

赛普拉斯 HyperFlash 系列编程指南

作者：Zhi Feng

相关器件系列：HyperFlash

相关代码示例：无

赛普拉斯 HyperFlash 存储器使用少引脚、高性能的 HyperBus™ 接口。它们是业界最快的 NOR 闪存器件，在 HyperBus 接口上传输速率高达 333 Mbytes 每秒。本文档向软件程序员和系统工程师介绍了使用 HyperFlash 系列的方法。

目录

1	简介	1	8	编程	7
2	HyperFlash 的基本特性	1	9	擦除	7
2.1	HyperFlash 系统框图	1	10	安全区域 (SSR)	7
2.2	存储器架构	2	11	INT# 输出引脚	8
2.3	与传统并行 NOR 器件比较	2	12	结论	8
2.4	与 SPI NOR 器件的比较	2	13	参考	8
3	配置 HyperFlash 器件	3		文档修订记录	9
3.1	对存储器阵列进行第一次编程或擦除操作前	3		全球销售和设计支持	10
3.2	设置所有配置后	3		产品	10
4	状态寄存器	3		PSoC® 解决方案	10
5	复位	5		赛普拉斯开发者社区	10
6	CFI / 器件 ID	5		技术支持	10
7	最大化读取性能	6			

1 简介

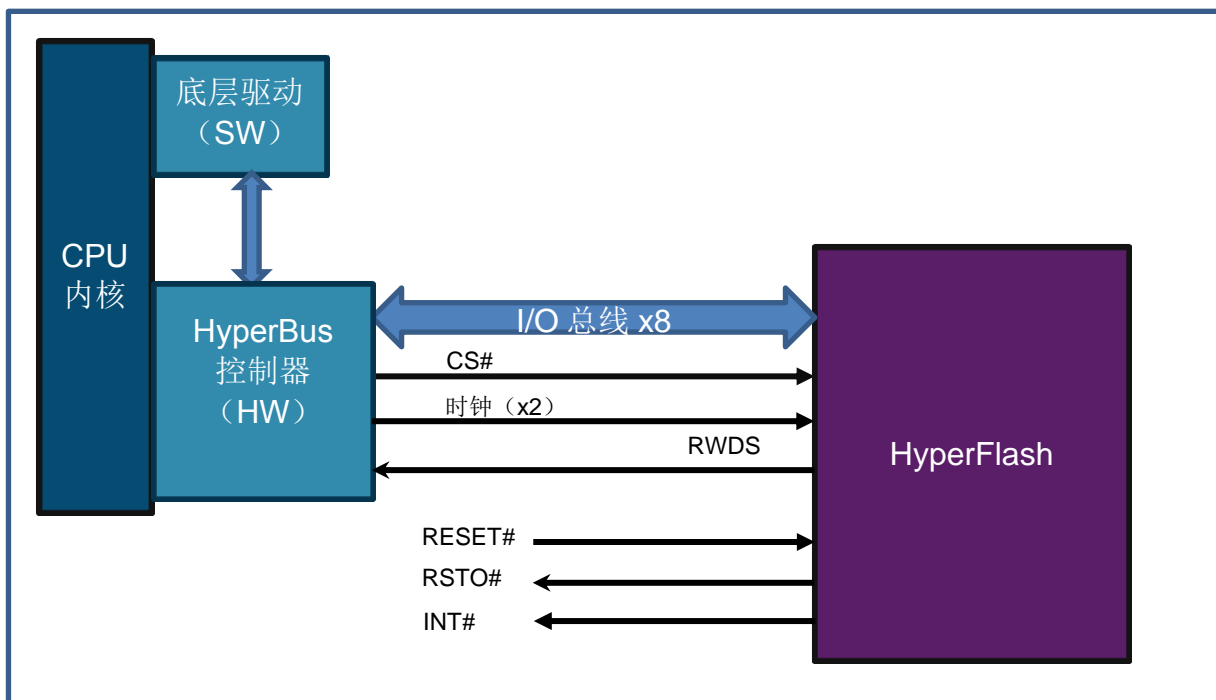
赛普拉斯 HyperFlash 存储器使用少引脚、高性能的 HyperBus™ 接口。它们是业界最快的 NOR 闪存器件，在 HyperBus 接口上传输速率高达 333 Mbytes 每秒。本文档向软件程序员和系统工程师介绍了使用 HyperFlash 器件的方法。要想了解 HyperFlash 器件的完整规范，请参考相应的数据手册。

