

带有实时时钟的 1 Mbit (128K x 8) 四线 SPI nvSRAM

特性

- 容量
 - 1 Mbit (128K x 8)
- 带宽
 - 提供速率为 108 MHz 的高速接口
 - 支持以 54 Mbps 的速度进行读写操作
- 串行外设接口
 - 时钟极性和相位模式 0 和 3
 - 多个 I/O 选项 — 单线 IO 模式中的 SPI (SPI)、双线 IO 模式下的 SPI (DPI) 以及四线 IO 模式下的 SPI (QPI)
- 可靠性较高
 - 无限次读、写和回读周期
 - 一百万次的存储周期, 用于将数据存储到 SONOS FLASH Quantum trap 中非易失性单元内
 - 数据保留时间: 温度为 85°C 时保留 20 年
- 读取
 - 命令: 正常、快速、双线 I/O 以及四线 I/O
 - 模式: 突发包、持续 (XIP)
- 写入
 - 命令: 正常、快速、双线 I/O 以及四线 I/O
 - 模式: 突发循环
- 数据保护
 - 硬件: 通过写保护引脚 (\overline{WP}) 提供保护功能
 - 软件: 通过写禁用指令提供保护
 - 块保护: 状态寄存器位用于控制保护
- 特殊指令
 - STORE/RECALL: 用于在 SRAM 和 Quantum Trap nvSRAM 间传输数据
 - 序列号: 用户可选的 8 字节 (OTP)
 - 标识号: 4 字节的制造商 ID 和产品 ID
- 支持将数据从 SRAM 存储到非易失性 SONOS FLASH Quantum Trap 内
 - 自动存储: 断电时, 通过使用小电容 (V_{CAP}) 自动存储数据
 - 软件: 使用 SPI 指令 (STORE)
 - 硬件: 使用 HSB 引脚进行存储
- 支持将数据从非易失性 SONOS FLASH Quantum Trap 回读到 SRAM 内
 - 自动回读: 加电时, 自动开始回读
 - 软件: 使用 SPI 指令 (RECALL) 进行回读
- 低功耗模式
 - 睡眠: 温度为 85 °C 时, 平均电流为 380 μ A
 - 休眠模式: 温度为 85 °C 时, 平均电流为 8 μ A
- 供电电压的工作范围
 - 内核电压 V_{CC} : 2.7 V ~ 3.6 V
 - I/O 电压 V_{CCQ} : 1.71 V ~ 2.0 V

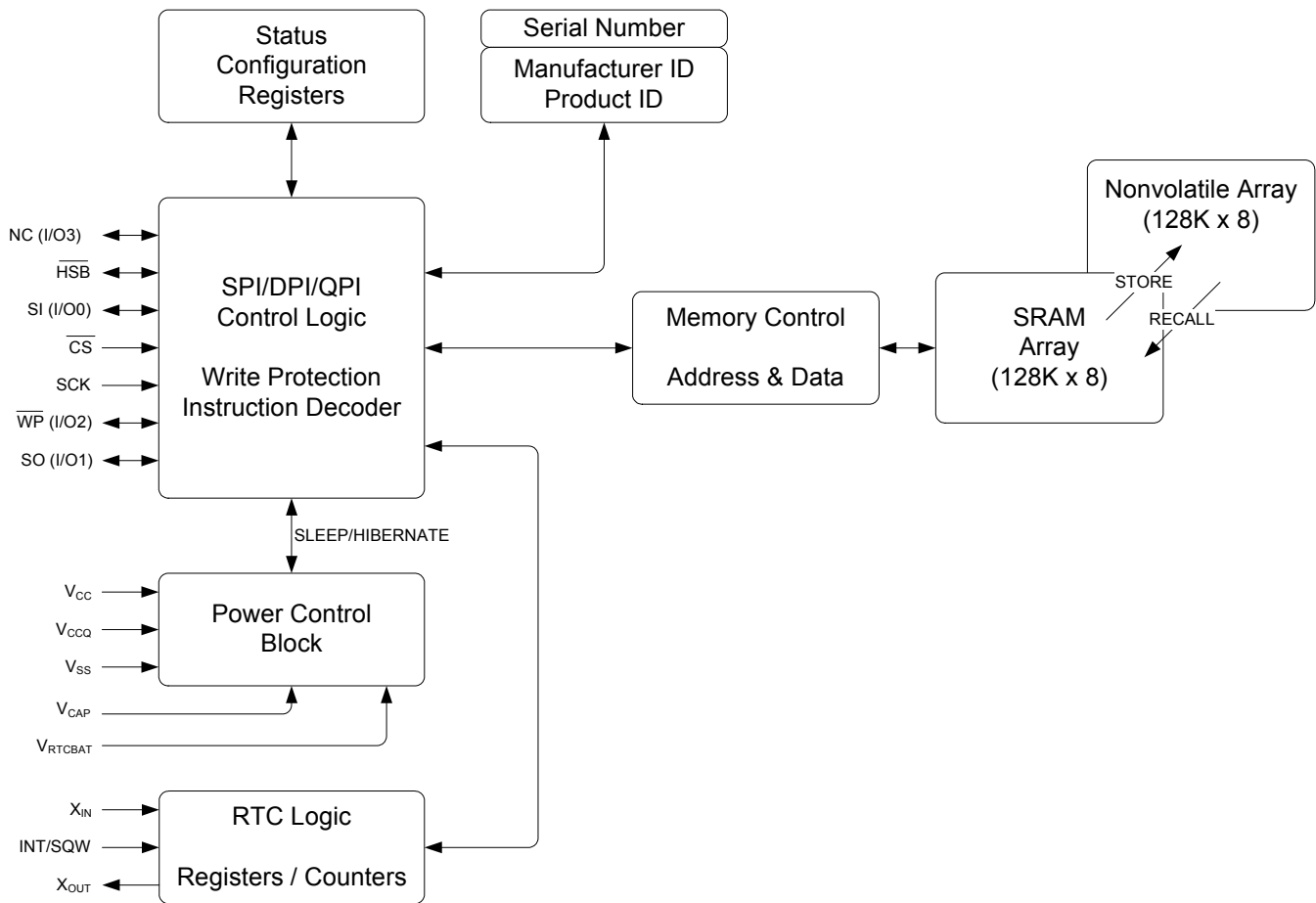
- 温度范围
 - 工业级: -40 °C~85 °C
- 封装
 - 16-SOIC

功能概述

赛普拉斯 CY14V101PS 将 1 Mbit nvSRAM 与 QPI 接口相结合。QPI 通过使用所选的操作码能够在单线模式 (在每个时钟周期内使用一个 I/O 通道传输一位)、双线模式 (在每个时钟周期内使用两个 I/O 通道传输两位) 或四线模式 (在每个时钟周期内使用四个 I/O 通道传输四位) 下对存储器进行读写操作。

存储器被组织为 128KB, 该存储器包括 SRAM 和非易失性 SONOS FLASH Quantum Trap 单元。SRAM 能够实现无限次的读写周期, 而非易失性单元则能够提供高可靠性的数据存储空间。断电时, 数据会自动从 SRAM 中转移到非易失性存储器中 (存储操作)。加电时, 数据会从非易失性存储器回读到 SRAM (回读操作) 内。用户可以通过 SPI 指令触发 “存储” 和 “回读” 操作。

逻辑框图

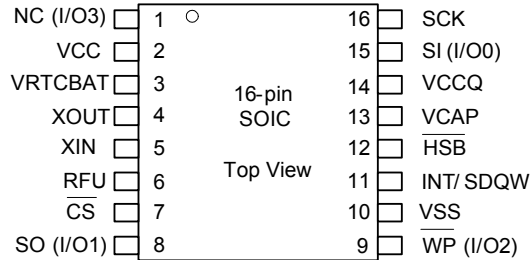


目录

引脚分布	4	自动存储使能 (ASEN) 指令	44
引脚定义	4	自动存储禁用 (ASDI) 指令	44
器件操作	6	实时时钟操作	45
SRAM 写入	6	nvTIME 操作	45
SRAM 读取	6	时钟操作	45
存储操作	6	读取时钟	45
自动存储操作	6	设置时钟	45
软件存储操作	7	备用电源	45
硬件存储和 HSB 引脚操作	7	停止和启动振荡器	45
回读操作	7	校准时钟	46
硬件回读 (加电)	7	警报	46
软件回读	7	看门狗定时器	46
禁用和使能自动存储	7	可编程方波生成器	47
四线串行外设接口	8	电源监视器	47
SPI 概述	8	备用电源监控器	47
双线和四线 I/O 模式	10	中断	47
SPI 模式	10	中断寄存器	47
SPI 操作功能	11	标志寄存器	48
加电	11	RTC 外部组件	49
断电	11	RTC 的 PCB 设计注意事项	50
有效功耗模式和待机模式	11	布局要求	50
SPI 功能说明	12	最大额定值	55
状态寄存器	14	工作范围	55
写入禁用 (WRDI) 指令	18	直流规范	55
写入使能 (WREN) 指令	18	数据保持时间与耐久性	56
使能 DPI (DPIEN) 指令	19	电容	56
使能 QPI (QPIEN) 指令	19	热阻	56
使能 SPI (SPIEN) 指令	19	交流测试负载和波形	57
SPI 存储器读取指令	20	交流测试条件	57
读指令	20	RTC 特性	57
快速读取指令	21	交流开关特性	58
写指令	24	开关波形	58
系统资源指令	28	自动存储或加电回读	59
软件复位 (RESET) 指令	28	切换波形	60
默认的恢复指令	29	软件控制的存储和回读周期	61
读取实时时钟 (RDRTC) 指令	29	切换波形	61
写入实时时钟 (WRRTC) 指令	31	硬件存储周期	62
休眠 (HIBEN) 指令	32	切换波形	62
睡眠 (SLEEP) 指令	33	订购信息	63
寄存器指令	35	订购代码定义	63
读取状态寄存器 (RDSR) 指令	35	封装图	64
写入状态寄存器 (WRSR) 指令	35	缩略语	65
读取配置寄存器 (RDCR) 指令	36	文档常规	65
写入配置寄存器 (WRCR) 指令	37	测量单位	65
标识寄存器 (RDID) 指令	38	文档修订记录	66
标识寄存器 (FAST_RDID) 指令	39	销售、解决方案和法律信息	67
序列号寄存器写入 (WRSN) 指令	40	全球销售和 design 支持	67
序列号寄存器读取 (RDSN) 指令	40	产品	67
快速读取序列号寄存器 (FAST_RDSN) 指令	42	PSoc® 解决方案	67
NV 特定指令	43	赛普拉斯开发者社区	67
软件存储 (STORE) 指令	43	技术支持	67
软件回读 (RECALL) 指令	43		

引脚分布

图 1. 16-SOIC 引脚分布



引脚定义

引脚名称	I/O 类型	说明
NC (I/O3)	输入	未连接。在单线或双线模式下，该引脚未被连接并处于悬空状态。这两种模式不支持使用 QSPI 指令。
	输入 / 输出	I/O3: 该器件处于四线模式时，NC (I/O3) 引脚将作为 I/O3 引脚执行输入 / 输出操作。 在支持 SPI/DPI 指令的四线模式下， \overline{CS} 被使能时，该引脚应处于三态。
V _{CCQ}	电源	供给器件上各 I/O 的电源电压。
V _{CC}	电源电压	供给器件内核的电源电压。
\overline{CS}	输入	芯片选择。将该引脚置于低电平时，会激活该器件。通过使该引脚置于高电平，可以使器件进入待机状态。
SO (I/O1)	输出	串行输出。用于通过 SPI 输出数据的引脚。
	输入 / 输出	I/O1: 当该器件处于双线模式或四线模式时，SO (I/O1) 引脚作为 I/O1 引脚执行输入 / 输出操作。
\overline{WP} (I/O2)	输入	写入保护。在 SPI/DPI 模式下实现硬件写保护。
	输入 / 输出	I/O2: 当该器件处于四线模式时， \overline{WP} (I/O2) 引脚将成为 I/O2 引脚执行输入 / 输出操作。
V _{SS}	接地	器件中内核和 I/O 的接地电压。
\overline{HSB}	输入 / 输出	硬件存储繁忙： 输出：该引脚为低电平时表示 nvSRAM 为繁忙状态。每次执行完硬件和软件存储后，都会使用标准输出高电流将 HSB 驱动为高电平一小段时间 (t_{HHHD})，然后使用一个较弱的内部上拉电阻使该引脚保持为高电平状态（可以选用外部上拉电阻连接）。 输入：通过外部将该引脚下拉为低电平，可以启动硬件存储。
V _{CAP}	电源	自动存储电容。断电期间为 nvSRAM 提供电源，以在该过程中将数据从 SRAM 存储到非易失性单元中。如果不需要自动存储，那么需要将该引脚置于“无连接”状态。该引脚绝对不能接地。
V _{RTCbat}	电源	RTC 的备用电池。
Xout	输出	晶振输出连接。如果不使用 RTC 功能，则必须让其保持未连接状态。
Xin	输入	晶振输入连接。如果不使用 RTC 功能，则必须让其保持未连接状态。
INT/SQW	输出	中断输出 / 校准 / 方波。可对此引脚进行编程，使之响应时钟警报、看门狗定时器以及电源监控器。此外，还可以将其设置为高电平（推或拉）或低电平（开漏）有效。在校准模式下，可输出 512 Hz 的方波。在方波模式下，用户可以选择 1 Hz、512 Hz、4096 Hz 或 32768 Hz 作为连续输出的频率。 如果不使用 RTC 功能，则必须让其保持未连接状态。

引脚定义 (续)

引脚名称	I/O 类型	说明
SI (I/O0)	输入	串行输入。用于输入所有 SPI 指令和数据的引脚。
	输入 / 输出	I/O0: 当该器件处于双线模式或四线模式时, SI (I/O0) 引脚将作为 I/O0 引脚执行输入 / 输出操作。
SCK	输入	串行时钟。以 f_{SCK} 的最大速度运行。在该时钟的上升沿上锁存串行输入。将在该时钟的下降沿上驱动串行输出。
NC	—	未连接。
RFU	—	留给将来使用

器件操作

CY14V101PS 是一个带 SONOS FLASH 非易失性元件并且大小为四线串行接口的 1 Mbit nvSRAM 存储器：在每个存储器单元中，SONOS FLASH 非易失性元件会与 SRAM 元件交叉存在。对 nvSRAM 进行的所有读和写操作都发生在 SRAM 上，它为 nvSRAM 提供了独特的处理无限次存储器写入操作的能力。SRAM 中的数据由 STORE 序列保护，该序列将数据并行传输到非易失性单元中。断电时，使用小电容 (V_{CAP}) 来将 SRAM 数据自动存储到非易失性单元中，从而确保数据的完整性。采用可靠的 SONOS 技术制造的非易失性单元，使 nvSRAM 成为保护数据存储的最理想选择。

这个 1 Mbit 的存储器阵列的组织方式为 128 KB。可以通过标准的 SPI 接口（单线模式、双线模式和四线模式）访问该存储器，该接口能够实现高达 40 MHz 的时钟速率，并且能够实现零周期延迟的读取和写入操作。该 SPI 接口还支持 108 MHz 的操作（单线模式、双线模式和四线模式），仅读取操作存在周期延迟。CY14V101PS 作为 SPI 从设备工作，并支持 SPI 模式 0 和 3（CPOL、CPHA = [0, 0] 和 [1, 1]）。在单线模式和双线模式下使用芯片选择（CS）、串行输入（SI）（I/O0）、串行输出（SO）（I/O1）和串行时钟（SCK）引脚执行所有指令。在四线模式下使用 WP（I/O2）和 I/O3 引脚来执行命令、寻址和数据输入等操作。

该器件使用 SPI 操作码执行存储器访问。操作码支持在 SPI、双线数据、双线地址 / 数据、双线 I/O、四线数据、四线地址 / 数据和四线 I/O 模式下进行读 / 写操作。此外，它还包含四条特殊指令，通过这四条特殊指令可以使用四项 nvSRAM 特定功能：STORE（存储）、RECALL（回读）、自动存储禁用（ASDI）和自动存储使能（ASEN）。

CY14V101PS 具有内置的数据安全功能。分别通过 \overline{WP} 引脚和 WRD1 指令提供了硬件和软件写保护。此外，存储器阵列块通过状态寄存器块保护位受到写保护。

SRAM 写入

对 nvSRAM 的所有写操作都在 SRAM 上执行的，不会消耗 SONOS FLASH 非易失性存储器的擦写次数。这允许您执行无限次的写入操作。通过下面一个写指令可以启动写周期：WRITE、DIW、QIW、DIOW 和 QIOW。写指令包括一个写操作码、三字节地址和一字节数据。以 SPI 总线速度完成对 nvSRAM 进行的写操作，并且周期延迟为零。

通过该器件可以执行突发模式写操作。这样，可以在连续的地址上启用写操作，而无需发出新的 WRITE 指令。在突发模式下到达存储器的最后一个地址时，地址将翻转为 0x00000，然后该器件继续执行写入操作。

第 12 页上的 SPI 功能说明中的 nvSRAM 读写指令部分对 SPI 写周期序列进行了明确的定义。

SRAM 读取

对 nvSRAM 的所有读取操作都是以 SPI 总线速度在 SRAM 单元上执行的。读指令（READ）以 40 MHz 的频率执行，其周期延迟为零。它包括一个读操作码字节，后面是三字节的地址。在数据输出引脚 / 各引脚上读取得到该数据。

速率高于 40 MHz（最多可达 108 MHz）时，需要快速读取指令：FAST_READ、DOR、QOR、DIOR 和 QIOR。快速读取指令包括：一个快速读取操作码字节、三个地址字节和一个虚拟 / 模式字节。在数据输出引脚 / 各引脚上读取得到该数据。

通过该器件可以执行突发模式读操作。这样便可以在连续的地址上执行读操作，而无需发出新的 READ 指令。在突发模式下到达存储器的最后一个地址时，地址将翻转为 0x00000，然后该器件继续执行读取操作。

第 12 页上的 SPI 功能说明中的 nvSRAM 读写指令部分对 SPI 读周期序列进行了明确的定义。

存储操作

存储操作将 SRAM 中的数据传送到非易失性单元内。通过以下三种存储操作中的一种，器件可将数据存储起来：自动存储（器件断电时激活（需要 V_{CAP} ））、软件存储（通过 STORE 指令激活）以及硬件存储（通过 HSB 引脚激活）。在存储周期内，先要擦除非易失性单元，然后再编程它。启动存储周期后，将禁止对器件进行读 / 写操作，直到周期完成为止。

系统将监控状态寄存器中的 \overline{HSB} 信号或 WIP 位，以检测是否在执行存储周期。通过将 \overline{HSB} 置于低电平或将 WIP 位置为 ‘1’，可以指示 nvSRAM 处于忙碌状态。为了避免不必要的非易失性存储，应忽略自动存储和硬件存储操作，除非在最新的存储周期后至少发生了一次 SRAM 写操作。但是，无论是否发生 SRAM 写操作，都会执行软件触发的存储周期。

自动存储操作

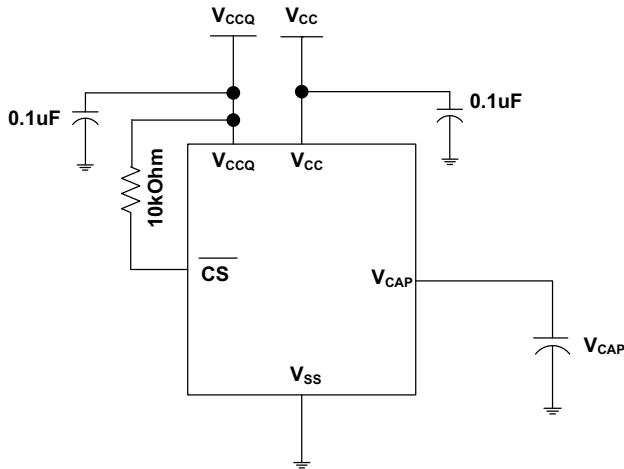
自动存储操作是 nvSRAM 独有的特性，该特性会在断电时自动将 SRAM 中的数据存储到 SONOS FLASH 非易失性单元内。断电时该存储操作利用外部电容（ V_{CAP} ）上所存的电能保证器件能够将数据安全存储到非易失性存储器中。

正常工作时，器件从 V_{CC} 得到电流，以给连接至 V_{CAP} 引脚的电容充电。断电时，如果 V_{CC} 引脚上的电压下降低于 V_{SWITCH} ，那么器件将禁止对 nvSRAM 进行的所有存储器访问，并通过使用来自 V_{CAP} 电容的电荷自动执行存储操作。如果在上一次回读之后未执行写入周期，则不启动自动存储操作。

注意：如果 V_{CAP} 引脚上没有连接电容，则必须通过发出自动存储禁用指令（第 44 页上的自动存储禁用（ASDI）指令）来禁用自动存储。如果 V_{CAP} 引脚上没有电容时启用了自动存储，则器件将在没有足够电荷的情况下尝试进行自动存储操作，以完成存储操作。这样会损坏存储在存储器阵列中的数据、序列号寄存器和状态寄存器。为了恢复正常的功能，需要更新它们。

图 2 显示了用于自动存储操作的存储电容（ V_{CAP} ）连接情况。请参考第 55 页上的直流规范，以了解 V_{CAP} 的大小。

图 2. 自动存储模式



软件存储操作

通过软件存储可以根据指令执行一次存储操作。无论是否已经执行了写操作，只要执行 STORE 指令便能够启动存储操作。

存储周期需要 t_{STORE} 时间才能完成，在此期间，对 nvSRAM 的所有存储器访问都被禁止。可以轮询状态寄存器中的 WIP 位或 HSB 引脚，以查找就绪状态或忙碌状态。完成 t_{STORE} 周期时间后，nvSRAM 将处于就绪状态以进行正常操作。

硬件存储和 HSB 引脚操作

器件中的 HSB 引脚是一个双目的引脚，用于启动存储操作或轮询 STORE/RECALL 的完成状态。如果并未进行存储或回读操作，则可以将 HSB 引脚驱动为低电平，从而开始执行硬件存储周期。

如果在 HSB 引脚上检测到低电平，那么 nvSRAM 将经过 t_{DELAY} 时间后启动存储操作。只有在上一个存储或回读周期后已经执行了 SRAM 写操作时，才会开始硬件存储周期。这样能够优化 SONOS FLASH 擦写次数。在 t_{STORE} 时间内禁止对存储器执行任何读写操作。HSB 引脚还起到开漏驱动器（内部 100 kΩ 弱上

拉电阻）的作用，它在进行存储 / 回读时内部进入低电平以指示状态繁忙。

注意：每次硬件和软件存储操作完成后，HSB 通过标准输出高电流在短时间（ t_{HHHD} ）内变为高电平，然后通过内部 100 kΩ 上拉电阻一直保持高电平。

注意：若想成功进行最后一个数据字节的存储操作，则在收到最后一个数据位 D0 后至少一个时钟周期才能启动硬件存储。

注意：建议仅在器件进入待机状态时执行硬件存储操作。同时，应该退出“芯片内执行”（XIP）模式。

存储操作完成后，如果 HSB 引脚返回为高电平，nvSRAM 存储器访问将在 t_{LZHSB} 时间内被禁止。如果不使用 HSB 引脚，则必须将它置于未连接状态。

回读操作

执行回读操作时，可将非易失性元件中所存储的数据传输到 SRAM 单元中。可通过下面某种方法启动回读操作：在加电时启动的硬件回读和通过 SPI 回读指令启动的软件回读。

在内部，回读需要执行两步流程。首先，清除 SRAM 中的数据（设置为“0”）。然后，将非易失性信息传输到 SRAM 单元中。在进行回读周期时，所有存储器访问都将被禁止。回读操作不会更改非易失性单元中的数据。

硬件回读（加电）

加电时，如果 V_{CC} 的电压超过 V_{SWITCH} ，则会启动自动回读序列操作，该操作可以将非易失性存储器中的内容传输到 SRAM 中。

需要 t_{FA} 的时间才能完成加电回读周期，在此期间，存储器访问将被禁用。HSB 引脚用于检测器件的就绪状态。

软件回读

用户可通过软件回读来启动回读操作，从而将非易失性存储器中的内容重新存储到 SRAM 中。使用 RECALL（回读）指令触发软件回读。

需要经过 t_{RECALL} 时间才能完成软件回读操作，在此期间，对 nvSRAM 的所有存储器访问都被禁止。

禁用和使能自动存储

如果应用不需要自动存储功能，可通过 ASDI 指令禁用该功能。禁用后，nvSRAM 不会在断电时执行存储操作。

使用 ASEN 指令可以重新使能自动存储功能。然而，ASEN 和 ASDI 操作需要一个存储操作来使它们成为非易失性数据。

注意：出厂时已经使能了器件的自动存储功能，并且已将 0x00 写入到所有单元中。

注意：如果禁用了自动存储功能，并且不需要 V_{CAP} ，那么 V_{CAP} 引脚必须保持为开路状态。 V_{CAP} 引脚绝对不能接地。不可禁用加电回读操作。

四线串行外设接口

SPI 概述

SPI 是带有芯片选择 (\overline{CS})、串行输入 (SI)、串行输出 (SO) 和串行时钟 (SCK) 引脚的四引脚接口。该器件通过 SPI 接口提供对 nvSRAM 进行串行访问。器件上的 SPI 总线的最大运行频率为 108 MHz。

SPI 是同步的串行接口，它使用时钟和数据引脚进行存储器访问并支持数据总线上的多个器件。使用 \overline{CS} 引脚可激活 SPI 总线上的器件。

芯片选择、时钟和数据之间的关系是由 SPI 模式决定的。该器件支持 SPI 的模式 0 和模式 3。在这两种模式下，数据都将在 SCK 上升沿（从 \overline{CS} 变为有效之后的第一个上升沿）记录到 nvSRAM 内。

SPI 协议由操作码控制。这些操作码规定了从总线主设备到从设备的所有命令。激活 \overline{CS} 后，总线主设备传输的第一个字节便是操作码。随后，可以传输任何地址和数据。在完成某个操作并发出新的操作码前， \overline{CS} 必须进入无效状态。下面列出了 SPI 协议中常用的术语：

SPI 主设备

SPI 主设备控制着 SPI 总线上进行的操作。SPI 总线上仅有一个主设备，但可有一个或多个从设备。所有从设备共享了同一个 SPI 总线。主设备可通过 \overline{CS} 引脚选择任意一个从设备。要启动操作，必须由主设备将从设备的 \overline{CS} 引脚置于低电平状态来激活从设备。主设备生成 SCK（串行时钟），以便 SI 和 SO 线上的所有数据传输均与该时钟同步。

SPI 从设备

SPI 从设备由主设备通过芯片选择线激活。来自 SPI 主设备的串行时钟 SCK 作为从设备的输入，所有通信均与该时钟同步。SPI 从设备从来不会在 SPI 总线上执行通信，并且它会执行主设备发出的指令。

该器件可作为 SPI 从设备工作，并与其他 SPI 从设备共享 SPI 总线。

芯片选择 (\overline{CS})

要选择任意一个从设备，主设备必须下拉相应 \overline{CS} 引脚。只有 \overline{CS} 引脚为低电平时，才能将指令发送给从设备。未选择器件时，将忽略通过 SI 引脚的数据，同时，串行输出引脚 (SO) 保持高阻抗状态。

注意：必须从 \overline{CS} 的下降沿上开始执行新指令。因此，每个有效芯片选择周期内只能发送一个操作码。

注意：建议将连接 V_{CCQ} 的 10 k Ω 的外部上拉电阻安装在 \overline{CS} 引脚上。

串行时钟 (SCK)

串行时钟由 SPI 主设备生成，在 \overline{CS} 变为低电平后，通信将与该时钟同步。

器件使能 SPI 模式 0 和 3，以便进行数据通信。在这两种模式下，从设备在 SCK 的上升沿上锁存输入，输出在下降沿上发出。因此，SCK 的第一个上升沿表示 SI 引脚上 SPI 指令已接收到第一位 (MSB)。此外，所有数据输入和输出均与 SCK 同步。

数据传输 — SI/SO

SPI 数据总线包括 SI 和 SO 两条线，它们都可用于执行串行数据通信。SI 又称为主出从入 (MOSI)，SO 又称为主入从出 (MISO)。主设备通过 SI 引脚将指令发送到从设备，同时从设备通过 SO 引脚进行响应。如上所述，多个从设备可共享 SI 和 SO 线。

器件为 SI 和 SO 提供了两个可连接至主设备的独立引脚，如第 9 页上的图 3 所示。

该 SI 输入信号用于将数据串行传输给器件。它接收需要编程的操作码、地址和数据。各个数值在串行 SCK 时钟信号的上升沿上被锁存。在执行扩展 SPI 和 DPI/QPI 命令过程中，SI 变成 I/O0（一个输入 / 输出），用于接收操作码、地址和数据来进行写入（锁存在串行 SCK 时钟信号的上升沿上的值），并移出数据（在 SCK 的下降沿上）。

SO 输出信号用于串行传输器件中的数据。在串行 SCK 时钟信号的下降沿上移出数据。在执行扩展 SPI 和 DPI/QPI 命令过程中，SO 变成 I/O1（一个输入 / 输出），用于接收操作码、地址和数据来进行编程（锁存在串行 SCK 时钟信号的上升沿上的值），并移出数据（在 SCK 的下降沿上）。SO 具有一个用于实现 Repeater/Bus-Hold 功能的电路。

写保护 (\overline{WP})

在 SPI 和 DSPI 模式下，如果 \overline{WP} 引脚被驱动为低电平，可防止对状态寄存器进行写入操作，并保护存储器区域中的所有数据字节，该存储器区域受状态寄存器中块保护位的保护。

如果 \overline{WP} 被驱动为低电平，那么在执行 WRSR 命令期间以及状态寄存器的 SRWD 位被设置为“1”时，便无法对状态寄存器和配置寄存器进行写操作。这样可以防止对块保护 (BP2、BP1、BP0) 以及 TBPROT 位进行任何更改。因此，如果 WP 在执行 WRSR 命令过程中处于低电平，那么防止对存储在存储器区域（受块保护和 TBPROT 位的保护）中的所有数据字节进行修改。

在四线传输模式下， \overline{WP} 功能不可用。在这些模式下，I/O2（作为输入 / 输出）会替代 WP 功能，以便接收操作码、地址和数据（用于进行写入 / 编程）以及移出数据。WP 具有一个内部上拉电阻；如果在四线传输模式下不使用该引脚，则在主机系统中它可以保持为未连接状态。在 SPI 模式下，WP 有一个 100 k Ω 的内部弱上拉电阻。

NC (I/O3)

在四线传输模式下，NC (I/O3) 引脚作为 I/O3 使用（作为输入/输出），可以接收操作码、地址和数据（用于进行写入/编程）以及移出数据。NC (I/O3) 具有一个内部上拉电阻；如果不使用该引脚，则在主机系统中它可以保持为非连接状态。在 SPI 模式下，NC (I/O3) 有一个 100 kΩ 的内部弱上拉电阻。

最高有效位 (MSB)

