

HWD29S01GPBGA64M3 型

NOR FLASH

产品手册

V1.0.0

成都华微电子科技股份有限公司

2025 年 7 月 20 日

版本历史

版次	更改说明	更改人	更改日期	备注
V1.0.0	初始发行版本	杨振越	20250720	

目 次

1 概述	1
2 产品特点	1
3 功能框图	2
4 封装信息	4
4.1 HWD29S01GPBGA64M3 封装信息	4
4.2 HWD29S01GPBGA64M3 焊盘推荐	6
4.3 HWD29S01GPBGA64M3 焊接推荐	6
5 绝对最大额定值及推荐工作范围	7
5.1 绝对最大额定值	7
5.2 推荐工作范围	7
6 电特性参数	8
7 功能描述	12
7.1 概述	12
7.2 引脚控制	13
7.3 时序特性	14
7.4 交流特性	16
7.5 功能描述	32
8 使用注意事项	86
9 订货信息	87

1 概述

HWD29S01GPBGA64M3 是基于 65nm 流片工艺的并行接口 NOR FLASH 产品，容量为 1Gbit。本产品支持 15ns 快速页访问和相应 100ns 的快速随机访问。同时具有写入缓冲区，允许在一次操作中最多编程 256 字/512 字节，从而比标准编程算法缩短有效编程时间。是当今需要更高密度、更好性能和更低功耗的嵌入式应用的理想选择。

2 产品特点

- 1) 具有 VIO CMOS 3.0V 核
- 2) 65nm NOR Flash 工艺技术
- 3) 读取/编程/擦除单电源供电 (2.7-3.6V)
- 4) VIO 控制
 - 所有输入电平（地址、控制和 DQ 输入电平）和输出均由 VIO 输入上的电压决定。
 - V_{IO} 范围为 1.65V 至 V_{CC} 。
- 5) 数据位宽 x8/x16
- 6) 异步 16 字/32 字节页面读取缓冲区
- 7) 256 字/512 字节写缓冲区，减少了多字编程的总体时间
- 8) 同一字具有单个字/字节编程和多次编程两种编程方式
- 9) 扇区擦除
 - 每个扇区大小 128-kbyte
- 10) 编程与擦除操作支持挂起和恢复指令
- 11) 解锁旁路编程指令以减少编程时间
- 12) 编程加速模式 (WP#/ACC)
- 13) 通过状态寄存器，数据轮询，就绪/忙引脚判断决定设备的状态
- 14) 先进扇区保护 (ASP)
 - 每个扇区具有易失性与非易失性保护措施
- 15) 用两个可锁的区域来分隔 1024-byte OTP (一次性编程) 阵列
- 16) 支持 CFI (通用闪存接口)
- 17) 温度范围/等级
 - N1 级 (-55°C ~ 125°C)
- 18) 擦写次数: 100,000 次
- 19) 数据保持时间: 20 年
- 20) 封装形式
 - PBGA64

3 功能框图

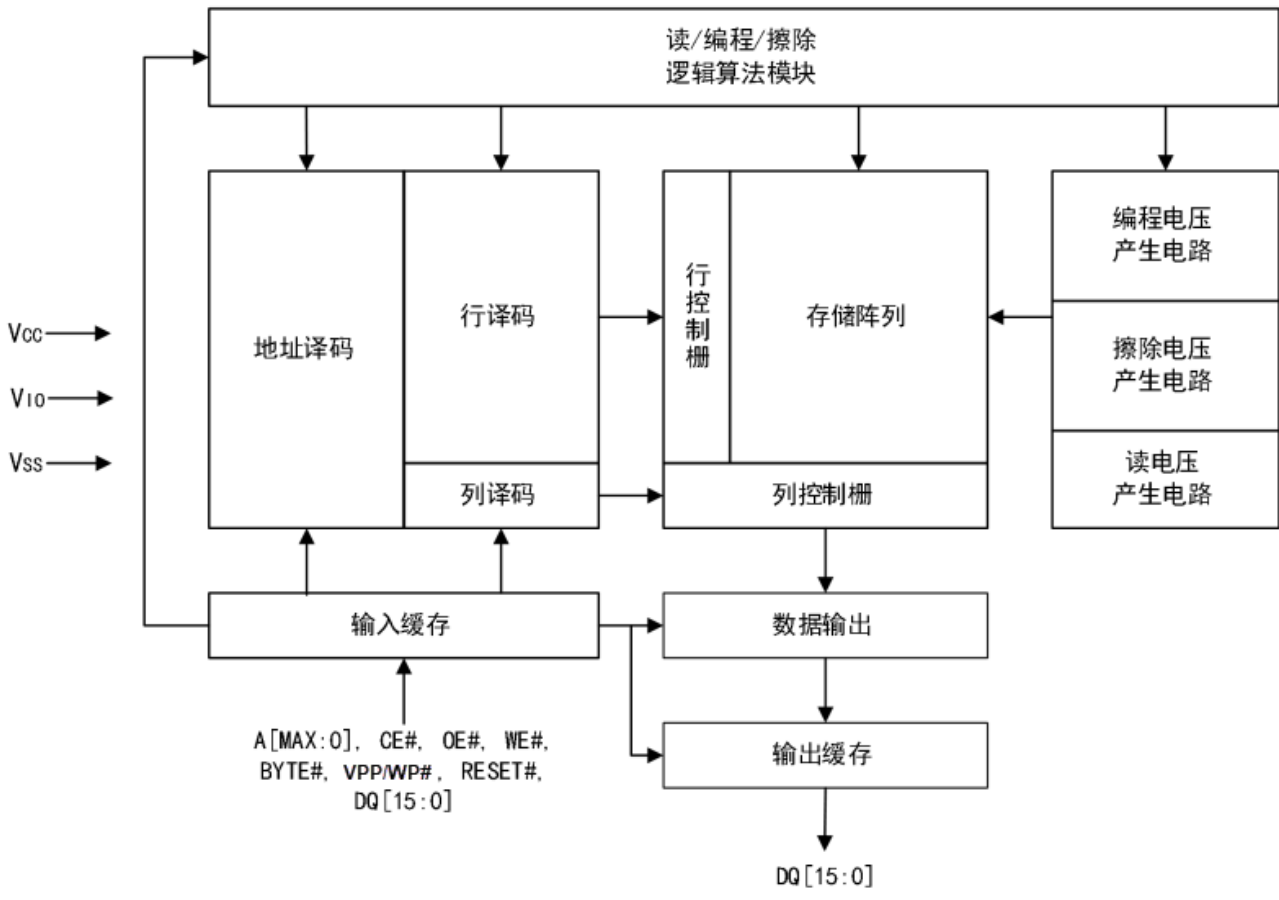


图 3-1 功能框图

HWD29S01GPBGA64M3典型应用场景如下图所示。

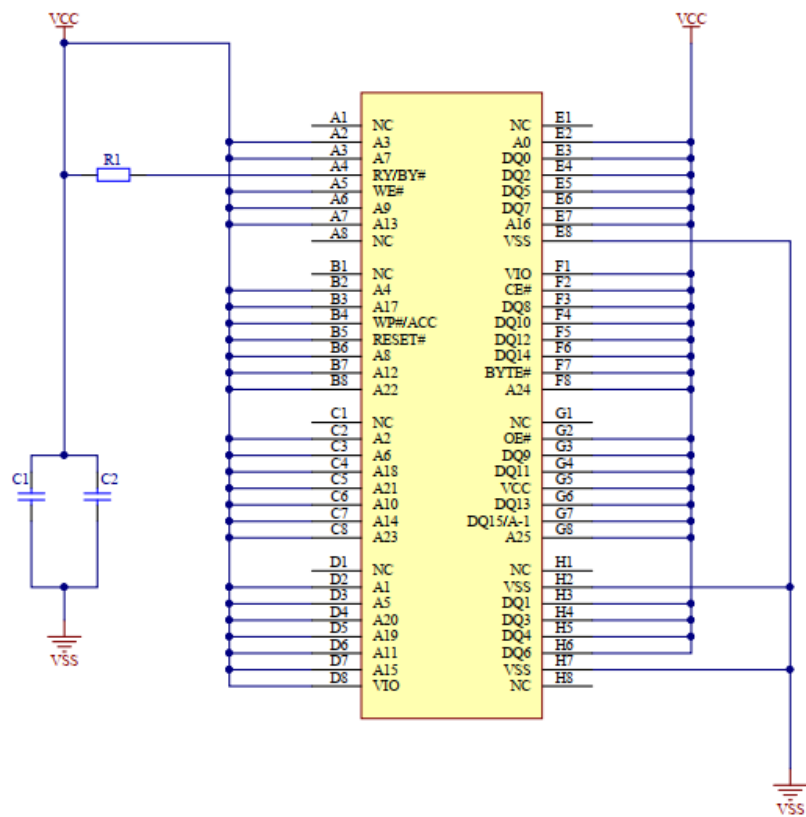


图 3-2 典型应用场景

注： $V_{CC}=3.3 \times (1 \pm 10\%)V$ ； $V_{SS}=0V$ ； $R1=4.7 \times (1 \pm 5\%)k\Omega$ ； $C1=10 \times (1 \pm 10\%) \mu F$ ； $C2=100 \times (1 \pm 10\%)nF$ ； NC 管脚悬空。

