

文档编号: AN2070

上海东软载波微电子有限公司

应用笔记

ES32H05x4

修订历史

版本	修订日期	修改概要
V1.0	2026-03-06	初版
V1.1	2026-09-08	1.Flash ECC 使能时不能在 SRAM 运行程序; 2.中断向量重映射功能仅可在 Flash 非 ECC 模式下且 SRAM 运行时使用。

地 址：中国上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：http://www.essemi.com/

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系

目 录

内容目录

第 1 章	嵌入式软件开发注意事项	4
1.1	配置字	4
1.2	开发环境	4
1.3	库函数	4
1.4	PLL 时钟	4
1.5	寄存器写保护	4
1.5.1	系统写保护	5
1.5.2	IAP 操作保护	5
1.5.3	IWDT 写保护	5
1.5.4	WWDT 写保护	5
1.6	IAP 操作程序	5
1.7	FLASH 读保护	5
1.8	GPIO 模块	6
1.9	IWDT 模块	6
1.10	WWDT 模块	6
1.11	I2C 模块	6
1.12	UART 模块	7
1.13	SPI 模块	7
1.14	ADC 模块	8
1.15	DMA 模块	9
1.16	低功耗模式	9
1.17	FLASH ECC 功能	9
1.17.1	ECC 使用效果	10
1.17.2	ECC 故障注入测试	10
1.18	LP16T 模块	12
1.19	ACMP 模块	13
1.20	OPA 模块	13
1.21	中断向量重映射功能	13
第 2 章	硬件开发注意事项	14
2.1	外部晶振	14
2.2	ES32H05x4 最小系统电路	15

第1章 嵌入式软件开发注意事项

1.1 配置字

芯片出厂时，配置字地址单元可能为空，在进行嵌入式软件开发前和量产烧录时，需要对目标芯片进行正确的配置字编程，避免配置字为空。用户使用 ES-Burner（PC 软件）+ES-LinkII（调试器）对目标芯片进行配置字编程。

注意：调试前必须将 GBRDP 设置为“读保护等级 Level0”，否则无法正常调试，量产烧录前可根据需求自行设置 GBRDP。

1.2 开发环境

推荐用户使用 Keil5 或者 IAR8.11 进行固件开发。由于 Keil4 不支持 PACK 机制，故不推荐用户使用 Keil4。

在使用 Keil5 或 IAR 进行开发前，需要使用 ESBurner+ES-LinkII 烧录合适的配置字选项，如：正确的启动地址（BOOT）、BORVS 配置到工作电压以下、看门狗关闭、FLASH 保护全部禁止。进行【擦除】→【配编】后才可以使使用 Keil5 或 IAR 进行程序调试。

1.3 库函数

ES32H05x4 系列芯片提供 MD 库函数：

- MD 库函数：提供外设模块初始化接口，以及寄存器级别的“读”、“写”接口，用户可方便地进行外设初始化操作。MD 库函数的时间和空间复杂度都比较精简，特别适合对性能和空间利用率要求高的系统。

1.4 PLL时钟

注意事项 1：PLL 支持 48MHz~80MHz 频率可配，当系统时钟大于 48MHz 时，需旁路系统时钟滤波器（CMU_CSR 的 CFT_CMD 位），否则可能会造成系统时钟失效。系统默认为旁路系统时钟滤波器。

注意事项 2：使能 FLASH ECC 纠错功能时，系统时钟最高支持 72MHz，并且此时 CPU 访问 Flash 会增加额外等待时间，整体运行速率会有降低。

注意事项 3：使用 PLL 时，需要先设置寄存器 ADC_CCR 的 IREFEN 位为 1 或设置寄存器 CMU_CLKENR 的 HRC48MEN 为 1，然后再使能 PLL 为 1。

1.5 寄存器写保护

为避免程序的异常导致运行错误，芯片写保护寄存器用于阻止对被保护的寄存器误操作。

系统控制单元，WDT 等模块支持寄存器写保护，对被保护的寄存器进行写之前需要解除写保护状态（允许写），否则无法对被写保护寄存器写入。操作完成后，再使能写保护（禁止写）。库函数中均提供相应宏定义进行解除保护和使能保护。

1.5.1 系统写保护

系统控制寄存器的访问操作会影响整个芯片的运行状态，芯片提供系统设置保护寄存器 SYSCFG_PROT。对 SYSCFG_PROT 寄存器以字方式写入 0x55AA6996 会解除写保护，对该寄存器写入其他任何值都会使能写保护。