SPI 协议要求发送的第一位是最高有效位 (MSB)。该条件也适用于地址和数据传输。

1 Mbit 串行 nvSRAM 需要一个 3 字节的地址，以进行读/写操作。由于地址只有 17 位，所以器件会忽略所载入的前七位。虽然前 7 位无需关注，但赛普拉斯建议将这些位全部设置为 0，以使能无缝切换实现更大的存储容量。

串行操作码

通过将 \overline{CS} 置于低电平选择从器件后，可将收到的第一个字节做为操作码进行期望的操作。该器件使用了标准操作码进行存储器访问。除了存储器访问外，它还提供了额外的操作码，用于实现 nvSRAM 特定的功能：STORE（存储）、RECALL（回读）、AutoStore Enable（自动存储使能）以及 AutoStore Disable（自动存储禁用）。更多信息，请查阅第 12 页上的表 2。

无效的操作码

如果收到无效的操作码，则该操作码将被忽略。器件将忽略 SI 引脚上存在的所有额外串行数据，直到 \overline{CS} 的下一个下降沿到来为止，与此同时，SO 引脚将保持三态。

指令

操作码、地址以及模式/虚拟周期的组合，用于发出命令。

模式位

地址位后面的控制位。器件使用控制位来使能“芯片内执行”(XIP) 模式。这些位被指定时，系统控制器将驱动它们。

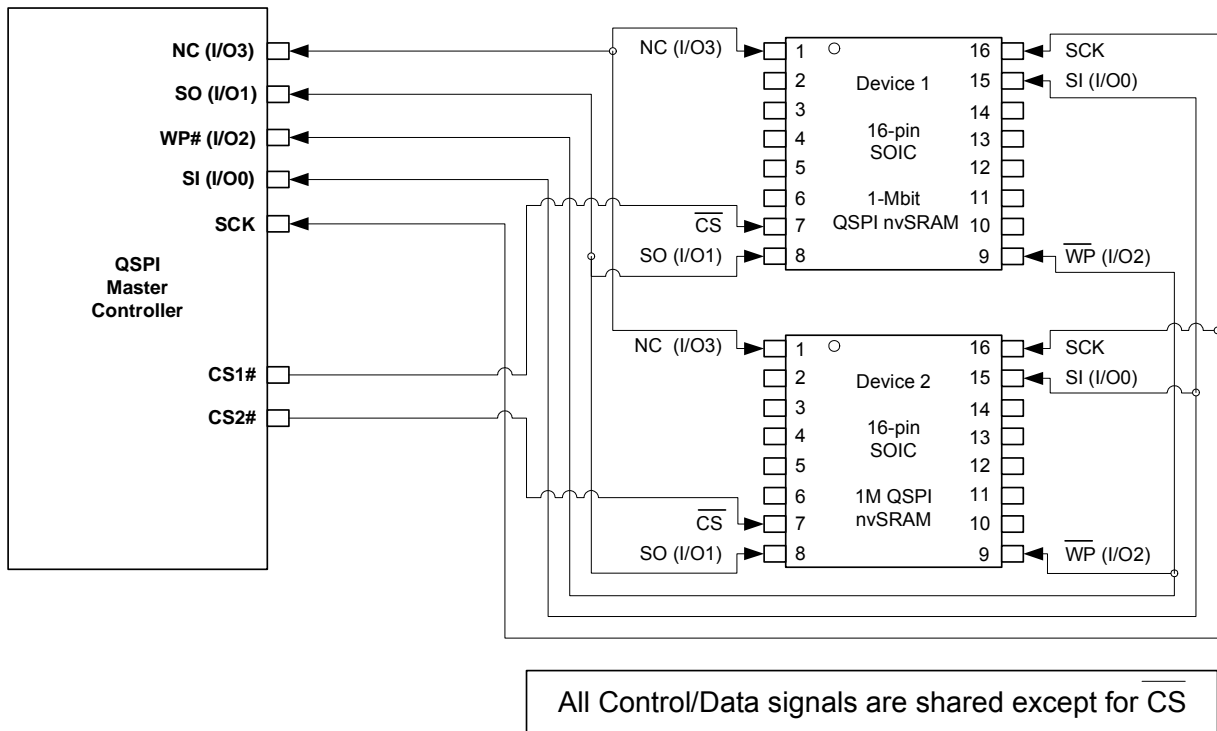
等待状态

在传输地址位或可选模式位后需要的虚拟时钟周期。

状态寄存器

该器件具有一个 8 位的状态寄存器。状态寄存器中的位用于配置 SPI 总线。第 14 页上的表 3 和表 4 对这些位进行了说明。

图 3. 采用多个 1 Mbit 四线 SPI nvSRAM 进行的系统配置



双线和四线 I/O 模式

CY14V101PS 还可以重新配置标准的 SPI 引脚，以便能够在双线和四线 I/O 模式下工作。

当器件处于双线 I/O 模式时，SI 引脚和 SO 引脚将变成 I/O0 引脚和 I/O1 引脚，以便传输操作码、地址和数据（双线 I/O 模式）或传输地址和数据（双线地址 / 数据模式）或只传输数据（双线数据模式）。

当器件处于四线 I/O 模式时，SI 引脚、SO 引脚、 \overline{WP} 引脚和 NC (I/O3) 引脚分别作为 I/O0 引脚、I/O1 引脚、I/O2 引脚和 I/O3 引脚，以便传输操作码、地址和数据（四线 I/O 模式）或传输地址和数据（四线地址 / 数据模式），或只传输数据（四线数据模式）。

表 1. I/O 模式

协议	命令输入	地址输入	数据输入 / 输出
SPI	SI	SI	SI/SO
DPI	I/O[1:0]	I/O[1:0]	I/O[1:0]
QPI	I/O[3:0]	I/O[3:0]	I/O[3:0]
双线数据模式 (双线输出)	I/O[0]	I/O[0]	I/O[1:0]
双线地址 / 数据 模式 (双线 I/O)	I/O[0]	I/O[1:0]	I/O[1:0]
四线数据模式 (四线输出)	I/O[0]	I/O[0]	I/O[3:0]
四线地址 / 数据 模式 (四线输入 / 输出)	I/O[0]	I/O[3:0]	I/O[3:0]

更多详细信息，请参考数据手册中读和写时序框图。

SPI 模式

该器件还有重新配置功能。可通过微控制器来驱动它，该微控制器的 SPI 外设可运行于下述任一模式：

■ SPI 模式 0 (CPOL = 0, CPHA = 0)

■ SPI 模式 3 (CPOL = 1, CPHA = 1)

在这两种模式下，都是在 SCK 的上升沿上锁存输入数据（SCK 上升沿是从 \overline{CS} 有效后的第一个上升沿开始的）。如果时钟由 HIGH 状态开始（处于模式 3 时），则时钟切换后的第一个上升沿被捕获。输出数据在 SCK 的下降沿上有效。

图 4 和图 5 分别显示了两种 SPI 模式。当总线主设备处于待机模式且不传输数据时，时钟的状态为：

■ 在模式 0 下，SCK 保持为“0”

■ 在模式 3 下，SCK 保持为“1”

当器件通过将 CS 引脚设置为低电平状态而被选中时，它将通过 SCK 引脚的状态检测出 SPI 模式。器件被选中时，如果 SCK 引脚处于低电平状态，它将工作于 SPI 模式 0；如果 SCK 引脚处于高电平状态，它将工作于 SPI 模式 3。

图 4. SPI 模式 0

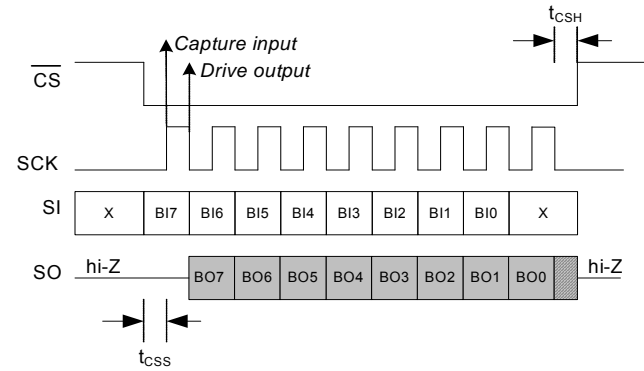
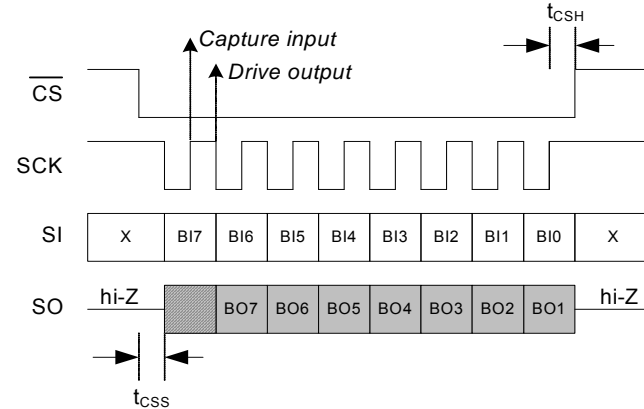


图 5. SPI 模式 3



SPI 操作功能

加电

加电是指启动电源和 V_{CC} 超过 V_{SWITCH} 电压时的状态。

如上面所述，加电时，nvSRAM 在 t_{FA} 的时间内进行加电回读操作。在此期间，所有存储器访问都被禁用。加电后，可检测 HSB 引脚以确定 nvSRAM 处于就绪 / 繁忙状态。

加电后的器件状态如下所述：

- SPI I/O 模式
- \overline{HSB} 的上拉电阻被激活
- SO 为三态
- 待机功能模式（如果 \overline{CS} 引脚处于高电平状态）。活动模式（如果 \overline{CS} 引脚处于低电平状态）。
- 状态寄存器的状态：
 - 写入使能（WEN）位被复位为 0。
 - 自前一个存储操作后，SRWD 保持不变。
 - 自前一个存储操作后，SNL 保持不变。
 - 自前一个存储操作后，块保护位保持不变。

- \overline{WP} 和 NC (I/O3) 功能由四线数据宽度 (QUAD) CR[1] 位定义。如果四线数据宽度 (QUAD) CR[1] 为逻辑 '0'，那么 \overline{WP} 和 NC (I/O3) 的电阻上拉有效。

断电

断电时 (V_{CC} 连续减小)，如果 V_{CC} 从正常工作电压下降到低于 V_{SWITCH} 电压阈值，则器件对发送到它的任何指令停止响应。

断电时，如果当前正在进行写周期，且已经接收到最后一个数据位 D0，则可以在 t_{DELAY} 的时间内完成写操作。然后，将禁止所有存储器访问并进行自动存储操作（如果在上一次回读周期后未执行写操作，那么便不会进行自动存储操作）。该功能用于防止断电时意外写入到 nvSRAM 内。

但是，为彻底避免在断电期间发生无意写入，需要确保器件未被选中且处于待机模式，并且确保 \overline{CS} 采用 V_{CC} 上的电压。

有效功耗模式和待机模式

当 \overline{CS} 处于低电平状态时，器件被选中且处于有效功耗模式。器件会消耗第 55 页上的直流规范中所指定的 I_{CC} ($I_{CC1} + I_{CCQ1}$) 电流。 \overline{CS} 处于高电平状态时，器件未被选中；如果未进行存储或回读周期，那么器件将进入待机状态。如果正在进行存储 / 回读周期，则存储或回读周期完成后器件将进入待机状态。

SPI 功能说明

该器件采用一个 8 位指令寄存器。表 2 列出了指令及其操作码。所有指令、地址和数据都在 \overline{CS} 从高电平转为低电平时开始传输。SPI 指令和 \overline{WP} 、NC (I/O3) 以及 \overline{HSB} 引脚会提供对 nvSRAM 中所有功能的访问操作。

表 2. 指令集

指令类别	指令名	操作码	SPI	双线输出	四线输出	双线 I/O	四线 I/O	DPI	QPI	最大频率 (MHz)
控制										
写禁用	WRDI	04h	有	—	—	—	—	有	有	108
写使能	WREN	06h	有	—	—	—	—	有	有	108
使能 DPI	DPIEN	37h	有	—	—	—	—	—	有	108
使能 QPI	QPIEN	38h	有	—	—	—	—	有	—	108
使能 SPI	SPIEN	FFh	—	—	—	—	—	有	有	108
存储器读取操作										
读取	READ	03h	有	—	—	—	—	有	有	40
快速读取	FAST_READ	0Bh	有	—	—	—	—	有	有	108
双线输出 (快速) 读取	DOR	3Bh	—	有	—	—	—	—	—	108
四线输出 (快速) 读取	QOR	6Bh	—	—	有	—	—	—	—	108
双线 I/O (快速) 读取	DIOR	BBh	—	—	—	有	—	—	—	108
四线 I/O (快速) 读取	QIOR	EBh	—	—	—	—	有	—	—	108
存储器写操作										
写入	WRITE	02h	有	—	—	—	—	有	有	108
双线输入写操作	DIW	A2h	—	有	—	—	—	—	—	108
四线输入写操作	QIW	32h	—	—	有	—	—	—	—	108
双线 I/O 写入	DIOW	A1h	—	—	—	有	—	—	—	108
四线 I/O 写入	QIOW	D2h	—	—	—	—	有	—	—	108
SR 命令										
软件复位使能	RSTEN	66h	有	—	—	—	—	有	有	108
软件复位	RESET	99H	有	—	—	—	—	有	有	108
读取 RTC	RDRTC	56h	有	—	—	—	—	有	有	40
写入 RTC	WRRTC	55h	有	—	—	—	—	有	有	108
快速读取 RTC	FAST_RDRT C	57H	有	—	—	—	—	有	有	108
进入休眠模式	HIBEN	BAh	有	—	—	—	—	有	有	108
进入睡眠模式	SLEEP	B9h	有	—	—	—	—	有	有	108
退出睡眠模式	EXSLP	ABh	有	—	—	—	—	有	有	108
寄存器命令										
读取状态寄存器	RDSR	05h	有	—	—	—	—	有	有	108
写入状态寄存器	WRSR	01h	有	—	—	—	—	有	有	108
读取配置寄存器	RDCR	35h	有	—	—	—	—	有	有	108
写入配置寄存器	WRCCR	87h	有	—	—	—	—	有	—	108

表 2. 指令集 (续)

指令类别	指令名	操作码	SPI	双线输出	四线输出	双线 I/O	四线 I/O	DPI	QPI	最大频率 (MHz)
读取 ID 寄存器	RDID	9Fh	有	—	—	—	—	有	有	40
快速读取 ID 寄存器	FAST_RDID	9Eh	有	—	—	—	—	有	有	108
写入序列号寄存器	WRSN	C2h	有	—	—	—	—	有	有	108
读取序列号寄存器	RDSN	C3h	有	—	—	—	—	有	有	40
快速读取序列号寄存器	FAST_RDSN	C9h	有	—	—	—	—	有	有	108
非易失性特定命令										
存储	STORE	8Ch	有	—	—	—	—	有	有	108
RECALL	回读	8Dh	有	—	—	—	—	有	有	108
自动存储使能	ASEN	8Eh	有	—	—	—	—	有	有	108
自动存储禁用	ASDI	8Fh	有	—	—	—	—	有	有	108
模式位										
模式位 (设置、复位)	—	Axh、无 Axh	有	—	—	—	—	有	有	—

根据它们的功能，SPI 指令被分为以下几种类型：

■ 控制指令：

- 写保护：WREN、WRDI 指令
- I/O 模式：DPIEN、QPIEN、SPIEN

■ 存储器读取指令：

- 存储器访问：READ、FAST_READ、DOR、QOR、DIOR、QIOR

■ 存储器写指令：

- 存储器访问：WRITE、DIW、QIW、DIOW、QIOW

■ 系统资源指令：

- 软件复位：RSTEN、RESET
- 实时时钟：RDRTC、WRRTC、FAST_RDRTC
- 功耗模式：HIBEN、SLEEP、EXSLP

■ 寄存器指令：

- 配置寄存器：RDCR、WRCR
- 状态寄存器：RDSR、WRSR
- 标识寄存器：RDID、FAST_RDID
- 序列号寄存器：RDSN、WRSN、FAST_RDSN

■ nvSRAM 特殊指令：

- 存储：STORE
- 回读：RECALL
- 使能 / 禁用：ASEN、ASDI

注意：下面部分所介绍的指令波形并不涉及 \overline{WP} (I/O2)、NC (I/O3) 以及 SO 上中继器 / 总线保持电路的电阻上拉影响。

注意：C5h、1Eh、C8h、CEh、CBh、CCh、CDh 均为赛普拉斯的保留指令操作码，用于修改器件配置。如果错误地输入了其中某个操作码，那么需要进行软件复位 (66h、99h) 才能恢复器件的正确配置。否则器件将无法准确工作。

状态寄存器

该器件具有一个状态寄存器；表 3 中列出了该状态寄存器及其位说明。状态寄存器中的位格式显示了对该位只能进行读操作（R）还是也能进行写操作（既可以写 / 也可以读操作（W/R））。唯一的例外是序列号锁定位（SNL）。SNL 仍为 ‘0’ 时，可使用

WRSN 指令多次写入序列号。将该位设置为 ‘1’ 时，可防止对序列号进行任何修改。该位在出厂时被设置为 ‘0’，并且只能写入一次。该位被设置为 ‘1’ 后，便不能将其恢复到 ‘0’。

表 3. 状态寄存器格式和位定义

位	字段名称	功能	类型	R/W	默认状态	说明
7	SRWD	状态寄存器写入操作禁用	NV	R/W	0	1 = 通过忽略 WRSR 命令， \overline{WP} 处于低电平时锁定 SR 的状态 0 = 不受保护，即使 \overline{WP} 处于低电平。
6	SNL	序列号锁定	OTP	R/W	0	锁定序列号
5	TBPROT	配置块的起始地址	NV	R/W	0	1 = BP 从底部（低地址）开始 0 = BP 从顶部（高地址）开始
4	BP2	块保护	NV	R/W	0	防止对块中的所选范围进行写入、编程或擦除操作
3	BP1		NV	R/W	0	
2	BP0		NV	R/W	0	
1	WEL	写使能锁存	V	R	0	1 = 器件接收写入寄存器（WRSR）、写入、编程或擦除命令 0 = 器件忽略写入寄存器（WRSR）、写入、编程或擦除命令 该位不受 WRSR 的影响，但受 WREN 和 WRDI 命令的影响
0	WIP	正在进行	V	R	0	1 = 器件处于繁忙状态，正在执行某个写入寄存器（WRSR）、编程、擦除或其他操作 0 = 就绪器件（Ready Device）处于待机状态，并且可以接收命令

状态寄存器写操作禁用（SRWD）SR[7]

当该位被设置为 1，并且 \overline{WP} 输入被驱动为低电平时，该器件将处于硬件保护模式。在这种模式下，所有 SRWD 位（非 WEL）都成为只读位，并不会再接收写寄存器（WRSR）命令来进行操作。如果 \overline{WP} 为高电平，那么可以通过 WRSR 命令修改各个 SRWD 位。如果 SRWD 被置为 ‘0’， \overline{WP} 不起任何作用，并且 SRWD 位可以通过 WRSR 命令得到修改。

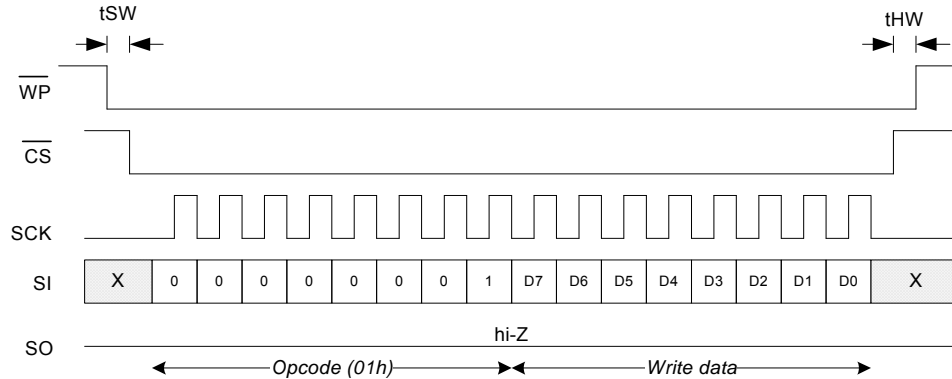
注意：设置好配置寄存器中的四线位 CR[1] 后， \overline{WP} 内部设置将为逻辑 ‘0’。SRWD 被置为逻辑 ‘1’，便不能改变保护状态，直到四线位 CR[1] 复位为逻辑 ‘0’ 为止。

表 4. SRWD、 \overline{WP} 、WEL 和保护

SRWD	\overline{WP}	WEL	受保护的块	不受保护的块	状态寄存器（WEL 除外）
X	X	0	受保护	受保护	受保护
0	X	1	受保护	可写	可写
1	低	1	受保护	可写	受保护
1	高电平	1	受保护	可写	可写

注意：执行写入状态寄存器指令时， \overline{WP} 会根据 \overline{CS} 进行采样，以确定硬件保护是否被使能。图 6 显示了时序波形。

图 6. 根据 \overline{CS} 的 \overline{WP} 时序



序列号锁定位 (SNL) SR[6]

将其设置为 ‘1’ 时，可防止对序列号进行所有修改。该位在出厂时被设置为 ‘0’，并且只能对其进行一次写操作。该位被设置为 ‘1’ 后，就不能将其恢复到 ‘0’。

顶部或底部保护 (TBPROT) CR[5]

该位定义了块保护位 (BP2、BP1 和 BP0) 的操作。必须在设备的系统制造过程中进行初始化配置时选择 TBPROT 所需的状态。

块保护 (BP2、BP1、BP0) SR[4:2]

这些位将存储器阵列域定义成了防止写入命令的软件保护。BP 位是非易失性的。当一个或多个 BP 位被设置为 ‘1’ 时，便不能对相关存储区进行写入、编程和擦除操作。

可通过各块保护位 (状态寄存器位 BP2、BP1 和 BP0) 与 TBPROT 位的组合来保护存储器阵列的地址范围。该范围大小由 BP 位的值确定，并且该范围的上限或下限起始值由状态寄存器的 TBPROT 位决定。

表 5. 保护部分的上层阵列起点 (TBPROT = 0)

状态寄存器内容			存储器阵列的受保护部分	地址范围
BP2	BP1	BP0		
0	0	0	无	无
0	0	1	上层的第 64 个扇区	0x1F800 - 0x1FFFF
0	1	0	上层的第 32 个扇区	0x1F000 - 0x1FFFF
0	1	1	上层的第 16 个扇区	0x1E000 - 0x1FFFF
1	0	0	上层的第 8 个扇区	0x1C000 - 0x1FFFF
1	0	1	上层的第 4 个扇区	0x18000 - 0x1FFFF
1	1	0	上半部分	0x10000 - 0x1FFFF
1	1	1	所有扇区	0x00000 - 0x1FFFF

表 6. 保护部分的下层阵列起点 (TBPROT = 1)

状态寄存器内容			存储器阵列的受保护部分	地址范围
BP2	BP1	BP0		
0	0	0	无	无
0	0	1	下层的第 64 个扇区	0x00000 - 0x007FFF
0	1	0	下层的第 32 个扇区	0x00000 - 0x00FFFF
0	1	1	下层的第 16 个扇区	0x00000 - 0x01FFFF
1	0	0	下层的第 8 个扇区	0x00000 - 0x03FFFF
1	0	1	下层的第 4 个扇区	0x00000 - 0x07FFFF
1	1	0	下半部	0x00000 - 0x0FFFFF
1	1	1	所有扇区	0x00000 - 0x1FFFFF

写使能 (WEL) SR[1]

必须将 WEL 位设置为 ‘1’，这样才能使能编程、写入、擦除操作，并能防止对存储器或寄存器值进行意外更改。执行写使能 (WREN) 命令会将写使能锁定位设置为 ‘1’，这样便能执行后面所有写命令。通过写禁用 (WRDI) 命令，将写使能锁定位设置为 0，这样可以防止所有写命令。在对寄存器的成功写操作、STORE、RECALL、编程或擦除操作结束时，将清除 WEL 位为 0 — 请注意，对存储器宏进行写操作后，它不会被清除。断电 / 上电序列、硬件复位或软件复位发生后，写使能锁定位将被设置为 ‘0’。WRSR 命令不会影响该位的状态。

注意：WEL 位不会对自动存储、上电回读和硬件存储（由 $\overline{\text{HSEB}}$ 启动）等操作产生任何影响。

表 7. 需要设置 WEL 位的指令

指令说明	指令名	操作码
存储器写入操作		
写入	WRITE	02h
双输入写入	DIW	A2h
四输入写入	QIW	32h
双线 I/O 写入	DIOW	A1h
四线 I/O 写入	QIOW	D2h
寄存器命令		
写状态寄存器	WRSR	01h
写入配置寄存器	WRCR	87h
写入序列号寄存器	WRSN	C2h
非易失性特定命令		
存储	STORE	8Ch
回读	RECALL	8Dh
自动存储使能	ASEN	8Eh
自动存储禁用	ASDI	8Fh

表 8. 配置寄存器

位	字段名称	功能	类型	R/W	默认状态	说明
7	RFU	预留	—	R/W	0	留给将来使用
6	RFU	预留	—	R/W	1	留给将来使用
5	RFU	预留	—	—	0	留给将来使用
4	RFU	预留	—	—	0	留给将来使用
3	RFU	预留	—	—	0	留给将来使用
2	RFU	预留	—	—	0	留给将来使用
1	QUAD	使器件进入四线模式	NV	R/W	0	1 = 四线；0 = 双线或串行
0	RFU	预留	—	—	0	留给将来使用

正在进行的操作 (WIP) SR[0]

表示器件是否正在执行编程、写入、擦除操作或任何其他操作，在此期间会忽略一条新的操作命令。当该位被设置为 ‘1’ 时，器件会执行一个后台操作。WIP 位为 ‘1’ 时，只能接收读状态 (RDSR) 命令。将 WIP 位清除为 ‘0’ 时，不会进行任何操作。这是一个只读位。

只有执行一个存储操作后，才会将写到 SR 中的所有值保存到非易失性存储器中。如果自动存储被禁用，则必须进行软件存储操作来保存对状态寄存器所做的修改。

如果对 SRAM 进行写入操作，那么硬件存储操作会只将状态寄存器的值传输到非易失性存储器内。

配置寄存器

该器件具有一个配置寄存器：表 8 中列出了该状态寄存器和其位说明。配置寄存器中的位格式显示了只能对该位进行读操作 (R) 还是也能进行写操作（既可以写 / 也可以读操作 (W/R)）。配置寄存器控制某些接口功能。

四线数据宽度 (QUAD) CR[1]

该位被设置为 ‘1’ 时，它会将数据宽度改变为 4 位，即： \overline{WP} 成为 I/O2，而 NC (I/O3) 成为 I/O3。 \overline{WP} 输入不被监控，因此它正常运行，并且它被内部设置为激活状态。可以正常执行串行、双线输出和双线 I/O 读取命令，但在使用不同数据路径宽度切换各个命令时就不用为这些命令驱动 \overline{WP} 输入。当使用四线输出读取、四线 I/O 读取、四线输入写入、四线 I/O 写入，以及所有四线 SPI 命令时，必须将四线位设置为 1。四线位是非易失性的。

注意：需要向配置寄存器内写入 0x42，才能设置四线位。同样，向配置寄存器内写入 0x40，可实现重置四线位。数据的所有其他组合会改变器件的配置情况，并使其无法工作。

注意：设置配置寄存器中的四线位 CR[1] 后， \overline{WP} 内部默认设置为逻辑 ‘0’。

注意：写入配置寄存器的有效值将在存储操作完成后被保存到非易失性存储器内。如果自动存储被禁用，则必须要进行软件存储操作来保存对配置寄存器所做的修改。如果对 SRAM 进行写入操作，那么硬件存储操作只将配置寄存器的值传输到非易失性存储器内。

SPI 控制指令

写入禁用 (WRDI) 指令

该写禁用指令通过将 WEL 位清除为“0”来禁用所有写操作，从而防止发生意外写入。CS 的下降沿和 WRDI 指令的操作码完成后，将发出该指令。WEL 位会在 CS 的上升沿上被清除。

图 7. SPI 模式下的 WRDI 指令

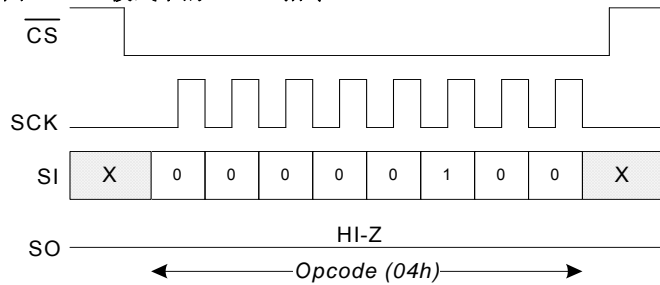


图 8. DPI 模式下的 WRDI 指令

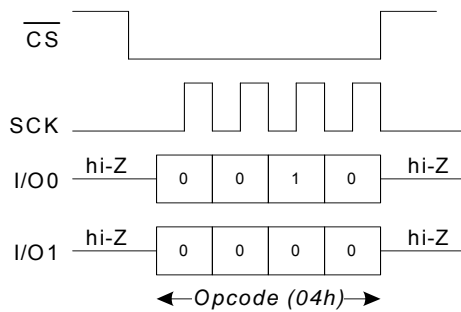
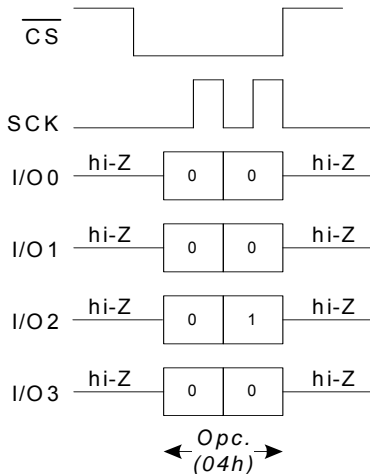


图 9. QPI 模式下的 WRDI 指令



写入使能 (WREN) 指令

加电时，器件保持写入禁用状态。因此，写指令或 nvSRAM 特殊指令的前面必须是一条写入使能指令。如果器件并非处于写入使能状态 (WEL = 0)，那么它会忽略写指令并在 CS 处于高电平状态时返回到待机状态。CS 下降沿过后会发出该指令，然后该指令会将状态寄存器的 WEL 位置为“1”。加电时，WEL 位默认为 0。

