT3_DET	UT2_DET	UT1_DET
复位	0	0	0	0	0	0	0	0

UT1\_DET 至 UT8\_DET = GPIO1 至 GPIO8 的 UT 故障状态，结果来自 UT 比较器检测。

#### 6.5.4.13.15 FAULT\_COMP\_GPIO

地址	0x0543							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	GPIO8_FAIL	GPIO7_FAIL	GPIO6_FAIL	GPIO5_FAIL	GPIO4_FAIL	GPIO3_FAIL	GPIO2_FAIL	GPIO1_FAIL
复位	0	0	0	0	0	0	0	0

GPIO1\_FAIL 至 指示 GPIO1 至 GPIO8 的 ADC 与 AUX ADC GPIO 测量诊断结果。  
GPIO8\_FAIL = 0 = 诊断通过  
1 = 诊断失败。来自主 ADC 的 GPIO 与 AUX ADC 测量值大于 [GPIO\_THR2:0]

#### 6.5.4.13.16 FAULT\_COMP\_VCCB1

地址	0x0545							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL16_FAIL	CELL15_FAIL	CELL14_FAIL	CELL13_FAIL	CELL12_FAIL	CELL11_FAIL	CELL10_FAIL	CELL9_FAIL
复位	0	0	0	0	0	0	0	0

CELL9\_FAIL 至 指示 cell9 至 cell16 的电压诊断结果。  
CELL16\_FAIL = 0 = 诊断通过  
1 = 诊断失败。VCELL 与 AUXCELL 测量值大于 [VCCB\_THR4:0]

#### 6.5.4.13.17 FAULT\_COMP\_VCCB2

地址	0x0546							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CELL8_FAIL	CELL7_FAIL	CELL6_FAIL	CELL5_FAIL	CELL4_FAIL	CELL3_FAIL	CELL2_FAIL	CELL1_FAIL
复位	0	0	0	0	0	0	0	0

CELL1\_FAIL 至 指示 cell1 至 cell8 的电压诊断结果。  
CELL8\_FAIL = 0 = 诊断通过  
1 = 诊断失败。VCELL 与 AUXCELL 测量值大于 [VCCB\_THR4:0]

#### 6.5.4.13.18 FAULT\_COMP\_VCOW1

地址	0x0548							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	VCOW16_FAIL	VCOW15_FAIL	VCOW14_FAIL	VCOW13_FAIL	VCOW12_FAIL	VCOW11_FAIL	VCOW10_FAIL	VCOW9_FAIL
复位	0	0	0	0	0	0	0	0
VCOW9_FAIL 至 指示 cell9 至 cell 16 的 VC OW 诊断结果。 VCOW16_FAIL = 0 = 诊断通过 1 = 诊断失败。VCELL 测量值小于 [OW_THR3:0]								

#### 6.5.4.13.19 FAULT\_COMP\_VCOW2

地址	0x0549							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	VCOW8_FAIL	VCOW7_FAIL	VCOW6_FAIL	VCOW5_FAIL	VCOW4_FAIL	VCOW3_FAIL	VCOW2_FAIL	VCOW1_FAIL
复位	0	0	0	0	0	0	0	0
VCOW1_FAIL 至 指示 cell1 至 cell8 的 VC OW 诊断结果。 VCOW8_FAIL = 0 = 诊断通过 1 = 诊断失败。VCELL 测量值小于 [OW_THR3:0]								

#### 6.5.4.13.20 FAULT\_COMP\_CBOW1

地址	0x054B							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CBOW16_FAIL	CBOW15_FAIL	CBOW14_FAIL	CBOW13_FAIL	CBOW12_FAIL	CBOW11_FAIL	CBOW10_FAIL	CBOW9_FAIL
复位	0	0	0	0	0	0	0	0
CBOW9_FAIL 至 CB FET9 至 CB FET16 的 CB OW 诊断结果。 CBOW16_FAIL = 0 = 通过 1 = 失败								

#### 6.5.4.13.21 FAULT\_COMP\_CBOW2

地址	0x054C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CBOW8_FAIL	CBOW7_FAIL	CBOW6_FAIL	CBOW5_FAIL	CBOW4_FAIL	CBOW3_FAIL	CBOW2_FAIL	CBOW1_FAIL
复位	0	0	0	0	0	0	0	0
CBOW1_FAIL 至 CB FET1 至 CB FET8 的 CB OW 诊断结果。 CBOW8_FAIL = 0 = 通过 1 = 失败								

#### 6.5.4.13.22 FAULT\_COMP\_CBFET1

地址	0x054E							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CBFET16_FAIL	CBFET15_FAIL	CBFET14_FAIL	CBFET13_FAIL	CBFET12_FAIL	CBFET11_FAIL	CBFET10_FAIL	CBFET9_FAIL
复位	0	0	0	0	0	0	0	0
CBFET9_FAIL 至 CB FET9 至 CB FET16 的 CB FET 诊断结果。 CBFET16_FAIL = 0 = 通过 1 = 失败								

#### 6.5.4.13.23 FAULT\_COMP\_CBFET2

地址	0x054F							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CBFET8_FAIL	CBFET7_FAIL	CBFET6_FAIL	CBFET5_FAIL	CBFET4_FAIL	CBFET3_FAIL	CBFET2_FAIL	CBFET1_FAIL
复位	0	0	0	0	0	0	0	0
CBFET1_FAIL 至 CBFET1 至 CBFET8 的 CBFET 诊断结果。 CBFET8_FAIL = 0 = 通过 1 = 失败								

#### 6.5.4.13.24 FAULT\_COMP\_MISC

地址	0x0550							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD						COMP_ADC_ABORT	LPF_FAIL
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
COMP_ADC_ABORT = 指示最近的 ADC 比较诊断由于设置不当而中止。仅当 ADC 比较诊断之一已启动时才有效。 0 = ADC 比较诊断运行至完成 1 = ADC 比较诊断中止								
LPF_FAIL = 指示 LPF 诊断结果。 0 = 通过 1 = 失败								

#### 6.5.4.13.25 FAULT\_PWR1

地址	0x0552							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CVSS_OPEN	DVSS_OPEN	REFHM_OPEN	CVDD_UV	CVDD_OV	DVDD_OV	AVDD_OSC	AVDD_OV
复位	0	0	0	0	0	0	0	0
CVSS_OPEN = 指示 CVSS 引脚上的开路情况。 0 = 无故障 1 = 故障								
DVSS_OPEN = 指示 DVSS 引脚上的开路情况。 0 = 无故障 1 = 故障								
REFHM_OPEN = 指示 REFHM 引脚上的开路情况。 0 = 无故障 1 = 故障								
CVDD_UV = 指示 CVDD LDO 上的欠压故障。 0 = 无故障 1 = 故障								
CVDD_OV = 指示 CVDD LDO 上的过压故障。 0 = 无故障 1 = 故障								
DVDD_OV = 指示 DVDD LDO 上的过压故障。 0 = 无故障 1 = 故障								
AVDD_OSC = 指示 AVDD 振荡超出可接受的限制。 0 = 无故障 1 = 故障 当从睡眠模式转换到活动模式时，可能会触发此故障。因此，如果设置了此故障，请忽略它并复位该故障。								

AVDD\_OV = 指示 AVDD LDO 上的过压故障。  
0 = 无故障  
1 = 故障

#### 6.5.4.13.26 FAULT\_PWR2

地址	0x0553							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留	PWRBIST_FAIL	被保留	REFH_OSC	NEG5V_UV	TSREF_OSC	TSREF_UV	TSREF_OV
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
PWRBIST_FAIL = 指示电源 BIST 运行出现故障。 0 = 无故障 1 = 故障								
REFH_OSC = 指示 REGH 基准振荡超出可接受的限制。 0 = 无故障 1 = 故障								
NEG5V_UV = 指示 NEG5V 电荷泵上存在欠压故障。 0 = 无故障 1 = 故障								
TSREF_OSC = 指示 TSREF 振荡超出可接受的限制。 0 = 无故障 1 = 故障								
TSREF_UV = 指示 TSREF LDO 上存在欠压故障。 0 = 无故障 1 = 故障								
TSREF_OV = 指示 TSREF LDO 上存在过压故障。 0 = 无故障 1 = 故障								

#### 6.5.4.13.27 FAULT\_PWR3

地址	0x0554							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD					RSVD	RSVD	AVDDUV_DRST
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
AVDDUV_DRST = 指示由于检测到 AVDD UV 而发生了数字复位。当器件在关闭或硬件复位事件后唤醒时，这也适用。 0 = 无复位 1 = 由于 AVDD UV 而发生了数字复位								

### 6.5.4.14 调试控制和状态

#### 6.5.4.14.1 DEBUG\_CTRL\_UNLOCK

地址	0x0700							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CODE[7:0]							
复位	0	0	0	0	0	0	0	0
CODE[7:0] = 将解锁代码 (0xA5) 写入该寄存器以激活 <i>DEBUG_COMM_CTRL</i> * 寄存器中的设置。 解锁代码以外的任何其他值都将停用 <i>DEBUG_COMM_CTRL</i> * 设置中的任何效果，并返回到器件的正常设置。								

#### 6.5.4.14.2 DEBUG\_COMM\_CTRL1

地址	0x0701							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD			UART_BAUD	UART_MIRROR_EN	UART_TX_EN	USER_UART_EN	USER_DAISSY_EN
复位	0	0	0	0	0	1	0	0
RSVD = 被保留								
UART_BAUD = 该位将 UART 波特率更改为 250kb/s。对 VIF 调试很有用。当系统将所有菊花链器件设置为 250kb/s 波特率时，它会减慢通过 VIF 的响应字节，从而提高 VIF 的稳健性以进行调试。 0 = 默认值 1Mb/s 1 = 250kb/s								
UART_MIRROR_EN = 该位使堆栈 VIF 通信能够镜像到 UART。要使用该调试功能，必须首先通过设置 [UARTTX_EN] = 1 来启用堆栈器件的 UART TX。 0 = 禁用 1 = 启用								
UART_TX_EN = 默认情况下，堆栈器件禁用 UART TX。该位使 UART TX 能够允许通过堆栈器件上的 UART 进行读取/写入。 0 = 禁用 1 = 启用								
USER_UART_EN = 该位启用调试 UART 控制位 [UART_TX_EN] 和 [UART_MIRROR_EN]。 0 = 上述位的设置无效。 1 = 器件根据 [UART_TX_EN] 和 [UART_MIRROR_EN] 设置配置 UART								
USER_DAISSY_EN = 该位启用 DEBUG_COMM_CTRL2 寄存器中的调试 COML 和 COMH 控制位 0 = DEBUG_COMM_CTRL2 寄存器的设置无效。 1 = 器件根据 DEBUG_COMM_CTRL2 寄存器设置配置 COML 和 COMH。								

#### 6.5.4.14.3 DEBUG\_COMM\_CTRL2

地址	0x0702							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留				COML_TX_EN	COML_RX_EN	COMH_TX_EN	COMH_RX_EN
复位	0	0	0	0	1	1	1	1
RSVD = 被保留								
COML_TX_EN = 启用 COML 发送器。 0 = 禁用 1 = 启用								
COML_RX_EN = 启用 COML 接收器。 0 = 禁用 1 = 启用								
COMH_TX_EN = 启用 COMH 发送器。 0 = 禁用 1 = 启用								
COMH_RX_EN = 启用 COMH 接收器。 0 = 禁用 1 = 启用								

#### 6.5.4.14.4 DEBUG\_COMM\_STAT

地址	0x0780							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD		HW_UART_DRV	HW_DAISSY_DRV	COML_TX_ON	COML_RX_ON	COMH_TX_ON	COMH_RX_ON
复位 (基底)	0	0	1	1	0	0	1	1

复位 (堆栈)	0	0	1	1	1	1	1	1
RSVD = 被保留								
<p>HW_UART_DRV = 指示 UART TX 是由器件本身控制还是由 MCU 控制。适用于器件配置为“STACK”后默认禁用 UART TX 的堆栈器件。            0 = <i>DEBUG_COMM_CTRL1</i>[USER_UART_EN] = 1。通过 <i>DEBUG_COMM_CTRL2</i> 寄存器对 UART TX 进行手动控制。            1 = UART TX 由器件控制</p>								
<p>HW_DAI5Y_DRV = 指示 COML 和 COMH 是由器件本身控制还是由 MCU 控制。            0 = <i>DEBUG_COMM_CTRL1</i>[USER_DAI5Y_EN] = 1。通过 <i>DEBUG_COMM_CTRL2</i> 寄存器对 COML 和 COMH 进行手动控制。            1 = COML 和 COMH 由器件控制</p>								
<p>COML_TX_ON = 显示当前 COML 发送器状态。            0 = 关闭            1 = 开启</p>								
<p>COML_RX_ON = 显示当前 COML 接收器状态。            0 = 关闭            1 = 开启</p>								
<p>COMH_TX_ON = 显示当前 COMH 发送器状态。            0 = 关闭            1 = 开启</p>								
<p>COMH_RX_ON = 显示当前 COMH 接收器状态。            0 = 关闭            1 = 开启</p>								

#### 6.5.4.14.5 DEBUG\_UART\_RC

地址	0x0781							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留		RC_IERR	RC_TXDIS	RC_SOF	RC_BYTE_ERR	RC_UNEXP	RC_CRC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
<p>RC_IERR = 检测到所接收命令帧中的初始化学字节错误。这可能是由于帧初始化学字节有停止错误、设置了不正确的帧类型或设置了保留命令类型位。在接收到通信 CLEAR 之前，后面的所有字节都将被忽略。            当忽略通信帧时，器件不会尝试检测被忽略帧中的任何通信错误，也不会将帧计数器中将其计为有效/丢弃。            0 = 无错误            1 = 检测到错误</p>								
<p>RC_TXDIS = 检测到 UART TX 已禁用，但主机 MCU 已发出从器件读取数据的命令。            0 = 无错误            1 = 检测到错误</p>								
<p>RC_SOF = 检测到帧起始 (SOF) 错误。也就是说，在当前帧完成之前，在 UART 上会接收到 UART CLEAR。            0 = 无错误            1 = 检测到错误</p>								
<p>RC_BYTE_ERR = 在接收到的命令帧中检测到除初始化学字节错误以外的任何字节错误。在接收到通信 CLEAR 之前，后面的所有字节都将被忽略。            当忽略通信帧时，器件不会尝试检测被忽略帧中的任何通信错误，也不会将帧计数器中将其计为有效/丢弃。            0 = 无错误            1 = 检测到错误</p>								
<p>RC_UNEXP = 在堆栈器件中 (即 [<i>STACK_DEV</i>] = 1 且 [<i>MULTIDROP</i>] = 0)，预计不会通过 UART 接口接收堆栈或广播命令。如果是，则将其检测为错误并设置该位。            如果器件配置为 [<i>MULTIDROP</i>] = 1，则不会设置该位。            0 = 无错误            1 = 检测到错误</p>								

RC\_CRC = 检测到从 UART 接收的命令帧中的 CRC 错误。该帧将被视为丢弃帧。  
0 = 无错误  
1 = 检测到错误

#### 6.5.4.14.6 DEBUG\_UART\_RR\_TR

地址	0x0782							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留			TR_SOF	TR_WAIT	RR_SOF	RR_BYTE_ERR	RR_CRC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
TR_SOF = 指示在器件仍在传输数据时接收到 UART CLEAR。 0 = 无错误 1 = 检测到错误								
TR_WAIT = 器件正在等待传输响应，但由于以下任一原因，此操作被终止： <ul style="list-style-type: none"> <li>该器件接收到 UART CLEAR 信号。</li> <li>该器件接收到新命令。</li> </ul> 当发出广播或堆栈读取命令时，该位有效。 0 = 无错误 1 = 检测到错误								
RR_SOF = 指示在接收响应帧时收到 UART CLEAR。UART 上的响应帧仅适用于多点模式。 0 = 无错误 1 = 检测到错误								
RR_BYTE_ERR = 在接收到的响应帧中检测到除初始化学字节错误以外的任何字节错误。在接收到通信 CLEAR 之前，后面的所有字节都将被忽略。 当忽略通信帧时，器件不会尝试检测被忽略帧中的任何通信错误，也不会将帧计数器中将其计为有效/丢弃。 0 = 无错误 1 = 检测到错误								
RR_CRC = 检测到从 UART 接收到的响应帧中存在 CRC 错误。该帧将被视为丢弃帧。 0 = 无错误 1 = 检测到错误								