可通过读 SYSCFG_PROT 寄存器确认写保护状态，读出值为 0x1，表示当前处于写保护状态；读出值为 0x0 表示当前处于写保护解除状态。

SYSCFG_PROT 保护的寄存器为除 SYSCFG_PROT 寄存器外的 SYSCFG、PMU、CMU、RMU 模块所有寄存器。

1.5.2 IAP 操作保护

软件通过写 MSC_FLASHKEY 寄存器，可解除对程序区的 IAP 操作保护，处于保护状态时，无法进行擦除和编程的操作。通过检查 MSC_FLASHKEY.STATUS 是否为 0，判断 Flash 是否处于保护状态。

1.5.3 IWDT写保护

对 IWDT_LOCK 寄存器以字方式写入 0x1ACCE551 会解除写保护，写入其他值使能写保护。

可通过读 IWDT_LOCK 寄存器确认 IWDT 模块是否处于写保护状态，读出值为 0x1，表示当前处于写保护状态；读出值为 0x0 表示 IWDT 模块处于写保护解除状态。

该寄存器保护除自身外的 IWDT 所有寄存器。

1.5.4 WWDT写保护

对 WWDT_LOCK 寄存器以字方式写入 0x1ACCE551 会解除写保护，写入其他值使能写保护。

可通过读 WWDT_LOCK 寄存器确认 WWDT 模块是否处于写保护状态，读出值为 0x1，表示当前处于写保护状态；读出值为 0x0 表示 WWDT 模块处于写保护解除状态。

该寄存器保护除自身外的 WWDT 所有寄存器。

1.6 IAP操作程序

注意事项 1: IAP 操作时会占用 FLASH 总线，芯片内置 ROM 固化 IAP 操作函数，由硬件电路实现。IAP 操作既可放在 SRAM 中执行寄存器操作，也可以在 FLASH 中调用 ROM 固化的 IAP 操作函数。推荐用户调用内置 ROM 固化的 IAP 操作函数，以减少 SRAM 中的 IAP 操作代码量。

注意事项 2: 在进行 FLASH 编程时，无论是否编写相同的数据，在 FLASH 编程前均必须先进行擦除。禁止通过对一个 word 的多次写入实现按 byte 或按 bit 修改。

1.7 FLASH读保护

当 FLASH 的读保护级别设置为 Level1 或 Level2 时，运行在 SRAM 中的程序不能有读 FLASH 的操作。

典型应用 1: 运行在 SRAM 中的程序如需读取 FLASH，将读 FLASH 操作放在 FLASH 中执行。

典型应用 2: 程序运行在 SRAM 中，响应中断请求。将中断向量表拷贝至 SRAM 中，并设置中断向量偏移地址（SYSCFG_VTOR 和 SYSCFG_MEMRMP.VTOEN）。

1.8 GPIO 模块

注意事项 1: 未使用的 GPIO 管脚（包括调试管脚）建议设置为输出低电平，若设置为输入，须加上拉或下拉电阻接到电源或地。

注意事项 2: IO 高速（20MHz 以上）需设定强电流驱动模式 1，即相应的 GPIO_ODRV 寄存器位设置为 01。PD0~PD8 端口输出仅支持普通驱动模式，不支持强驱动模式。

注意事项 3: 建议 VDD 上电时间在 3ms 以上，这样 IO 引脚在上电过程中不会出现明显的瞬态高脉冲（超过 0.5V）。

注意事项 4: 如果 VDD 上电时间小于 1.5ms，大部分 IO 引脚可通过外加 10K Ω 下拉电阻（对 PC0 和 PD1，则建议下拉电阻小于 5K Ω ），将 IO 引脚的瞬态脉冲幅度拉低至 0.3V 以下；VDD 上电时间在 1.5ms~3ms 之间时，所有 IO 都可通过外加 10K Ω 下拉电阻消除上电过程中的瞬态高脉冲。

1.9 IWDG 模块

注意事项 1: 通过配置字使能 IWDG 为硬件看门狗（硬件强制使能），可使 IWDG 脱离软件配置，系统则会更可靠。

注意事项 2: 当 IWDG 作为软件看门狗使用时，配置字 IWDTEN 配置成“由软件使能”，并通过寄存器对 IWDG 进行初始化。

注意事项 3: IWDG 如果通过配置字使能为硬件看门狗，则计数时钟固定为 LRC 时钟，上电默认的溢出周期最小值约 0.5 秒，用户可通过程序修改 IWDG_LOAD 寄存器来调整计数周期。IWDG 模块固定为使能，IWDG 复位和中断也固定为使能，软件无法关闭，寄存器 IWDG_CON 的 CLKS, RSTEN, IE, EN 位均无效。