注意：成功写入寄存器内或结束 STORE、RECALL、ASEN 或 ASDI 操作时，WEL 位被清除为 0。对存储器宏进行写操作后，不会清除它。

图 10. SPI 模式下的 WREN 指令

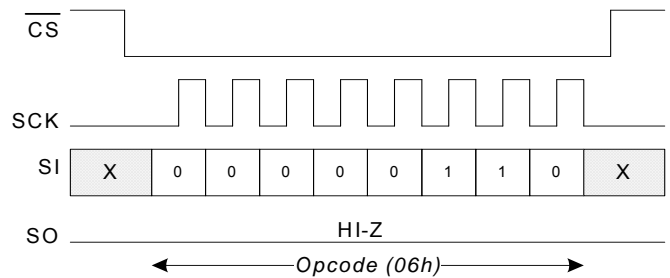


图 11. DPI 模式下的 WREN 指令

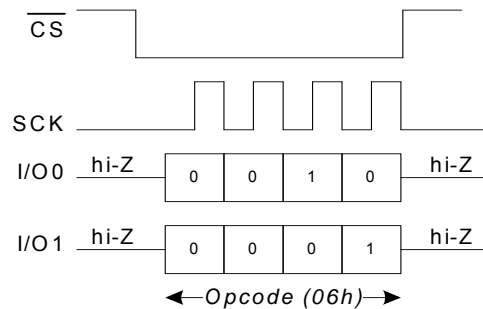
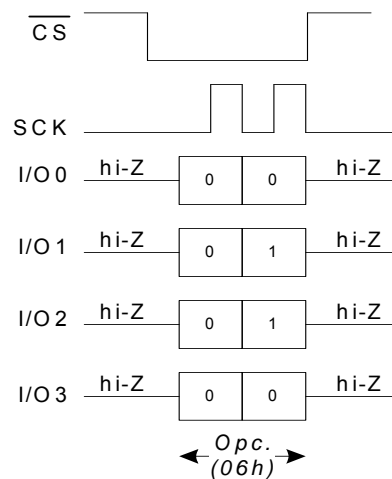


图 12. QPI 模式下的 WREN 指令



使能 DPI (DPIEN) 指令

DPIEN 使能双线 I/O 模式，在该模式下，通过 I/O0 和 I/O1 传送操作码、地址、各模式位和数据。

图 13. SPI 模式下的使能双线 I/O 指令

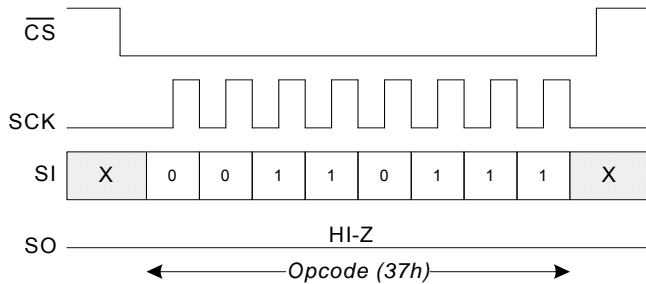
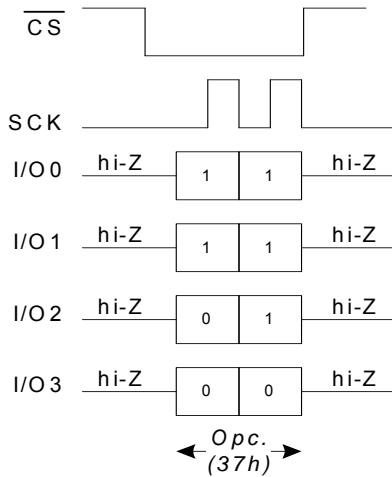


图 14. QPI 模式下的使能双线 I/O 指令



使能 QPI (QPIEN) 指令

QPIEN 使能 QPI 模式，在该模式下，操作码、地址、虚拟 I 模式位和数据会通过 IO0、IO1、IO2 和 IO3 被传送。QPIEN 指令不会设置配置寄存器中的四线位 CR[1]。因此，用来设置四线位 CR[1] 的 WRCR 指令必须执行 QPIEN 指令。

注意：禁用 QPI 模式，不会重置四线位 CR[1]。

图 15. SPI 模式下的使能四线 I/O 指令

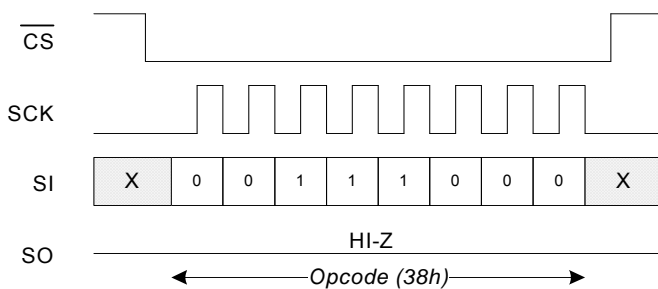
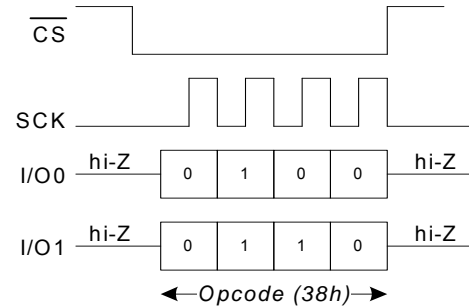


图 16. DPI 模式下的使能四线 I/O



使能 SPI (SPIEN) 指令

SPIEN 将禁用双线 I/O 或四线 I/O 模式，并将器件返回 SPI 模式。QPIEN 指令不会对配置寄存器中的四线位 CR[1] 进行复位。

图 17. DPI 模式下的使能 SPI 指令

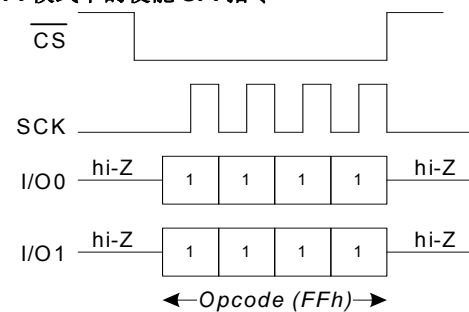
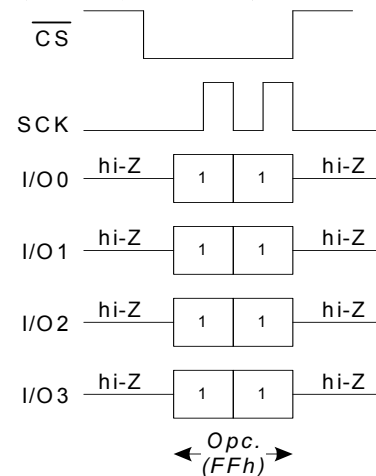


图 18. QPI 模式下的使能 SPI 指令



SPI 存储器读取指令

通过读指令可以访问存储器阵列。在进行存储或回调周期过程中无法使用这些指令。可通过状态寄存器的 WIP 位和 HSB 引脚来表示正在进行的存储操作。

读指令

如果在 SI 引脚上提供了读指令操作码，那么器件将进行读取操作。它将在 SPI 模式下读取 SO 引脚上的数据，在双线 I/O 模式下读取 I/O1 和 I/O0 引脚上的数据，并在四线 I/O 模式下读取 I/O3、I/O2、I/O1 和 I/O0 引脚上的数据。将 CS 引脚设置为低电平选中器件后，将发送读取操作码，然后输入三个字节地址。该器件还包含一个用于 1 Mbit 配置的 17 位地址空间。

最高有效地址字节在位 0 上包含的是 A16，其他则为 ‘无需关注’ 位。地址位 A15 到 A0 通过后面的两个地址字节发送。发送最后地址位后，会在 SCK 下降沿上移出指定地址上的数据 (D7–D0) (以 D7 开始)。如果将 CS 设置为低电平，则可以进行突发读操作。

输出每一个数据字节后，器件自动递增至下一个更高地址。当到达最后一个数据存储器地址 (0x1FFFF) 时，地址将翻转为 0x0000，然后器件继续进行读操作。输出数据时可以随时将 CS 置于高电平，以停止读指令。

注意：读指令的最高工作频率为 40 MHz。在双线和四线 I/O 模式下，地址字节传输后需要经过一个虚拟周期。这样便允许器件预提取第一个字节并开始执行流水线操作。

READ 指令

读指令可以在 SPI、双线 I/O (DPI) 或四线 I/O (QPI) 模式下使用。在 SPI 模式下，操作码和地址字节都通过 SI 引脚被传输，每个时钟周期传输一位。在最后一个地址周期的 SCK 的下降沿上，特定地址上的数据 (D7–D0) 将被移出，每个时钟周期一位 (以 D7 开始)。

在 DPI 模式下，操作码和地址字节通过 I/O1 和 I/O0 引脚传送，每个时钟周期两位。在最后一个地址周期的 SCK 下降沿上，位于特定地址的数据 (D7–D0) 被移出，每个时钟周期移出两位 (在 I/O1 上以 D7 开始，在 I/O0 上以 D6 开始)。在 QPI 模式下，操作码和地址字节通过 I/O3、I/O2、I/O1 和 I/O0 引脚被传送，并且每个时钟周期传送四位。在 SCK 最后一个地址周期的下降沿上，特定地址上的数据 (D7–D0) 将被移出，并且每个时钟周期移出四位 (在 I/O3 上以 D7 开始，在 I/O2 上以 D6 开始，在 I/O1 上以 D5 开始，在 I/O0 上以 D4 开始)。

图 19. SPI 模式下的读指令

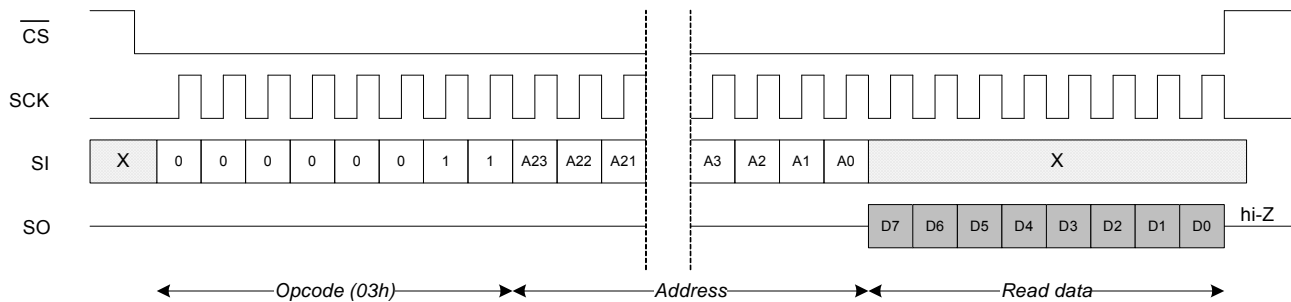


图 20. SPI 模式下的突发模式读指令

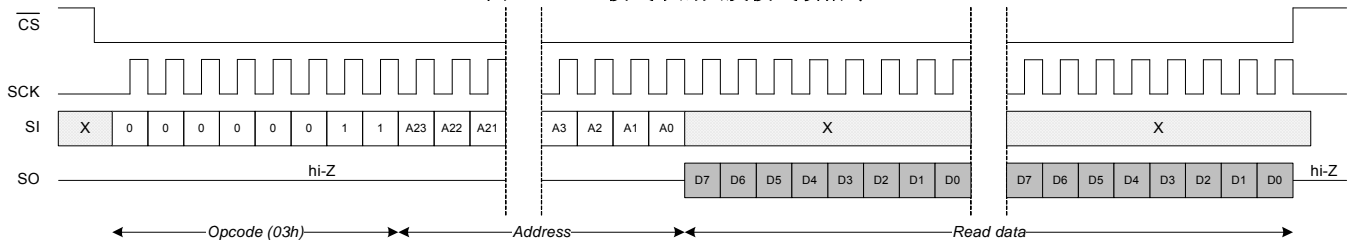


图 21. DPI 模式下的读指令

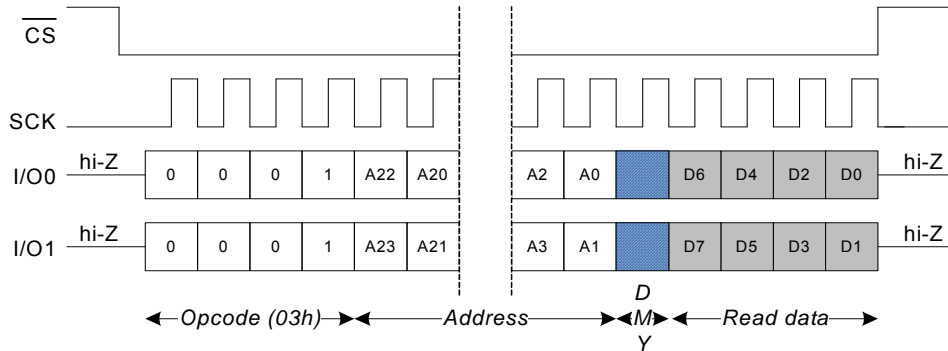
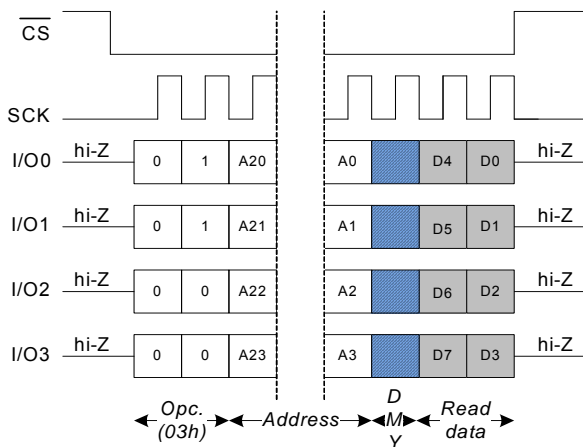


图 22. QPI 模式下的读指令



注意: 在 QPI 模式下执行读指令之前，必须将四线位 CR[1] 设置为逻辑 ‘1’。

快速读取指令

利用快速读取指令，能够以高达 108 MHz（最大值）的 SPI 频率读取存储器。与通用的读指令相似，该指令的所有 I/O 配置中都有一个等待状态；在发送地址后、发送第一个数据前，必须发送模式字节。这样便允许器件预提取第一个字节并开始执行流水线操作。主机系统首先要通过将 CS 设置为低电平来选中器件，然后输入三个地址字节和一个模式字节。在下一个 SCK 下降沿

上，特定地址的数据分别在 SPI 模式的 SO 引脚、双线 I/O 模式的 I/O1 和 I/O0 引脚以及四线 I/O 模式的 I/O3、I/O2、I/O1 和 I/O0 引脚上移出。指定的第一个字节可处于任何位置。输出每一个数据字节后，器件自动递增至下一个更高地址。因此，只要通过单一快速读取指令就可读取整个存储器阵列。当到达存储器阵列中的最高地址时，地址计数器将翻转为起始地址 0x00000，并允许读取序列无限期继续。在数据输出时可以随时将 CS 置于高电平以终止快速读取指令。

注意: 这些指令可以在最高 108 MHz SPI 工作频率下运行。

FAST_READ 指令

FAST_READ 指令可以在 SPI、双线 I/O (DPI) 或四线 I/O (QPI) 模式下使用。在 SPI 模式下，操作码、地址和模式字节都通过 SI 引脚被传输，每个时钟周期传送一位。在最后模式字节周期的 SCK 的下降沿上，特定地址上的数据 (D7-D0) 在 SO 引脚上移出，每个时钟周期一位 (以 D7 开始)。在 DPI 模式下，操作码、地址和模式字节通过 I/O1 和 I/O0 引脚传送，每个时钟周期传送两位。在最后模式周期的下降沿上，特定地址上的数据 (D7-D0) 将被移出，每个时钟周期两位 (D7 通过 I/O1 移出，D6 通过 I/O0 移出)。在 QPIO 模式下，操作码和地址字节通过 I/O3、I/O2、I/O1 和 I/O0 引脚传送，每个时钟周期传送四位。在最后模式周期的 SCK 的下降沿上，特定地址上的数据 (D7-D0) 将被移出，每个时钟周期四位 (D7 通过 I/O3 移出，D6 通过 I/O2 移出，D5 通过 I/O1 移出，并 D4 通过 I/O0 移出)。

图 23. SPI 模式下的 FAST_READ 指令

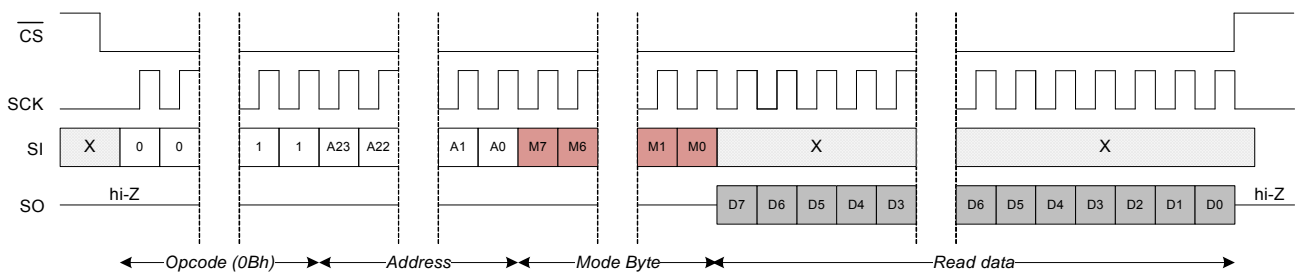


图 24. DPI 模式下的 FAST_READ 指令

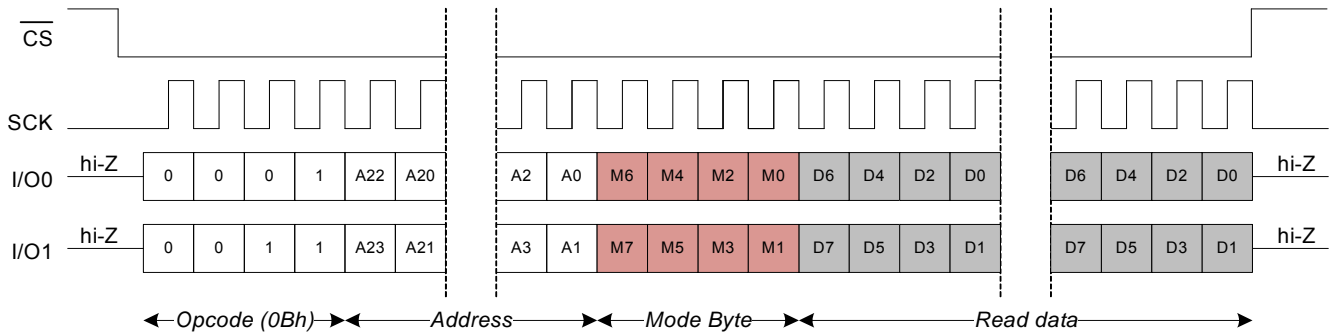
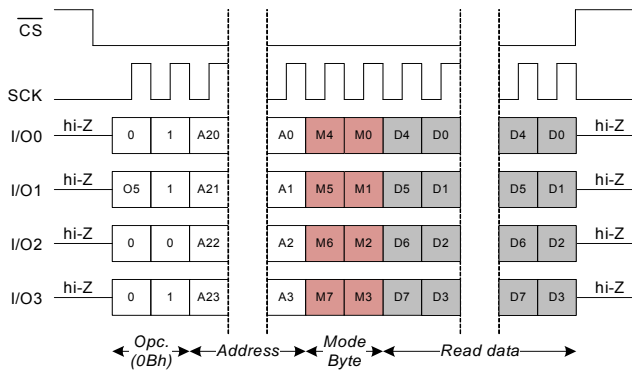


图 25. QPI 模式下的 FAST_READ 指令



DOR 指令

DOR 指令用于双数据模式，它属于 SPI 扩展读命令的一部分。在双线数据模式下，操作码、地址和模式字节通过 SI 引脚被传送，每个时钟周期传送一位。在最后一个模式周期的 SCK 的下降沿上，这些引脚被重新配置：SO 作为 I/O1，SI 作为 I/O0。特定地址上的数据（D7-D0）在 I/O1 和 I/O0 引脚上被移出，每个时钟周期两位（D7 通过 I/O1 移出，D6 通过 I/O0 移出）。

QOR 指令

QOR 指令在四数据模式下使用，它是 SPI 扩展读命令的部分。在四线数据模式下，操作码、地址和模式字节通过 SI 引脚被传送，每个时钟周期传送一位。在最后一个模式周期的 SCK 的下降沿上，这些引脚被重新配置：NC 变成 I/O3，WP 变成 I/O2，SO 变成 I/O1 和 SI 变成 I/O0。特定地址上的数据（D7-D0）在 I/O3、I/O2、I/O1 和 I/O0 引脚上被移出，每个时钟周期四位（D7 通过 I/O3 移出，D6 通过 I/O2 移出，D5 通过 I/O1 移出，并 D4 通过 I/O0 移出）。

注意 执行 QOR 指令前，必须将四线位 CR[1] 设置为逻辑“1”。

图 26. DOR 指令

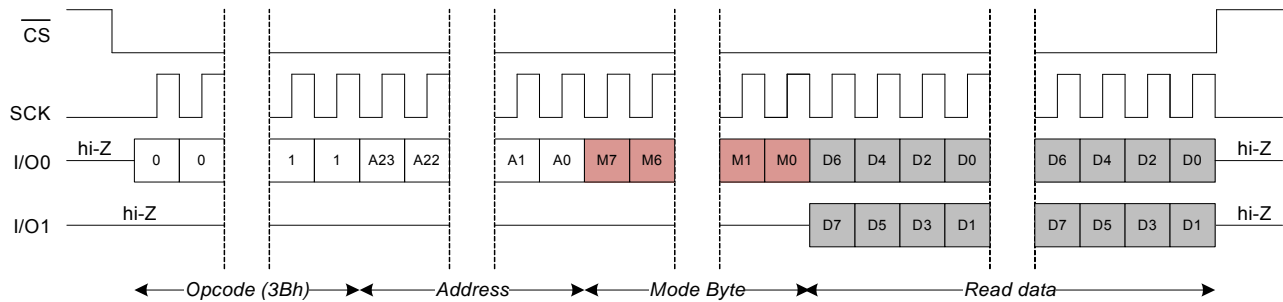
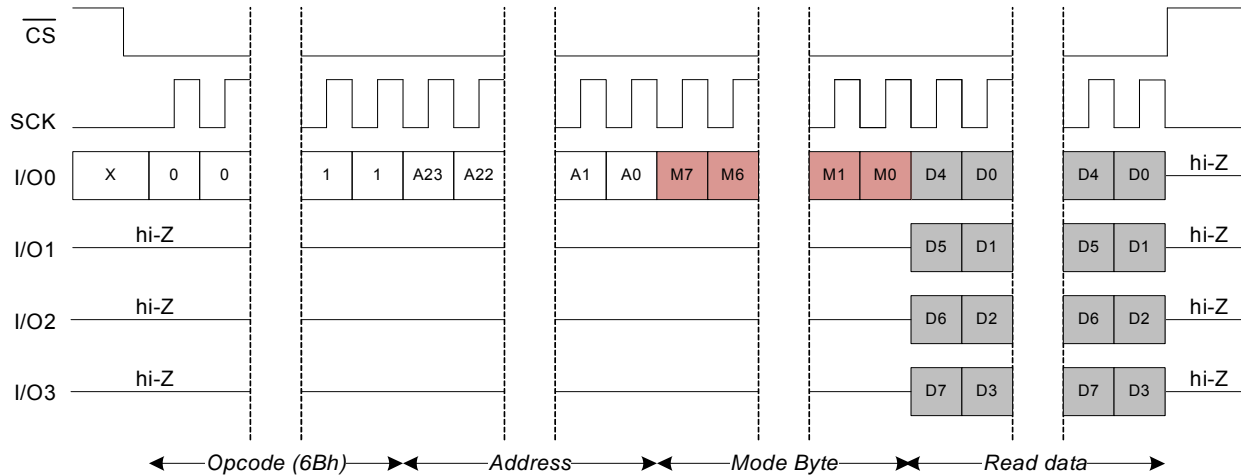
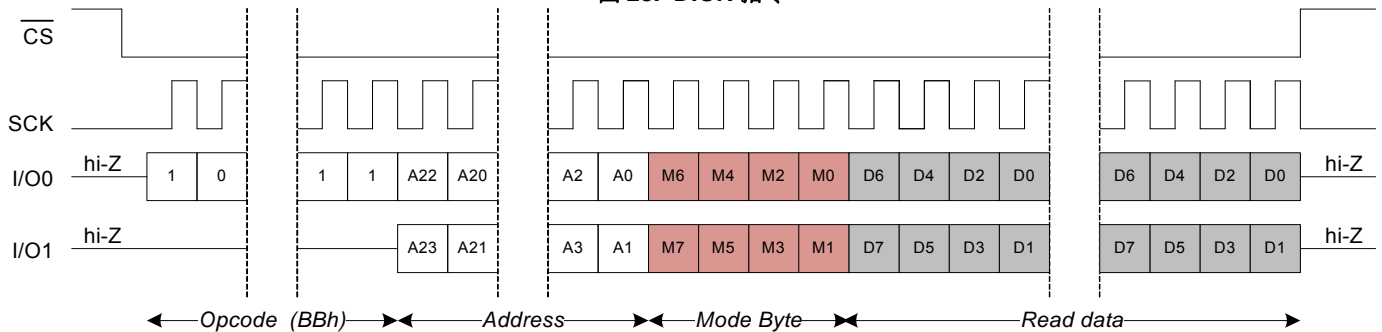


图 27. QOR 指令


DIOR 指令

DIOR 指令在双地址 / 数据模式下使用，它是 SPI 扩展读命令的一部分。在双线地址 / 数据模式下，操作码通过 SI 引脚被传送，每个时钟周期传送一位。在传输操作码的最后位后，这些引脚被重新配置：SO 变成 I/O1 和 SI 变成 I/O0。然后，该地址通过 I/O1 和 I/O0 引脚传送到器件，每个时钟周期传送两位（在 I/O1 上以

A23 开始，在 I/O0 上以 A22 开始），直到输入三字节地址为止。位于特定地址的数据（D7-D0）在 I/O1 和 I/O0 引脚上被移出，每个时钟周期移出两位（在 I/O1 上以 D7 开始，在 I/O0 上以 D6 开始）。

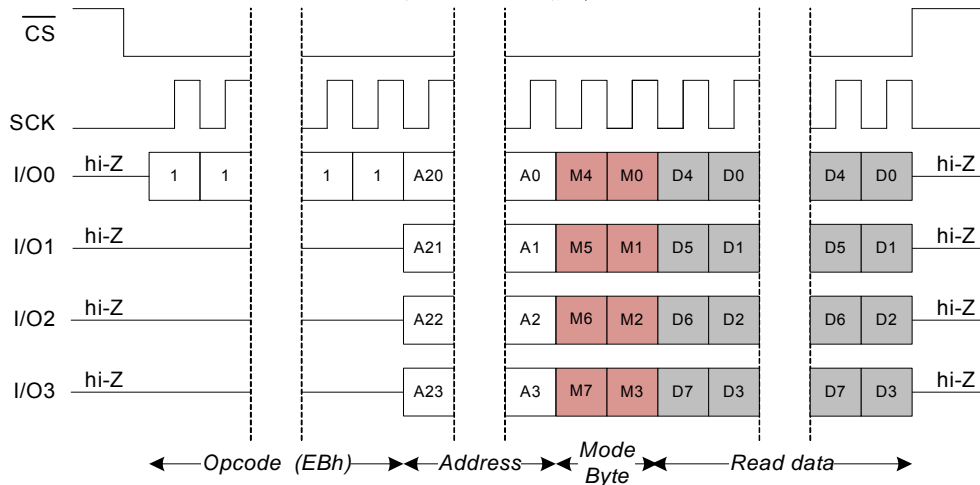
图 28. DIOR 指令


QIOR 指令

QIOR 指令在四地址 / 数据模式下被使用，它是 SPI 扩展读命令的一部分。在四线地址 / 数据模式下，操作码通过 SI 引脚被传送，并且每个时钟周期传送一位。在操作码的最后位传输后，这些引脚被重新配置：NC 作为 I/O3，WP 作为 I/O2，SO 作为 I/O1 和 SI 作为 I/O0。然后，该地址通过 I/O3、I/O2、I/O1 和 I/O0 引脚传送到器件，每个时钟周期四位（A23 通过 I/O3 传输，A22 通过 I/O2 传输，并 A20 通过 I/O0 传输），直到输入三字节地址为止。特定地址上的数据（D7-D0）在 I/O3、I/O2、I/O1 和 I/O0 引脚上被移出，每个时钟周期传送四位（D7、D6、D5 和 D4 分别通过 I/O3、I/O2、I/O1 和 I/O0 移出）。

注意：在执行 QIOR 指令前，四线位 CR[1] 必须是逻辑 '1'。

图 29. QIOR 指令



写指令

将写指令操作码和写数据提供给 SPI 模式的 SI 引脚、双线 I/O 模式的 I/O1 和 I/O0 引脚或四线 I/O 模式的 I/O3、I/O2、I/O1 和 I/O0 引脚时，器件将进行写操作。如果器件处于写入禁用状态，首先要通过 WREN 指令使能写入功能，这样才能进行写操作。使能写入功能（WEL = ‘1’）后，将在 \overline{CS} 的下降沿后发出 WRITE 指令。nvSRAM 允许进行突发写入操作，这样可以无需发出新的写入指令，仍能写入连续的地址。如果只写入一个字节，发送 D0（数据的 LSB）后必须将 \overline{CS} 引脚置为高电平。如果要写入多个字节，则必须使 \overline{CS} 引脚保持问低电平状态，地址会自动被递增。输入引脚上的数据字节被写入到连续地址内。当达到最后数据存储地址（0x1FFFF）时，地址将翻转为 0x00000，并且该器件将继续进行写操作。

注意：对存储器阵列进行写序列后，状态寄存器中的 WEL 位不会被复位到 ‘0’。

注意：当突发写入达到受保护的块地址时，它仍在受保护空间内保持地址递增，但是不将任何数据写入到受保护的存储器内。如果地址翻转并采用突发写入到不受保护的块，则可恢复写入。如果在写入保护的块中启动了突发写入，将执行同一操作。

注意：这些指令的最高工作频率可达 108 MHz。

将 \overline{CS} 引脚置于低电平并选择器件后，进行传输写操作码，然后再传输三字节地址。该器件还包含一个用于 1 Mbit 配置的 17 位地址空间。最高有效地址字节在位 0 中包含 A16，其他为 ‘无需关注’ 位。地址位 A15 至 A0 在随后的两个地址字节中发送。传送最后地址位后，将立即通过输入线传送数据（D7-D0）。可以在 SPI、DPI 或 QPI 模式下使用该命令。

WRITE 指令

可以在 SPI、DPI 或 QPI 模式下使用写指令。在 SPI 模式下，操作码、地址字节和数据字节通过 SI 引脚传送，每个时钟周期一位（以 D7 开始传输）。在 DPI 模式下，操作码、地址字节和数据字节通过 I/O1 和 I/O 引脚传送，每个时钟周期两位（D7 和 D6 分别通过 I/O1、I/O0 传送）。在 QPI 模式下，操作码、地址字节和数据字节通过 I/O3、I/O2、I/O1 和 I/O0 引脚传送，每个时钟周期四位（D7、D6、D5 和 D4 分别通过 I/O3、I/O2、I/O1 和 I/O0 传送）。