#### 6.5.4.14.7 DEBUG\_COMH\_BIT

地址	0x0783							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD			PERR	BERR_TAG	Sync2	Sync1	位
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
PERR = 检测到传入通信帧异常，因此器件将转发设置了 [BERR] 位的通信帧。该寄存器中设置的任何错误位也会设置 [PERR] 位。然而，该寄存器中未分类的异常也可能触发 [PERR] 位（例如，检测到数据丢失或数据顺序错误）。 0 = 未检测到通信错误，转发的通信帧未插入 [BERR]。 1 = 检测到接收的通信帧异常。[BERR] 对转发的通信生效。								
BERR_TAG = 当接收到的通信带有 [BERR] = 1 标记时设置。 0 = 接收到的通信帧无 [BERR] 1 = 接收到的通信帧有 [BERR]								
SYNC2 = 检测到前导码半位和 [SYNC1:0] 位。器件正在使用从这些位中提取的时序信息，但无法检测到有效数据。 0 = 无错误 1 = 检测到错误								
SYNC1 = 无法检测到前导码半位或任何 [SYNC1:0] 位。这可能是由于位丢失或信号严重失真而无法检测到。 0 = 无错误 1 = 检测到错误								

BIT = 器件已经检测到一个数据位；但是，检测样本不足以确保强 1 或 0。  
0 = 无错误  
1 = 检测到错误

#### 6.5.4.14.8 DEBUG\_COMH\_RC

地址	0x0784							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD		RC_IERR	RC_TXDIS	RC_SOF	RC_BYTE_ERR	RC_UNEXP	RC_CRC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
<p>RC_IERR = 检测到所接收命令帧中的初始化字节错误。这可能是由于字节格式不正确造成的，例如，需要帧初始化字节，但未设置帧起始 (SOF) 位，或者选择了无效的帧类型。由于在 COMH/COML 上接收到的字节会在堆栈中向上传播，因此上层堆栈中的器件很可能也会检测到此错误。后面的所有字节都将被忽略，直至收到 SOF 位。当忽略通信帧时，器件不会尝试检测被忽略帧中的任何通信错误，也不会将帧计数器中将其计为有效/丢弃。</p> <p>0 = 无错误 1 = 检测到错误</p>								
<p>RC_TXDIS = 当 [DIR_SEL] = 1 时有效。器件检测到 COMH TX 已禁用，但器件接收到读取数据的命令 (即，将数据传出去)。命令帧将计为丢弃帧。</p> <p>0 = 无错误 1 = 检测到错误</p>								
<p>RC_SOF = 当 [DIR_SEL] = 1 时有效。在 COMH 上检测到帧起始 (SOF) 错误。SOF 位仅应在初始化帧中设置，但 SOF 位在当前帧中设置，这不符合预期。</p> <p>0 = 无错误 1 = 检测到错误</p>								
<p>RC_BYTE_ERR = 当 [DIR_SEL] = 1 时有效。在接收到的命令帧中检测到除初始化字节错误以外的任何字节错误。此错误可能会触发在 DEBUG_COMMH_BIT 寄存器中设置的一个或多个错误位。</p> <p>0 = 无错误 1 = 检测到错误</p>								
<p>RC_UNEXP = 如果 [DIR_SEL] = 0，但器件从 COMH 接收命令帧 (这是无效条件)，器件将设置该错误位。</p> <p>0 = 无错误 1 = 检测到错误</p>								
<p>RC_CRC = 指示导致一个或多个 COMH 命令帧被丢弃的 CRC 错误。由于帧被丢弃，因此不会指示该帧中的任何其他错误。</p> <p>0 = 无错误 1 = 检测到错误</p>								

#### 6.5.4.14.9 DEBUG\_COMH\_RR\_TR

地址	0x0785							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD		TR_WAIT	RR_TXDIS	RR_SOF	RR_BYTE_ERR	RR_UNEXP	RR_CRC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
<p>TR_WAIT = 器件正在等待传输响应，但由于器件收到新命令，因此该操作被终止。当发出广播或堆栈读取命令时，该位有效。</p> <p>0 = 无错误 1 = 检测到错误</p>								
<p>RR_TXDIS = 当 [DIR_SEL] = 0 时有效，此时器件接收到响应，但由于 COMH TX 被禁用，无法发送到下一个器件。该帧计为丢弃帧。</p> <p>0 = 无错误 1 = 检测到错误</p>								

RR_SOF = 当 $[DIR\_SEL] = 0$ 时有效。在 COMH 上检测到帧起始 (SOF) 错误。SOF 位仅应在初始化帧中设置，但 SOF 位在当前帧中设置，这不符合预期。 0 = 无错误 1 = 检测到错误
RR_BYTE_ERR = 当 $[DIR\_SEL] = 0$ 时有效。在接收到的响应帧中检测到除初始化学节错误以外的任何字节错误。此错误可能会触发 $DEBUG\_COMMH\_BIT$ 寄存器中设置的一个或多个错误位。 0 = 无错误 1 = 检测到错误
RR_UNEXP = 如果 $[DIR\_SEL] = 1$ ，但器件从 COMH 接收到响应帧（这是无效条件），器件会设置该错误位。 0 = 无错误 1 = 检测到错误
RR_CRC = 指示导致一个或多个 COMH 响应帧被丢弃的 CRC 错误。由于帧被丢弃，因此不会指示该帧中的大多数其他错误。如果在 CRC 的最后一个字节上观察到 $[RR\_BYTE\_ERR]$ ，则会同时指示 CRC 和 BERR。 0 = 无错误 1 = 检测到错误

#### 6.5.4.14.10 DEBUG\_COML\_BIT

地址	0x0786							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD			PERR	BERR_TAG	Sync2	Sync1	位
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
PERR = 检测到传入通信帧异常，并且将对传出通信帧设置 BERR。该寄存器中设置的任何错误位也会设置 $[PERR]$ 位。然而，该寄存器中未分类的异常也可能触发 $[PERR]$ 位（例如，检测到数据丢失或数据顺序错误）。 0 = 未检测到通信错误，转发的通信帧未插入 $[BERR]$ 1 = 检测到接收的通信帧异常。BERR 对转发的通信生效。								
BERR_TAG = 当接收到的通信带有标记时设置。 0 = 接收到的通信帧无 BERR 1 = 接收到的通信帧有 BERR								
SYNC2 = 检测到前导码半位和 $[SYNC1:0]$ 位。器件正在使用从这些位中提取的时序信息，但无法检测到有效数据。 0 = 无错误 1 = 检测到错误								
SYNC1 = 无法检测到前导码半位或任何 $[SYNC1:0]$ 位。这可能是由于位丢失或信号严重失真而无法检测到。 0 = 无错误 1 = 检测到错误								
BIT = 器件检测到一个数据位。然而，检测样本不足以确保强 1 或 0。 0 = 无错误 1 = 检测到错误								

#### 6.5.4.14.11 DEBUG\_COML\_RC

地址	0x0787							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留		RC_IERR	RC_TXDIS	RC_SOF	RC_BYTE_ERR	RC_UNEXP	RC_CRC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
RC_IERR = 检测到所接收命令帧中的初始化学节错误。这可能是由于字节格式不正确造成的，例如，需要帧初始化学节，但未设置帧起始 (SOF) 位，或者选择了无效的帧类型。由于在 COMH/COML 上接收到的字节会在堆栈中向上传播，因此上层堆栈中的器件很可能也会检测到此错误。后面的所有字节都将被忽略，直至收到 SOF 位。 当忽略通信帧时，器件不会尝试检测被忽略帧中的任何通信错误，也不会将帧计数器中将其计为有效/丢弃。 0 = 未检测到错误 1 = 检测到错误								

<p>RC_TXDIS = 当 <math>[DIR\_SEL] = 0</math> 时有效。器件检测到 COML TX 已禁用，但器件接收到读取数据的命令（即，将数据传送去）。命令帧将计为丢弃帧。 0 = 无错误 1 = 检测到错误</p>
<p>RC_SOF = 当 <math>[DIR\_SEL] = 0</math> 时有效。在 COML 上检测到帧起始 (SOF) 错误。SOF 位仅应在初始化帧中设置，但 SOF 位在当前帧中设置，这不符合预期。 0 = 无错误 1 = 检测到错误</p>
<p>RC_BYTE_ERR = 当 <math>[DIR\_SEL] = 0</math> 时有效。在接收到的命令帧中检测到除初始化学节错误以外的任何字节错误。此错误可能会触发 <i>DEBUG_COMML_BIT</i> 寄存器中设置的一个或多个错误位。 0 = 无错误 1 = 检测到错误</p>
<p>RC_UNEXP = 如果 <math>[DIR\_SEL] = 1</math>，但器件从 COML 接收到命令帧（这是无效条件），器件将设置该错误位。 0 = 无错误 1 = 检测到错误</p>
<p>RC_CRC = 指示导致一个或多个 COML 命令帧被丢弃的 CRC 错误。由于帧被丢弃，因此不会指示该帧中的任何其他错误。 0 = 无错误 1 = 检测到错误</p>

#### 6.5.4.14.12 DEBUG\_COML\_RR\_TR

地址	0x0788							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	RSVD		TR_WAIT	RR_TXDIS	RR_SOF	RR_BYTE_ERR	RR_UNEXP	RR_CRC
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
<p>TR_WAIT = 器件正在等待传输响应，但由于器件收到新命令，因此该操作被终止。当发出广播或堆栈读取命令时，该位有效。 0 = 无错误 1 = 检测到错误</p>								
<p>RR_TXDIS = 当 <math>[DIR\_SEL] = 1</math> 时有效，此时器件接收到响应，但由于 COML TX 被禁用，无法发送到下一个器件。该帧计为丢弃帧。 0 = 无错误 1 = 检测到错误</p>								
<p>RR_SOF = 当 <math>[DIR\_SEL] = 1</math> 时有效。在 COML 上检测到帧起始 (SOF) 错误。SOF 位仅应在初始化帧中设置，但 SOF 位在当前帧中设置，这不符合预期。 0 = 无错误 1 = 检测到错误</p>								
<p>RR_BYTE_ERR = 当 <math>[DIR\_SEL] = 1</math> 时有效。在接收到的响应帧中检测到除初始化学节错误以外的任何字节错误。此错误可能会触发 <i>DEBUG_COMML_BIT</i> 寄存器中设置的一个或多个错误位。 0 = 无错误 1 = 检测到错误</p>								
<p>RR_UNEXP = 如果 <math>[DIR\_SEL] = 0</math>，但器件从 COML 接收到响应帧（这是无效条件），器件将设置该错误位。 0 = 无错误 1 = 检测到错误</p>								
<p>RR_CRC = 指示导致一个或多个 COML 响应帧被丢弃的 CRC 错误。由于帧被丢弃，因此不会指示该帧中的大多数其他错误。如果在 CRC 的最后一个字节上观察到 <math>[RR\_BYTE\_ERR]</math>，将同时指示 CRC 和 BERR。 0 = 无错误 1 = 检测到错误</p>								

#### 6.5.4.14.13 DEBUG\_UART\_DISCARD

地址	0x0789							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							

复位	0	0	0	0	0	0	0	0
COUNT[7:0] = UART 帧计数器，用于跟踪接收或发送的丢弃帧的数量。当读取该寄存器时， <i>DEBUG_UART_DISCARD</i> 和 <i>DEBUG_UART_VALID*</i> 的寄存器锁存且相关计数器复位。								

#### 6.5.4.14.14 DEBUG\_COMH\_DISCARD

地址	0x078A							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = COMH 帧计数器，用于跟踪接收或发送的丢弃帧的数量。当读取该寄存器时， <i>DEBUG_COMH_DISCARD</i> 和 <i>DEBUG_COMH_VALID*</i> 的寄存器锁存且相关计数器复位。								

#### 6.5.4.14.15 DEBUG\_COML\_DISCARD

地址	0x078B							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = COML 帧计数器，用于跟踪接收或发送的丢弃帧的数量。当读取该寄存器时， <i>DEBUG_COML_DISCARD</i> 和 <i>DEBUG_COML_VALID*</i> 的寄存器锁存且相关计数器复位。								

#### 6.5.4.14.16 DEBUG\_UART\_VALID\_HI/LO

##### DEBUG\_UART\_VALID\_HI

地址	0x078C							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = UART 帧计数器的高字节用于跟踪接收或发送的有效帧数。当 <i>DEBUG_UART_VALID_HI/LO</i> 均为 0xFF 时，计数器饱和。当读取 <i>DEBUG_UART_DISCARD</i> 时，该寄存器锁存且相关计数器复位。								

##### DEBUG\_UART\_VALID\_LO

地址	0x078D							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = UART 帧计数器的低字节用于跟踪接收或发送的有效帧数。当 <i>DEBUG_UART_VALID_HI/LO</i> 均为 0xFF 时，计数器饱和。当读取 <i>DEBUG_UART_DISCARD</i> 时，该寄存器锁存且相关计数器复位。								

#### 6.5.4.14.17 DEBUG\_COMH\_VALID\_HI/LO

##### DEBUG\_COMH\_VALID\_HI

地址	0x078E							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = COMH 帧计数器的高字节用于跟踪接收或发送的有效帧数。当 <i>DEBUG_COMH_VALID_HI/LO</i> 均为 0xFF 时，计数器饱和。当读取 <i>DEBUG_COMH_DISCARD</i> 时，该寄存器锁存且相关计数器复位。								

### DEBUG\_COMH\_VALID\_LO

地址	0x078F							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = COMH 帧计数器的低字节用于跟踪接收或发送的有效帧数。当 <i>DEBUG_COMH_VALID_HI/LO</i> 均为 0xFF 时，计数器饱和。当读取 <i>DEBUG_COMH_DISCARD</i> 时，该寄存器锁存且相关计数器复位。								

#### 6.5.4.14.18 DEBUG\_COML\_VALID\_HI/LO

### DEBUG\_COML\_VALID\_HI

地址	0x0790							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = COML 帧计数器的高字节用于跟踪接收或发送的有效帧数。当 <i>DEBUG_COML_VALID_HI/LO</i> 均为 0xFF 时，计数器饱和。当读取 <i>DEBUG_COML_DISCARD</i> 时，该寄存器锁存且相关计数器复位。								

### DEBUG\_COML\_VALID\_LO

地址	0x0791							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	COUNT[7:0]							
复位	0	0	0	0	0	0	0	0
COUNT[7:0] = COML 帧计数器的低字节用于跟踪接收或发送的有效帧数。当 <i>DEBUG_COML_VALID_HI/LO</i> 均为 0xFF 时，计数器饱和。当读取 <i>DEBUG_COML_DISCARD</i> 时，该寄存器锁存且相关计数器复位。								

#### 6.5.4.14.19 DEBUG\_OTP\_SEC\_BLK

地址	0x07A0							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	BLOCK[7:0]							
复位	0	0	0	0	0	0	0	0
BLOCK[7:0] = 保留发生 SEC 的最后一个 OTP 块地址。仅当 <i>FAULT_OTP[SEC_DET]</i> = 1 时才有效。								

#### 6.5.4.14.20 DEBUG\_OTP\_DED\_BLK

地址	0x07A1							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	BLOCK[7:0]							
复位	0	0	0	0	0	0	0	0
BLOCK[7:0] = 保留发生 DED 的最后一个 OTP 块地址。仅当 <i>FAULT_OTP[DED_DET]</i> = 1 时才有效。								

### 6.5.4.15 OTP 编程控制和状态

#### 6.5.4.15.1 OTP\_PROG\_UNLOCK1A 至 OTP\_PROG\_UNLOCK1D

地址	0x0300 至 0x0303							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CODE[7:0]							

复位	0	0	0	0	0	0	0	0
CODE[7:0] = 在执行 OTP 编程之前, 需要将第一个 32 位 OTP 编程解锁代码作为 OTP 编程解锁序列的一部分。此 32 位代码按从 <code>OTP_PROG_UNLOCK1A</code> 到 <code>OTP_PROG_UNLOCK1D</code> 的顺序输入。这些寄存器始终读回 0。								