2 HyperFlash 的基本特性

HyperFlash 系列包括 3.0 V KL-S 和 1.8 V KS-S 器件。3 V 和 1.8 V 器件均使用相同的软件命令集。HyperFlash 包含了如 SPI NOR 少信号引脚的优势，以及比并行 NOR 器件的更高的读性能。本节对 HyperFlash 与传统的 PNOR 和 SPI NOR 器件进行了比较。

2.1 HyperFlash 系统框图

下图显示的是一个典型的嵌入式系统芯片如何和 HyperFlash 进行连接。该连接需要 HyperBus 存储器控制器，通过控制引脚和 I/O 引脚与 HyperFlash 通信。底层驱动软件为上层应用提供了基本的读、编程和擦除功能。



2.2 存储器架构

HyperFlash 具有统一的扇区架构，扇区大小为 256 KB。在一个 FLASH 器件中，扇区是可擦除的最小单位。如果用户想要灵活使用更小的扇区，则可以通过用户配置选项来设置第一个或最后一个扇区。这些更小扇区被称为参数扇区，用户通常会将系统参数存储在这些扇区内。

HyperFlash 有一个写缓冲器的 RAM 区域，以 512 字节边界对齐。当使用 Write Buffer 对存储器进行编程时，HyperFlash 允许在单次操作中最多可编程 512 个字节。

通常，设备内部操作需要一段时间才能完成，如擦除扇区和编程，通常被称为嵌入式操作（EO）。在 EO 期间，器件处于 busy 态，并且大多数指令被禁止。用户可以使用读状态寄存器指令来确定 EO 是否完成。

这些读指令与页相关联。一页是以 16 字边界对齐的 16 个 word（32 字节）区域。在读操作跨越页边界时，可能会引入其他延迟。更多详细信息，请参考第 7 节。半页被定义为存储器的 16 字节，页的一半大小。在第 8 节中对半页进行了介绍，该节也说明了编程主题。

2.3 与传统并行 NOR 器件比较

HyperFlash 中的软件操作与传统 PNOR 器件（如赛普拉斯 GL-P 和 GL-S 系列）相同。虽然 HyperFlash 的物理引脚连接与 PNOR 器件的物理引脚连接不一样，但它将 PNOR 指令集作为它自己的基础指令集。

例如，要想擦除器件中的某个扇区，用户需要发送写指令序列（该序列与赛普拉斯 GL-P 和 GL-S 系列所使用的擦除指令序列相同）。HyperFlash 和 PNOR 之间的电气信号区别对软件是不可见的。主机系统中的 HyperFlash 存储器控制器会将软件的读写访问转换成 HyperBus 信号协议。因此，对 HyperFlash 设备，用户可以使用与 PNOR 相同的底层驱动。除了基础指令集外，HyperFlash 器件还有可选指令，使用这些指令可使能器件的其他特性。

2.4 与 SPI NOR 器件的比较

HyperFlash 和 SPI NOR 器件的信号引脚在封装上的物理位置是相同的。而 HyperFlash 中 I/O 总线上的指令格式与 SPI 器件中的完全不同。如上所述，从软件指令集角度来说，HyperFlash 与并行 NOR 器件向后兼容。这意味着不需要作任何更改仍可以在 HyperFlash 器件上重新使用 PNOR 底层软件驱动。

3 配置 HyperFlash 器件

HyperFlash 器件提供了一个非易失性配置寄存器（NVCR）以及对应的易失性配置寄存器（VCR）。VCR 适用于暂时更改配置设置以进行测试的用户。在下个电源周期内，VCR 值将被复位为 NVCR 中的值。如果需要非易失性配置，则 NVCR 应该更新所需数值。NVCR 和 VCR 均被称为 xVCR 寄存器。表 1 显示的是 xVCR 的位分配。