4 封装信息

4.1 HWD29S01GPBGA64M3 封装信息

HWD29S01GPBGA64M3采用PBGA64封装，外形及引脚排列图如下图所示。

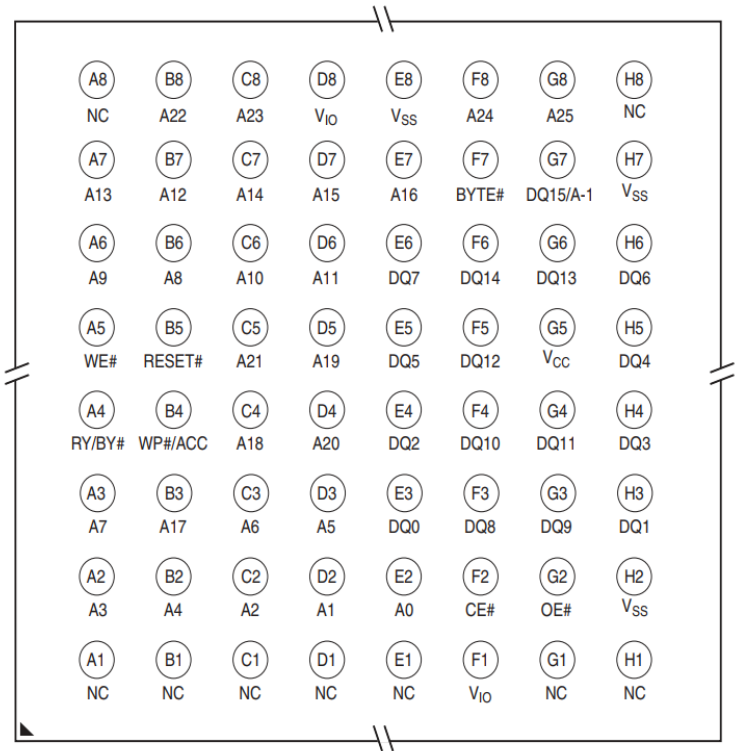


图 4-1 PBGA64 的引脚配置

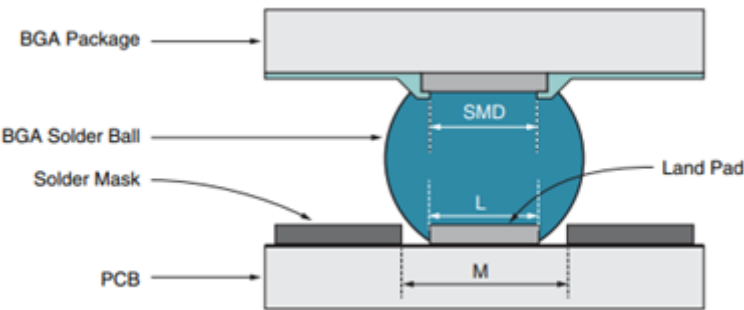
HWD29S01GPBGA64M3 引脚描述描述如下表所示。

表 4-1 信号描述

管脚名	类型	功能描述
A25-A0	输入	地址线 A_{MAX} -A0
DQ14 - DQ8	输入/输出	数据 I/O: 在读取操作期间, 当 BYTE# 为高时, 输出存储在选定地址的数据。当 BYTE# 为低时, 这些引脚为高阻 Z 且不使用。在写操作期间, 这些位为不使用。读取数据轮询寄存器时, 应忽略这些位。
DQ7-DQ0	输入/输出	数据 I/O: 在读取操作期间, 输出存储在选定地址的数据。在写入操作期间, 表示发送到命令界面的命令。
DQ15/A-1	输入/输出	DQ15: 数据输入/输出 (x16) A-1: LSB 地址输入 (x8)
CE#	输入	芯片使能
OE#	输入	输出使能
WE#	输入	写使能
VCC	电源	设备电源 (核心电源)
VIO	电源	多功能 I/O 电源输入
VSS	电源	地
NC	不连接	内部未连接。此引脚或焊球位置可能用于 PCB 布线通道的一部分
RY/BY#	输出	就绪/忙碌。指示嵌入式算法是正在进行中还是已完成。 RY/BY#=V _{IL} : 设备忙碌, 正在进行擦除或编程。设备主动参与嵌入算法, 例如擦除或编程。 RY/BY#=高阻: 设备就绪。设备准备好读取或写入新命令-需要外部上拉电阻来检测高 Z 状态。多个设备可以将它们的 RY/BY# 输出绑定在一起, 以检测所有设备何时准备好了。
BYTE#	输入	选择数据总线宽度。 BYTE#=V _{IL} : 器件处于 x8 模式, 数据 I/O 引脚 DQ0-DQ7 有效, DQ15/A-1 成为 LSB 地址输入。 BYTE#=V _{IH} : 器件处于 x16 模式, 数据 I/O 引脚 DQ0-DQ15 有效。 在字节配置下, 在任何写操作期间不应切换 BYTE#。注意: 此引脚不能浮动。
RESET#	输入	硬件复位。低=设备复位
WP#/ACC	输入	写保护。在 V _{IL} , 禁用设备最低或最高地址 64-k word (128-kB) 扇区中的编程和擦除功能。在 V _{IH} , 该扇区不受保护。WP# 有内拉; 当未连接的 WP# 为 V _{IH} 时。在程序加速模式下, ACC 引脚强制为 V _{IH} 。

4.2 HWD29S01GPBGA64M3 焊盘推荐

HWD29S01GPBGA64M3焊盘推荐如下图所示，用户可根据焊接要求进行调整。



mm			
SMD	L	M	pitch
0.53	0.53	0.63	1.00

图 4-2 推荐焊盘

4.3 HWD29S01GPBGA64M3 焊接推荐

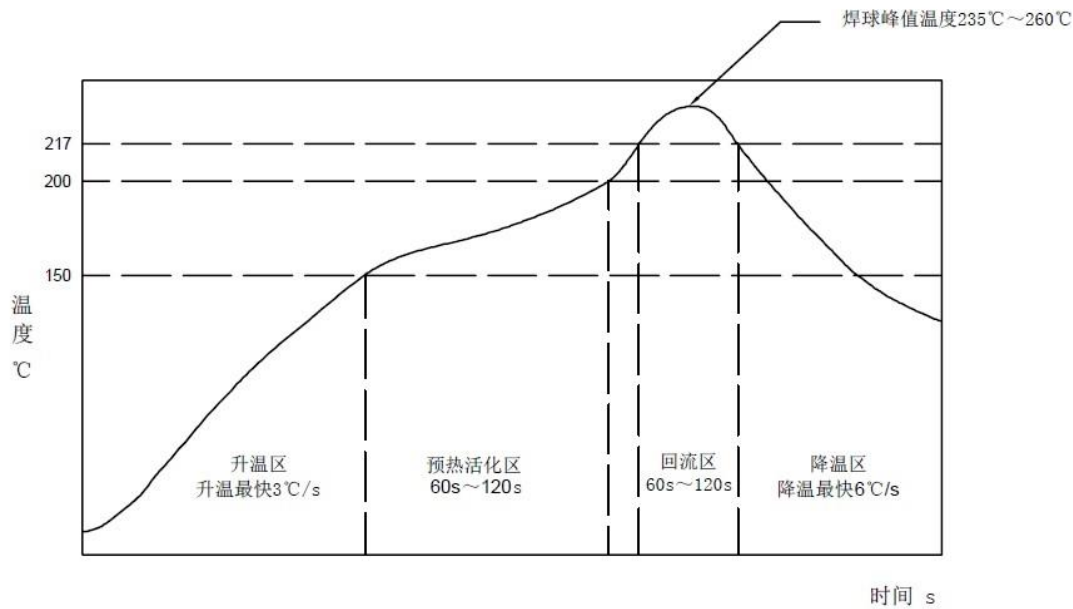


图 4-3 建议条件下的无铅回流曲线

5 绝对最大额定值及推荐工作范围

5.1 绝对最大额定值

序号	参数名称	参数值
1	储存时的环境温度 (T_{stg})	$-65^{\circ}\text{C} \sim +150^{\circ}\text{C}$
2	内核电源电压 (V_{CC})	$-0.5\text{V} \sim 4.0\text{V}$
3	I/O 电源电压 (V_{IO})	$-0.5\text{V} \sim 4.0\text{V}$
4	除 RESET#外, 所有引脚 (注 1)	$-0.5\text{V} \sim V_{IO} + 0.5\text{V}$
5	RESET#(注 1)	$-0.5\text{V} \sim V_{CC} + 0.5\text{V}$
6	A9 和 ACC	$-0.5\text{V} \sim +12.5\text{V}$

注 1: 输入或 I/O 引脚上的最小 DC 电压为 -0.5V 。在电压转换期间, 输入或 I/O 引脚可能会在长达 20 ns 的时间内将 V_{SS} 下冲至 -2.0V 。参见图 6-3。最大直流电压输入或 I/O 引脚上的电压为 $V_{CC}+0.5\text{V}$ 。在电压转换期间, 输入或 I/O 引脚可能会在长达 20 ns 的时间内过冲到 $V_{CC}+2.0\text{V}$ 。见图 6-4。

注 2: 一次不超过一个输出可以短路到地。短路持续时间不应大于 1 秒。

注 3: 高于绝对最大额定值下列出的应力可能会对设备造成永久性损坏。这仅是一个应力等级; 本数据手册操作部分所示的这些条件或任何其他条件下器械的功能操作并不意味着器械暴露于长期处于绝对最大额定值条件可能会影响器械可靠性。

注 4: V_{HH} 不得保持在 9.5V 累计超过 80 小时。

注 5: 器件在上述所列“绝对最大额定值”下工作, 可能对该器件造成永久性损坏。

5.2 推荐工作范围

序号	参数名称	参数值
1	储存时的环境温度 (T_{stg})	$-65^{\circ}\text{C} \sim +125^{\circ}\text{C}$
2	内核电源电压 (V_{CC})	$2.7\text{V} \sim 3.6\text{V}$
3	I/O 电源电压 (V_{IO})	$1.65\text{V} \sim V_{CC} + 200\text{mV}$

6 电特性参数

表 6-1 直流特性 (−40℃~85℃)