#### 6.5.4.15.2 OTP\_PROG\_UNLOCK2A 至 OTP\_PROG\_UNLOCK2D

地址	0x0352 至 0x0355							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	CODE[7:0]							
复位	0	0	0	0	0	0	0	0
CODE[7:0] = 在执行 OTP 编程之前, 需要将第二个 32 位 OTP 编程解锁代码作为 OTP 编程解锁序列的一部分。此 32 位代码按从 <code>OTP_PROG_UNLOCK2A</code> 到 <code>OTP_PROG_UNLOCK2D</code> 的顺序输入。这些寄存器始终读回 0。								

#### 6.5.4.15.3 OTP\_PROG\_CTRL

地址	0x030B							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留						PAGESEL	PROG_GO
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
PAGESEL = 选择要编程的客户 OTP 页面。 0 = 第 1 页 1 = 第 2 页								
PROG_GO = 启用对 <code>OTP_PROG_CTRL[PAGESEL]</code> 选择的 OTP 页面的编程。需要将 <code>OTP_PROG_UNLOCK1*</code> 和 <code>OTP_PROG_UNLOCK2*</code> 寄存器设置为正确的代码。 0 = 就绪 1 = 开始 OTP 编程								

#### 6.5.4.15.4 OTP\_ECC\_TEST

地址	0x034C							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	被保留				DED_SEC	MANUAL_AUTO	ENC_DEC	ENABLE
复位	0	0	0	0	0	0	0	0
RSVD = 被保留								
DED_SEC = 设置要测试的解码器功能 ( SEC 或 DED )。该位在编码器测试期间被忽略。 0 = 测试 SEC 功能。设置 <code>FAULT_OTP[SEC_DETECT]</code> 标志并将测试结果输出到 <code>OTP_ECC_DATAOUT*</code> 寄存器。 1 = 测试 DED 功能。设置 <code>FAULT_OTP[DED_DETECT]</code> 标志并将测试结果输出到 <code>OTP_ECC_DATAOUT*</code> 。 注意: 如果检测到 SEC 或 DEC 故障, 主机将设置 <code>[RST_OTP_DATA] = 1</code> 以复位相应的故障。切换到运行 SEC 测试不会清除 DEC 故障, 反之亦然。								
MANUAL_AUTO = 设置用于 ECC 测试的数据的位置。 0 = 自动模式。使用内部数据进行测试。 1 = 手动模式。使用 <code>ECC_DATAIN_n</code> 寄存器中的数据进行测试。用于 MPF 测试。								
ENC_DEC = 将编码器/解码器测试设置为在 <code>OTP_ECC_TEST[ENABLE] = 1</code> 时运行。 0 = 运行解码器测试 1 = 运行编码器测试								
ENABLE = 执行由 <code>[ENC_DEC]</code> 和 <code>[DED_SEC]</code> 位配置的 OTP ECC 测试。 0 = 正常运行, 禁用 ECC 测试 1 = 启动测试								

#### 6.5.4.15.5 OTP\_ECC\_DATAIN1 至 OTP\_ECC\_DATAIN9

地址	0x0343 至 0x034B							
RW	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	DATA[7:0]							
复位	0	0	0	0	0	0	0	0
DATA[7:0] = 在手动模式下启用 ECC 时, $CUST\_ECC\_TEST[MANUAL\_AUTO] = 1$ , $OTP\_ECC\_DATAIN1 \dots 9$ 寄存器用于测试 ECC 编码器/解码器。 如果 $CUST\_ECC\_TEST[ENC\_DEC] = 1$ , $ECC\_DATAIN8$ 至 $ECC\_DATAIN1$ 会被馈送到编码器中。 如果 $CUST\_ECC\_TEST[ENC\_DEC] = 0$ , $ECC\_DATAIN9$ 至 $ECC\_DATAIN1$ 会被馈送到解码器中。必须读回 $ECC\_DATAOUT0 \dots 8$ 字节以验证功能。								

#### 6.5.4.15.6 OTP\_ECC\_DATAOUT1 至 OTP\_ECC\_DATAOUT9

地址	0x0510 至 0x0518							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	DATA[7:0]							
复位	0	0	0	0	0	0	0	0
DATA[7:0] = $OTP\_ECC\_DATAOUT^*$ 字节输出 ECC 解码器和编码器测试的结果。 如果 $CUST\_ECC\_TEST[ENC\_DEC] = 0$ , 则会读取 $ECC\_DATAOUT8$ 至 $ECC\_DATAOUT1$ , 以确定解码器测试是否成功。 如果 $CUST\_ECC\_TEST[ENC\_DEC] = 1$ , 则会读取 $ECC\_DATAOUT9$ 至 $ECC\_DATAOUT1$ , 以确定编码器测试是否成功。正确的结果取决于测试的输入。								

#### 6.5.4.15.7 OTP\_PROG\_STAT

地址	0x0519							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	UNLOCK	OTERR	UVERR	OVERR	SUVERR	SOVERR	PROGERR	DONE
复位	0	0	0	0	0	0	0	0
UNLOCK = 指示 OTP 编程功能解锁状态。设置此位 (即启用 OTP 编程) 后, 主机写入 $OTP\_PROG\_CTRL$ 寄存器以开始 OTP 编程。写入任何其他寄存器都会重新锁定 OTP 编程功能并将该位清除为 0。[ $PROG\_GO$ ] = 1 也会将该位清除为 0。 0 = OTP 编程已锁定 1 = OTP 编程已解锁								
OTERR = 指示芯片温度高于 $T_{OTP\_PROG}$ 并且器件未开始 OTP 编程。 0 = 无故障 1 = 检测到的芯片温度高于 $T_{OTP\_PROG}$ 。中止 OTP 编程。								
UVERR = 表示在 OTP 编程期间在编程电压上检测到欠压错误。通过 [ $PROG\_GO$ ] = 1 清除该位。 0 = 无错误 1 = 检测到 UV 错误								
OVERR = 表示在 OTP 编程期间在编程电压上检测到过压错误。通过 [ $PROG\_GO$ ] = 1 清除该位。从存在此错误的器件收到的信息不得视为可靠的信息。 0 = 无错误 1 = 检测到 OV 错误								
SUVERR = 在开始实际的 OTP 编程之前, 将执行编程电压稳定性测试。该位指示在电压稳定性测试期间检测到欠压错误。通过 [ $PROG\_GO$ ] = 1 清除该位。 0 = 无错误 1 = 在 OTP 编程电压稳定性测试期间检测到 UV 错误								
SOVERR = 在开始实际的 OTP 编程之前, 将执行编程电压稳定性测试。该位指示在电压稳定性测试期间检测到过压错误。通过 [ $PROG\_GO$ ] = 1 清除该位。 0 = 无错误 1 = 在 OTP 编程电压稳定性测试期间检测到 OV 错误								

PROGERR 指示当由于以下任何原因导致的页面设置不正确而检测到错误时：

- 正在尝试编程，但 OTP 编程  $[UNLOCK] = 0$ 。
- 正在尝试对  $[TRY] = 1$  的页面进行编程。
- 尝试对  $[FMterr] = 1$  的页面进行编程。

通过  $[PROG\_GO] = 1$  清除该位。

0 = 无错误或未尝试编程

1 = 检测到错误

DONE = 指示所选页的 OTP 编程状态。通过  $[PROG\_GO] = 1$  清除该位。

0 = 未完成或未尝试编程

1 = 完成。

#### 6.5.4.15.8 OTP\_CUST1\_STAT

地址	0x051A							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	LOADED	LOADWRN	LOADERR	FMterr	PROGOK	UVOK	OVOK	TRY
复位	0	0	0	0	0	0	0	0
LOADED = 指示已选择 OTP 第 1 页以加载到相关寄存器中。有关错误和警告状态，请参阅 $[LOADERR]$ 和 $[LOADWRN]$ 。 0 = 未选择任何内容进行加载 1 = 选择并加载第 1 页								
LOADWRN 指示 OTP 第 1 页已加载，但带有一个或多个 SEC 警告。 = 0 = 无警告，或未尝试加载 1 = 警告								
LOADERR = 指示尝试加载 OTP 第 1 页时发生错误；那就是，在加载所选页面时检测到 DED。 0 = 无错误，或未尝试加载。 1 = 检测到错误								
FMterr = 指示 OTP 第 1 页中存在格式错误；那就是，当设置 $[UVOK]$ 或 $[OVOK]$ 时，但 $[TRY] = 0$ 。如果设置了此位，则不要编程。 0 = 无错误 1 = 检测到错误								
PROGOK = 指示 OTP 第 1 页加载的有效性。有效页表明编程成功。 0 = 无效 1 = 有效								
UVOK = 指示在尝试对 OTP 第 1 页进行编程时检测到 OTP 编程电压欠压情况。OV 条件也可能在关断过程中触发 UV。 0 = 检测到 UV 条件。如果未执行编程尝试，也读取为 0。 1 = 未检测到 UV 条件								
OVOK = 指示在尝试对 OTP 第 1 页进行编程时检测到 OTP 编程电压过压情况。OV 条件将在关断过程中触发 UV。器件必须停止使用。 0 = 检测到 OV 条件。如果未执行编程尝试，也读取为 0。 1 = 未检测到 OV 条件								
TRY = 指示对 OTP 第 1 页的第一次编程尝试。 0 = 未进行第一次尝试 1 = 进行了第一次尝试								

#### 6.5.4.15.9 OTP\_CUST2\_STAT

地址	0x051B							
只读	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
名称	LOADED	LOADWRN	LOADERR	FMterr	PROGOK	UVOK	OVOK	TRY
复位	0	0	0	0	0	0	0	0
LOADED = 指示已选择 OTP 第 2 页以加载到相关寄存器中。有关错误和警告状态，请参阅 $[LOADERR]$ 和 $[LOADWRN]$ 。 0 = 未选择任何内容进行加载 1 = 选择并加载第 2 页								

<p><b>LOADWRN</b> 指示 OTP 第 2 页已加载，但带有一个或多个 SEC 警告。 = 0 = 无警告，或未尝试加载 1 = 警告</p>
<p><b>LOADERR</b> 指示尝试加载 OTP 第 2 页时发生错误；那就是，在加载所选页面时检测到 DED。 = 0 = 无错误，或未尝试加载。 1 = 检测到错误</p>
<p><b>FMTERR</b> = 指示 OTP 第 2 页中存在格式错误；那就是，当设置 [UVOK] 或 [OVOK] 时，但 [TRY] = 0。如果设置了此位，则不要编程。 0 = 无错误 1 = 检测到错误</p>
<p><b>PROGOK</b> = 指示 OTP 第 2 页加载的有效性。有效页表明编程成功。 0 = 无效 1 = 有效</p>
<p><b>UVOK</b> = 指示在尝试对 OTP 第 2 页进行编程时检测到 OTP 编程电压欠压情况。UV 条件也可能在关断过程中触发 UV。 0 = 检测到 UV 条件。如果未执行编程尝试，也读取为 0。 1 = 未检测到 UV 条件</p>
<p><b>OVOK</b> = 指示在尝试对 OTP 第 2 页进行编程时检测到 OTP 编程电压过压情况。UV 条件将在关断过程中触发 UV。器件必须停止使用。 0 = 检测到 OV 条件。如果未执行编程尝试，也读取为 0。 1 = 未检测到 OV 条件</p>
<p><b>TRY</b> = 指示对 OTP 第 2 页的第一次编程尝试。 0 = 未进行第一次尝试 1 = 进行了第一次尝试</p>

## 7 应用和实例

### 备注

以下应用部分中的信息不属于 TI 元件规格，TI 不担保其准确性和完整性。TI 的客户负责确定元件是否适合其用途，以及验证和测试其设计实现以确认系统功能。

### 7.1 应用信息

BQ79616 器件可为 6 节至 16 节串联电池模块提供高精度的电芯电压和电流测量。

### 7.2 典型应用

#### 7.2.1 基底器件应用电路

以下应用电路 ( 请参阅图 7-1 ) 基于连接到 16S 模块的 BQ79616 器件。

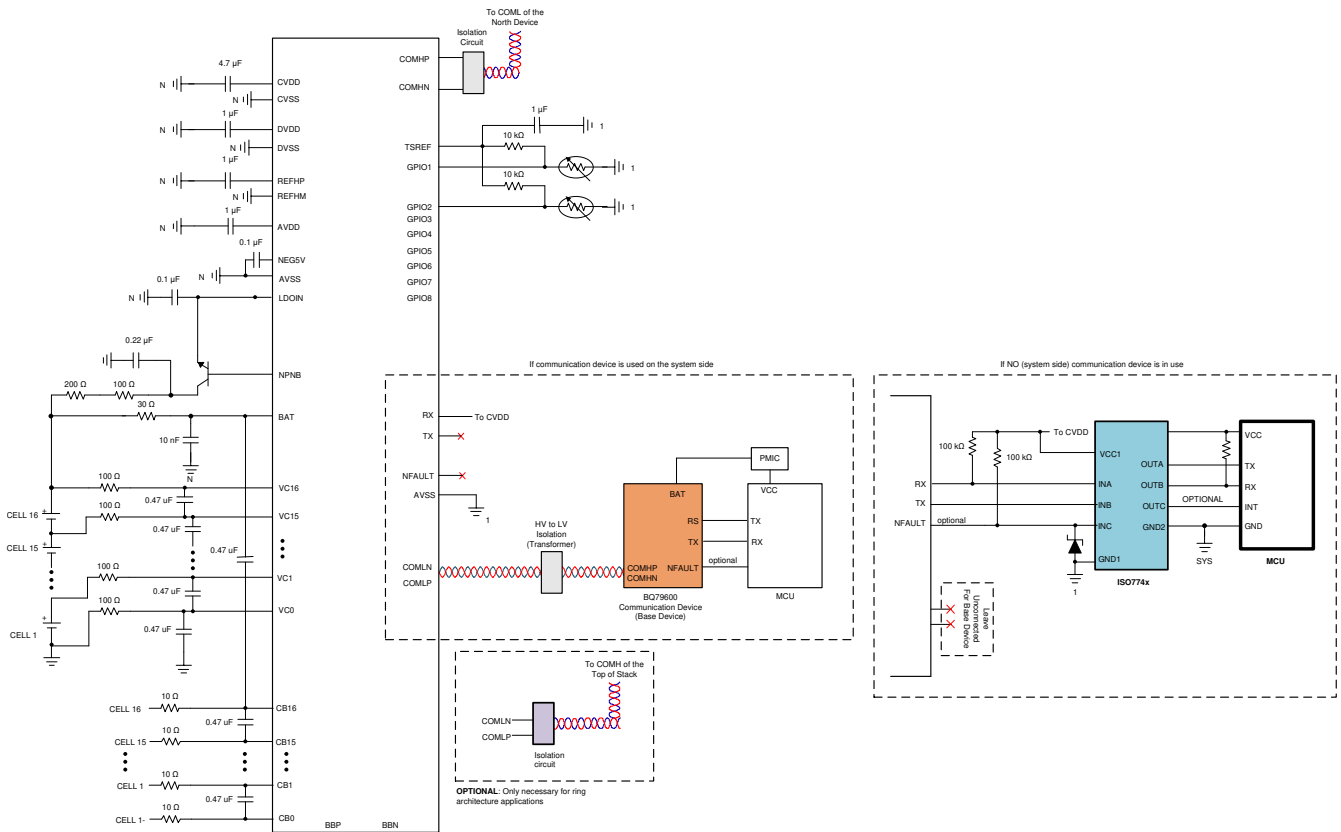


图 7-1. 具有测量应用电路的典型基底器件

#### 7.2.1.1 设计要求

下面的表 7-1 展示了设计参数。

表 7-1. 推荐的设计要求

参数	值
模块电压范围 ( BAT 引脚上的电压 )	9V 至 80V
电芯数 ( 单个器件 )	6 至 16 个电芯 (BQ79616)、6 至 14 个电芯 (BQ79616)、6 至 12 个电芯 (BQ79616)
电芯电压范围	0V 至 5V