表 1. 易失性与非易失性配置寄存器

xVCR位	功能	设置（二进制）
xVCR.15	保留	1 — 保留（默认）
xVCR14 – xVCR12	驱动强度	000 — （默认）（请参考数据手册，了解器件的实际相关阻抗）
xVCR.11	xVCR冻结	0 — 锁定VCR或NVCR（不能对NVCR进行任何编程/擦除操作，VCR没有任何改变） 1 — 解锁VCR和NVCR（出厂默认设置）
xVCR.10	SSR冻结	0 — 锁定安全区域（禁止编程） 1 — 解锁安全区域（出厂默认设置）
xVCR.9 – xVCR.8	参数扇区映射	00 — 参数扇区和读取密码扇区被映射到最低地址 01 — 参数扇区和读取密码扇区被映射到最高地址 10 — 统一扇区和读取密码扇区被映射到最低地址（出厂默认设置） 11 — 统一扇区和读取密码扇区被映射到最高地址
xVCR.7 – xVCR.4	读延迟	1011 — 16个时钟周期延迟（出厂默认设置） （请参考数据手册，了解器件的实际相关阻抗）
xVCR.3	保留	1 — 保留（默认）
xVCR.2	保留	0 — 保留（默认）
xVCR.1 – xVCR.0	突发长度	00 — 保留 01 — 64个字节 10 — 16个字节 11 — 32个字节（出厂默认设置）

3.1 对存储器阵列进行第一次编程或擦除操作前

HyperFlash 系列具有统一的扇区架构，默认扇区大小为 256 KB。通过用户配置选项，可以把第一个扇区或最后一个扇区配置成 8 个 4 KB 的参数扇区。该配置由非易失性配置寄存器的 9-8 比特控制。请参考上表了解实际设置值。

如果用户想配置参数扇区，应该在对闪存阵列进行任何编程或擦除操作前将这两位编程为所需数值；否则，将丢失闪存阵列中覆盖部分的数据。

3.2 设置所有配置后

在确定和完全调试上电默认配置后，用户可以使用 NVCR 的位 11 来永久性锁定 xVCR 寄存器。将该位编程为 0 后，不能再对 xVCR 进行任何修改。该器件配置永久性被锁定。

4 状态寄存器

HyperFlash 具有一个单 16 位状态寄存器（SR），通过使用该寄存器可以检查器件当前的状态、嵌入式操作的状态或之前的擦除状态。只定义了状态寄存器的低 8 位。