序号	参数	描述	测试条件	最小	典型值	最大	单位
1	I_{LI}	输入负载电流	$V_{IN} = V_{SS}$ 至 V_{CC} , $V_{CC} = V_{CC\ max}$	—	—	±10.0	μA
2	I_{LO}	输出漏电流	$V_{OUT} = V_{SS}$ 至 V_{CC} , $V_{CC} = V_{CC\ max}$	—	—	±2.0	μA
3	I_{CC1}	V_{CC} 有效读取电流	$CE\ \# = V_{IL}$, $OE\ \# = V_{IH}$, $V_{CC} = V_{CC\ max}$, 地址转换 $f = 5\text{MHz}$	—	55	110	mA
4	I_{CC2}	V_{CC} 页内读取电流	$CE\ \# = V_{IL}$, $OE\ \# = V_{IH}$, $V_{CC} = V_{CC\ max}$, 地址转换 $f = 33\text{MHz}$	—	9	20	mA
5	I_{CC3}	V_{CC} 擦除/编程电流 (注 1)	$CE\ \# = V_{IL}$, $OE\ \# = V_{IH}$, $V_{CC} = V_{CC\ max}$	—	45	100	mA
6	I_{CC4}	V_{CC} 旁路电流	$CE\ \# = OE\ \# = RESET\ \# = V_{IH}$, $V_{IH}=V_{IO}$ $V_{IL}=V_{SS}$, $V_{CC} = V_{CC\ max}$	—	—	3	mA
7	I_{CC5}	V_{CC} 复位电流 (注 5)	$CE\ \# = RESET\ \# = V_{IL}$, $V_{CC} = V_{CC\ max}$	—	10	20	mA
8	I_{CC6}	V_{CC} 深度断电流	$CE\ \# = OE\ \# = RESET\ \# = V_{IH}$, $V_{IH}=V_{IO}$ $V_{IL}=V_{SS}$, $V_{CC} = V_{CC\ max}$	—	30	70	μA
9	I_{CC7}	V_{CC} 上电电流 (注 4)	$CE\ \# = OE\ \# = RESET\ \# = V_{IO}$, $V_{CC} = V_{CC\ max}$	—	53	80	mA
10	V_{IL}	输入低电平 (注 2)		−0.1	—	$0.3 \times V_{IO}$	V
11	V_{IH}	输入高电平 (注 2)		$0.7 \times V_{IO}$	—	$V_{IO} + 0.3$	V
12	V_{OL}	输出低电平 (注 2, 注 6)	DQ15-DQ0: $I_{OL} = 100\mu\text{A}$ RY/BY#: $I_{OL}=2\text{mA}$	—	—	$0.15 \times V_{IO}$	V
13	V_{OH}	输出高电平 (注 2)	$I_{OH} = 100\mu\text{A}$	$0.85 \times V_{IO}$	—	—	V
14	V_{LKO}	低 V_{CC} 输出锁定电压		2.25	—	2.5	V
15	V_{RST}	低 V_{CC} 上电复位电压		—	1.0	—	V

注 1: 当嵌入式算法正在进行时, I_{CC} 处于活动状态。

注 2: $V_{IO}=1.65\text{V}$ 至 V_{CC} 或 2.7V 至 V_{CC} , 具体取决于型号。

注 3: $V_{CC}=3\text{V}$ 且 $V_{IO}=3\text{V}$ 或 1.8V 。当 V_{IO} 为 1.8V 时, I/O 引脚不能在 $>1.8\text{V}$ 下工作。

注 4: 在上电期间存在电流需求峰值, 系统需要能够提供该电流以确保部分正确初始化。

注 5: 如果嵌入式操作在复位开始时, 电流消耗将保持在嵌入式操作规范, 直到嵌入式操作因复位而停

止。当复位开始时如果没有嵌入式操作，或在嵌入式操作停止后， I_{CC5} 将在 t_{RPH} 的剩余时间内被拉起。 t_{RPH} 结束后，设备将进入待机模式直到下一次读取或写入。

注 6: RY/BY#输出的推荐上拉电阻为 5k 至 10k 欧姆。

表 6-2 直流特性 (−55℃~125℃)

序号	参数	描述	测试条件	最小	典型值	最大	单位
1	I_{LI}	输入负载电流	$V_{IN} = V_{SS}$ 至 V_{CC} , $V_{CC} = V_{CC\ max}$	—	—	± 10.0	μA
2	I_{LO}	输出漏电流	$V_{OUT} = V_{SS}$ 至 V_{CC} , $V_{CC} = V_{CC\ max}$	—	—	± 1.0	μA
3	I_{CC1}	V_{CC} 有效读取电流	$CE\ \# = V_{IL}$, $OE\ \# = V_{IH}$, $V_{CC} = V_{CC\ max}$, 地址转换 $f = 5MHz$	—	55	60	mA
4	I_{CC2}	V_{CC} 页内读取电流	$CE\ \# = V_{IL}$, $OE\ \# = V_{IH}$, $V_{CC} = V_{CC\ max}$, 地址转换 $f = 33MHz$	—	9	30	mA
5	I_{CC3}	V_{CC} 擦除/编程电流 (注 1)	$CE\ \# = V_{IL}$, $OE\ \# = V_{IH}$, $V_{CC} = V_{CC\ max}$	—	45	100	mA
6	I_{CC4}	V_{CC} 旁路电流	$CE\ \# = OE\ \# = RESET\ \# = V_{IH}$, $V_{IH}=V_{I0}$ $V_{IL}=V_{SS}$, $V_{CC} = V_{CC\ max}$	—	—	2	mA
7	I_{CC5}	V_{CC} 复位电流 (注 5)	$CE\ \# = RESET\ \# = V_{IL}$, $V_{CC} = V_{CC\ max}$	—	10	20	mA
8	I_{CC6}	V_{CC} 深度断电流	$CE\ \# = OE\ \# = RESET\ \# = V_{IH}$, $V_{IH}=V_{I0}$ $V_{IL}=V_{SS}$, $V_{CC} = V_{CC\ max}$	—	30	100	μA
9	I_{CC7}	V_{CC} 上电电流 (注 4)	$CE\ \# = OE\ \# = RESET\ \# = V_{I0}$, $V_{CC} = V_{CC\ max}$	—	53	80	mA
10	V_{IL}	输入低电平 (注 2)		-0.1	—	$0.3 \times V_{I0}$	V
11	V_{IH}	输入高电平 (注 2)		$0.7 \times V_{I0}$	—	$V_{I0} + 0.3$	V
12	V_{OL}	输出低电平 (注 2, 注 6)	DQ15-DQ0: $I_{OL} = 100\mu A$ RY/BY#: $I_{OL}=2mA$	—	—	$0.15 \times V_{I0}$	V
13	V_{OH}	输出高电平 (注 2)	$I_{OH} = 100\mu A$	$0.85 \times V_{I0}$	—	—	V
14	V_{LKO}	低 V_{CC} 输出锁定电压		2.25	—	2.5	V
15	V_{RST}	低 V_{CC} 上电复位电压		—	1.0	—	V

注 1: 当嵌入式算法正在进行时, I_{CC} 处于活动状态。

注 2: $V_{I0}=1.65V$ 至 V_{CC} 或 $2.7V$ 至 V_{CC} , 具体取决于型号。

注 3: $V_{CC}=3V$ 且 $V_{I0}=3V$ 或 $1.8V$ 。当 V_{I0} 为 $1.8V$ 时, I/O 引脚不能在 $>1.8V$ 下工作。

注 4: 在上电期间存在电流需求峰值, 系统需要能够提供该电流以确保零件正确初始化。

注 5: 如果嵌入式操作在复位开始时, 电流消耗将保持在嵌入式操作规范, 直到嵌入式操作因复位而停

止。当复位开始时如果没有嵌入式操作，或在嵌入式操作停止后， I_{CC5} 将在 t_{RPH} 的剩余时间内被拉起。 t_{RPH} 结束后，设备将进入待机模式直到下一次读取或写入。

注 6：RY/BY#输出的推荐上拉电阻为 5k 至 10k 欧姆。

在上电或断电期间， V_{CC} 必须始终大于或等于 V_{I0} ($V_{CC} \geq V_{I0}$)。

设备忽略所有输入，直到在 V_{CC} 和 V_{I0} 都上升到并保持在最小 V_{CC} 和 V_{I0} 阈值以上的时刻之后经过 t_{VCS} 的时间延迟。在 t_{VCS} 期间，设备执行通电复位操作。

在断电或电压降至 V_{CC} 锁定最大值 (V_{LKO}) 以下期间， V_{CC} 和 V_{I0} 电压必须在 t_{PD} 期间降至 V_{CC} 复位 (V_{RST}) 最小值以下，以便当 V_{CC} 和 V_{I0} 再次上升到它们的工作范围。见图 6-2。如果在电压下降期间 V_{CC} 保持高于 V_{LKO} 最大值，则该部件将保持初始化状态，并且当 V_{CC} 再次高于 V_{CC} 最小值将正常工作。如果部分因初始化不当而锁定，则可以使用硬件复位来正确初始化这部分。

必须采取正常的电源去耦预防措施，以稳定 V_{CC} 和 V_{I0} 电源。系统中的每个设备都应该通过适当的电容将 V_{CC} 和 V_{I0} 电源解耦或靠近封装连接（该电容通常在 0.1 μ F 量级）。 V_{I0} 在任何时候都不应大于 $V_{CC}200\text{mV}$ ($V_{CC} \geq V_{I0} - 200\text{mV}$)。

表 6-3 上下电电压和时间

Symbol	Parameter	Min	Max	Unit
V_{CC}	V_{CC} Power Supply	2.7	3.6	V
V_{LKO}	V_{CC} level below which re-initialization is required (Note 1)	2.25	2.5	V
V_{RST}	V_{CC} and V_{I0} Low voltage needed to ensure initialization will occur (Note 1)	1.0	–	V
t_{VCS}	V_{CC} and $V_{I0} \geq$ minimum to first access (Note 1)	300	–	μ s
t_{PD}	Duration of $V_{CC} \leq V_{RST}(\text{min})$ (Note 1)	15	–	μ s