图 30. SPI 模式下的写指令

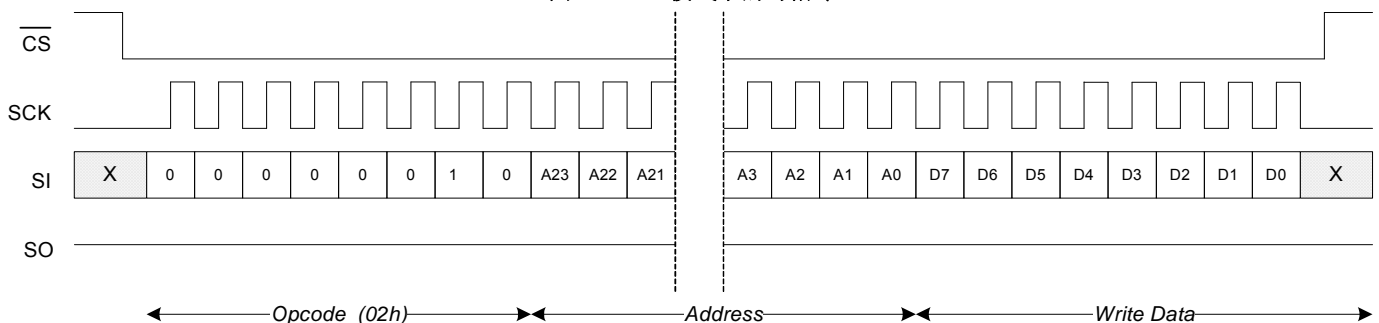


图 31. SPI 模式下的突发写指令

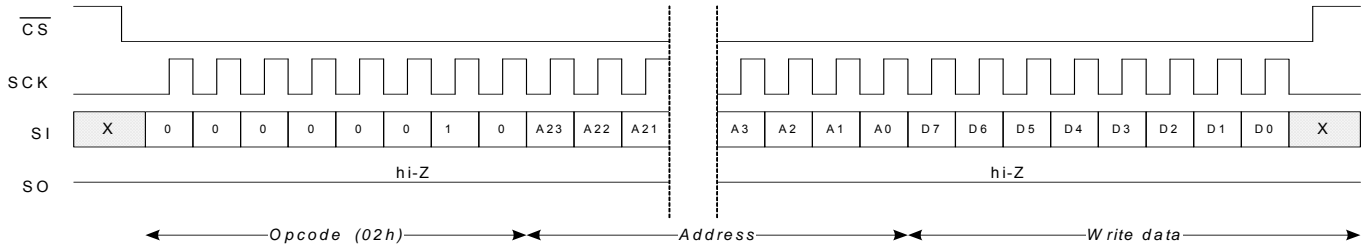


图 32. DPI 模式下的写指令

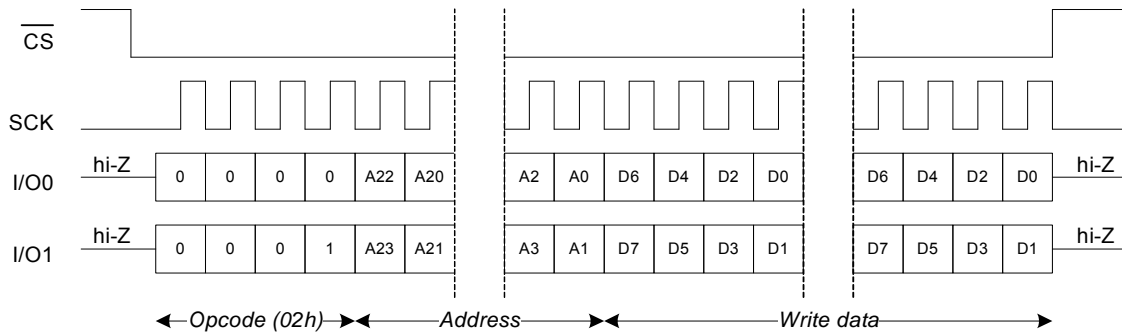
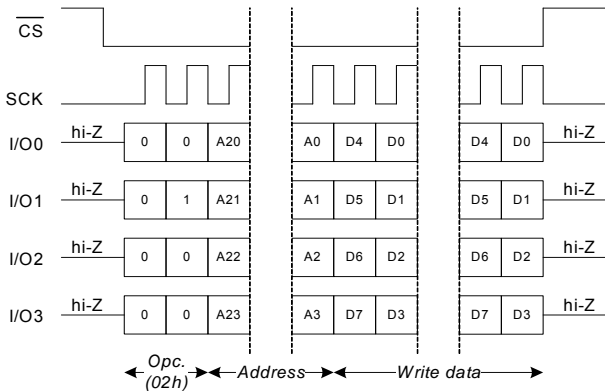


图 33. QPI 模式下的写指令

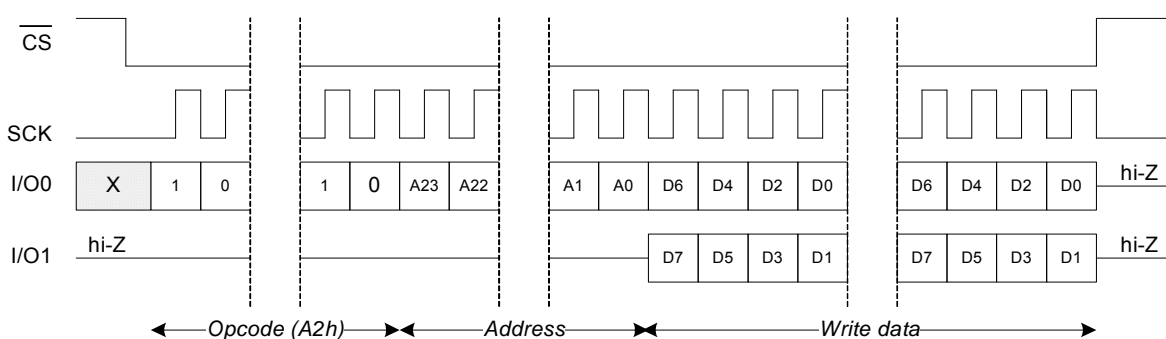


注意: 在 QPI 模式下执行写指令前, 必须将四线位 CR[1] 设置为逻辑 '1'。

DIW 指令

DIW 指令在双数据模式下使用, 它是 SPI 扩展写命令的一部分。在双数据模式下, 操作码和地址字节通过 SI 引脚传送, 每个时钟周期传送一位。传送最后地址位后, 将立即重新配置各引脚: SO 变成 I/O1 和 SI 变成 I/O0。另外, 数据 (D7-D0) 被传送到 I/O1 和 I/O0 引脚, 每个时钟周期两位 (D7 和 D6 分别通过 I/O1 和 I/O0 传送)。

图 34. DIW 指令



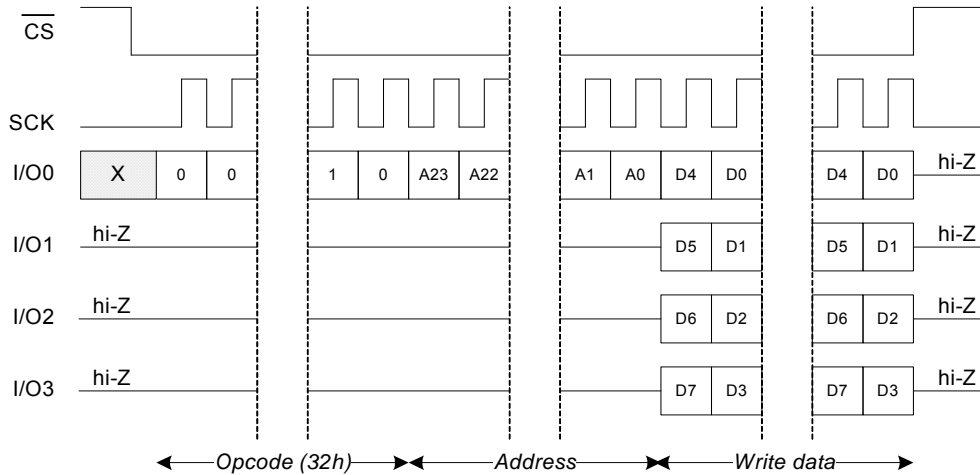
QIW 指令

QIW 指令在四线数据模式下被使用，它是 SPI 扩展写命令的一部分。在四线数据模式下，操作码和地址字节通过 SI 引脚被传送，每个时钟周期传送一位。传送最后地址位后，将立即重新配置各引脚：NC 变成 I/O3，WP 变成 I/O2，SO 变成 I/O1 和 SI 变成

I/O0。另外，数据（D7-D0）被传送到 I/O3、I/O2、I/O1 和 I/O0 引脚，每个时钟周期四位（D7、D6、D5 和 D4 通过 I/O3、I/O2、I/O1 和 I/O0 传送）。

注意：在执行 QIW 指令前，四线位 CR[1] 必须是逻辑 ‘1’。

图 35. QIW 指令

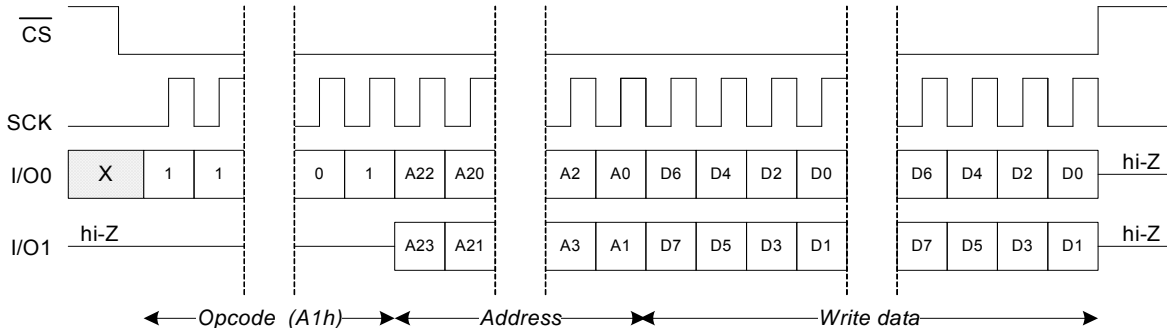


DIOW 指令

DIOW 指令在双线地址 / 数据模式下被使用，它是 SPI 扩展写命令的一部分。在双线地址 / 数据模式下，操作码通过 SI 引脚被传送，每个时钟周期传送一位。传送最后操作码位后，将立即重新配置各引脚：SO 变成 I/O1 和 SI 变成 I/O0。另外，地址通过 I/O1

和 I/O0 引脚传送到器件，每个时钟周期两位（首先 A23 通过 I/O1 传送，A22 通过 I/O0 传送），直到输入三字节地址为止。传送最后地址位后，会通过 I/O1 和 I/O0 将数据（D7-D0）传送到器件，每个时钟周期两位（首先 D7 通过 I/O1 传输，D6 通过 I/O0 传输）。

图 36. DIOW 指令

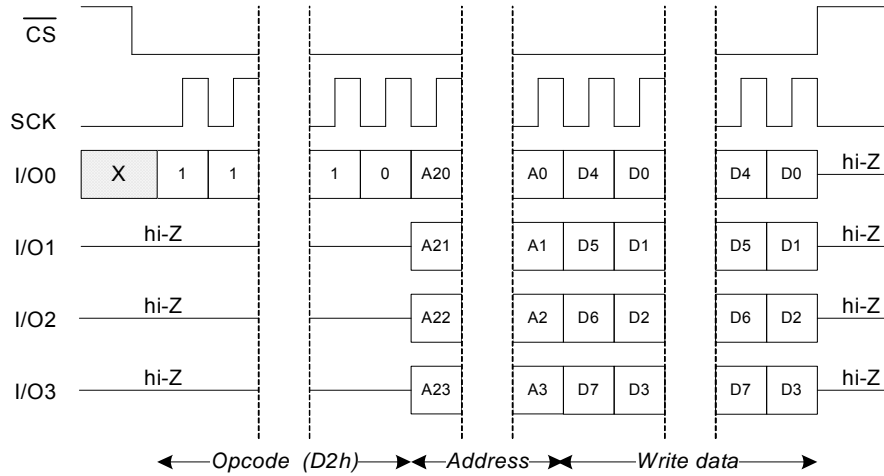


QIOW 指令

QIOW 指令在四线地址 / 数据模式下被使用，它是 SPI 扩展写命令的一部分。在四线地址 / 数据模式下，操作码通过 SI 引脚被传送，并且每个时钟周期传送一位。传送最后操作码位后，将立即重新配置各引脚：NC 变成 I/O3，WP 变成 I/O2，SO 变成 I/O1 和 SI 变成 I/O0。另外，地址通过 I/O3、I/O2、I/O1 和 I/O0 引脚传送到器件，每个时钟周期四位（开始于 A23 通过 I/O3 传送，A22 通过 I/O2 传送，A21 通过 I/O1 传输，并 A20 通过 I/O0 传送），直到输入三字节地址为止。传送最后地址位后，会通过 I/O3、I/O2、I/O1 和 I/O0 将数据（D7-D0）传送给器件，每个时钟周期传送四位（在 I/O3 上以 D7 开始，在 I/O2 上以 D6 开始，在 I/O1 以 D5 开始，在 I/O0 以 D4 开始）。

注意：在执行 QIOW 指令前，四线位 CR[1] 必须是逻辑 ‘1’。

图 37. QIOW 指令



Execute In Place (芯片内执行 — XIP)

Execute-in-place (XIP) 模式允许存储器进行开始于不同地址的一系列读操作，而不需要加载每个读操作的命令代码。这样会节省随机访问时间并且不需要将代码映像到 RAM 以快速运行。XIP 模式支持的读命令是 FAST_READ (在 SPI、DPI 和 QPI 模式下)、DOR、DIOR、QOR 和 QIOR。

通过输入 Mode 位可以设置或复位这些命令的 XIP 模式。Mode 位的高位半字节 (位 7-4) 通过添加或删除第一字节指令代码控制上述读命令的长度。Mode 位的低位半字节 (位 3-0) 无需关

注，它们可以为高阻抗。微控制器通常使用这些字节反转总线以读取数据。如果 Mode 位等于 Axx，器件将被设置为 / 保持读取模式，并且可以输入下个地址而不需要操作码，如下图所示；这样将清除操作码序列的某些周期。如果 Mode 位不等于 Axx，XIP 模式将被复位，而且在当前数据传输结束后，器件将等待操作码。

在执行这些命令期间，可以在任何序列中随时进入或退出 XIP。如果需要执行不受 XIP 支持的另一个操作 (如写操作)，那么在输入新的命令代码以实现所需操作前需要退出 XIP。

图 38. SPI 模式和 FAST_READ 指令的 XIP (0Bh)

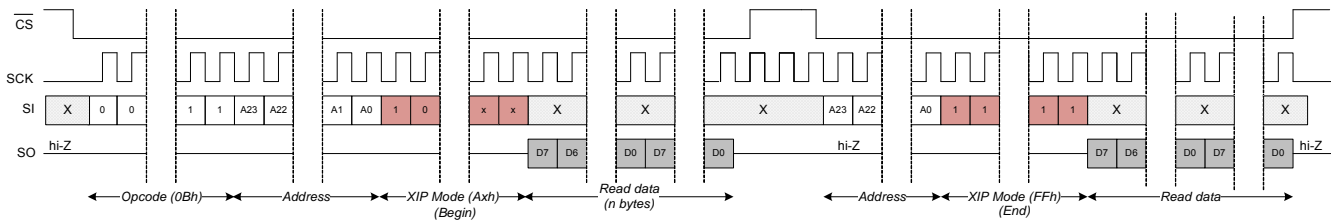
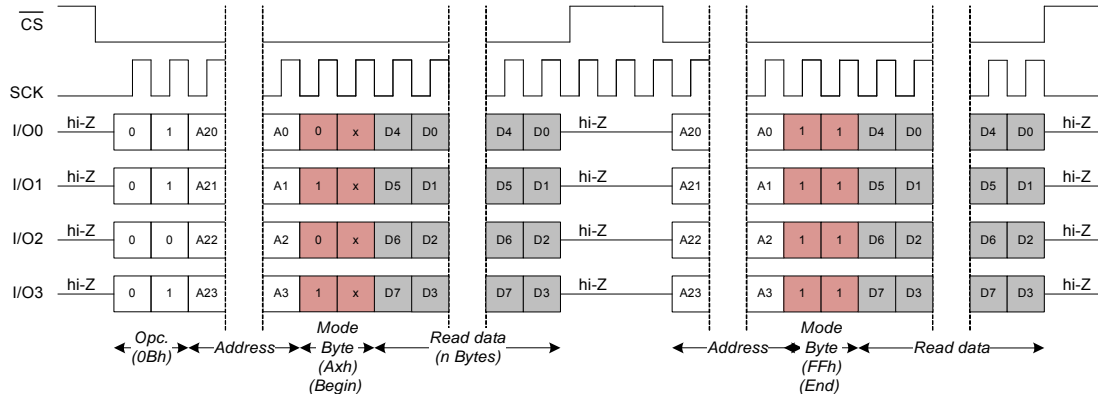


图 39. QPI 模式和 FAST_READ 指令的 XIP (0Bh)



系统资源指令

软件复位（RESET）指令

RESET 指令会复位整个器件，并使其可接收各条命令。I/O 模式配置为 SPI。所有非易失性寄存器或非易失性寄存器位都保持其各自的数值。所有易失性寄存器或易失性寄存器位的默认值为逻辑‘0’。该操作需要占用 t_{RESET} 的时长。没有任何存储 / 回读操作被执行。要想启动软件复位程序，需要使用复位使能（RSTEN）指令。这样可以防止所有意外复位。因此，软件复位是一个执行两条命令的序列。

注意：RSTEN 指令之后的任何命令（RESET 除外）将清除复位使能条件，并防止识别随后的 RESET 命令。

注意：如果将 WIP（SR[0]）位置于高电平并输入了 RSTEN/RESET 指令，则器件将忽略 RSTEN/RESET 指令。

注意： $\overline{\text{WP}}$ 和 NC（I/O3）引脚的功能由配置寄存器中的四线位 CR[1] 控制。如果将四线位配置为逻辑‘1’，则 $\overline{\text{WP}}$ 和 NC（I/O3）分别被配置为 I/O2 和 I/O3。在其他情况下，将配置 $\overline{\text{WP}}$ 和 NC（I/O3）功能。

表 9 汇总了软件复位后的器件状态。

表 9. 软件复位状态

状态 1	状态 2	状态 3	I/O 模式和寄存器位
待机	软件复位	待机	I/O 模式: SPI SRWD SR[7]: 与状态 1 相同 SNL SR[6]: 与状态 1 相同 TBPROT SR[5]: 与状态 1 相同 BP2 SR[4]: 与状态 1 相同 BP1 SR[3]: 与状态 1 相同 BP0 SR[2]: 与状态 1 相同 WEL SR[1]: 0 WIP SR[0]: 0 QUAD CR[1]: 与状态 1 相同

图 40. SPI 模式下的复位指令

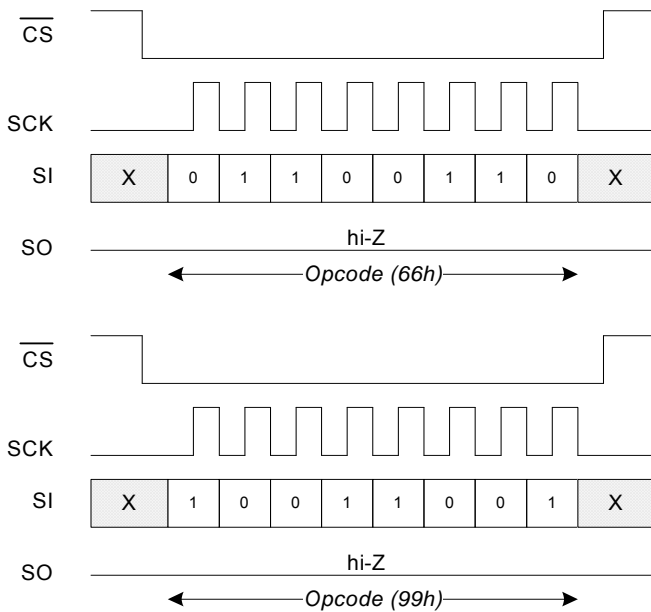


图 41. DPI 模式下的复位指令

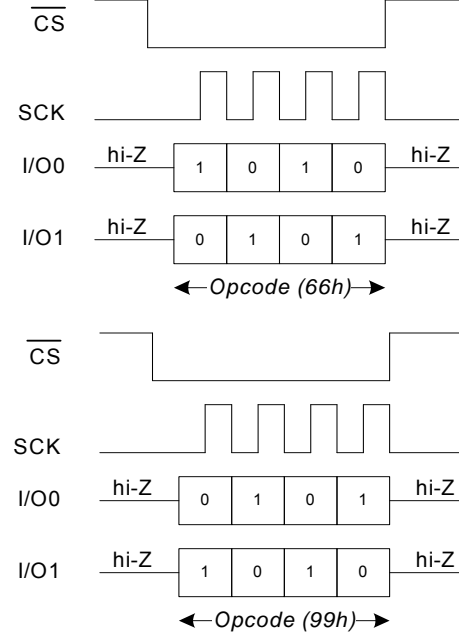
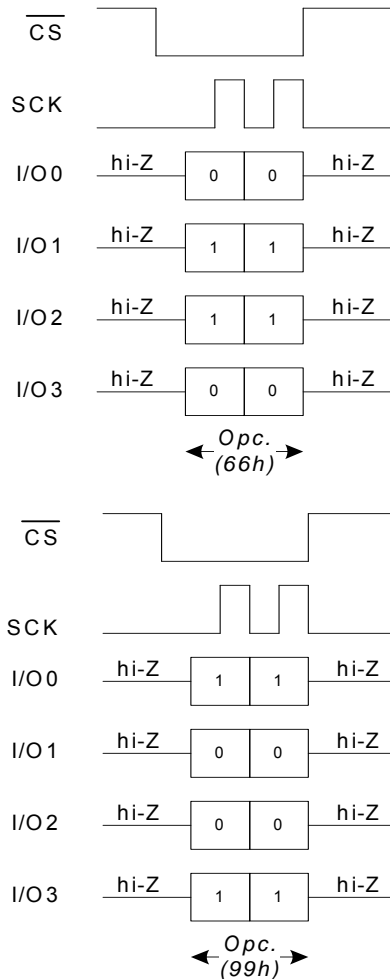


图 42. QPI 模式下的复位指令



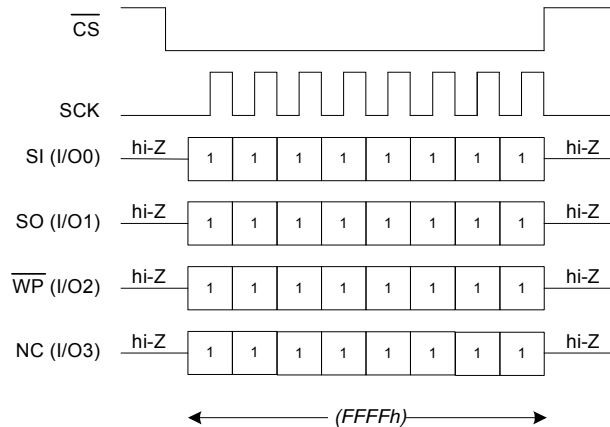
注意：在 QPI 模式下执行 RSTEN/RESET 指令前，必须将四线位 CR[1] 设置为逻辑 ‘1’。

默认的恢复指令

CY14V101PS 提供默认的恢复模式，从而允许器件返回 SPI 模式。在八个 SCLK 周期时间内，将一个逻辑高放置在所有 I/O (I/O3、I/O2、I/O1、I/O0) 上会使器件将进入已知模式 (SPI)，这样可以在未知起始模式的情况下，主机仍能与器件通信。

注意： \overline{WP} 和 NC (I/O3) 引脚的功能由配置寄存器中的四线位 CR[1] 控制。如果将四线位配置为逻辑 ‘1’，则 \overline{WP} 和 NC (I/O3) 分别被配置为 I/O2 和 I/O3。在其他情况下，将配置 \overline{WP} 和 NC (I/O3) 功能。

图 43. 默认的恢复指令



读取实时时钟 (RDRTC) 指令

通过读取 RTC (RDRTC) 指令能以高达 40 MHz 的 SPI 频率读取 RTC 寄存器中的内容。在 SPI 模式下，将 CS 线置于低电平并选取器件后，通过 SI 线发送 RDRTC 操作码，后跟八个用于选取寄存器的地址位。位于指定地址上的数据 (D7-D0) 被移出到 SO 线上。也可以使用 RDRTC 在突发模式下进行读取操作。从 RTC 寄存器中读取多个字节时，在到达最后一个 RTC 寄存器地址 (0x0F) 后，地址将翻转为 0x00。DPI 和 QPI 模式与 SPI 模式相同，但在 DPI 模式下，I/O1 和 I/O0 引脚被使用；在 QPI 模式下，I/O3、I/O2、I/O1 和 I/O0 引脚被使用。读取 RTC 计时寄存器前须将 RTC 标志寄存器中的 “R” 位设为 “1”，从而可以避免读取过渡数据。修改 RTC 标志寄存器时，需要一个写入 RTC 周期。完成读取操作后，必须将 R 位清除为 ‘0’。读取 RTC 寄存器的最简单方法是在突发模式下执行 RDRTC。从第一个 RTC 寄存器 (0x00) 开始读取，同时使 CS 保持为低电平状态，这样才能通过 SO 引脚传输来自 16 个 RTC 寄存器的数据。

注意：访问 RTC 结构后，RTC 地址通过加 ‘1’ 被更新。因此，更新在 RTC 结构中环绕。先访问 RTC 结构 (RTC 地址 ‘15’) 中的最后字节，然后访问第一个字节 (RTC 地址 ‘0’)。

图 44. SPI 模式下的 RDRTC 指令

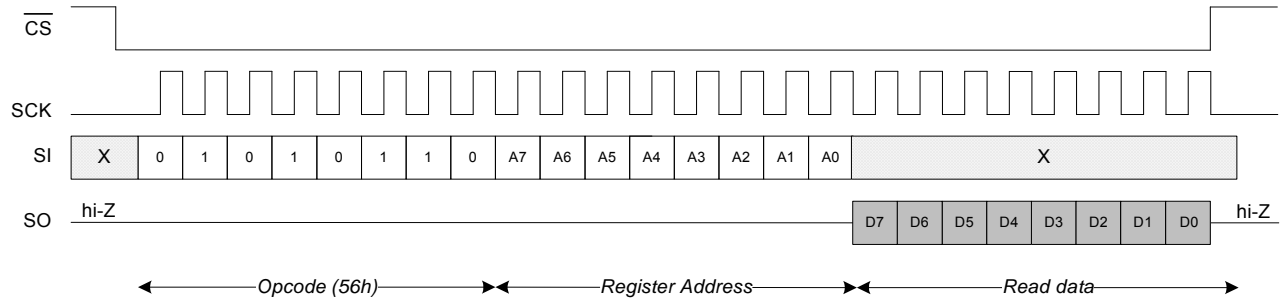


图 45. DPI 模式下的 RDRTC 指令

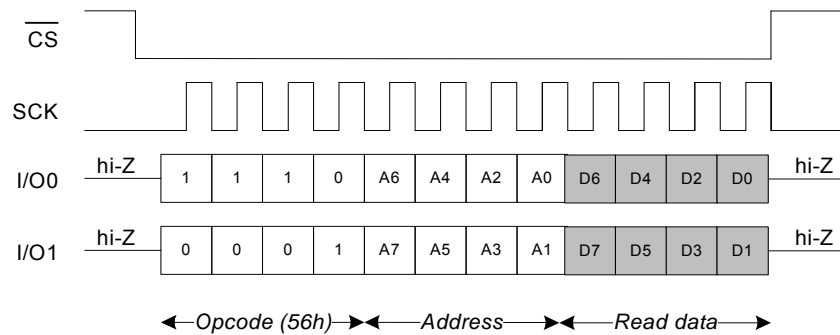
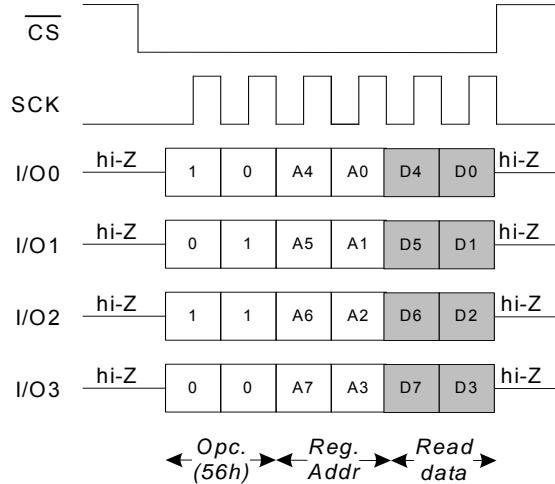


图 46. QPI 模式下的 RDRTC 指令



注意：在 QPI 模式下执行 RDRTC 指令前，必须将四线位 CR[1] 设置为逻辑 ‘1’。

快速读取实时时钟（FAST_RDRTC）指令

快速读取 RTC（FAST_RDRTC）指令与 RDRTC 相同，但它允许在操作码后面放置虚拟字节而且它的工作频率可达 108 MHz。

图 47. SPI 模式下的 RDRTC 指令

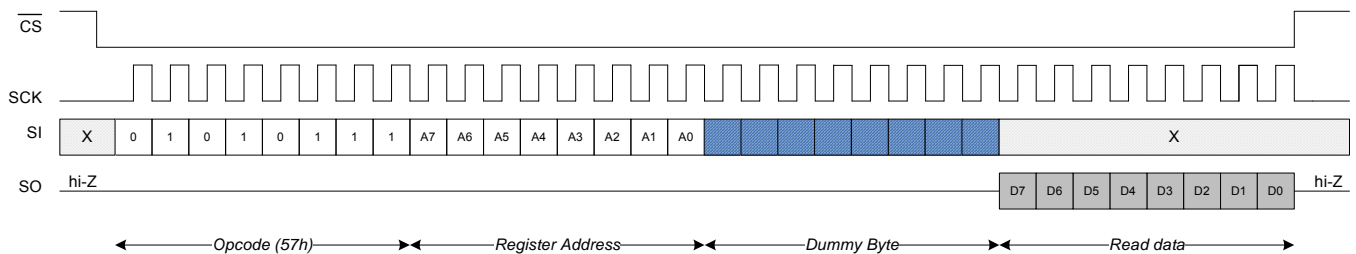
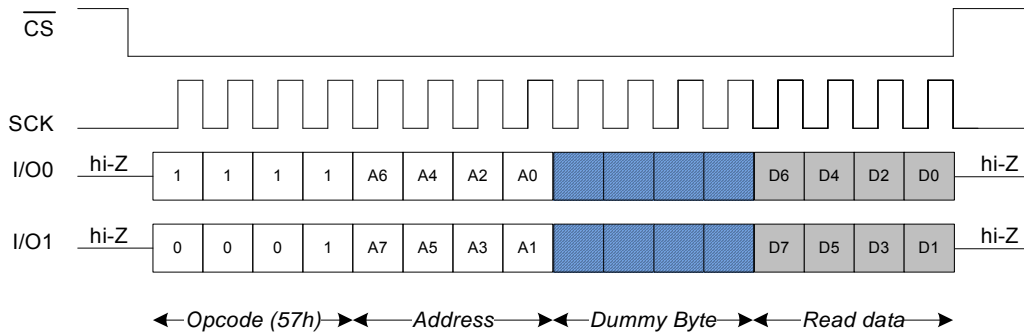
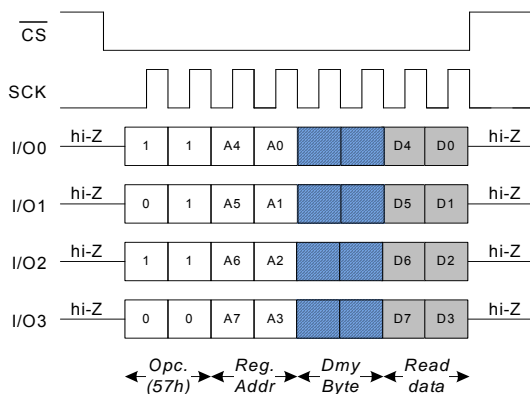