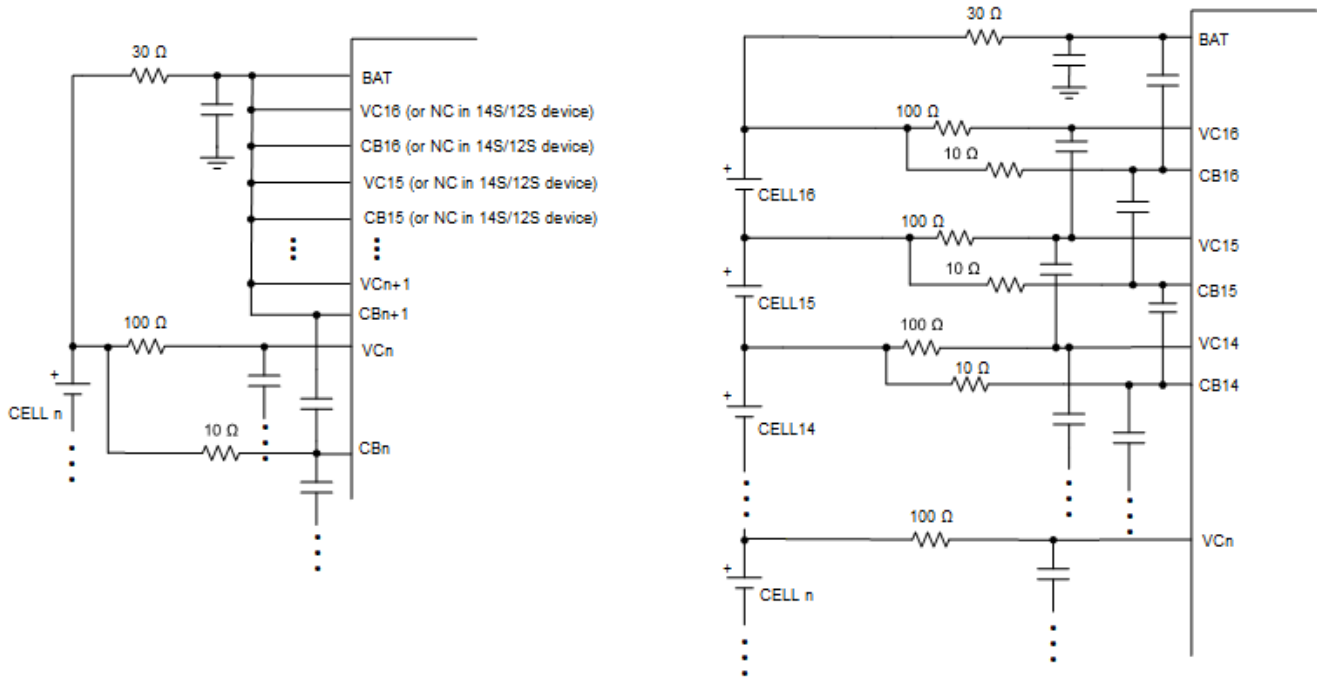
### 7.2.1.2 详细设计过程

#### 7.2.1.2.1 电芯检测和均衡输入

相关引脚	元件	值	说明
VC0 至 VC16	滤波电阻器	100Ω	VC 通道仅需要差分 RC 滤波器。除了用于输入信号过滤之外，这些元件还需要支持电芯模块插入期间的热插拔事件。因此，强烈建议使用建议的元件值。
	滤波电容器	0.47 μ F/16V 或 1 μ F/16V	
CB0 至 CB16	滤波电阻器	取决于系统的均衡 电流要求	CB 通道仅需要差分 RC 滤波器。除了用于输入信号过滤之外，这些元件还需要支持电芯模块插入期间的热插拔事件。因此，强烈建议使用建议的元件值。
	滤波电容器	0.47 μ F/16V 或 1 μ F/16V	

### 电芯连接

从底部通道 ( VC 和 CB 通道 ) 向上填充电池电芯，如果电芯模块尺寸小于 BQ79616 器件的最大通道尺寸，则将上部通道保留为未使用通道。BQ79616 器件中未使用的通道按图 7-2 所示方式进行连接。开路/NC 引脚的 PCB 布局应具有最短走线长度，并且不得连接到电线或电缆。



(a) Customized PCB for certain channels applications - Short unused pins to BAT Pin

(b) One PCB for all channels applications - For BQ79616: Configured for 16 VC and CB

图 7-2. 未使用的 VC 和 CB 通道

#### 7.2.1.2.2 BAT 和外部 NPN

相关引脚	组件	值	说明
BAT	滤波电阻器	30Ω	单端 RC 滤波器，必须使用建议值以实现热插拔性能。
	滤波电容器	10nF/100V 可以根据模块尺寸使用较低的额定电压	

相关引脚	组件	值	说明
NPNB	NPN (Q1)	集电极-发射极击穿电压 80V 至 100V, 但可以根据模块尺寸使用较低的额定值 额定功率 $\geq 1W$ 在预期负载电流下增益 $> 80$ 电流处理能力 $> 100mA$	外部 NPN 用于形成预稳压电路, 以便为 LDOIN 引脚提供 6V (典型值) 输入。 可以通过以下公式来优化 NPN 的额定电压: <b>NPN 额定电压 = 最大 VModule - 最小 VLDOIN + 裕度</b> 其中: 最大 VModule = 充满电的电芯的最大模块电压 最小 VLDOIN = VLDOIN 参数的最小规格 裕度 = 系统瞬态电压 + 符合应用要求的设计裕度
	外部 NPN 集电极上的电阻器 ( $R_{NPN}$ )	因模块电压而异	该电阻器有两个用途: (a) 构成 NPN 预稳压电路的 RC 滤波器 (b) 与 NPN 共享散热
	外部 NPN 集电极上的电容器	0.22 $\mu F/100V$ 可以根据模块尺寸使用较低的额定电压	该电容器构成 NPN 预稳压电路的 RC 滤波器 该电容器的额定值基于模块上的峰值电压尖峰。对于更小的模块尺寸, 可以使用额定电压小于 100V 的电容器。系统设计人员根据其系统容差和要求选择额定电压经过优化的电容器。

为了降低外部 NPN (Q1) 所需的额定功率, 系统设计人员可以在 NPN 集电极上放置功率电阻器, 以通过模块电压 (VModule) 生成 IR 压降。图 7-3 展示了为 BQ79616 器件供电的电流路径。

对于 TI 建议的元件, 典型的  $I_{STARTUP}$  电流 (即器件从 SHUTDOWN 状态进入 ACTIVE, 状态时的浪涌启动电流) 为 20mA。该电流是  $I_{BAT} + I_{LDOIN}$  的总和, 取决于 PCB 板元件和布局。设计人员必须在其端部对其进行表征。

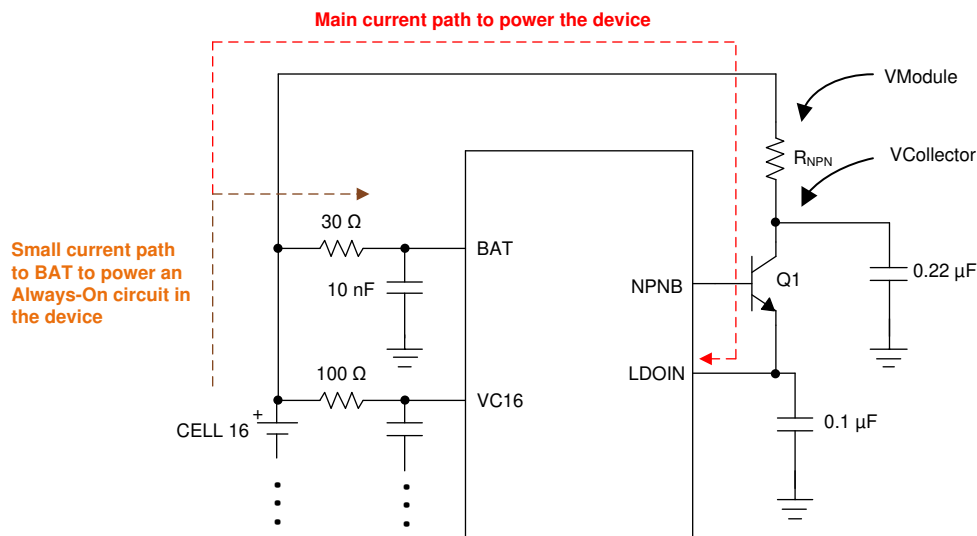


图 7-3. 功耗路径

为确保有足够的余量来保持 LDOIN 引脚上的 6V (典型值) 稳定电压, 系统设计人员应确保 VCollector 的电压在任何时候都不低于 8V (假设 NPN 上的压降约为 2V)。

因此, 允许的最大  $R_{NPN}$  值 =  $((\text{最小 VModule}) - (V_{Collector})) / (\text{最大峰值电流})$

其中:

最小 VModule: 基于模块尺寸和每个应用的最小电芯电压

VCollector: 8V, 假设 NPN 上的压降约为 2V

最大峰值电流: 最高工作电流, 即所有功能都开启时的有功电流。请注意, 不同的通信隔离元件 (例如电容器隔离与变压器或变压器类型) 会为总功耗产生不同的负载。

从电池组顶部单独为器件供电:

该器件设计为由电池组供电。如果需要使用单独的电源为器件供电（如图 7-4 所示），则 BAT 引脚上的电压与最高 VC 引脚电压（相对于接地）之间的关系为  $BAT \text{ 电压} \geq (0.5 \times \text{最高 VC 电压}) + 2$ 。

例如，如果器件连接至最大电芯电压为 4.2V/电芯的 14S 模块，则最高 VC 引脚为 VC14，最高 VC14 电压为  $(4.2V \times 14) = 58.8V$ 。如果单独为 BAT 引脚供电，则 BAT 电压必须  $\geq 31.4V$ 。

同样，如果 BBP/N 通道连接在最高电芯组上方，则 BBP 引脚的电压（相对于接地）将高于 VC 引脚。在这种情况下， $V_{BAT} \geq [(V_{BBP}-2.5) \times 0.84] + 4.5$ 。该要求适用于为 BAT 单独供电的情况，用于确保内部电平转换器正常工作。如果无法维持电压关系，则会导致 VC 和 BB 通道上的 ADC 测量误差增加。

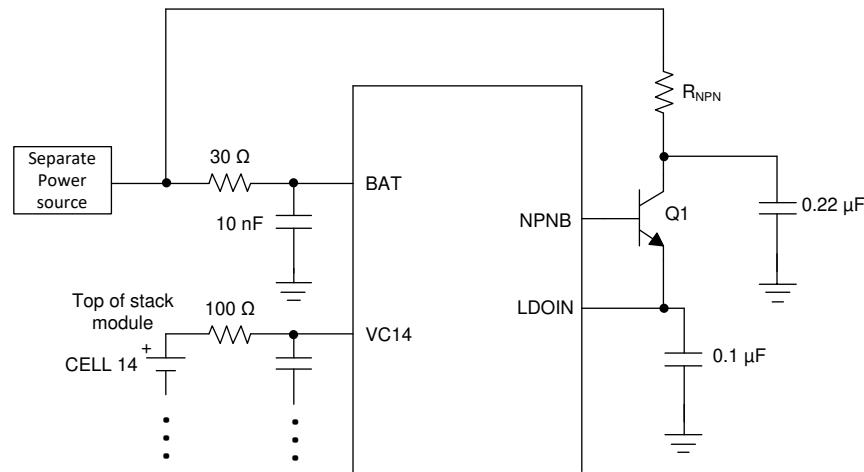


图 7-4. BAT 的单独电源

### 7.2.1.2.3 电源、基准输入

相关引脚	组件	值	说明
AVDD、DVDD、TSREF	旁路电容器	1 μF/10V	用于内部 LDO 的旁路电容器
CVDD	旁路电容器	4.7μF/10V	用于 CVDD 的旁路电容器
NEG5V	旁路电容器	0.1 μF/10V	用于负电荷泵的旁路电容器

### 7.2.1.2.4 适用于热敏电阻输入的 GPIO

当外部热敏电阻仅用于进行 ADC 测量时，对于热敏电阻的类型（NTC 或 PTC）或偏置电阻 (R1) 值没有限制，也不限制热敏电阻相对于偏置电阻是位于高侧还是低侧。

然而，当与集成式 OTUT 比较器一起使用时，可编程 OT 和 UT 阈值范围设计为与 103NTC（25°C 时为 10kΩ）类 NTC 热敏电阻一起工作，遵循图 7-5 所示的连接，R1 和 R2 电阻器具有不同的选项。

- 选项 1：R<sub>1</sub> = 10kΩ，无 R<sub>2</sub>
- 选项 2：R<sub>1</sub> = 10kΩ，并且 R<sub>2</sub> = 100kΩ，可在低温下实现更好的线性度
- 选项 3：R<sub>1</sub> = 3.6kΩ，并且 R<sub>2</sub> = 15kΩ。假设系统设计人员允许 PCB 温度在均衡期间高于电芯温度，则该基本选项可用于旨在实现 OTCB 特性的 NTC。由于该器件不区分电芯与 PCB 上使用的 NTC，因此，使用此选项的 NTC 偏置会以不同方式缩放 NTC 的高温曲线，从而允许在较低的 GPIO 电压下触发为 OT 比较器设置的阈值。因此，使器件仅在该 NTC 上触发 OTCB 阈值。

该器件不需要外部 RC 来进行温度测量。然而，系统设计人员通常会在 NTC 电路的 GPIO 引脚上添加 RC 滤波器。系统设计人员可根据应用需求选择 RC 值。示例：R<sub>GPIO</sub> = 1kΩ，C<sub>GPIO</sub> = 0.1 μF 至 1 μF。

未使用的 GPIO 必须通过 10kΩ 电阻器接地至 AVSS。

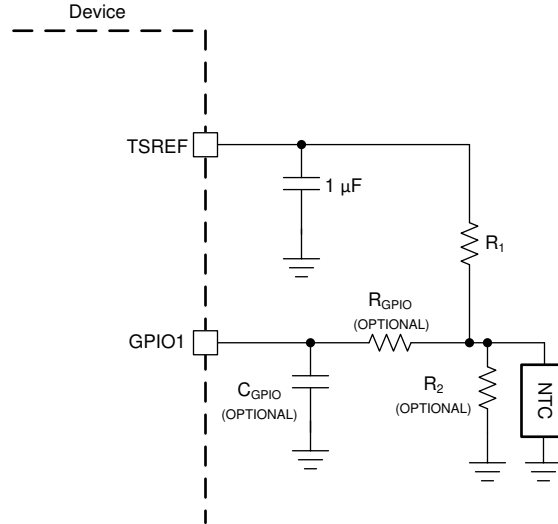


图 7-5. NTC 连接

#### 7.2.1.2.5 内部均衡电流

当使用内部电芯均衡时，器件可以支持的最大均衡电流（在进入热暂停之前）可能会根据环境温度而变化。

#### 7.2.1.2.6 UART、NFAULT

如果器件用作基底器件，则 UART 接口要求通过  $10\text{k}\Omega$  至  $100\text{k}\Omega$  电阻来上拉 TX 和 RX 引脚。不要将 TX 和 RX 悬空。必须将 TX 拉高，以防止在空闲状态期间触发无效的通信帧。当使用串行电缆连接主机控制器时，请将主机侧的 TX 上拉电阻和器件侧的 RX 上拉电阻连接至 CVDD。

如果器件用作堆栈器件，则默认情况下禁用 TX 引脚，并使其保持悬空。RX 引脚短接至 CVDD。

基底器件的 NFAULT 引脚如果未使用，则必须保持悬空。否则，使用  $100\text{k}\Omega$  将其上拉至 CVDD。堆栈器件上的 NFAULT 引脚悬空。

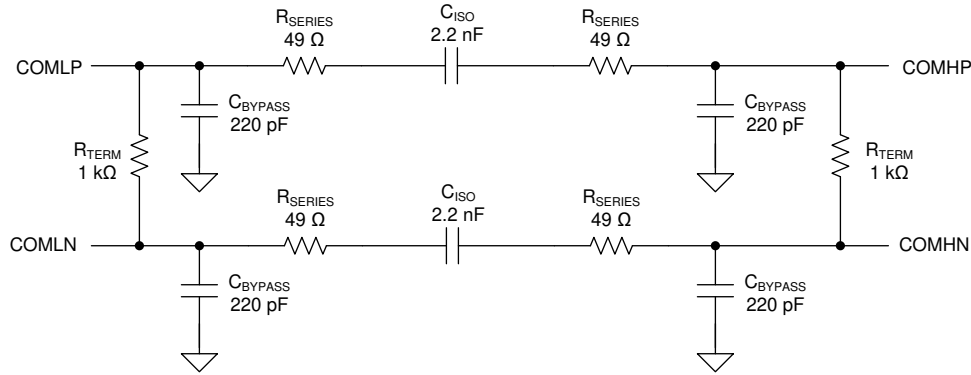
#### 7.2.1.2.7 菊花链隔离

该器件适用于多种菊花链隔离类型：电容器隔离、电容器/扼流圈隔离和变压器隔离。对于以菊花链形式连接到同一 PCB 上的器件，图 7-6 所示的无 ESD 二极管的电容器隔离就足够了。未使用的 COMLP/H 或 COMHP/N 引脚必须连接  $1\text{k}\Omega$  的端接电阻器。