表 2. 状态寄存器

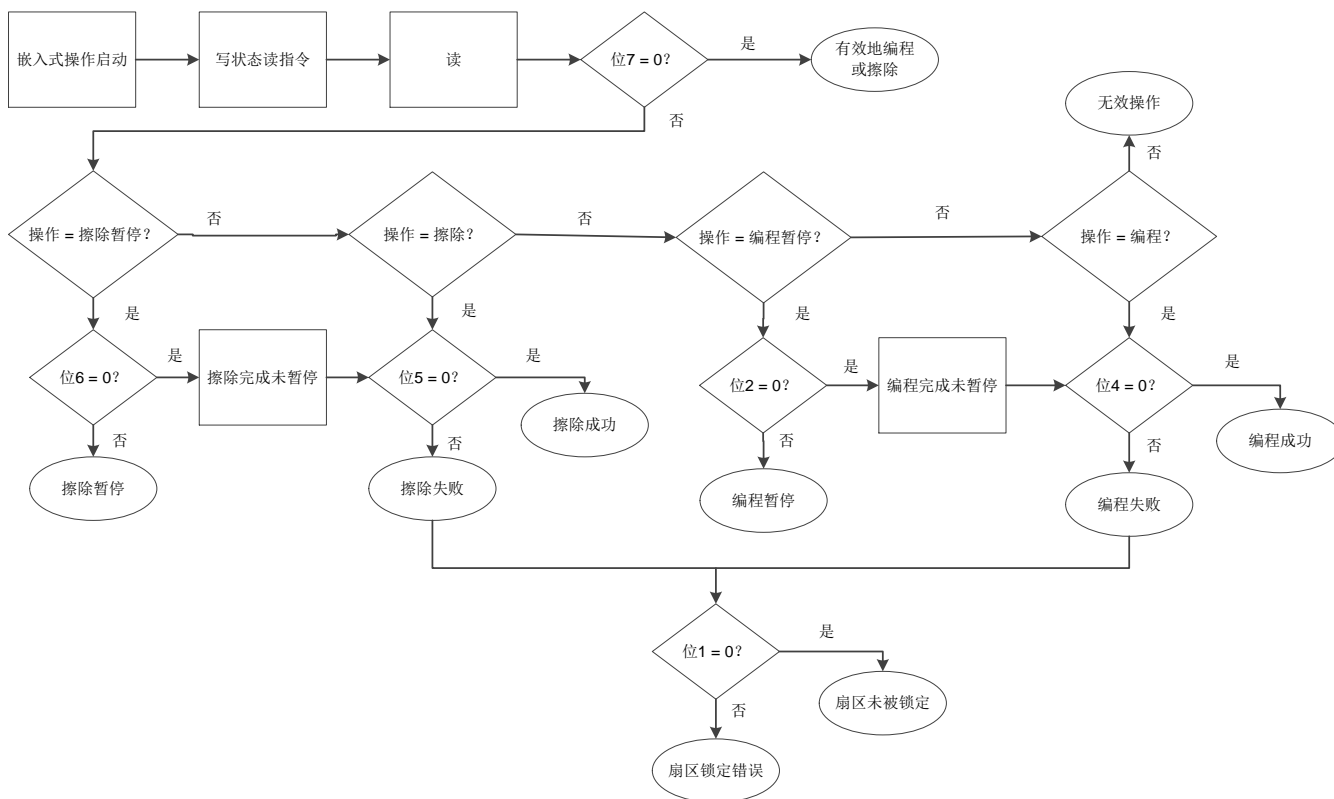
位编号	15:9	8	7	6	5	4	3	2	1	0
位说明	保留	保留	器件准备就绪位	擦除挂起状态位	擦除状态位	编程状态位	写入缓冲器终止状态位	编程挂起状态位	扇区锁定状态位	扇区擦除状态位
位名			DRB	ESSB	ESB	PSB	WBASB	PSSB	SLSB	ESTAT
复位状态	X	0	1	0	0	0	0	0	0	0
繁忙状态	无效	无效	0	无效	无效	无效	无效	无效	无效	无效
准备就绪状态	X	X	1	0 = 无擦除被挂起 1 = 擦除被挂起	0 = 擦除成功 1 = 擦除失败	0 = 编程成功 1 = 编程失败	0 = 编程未中止 1 = 编程操作在进行写入缓冲器指令期间被异常中止	0 = 无编程被挂起 1 = 编程被挂起	0 = 扇区在操作期间未被锁定 1 = 扇区锁定错误	0 = 扇区擦除状态指令结果 = 上次擦除没有完全成功 1 = 扇区擦除状态指令结果 = 上次擦除完全成功

在发送一个指令以触发嵌入式操作（如编程或擦除一个扇区），用户始终应该检查状态寄存器，以确保 FLASH 内部的操作已经完成，然后才能执行下一条指令。在进行嵌入式操作期间，只接受编程/擦除挂起¹指令或状态寄存器读取指令。所有其他指令均被忽略。

如果在进行嵌入式操作期间出现错误，则用户需要使用清除状态寄存器指令来清除错误位，然后才能执行下一条指令。

下图显示的是一个轮询函数，在启动嵌入式操作（如编程或擦除扇区）后，将使用状态寄存器来检查器件的状态。该算法假设最近执行的操作类型（显示在该图中间的菱形框内）被传递给软件轮询函数。这样便实现简化了软件流程。可以设计一个算法而不需要知道它正在轮询的操作类型，但该操作将更加复杂。