图 48. DPI 模式下的 FAST_RDRTC 指令

图 49. QPI 模式下的 FAST_RDRTC 指令


写入实时时钟 (WRRTC) 指令

通过写入 WRITE RTC (WRRTC) 指令可以修改 RTC 寄存器的内容。发送 WRRTC 指令前，需要将状态寄存器中的 WEL 位设置为 '1'。如果 WEL 位为 '0'，则需要发送 WREN 指令才能使用 WRRTC。在 SPI 模式下，将 CS 线置于低电平以选择器件后，可通过 SI 线发送 WRRTC 操作码，后跟八个用于识别需要写入哪个寄存器的地址位和一个或多个数据字节。WRRTC 允许在突发模式下进行写入操作。将多个字节写入 RTC 寄存器时，在到达最后一个 RTC 寄存器地址 (0x0F) 后，地址将翻转为 0x00。DPI 和 QPI 模式与 SPI 模式相同，但在 DPI 模式下，I/O1 和 I/O0 引脚被使用；在 QPI 模式下，I/O3、I/O2、I/O1 和 I/O0 引脚被使用。

注意：要想写入 RTC 计时寄存器和控制寄存器，需要将 W 位设置为 '1'。只有将 W 位清除为 '0' 后，RTC 寄存器中的值才会有效。完成 WRRTC 指令后，写入使能位 (WEN) 会自动被清除为 '0'。

注意：在 QPI 模式下执行 FAST_RDRTC 指令前，必须将四线位 CR[1] 设置为逻辑 '1'。

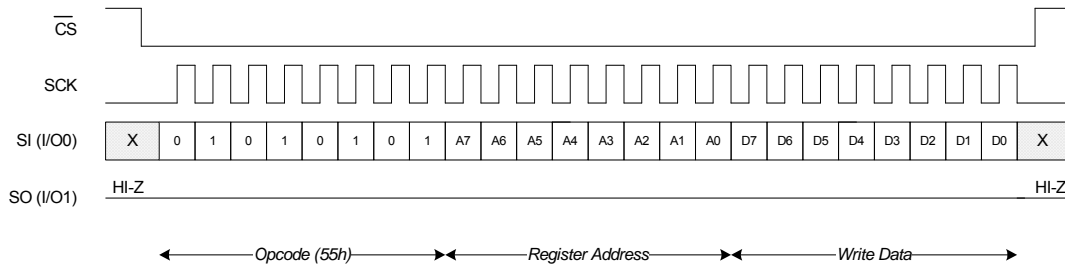
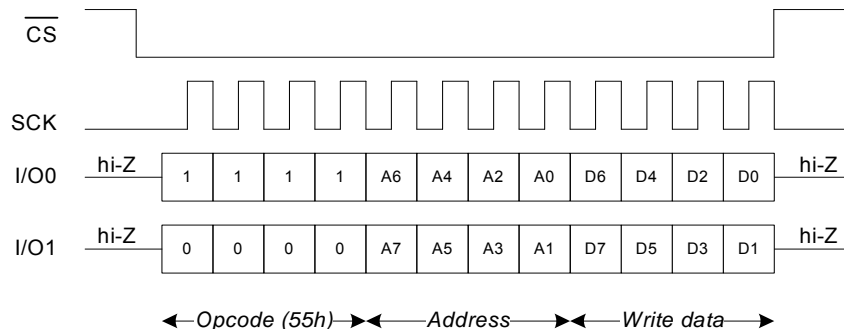
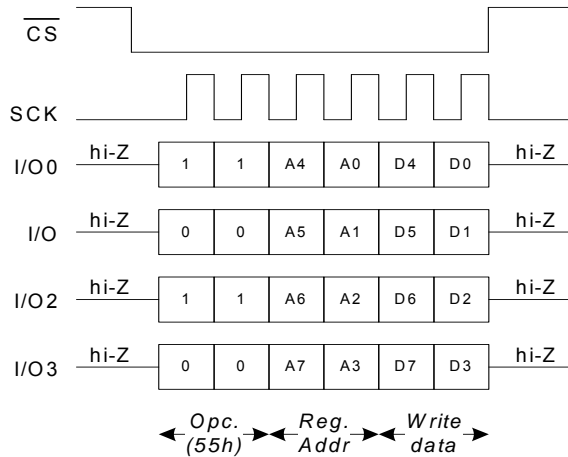
图 50. SPI 模式下的 WRRTC 指令

图 51. DPI 模式下的 WRRTC 指令


图 52. QPI 模式下的 WRTTC 指令



注意: 在 QPI 模式下执行 WRTTC 指令前, 必须将四线位 CR[1] 设置为逻辑 ‘1’。

表 10. 唤醒 (退出休眠) 状态

状态 1	状态 2	状态 3	I/O 模式和寄存器位
待机	休眠	待机	I/O 模式: 与状态 1 相同的模式 (SPI/DPI/QPI) SRWD SR[7]: 与状态 1 相同 SNL SR[6]: 与状态 1 相同 TBPROT SR[5]: 与状态 1 相同 BP2 SR[4]: 与状态 1 相同 BP1 SR[3]: 与状态 1 相同 BP0 SR[2]: 与状态 1 相同 WEL SR[1]: 0 WIP SR[0]: 0 QUAD CR[1]: 与状态 1 相同

图 53. SPI 模式下的 HIBEN 指令

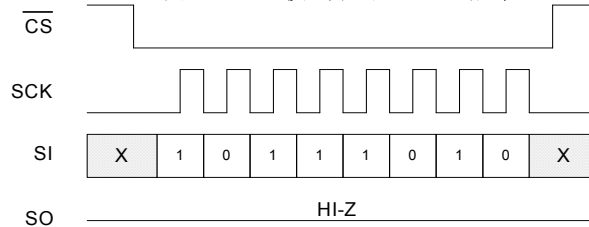
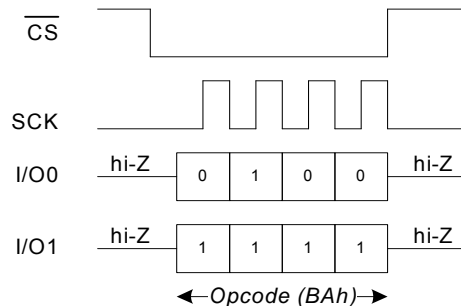


图 54. DPI 模式下的 HIBEN 指令



休眠 (HIBEN) 指令

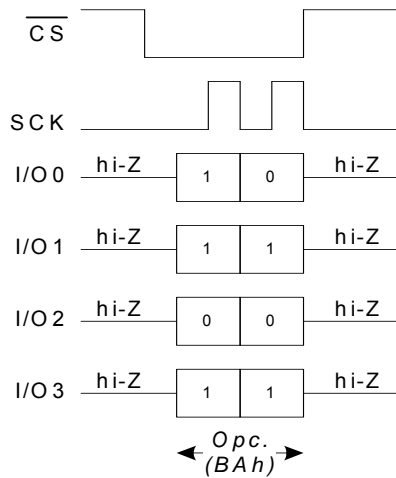
通过 HIBEN 指令可将 nvSRAM 进入休眠模式。发出 HIBEN 指令后, nvSRAM 需要 tSS 的时间来处理 HIBEN 请求。成功寄存和处理 HIBEN 命令后, nvSRAM 将 HSB 置于低电平并执行存储操作以将数据存入非易失性单元内, 然后进入休眠模式。从寄存 HIBEN 指令起, 器件在 t_{HIBEN} 时间后开始消耗 I_{ZZ} 电流。发出 HIBEN 指令后, 器件无法进行正常操作。在休眠模式下, 将忽略 SCK 和 SI 引脚, 并且将 SO 引脚置于高阻态 (HI-Z), 但是器件仍继续监控 CS 引脚。

要从休眠模式唤醒 nvSRAM, 必须通过将 CS 引脚从高电平跳转到低电平状态来选择器件。检测到 CS 引脚的下降沿后, 持续 t_{WAKE} 后, 器件被唤醒并可通过正常操作访问。器件唤醒后, 将返回到进入休眠模式前的工作模式。

注意: 只要进入休眠模式, nvSRAM 就会启动非易失性存储周期, 在每次执行休眠命令后都会产生耐久性周期。仅在上一个存储或回读周期之后对 SRAM 进行了写操作时, 存储周期才会开始。

表 10 汇总了从休眠器件状态唤醒的状态。

图 55. QPI 模式下的 HIBEN 指令



注意：在 QPI 模式下执行 HIBEN 指令前，必须将四线位 CR[1] 设置为逻辑 ‘1’。

表 11. 退出睡眠 (EXSLP) 状态

状态 1	状态 2	状态 3	I/O 模式和寄存器位
待机	睡眠	待机	I/O 模式：与状态 1 相同的模式 (SPI/DPI/QPI) SRWD SR[7]: 与状态 1 相同 SNL SR[6]: 与状态 1 相同 TBPROT SR[5]: 与状态 1 相同 BP2 SR[4]: 与状态 1 相同 BP1 SR[3]: 与状态 1 相同 BP0 SR[2]: 与状态 1 相同 WEL SR[1]: 与状态 1 相同 WIP SR[0]: 0 QUAD CR[1]: 与状态 1 相同

图 56. SPI 模式下的睡眠指令

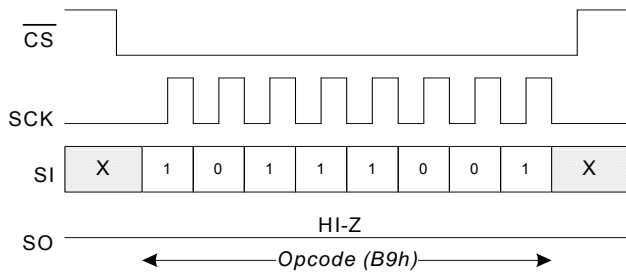
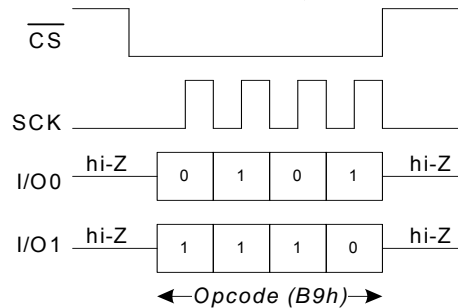


图 57. DPI 模式下的睡眠指令

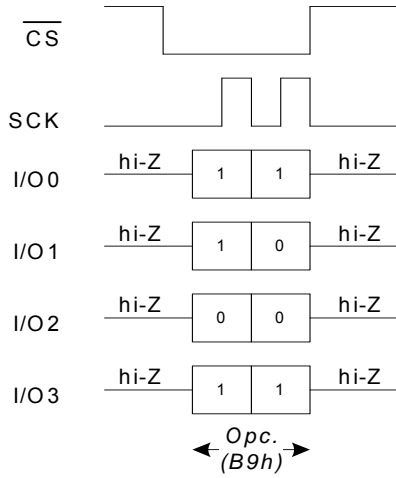
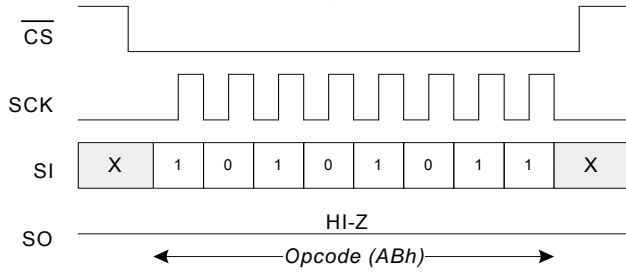
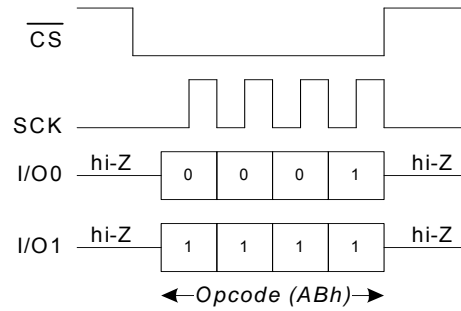
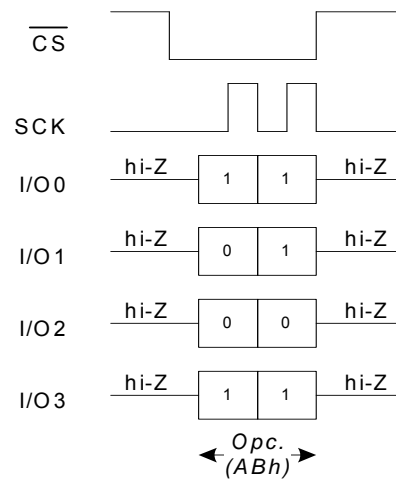


睡眠 (SLEEP) 指令

可通过睡眠指令使 nvSRAM 进入睡眠模式。发出睡眠指令后，nvSRAM 需要 t_{SLEEP} 的时间来处理 SLEEP 请求并开始消耗 I_{SLEEP} 电流。发出睡眠指令后，器件无法进行正常操作。在睡眠模式下，所有引脚均有效。

为了将 nvSRAM 从睡眠模式唤醒，必须输入 EXSLP 指令。经过 t_{EXSLP} 的时间后，可以对 nvSRAM 进行正常操作。器件唤醒后，将返回到进入睡眠模式前的工作模式。当器件在睡眠模式下，除了 EXSLP 指令和 RDSR 指令外，其他任何指令都被忽略。

表 11 汇总了器件从睡眠模式唤醒后的状态。

图 58. QPI 模式下的睡眠指令

图 59. SPI 模式下的 EXSLP 指令

图 60. DPI 模式下的 EXSLP 指令

图 61. QPI 模式下的 EXSLP 指令


寄存器指令

读取状态寄存器（RDSR）指令

通过使用 RDSR 指令能够以高达 108 MHz 的 SPI 频率访问状态寄存器。该指令用于检测器件的状态。

注意： 读取状态寄存器的最后一位后，该器件将返回到状态寄存器的第一位。

图 62. SPI 模式下的 RDSR 指令

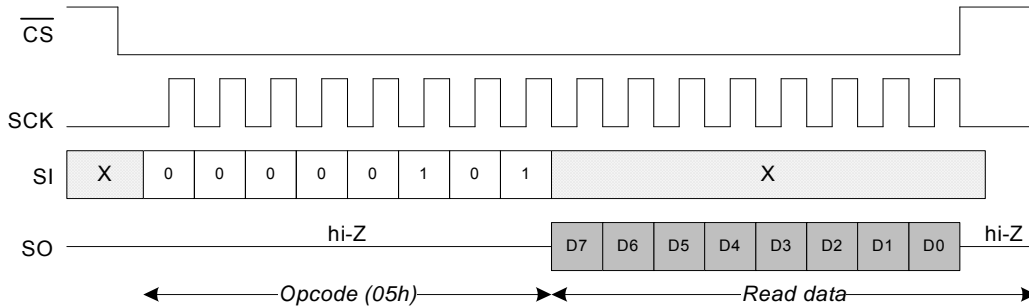


图 63. DPI 模式下的 RDSR 指令

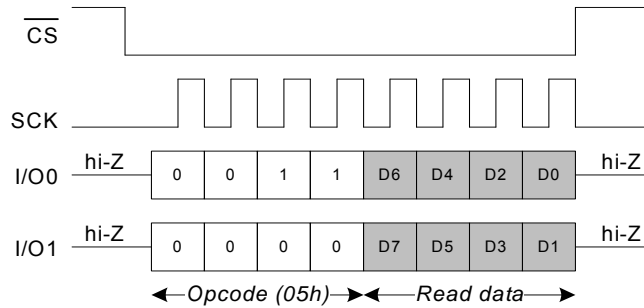
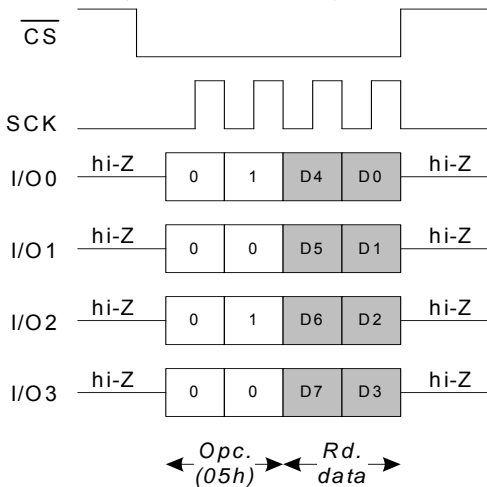


图 64. QPI 模式下的 RDSR 指令



写入状态寄存器（WRSR）指令

用户可通过 WRSR 指令对状态寄存器进行写操作。但该指令只能修改可写位，即：位 2（BP0）、位 3（BP1）、位 4（BP2）、位 5（TBPROT）、位 6（SNL）和位 7（SRWD）。WRSR 指令是一条写入指令，需要在发送该指令前将 WEL 位设置为 ‘1’（通过使用 WREN 指令实现）。WRSR 指令操作码在 CS 下降沿后发送，后面紧接的是要存储到状态寄存器中的 8 位数据。如上面所述，WRSR 指令只能修改状态寄存器的位 2、位 3、位 4、位 5、位 6 和位 7。

注意： 写入状态寄存器的有效值将在存储操作完成后被保存到非易失性存储器内。如果自动存储已禁用，必须进行软件存储操作以保存对状态寄存器所做的修改。

注意： 对状态寄存器进行写序列后，状态寄存器中的 WEL 位将被复位到 ‘0’。

Timing diagram for the SPI interface. The diagram shows the relationship between the CS (Chip Select), SCK (Serial Clock), SI (Serial Input), and SO (Serial Output) signals during a write operation.

- CS (Chip Select):** Active-low signal, shown as a low pulse during the transfer.
- SCK (Serial Clock):** Continuous clock signal.
- SI (Serial Input):** Data being written to the device. The sequence is: **X** (start), 0, 0, 0, 0, 0, 0, 0, 1, D7, D6, D5, D4, D3, D2, D1, D0, **X** (end). The first and last bits are marked as **X** (unknown or don't care).
- SO (Serial Output):** Output signal. It is in a high-impedance state (HI-Z) during the write operation.
- Data Flow:** The data is divided into two sections: **Opcode (01h)** and **Write Data**.

The diagram illustrates the timing for the I/O operation. The CS (Chip Select) signal is active low. The SCK (Serial Clock) signal is a clock signal. The I/O data bus is divided into two sections: the first four bits (0, 0, 0, 1) represent the Opcode (01h), and the next four bits (D6, D4, D2, D0) represent the Write data. The CS signal is active low, and the SCK signal is a clock signal. The I/O data bus is labeled hi-Z when not active.

通过使用 RDCR 指令能够以高达 108 MHz 的 SPI 频率访问配置寄存器。下图显示的是 SPI、DPI 和 QPI 模式下的配置寄存器指令传输波形。

注意：在读取配置寄存器的最后一位之后，器件将返回到配置寄存器的第一位。

The timing diagram illustrates the communication protocol for the 24C02. The **CS** signal is active-low. The **SCK** signal is a square wave. The **I/O0**, **I/O1**, **I/O2**, and **I/O3** signals are bidirectional data lines. The diagram shows a sequence of operations: a read cycle (Opc. 01h) and a write cycle (Wr. data).

Timing diagram for the 2-wire interface. The diagram shows the relationship between the Chip Select (CS), Serial Clock (SCK), Serial Input (SI), and Serial Output (SO) signals during a read operation.

- CS (Chip Select):** Active-low signal, shown as a low pulse.
- SCK (Serial Clock):** Periodic clock signal.
- SI (Serial Input):** Data being sent to the device. It starts with a start bit (X), followed by the opcode `00110101` (35h), and then a shaded area representing the read data.
- SO (Serial Output):** Data being received from the device. It is in a high-impedance state (hi-Z) until the start of the read data, then outputs the data bits D7 through D0.

The diagram illustrates the sequence of events: the CS signal is asserted, the SCK signal is provided, the SI signal is driven with the opcode and read data, and the SO signal outputs the read data.

图 69. DPI 模式下的 RDCR 指令

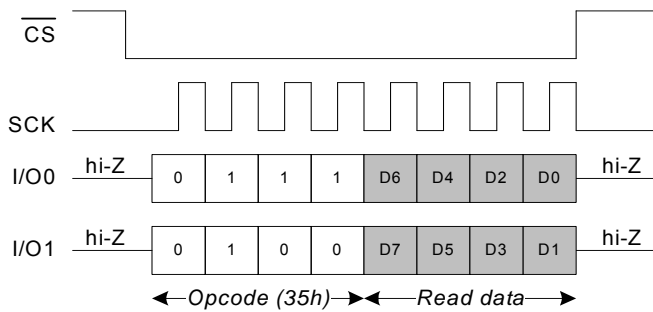
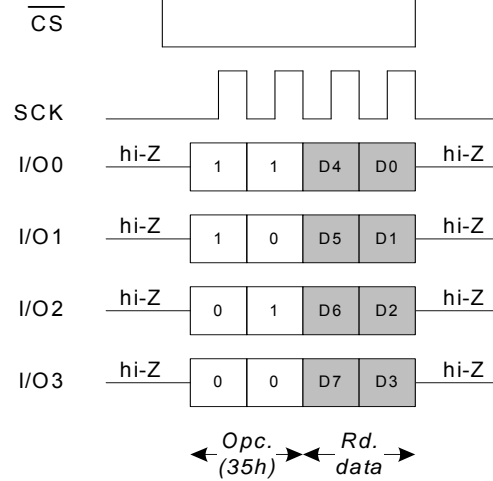


图 70. QPI 模式下的 RDCR 指令



注意：在 QPI 模式下执行 RDCR 指令前，必须将四线位 CR[1] 设置为逻辑 ‘1’。

写入配置寄存器（WRCR）指令

通过设置四线位，用户可以使用写入配置寄存器（WRCR）指令更改器件的数据宽度。当使用读取四线输出、四线 I/O 读取和四线输入写入命令时，必须将四线位设置为 1。四线位是非易失性的。

注意：使能 QPI 模式（QPIEN）指令不会设置配置寄存器中的四线位。

注意：建议始终使用表 8 中所提供的 RFU 位。

图 71. SPI 模式下的 WRCR 指令

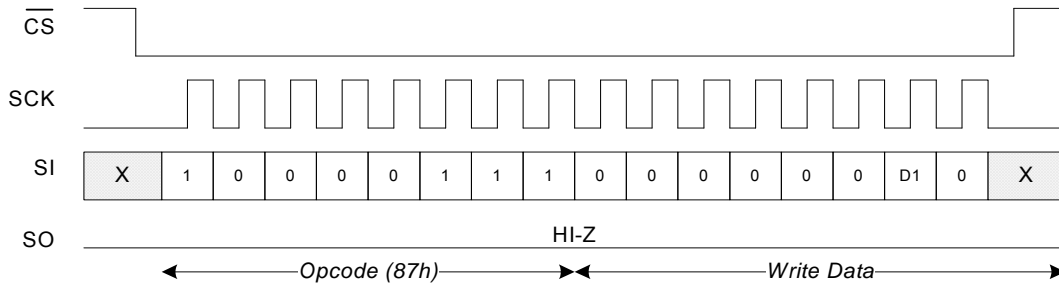
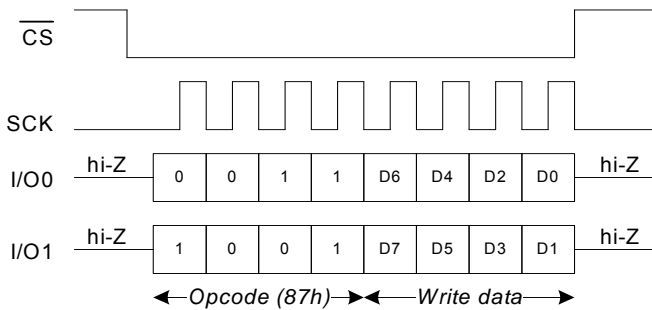


图 72. DPI 模式下的 WRCR 指令



标识寄存器（RDID）指令

通过使用 RDID 指令能以高达 40 MHz 的 SPI 频率读取 JEDEC 分配的制造商 ID 和产品 ID。该指令可用于识别总线上的器件。CS# 进入低电平状态后，通过移入 RDID 的操作码可以发出 RDID 指令。

器件 ID 是 4 字节的只读代码，用于独特标识 1 Mbit QPI nvSRAM 产品。它包括产品系列号、产品配置和密度。

RDID 命令读取 4 字节的器件 ID 结构（不能写入到该结构）。每次可以访问该结构的一个字节。第一个被访问的字节是该结构的

最高有效字节 ID[31:24]，第二个被访问的字节是 ID[23:16]，... 最后被访问的字节是 ID[7:0]。

注意：由于始终按相同的顺序访问该结构，因此不需要进行地址传输。但是一个内部的 2 位地址指针被使用。当解码该操作码时，该指针被初始化为“0”。访问每个字节后，内部地址指针将递增。地址指针从“3”环绕到“0”；在访问第四个字节 ID[7:0] 后，将访问第一个字节 ID[31:24]。可以在 SPI、DPI 或 QPI 模式下发出该命令。

表 12. 器件标识

器件	制造商 ID	产品 ID	容量	裸片版本
	31-21	20-7	6-3	2-0
	11 位	14 位	4 位	3 位
CY14V101PS	00000110100	00001110000001	0100	001

图 73. SPI 模式下的 RDID 指令

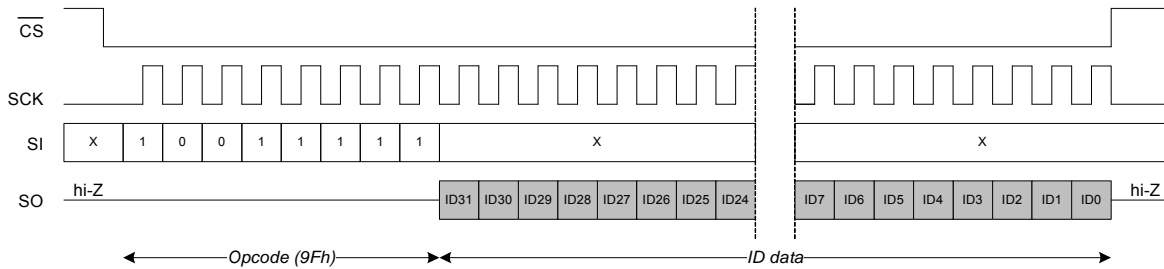


图 74. DPI 模式下的 RDID 指令

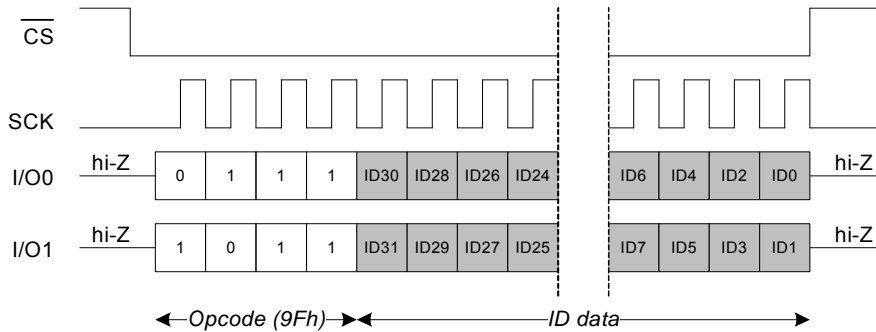
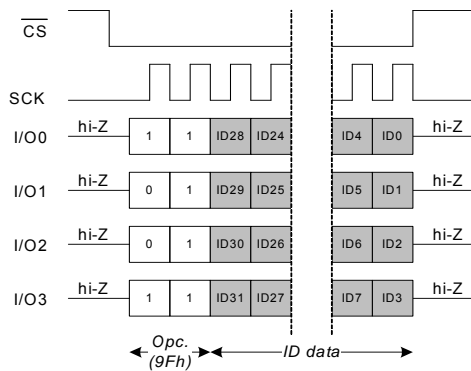


图 75. QPI 模式下的 RDID 指令



注意：在 QPI 模式下执行 RDID 指令前，必须将四线位 CR[1] 设置为逻辑“1”。

标识寄存器（FAST_RDID）指令

FAST_RDID 指令与 RDID 指令相同，但它允许在操作码后放置一个虚拟字节。通过使用 FAST_RDID 指令能以高达 108 MHz 的 SPI 频率读取 JEDEC 分配的制造商 ID 和产品 ID。