注意事项 4: 可靠的系统在休眠下也应该保持 IWDG 处于工作状态，用户可以使用 IWDG 或 WWDG 定时唤醒后对 IWDG 进行喂狗。

1.10 WWDG 模块

注意事项 1: WWDG 禁止用户在喂狗窗口外喂狗，建议用户在 WWDG 中断服务程序内喂狗。或使用一个全局变量做为软件标志，在中断服务程序里置位软件标志，在主循环里判断软件标志，如果软件标志被置位再喂狗。

1.11 I2C 模块

注意事项 1: 最大传输 NBYTE

I2C 支持最大 $2^{16}=65536$ Bytes 的传输数量，分别配置 I2C_CON1.NBYTES 与

I2C_CON2.NBYTES 寄存器; 当传输数量需求大于 65536 时, 在传输前配置 I2C_CON2.RELOAD 位, 当第一笔 65536 Bytes 传输完后, SCL 将被拉低, 此时可重新填写第二笔传输的 NBYTES 数量; 当不使用 NBYTES 重载模式时, 需将 I2C_CON2.RELOAD 清除, 如果开启 I2C_CON2.AUTOEND 功能, I2C 会在 I2C_CON2.RELOAD 清除后, 自动发送 STOP 信号。

注意事项 2: 中断标志位处理方式

发送空中断标志位 TXE 置起后, 当状态寄存器 STAT 的 TXE 位为 1 时, 清除中断标志位无效。所以, TXE 置起后, 必须先填写 I2C_TXDATA, 方可清除该中断标志位。

接收非空中断标志位 RXNE 置起后, 当状态寄存器 STAT 的 RXNE 位为 1 时, 清除中断标志位无效。所以, RXNE 置起后, 必须先读空 I2C_RXDATA 里的数据, 方可清除该中断标志位。

注意事项 3: 当总线 SDA 遇到异常下拉的波形时, 避免传输状态错误的处理方式

请开启检测停止中断功能 (I2C_IER.STOP), 当 STOP 中断事件发生时, 检查 I2C_STAT.BUSY 是否为 1。如 BUSY 为 1, 关闭 I2C_CON1.PE, 再开启 I2C_CON1.PE。

如果上述状态发生之前, 已设置 I2C_CON2.START, 则必须重新设置一次。

1. 12 UART 模块

注意事项 1: 自动波特率

若发生侦测波特率超时(UART_RIF.ABTO), 波特率开关会自动清除(UART_MCON.ABREN), 建议启用自动波特率侦测时, 同时启用重复侦测自动波特率(UART_MCON.ABRREPT)功能。若自动波特率侦测成功, 但数据非预期时, 用户需重新使能自动波特率侦测(UART_MCON.ABREN)功能。

注意事项 2: 中断标志位处理方式

发送空中断标志位 TFEMPTY 置起后, 当状态寄存器 STAT 的 TFEMPTY 位为 1 时, 清除中断标志位无效。所以, TFEMPTY 置起后, 必须先填写 UART_TXBUF, 方可清除该中断标志位。

接收非空中断标志位 RFEMPTY 置起后, 当状态寄存器 STAT 的 RFEMPTY 位为 1 时, 清除中断标志位无效。所以, RFEMPTY 置起后, 必须先读空 UART_RXBUF 里的数据, 方可清除该中断标志位。

1. 13 SPI 模块

注意事项 1: 发送 FIFO 缓存中断处理方式

SPI 发送 FIFO 中断标志分为两种, 发送 FIFO 缓存空中断(SPI_IFM.TXE)与发送 FIFO 缓存低于阈值中断(SPI_IFM.TXTH)。当进入中断处理流程时, 务必优先填写 SPI_DATA 再清除中断标志位。根据不同的应用, 为了避免填写 SPI_DATA 时, 发生 FIFO 溢出的错误操作, 提供以下两种方式检查:

- 方法 1: 确认 FIFO 剩余空间, 读取 SPI_STAT.TXFLV 寄存器, 表示 FIFO 已填入数据数量 (Bytes), 可判断再填入多少数据。
- 方法 2: 确认 FIFO 是否已满, 读取 SPI_STAT.TXF 寄存器, 表示 FIFO 已满, 不可再填入数据。

注意事项 2: 接收 FIFO 缓存中断处理方式

SPI 接收 FIFO 中断标志分为三种, 接收 FIFO 缓存非空中断(SPI_IFM.RXNE)、接收 FIFO 缓存满中断(SPI_IFM.RXF)与接收 FIFO 缓存超过阈值中断(SPI_IFM.RXTH)。FIFO 功能主要为了提升 SPI 传输效率, 因此大多使用 RXF 与 RXTH 中断事件, 效率比较显著; 当进入中断处理流程时, 务必优先读取(搬移)SPI_DATA, 避免 FIFO 满溢造成数据丢失, 提供以下两种方式检查:

- 方法 1: 确认 FIFO 缓存空间, 读取 SPI_STAT.RXFLV 寄存器, 表示 FIFO 接收 Bytes 数量, 可判断读取多少数据。
- 方法 2: 确认 FIFO 是否已空, 读取 SPI_DATA 数据, 判断 SPI_STAT.RXNE 寄存器为 0, 表示 FIFO 已空, 数据已读取完毕。

注意事项 3: 从机模式下, 发送数据操作方法

SPI 配置为从机模式时, 填写 SPI_DATA 数据的依据, 建议使用发送 FIFO 缓存低于阈值状态标志位(SPI_STAT.TXTH)或中断(SPI_IFM.TXTH)。

注意事项 4: 高速 SPI 主机配置

当 SPI 作为主机, SCK 达到 18MHz 以上时, 建议如下:

- ①SPI 通信管脚对应的 IO 配置为输出高电平为强电流驱动, 输出低电平为普通电流驱动, 即设置 SPI 时钟和数据管脚所对应寄存器 GPIO_ODRV 的 ODRVy<1:0>=01;
- ②SPI 通信管脚的负载电容小于 30pF。

注意事项 5: SPI 主机单工只接收模式

SPI 单工只接收模式效率较高, 若接收数据传输不及时, 容易造成接收 FIFO 溢出, 如需使用该模式, 有如下注意事项:

- ①须配合使用 DMA 搬运接收数据;
- ②此模式下无需读取 BUSY 标志, 因为开始通信后此标志一直为 1。

1. 14 ADC模块

注意事项 1: 因 ADC 在使能后需要一定的建立时间(小于 1us), 才能正常工作, 所以在 ADCEN 使能后需延迟至少 1us 再进行第一次转换, 否则转换结果可能不准确。

注意事项 2: 内部基准电压 VREF1.2V 工作时, 需使能 IREFEN=1, 否则会导致 ADC 工作异常; 在 STOP 模式下 ADC 不工作, 需设置 IREFEN=0, 否则会增大芯片功耗。

注意事项 3: ADC 工作时, 需设置 PWRMODSEL=0, 选择为高速模式, 提高转换结果精度。

注意事项 4: ADC 时钟分频倍数由低到高切换时无延迟，但由高到低（2 分频除外）切换时（例如由 32 分频切换到 4 分频）可能会有最多 1024 个 ADC 原始时钟（HCLK）周期的延迟，在该延迟时间内，ADC 采样和转换过程暂停；应用中为了避免该延迟，当分频倍数由高到低切换时，推荐先切换至 2 分频，然后再切换到目标分频倍数。

注意事项 5: 在采样时间内及其之后的 2 个 T_{adclk} 周期内，ADC 输入通道信号电压需保持稳定，否则会影响采样结果。

注意事项 6: 使用 ADC 内部参考电压 V_{REFINT} 作为模拟通道 18 时，ADC 时钟频率需不超过 5MHz，采样时间需设为大于 3 μ s。

注意事项 7: ADC 插入通道模式建议使用 MODF 快速模式（OPA_CON.MODF），可额外增加 12 个 ADC 采样周期，提升 ADC 转换稳定性。

1. 15 DMA模块

注意事项 1: DMA 无法直接搬运 Flash 区域的数据。

注意事项 2: DMA 控制器工作在外设到外设模式时，需将仲裁前的 DMA 传输次数设置为 1，即设置 DMA_CONn 寄存器的 $MAX_BURST<3:0>=0$ ，并且对所涉及的每个外设，只能使能该外设的一个 DMA 请求功能。