¹ 在进行编程或擦除操作期间，用户会发送一个挂起指令以挂起 EO，从而快速返回到读取模式。该操作一直被挂起，直到输入恢复指令为止。



5 复位

HyperFlash 器件中存在三种复位类型：上电复位（POR）（也被称为冷复位）、由 RESET#信号触发的硬件复位（也被称为热复位）以及由软件复位指令（F0h）触发的软件复位。

进行冷复位和热复位时，将从相应的 NVCR 位值加载所有 VCR 位。

可以在系统上使用 RSTO#引脚以表示 POR 的完成。器件完成上电过程，并经过用户定义的超时周期后，该引脚将从低电平状态转换到高电平状态。

软件复位通常用于清除状态寄存器或使器件从 ASO 状态或未知状态返回到读取阵列模式。软件复位不会影响正在进行的嵌入式操作或任何配置寄存器值。

6 CFI/器件 ID

同早期的 PNOR 器件相似，HyperFlash 系列遵循 JEDEC 通用闪存接口（CFI）规范，以提供一个标准化的数据结构。通过使用一条指令可以读取该结构。数据结构包含信息（如各种电气和时序参数）以及器件支持的特殊功能。这样，软件支持就变得与器件和器件 ID 无关，并且对整个闪存器件系列前后兼容。

通过使用 HyperFlash 器件中的 ID（自动选择）和 CFI 指令，用户可以访问复合的 ID/CFI 数据集。使用 ID（自动选择）或 CFI 进入指令序列后，可以访问 ID/CFI 数据集内的所有数据。

有关 ID/CFI 数据结构的完整说明，请参考 HyperFlash 系列数据手册（提供在本应用笔记的后面）。

7 最大化读取性能

在该数据手册中，HyperFlash 存储器阵列的初始访问时间被定义为 t_{ACC} 。需要在这段时间内将数据从闪存阵列传输到数据 I/O 内； t_{ACC} 独立于用于所选的时钟频率。时钟频率越高，需要的时钟周期越多，以便器件送出数据。用时钟周期表示的持续时间被称为延迟时钟。

在 HyperFlash 器件中，延迟时钟数量由四个延迟代码配置寄存器位（xVCR 位 7-4）控制。下表显示的是延迟代码、延迟时钟以及满足 t_{ACC} 初始读取延迟要求的最大时钟频率。

表 7.1 延迟设置

延迟代码	延迟时钟	最大工作频率 (MHz)
0000	5	52
0001	6	62
0010	7	72
0011	8	83
0100	9	93
0101	10	104
0110	11	114
0111	12	125
1000	13	135
1001	14	145
1010	15	156
1011	16	166
1100	保留	NA
1101	保留	NA
1110	保留	NA
1111	保留	NA

注意：

1. 器件出厂时，默认的NVCR延迟被设置为16个时钟周期。
2. 加载到（非）易失性配置寄存器位xVCR[7:4]内的数值便为延迟代码。
3. 假设最大的操作频率为 $t_{ACC} = 96 \text{ ns}$ 。

为了最大化读取性能，用户需要使用单个读指令序列来读取连续数据的最大长度，用于避免多初始延迟周期。这便表示最好使主控制器保持 CS# 为低电平，直到读取所有数据为止。为了实现该操作，用户不能使用 memcpy() 或类似的读取函数，一次读取一个字。用户需要实现控制器特定的函数，以便为应用级软件读取连续数据的多个字。该函数应该将数据长度作为一个参数，并且要具有所需要的主机存储器控制器特定寄存器设置，以便可使存储器控制器将正确的信号协议序列发送至 HyperFlash 存储器。

在 HyperFlash 中，一个完整的读取页为 32 个字节。用户可以将读操作的突发长度配置为 16、32、64 字节，或者一个线性的读序列。

如果将读操作配置为线性的，并且读地址没有从 16 字节的对齐边界（0h 或 8h 字节地址多偏移）开始，那么除了初始延迟外，用户需要插入一个页交叉延迟，然后才能越过第一个 32 字节对齐边界（到第二页）。初始和第一个页交叉延迟周期取决于时钟频率和起始地址偏移。请参考该数据手册中的列表，了解实际计数。