图 6-1 上电

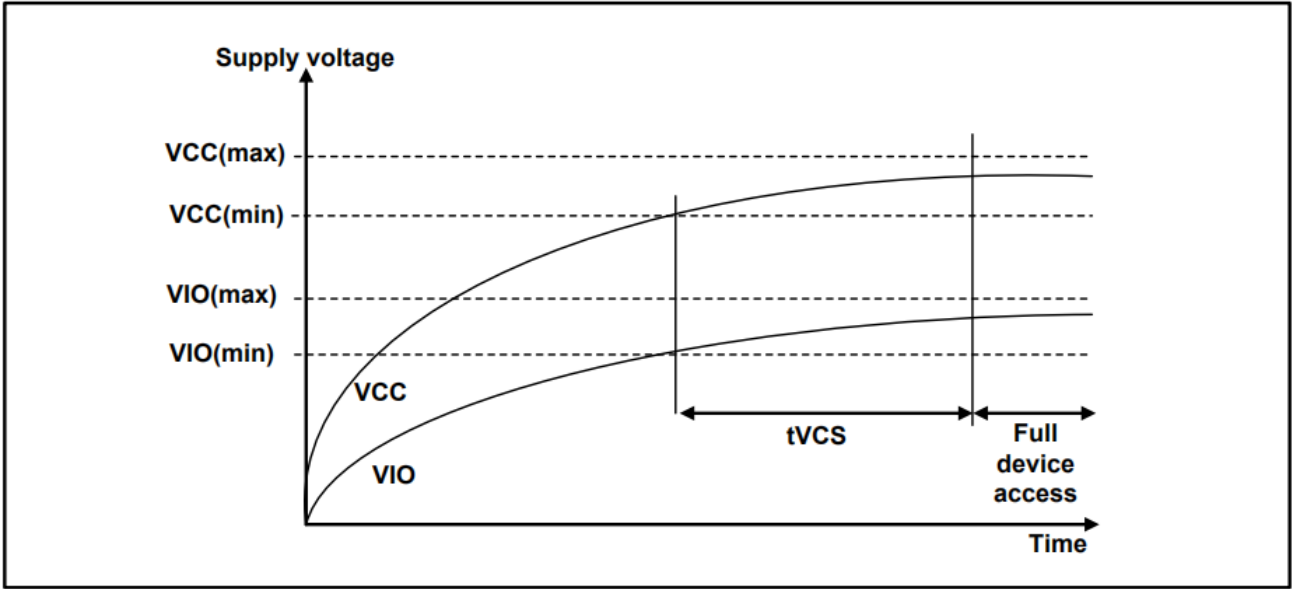
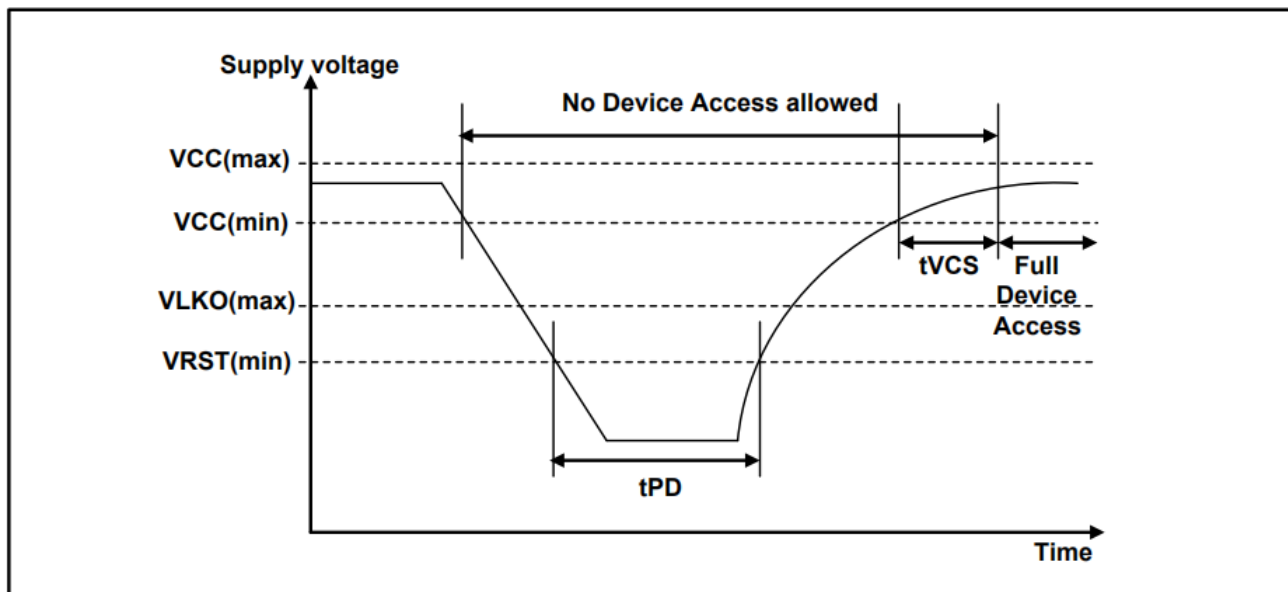


图 6-2 下电和电压下降



输入信号过冲

图 6-3 最大反向过冲波形

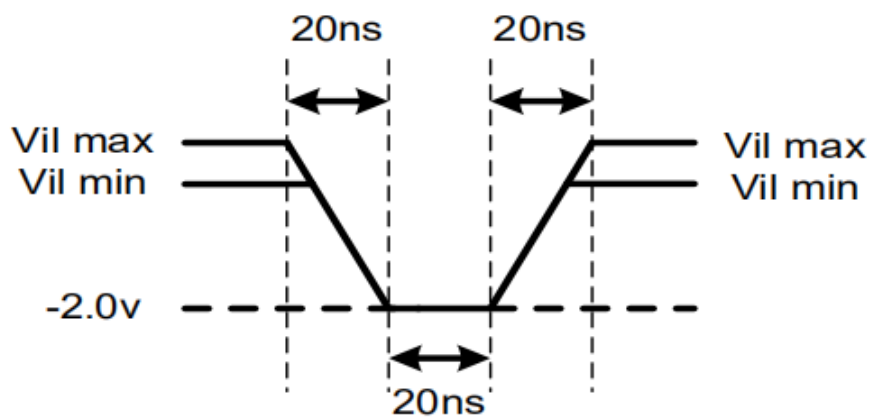
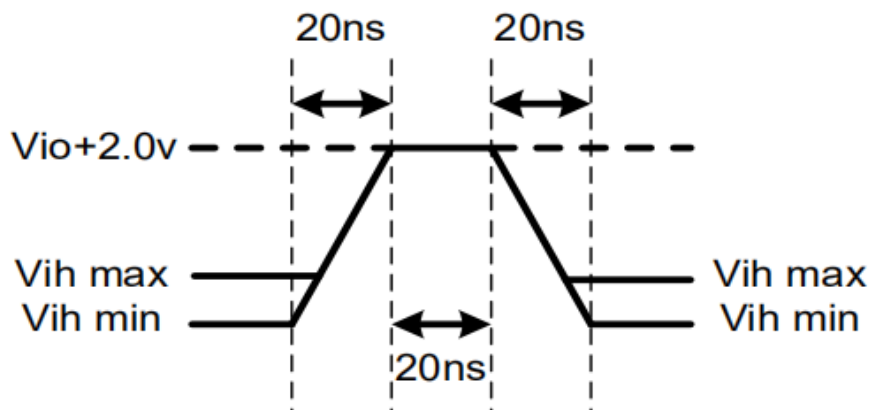


图 6-4 最大正向过冲波形



7 功能描述

7.1 概述

HWD29S01GPBGA64M3 是具有 1Gbit 容量, 3.0V 核, 多功能 I/O, 非易失性的闪存设备。该设备有 8/16 位 (word) 数据总线。在每一个总线传输周期里为所有读访问提供 8/16 位数据。每一个总线传输周期里, 写操作用到 8/16 位数据。

HWD29S01GPBGA64M3 包含了最好的直接存储 (XIP) 和数据存储闪存功能, 是具有高密度和快速编程数据存储速度的快速随机访问 XIP 闪存。

根据 I/O 电压, 随机位置读访问花费 100ns 到 130ns。每随机访问顺序相邻 32 字节数据称为一页。访问读取同一页其它地址可以通过改变低 4 位字地址实现。同一页每次访问花费 15ns 到 30ns, 这种叫页读取模式。改变更高的字地址将会选择不同的页, 从而开始新的访问。所有的读访问都是异步的。

设备控制逻辑分为两种并行操作部分, HIC (Host Interface Controller) 和 EAC (Embedded Algorithm Controller)。HIC 监控在和主机系统读写数据传输时, 设备输入和驱动输出信号水平。读传输时, HIC 从当前输入地址映射处传输数据; 放置写传输地址和数据信息到 EAC 命令存储器; 通知 EAC 开启传输, 硬件复位, 写传输。EAC 查找存储指令, 然后启动一个写传输, 为合法的命令序列执行相应的嵌入算法。

改变存储阵列里的非易失数据需要一个复杂的操作序列, 称之为嵌入算法 EA (Embedded Algorithm)。这个算法管理整个设备内部的 EAC, 主要的算法执行主阵列数据的编程和擦除。主机系统写命令指令到闪存器件地址空间, EAC 接收到命令, 执行所有完整命令中的必要步骤, 并通过 EA 过程提供状态信息。

每个存储位的擦除状态是逻辑 1, 程可以将逻辑 1 改变到逻辑 0, 仅擦除操作可以将 0 值改变到 1 值。一个擦除操作必须一次性执行顺序相邻的 128-kbyte 且整组数据处于一个扇区。

编程是通过 512-byte 写缓冲区完成的。在开始编程操作前, 可以将 1 到 256 个的任意字写入到写缓冲区。在闪存阵列里, 每 512byte 被顺序串在一个线上, 称之为线。一次编程操作将易失性的数据从写缓冲区写到非易失性存储阵列线上。这种操作称之为写缓冲区编程。

复位或完成写缓冲区的任何操作之后写缓冲区被填充 1。任何地址都不能通过写缓冲命令写成 0, 只能默认填充的是 1。在编程操作时任何写缓冲区的 1 并不影响存储阵列中的数据。

每页数据是通过加载到写缓冲区, 再传输到存储阵列线上。

在编程和擦除操作时, 扇区是被 ASP (Advanced Sector Protection, 先进扇区保护) 功能设置所单独保护的。ASP 提供几个硬件和软件控制, 易失和非易失性的方法来选择哪些扇区在编程和擦除操作时被保护。