1. 16 低功耗模式

注意事项 1: 芯片执行 WFI 进入低功耗模式前：

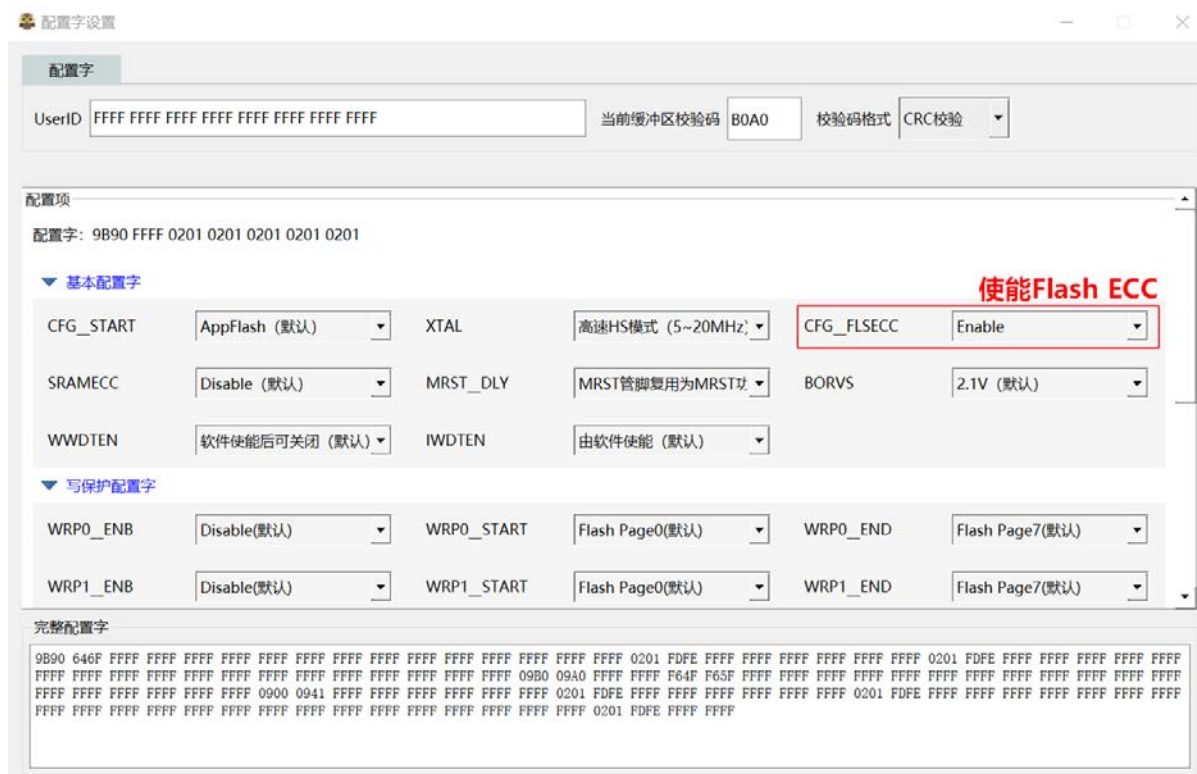
- 如果 SysTick 为使能状态，需要临时关闭 SysTick 计数器使能位(ENABLE@SysTick->CTRL)；
- 如果有非唤醒源的中断为使能状态，需要临时禁止对应的 IRQ 中断 (NVIC_DisableIRQ(XXX_IRQn))。

注意事项 2: 在 STOP 模式下如果不使用 PLL，内部基准电压 VREF1.2V 和模拟比较器 ACMP0~2 的高速模式，需设置 IREFEN=0，否则会增大芯片功耗。

1. 17 FLASH ECC功能

使用 Flash ECC 纠错功能时，必须要在配置字中使能 Flash ECC，其他配置字按需设置，确定后点击“配编”，配编成功后，ECC 方能生效。

注意事项 1: 在使能 Flash ECC 功能的情况下，程序不能在 SRAM 中运行，否则可能发生运行异常。在 Flash 非 ECC 模式下，SRAM 运行程序没有影响。



1. 17. 1 ECC使用效果

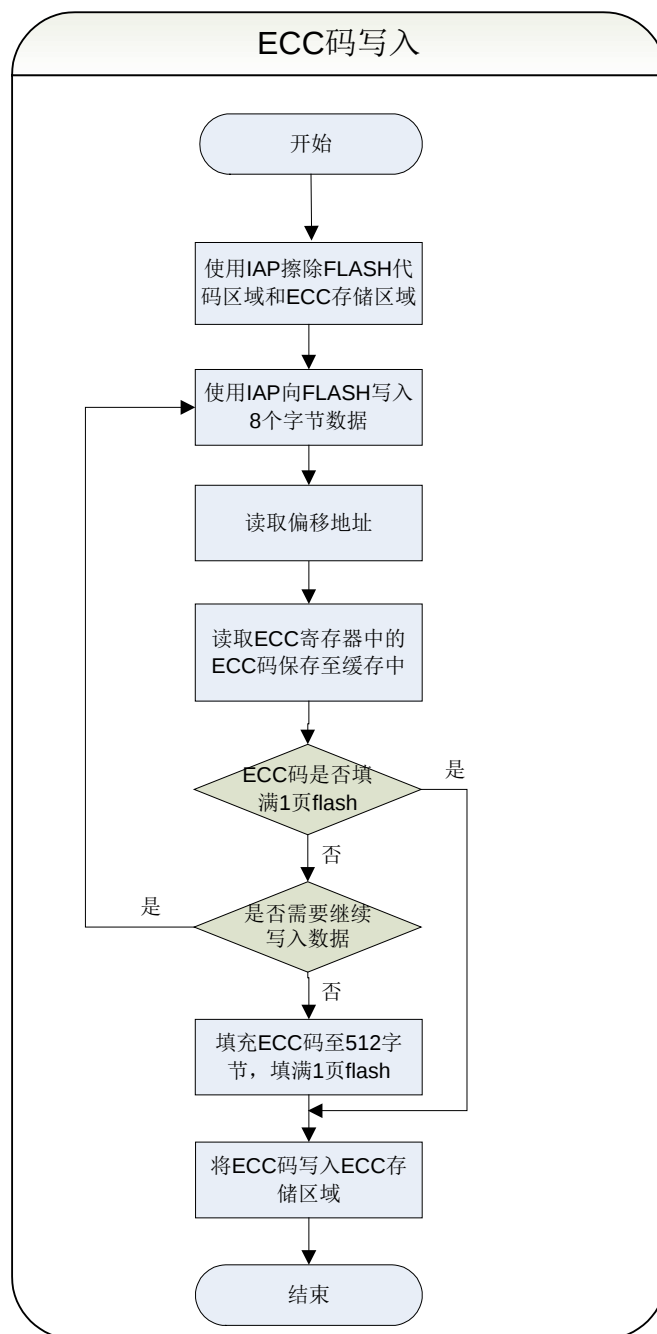
按照上述方法使能 ECC 后, ECC 便生效, IDE 和 ESBurner 均会自动适配 ECC 烧录算法, 正常烧录程序即可。若程序在运行过程中出现 1bit 数据错误, ECC 会自动纠错, 若同时使能了 ECC 中断 (MSC_FLASHCR.FLSECIE = 1) 且中断发生条件选择位为 0 (MSC_FLASHCR.FLSECIFS = 0), 则程序响应 ECC 中断。若程序在运行过程中出现 2bit 数据错误, ECC 不会自动纠错, 但 FLASH ECC 状态位会置起 (MSC_FLASHSR.FLSECIF = 2 或 3), 若同时使能了 ECC 中断