请注意，同一个线性访问中的所有后续页交叉不需要任何其他延迟时钟。对 16 字节和 32 字节突发进行的回卷读操作始终是在页范围内完成的，并不会因为超出页界限而引起延迟。对于一个 64 字节的回卷突发读取操作，根据具体的起始地址，在跨越页边界期间，可能需要插入另一个延迟。

8 编程

HyperFlash 器件系列有一个 512 字节的写缓冲器，该缓冲器与 512 字节边界对齐。按 512 字节长度及与此对齐的增量写入，是将数据编程到 FLASH 的最有效方式。虽然可以写入更小的数据，但为了使性能最佳，软件应该以完整、地址对齐、全缓冲增量的形式编程数据。

在 HyperFlash 中，闪存存储器阵列被组织为 16 字节长度，并以半页对齐。虽然不推荐在一个半页中进行多次编程操作，但依然允许这样的操作以便和传统的并行 NOR 产品互相兼容。在同一个半页上编程多次，每次擦除包含半页的扇区后，会降低该半页中的数据完整性。如果在一个擦除操作后将数据多次编程到半页内（该数据小于半页的大小），则建议为这些半页中的数据添加软件错误纠正信息。

例如，简单的闪存文件系统会编写 512 字节的文件记录，每个记录带有 30 字节的元数据。如果用户编程在 30 字节元数据后的下一个文件扇区，则所有后续扇区将不再对齐，如表 3 所示。在这种情况下，强烈建议用户在 30 字节元数据结构后面插入 2 字节的填充数据，以便使所有文件扇区和元数据与内部 16 字节半页和 512 字节写缓冲器页对齐。

表 3. 未对齐的数据存储

文件扇区	第一个记录（512 个字节）			第一个元数据（30 个字节）		第二个记录（512 个字节）			第二个元数据（#0 字节）	
闪存页	HP 0	...	HP 31	HP 32	HP 33	HP 34	...	HP 64	HP 65	HP 66

注意：HP：半页（16 个字节）

表 4. 对齐的数据存储

文件扇区	第一个扇区（512 个字节）			第一个元数据（30 个字节）+ 2 个填充字节		第二个扇区（512 个字节）			第二个元数据（30 个字节）+ 2 个填充字节	
闪存页	HP 0	...	HP 31	HP 32	HP 33	HP 34	...	HP 65	HP 66	HP 67

9 擦除

HyperFlash 器件系列具有统一的闪存大小（256 KB）。第一个（扇区 0）或最后一个扇区的某些部分可拆分成八个参数扇区（每个大小为 4 KB）。用户要注意，在这种交叉配置中，参数扇区和剩余的统一为 256 KB 扇区之间的参数扇区（大小为 224 KB）上面或下面存在一个扇区。当发送擦除指令时，该地址必须位于预期扇区内。

10 安全区域（SSR）

HyperFlash 器件系列提供了一个大小为 1024 个字节的安全硅区域（SSR），它是一个独立于主闪存阵列的一次性可编程（OTP）区域。该区域分为 32 个可独立锁定的 32 字节对齐区域（32 × 32 个字节 = 1024 个字节）。通过进入 SSR 地址空间重叠（ASO）模式，用户可以访问 SSR。

当读取 SSR 时，将进入 SSR ASO 模式，并对 SSR 地址偏移进行读取。如果输入地址超过了 1024 字节的 SSR 地址范围，将检索主阵列数据。

编程 SSR 时，将进入 SSR ASO 模式，并按照与通用阵列相同的方法编程。

SSR 由 xVCR 位 10 中的 FREEZE 位保护。如果设置了 FREEZE 位，将忽略编程指令。无错误被记录。

SSR 的区域 0（前 32 位）是一个特殊区域。区域 0 的前 16 个字节被保留，以便使赛普拉斯能够在随机数中进行编程（该随机数可以作为独特器件标识（如序列号）使用）。接下来的四个字节为锁定位。每个锁定位控制着相应的 32 个 SSR 区域（从区域 0 到区域 31）。通过编程这些位，用户可以单独锁定任何 SSR 区域。一旦锁定某个区域，它将永久被锁定。