7.2 引脚控制

7.2.1 硬件 RESET#输入操作

RESET#输入提供了将设备重置为待机状态的硬件方法。当 RESET#被驱动为低电平至少一个 t_{RP} 周期时，器件立即：

- 终止任何正在进行的操作，
- 退出任何 AS0，
- 三态所有输出，
- 重置状态寄存器，
- 将 EAC 重置为待机状态。
- CE#在重置操作期间被忽略 (t_{RPH})。
- 满足复位电流规格 (ICC5) CE#的，必须保持高。

为了确保数据完整性，一旦设备准备好接受另一个命令序列，任何被中断的操作都应该重新启动。

7.2.2 输出禁用 (OE#)

当 OE#输入为 V_{IH} 时，器件的输出被禁用。输出置于高阻抗状态。(RY/BY#除外。)

7.2.3 多功能 I/O (VIO) 控制

多功能 I/O (VIO) 控制允许主机系统设置设备在所有输入和输出 (地址、控制和 DQ 信号) 上产生和容许的电压电平。VIO 范围为 1.65V 至 VCC。

例如，1.65-3.6V 的 VIO 允许在 1.8 或 3 伏电平的 I/O，驱动和接收来自同一数据总线上其他 1.8 或 3V 设备的信号。

设备驱动的最大输出电压电平和设备可接受的输入电平由 VIO 电源确定。该电源允许设备驱动和接收信号同一总线上具有不同于器件核心电压的接口信号电平的其他器件。

7.2.4 RY/BY#

RY/BY#是一个专用的开漏输出引脚，指示嵌入式算法，上电复位 (POR) 或硬件复位是正在进行中还是完成。当 POR 期间 VCC 高于 VCC 最小值时，或在 RESET#的下降沿之后，RY/BY#状态在命令序列中最后一个 WE#脉冲的上升沿后有效。由于 RY/BY#是漏极开路输出，因此几个 RY/BY#引脚可以与一个上拉电阻并联到 VIO。

如果输出为低 (忙)，则设备正在主动擦除、编程或复位。(这包括在擦除挂起模式下编程)。如果输出为高电平 (就绪)，则设备准备好读数据 (包括在擦除挂起模式期间)，或处于待机模式。

数据轮询状态显示了每个操作中 RY/BY#的输出。

如果嵌入式算法失败 (由于最大脉冲或扇区被锁定而导致编程/擦除失败)，RY/BY#将保持低电平 (忙)，直到状态寄存器位 4 和 5 被清除并且复位命令发出。这包括对锁定扇区的擦除或编程。

7.2.5 字/字节编程 (BYTE#)

BYTE#管脚控制器件数据 I/O 管脚是在 x8 模式还是 x16 模式下工作。如果 BYTE#引脚设置为逻辑“1”，则器件处于 x16 模式，DQ0-DQ15 有效并由 CE#和 OE#控制。

如果 BYTE#引脚设置为逻辑“0”，则器件处于 x8 模式，只有数据 I/O 引脚 DQ0-DQ7 有效并由 CE#和 OE#控制。数据 I/O 引脚 DQ8-DQ14 为三态，DQ15 引脚用作 LSB (A-1) 地址功能的输入。

7.3 时序特性

7.3.1 测试条件

图 7-1 测试设置

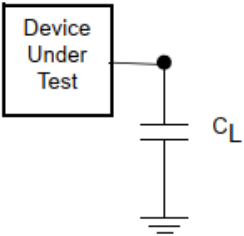


表 7-1 测试条件

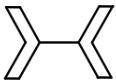
测试条件	数值	单位
输出负载电容，CL（包括夹具电容）	30	pF
输入上升和下降时间	1.5	ns
输入脉冲电平	0.0 - V _{IO}	V
输入时序测量参考电平	0.5V _{IO}	V
输出时序测量参考电平	0.5V _{IO}	V

注 1：在 V_{ILmax} 和 V_{IHmin} 之间测试。

7.3.2 波形变化描述

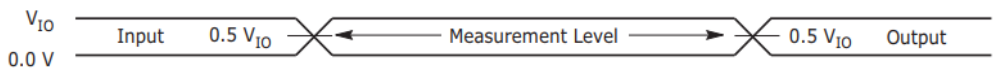
表 7-2 波形变化描述

波形	输入	输出
	稳定状态	
	从 H 变为 L	
	从 L 变为 H	

	不在乎，允许任何更改	变化，状态未知
	不适用	中心线为高阻抗状态 (High Z)

7.3.3 输入波形和测量电平

图 7-2 输入波形和测量电平



注 1：如果 $V_{I/O} < V_{CC}$ ，输入测量参考电平为 $0.5V_{I/O}$ 。

7.3.4 上电复位（POR）和热复位

必须采取正常的电源去耦预防措施，以稳定 V_{CC} 和 $V_{I/O}$ 电源。系统中的每个设备都应该通过适当的电容将 V_{CC} 和 $V_{I/O}$ 电源解耦或靠近封装连接（该电容通常在 $0.1\mu\text{F}$ 量级）。

表 7-3 上电和复位参数

参数	描述	最小	单位
t_{VCS}	V_{CC} 设置时间到第一次访问（注 1）	300	μs
t_{VIOS}	$V_{I/O}$ 设置时间到第一次访问（注 1）	300	μs
t_{RPH}	RESET#低电平到 CE#低电平	35	μs
t_{RP}	RESET#低电平脉宽	5200	ns
t_{RH}	RESET#高电平到 CE#低电平时间	150	ns
t_{CEH}	CE#高电平脉宽	20	ns

注 1：从 V_{CC} 达到 V_{CC} 最小值和 $V_{I/O}$ 达到 $V_{I/O}$ 最小值到复位时的 V_{IH} 和 CE#上的 V_{IL} 测量的定时。

注 2：POR 期间 RESET#低是可选的。如果 POR 中 RESET 置高， t_{RPH} 、 t_{VIOS} 或 t_{VCS} 中较晚的一个将决定 CE#何时可能走低。如果 RESET#在 t_{VIOS} 或 t_{VCS} 满足后保持低电平，则从 t_{VIOS} 或 t_{VCS} 结束时测量 t_{RPH} 。在 CE#走低之前 t_{RH} RESET 必须置高。

注 3：上电期间的 $V_{CC} \geq V_{I/O}$ 。

注 4： V_{CC} 和 $V_{I/O}$ 斜坡速率可以是非线性的。

注 5： t_{RP} 和 t_{RH} 之和必须等于或大于 t_{RPH} 。

7.3.4.1 上电（冷）复位（POR）

在电源上升期间， $V_{I/O}$ 电源电压必须保持小于或等于 V_{CC} 电源电压。 V_{IH} 也必须保持小于或等于 $V_{I/O}$ 供应量。

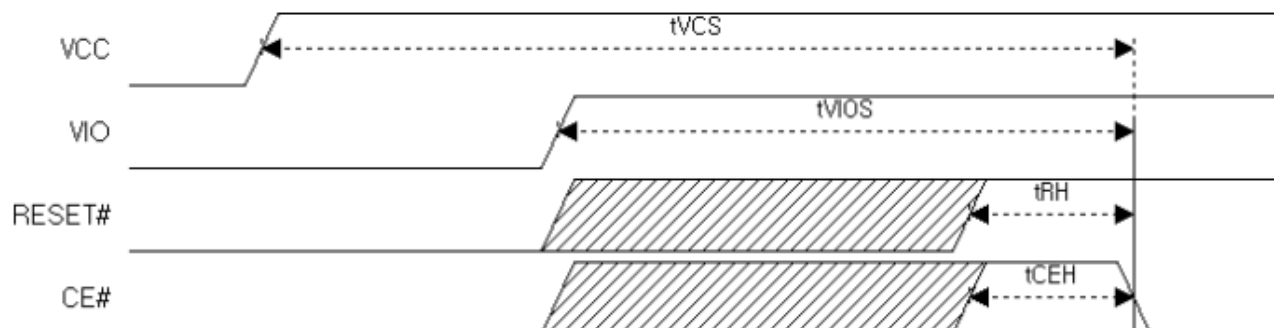
冷复位嵌入式算法需要相对较长的数百 μs 周期（ t_{VCS} ）来从非易失性存储器加载所有 EAC 算法和默认状态。在冷复位期间包括 CE#和 RESET#在内的所有控制信号都被忽略。如果在 t_{VCS} 期间 CE#较低，则器件在 t_{VCS} 期间可能会消耗高于正常 POR 电流，但 CE#的水平不会影响冷复位 t_{EA} 。CE#或 OE#必须在 t_{VCS} 之后从高转换到低，才能进行有效的读或写操作。在 t_{VCS} 期间，RESET#可能为高或低。如果 RESET#在 t_{VCS} 期间为低电平，则在 t_{VCS} 结束时可以保持低电平将设备保持在硬件复位状态。如果 RESET#在 t_{VCS} 结束时为高电平，则设备将进入待机状态。

当首次通电时，电源电压低于 V_{RST} ，然后上升到工作范围最小值，启动内部器件配置和热复位活动。POR 操作的持续时间（ t_{VCS} 或 t_{VIOS} ）忽略 CE#。此 POR 期间的 RESET#Low 是可选的。如果在 POR 期间 RESET#被驱

动为低电平，则它必须满足硬件复位参数 t_{RP} 和 t_{RPH} 。其中在这种情况下，复位操作将在 t_{VCS} 或 t_{VLOS} 或 t_{RPH} 中的较晚者完成。

在冷复位期间，器件将消耗 I_{CC7} 电流。

图 7-3 上电时序



7.3.4.2 硬件（热）复位

在硬件复位 (t_{RPH}) 期间，器件将消耗 I_{CC5} 电流。

当 RESET# 继续保持在 V_{SS} 时，器件消耗 CMOS 待机电流 (I_{CC4})。如果 RESET# 保持在 V_{IL} ，而不是 V_{SS} ，则待机电流更大。

如果在 t_{VCS} 后 RESET# 被断言为低电平时，设备尚未完成冷复位，则将执行冷 RESET#EA 而不是热 RESET#，需要 t_{VCS} 时间来完成。参见