图 76. SPI 模式下的 FAST_RDID 指令

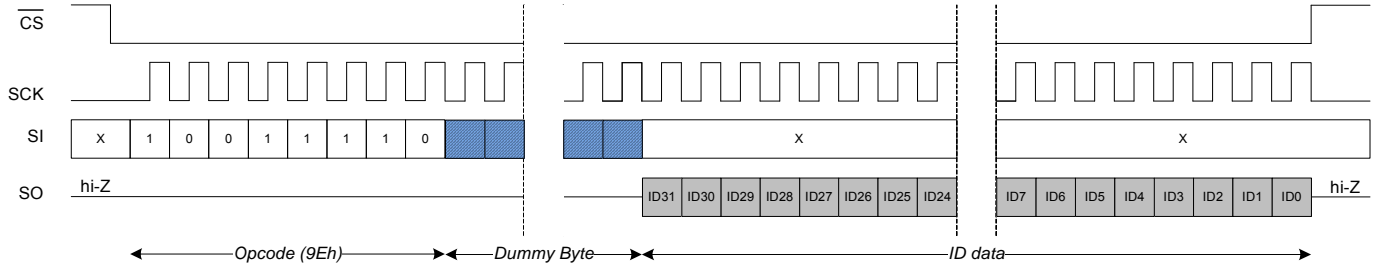


图 77. DPI 模式下的 FAST_RDID 指令

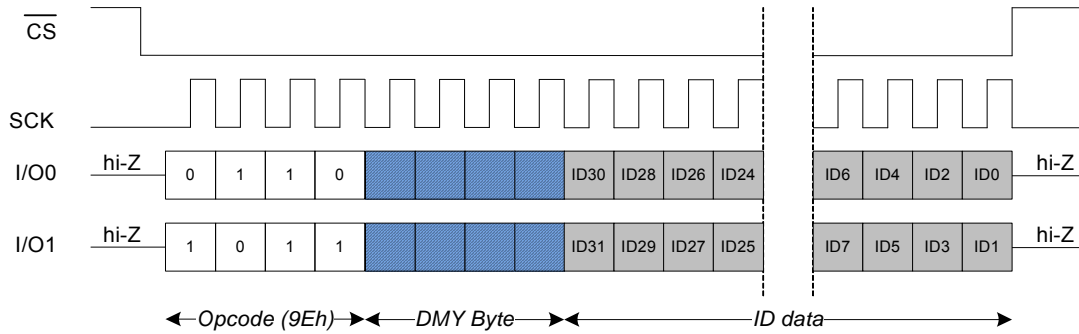
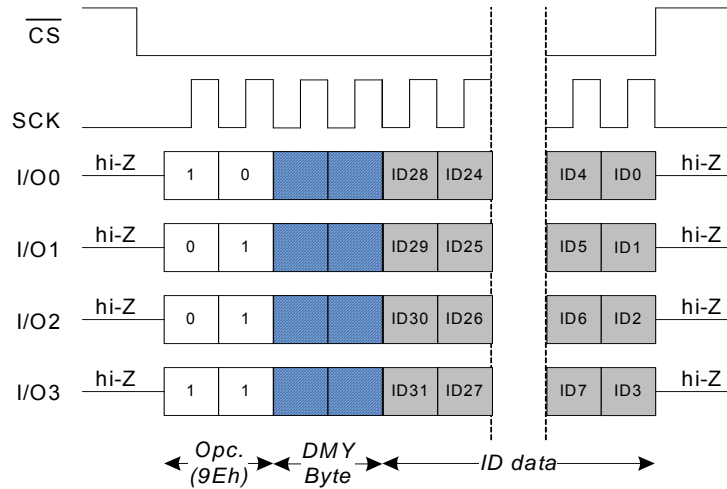


图 78. QPI 模式下的 FAST_RDID 指令



序列号寄存器写入（WRSN）指令

序列号为提供给用户以对器件进行唯一标识的 8 字节可编程存储空间。它通常由一个两字节的客户 ID、五字节的唯一序列号及一字节的循环冗余校验组成。器件不计算循环冗余校验，系统设计者可通过任何方式利用八个字节的存储空间。八个字节位置的默认值为“0x00”。

通过使用 WRSN 命令可以编写该序列号。要想编写序列号，必须使用 WREN 命令使能写功能。通过在突发模式下使用 WRSN 命令可以编写 8 字节的序列号。编写序列号的最后字节后，器件将返回到序列号的第一个（MSB）字节。可使用状态寄存器的 SNL 位锁定序列号。该位被置为“1”后，无法对序列号进行任何修改。将 SNL 位设置为“1”后，使用 WRSN 命令不会对序列号产生任何影响。执行该命令前需要设置 WEL 位。如果未将状态寄存器中的 SRWD 位设置为“1”，那么在执行该命令后，

WEL 位将复位为“0”。可以在 SPI、DPI 或 QPI 模式下发出该命令。

通过使用 WRSN 指令能够以高达 108 MHz 的 SPI 频率写入序列号。

注意：需要通过存储操作（自动存储或软件存储）将序列号存储到非易失性存储器中。如果已禁用自动存储，则必须执行软件存储操作来保存和锁定序列号。如果 SNL 位被设置为“1”，并且未被保存（自动存储被禁用），那么在下次电源循环中，SNL 位和序列号将默认被设置为“0”。如果 SNL 位被设置为“1”并保存其设置，无法将它清除为“0”。执行该指令前需要设置 WEL 位。可以在 SPI、DPI 或 QPI 模式下发出该指令。

注意：完成该指令后，WEL 位被复位为“0”。

图 79. SPI 模式下的 WRSN 指令

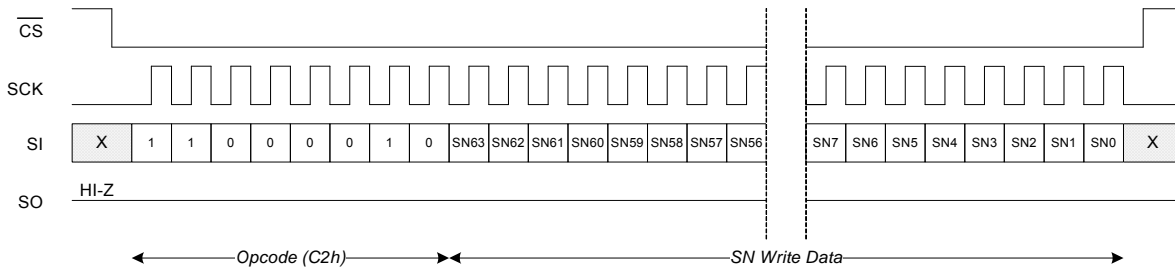


图 80. DPI 模式下的 WRSN 指令

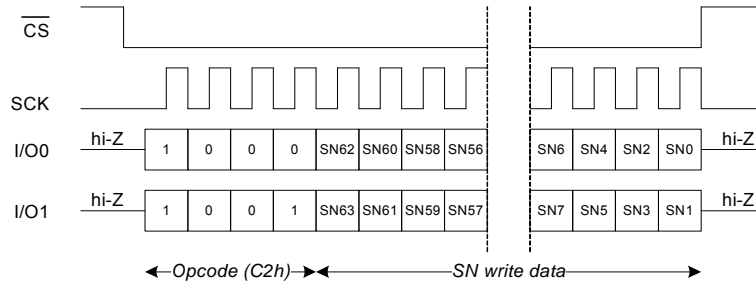
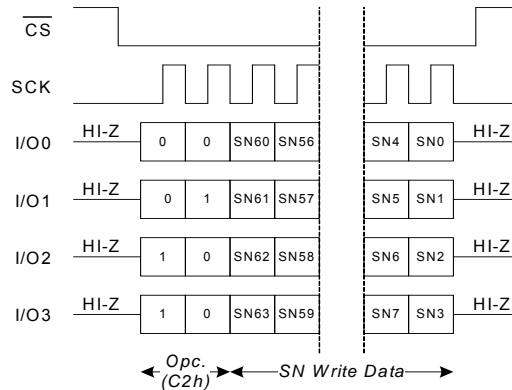


图 81. QPI 模式下的 WRSN 指令



序列号寄存器读取（RDSN）指令

通过使用 RDSN 指令能够以高达 40 MHz 的 SPI 频率读取序列号。可以在突发模式下读取序列号，从而可以一次性读取全部八个字节。读取序列号的最后一个字节后，器件将返回到序列号的

第一个（MSB）字节。CS 进入低电平状态后，通过移入 RDSN 的操作码可以发送 RDSN 指令。此后，nvSRAM 将移出序列号的八个字节。可以在 SPI、DPI 或 QPI 模式下发出该指令。

图 82. SPI 模式下的 RDSN 指令

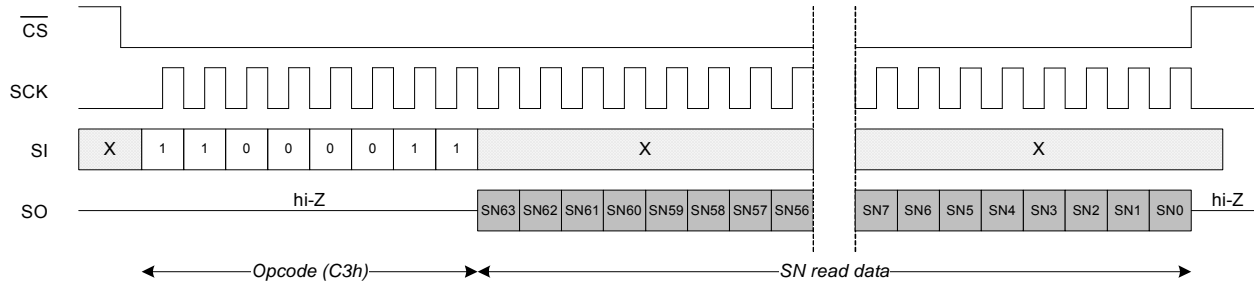


图 83. DPI 模式下的 RDSN 指令

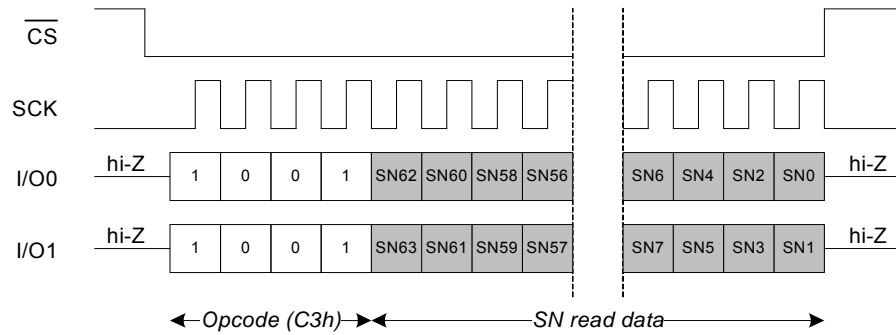
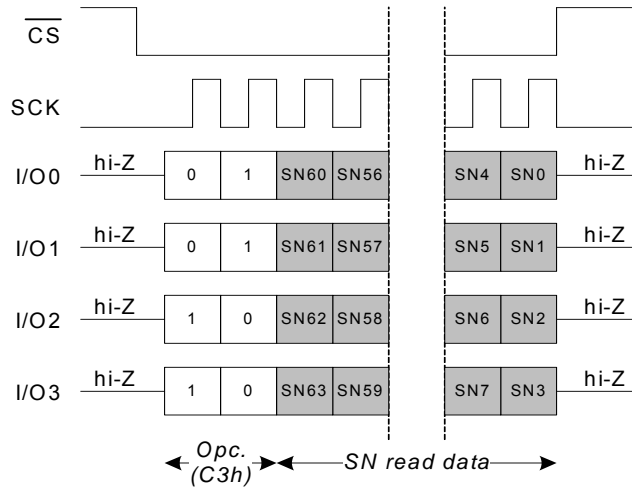


图 84. QPI 模式下的 RDSN 指令



注意: 在 QPI 模式下执行 RDSN 指令前, 必须将四线位 CR[1] 设置为逻辑 ‘1’。

快速读取序列号寄存器（FAST_RDSN）指令

FAST_RDSN 指令与 RDSN 指令相同，但它允许在操作码后放置一个虚拟字节。能够以高达 108 MHz 的频率使用 FAST_RDSN 指令。

图 85. SPI 模式下的 FAST_RDSN 指令

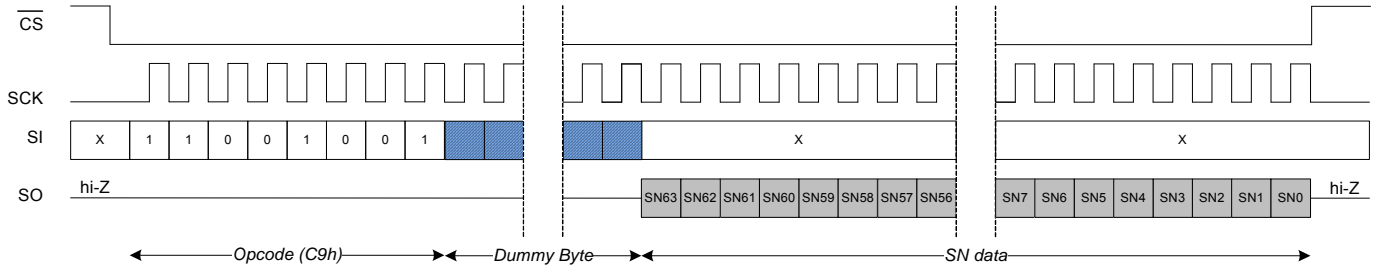


图 86. DPI 模式下的 FAST_RDSN 指令

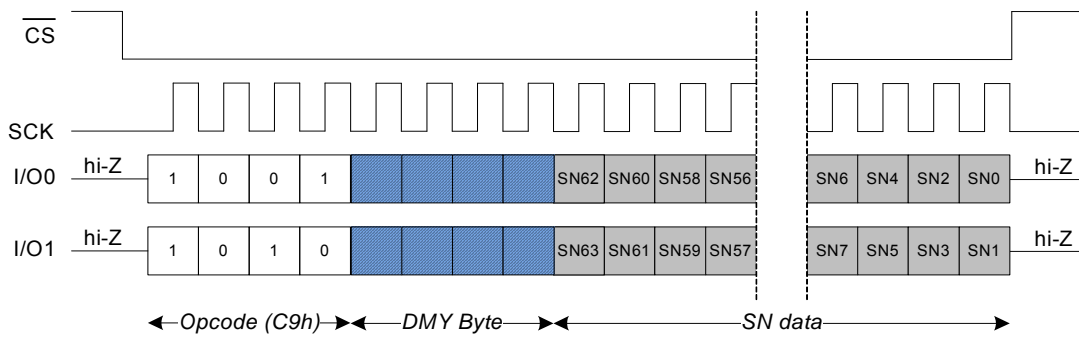
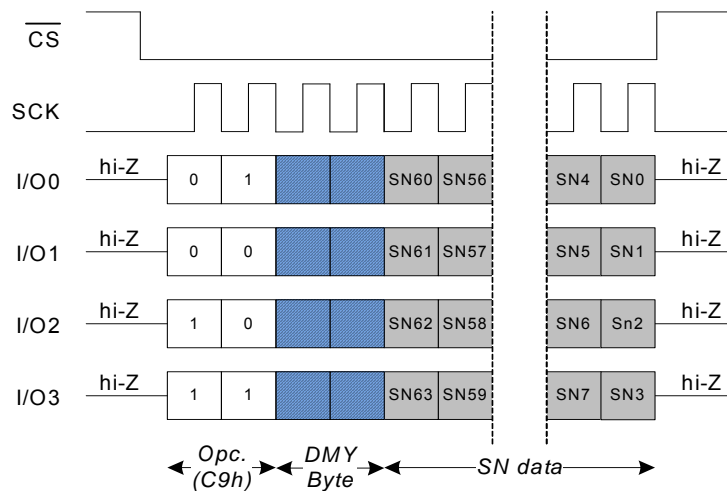


图 87. QPI 模式下的 FAST_RDSN 指令



NV 特定指令

nvSRAM 器件提供了四条用于访问 nvSRAM 特定功能的特殊指令，它们分别为：STORE、RECALL、ASEN 和 ASDI。

软件存储（STORE）指令

执行 STORE 指令后，nvSRAM 进行软件存储操作。无论从上次存储或回读操作后是否发生了写操作，都会进行存储操作。要想发送该指令，器件必须使能写入功能（WEL 位 = ‘1’）。可以在 SPI、DPI 或 QPI 模式下发送该指令。

注意：发送 STORE 指令后，将在 \overline{CS} 的上升沿上清除 WEL 位。

图 88. SPI 模式下的存储指令

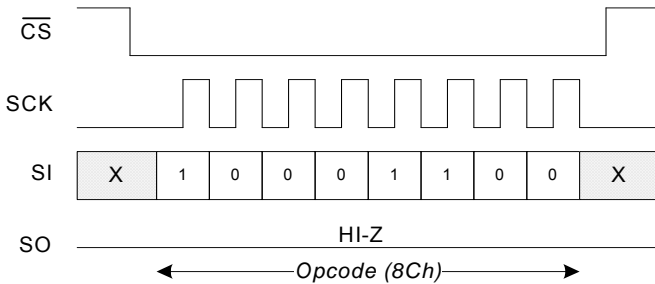
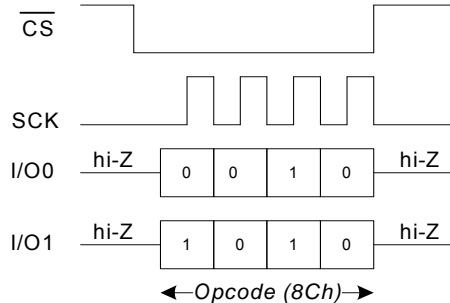


图 89. DPI 模式下的存储指令



软件回读（RECALL）指令

执行 RECALL 指令后，nvSRAM 将执行软件回读操作。要想发送该指令，器件必须使能写入功能（WEL = ‘1’）。可以在 SPI、DPI 或 QPI 模式下发送该指令。

注意：发送 RECALL 指令后，将在 \overline{CS} 的上升沿上清除 WEL 位。

图 90. QPI 模式下的存储指令

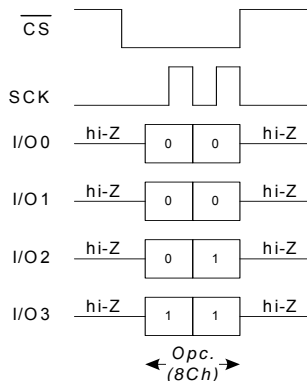


图 91. SPI 模式下的回读指令

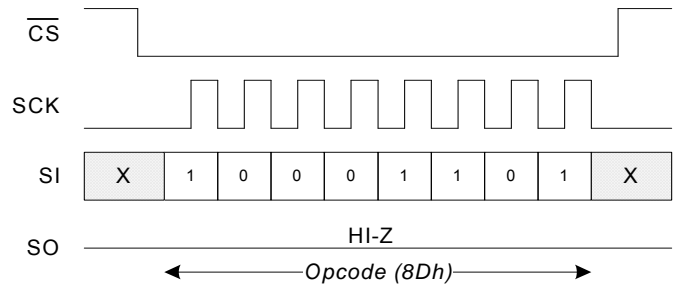


图 92. DPI 模式下的回读指令

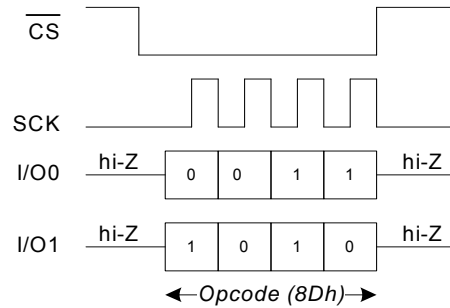
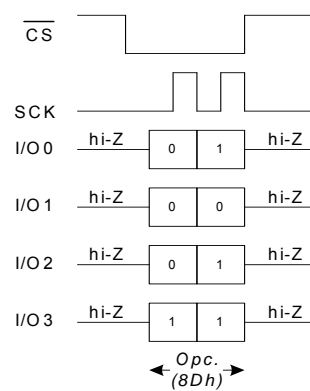


图 93. QPI 模式下的回读指令



自动存储使能 (ASEN) 指令

通过自动存储使能指令可以在 nvSRAM 器件上使能自动存储功能。该设置不是非易失性操作，随后需要一个存储序列来保持加电周期。想要发出该指令，器件必须使能写入功能 (WEL = '1')。可以在 SPI、DPI 或 QPI 模式下发出该指令。

注意：如果执行 ASDI 和 ASEN 指令，在软件序列处理期间 (t_{SS})，器件将处于繁忙状态。

注意：发出 ASEN 指令后，将在 \overline{CS} 的上升沿上清除 WEL 位。

图 94. SPI 模式下的 ASEN 指令

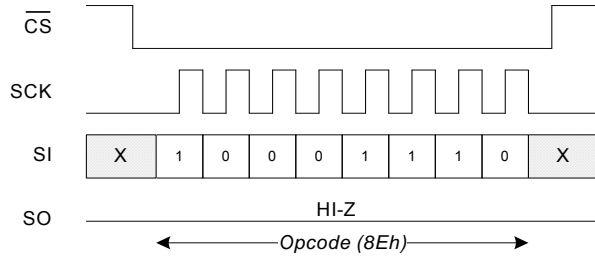


图 95. DPI 模式下的 ASEN 指令

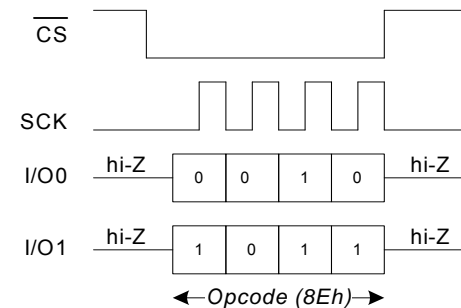
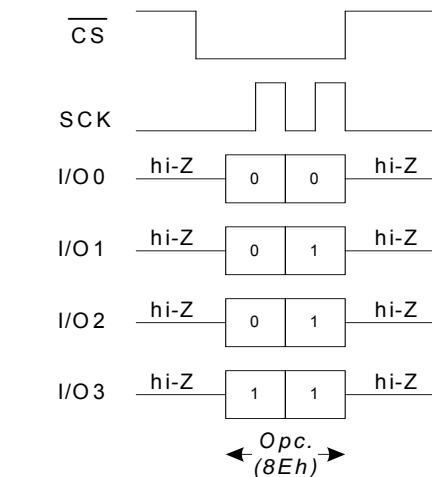


图 96. QPI 模式下的 ASEN 指令



自动存储禁用 (ASDI) 指令

在该器件中默认使能自动存储功能。ASDI 指令禁用自动存储功能。该设置不是非易失性操作，随后需要一个存储序列来保持加电周期。要想发送该指令，器件必须使能写入功能 (WEL = '1')。可以在 SPI、DPI 或 QPI 模式下发出该指令。

注意：发送 ASDI 指令后，将在 \overline{CS} 的上升沿上清除 WEL 位。

图 97. SPI 模式下的 ASDI 指令

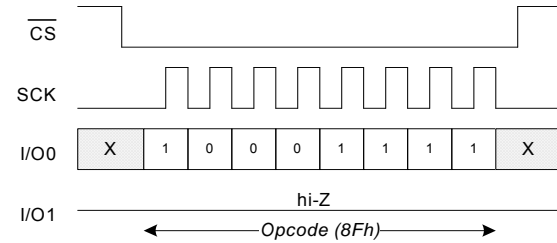


图 98. DPI 模式下的 ASDI 指令

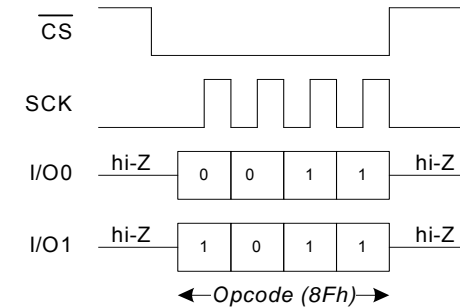
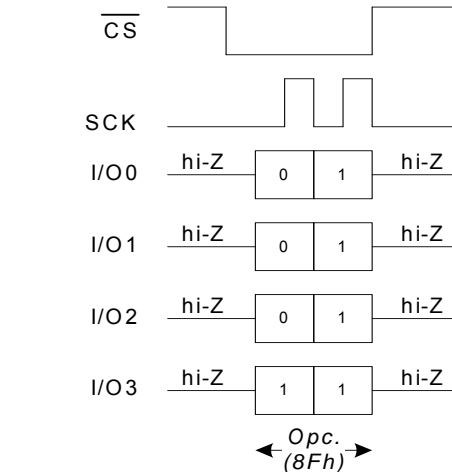


图 99. QPI 模式下的 ASDI 指令



注意：在 QPI 模式下执行 ASDI 指令前，必须将四线位 CR[1] 设置为逻辑 '1'。

实时时钟操作

nvTIME 操作

该器件提供了具有时钟、警报、看门狗、中断和控制功能的内部寄存器。RTC 寄存器占用单独的 nvSRAM 地址空间，可通过寄存器地址为 0x00 至 0x0F 的 RTC 寄存器读和写序列进行访问。计时寄存器间的内部双缓冲可阻止在读或写期间访问被传输的内部时钟数据。双缓冲技术还避免了在访问时钟数据期间影响正常的定时计数或内部时钟的准确性。时钟和警报寄存器以 BCD 格式存储数据。

时钟操作

时钟寄存器以一秒的增量保存时间，最长达 9,999 年。时间可被设置为任何一种日历时间，并且时钟会自动记录某月某日、某周某日、闰年及世纪转换。共有八个专用于时钟功能的寄存器，可用于使用写周期设置时间以及基于读周期读取时间。这些寄存器包含 BCD 格式的日期。定义为 ‘0’ 的位目前不可用，并被保留以供赛普拉斯将来使用。

读取时钟

双缓冲 RTC 寄存器结构降低了从时钟读取错误数据的可能性。当读取位 “R”（位于标志寄存器 0x00 位置）设置为 “1” 时，在读取时钟数据前对器件的计时寄存器的内部更新将停止，以防止读取过渡数据。停止寄存器的更新不会影响时钟的准确度。

当 RTC 器件读取序列被启动后，用户计时寄存器更新将被停止，直到向读取位 “R”（在位于 0x00 的标志寄存器中）写入 “0” 后才会重新开始更新。读取序列完成后，所有 RTC 寄存器在 20 ms 内同时更新。

设置时钟

对 RTC 器件进行写访问将使计时寄存器更新停止，而当写入位 “W”（位于标志寄存器 0x00 位置）设置为 “1” 时允许用户设置时间。然后，正确的星期、日期和时间被写入到寄存器内，且必须为 24 小时 BCD 格式。写入的时间被称为 “基准时间”。该值保存在非易失性寄存器中，用于计算当前时间。通过向 W 位写入 “0” 来清除该位时，计时寄存器的值将被传输到实际的时钟计数器内，然后该时钟会恢复正常运行。

如果写入 RTC 寄存器的时间不是正确的 BCD 格式，则 RTC 寄存器中每个无效的半字节在翻滚至 0x0 前继续计数至 0xF，然后 RTC 寄存器恢复正常操作。

注意：“W” 位被设置为 “0” 后，写入计时、警报、校准和中断寄存器的值在 t_{RTC0} 时间后会被传输到 RTC 计时计数器中。这些计数器值必须通过启动软件 / 硬件存储或自动存储操作保存在非易失性存储器中。在 “自动存储禁用” 模式下， t_{RTC0} 时间后将执行存储操作，同时写入到 RTC 寄存器内以正确记录所进行的修改。

备用电源

器件中的 RTC 适合永久性带电操作。将 V_{RTCbat} 或 V_{RTCbat} 引脚连接到某个电池上。推荐使用 3 V 的锂电池，当主电源被移除时，器件会使用该电池提供的电流。然而，它在任何时候都不会对电池进行充电。必须根据系统生命周期期间总的预期累计断电时间选择电池容量。

当主电源 V_{CC} 断电并下降至 V_{SWITCH} 以下时，器件会切换至备用电源。时钟振荡器消耗的电流非常少，因此最大程度地延长了备用电源的供电时长。主电源被移除后，无论时钟操作如何，存储在 nvSRAM 中的数据都是安全的，因为断电后这些数据将被存储在非易失性单元中。在备用电源操作期间，室温下消耗 0.45 μA （典型值）的电流。

注意：如果电池在连接到 V_{CC} 前先连接了 V_{RTCbat} 引脚，则芯片将吸收高达 I_{BAK} 的电流。尽管禁用了振荡器，仍会发生此情况。为以最大程度延长电池寿命，将电池接到 V_{RTCbat} 引脚前，必须先将其连接到 V_{CC} 。

停止和启动振荡器

校准寄存器中 0x08 位置的 OSCEN 位控制振荡器的使能和禁用。该位是非易失性的，交付给客户时处于 “使能”（设置为 “0”）状态。系统长期存放不用时，为了保持电池寿命，须将 OSCEN 设置为 “1”。这样可关闭振荡器电路，以延长电池寿命。如果 OSCEN 位从 “禁用” 变为 “使能”，振荡器大约需要一秒钟（最多两秒）的时间进行启动。

当系统电源被关闭时，如果备用电源（ V_{RTCcap} 或 V_{RTCbat} ）的电压降低到各自最低值以下，那么振荡器可能会掉电。当系统电源恢复时，器件能够检测振荡器是否掉电。该过程会被记录在标志寄存器中 0x00 位置的振荡器掉电标志（OSCF）内。当系统通电（ V_{CC} 大于 V_{SWITCH} ）时，会检查 OSCEN 位是否处于 “使能” 状态。如果 OSCEN 位处于 “使能” 状态并且在 5 ms 内未激活振荡器，那么 OSCF 位将被设置为 “1”。系统必须检查该条件，然后写入 “0” 以清除标志。

注意：除设置 OSCF 标志位外，时间寄存器被复位为 “基准时间”，该值是上次吸入计时寄存器的值。控制寄存器或校准寄存器与 OSCEN 位不受 “振荡器失败” 条件的影响。

首次对时间寄存器进行写入时，OSCF 值必须重置为 “0”。这将初始化该位的状态（系统首次加电时可能已设置）。

要想复位 OSCF，需要将 “W” 位（在位于 0x00 的标志寄存器中）设置为 “1” 以便使能对标志寄存器的写操作。向 OSCF 位写入 “0”，然后将 W 位复位为 “0”，从而禁用写操作。

校准时钟

RTC 由一个石英控制的晶振驱动，该晶振的额定频率为 32.768 kHz。时钟的准确度取决于晶振质量和校准。市场上的晶振通常有 ± 20 ppm 到 ± 35 ppm 的误差。然而，器件应用一种在 25 °C 下可将准确度提高至 $\pm 1/-2$ ppm 的校准电路。这表示每个月有 +2.5 秒到 -5 秒的误差。

校准电路通过增加或减少振荡器分频器电路的计数来达到这种准确度。抑制（减少，负校准）或拆分（增加，正校准）的脉冲数量取决于加载到位于 0x08 的校准寄存器中的五个校准位的值。校准位占用校准寄存器中的五个低位。这些位被设置为以二进制形式表示的 0 和 31 之间的任何值。D5 位是符号位，其中 ‘1’ 表示正校准，‘0’ 表示负校准。增加计数可使时钟加速，减少计数可使时钟减速。如果将一个二进制 ‘1’ 加载到寄存器中，实现对应振荡器误差中 4.068 或 -2.034 ppm 偏移的调整，具体大小取决于符号。

校准周期时间为 64 分钟。对于周期内的前 62 分钟（每分钟一次），可能会每 128 个振荡器周期缩短一秒或每 256 个振荡器周期延长一秒。如果将二进制 ‘1’ 加载到寄存器中，只会修改周期时长为 64 分钟的前两分钟。如果将二进制 6 加载到寄存器中，则影响前 12 分钟，以此类推。因此，每个校准步骤对每 125、829 和 120 个实际振荡器周期具有增加 512 个或减少 256 个振荡器周期的作用，即校准寄存器中的每个校准步骤有 4.068 或 -2.034 ppm 的调整。

为了保证所需校准，标志寄存器（0x00）中的 CAL 位必须设置为 ‘1’。这便导致 INT 引脚以 512 Hz 的额定频率进行切换。任何偏离 512 Hz 的偏差表示的都是所需纠正的大小和方向。例如，读数 512.01024 Hz 表示误差为 +20 ppm。因此，必须将十进制值 -10（001010b）加载到校准寄存器中来抵消该误差。

注意：设置或改变校准寄存器不影响测试输出频率。

要想设置或清除 CAL，需要将 W 位（在位于 0x00 的寄存器中）设置为 ‘1’，从而使能对标志寄存器的写入操作。将某个值写入到 CAL 内，然后将 W 位复位为 ‘0’，从而禁用写入操作。