#### 7.2.1.2.7.1 同一 PCB 上连接的器件

表 7-2. 连接在同一 PCB 上的器件的隔离元件

组件	值	说明（同一 PCB 上的电容器隔离）
$R_{\text{TERM}}$	$1\text{k}\Omega$	端接电阻器
$R_{\text{SERIES}}$	$49\Omega$	滤波电阻器和阻抗匹配电阻器。 器件间的连接必须约为 $120\Omega$ （器件信号连接的每端约为 $50\Omega$ 加上 $10\Omega$ 内阻）
$C_{\text{BYPASS}}$	$220\text{pF}/50\text{V}$	旁路电容器
$C_{\text{ISO}}$	$2.2\text{nF}$	隔离电容器 额定电压取决于应用要求。通常选择 2 倍的模块额定电压，以便在系统出现故障时提供隔离裕度。



Components Required for Cap Coupled Daisy Chain on the same PCB

图 7-6. 与同一 PCB 上的器件进行电容器隔离

### 7.2.1.2.7.2 不同 PCB 上连接的器件

对于通过一对双绞线以菊花链方式连接到不同 PCB 的器件，所有这三种隔离类型都可用于菊花链隔离，但无法在菊花链的一侧使用一种隔离类型（例如，COMLP/N 上的变压器与电池管理单元的隔离），同时在菊花链的另一侧使用不同的隔离类型（例如，COMH/N 上的电容器与电芯模块单元的隔离）。

#### 选项 1：电容器隔离

表 7-3. 用于在不同 PCB 上进行电容器隔离的元件

组件	值	说明 (PCB 之间的电容器隔离)
R <sub>TERM</sub>	1kΩ	端接电阻器
R <sub>SERIES</sub>	49Ω	滤波电阻器和阻抗匹配电阻器。 器件间的连接必须约为 120 Ω (器件信号连接的每端约为 50 Ω 加上 10 Ω 内阻)。
C <sub>BYPASS</sub>	220pF/50V	旁路电容器
C <sub>ISO</sub>	2.2nF	隔离电容器 额定电压取决于应用要求。通常选择 2 倍的模块额定电压，以便在系统出现故障时提供隔离裕度。
ESD 二极管	TVS 二极管	ESD 保护器应在热插拔事件期间为通信接口引脚提供保护，也在服务断开或重新连接期间吸收瞬态高压。选择 ESD 二极管，以将 COM* 总线上的最大电压限制为低于最大额定值。为了实现更好的 EMC 性能，建议使用接近最大电压的额定电压来提供尽可能高的共模电压范围。电容必须低于耦合电容 (如果使用电容器耦合)。

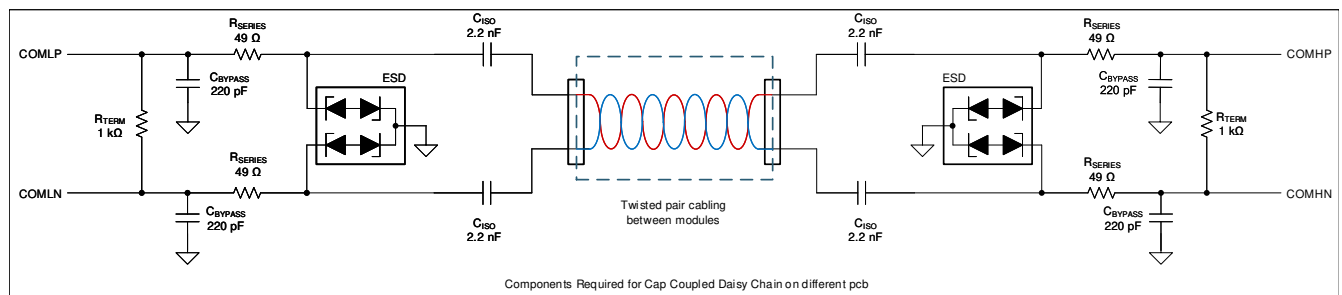


图 7-7. 不同 PCB 上的电容器隔离

图 7-7 展示了在 PCB 之间连接的器件的电容器隔离电路。与同一 PCB 外壳上的电容器隔离类似，此电容器的额定电压必须足够高，以便在系统发生故障导致器件暴露于局部危险电压时提供隔离裕度。电压由应用要求决定，但通常选择 2 倍的模块电压。

菊花链总线上的电容对性能有直接影响。在设计 EMC 的通信稳健性时，必须考虑支持元件和布线中的所有寄生电容。来自电缆、ESD 二极管、旁路电容和扼流圈的电容与可能会影响性能的隔离电容器形成电容分压器。此外，总线上的电容大小对通信期间的工作电流（电容器充电或放电）有直接影响。

### 选项 2：电容器加共模扼流圈隔离

表 7-4. 用于电容器加共模扼流圈隔离的元件

组件	值	描述 (PCB 之间的电容器加扼流圈隔离)
$R_{TERM}$	1k $\Omega$	端接电阻器
$R_{SERIES}$	49 $\Omega$	滤波电阻器和阻抗匹配电阻器。 器件间的连接必须约为 120 $\Omega$ (器件信号连接的每端约为 50 $\Omega$ 加上 10 $\Omega$ 内阻)
$C_{BYPASS}$	220pF/50V	旁路电容器
$C_{ISO}$	2.2nF	隔离电容器 额定电压取决于应用要求。通常选择 2 倍的模块额定电压，以便在系统出现故障时提供隔离裕度。
共模扼流圈	100 $\mu$ H 至 500 $\mu$ H	共模扼流圈 (电感范围 100 $\mu$ H 至 500 $\mu$ H 是一个通用指导值，而不是保证的范围，因为有许多参数会影响共模扼流圈的性能。)
ESD 二极管 (可选)	TVS 二极管	可选的 ESD 保护取决于 PCB 级 ESD 要求 (是否添加此二极管取决于用户系统级 ESD 要求)

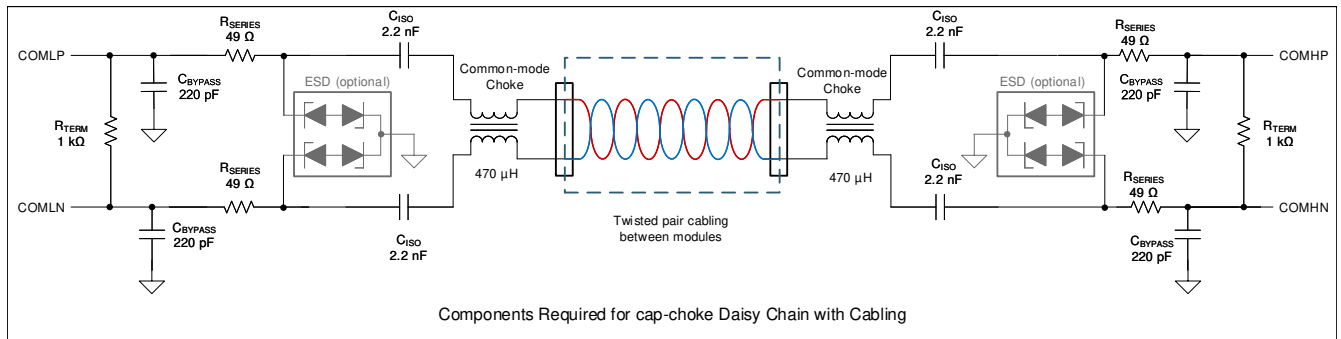


图 7-8. 电容器加扼流圈隔离

较长的电缆长度或异常嘈杂的应用可能要求使用共模扼流圈滤波器。电容器加扼流圈隔离比仅使用电容器具有更好的抗噪声性能。对于这些应用，为确保正常运行，至少应使用 100  $\mu$ H 至 500  $\mu$ H 的汽车级共模滤波器。为了在嘈杂的环境中实现更佳性能，应使用双共模滤波器 (470 $\mu$ H)。在 1MHz 至 100MHz 范围内，推荐的扼流圈阻抗至少为 1k $\Omega$ ，对于更高频率，建议阻抗高于 300  $\Omega$ 。

### 选项 3：变压器隔离

表 7-5. 变压器隔离的元件

组件	值	描述 (PCB 之间的电容器加扼流圈隔离)
$R_{TERM}$	1k $\Omega$	端接电阻器
$R_{SERIES}$	49 $\Omega$	滤波电阻器和阻抗匹配电阻器。 器件间的连接必须约为 120 $\Omega$ (器件信号连接的每端约为 50 $\Omega$ 加上 10 $\Omega$ 内阻)
$C_{BYPASS}$	220pF/50V	旁路电容器
变压器	电感：150 $\mu$ H 至 1400 $\mu$ H	电感范围 150 $\mu$ H 至 1400 $\mu$ H 是一个通用指导值，而不是保证的范围，因为有许多参数会影响变压器的性能。有关具体的推荐器件，请参阅 SLVAEP4 BQ79616 菊花链通信应用报告。用户应在其环境中进行全面测试。
ESD 二极管 (可选)	TVS 二极管	可选的 ESD 保护取决于 PCB 级 ESD 要求 (是否添加此二极管取决于用户系统级 ESD 要求。)