(MSC_FLASHCR.FLSECIE = 1) 且中断发生条件选择位为 1 (MSC_FLASHCR.FLSECIFS = 1), 则程序响应 ECC 中断。

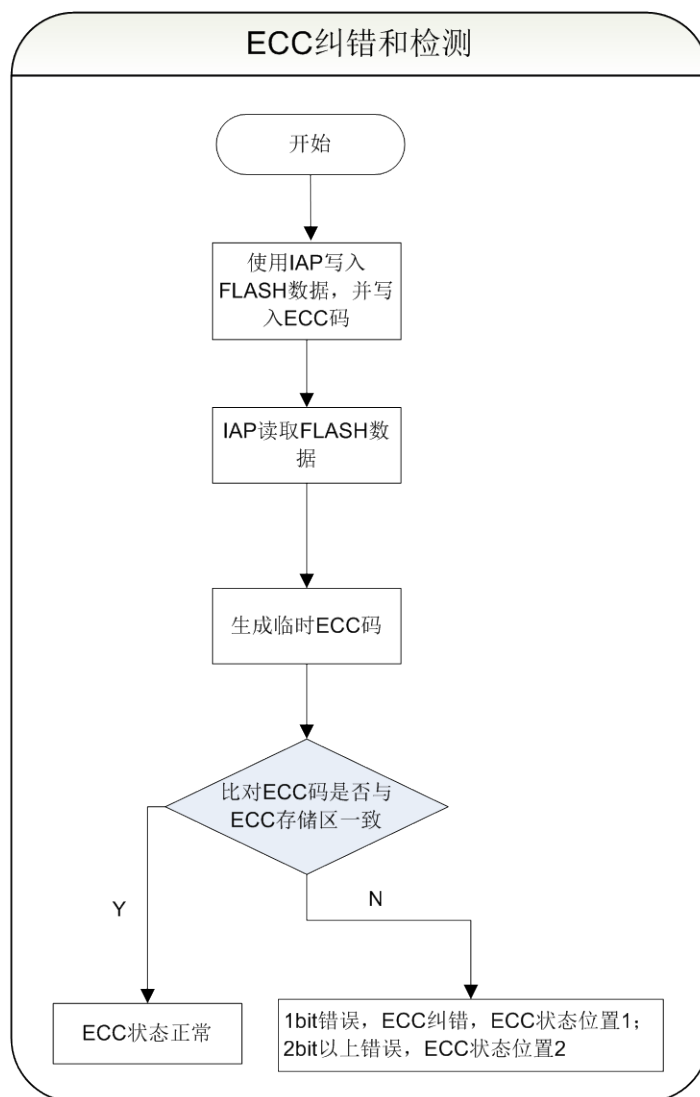
1. 17. 2 ECC故障注入测试

如需验证 ECC 功能是否有效, 可按如下方法进行故障注入测试。

ECC 启动后, 按 8 比 1 字节数写入 ECC 码到 ECC 存储地址, 每 8 个字节数据产生 1 个字节的 ECC 码, 通过 IAP 按字写入, 即 32 个字节产生 1 个字的 ECC 码, 才能通过 IAP 写入到 flash, 所以可以考虑使用 4K 字节对齐方式, 每 4K 字节产生 1 页的 ECC 码。在 IAP 写入一个 8 字节后, 需要去对应 flash 偏移地址中读取写入的数据, ECC 寄存器才会产生对应 ECC 码, 即每写 8 个字节, 读一次偏移地址, 再读 SCU 中对应 ECC 寄存器的 code。