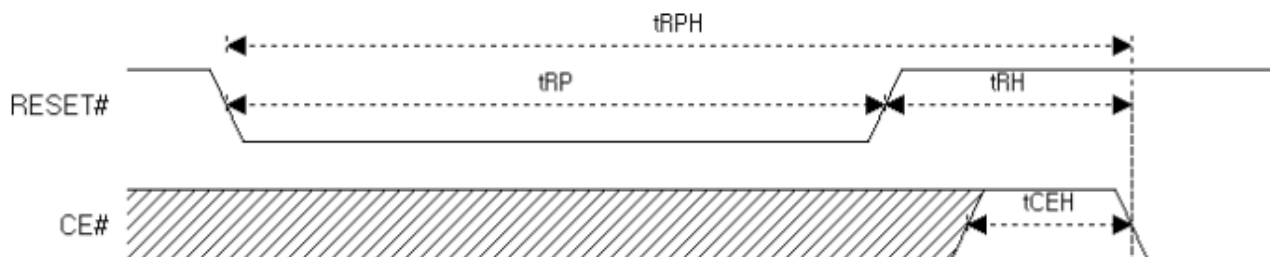
图 7-4。

在设备完成 POR 并进入待机状态后，任何后续到硬件复位状态的转换都将启动热复位嵌入式算法。热复位比冷复位，需要几十 μs (t_{RPH}) 才能完成。在热复位 EA 期间，停止任何正在进行的嵌入算法，并且 EAC 返回到其 POR 状态，而无需重新加载 EAC 算法来自非易失性存储器。热复位 EA 完成后，如果 RESET# 保持低电平，接口将保持在硬件复位状态。当 RESET# 返回高电平时，接口将转换到旁路状态。如果在热复位 EA 结束时 RESET# 为高电平，接口将直接转换到待机状态。

如果 POR 在 t_{VCS} 结束时尚未正确完成，则稍后过渡到硬件复位状态将导致过渡到上电复位接口状态并启动冷复位 EA 分层算法。这确保了即使系统上电压上升的某些方面导致 POR 无法正确启动或完成，设备也可以完成冷复位。RY/BY# 引脚为冷复位或热复位期间为低电平，表示设备正忙于执行复位操作。

硬件复位由启动 V_{IL} 的 RESET# 信号启动。

图 7-4 硬件复位时序



7.4 交流特性

7.4.1 异步读取操作

表 7-4 读取操作 $V_{IO}=V_{CC}=2.7V \sim 3.6V (-40^{\circ}C \sim 85^{\circ}C)$

序号	参数	参数	描述	速度	速度	速度	单位
1	tAVAV	tRC	读取周期时间	90	100	110	ns
2	tAVQV	tACC	地址到输出延迟 CE#=VIL, OE#=VIL	—	100	110	ns
3	tELQV	tCE	芯片使能到输出延迟 OE#=VIL	—	100	110	ns
4		tPACC	页面访问时间	—	15	20	ns
5	tGLQV	tOE	输出使能到输出延迟	—	25	20	ns
6	tAXQX	tOH	地址输出保持时间, CE#或 OE#, 任何一个先发生	—	0	25	ns
7	tEHQZ	tDF	芯片使能或输出使能 到输出高阻	—	15	0	ns
8		tOEH	输出使能保持时间, 读	—	0	15	ns
9		tOEH	输出使能保持时间, 翻转和 Data#翻转	—	10	10	ns
10	tELBL	tELFL	CE#低到 BYTE#低	—	10	10	ns
11	tELBH	tFLQV	CE#低到 BYTE#高	—	10	10	ns
12	tBLQV	tFLQV	BYTE#低到输出有效	—	1	1	us
13	tBHQV	tFHQV	BYTE#高到输出有效	—	1	1	us

表 7-5 读取操作 $V_{IO}=1.65V \sim V_{CC}$, $V_{CC}=2.7V \sim 3.6V (-40^{\circ}C \sim 85^{\circ}C)$

序号	参数	参数	描述	速度	速度	速度	单位
1	tAVAV	tRC	读取周期时间	100	110	120	ns
2	tAVQV	tACC	地址到输出延迟 CE#=VIL, OE#=VIL	—	110	120	ns
3	tELQV	tCE	芯片使能到输出延迟 OE#=VIL	—	110	120	ns
4		tPACC	页面访问时间	—	25	30	ns
5	tGLQV	tOE	输出使能到输出延迟	—	35	35	ns
6	tAXQX	tOH	地址输出保持时间, CE#或 OE#, 任何一个先发生	—	0	0	ns
7	tEHQZ	tDF	芯片使能或输出使能 到输出高阻	—	20	20	ns
8		tOEH	输出使能保持时间, 读	—	0	0	ns
9		tOEH	输出使能保持时间, 翻转和 Data#翻转	—	10	10	ns
10	tELBL	tELFL	CE#低到 BYTE#低	—	10	10	ns
11	tELBH	tFLQV	CE#低到 BYTE#高	—	10	10	ns
12	tBLQV	tFLQV	BYTE#低到输出有效	—	1	1	us
13	tBHQV	tFHQV	BYTE#高到输出有效	—	1	1	us

表 7-6 读取操作 $V_{IO}=V_{CC}=2.7V \sim 3.6V (-55^{\circ}C \sim 125^{\circ}C)$

序号	参数	参数	描述	速度	速度	速度	单位
1	tAVAV	tRC	读取周期时间	—	110	130	ns
2	tAVQV	tACC	地址到输出延迟 CE#=VIL, OE#=VIL	—	110	130	ns
3	tELQV	tCE	芯片使能到输出延迟 OE#=VIL	—	110	130	ns
4		tPACC	页面访问时间	—	15	30	ns
5	tGLQV	tOE	输出使能到输出延迟	—	25	25	ns
6	tAXQX	tOH	地址输出保持时间, CE#或 OE#, 任何一个先发生	—	0	—	ns
7	tEHQZ	tDF	芯片使能或输出使能 到输出高阻	—	15	15	ns
8		tOEH	输出使能保持时间, 读	—	0	0	ns
9		tOEH	输出使能保持时间, 翻转和 Data#翻转	—	10	10	ns
10	tELBL	tELFL	CE#低到 BYTE#低	—	10	10	ns
11	tELBH	tFLQV	CE#低到 BYTE#高	—	10	10	ns
12	tBLQV	tFLQV	BYTE#低到输出有效	—	1	1	us
13	tBHQV	tFHQV	BYTE#高到输出有效	—	1	1	us

表 7-7 读取操作 $V_{IO}=1.65V \sim V_{CC}$, $V_{CC}=2.7V \sim 3.6V (-55^{\circ}C \sim 125^{\circ}C)$

序号	参数	参数	描述	速度	速度	速度	单位
1	tAVAV	tRC	读取周期时间	—	120	130	ns
2	tAVQV	tACC	地址到输出延迟 CE#=VIL, OE#=VIL	—	120	130	ns
3	tELQV	tCE	芯片使能到输出延迟 OE#=VIL	—	120	130	ns
4		tPACC	页面访问时间	—	25	30	ns
5	tGLQV	tOE	输出使能到输出延迟	—	35	35	ns
6	tAXQX	tOH	地址输出保持时间, CE#或 OE#, 任何一个先发生	—	0	—	ns
7	tEHQZ	tDF	芯片使能或输出使能 到输出高阻	—	20	20	ns
8		tOEH	输出使能保持时间, 读	—	0	0	ns
9		tOEH	输出使能保持时间, 翻转和 Data#翻转	—	10	10	ns
10	tELBL	tELFL	CE#低到 BYTE#低	—	10	10	ns
11	tELBH	tFLQV	CE#低到 BYTE#高	—	10	10	ns
12	tBLQV	tFLQV	BYTE#低到输出有效	—	1	1	us
13	tBHQV	tFHQV	BYTE#高到输出有效	—	1	1	us