支持变压器隔离。[变压器隔离](#)展示了其实现方式。例如，变压器隔离可用于低压和高压边界之间，以实现电隔离。

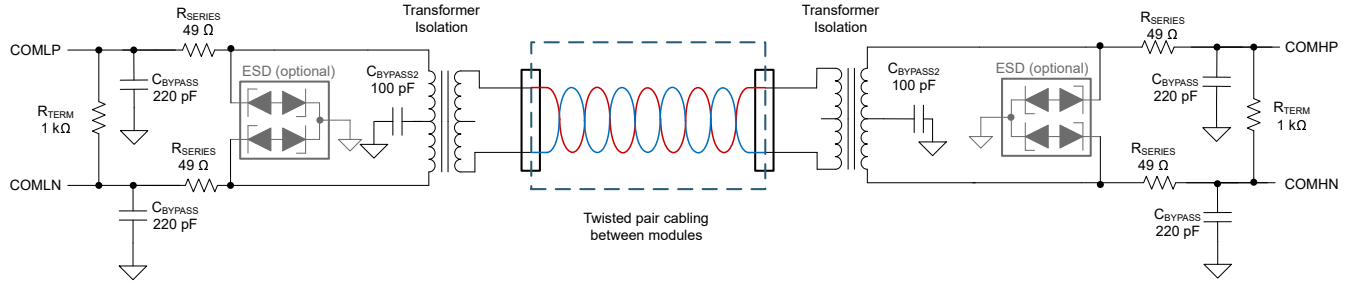


图 7-9. 变压器隔离

### 7.2.1.3 应用曲线

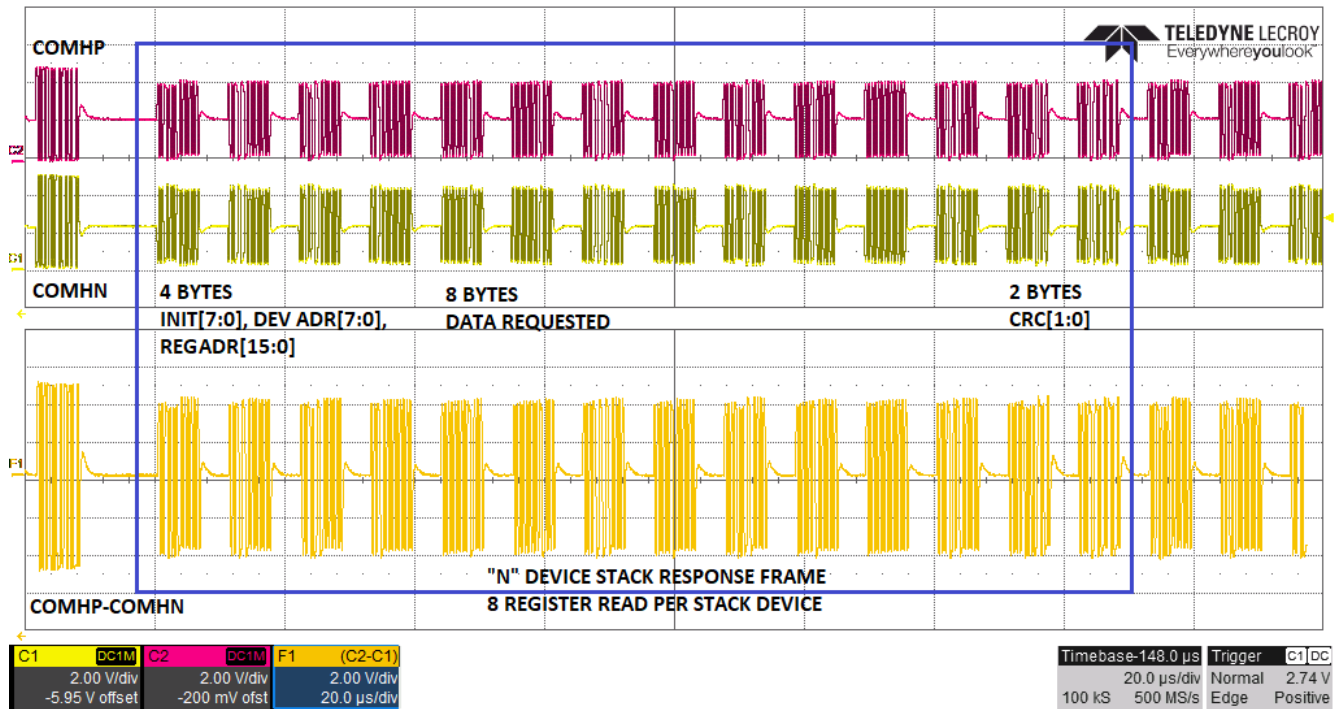


图 7-10. 从堆栈器件读取 8 个寄存器的响应帧

### 7.2.2 菊花链器件应用电路

以下应用电路 ( 请参阅图 7-11 ) 基于连接到 16S 模块的 BQ79616 器件。

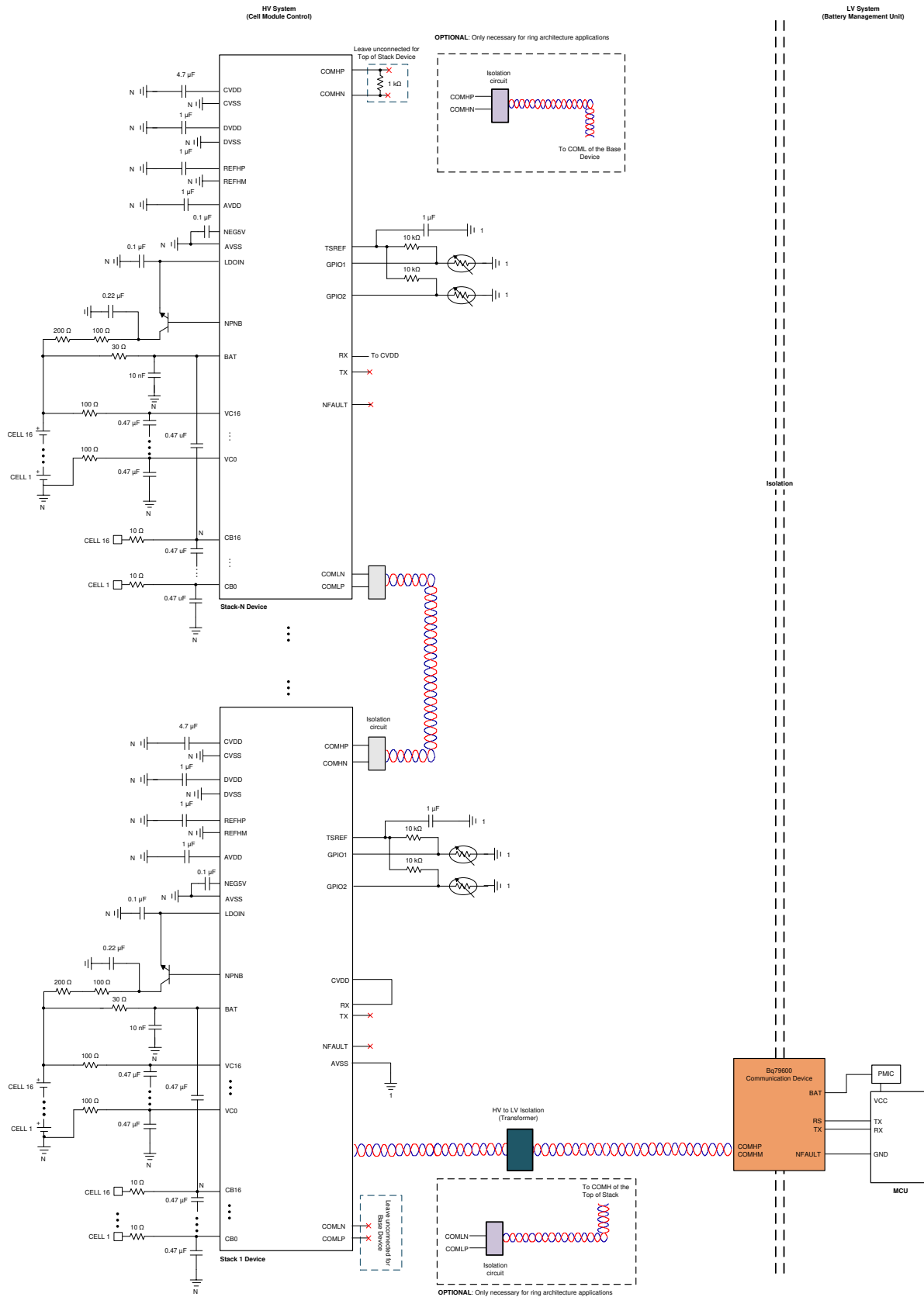


图 7-11. 菊花链器件应用电路

#### 7.2.2.1 设计要求

有关设计要求，请参阅节 [7.2.1.1](#)。

#### 7.2.2.2 详细设计过程

有关详细的设计过程，请参阅节 [7.2.1.2](#)。

## 8 电源相关建议

该器件由 BAT 引脚和 LDOIN 引脚供电，LDOIN 引脚由外部 NPN 构成的预稳压电路进行稳压。该器件可由 BAT 引脚上低至 9V 的电池模块供电（无 OTP 编程）。然而，系统设计人员必须相应地调整  $R_{NPN}$  电阻器，以确保在  $R_{NPN}$  和外部 NPN 上的 IR 压降之后 LDOIN 引脚上有足够的余量来提供 6V 电压。例如，如果 BAT 电压为 9V，则  $R_{NPN}$  减小至  $10\Omega$ ，以在 LDOIN 引脚上提供足够的电压。

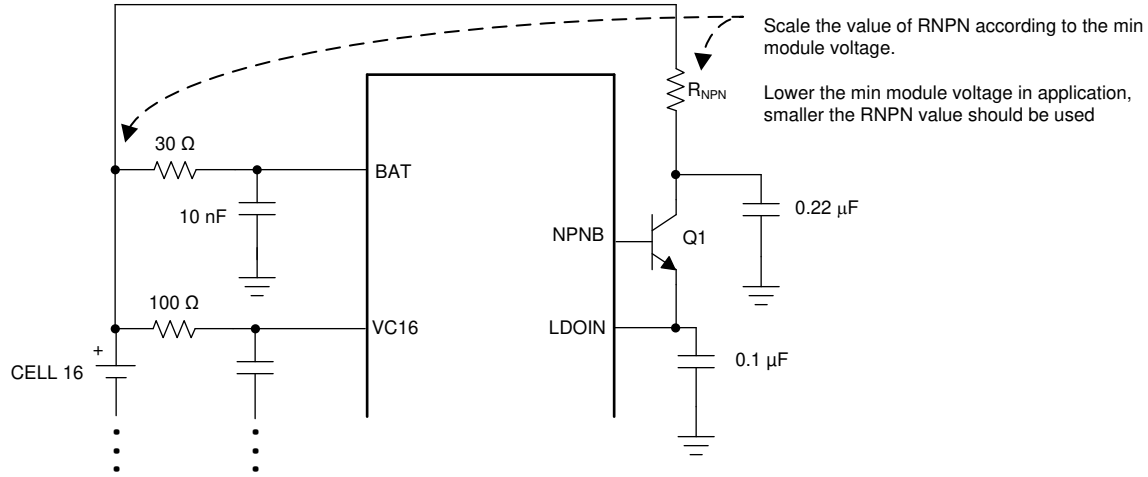


图 8-1. 器件供电路径

通过 BQ79616 系列的汇流条支持，可以将多个电芯模块连接到同一个器件。将从每个电芯模块汲取相同的功率。

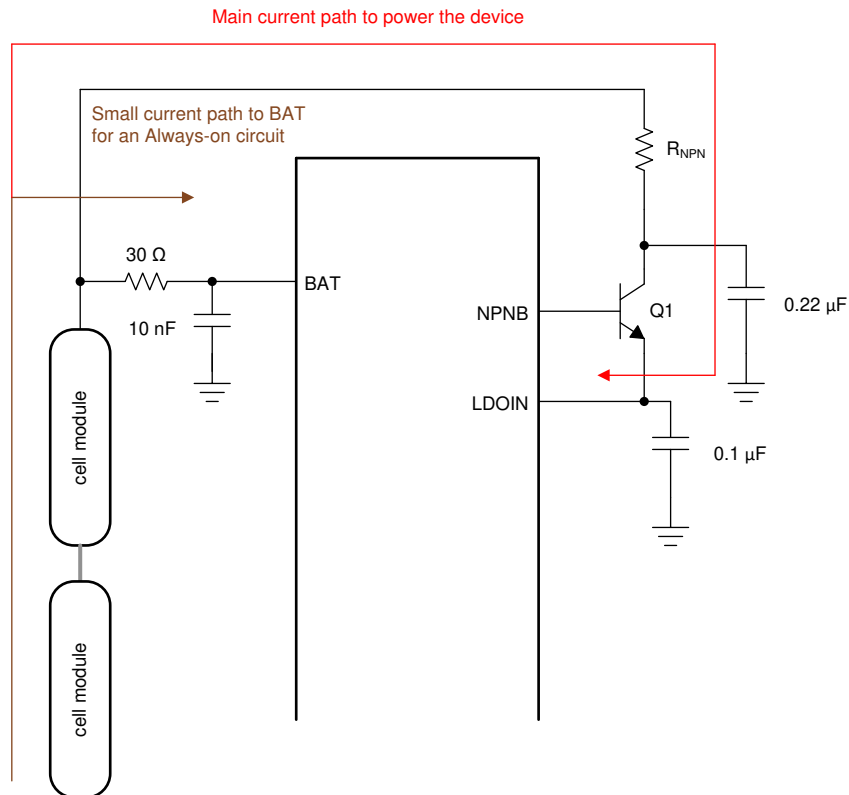


图 8-2. 器件可由堆栈中的多个电芯模块供电

## 9 布局

必须仔细设计该器件的布局。任何不遵守以下准则的设计都可能影响 ADC 精度和 EMI 性能。必须注意进出器件的信号的布局，以避免将噪声耦合到敏感输入上。接地和电源连接以及通信信号的布局也应该仔细进行。

### 9.1 布局指南

#### 9.1.1 接地平面

为确保器件的最佳性能，建立干净的接地方案非常重要。有三个接地引脚 ( AVSS、DVSS、CVSS ) 用于连接器件的内部电源，有一个接地基准 ( REFHM ) 用于连接精密基准。存在高噪声接地和低噪声接地，必须在最初的布局中将其分开，并在较低的 PCB 层中将其重新连接在一起。如果可能，必须将外部元件 ( 例如旁路电容器 ) 连接到适当的接地组，以使高噪声接地和低噪声接地分开。

- AVSS 接地：
  - 用于以下引脚的旁路电容器：BAT、VC0、CB0 和 AVDD。
  - 封装电源板
- DVSS 接地：
  - 用于 DVDD 的旁路电容器。
  - GPIO 滤波电容器 ( 如果使用 )。如果需要，它还可以连接到 AVSS 接地平面。
- CVSS 接地：
  - 用于 GPIO、CVDD、TSREF、NEG5V、LDOIN、COMHP/N 和 COMLP/N 的旁路电容器。
- REFHM 接地：
  - 用于 REFHP 的旁路电容器。
  - 如果可能，在信号连接层上将 REFHM 与 AVSS 分开，并在较低层将 REFHM 重新连接到 AVSS 接地平面。

即使在主要用于信号路由的 PCB 层上，如果可能，最好有一个小的接地覆铜岛来提供低阻抗接地，而不是简单地提供一个过孔来通过接地引线连接到较低接地平面。

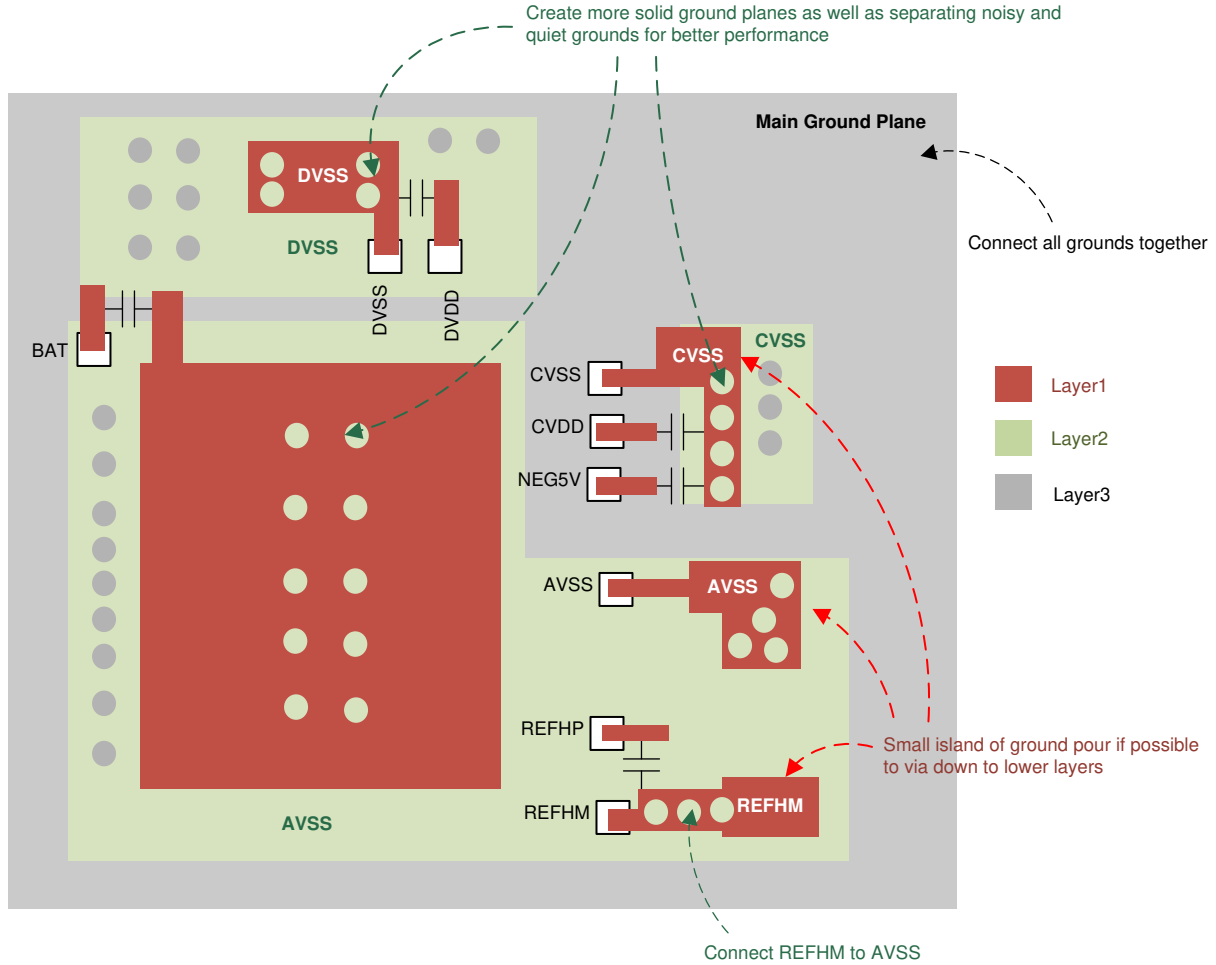


图 9-1. 接地布局注意事项

如果多个器件放置在同一个 PCB 上，则每个器件必须有自己的接地平面，并具有适当的布局间隙。

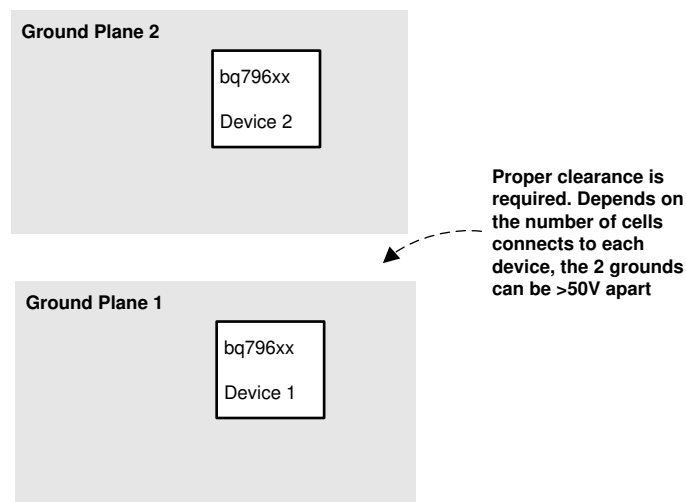


图 9-2. 同一 PCB 上每个器件的单独接地平面

### 9.1.2 电源和基准的旁路电容器

以下引脚的旁路电容器必须尽可能靠近器件引脚放置，以确保性能正常，尤其是对于 REFHP 电容器。

- REFHP、BAT、LDOIN、AVDD、DVDD、CVDD、TSREF 和 NEG5V

### 9.1.3 电芯电压检测

电芯电压检测走线 ( VC 引脚和 CB 引脚 ) 必须与阻抗匹配并联放置。均衡走线 ( CB 引脚 ) 的尺寸必须合适，才能承载应用的最大均衡电流和适当的热性能。

建议对 BAT 引脚和顶部 VC 引脚连接使用单独的电缆、连接端子和 PCB 走线。这同样适用于 AVSS 和 VC0 连接。这可以避免器件电流对顶部和底部电芯电压测量的影响。

如果相同的电缆和连接器端子用于 BAT/顶部 VC 引脚连接和 AVSS/VC0 引脚连接，则连接到 BAT/顶部 VC 引脚和 AVSS/VC0 引脚的 PCB 走线必须在连接端子处分离。请注意，器件电流仍会经由电芯进入 PCB 电缆，这可能会跨电缆连接对顶部和底部的电芯测量引入 IR 误差。

### 9.1.4 菊花链通信

为了获得最稳健的菊花链通信，在 COMHP/N 和 COMLP/N 电路上进行正确的布局非常重要。

- 保持差分引线尽可能短并且尽可能直。尽量减少引线转弯并避免引线出现任何绕圈。
- 将差分引线保持在同一层。排布引线时应屏蔽并匹配引线阻抗。
- 将隔离元件放置在靠近连接器的位置。
- 使用电容隔离时，请将 COMxP/N 对 ( 其中 x = H 或 L ) 的高压电容器沿平行引线彼此靠近放置。
- 在所有 PCB 层中的菊花链元件周围创建一个禁止区域 ( 没有其他引线，也没有接地平面 )。