警报

警报功能将用户编写的警报时间值和日期（存储在寄存器 0x01-5 中）与相应的时间和日期值进行比较。当这两组值互相匹配时，将设置警报内部标志（AF），并且如果设置了警报中断使能（AIE）位，那么将在 INT 引脚上生成一个中断。

有四个警报匹配字段——日期、小时、分钟和秒。上述每个字段有一个匹配位，用于确定字段是否被用于警报匹配逻辑。将匹配位设置为 ‘0’ 表示相应的字段用于进行匹配处理。根据匹配位的不同，警报可以明确到每个月发生一次或频繁到每分钟发生一次。如果没有选择任何匹配位（所有位都为 1），则表示不需要匹配，因此禁用警报。选择所有的匹配位（所有位都为 0）会引起精确的时间和日期匹配。

可通过下面的两种方法来检测警报事件：读取 AF 标志或监控 INT 引脚。位于标志寄存器 0x00 中的 AF 标志表示发生了日期或时间匹配。当发生匹配时，AF 位设置为 ‘1’。读取标志寄存器会清除警报标志位（和所有其他位）。硬件中断引脚也可能被用于检测警报事件。

要想设置、清除或使能警报，需要将 W 位（在标志寄存器 0x00 中）设置为 ‘1’，从而使能对警报寄存器的写入。写入警报值后，将 W 位清除为 ‘0’，从而使更改生效。

注意：器件要求将对秒的警报匹配位（即为警报秒寄存器 0x02 中的 ‘D7’ 位）设置为 ‘0’，从而正确操作警报标志和中断。

看门狗定时器

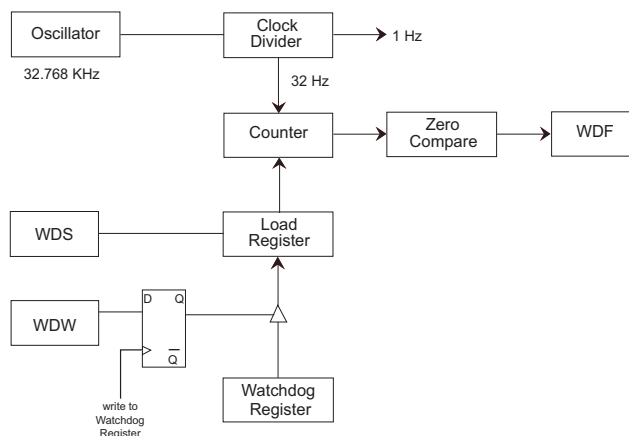
看门狗定时器是一个自由运行的递减计数器，它使用从晶体振荡器派生的 32 Hz 时钟脉冲（31.25 ms）。必须运行振荡器才能使看门狗正常运行。看门狗定时器将从看门狗定时器寄存器中加载的值开始递减计数。

定时器由一个可加载的寄存器和一个自由运行的计数器组成。在加电时，寄存器 0x07 中的看门狗超时值被加载到计数器加载寄存器中。计数操作从加电时开始，并在看门狗探针（WDS）位被设置为 ‘1’ 后随时都能从可加载值重新开始计数。计数器与终止值 ‘0’ 进行比较。如果计数器达到此值，则产生内部标志和可选中断输出。可以通过在计数器到达 ‘0’ 之前将 WDS 位设置为 ‘1’ 来阻止超时中断。这样会导致计数器重新加载看门狗超时值并重启。只要用户在计数器到达最终值之前设置了 WDS 位，就不会产生中断和 WDT 标志。

通过将看门狗写入位设置为 ‘0’，可以写入新的超时值。WDW 位为 ‘0’ 时，将使能看门狗超时值位 D5-D0 的写功能，以修改超时值。当 WDW 为 ‘1’ 时，会忽略对 D5-D0 位进行的写操作。采用 WDW 功能，用户能在无需考虑看门狗定时器值被修改的情况下设置 WDS 位。看门狗定时器逻辑图如第 47 页上的图 100 所示。注意：将看门狗超时值设置为 ‘0’ 会禁用看门狗功能。

看门狗定时器的输出是标志位 WDF（如果看门狗允许超时，将设置该标志位）。如果中断寄存器中的看门狗中断使能（WIE）位被设置，那么看门狗超时将生成 INT 引脚硬件中断。当用户读取标志寄存器时，标志和硬件中断都会被清除。

图 100. 看门狗定时器框图



可编程方波生成器

方波生成器块使用晶振输出在器件的 INT 引脚上生成所需的频率。输出频率可编程为以下频率之一：

- 1 Hz
- 512 Hz
- 4096 Hz
- 32768 Hz

当器件使用备用电源运行时，不生成方波输出。

电源监视器

CY14V101PS提供了具有断电中断功能的电源管理方案。它同样也控制着内部开关,以便为时钟提供备用电源并保护存储器在低 V_{CC} 条件下不被访问。功耗监控器以内部带隙参考电路为基础,该电路将 V_{CC} 电压与 V_{SWITCH} 阈值进行比较。

如中所述第 6 页上的自动存储操作，达到 V_{SWITCH} 后，由于 V_{CC} 由于断电而发生衰减，将启动从 SRAM 到非易失性元素的数据存储操作，以保存最后的 SRAM 数据状态。电源也会从 V_{CC} 切换到备用电源（电池）来运行 RTC 振荡器。

当使用备用电源运行时，对nvSRAM执行的读取和写入操作都被禁止，并且RTC功能对于用户不可用。RTC时钟继续在后台运行。V_{CC}存储到器件后，用户可以使用更新后的RTC计时寄存器（请参见第59页上的自动存储或加电回读）。

备用电源监控器

提供一个可检测备用电源（备用电池）故障的功耗监控系统。如果发生备用电源故障，备用电源故障标志（BPF）在下次加电时发出。如果备用电压低于 $V_{BAKFAIL}$ ，将设置 BPF 标志。即使 RTC 在备用模式下运行，仍会监控备用电源。备用模式运行中检测到的低电压通过 BPF 标志进行标记。BPF 只能将数据保持到出现定义的备用电压（ V_{DR} ）低电压电平。

中断

CY14X101Q 有一个标志寄存器、中断寄存器和将中断信号传送到微控制器的中断逻辑。有三个潜在中断源：看门狗定时器、功耗监控器和警报定时器。通过在中断寄存器（0x06）中准确设置每个中断源，可以单独使能它们来驱动 INT 引脚。此外，在标志寄存器（0x00）中，每种断源有相应的标志位，主机处理器使用这些标志位来确定中断原因。当中断发生时，INT 引脚驱动器有两个能指定其行为的位。

只有当三个中断源中的一个发出中断标志并且各自位于中断寄存器中的中断使能位被使能（设置为‘1’）时，才会发出中断。中断源处于活动状态后，两个可编程位（即 H/L 和 P/L）会决定 INT 引脚上输出引脚驱动器的行为。这两位位于中断寄存器中，可用于驱动 INT 引脚上的电平或脉冲模式输出。在脉冲模式下，脉冲宽度内部固定为大约 200 ms。该模式用于在复位主机微控制器。在电平模式下，引脚进入其活动极性，直到用户读取标志寄存器为止。此模式用作主机微控制器的中断。[中断寄存器](#)一节对控制位进行了总结。

系统仅在常规电源运行时才会生成中断，另外系统以备用电源模式运行时并不会触发中断。

注意：只有完成加电回读序列后，器件才会生成有效中断。加电后，必须在 t_{EA} 的时间内忽略 INT 引脚上的所有事件。

中断寄存器

看门狗中断使能 (WIE)：被设置为 ‘1’ 时，如果发生看门狗超时，则看门狗定时器将驱动 INT 引脚和一个内部标志。当 WIE 被设置为 ‘0’ 时，看门狗定时器只会影响标志寄存器中的 WDF 标志。

警报中断使能 (AIE): 当设置为 ‘1’ 时, 警报匹配将驱动 INT 引脚和一个内部标志。当 AIE 设置为 ‘0’ 时, 警报匹配只影响标志寄存器中的 AF 标志。

电源掉电中断使能 (PFE)：当设置为 ‘1’ 时，断电监控器将驱动引脚和一个内部标志。当 PFE 被设置为 ‘0’ 时，断电监控器只会影响标志寄存器中的 PF 标志。

方波使能 (SQWE)：当设置为‘1’，INT 引脚生成具有可编程频率的方波。频率由中断寄存器的 SQ1 和 SQ0 位决定。该位是非易失性的，在电源循环后保持不变。SQWE 位覆盖了所有其他中断。然而，CAL 位优先于方波发生器。该位在出厂时被默认设置为‘0’。

高/低 (H/L): 当设置为 ‘1’ 时, INT 引脚为高电平有效且驱动器模式为推挽式。仅在 V_{CC} 高于 V_{SWITCH} 时, INT 引脚才被驱动为高电平。当 H/L 被设置为 ‘0’ 时, INT 引脚为低电平有效, 并且驱动器模式为开漏模式。必须通过一个 $10\text{ k}\Omega$ 电阻将 INT 引脚上拉至 V_{CC} , 同时使用低电平有效模式的中断。

脉冲 / 电平 (P/L): 当设置为 ‘1’ 且发生中断时, 会将 INT 引脚驱动约 200 ms 的时间。当 P/L 设置为 ‘0’ 时, INT 引脚被驱动至高电平或低电平 (由 H/L 决定), 直到标志寄存器被读取为止。

SQ1 和 SQ0。当 SQWE 位被设置为 ‘1’ 时，这些位一起被用于修正 INT 引脚输出的中方波频率。这些位是非易失性的，在电源周期后保持不变。下表显示了输出频率的决定因素。

表 13. SQW 输出选择

SQ1	SQ0	频率	注释
0	0	1 Hz	1 Hz 信号
0	1	512 Hz	对校准操作很有帮助
1	0	4096 Hz	4 kHz 时钟输出
1	1	32768 Hz	振荡器输出频率

当使能的中断源激活 INT 引脚时，外部主机将读取标志寄存器来确定原因。请记住，当寄存器被读取时，所有标志都被清除。如果 INT 引脚被编程为电平模式，则条件将清除并且 INT 引脚会返回到其非活动状态。如果引脚被编程为脉冲模式，那么读取标志也能清除引脚中的标志。如果读取了标志寄存器，脉冲模式便不能完成指定的持续时间。如果 INT 引脚用于进行主机的复位，则在复位期间，标志寄存器不被读取。

该汇总表显示了 INT 引脚的状态。

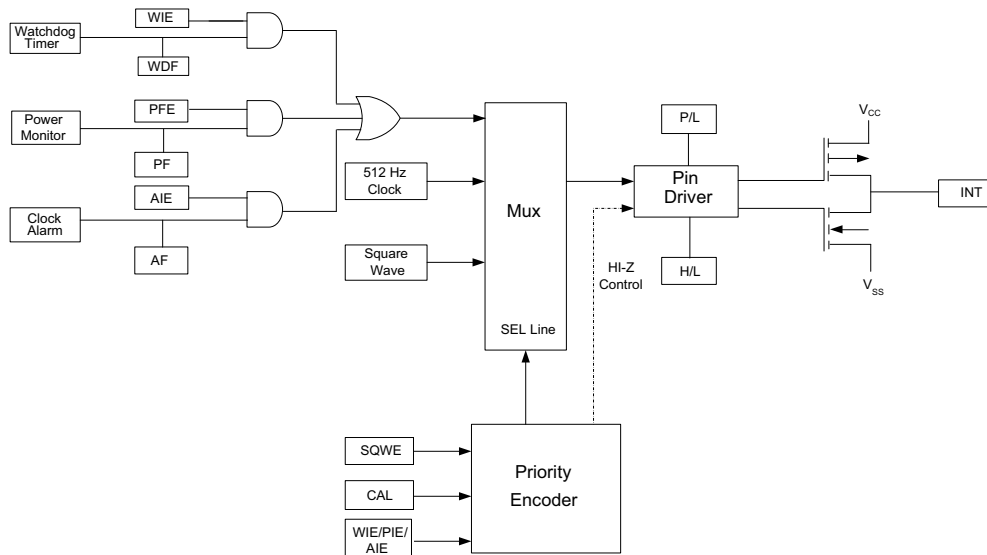
表 14. INT 引脚状态

CAL	SQWE	WIE/AIE/PFE	INT 引脚输出
1	X	X	512 Hz
0	1	X	方波输出
0	0	1	警报
0	0	0	高阻

标志寄存器

标志寄存器具有三个可用于生成中断的标志位，即：WDF、AF 和 PF 位。这些标志分别由看门狗超时、警报匹配或电源掉电监控器设置。处理器可通过轮询该寄存器或使能中断来确认何时设置了标志。寄存器被读取后，这些标志会自动复位。在加电时，标志寄存器会自动加载数值 0x00（OSCF 位除外）。请参见第 45 页上的停止和启动振荡器。

图 101. 中断框图

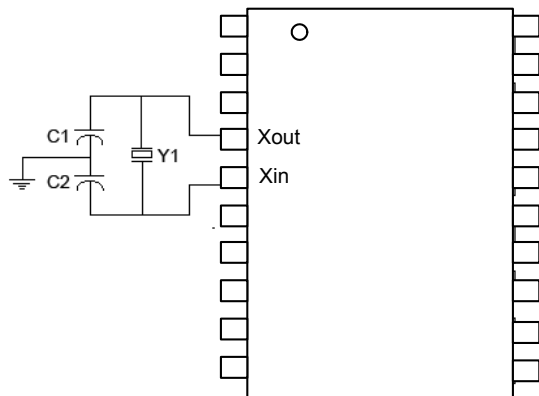


WDF — 看门狗定时器标志
WIE — 看门狗中断使能
PF — 断电标志
PFE — 断电使能
AF — 警报标志
AIE — 警报中断使能
P/L — 脉冲电平
H/L — 高电平 / 低电平
SQWE — 方波使能

RTC 外部组件

RTC 要求将外部 32.768kHz 晶体和 C_1 、 C_2 负载电容连接起来，如图 102 所示。该图显示了推荐的 RTC 外部组件的值。负载电容 C_1 和 C_2 包含印刷电路板（PCB）的寄生电容。PCB 寄生电容包括由晶体焊盘 / 引脚的地层、 X_{in}/X_{out} 焊盘以及与晶体和器件引脚相连接的焊盘和铜线导致的电容。

图 102. RTC 组件的建议配置^[1]



推荐值

$Y1 = 32.768 \text{ kHz}$

$C_1 = 20 \text{ pF}$

$C_2 = 40 \text{ pF}$

注意 所推荐的 C_1 和 C_2 值包括了电路板走线电容。

注释:

1. 有关 nvSRAM RTC 的设计指南和最佳实践，请参见应用笔记 AN61546。

RTC 的 PCB 设计注意事项

RTC 晶振是一个低电流电路，其晶体引脚上的节点处于高阻抗状态。由于 RTC 的较低计时电流，所以晶振连接对电路板上的噪声非常敏感。因此，必须将 RTC 电路与电路板上的其他信号隔离开。

此外，最小化 PCB 上的杂散电容也非常重要。杂散电容被添加到晶体的总负载电容内，这样会使振荡器频率出现误差。为获取 RTC 的最佳性能，要求实现适当的旁路并谨慎设计布局。

布局要求

布线 RTC 电路时，电路板布局必须符合（但不限于）下面的指南。按照这些指导，您能够得到 RTC 设计的最佳性能。

- 进行放置时，尽可能将晶体放置在接近 X_{in} 和 X_{out} 引脚的位置。使晶振和 RTC 之间的走线长度相等，并尽量缩短它们的长度，以便可以通过缩短天线来降低噪声耦合的可能性。

- X_{in} 和 X_{out} 走线宽度必须小于 8 mil。走线越宽，引起的走线电容也越大。这些连接焊盘和走线的宽度越大，噪声从相邻信号耦合的可能性也越大。

- 通过在晶振电路周围提供一个保护环来屏蔽 X_{in} 和 X_{out} 信号。该保护环阻止来自相邻信号的噪声耦合。

- 在 RTC 走线附近布置其他任何高速度信号时，需要特别注意。晶体与电路板上其他信号的相隔距离越长，则噪声耦合到晶体的可能性越小。在电路板上，保持 X_{in} 、 X_{out} 走线以及其他任何高速度信号之间的距离最小为 200 mil。

- 在 PCB 的同一层上，请勿在晶体组件下面布置任何信号。

- 在邻近 PCB 层上创建一个独立、实心的铜质层，该层位于晶体电路下面，其目的是阻止布置在 PCB 其他信号层上的走线的意外噪声耦合。在同一个 PCB 层上，本地层与其相邻层之间的距离最少为 40 mil。实心层只应该位于 RTC 组件附近的范围内，其外围要等于保护环的外围。图 103 显示了 RTC 电路的推荐布局。

图 103. RTC 的推荐布局

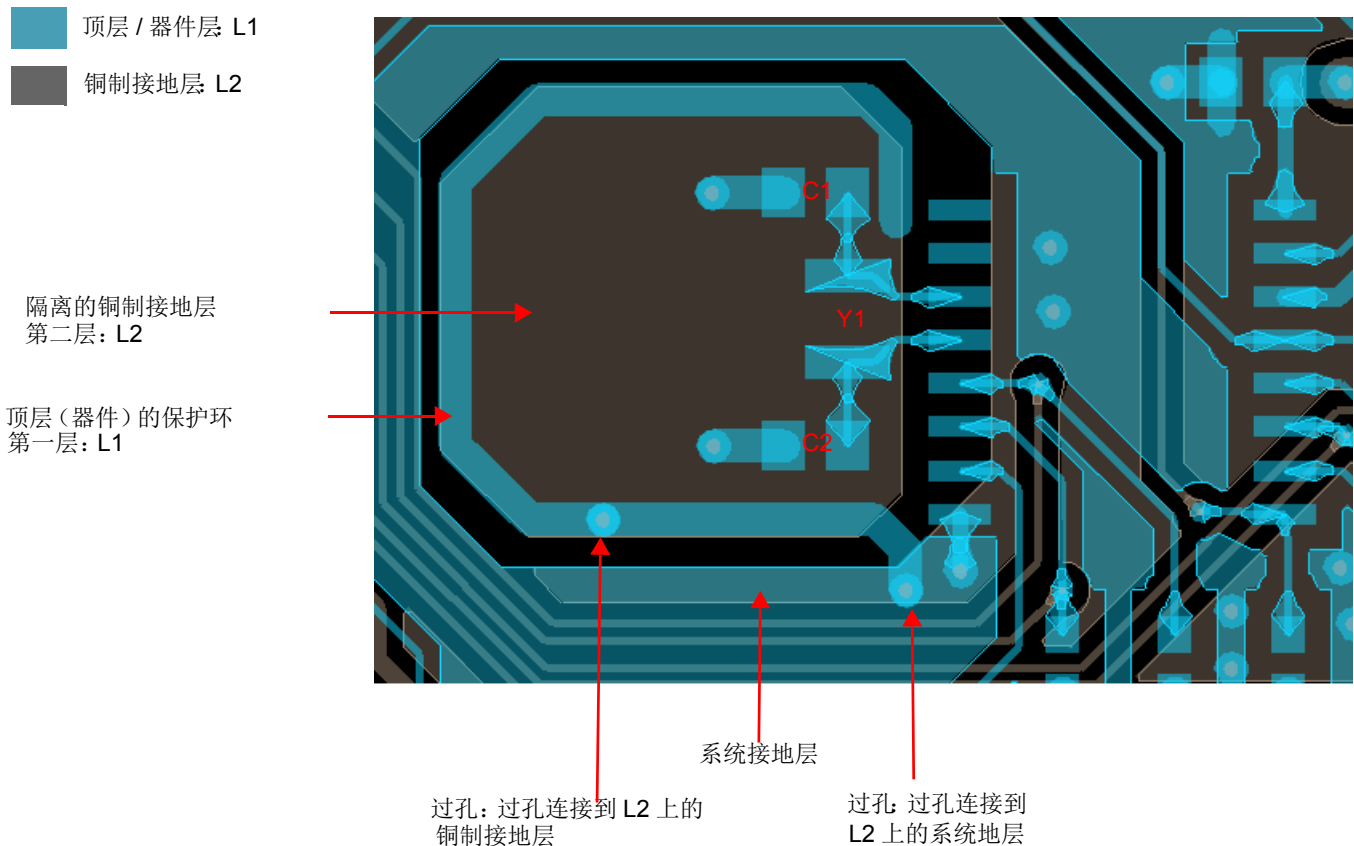


表 15. RTC 寄存器映射^[2、3]

寄存器	BCD 格式数据								功能 / 范围
	D7	D6	D5	D4	D3	D2	D1	D0	
0x0F	年 (x10)				年				年数: 00–99
0x0E	0	0	0	月 (x10)	月				月数: 01–12
0x0D	0	0	日期 (x10)		日期				某月某日: 01–31
0x0C	0	0	0	0	0	星期			某周某日: 01–07
0x0B	0	0	小时 (x10)		小时				小时: 00–23
0x0A	0	分钟 (x10)			分钟				分钟: 00–59
0x09	0	秒 (x10)			秒				秒钟: 00–59
0x08	OSCEN (0)	0	校准符号 (0)	校准 (00000)					校准值 ^[4]
0x07	WDS (0)	WDW (1)	WDT (000000)						看门狗 ^[4]
0x06	WIE (0)	AIE (0)	PFE (0)	SQWE (0)	H/L (1)	P/L (0)	SQ1 (0)	SQ0 (0)	中断 ^[4]
0x05	M (1)	0	日期警报 (x10)		日期警报				警报, 日期: 01–31
0x04	M (1)	0	小时警报 (x10)		小时警报				警报, 小时: 00–23
0x03	M (1)	分钟警报 (x10)			分钟警报				警报, 分钟: 00–59
0x02	M (1)	秒警报 (x10)			秒警报				警报, 秒钟: 00–59
0x01	世纪 (x10)				世纪				世纪: 00–99
0x00	WDF	AF	PF	OSCF ^[5]	BPF ^[5]	CAL (0)	W (0)	R (0)	标志 ^[4]

注释:

- () 指明出厂数值。
- RTC 寄存器未使用的位被保留以供将来使用, 并将该位设置为 ‘0’。
- 该值为二进制值, 并非 BCD 格式的值。
- 用户复位 OSCF 和 BPF 标志位后, 应在 t_{RTCp} 时间后更新标志寄存器。

表 16. 寄存器映射的详细信息

0x0F	计时一年							
	D7	D6	D5	D4	D3	D2	D1	D0
	年 (x10)				年			
	包含表示年的两个低位 BCD 数字。低位半字节 (四位) 包含了表示年的数值; 高位半字节 (四位) 包含的是以 10 年为单位的值。每半字节的取值范围为 0 到 9。该寄存器的范围为 0 到 99。							
0x0E	计时一月							
	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	月 (x10)	月			
	包含表示月的 BCD 数字。低位半字节 (四位) 包含了低位数字, 其取值范围为 0 到 9; 高位半字节 (一位) 包含了高位数字, 取值范围为 0 到 1。该寄存器的范围为 1 到 12。							
0x0D	计时一日期							
	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	日期 (x10)		日期			
	包含日期的 BCD 数字。低位半字节 (四位) 包含低位数字, 其取值范围为 0 到 9; 高位半字节 (两位) 包含高位数字, 其取值范围为 0 到 3。该寄存器的范围为 1 到 31。可针对闰年进行自动调整。							
0x0C	计时一星期							
	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	0	0	星期		
	低位半字节 (三位) 包含一个与星期相关的值。星期是一个循环计数器, 它从 1 计数到 7, 然后再返回为 1。用户必须为星期值提供一个它们所表达的意义, 因为星期与日期并不相关。							
0x0B	计时一小时							
	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	小时 (x10)		小时			
	包含小时 (二十四制式格式) 的 BCD 值。低位半字节 (四位) 包含低位数字, 范围为 0 到 9; 高位半字节 (两位) 包含高位数字, 范围为 0 到 2。该寄存器的范围为 0 到 23。							
0x0A	计时 一分钟							
	D7	D6	D5	D4	D3	D2	D1	D0
	0	分钟 (x10)			分钟			
	包含分钟的 BCD 值。低位半字节 (四位) 包含低位数字, 范围为 0 到 9; 高位半字节 (三位) 包含高位数字, 范围为 0 到 5。该寄存器的范围为 0 到 59。							
0x09	计时 一秒							
	D7	D6	D5	D4	D3	D2	D1	D0
	0	秒 (x10)			秒			
	包含秒的 BCD 值。低位半字节 (四位) 包含低位数字, 其范围为 0 到 9; 高位半字节 (三位) 包含高位数字, 其范围为 0 到 5。该寄存器的范围为 0 到 59。							
0x08	校准 / 控制							
	D7	D6	D5	D4	D3	D2	D1	D0
	OSCEN	0	校准符号	校准				
OSCEN	振荡器使能。被设置为 ‘1’ 时, 振荡器将停止。被设置为 ‘0’ 时, 振荡器将运行。通过禁用振荡器, 可以在存储过程中节省电能。							
校准符号	确定对时基进行哪种校准调整: 增加 (1) 还是减少 (0)。							
校准	这五个位控制时钟的校准。							
0x07	看门狗定时器							
	D7	D6	D5	D4	D3	D2	D1	D0
	WDS	WDW	WDT					

表 16. 寄存器映射的详细信息（续）

WDS	看门狗探针。将该位设置为“1”，可重新加载并重启看门狗定时器。将该位设置为“0”不起作用。看门狗定时器复位后，该位被自动清除。WDS 位是只写位。读取该位始终返回“0”。							
WDW	看门狗写使能。通过将该位设置为“1”，可禁用对看门狗超时值（D5–D0）进行的任何写入操作。这使用户可以置位看门狗探针位，而不影响超时值。当完成下一个写周期时，通过将该位设置为“0”，可以将 D5–D0 位写到看门狗寄存器内。 第 46 页上的看门狗定时器 中详细介绍了该功能。							
WDT	看门狗超时选择。可通过该寄存器中的 6 位值选择看门狗定时器的间隔。它代表一个 32 Hz 计数（31.25 ms）的倍数。超时值范围为 31.25 毫秒（设置为 1）到 2 秒（设置为 3 Fh）。将看门狗定时器寄存器设置为 0 将禁用定时器。只有在上一个周期中将 WDW 位设置为“0”时，才能对这些位进行写操作。							
0x06	中断状态 / 控制							
	D7	D6	D5	D4	D3	D2	D1	D0
	WIE	AIE	PFE	SQWE	H/L	P/L	SQ1	SQ0
WIE	看门狗中断使能。如果设置为“1”，并且发生看门狗超时，那么看门狗定时器将驱动 INT 引脚和 WDF 标志。将其设置为“0”时，看门狗定时器仅对 WDF 标志产生影响。							
AIE	警报中断使能。将该位设置为“1”时，警报匹配将驱动 INT 引脚和 AF 标志。将其设置为“0”时，警报匹配只影响 AF 标志。							
PFE	断电使能。将该位设置为“1”时，警报匹配将驱动 INT 引脚和 PF 标志。将其设置为“0”时，断电监控器只影响 PF 标志。							
SQWE	方波使能。设置为“1”时，在 INT 引脚上按照 SQ1 和 SQ0 位所编程的频率驱动方波。方波输出优先于中断逻辑。如果将 SQWE 位设置为“1”，则使能的中断源处于活动状态时，仅产生相应的标志，INT 引脚继续驱动方波。							
H/L	高电平 / 低电平。将该值设置为“1”时，将 INT 引脚驱动为高电平有效。当将该位设置为“0”时，INT 引脚为开漏、低电平有效。							
P/L	脉冲 / 电平。当将该位设置为“1”时，INT 引脚会由一个中断源驱动为有效状态（由 H/L 决定）约 200 ms。当设置为“0”时，INT 引脚被驱动到有效电平状态（如 H/L 设置），直到标志寄存器被读取为止。							
SQ1、SQ0	SQ1、SQ0。当 SQWE 位被设置为“1”时，这些位用于决定 INT 引脚上输出方波的频率。以下为每个（SQ1、SQ0）组合的频率输出： (0, 0) — 1 Hz (0, 1) — 512 Hz (1, 0) — 4096 Hz (1, 1) — 32768 Hz							
0x05	警报一日							
	D7	D6	D5	D4	D3	D2	D1	D0
	M	0	警报日期（x10）			警报日期		
	包含警报日期值和用于选择或取消选择日期值的掩码位。							
M	匹配。当该位设置为“0”时，在警报匹配中使用日期值。将该位设置为“1”会导致匹配电路忽略日期值。							
0x04	警报 — 小时							
	D7	D6	D5	D4	D3	D2	D1	D0
	M	0	警报小时（x10）			警报小时		
	包含警报小时值和用于选择或取消选择小时值的掩码位。							
M	匹配。当该位被设置为“0”时，在警报匹配中使用小时值。将该位设置为“1”时，匹配电路将忽略小时值。							
0x03	警报 — 分钟							
	D7	D6	D5	D4	D3	D2	D1	D0
	M	警报分钟（x10）			警报分钟			
	包含警报分钟值和用于选择或取消选择分钟值的掩码位。							
M	匹配。将该位设置为“0”时，在警报匹配中使用分钟值。将该位设置为“1”会导致匹配电路忽略分钟值。							
0x02	警报 — 秒							
	D7	D6	D5	D4	D3	D2	D1	D0
	M	警报秒（x10）			警报秒			
	包含警报秒值和用于选择或取消选择秒值的掩码位。							

表 16. 寄存器映射的详细信息（续）

M	匹配。将该位设置为‘0’时，在警报匹配中使用秒值。将该位设置为‘1’会导致匹配电路忽略秒值。							
0x01	计时 — 世纪							
	D7	D6	D5	D4	D3	D2	D1	D0
	世纪（x10）				世纪			
	包含表示世纪的 BCD 值。低位半字节（四位）包含低位数字，范围为 0 到 9；高位半字节（四位）包含高位数字，范围为 0 到 9。该寄存器的范围为 0 到 99。							
0x00	标志							
	D7	D6	D5	D4	D3	D2	D1	D0
	WDF	AF	PF	OSCF	BPF	CAL	W	R
WDF	看门狗定时器标志。当在没有被用户复位情况下允许看门狗定时器达到 0 时，将该只读位设置为‘1’。当标志寄存器被读或加电时，该位被清除为‘0’。							
AF	警报标志。当时间和日期与储存在警报寄存器中的值相匹配，并且匹配位为“0”时，该只读位将被设置为‘1’。当标志寄存器被读取或加电时，会清除警报标志。							
PF	断电标志。当电源下降到低于断电阈值 V_{SWITCH} 时，该只读位被设置为‘1’。读取标志寄存器时，会清除断电标志。							
OSCF	振荡器失败标志。如果振荡器被使能，并且在头 5 毫秒操作时间内未运行，则该位将在加电时被设置为‘1’。这表示 RTC 备用电源中断而且时钟值不再有效。该位在电源周期内保持不变，并永远不会被芯片内部清除。用户必须检查此条件并写入‘0’以清除该标志。当用户复位 OSCF 标志位时，在 t_{RTCp} 时间后该位将被更新。							
BPF	备用电源中断标志。如果备用电源（电池）中断，则加电时将该位设置为‘1’。如果电压下降到其各自最低指定电压以下时，会发生备用电源中断。BPF 只能将数据保持到出现定义的备用电压（ V_{DR} ）低电压电平。用户必须复位此位以清除该标志。用户复位 BPF 标志位时，经过 t_{RTCp} 时间后将更新该位。							
CAL	校准模式。当该位设置为‘1’时，INT 引脚会输出 512 Hz 的方波。当设置为‘0’时，则 INT 引脚恢复正常操作。该位优先于 SQ0/SQ1 和其他功能。加电时，该位默认为‘0’（禁用）。							
W	写使能：将‘W’位设置为‘1’会冻结对 RTC 寄存器进行的更新。然后用户可写入 RTC 寄存器、警报寄存器、校准寄存器、中断寄存器以及标志寄存器。如果时间被更改，那么将“W”设置为“0”时，RTC 寄存器中的内容将被传输到计时计数器中。完成该传输过程会需要 t_{RTCp} 时间。加电时，该位默认为 0。							
R	读使能：将“R”位设置为‘1’会停止对用户 RTC 寄存器中时钟进行的更新，所以在读取过程中不显示时钟更新。将“R”位设置为‘0’可恢复对保持寄存器的时钟更新。设置该位时不需要将“W”位设置为“1”。加电时，该位默认为 0。							