图 7-5 背对背读 (tACC) 操作时序图

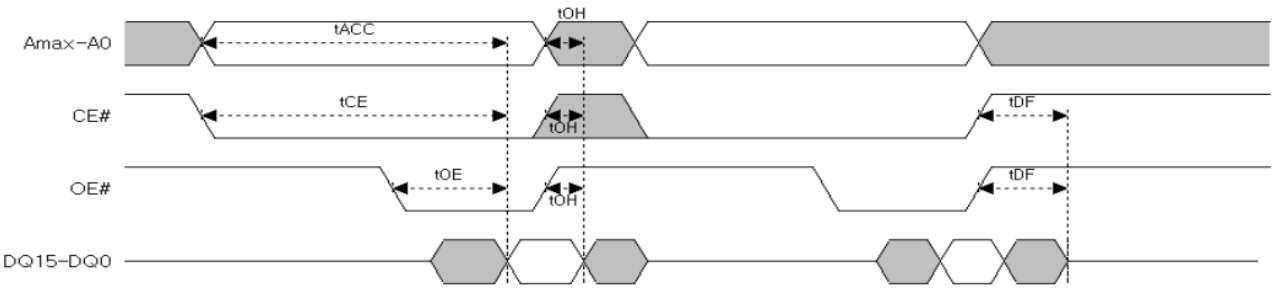


图 7-6 背对背读 (t_{RC}) 操作时序图

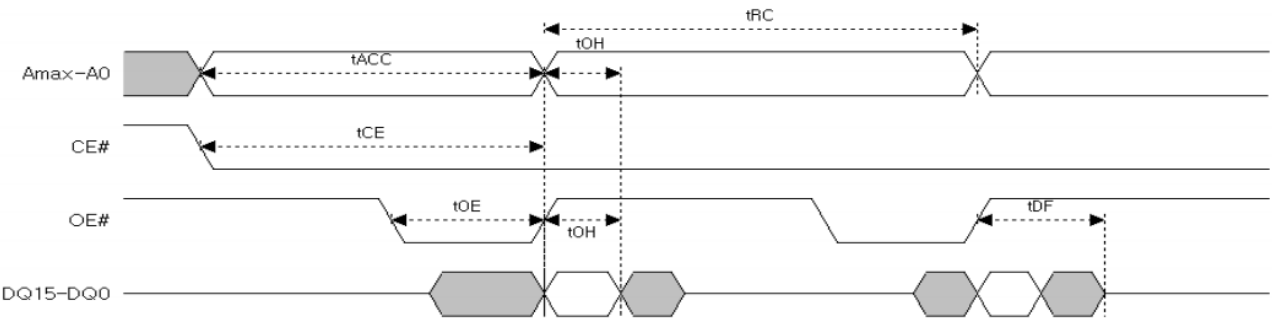


图 7-7 页读取操作时序图

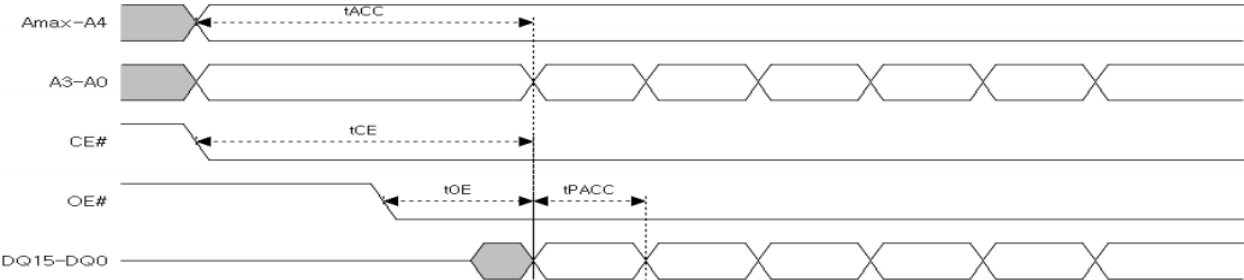


图 7-8 随机读 AC 时序图 (8 位模式)

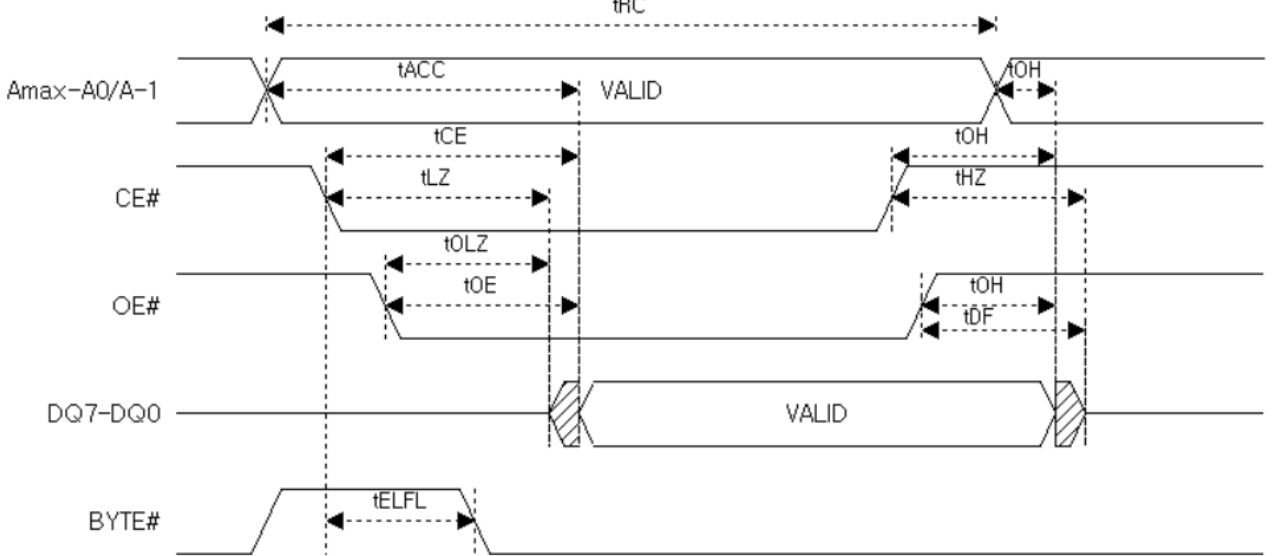


图 7-9 随机读 AC 时序图 (16 位模式)

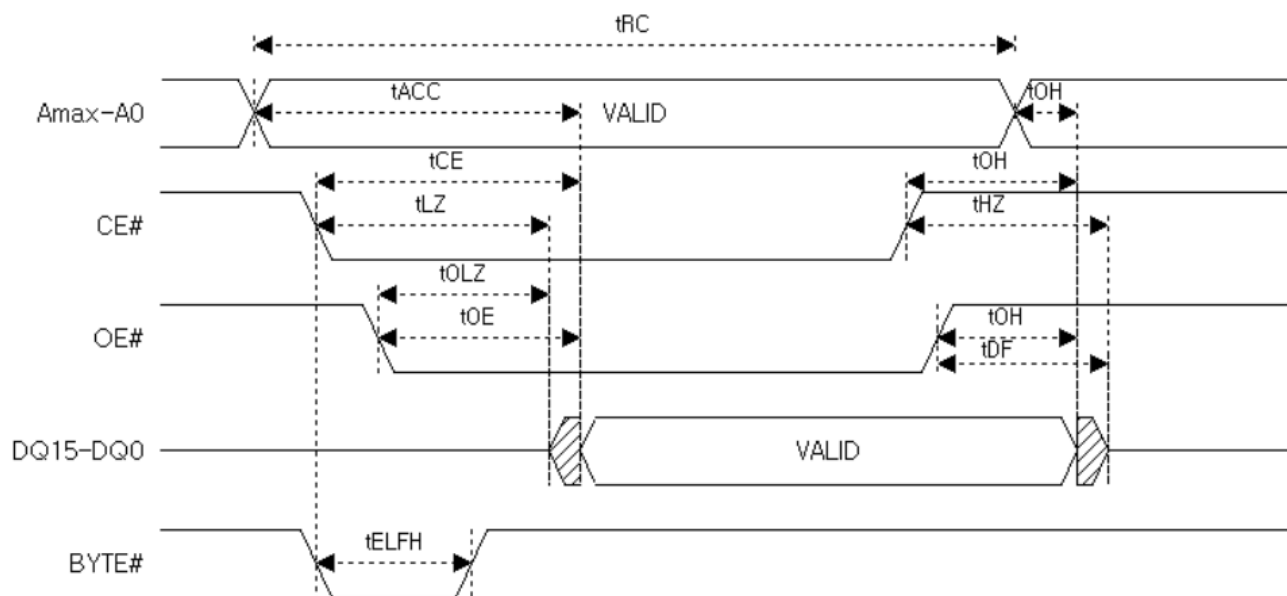
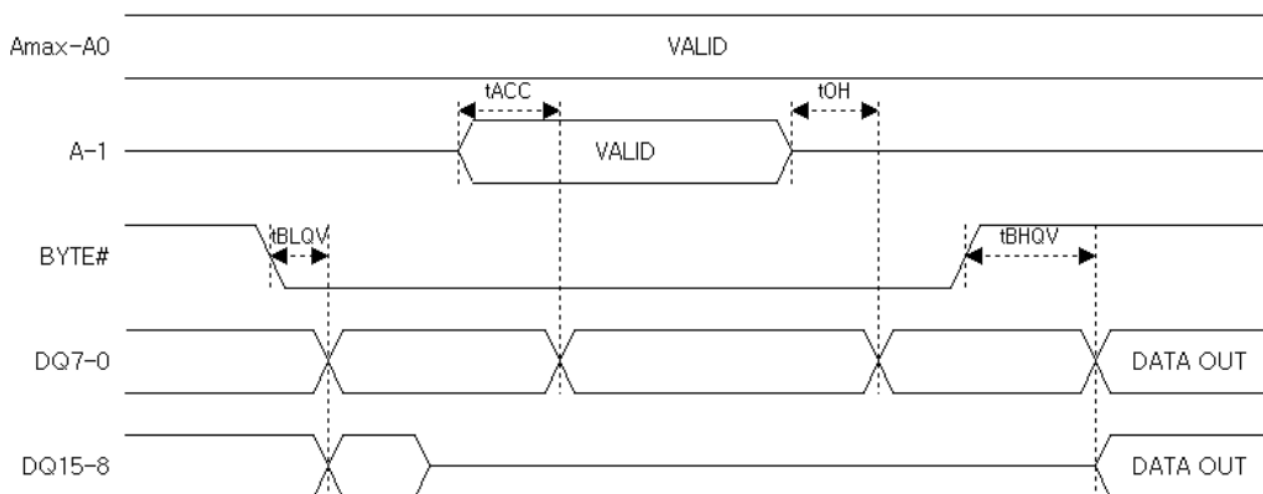


图 7-10 BYTE#传输读 AC 时序图



7.4.2 异步写操作

表 7-8 写操作

序号	JEDEC 参数	Std 参数	描述		$V_{I0}=2.7V\sim V_{CC}$	$V_{I0}=21.65V\sim V_{CC}$	单位
1	t_{AVAV}	t_{WC}	写周期时间	Min	60		ns
2	t_{AVWL}	t_{AS}	地址建立时间	Min	0		ns
3		t_{ASO}	翻转位轮询期间地址建立时间到 OE#低	Min	15		ns
4	t_{WLAX}	t_{AH}	地址保持时间	Min	45		ns
5		t_{AHT}	翻转位轮询期间从 CE#或 OE#高开始的地址保持时间	Min	0		ns
6	t_{DVWH}	t_{DS}	数据建立时间	Min	30		ns
7	t_{WHDX}	t_{DH}	数据保持时间	Min	0		ns
8		t_{OEPH}	翻转位轮询期间或状态寄存器读之后输出使能高	Min	20		ns
9	t_{GHWL}	t_{GHWL}	写之前的读恢复时间 (OE#高到 WE#低)	Min	0		ns
10	t_{ELWL}	t_{CS}	CE#建立时间	Min	0		ns
11	t_{WHEH}	t_{CH}	CE#保持时间	Min	0		ns
12	t_{WLWH}	t_{WP}	WE#脉冲宽度	Min	35		ns
13	t_{WHWL}	t_{WPH}	WE#高脉冲宽度	Min	30		ns