如果尝试对随机数区域进行编程，会导致编程错误。如果 SSR 区域由它的锁定位锁定，则尝试对该区域进行编程会引起编程错误。

编程 SSR 时，编程页的大小与通用闪存阵列编程页的大小相同，都是 512 个字节。这意味着用户可以使用同一个编程指令来编程多个 SSR 区域。

如果输入的编程数据大于页大小，则该数据将回卷到该页的起始位置，同一个通用页编程指令一样。在这种情况下，回卷数据可能与区域 0 相重叠（区域 0 即为上述特殊区域）。这时，如果编程指令包含用于区域 0 前 16 个字节的数据，则该指令会失败。不建议加载超过页编程缓冲器结尾的数据

11 INT#输出引脚

HyperFlash 器件系列提供了一个 INT#输出引脚。用户可以配置该输出，当从繁忙状态转换到就绪状态时（例如，在完成嵌入式操作时），用于向系统生成一个中断。

通过读取中断状态寄存器（ISR），用户可以验证是否发生了“繁忙-就绪”事件，或者最后的 POR 是否成功完成，然后通过向相应位写入 1 来复位 ISR。

12 结论

赛普拉斯 HyperFlash 系列和 SPI 器件提供了相同的物理封装，而且 HyperFlash 系列和并行的 NOR 器件具有相同的软件指令接口。另外，它还将 SPI 器件的低引脚数量优势与 NOR 器件的高吞吐量相结合。实际上，它的性能甚至比并行的 NOR 器件更大。

HyperFlash 系列为使用 SPI NOR 或并行 NOR 器件的用户提供了简单的转换功能，从而降低了系统成本，但依然可以使系统具有更好的性能。

当使用 HyperFlash 器件系列时，请联系赛普拉斯客户支持获取其他帮助。

13 参考

- [S26KL/KSxxxS 数据手册](#)
- [S26KL/KSxxxS 底层驱动](#)

文档修订记录

文档标题： AN99195 — 赛普拉斯 HyperFlash 系列编程指南

文档编号： 002-00241

版本	ECN	变更者	提交日期	变更说明
**	4914887	SHWU	09/11/2015	本文档版本号为 Rev**，译自英文版 001-99195 Rev**。

全球销售和 design 支持

赛普拉斯公司维护由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。要想查找离您最近的办事处，请访问赛普拉斯所在地。

产品

汽车级产品	cypress.com/go/automotive
时钟与缓冲器	cypress.com/go/clocks
接口	cypress.com/go/interface
照明与电源控制	cypress.com/go/powerpsoc
存储器	cypress.com/go/memory
PSoC	cypress.com/go/psoc
触摸感应	cypress.com/go/touch
USB 控制器	cypress.com/go/usb
无线/射频	cypress.com/go/wireless

PSoC® 解决方案

psoc.cypress.com/solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#)

赛普拉斯开发者社区

[社区](#) | [论坛](#) | [博客](#) | [视频](#) | [培训](#)

技术支持

cypress.com/go/support

PSoC 是赛普拉斯半导体公司的注册商标，且 PSoC Creator 是赛普拉斯半导体公司的商标。此处引用的所有其他商标或注册商标归其各自所有者所有。



赛普拉斯半导体公司
198 Champion Court
San Jose, CA 95134-1709
电话 : 408-943-2600
传真 : 408-943-4730
网址 : www.cypress.com

©赛普拉斯半导体公司，2015。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品内嵌的电路外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯不保证产品能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于有可能因为发生功能异常和故障而对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

该源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途外，未经赛普拉斯明确的书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做出通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能因为发生功能异常和故障，而对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受限于赛普拉斯软件许可协议。