最大额定值

超过最大额定值可能会缩短器件的使用寿命。这些用户指南未经过测试。

存储温度 -65 °C 到 +150 °C

最长存储时间

在 150°C 环境温度下 1000 小时

在 85°C 环境温度下 20 年

最高结温 150 °C

V_{CC} (相对于 V_{SS}) 的供电电压 -0.5 V 到 +4.1 V

V_{CCQ} (相对于 V_{SS}) 的供电电压 -0.5 V 到 +2.45 V

在高阻态下

输出的直流电压 -0.5 V 到 $V_{CCQ} + 0.5$ V

输入电压 -0.5 V 到 $V_{CCQ} + 0.5$ V

在接地电位的所有引脚上的

瞬变电压 (瞬变时间 < 20 ns) -2.0 V 到 $V_{CCQ} + 2.0$ V

封装功率耗散能力 ($T_A = 25$ °C)

16-SOIC 1.0 W

封装功率耗散能力

($T_A = 25$ °C) 1.0 W

表面组装铅焊温度 (3 秒) +260 °C

直流输出电流 (一次一个输出, 持续时间为 1 秒) 15 mA

静电放电电压

(根据 MIL-STD-883, 方法 3015) > 2001 V

栓锁电流 > 140 mA

工作范围

范围	环境温度	V_{CC}	V_{CCQ}
工业级	-40°C ~ +85°C	2.7 V ~ 3.6 V	1.71 V ~ 2.0 V

直流规范

参数	说明	测试条件	最小值	典型值 ^[6]	最大值	单位
V_{CC}	电源 — 内核电压	—	2.70	3.00	3.60	V
V_{CCQ}	电源 — I/O 电压	—	1.71	1.80	2.00	V
I_{CC1}	V_{CC} 的平均读 / 写电流 (所有输入均进行切换, 没有任何输出负载)	SPI = 1 MHz	—	—	1.00	mA
		SPI = 40 MHz	—	—	3.00	mA
		QPI = 108 MHz	—	—	33.00	mA
I_{CCQ1}	V_{CCQ} 的平均电流 (所有输入均进行切换, 没有任何输出负载)	SPI = 1 MHz	—	—	150.00	μA
		SPI = 40 MHz	—	—	1.00	mA
		QPI = 108 MHz	—	—	5.00	mA
I_{SB1}	温度为 85 °C 条件下的待机模式电流 ($V_{CC} + V_{CCQ}$)	$\overline{CS} > (V_{CCQ} - 0.2 \text{ V})$ 。 非易失性循环完成后的待机电流强度 (\overline{CS} 为高电平, 其他 I/O 都没有限制, $f_{SCK} \leq 108 \text{ MHz}$)。	—	—	1.8	mA
I_{SB2}	温度为 85 °C 条件下的待机模式电流 ($V_{CC} + V_{CCQ}$)	$\overline{CS} > (V_{CCQ} - 0.2 \text{ V})$ 。 非易失性循环完成后的待机电流强度 (所有 I/O 均为静态, $f_{SCK} = 0 \text{ MHz}$)。	—	—	380.00	mA
I_{CC2}	存储期间 V_{CC} 平均电流	—	—	—	6.00	mA
I_{CC4}	AUTOSTORE (自动存储) 过程中的 V_{CAP} 平均电流	—	—	—	6.00	mA
I_{SLEEP}	温度为 85 °C 条件下的睡眠模式电流 ($V_{CC} + V_{CCQ}$)	$\overline{CS} > (V_{CCQ} - 0.2 \text{ V})$ 。 非易失性循环完成后的睡眠电流强度。所有 I/O 均为静态, $f_{SCK} = 0 \text{ MHz}$	—	—	380	μA
I_{ZZ}	温度为 85 °C 条件下的休眠模式电流 ($V_{CC} + V_{CCQ}$)	$\overline{CS} > (V_{CCQ} - 0.2 \text{ V})$ 。HIBEN 指令得到寄存后 t_{HIBEN} 时间所有输入均为静态且配置为 CMOS 逻辑电平。	—	—	8.00	μA

注释:

6. 各典型值均是在温度为 25 °C, $V_{CC} = V_{CC}(\text{Typ})$, 且 $V_{CCQ} = V_{CCQ}(\text{Typ})$ 的条件下测量得到的。并非 100% 经过了测试。

参数	说明	测试条件	最小值	典型值 ^[6]	最大值	单位
I_{IX}	输入漏电流 (HSB 除外)	$V_{CCQ} = \text{最大值}, V_{SS} < V_{IN} < V_{CCQ}$ —	-1.00	—	1.00	μA
	输入漏电流 (用于 HSB)		-100.00	—	1.00	μA
	输入漏电流 (用于 SPI/DPI 模式下的 WP)		-2	—	1	μA
I_{OZ}	断开状态时的输出漏电流	$V_{CCQ} = \text{最大值}, V_{SS} < V_{IN} < V_{CCQ}$	-1.00	—	1.00	μA
V_{IH}	输入高电平电压	—	$0.70 * V_{CCQ}$	—	$V_{CCQ} + 0.30$	V
V_{IL}	输入低电平电压	—	-0.30	—	$-0.30 * V_{CCQ}$	V
V_{OH}	$I_{OH} = -2 \text{ mA}$ 时的高电平电压输出	$I_{OH} = -2 \text{ mA}$	$V_{CCQ} - 0.45$	—	—	V
V_{OL}	$I_{OL} = 2 \text{ mA}$ 时的低电平电压输出	$I_{OL} = 2 \text{ mA}$	—	—	0.45	V
V_{CAP} ^[7]	存储电容	在 V_{CAP} 引脚和 V_{SS} 之间	61.00	68.00	120.00	μF
V_{VCAP} ^[7]	在 V_{CAP} 引脚上驱动的最大电压	—	—	—	V_{CC}	V

数据保持时间与耐久性

参数	说明	最小值	单位
DATA_R	温度为 85 °C 条件下的数据保留时间	20	年
NV_C	执行非易失性存储操作的次数	1,000	K

电容

参数 ^[8]	说明	测试条件	最大值	单位
C_{IN}	输入电容	$T_A = 25 \text{ }^\circ\text{C}, f = 1 \text{ MHz}, V_{CC} = V_{CC(\text{typ})}, V_{CCQ} = V_{CCQ(\text{typ})}$	6.00	pF
C_{SCK}	时钟输入电容			
C_{OUT}	输出引脚电容			

热阻

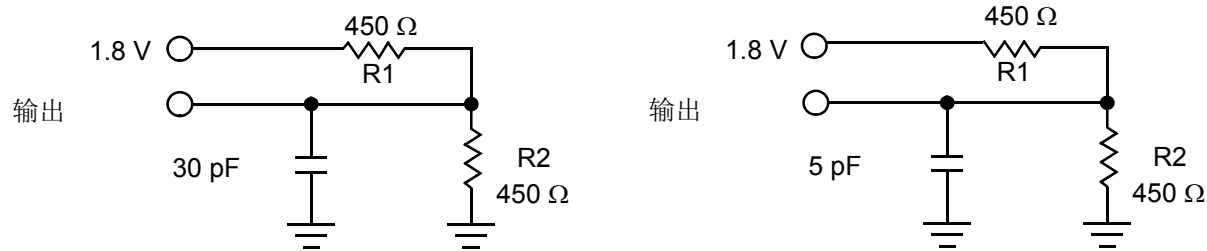
参数 ^[8]	说明	测试条件	16-SOIC	单位
Θ_{JA}	热阻 (结至环境)	根据 EIA/JESD51 的要求, 测试条件遵循测试热阻的标准测试方法和过程。	61.21	$^\circ\text{C/W}$
Θ_{JC}	热阻 (结至外壳)		26.20	

注释:

- V_{CAP} 的最小值可保证提供足够的电荷用于成功完成自动存储操作。 V_{CAP} 的最大值可确保在加电回读周期内电容电压 V_{CAP} 可被充电到最小值。这样, 在紧急断电时, 仍能够顺利完成自动存储操作。因此, 建议所使用的电容值属于指定的最小极限值到最大极限值范围内。请参考应用笔记 [AN43593](#), 了解有关 V_{CAP} 选项的详细信息。
- 这些参数由设计保证, 并未经过测试。

交流测试负载和波形

图 104. 交流测试负载和波形



交流测试条件

说明	CY14V101PS
输入脉冲电平	0 V 到 1.8 V
输出上升和下降时间（10%–90%）	≤ 1.8 ns
输入和输出时序参考电平	0.9 V

RTC 特性

参数	说明	最小值	典型值 ^[9]	最大值	单位
V _{RTCbat}	RTC 电池引脚电压	1.80	3.00	3.60	V
I _{BAK} ^[10]	RTC 备用电流 (请参见图 102, 了解推荐的 RTC 外部组件)	–	0.60	1.00	μA
t _{OCS}	RTC 振荡器启动时间	–	1.00	2.00	s
V _{BAKFAIL}	备用电压故障阈值	1.80	–	2.50	V
t _{RTCP}	将 ‘W’ 位置为 ‘0’ 后的 RTC 处理时间	–	–	350.00	μs

注释:

9. 各典型数值均是在温度为 25 °C, V_{CC} = V_{CC(Typ)} 的条件下测量得到的。并非 100% 经过了测试。

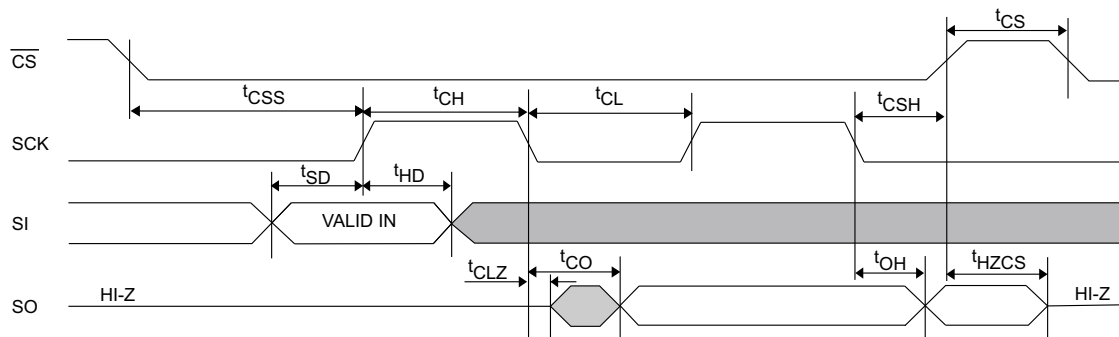
10. 当 V_{CC} < V_{SWITCH} 时, 由 V_{RTCbat} 驱动电流。

交流开关特性

参数 ^[11]	说明	最小值	最大值	单位
f_{SCK}	时钟频率 (QPI)	–	108.00	MHz
t_{CL}	时钟脉冲宽度低电平	$0.45 * 1/f_{SCK}$	–	ns
t_{CH}	时钟脉冲宽度高电平	$0.45 * 1/f_{SCK}$	–	ns
t_{CS}	\overline{CS} 为高电平的时间			
	结束读取操作所需的时间	10.00	–	ns
	结束写入操作所需的时间	10.00	–	ns
t_{CSS}	\overline{CS} 建立时间	5.00	–	ns
t_{CSH}	\overline{CS} 保持时间	5.00	–	ns
t_{SD}	数据输入的建立时间	2.00	–	ns
t_{HD}	数据输入的保持时间	3.00	–	ns
t_{SW}	\overline{WP} 的建立时间	2.00	–	ns
t_{HW}	\overline{WP} 保持时间	2.00	–	ns
t_{CO}	输入有效的时间	–	7.00	ns
t_{CLZ}	从时钟为低电平到输出为低阻态的时间	0.00	–	ns
t_{OH}	输出保持时间	1.00	–	ns
t_{HZCS} ^[12]	输出被禁用的时间	–	7.00	ns

开关波形

图 105. 同步数据时序 (模式 0)



注释:

11. 测试条件采用了不超过 1.8 ns 的信号跳变时间, $V_{CCQ}/2$ 的时序参考电平, 0 至 V_{CCQ} (t_{yp}) 的输入脉冲电平以及第 57 页上的图 104 中所示的指定 I_{OL}/I_{OH} 的输出负载和负载电容。
12. 这些参数由设计保证, 并未经过测试。

自动存储或加电回读

在工作范围内

参数	说明	最小值	最大值	单位
$t_{FA}^{[13]}$	加电回读时间	—	20.00	ms
$t_{STORE}^{[14]}$	存储周期时间	—	8.00	ms
$t_{DELAY}^{[15]}$	启动存储周期需要的时间	—	25.00	ns
V_{SWITCH}	V_{CC} 的低电压触发电平	—	2.60	V
$t_{VCCRRISE}^{[16]}$	V_{CC} 上升时间	150.00	—	μs
$V_{HDIS}^{[16]}$	\overline{HSB} 输出禁用电压	—	1.90	V
$V_{IODIS}^{[17]}$	禁用 I/O 时的 V_{CCQ} 电压	—	1.50	V
$t_{LZHSB}^{[16]}$	\overline{HSB} 为高电平到 nvSRAM 有效的时间	—	5.00	μs
$t_{HHHD}^{[16]}$	\overline{HSB} 为高电平有效的时间	—	500.00	ns
t_{WAKE}	nvSRAM 从休眠模式唤醒的时间	—	20.00	ms
t_{HIBEN}	发出 HIBEN 指令后进入休眠模式的时间	—	8.00	ms
t_{SLEEP}	\overline{CS} 变为高电平后进入睡眠模式的时间	—	0.00	μs
t_{EXSLP}	\overline{CS} 变为高电平后退出睡眠模式的时间	—	0.00	μs
t_{RESET}	软件复位时间	—	500.00	μs

注释:

13. 当 V_{CC} 超过 V_{SWITCH} 时, 开始 t_{FA} 。

14. 如果执行完最后一次非易失性周期后尚未对 SRAM 进行写操作, 将不启动自动存储或硬件存储操作。

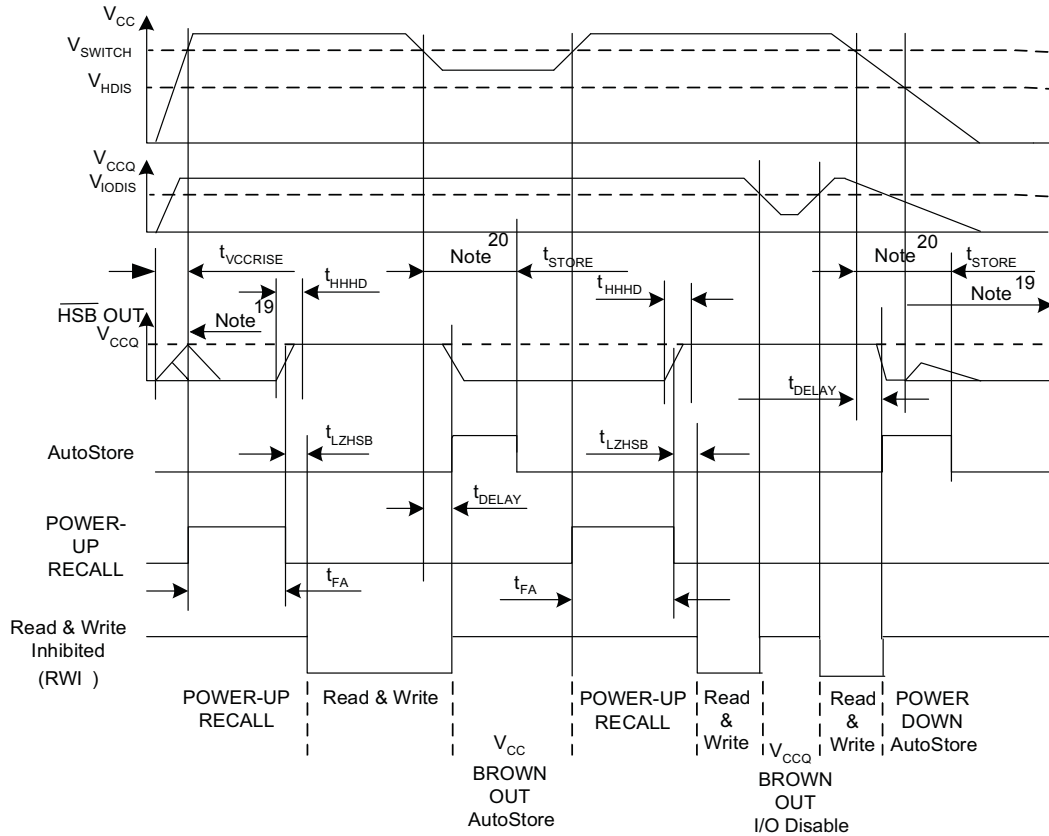
15. 在执行硬件存储、软件存储 / 回读、自动存储启用 / 禁用和自动存储时, 会在 t_{DELAY} 时间内持续启用 SRAM 操作。

16. 这些参数由设计保证, 但未进行过测试。

17. 当电压低于 V_{IODIS} 电压时, 便不可定义 \overline{HSB} 。

切换波形

图 106. 自动存储或加电回读^[18]



注释:

18. 在 V_{CC} 小于 V_{SWITCH} 的情况下, 将在存储、回读过程中忽略读写周期。
19. 在通电和断电期间, 如果通过外部电阻上拉 HSB 引脚电压, 那么 HSB 会产生瞬时脉冲。
20. 如果执行完最后一次非易失性周期后尚未对 SRAM 进行写操作, 将不启动自动存储或硬件存储操作。

软件控制的存储和回读周期

在工作范围内

参数	说明	最小值	最大值	单位
t_{RECALL}	回读持续时间	—	500	μs
$t_{\text{SS}}^{[21, 22]}$	软序列处理时间	—	500	μs

切换波形

图 107. 软件存储周期^[22]

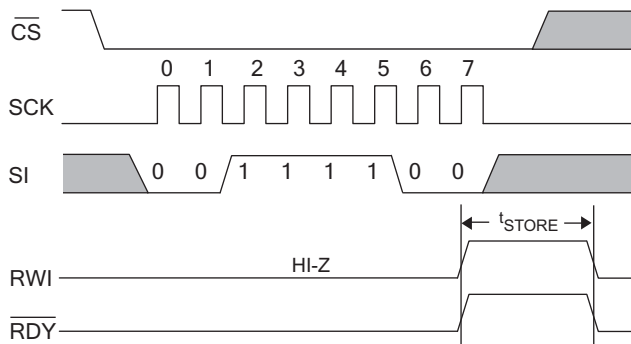


图 108. 软件回读周期^[22]

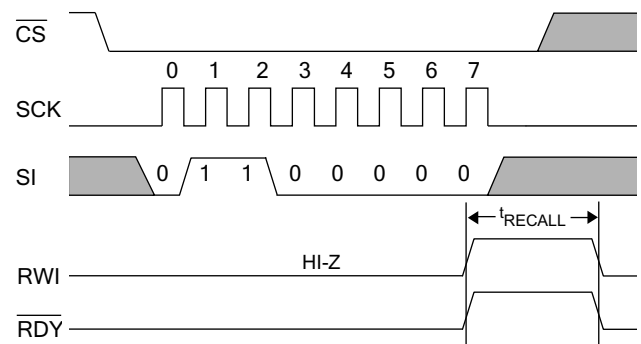


图 109. 自动存储使能周期

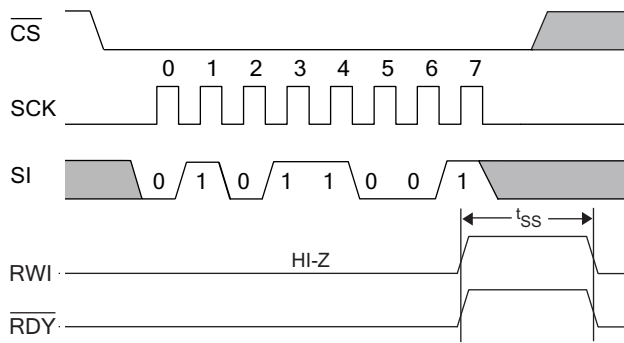
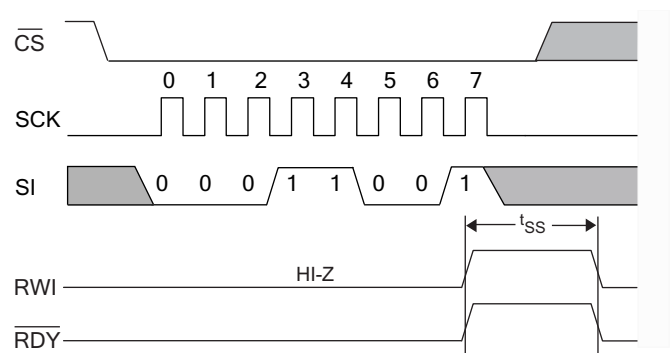


图 110. 自动存储禁用周期



注释:

21. 这是执行软序列命令所占用的时间。Vcc 电压必须保持为高电平，以保证有效地寄存命令。
 22. 存储和回读等命令会锁定 I/O，直到操作完成为止，这样可以延长该时间。请参见特定的命令。

硬件存储周期

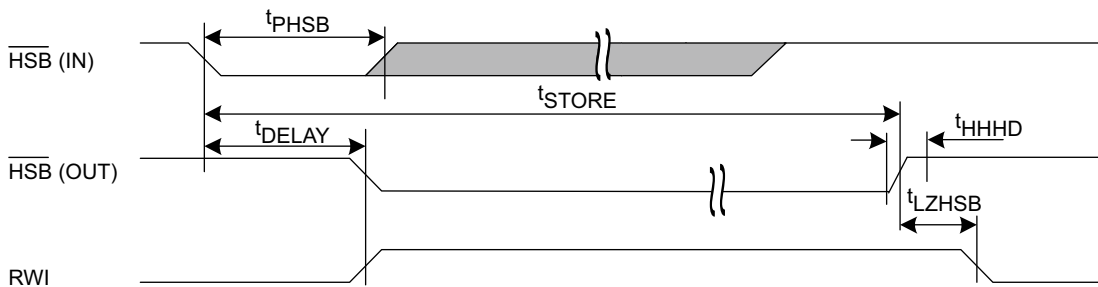
在工作范围内

参数	说明	最小值	最大值	单位
t_{PHSB}	硬件存储脉冲宽度	15	600	ns

切换波形

图 111. 硬件存储周期^[23]

Write Latch set



Write Latch not set

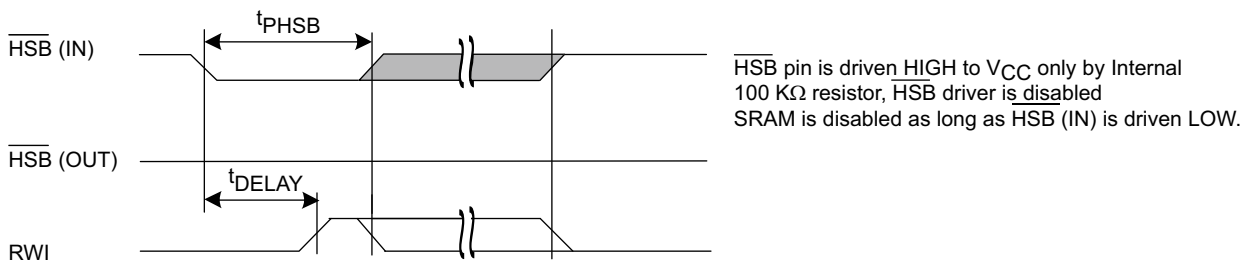
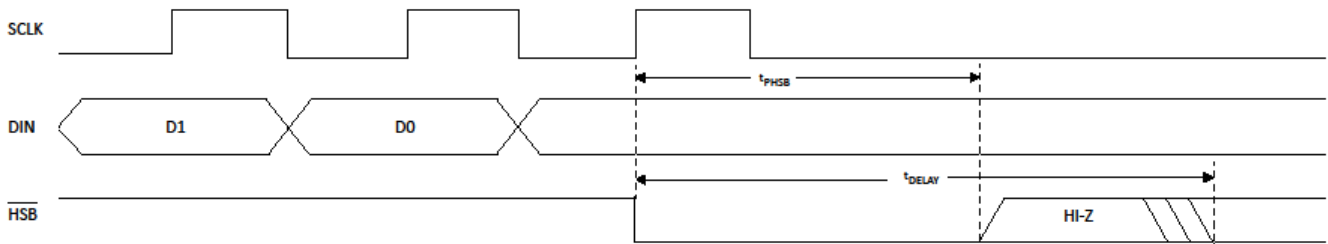


图 112. 数据有效到 HSB 的时间



注释:

23. 如果执行完最后一次非易失性周期后尚未对 SRAM 进行写操作，将不启动自动存储或硬件存储操作。

订购信息

订购代码	封装图	封装类型, 引脚分布	工作范围
CY14V101PS-SF108XI	51-85022	16-SOIC	工业级
CY14V101PS-SF108XIT			

这些都是无铅芯片。想要了解这些芯片的供应情况, 请联系赛普拉斯本地销售代表。

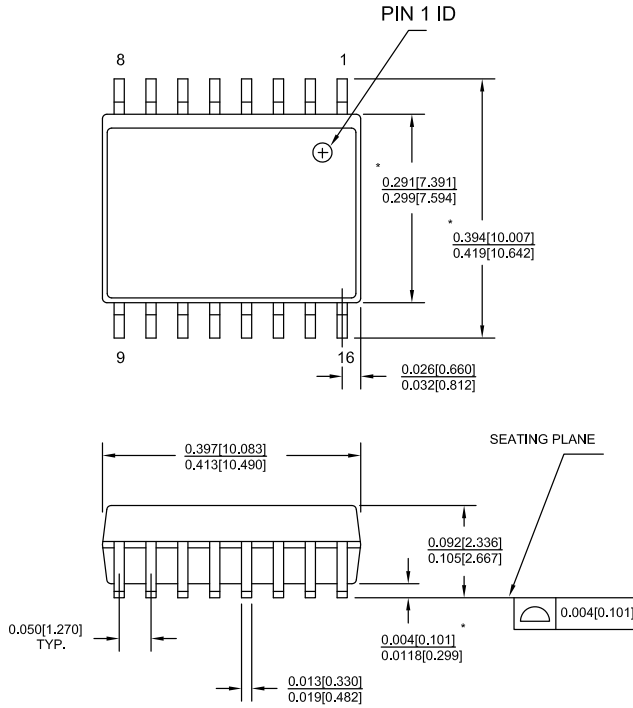
订购代码定义

CY 14 V 101 PS - SF 108 X I T

	Option: T - Tape and Reel, Blank - Std.
	Temperature: I - Industrial
	Pb-free
	Frequency: 108 - 108 MHz
	Package: SF - 16 SOIC Standard, SE - 16 SOIC Custom
	QS - Quad SPI, PS - Quad SPI with RTC
	Density: 101 - 1-Mbit
	Voltage: V - 3.0 V, 1.8 V I/O
	14 - nvSRAM
	CY - Cypress

封装图

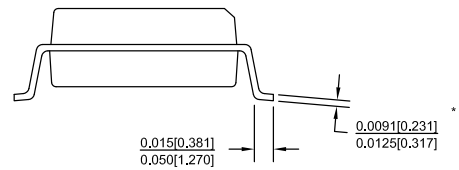
图 113. 16-SOIC (0.413 × 0.299 × 0.0932 英寸) 封装外形, 51-85022



DIMENSIONS IN INCHES[MM] MIN.
MAX.

REFERENCE JEDEC MO-119

PART #	
S16.3	STANDARD PKG.
SZ16.3	LEAD FREE PKG.



缩略语

缩略语	说明
CPHA	时钟相位
CPOL	时钟极性
CMOS	互补金属氧化物半导体
CRC	循环冗余校验
EEPROM	电可擦除可编程只读存储器
EIA	电子工业联盟
I/O	输入 / 输出
JEDEC	联合电子设备工程委员会
LSB	最低有效位
MSB	最高有效位
nvSRAM	非易失性静态随机存取存储器
RWI	禁止读和写
RoHS	有害物质限制
SNL	序列号锁定
SPI	串行外设接口
SONOS	硅 - 氧化 - 氮化 - 氧化半导体
SOIC	小外型集成电路
SRAM	静态随机存取存储器

文档常规

测量单位

符号	测量单位
°C	摄氏度
Hz	赫兹
kHz	千赫兹
kΩ	千欧
Mbit	兆位
MHz	兆赫兹
μA	微安
μF	微法
μs	微秒
mA	毫安
ms	毫秒
ns	纳秒
Ω	欧姆
%	百分比
pF	皮法
V	伏特
W	瓦特

文档修订记录

文档标题: CY14V101PS — 带有实时时钟的 1 Mbit (128K x 8) 四线 SPI nvSRAM 文档编号: 001-96331				
版本	ECN 编号	变更者	提交日期	变更说明
**	4669738	YLIU	03/16/2015	本文档版本号为 Rev**, 译自英文版 001-94176 Rev*B。
*A	4802534	YLIU	07/09/2015	本文档版本号为 Rev*A, 译自英文版 001-94176 Rev*E。
*B	5088628	YLIU	01/25/2016	本文档版本号为 Rev*B, 译自英文版 001-94176 Rev*I。

销售、解决方案和法律信息

全球销售和设计支持

赛普拉斯公司具有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。要想找到离您最近的办事处，请访问[赛普拉斯所在地](#)。

产品

汽车级产品.....cypress.com/go/automotive
时钟与缓冲区.....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
触摸感应产品.....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

© 赛普拉斯半导体公司，2014-2016。此处所包含的信息可随时更改，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，对于可能发生运转异常和故障并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。

所有源代码（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。对于可能发生运转异常和故障，并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。

产品使用可能受赛普拉斯相应软件许可协议的限制。