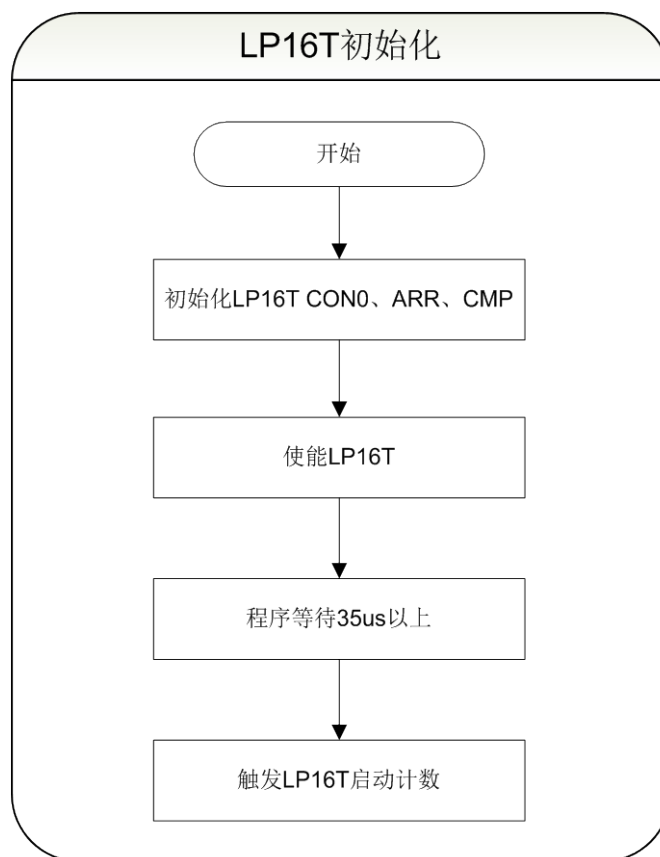


当ECC码和数据都写入flash后,可以对ECC纠错和检测错误功能进行验证,如在0x0000 5000写入数据,在ECC存储区域0x0003 8A00写完数据后,如果重新对0x0000 5000地址IAP写入数据,但是没对0x0003 8A00更改对应ECC码,在这个条件下,去读取0x0000 5000数据,就会触发ECC检测错误和纠错机制,因为读取0x0000 5000数据时,芯片对0x0000 5000地址生成临时ECC码并和ECC存储区域0x0003 8A00的ECC码比对,如果比对不正确,会发生ECC纠错和ECC错误状态置位。



1. 18 LP16T模块

LP16T 为低功耗定时器，在深度休眠模式下可以利用 LP16T 来唤醒 MCU。初始化 LP16T，先通过 CMU_CFGR.LPTCLKS 选择 LP16T 的时钟，并通过 CMU_CFGR.LPTCLKEN 使能该时钟，然后配置 LP16T 的相关参数，重载值、比较值、分频系数和工作模式。如果 LP16T 选择了 LRC 时钟，但 MCU 主时钟是 HRC 或 PLL，LP16T 使能后，需要等待 1 个 LRC 时钟周期再启动计数，软件等待时间需大于 35us。



1. 19 ACMP模块

注意事项 1: 在使用模拟比较器 ACMP 的高速模式时,需要先设置寄存器 ADC_CCR 的 IREFEN 位为 1,然后再设置模拟比较器的运行模式选择位 MODSEL 为 1;使用 ACMP 低速模式时,可设置 IREFEN 位为 0,以降低功耗。

注意事项 2: 当模拟比较器 ACMP 的负端输入选择为内部基准电压 VREF_1.2V 时,需要设置寄存器 ADC_CCR 的内部偏置电流和基准电压使能位 IREFEN =1。

1. 20 OPA模块

注意事项 1: 在内部增益放大 PGA 双端输入模式下,推荐将负端偏置电压选择为 VINN3 或 VREFN_IO,即设置 OPA_CON.VREFN_SEL=0x3 或 0x4,可提高增益放大倍数精度。