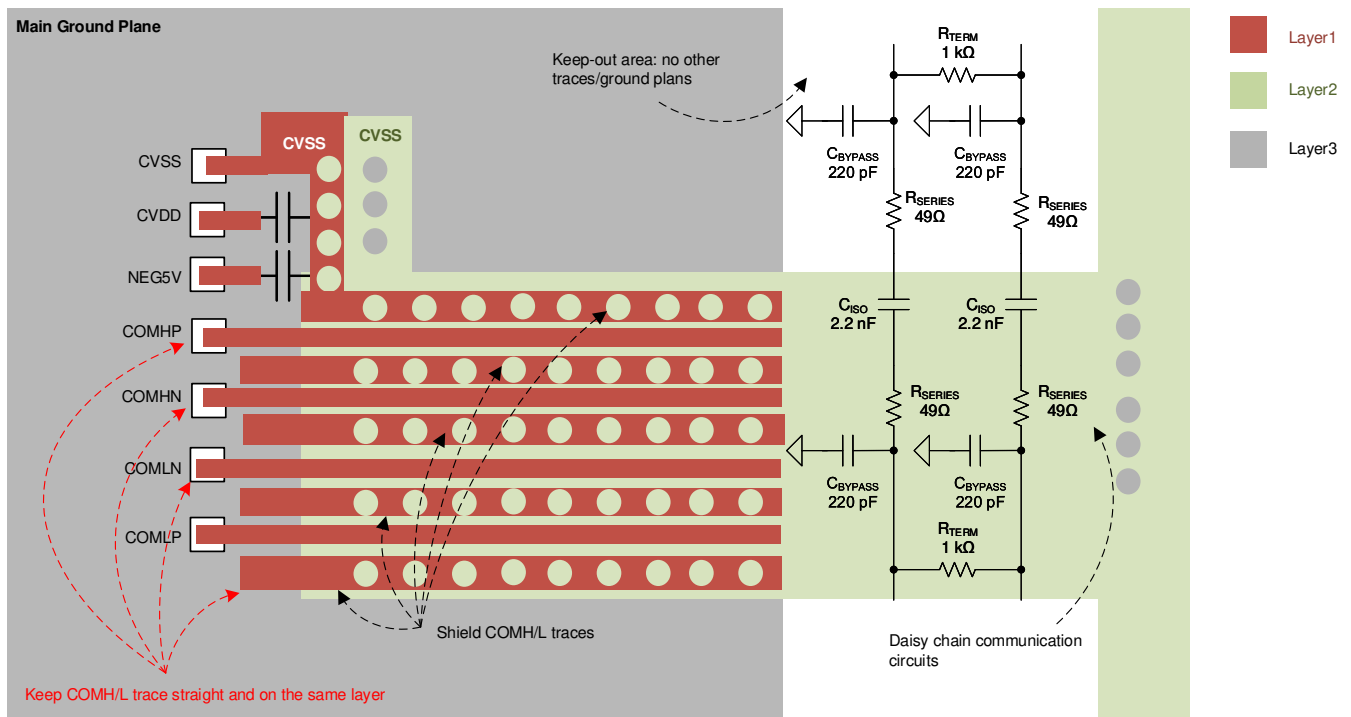


图 9-3. 菊花链布局注意事项

## 9.2 布局示例

本节通过一个布局示例介绍了 BQ79616 评估模块 (EVM) 设计。

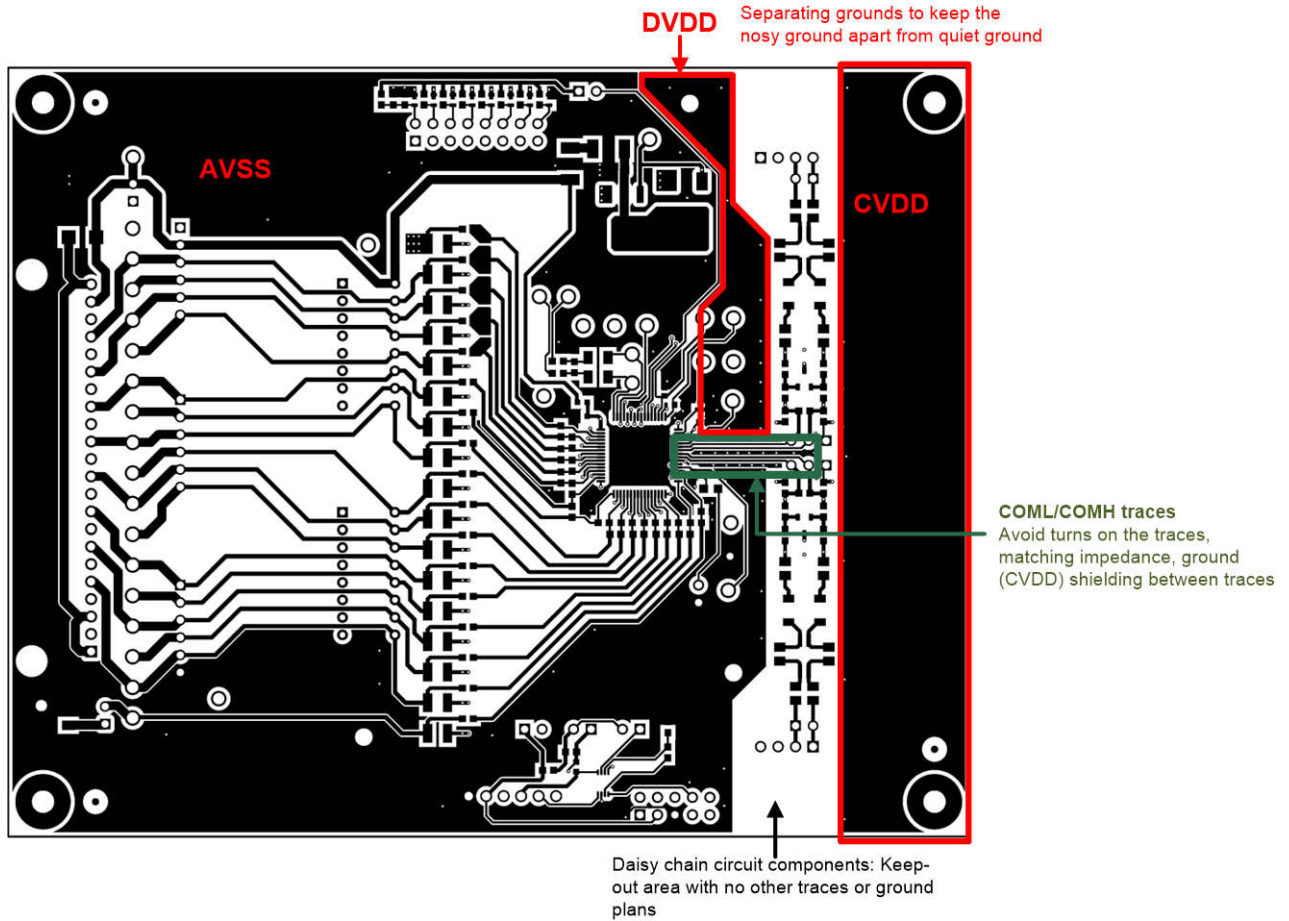


图 9-4. 顶端信号层

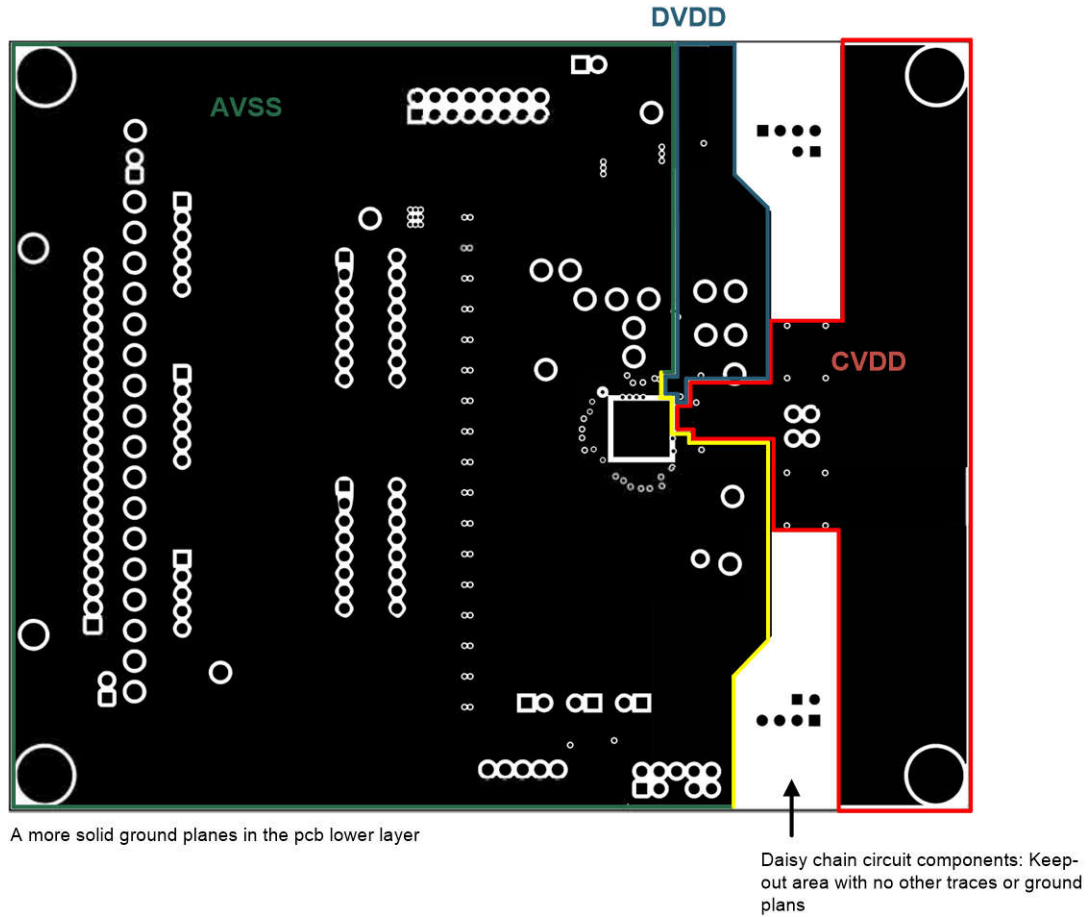


图 9-5. 具有实心独立接地平面的第二层

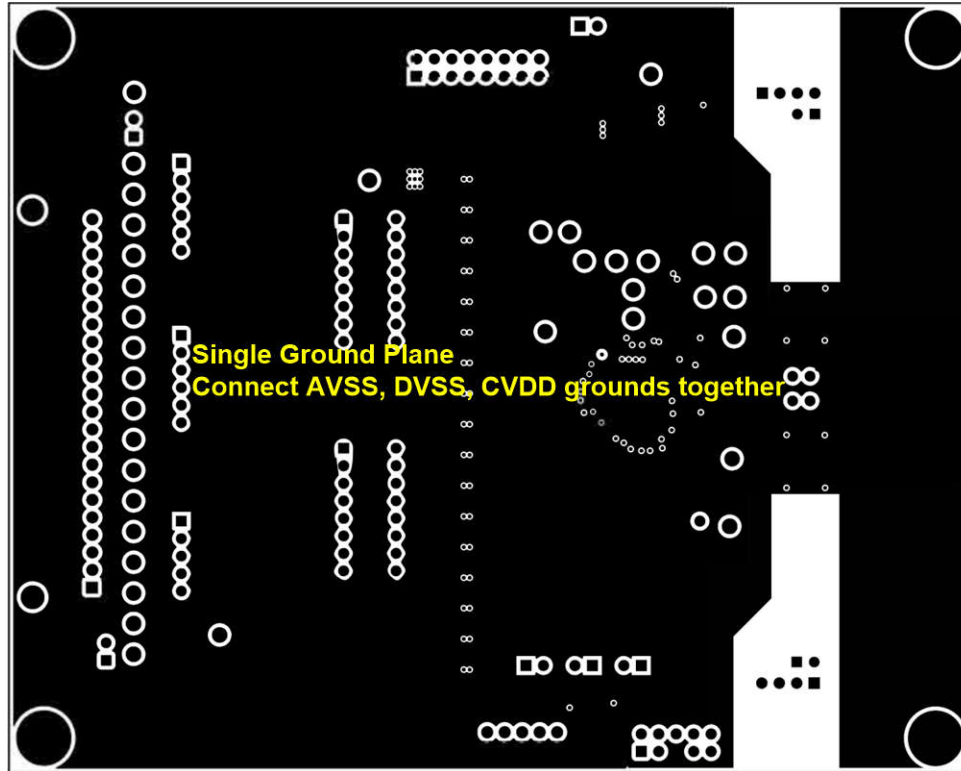


图 9-6. 具有单个接地平面的第三层

## 10 器件和文档支持

### 10.1 器件支持

#### 10.1.1 第三方产品免责声明

TI 发布的与第三方产品或服务有关的信息，不能构成与此类产品或服务或保修的适用性有关的认可，不能构成此类产品或服务单独或与任何 TI 产品或服务一起的表示或认可。

### 10.2 接收文档更新通知

要接收文档更新通知，请导航至 [ti.com](http://ti.com) 上的器件产品文件夹。点击 [通知](#) 进行注册，即可每周接收产品信息更改摘要。有关更改的详细信息，请查看任何已修订文档中包含的修订历史记录。

### 10.3 支持资源

[TI E2E™ 中文支持论坛](#) 是工程师的重要参考资料，可直接从专家处获得快速、经过验证的解答和设计帮助。搜索现有解答或提出自己的问题，获得所需的快速设计帮助。

链接的内容由各个贡献者“按原样”提供。这些内容并不构成 TI 技术规范，并且不一定反映 TI 的观点；请参阅 TI 的 [使用条款](#)。

### 10.4 商标

TI E2E™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

### 10.5 静电放电警告



静电放电 (ESD) 会损坏这个集成电路。德州仪器 (TI) 建议通过适当的预防措施处理所有集成电路。如果不遵守正确的处理和安装程序，可能会损坏集成电路。

ESD 的损坏小至导致微小的性能降级，大至整个器件故障。精密的集成电路可能更容易受到损坏，这是因为非常细微的参数更改都可能会导致器件与其发布的规格不相符。

### 10.6 术语表

[TI 术语表](#) 本术语表列出并解释了术语、首字母缩略词和定义。

## 11 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (June 2023) to Revision A (June 2026)	Page
• 删除了 <a href="#">图 7-2</a> 中的图 (c) 和 (d).....	183

日期	修订版本	说明
June 2023	*	初始发行版

## 12 机械、封装和可订购信息

以下页面包含机械、封装和可订购信息。这些信息是指定器件可用的最新数据。数据如有变更，恕不另行通知，且不会对此文档进行修订。有关此数据表的浏览器版本，请查阅左侧的导航栏。

**PACKAGING INFORMATION**

Orderable part number	Status (1)	Material type (2)	Package   Pins	Package qty   Carrier	RoHS (3)	Lead finish/ Ball material (4)	MSL rating/ Peak reflow (5)	Op temp (°C)	Part marking (6)
<a href="#">BQ79616PAPR</a>	Active	Production	HTQFP (PAP)   64	1000   LARGE T&R	Yes	NIPDAU	Level-3-260C-168 HR	-40 to 125	BQ75216
BQ79616PAPR.A	Active	Production	HTQFP (PAP)   64	1000   LARGE T&R	Yes	NIPDAU	Level-3-260C-168 HR	-40 to 125	BQ75216

(1) **Status:** For more details on status, see our [product life cycle](#).

(2) **Material type:** When designated, preproduction parts are prototypes/experimental devices, and are not yet approved or released for full production. Testing and final process, including without limitation quality assurance, reliability performance testing, and/or process qualification, may not yet be complete, and this item is subject to further changes or possible discontinuation. If available for ordering, purchases will be subject to an additional waiver at checkout, and are intended for early internal evaluation purposes only. These items are sold without warranties of any kind.

(3) **RoHS values:** Yes, No, RoHS Exempt. See the [TI RoHS Statement](#) for additional information and value definition.

(4) **Lead finish/Ball material:** Parts may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

(5) **MSL rating/Peak reflow:** The moisture sensitivity level ratings and peak solder (reflow) temperatures. In the event that a part has multiple moisture sensitivity ratings, only the lowest level per JEDEC standards is shown. Refer to the shipping label for the actual reflow temperature that will be used to mount the part to the printed circuit board.

(6) **Part marking:** There may be an additional marking, which relates to the logo, the lot trace code information, or the environmental category of the part.

Multiple part markings will be inside parentheses. Only one part marking contained in parentheses and separated by a "~" will appear on a part. If a line is indented then it is a continuation of the previous line and the two combined represent the entire part marking for that device.

**Important Information and Disclaimer:**The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

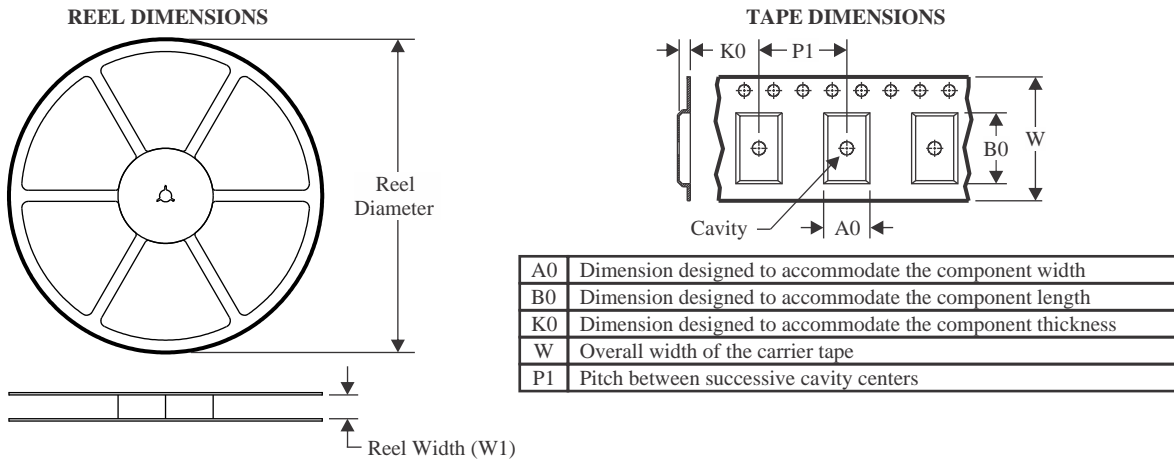
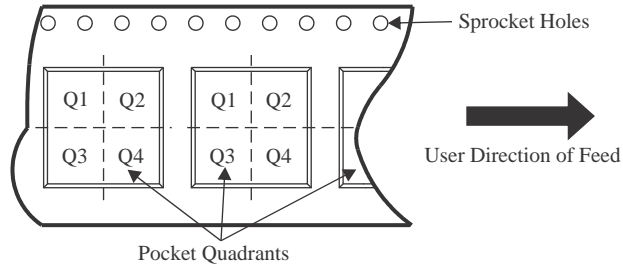
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

**OTHER QUALIFIED VERSIONS OF BQ79616 :**

- Automotive : [BQ79616-Q1](#)

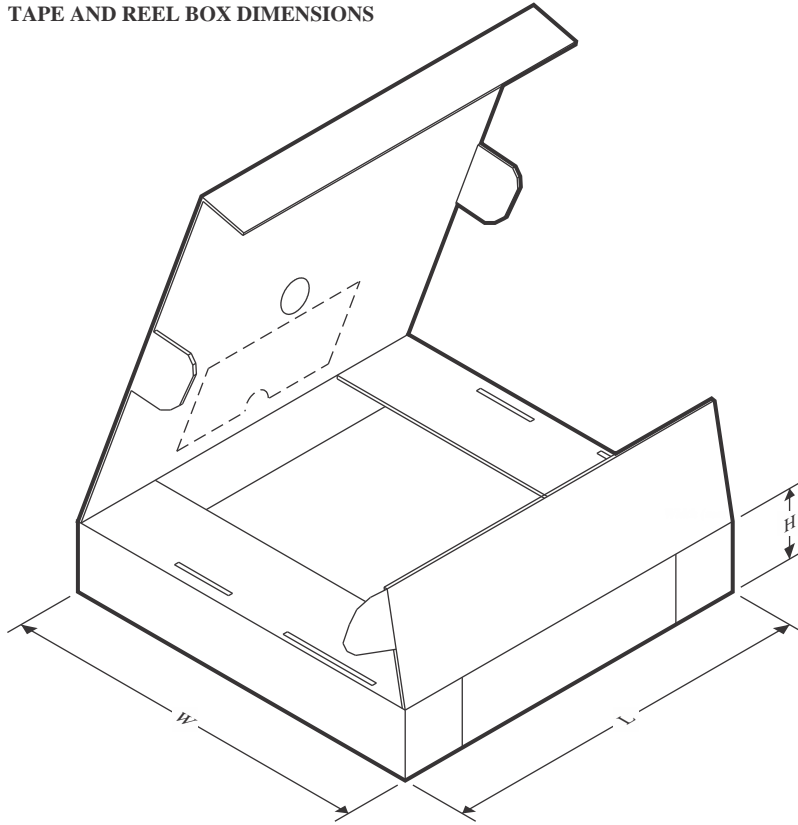
NOTE: Qualified Version Definitions:

- Automotive - Q100 devices qualified for high-reliability automotive applications targeting zero defects

**TAPE AND REEL INFORMATION**

**QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE**


\*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
BQ79616PAPR	HTQFP	PAP	64	1000	330.0	24.4	13.0	13.0	1.5	16.0	24.0	Q2

**TAPE AND REEL BOX DIMENSIONS**


\*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
BQ79616PAPR	HTQFP	PAP	64	1000	367.0	367.0	55.0

## GENERIC PACKAGE VIEW

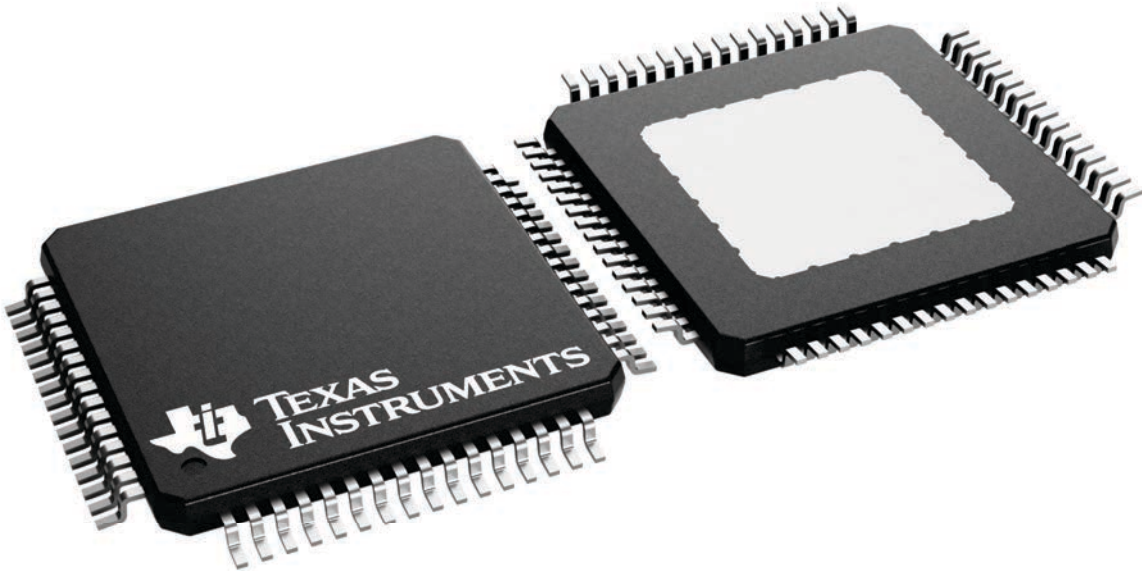
**PAP 64**

**HTQFP - 1.2 mm max height**

10 x 10, 0.5 mm pitch

QUAD FLATPACK

This image is a representation of the package family, actual package may vary.  
Refer to the product data sheet for package details.



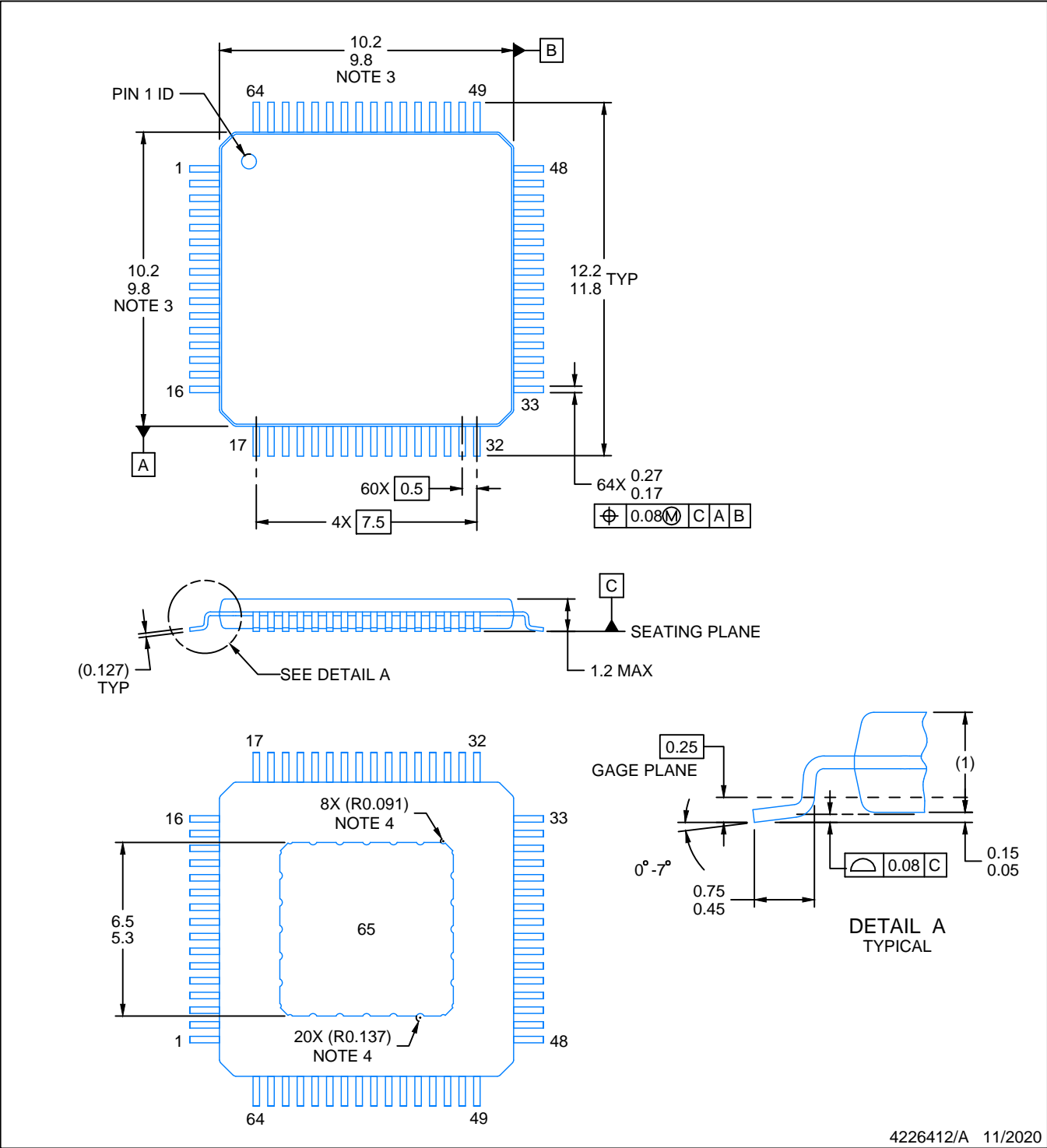
4226442/A

# PACKAGE OUTLINE

PAP0064F

PowerPAD™ TQFP - 1.2 mm max height

PLASTIC QUAD FLATPACK



4226412/A 11/2020

**NOTES:**

PowerPAD is a trademark of Texas Instruments.

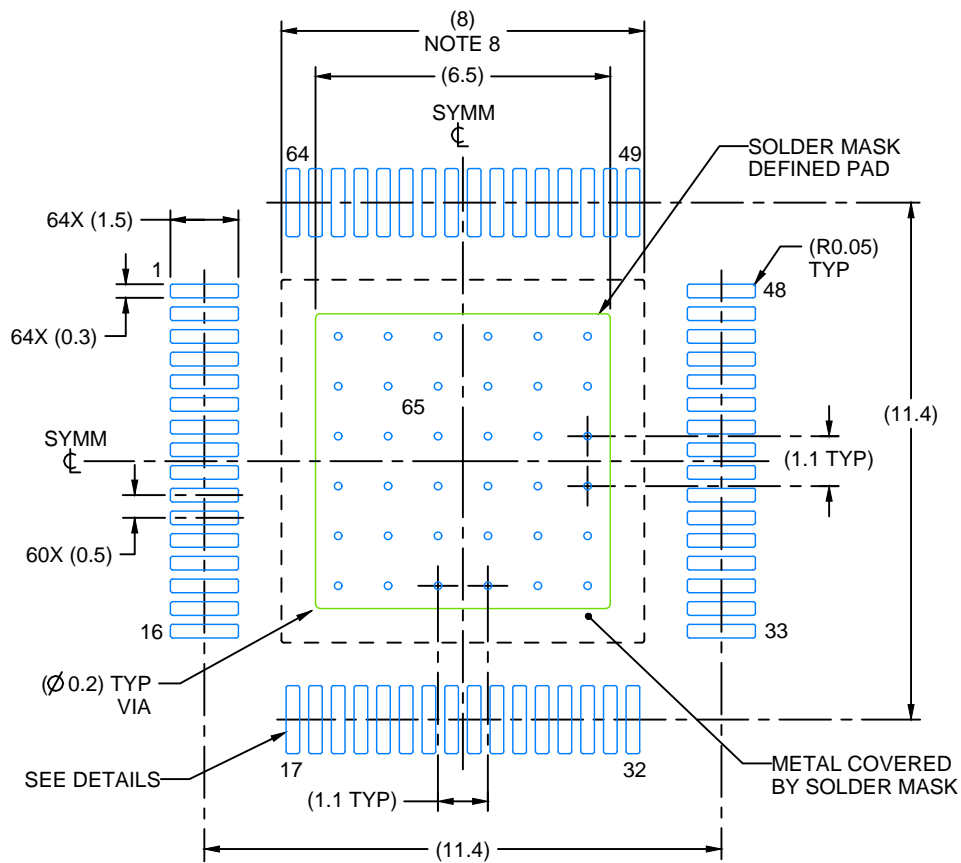
1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.
3. This dimension does not include mold flash, protrusions, or gate burrs.
4. Strap features may not be present.
5. Reference JEDEC registration MS-026.

# EXAMPLE BOARD LAYOUT

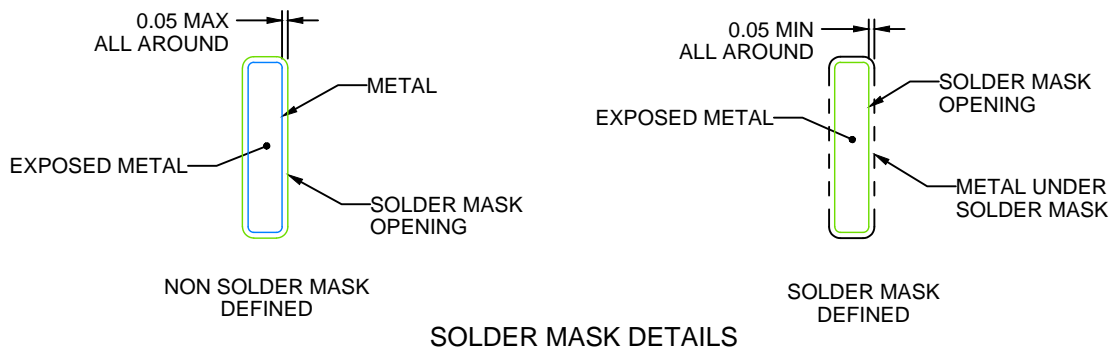
PAP0064F

PowerPAD™ TQFP - 1.2 mm max height

PLASTIC QUAD FLATPACK



LAND PATTERN EXAMPLE  
EXPOSED METAL SHOWN  
SCALE:6X



SOLDER MASK DETAILS

4226412/A 11/2020

NOTES: (continued)

6. Publication IPC-7351 may have alternate designs.
7. Solder mask tolerances between and around signal pads can vary based on board fabrication site.
8. This package is designed to be soldered to a thermal pad on the board. See technical brief, Powerpad thermally enhanced package, Texas Instruments Literature No. SLMA002 ([www.ti.com/lit/slma002](http://www.ti.com/lit/slma002)) and SLMA004 ([www.ti.com/lit/slma004](http://www.ti.com/lit/slma004)).
9. Vias are optional depending on application, refer to device data sheet. It is recommended that vias under paste be filled, plugged or tented.
10. Size of metal pad may vary due to creepage requirement.



## 重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月