1. 21 中断向量重映射功能

注意事项 1: 在使能 Flash ECC 功能的情况下,不能使用中断向量重映射功能。在未使能 Flash ECC 功能的情况下,需在 SRAM 运行时使能中断向量重映射功能(即设置 SYSCFG_MEMRMP 的 VTOEN 为 1),且使能后程序只能运行在 SRAM,在禁止 VTOEN 后程序才能切回 Flash 运行,若程序在 Flash 中运行时使能 VTOEN 会发生异常。

第2章 硬件开发注意事项

2.1 外部晶振

外部振荡器的典型应用连接:

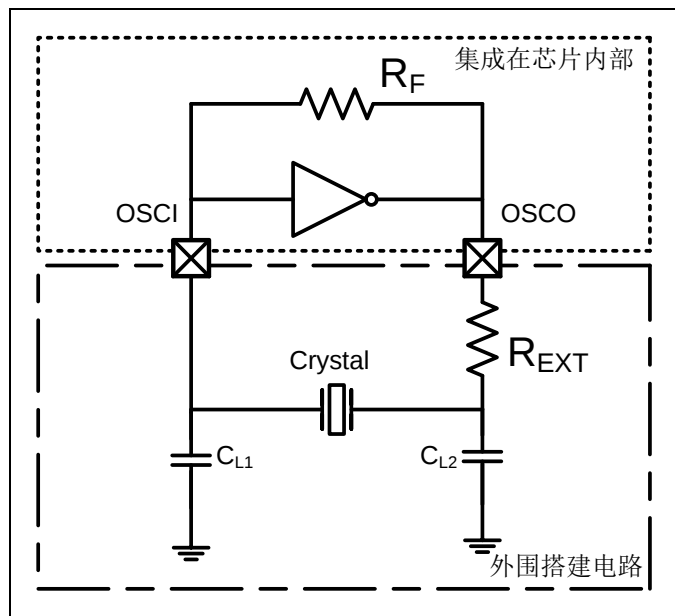
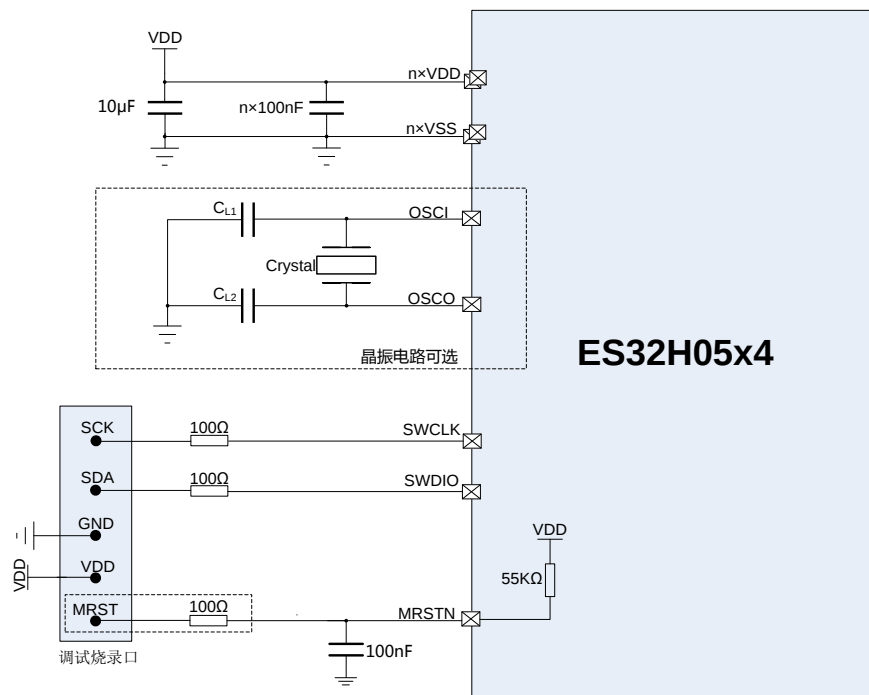


图 2-1 外部振荡器连接图

注意事项 1: R_{EXT} 阻值取决于晶振规格特性，为可选配置；

注意事项 2: 为达到理想的晶振起振和稳定工作状态，对 4~8MHz 晶振选型时，推荐 $ESR \leq 200 \Omega$ ， $C_L \leq 16pF$ （晶振的负载电容参数 $\leq 16pF$ ）；对 9~20MHz 晶振选型时，推荐 $ESR \leq 50 \Omega$ ， $C_L \leq 16pF$ ；对 32.768KHz 晶振选型时，推荐 $ESR \leq 40K \Omega$ ， $C_L \leq 12pF$ 。

2.2 ES32H05x4 最小系统电路



注 1: SWD 接口支持 4 线模式。

注 2: 芯片使用内部高速时钟 HRC 作为主系统时钟时，不需要外部晶振电路。