图 7-11 背对背写操作时序图

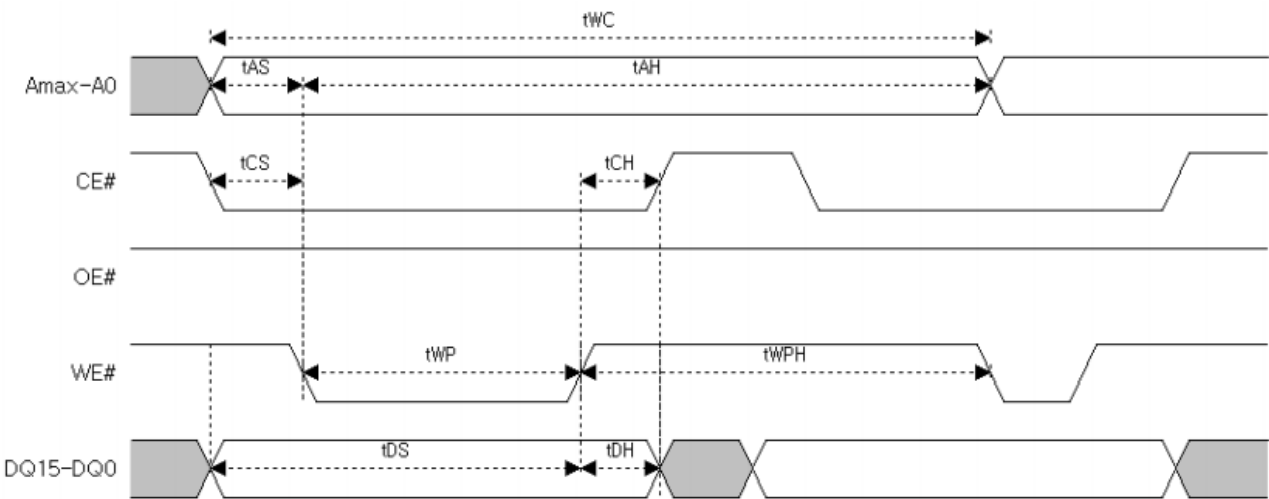


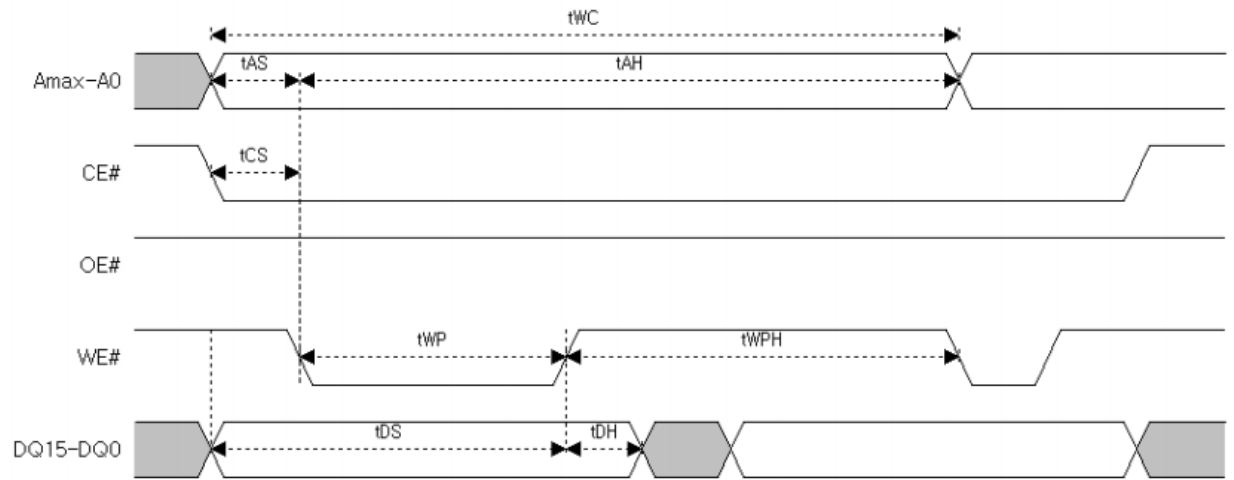
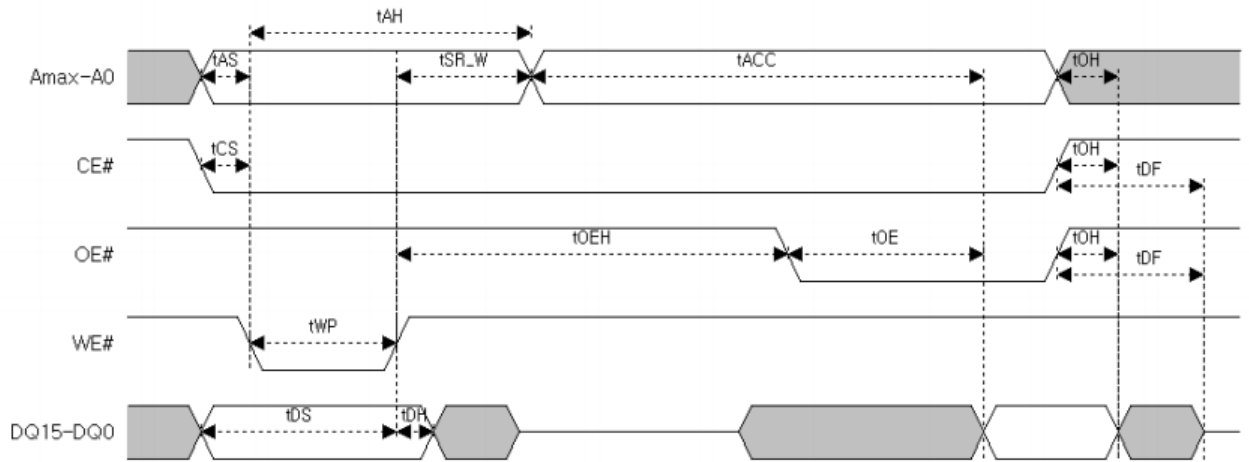
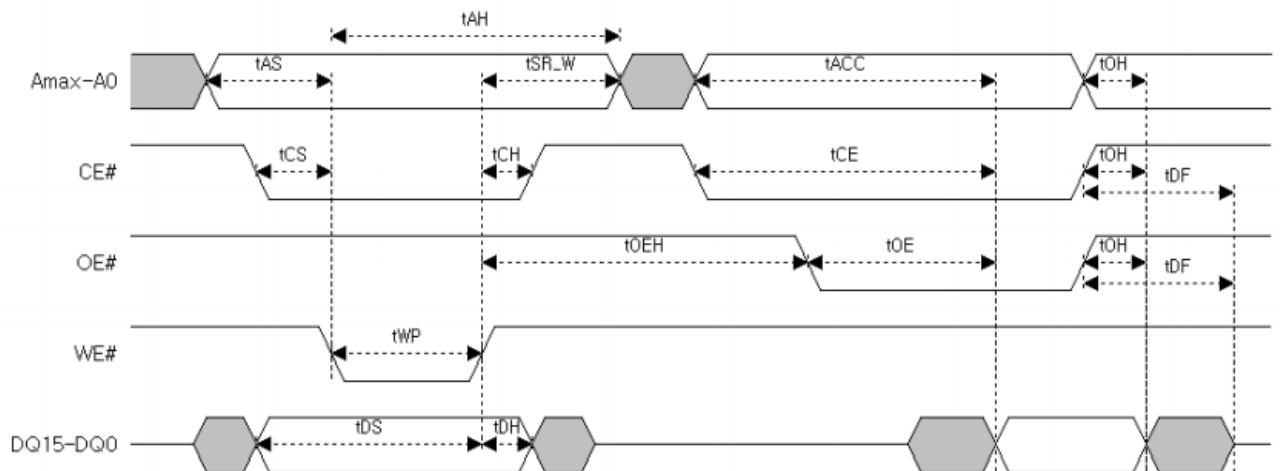
图 7-12 背对背 (CE#V_{IL}) 写操作时序图图 7-13 写到读 (t_{acc}) 操作时序图图 7-14 写到读 (t_{ce}) 操作时序图

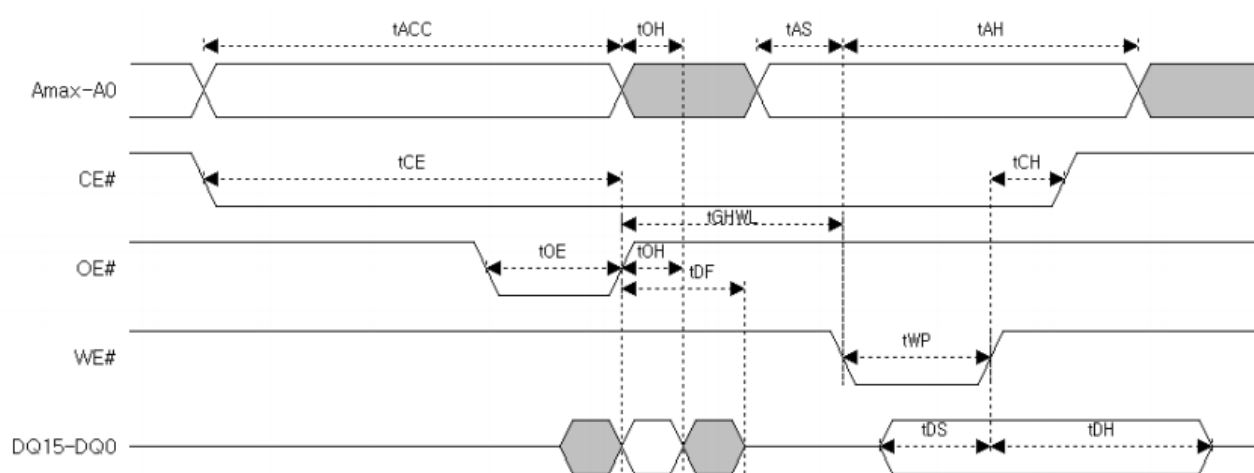
图 7-15 读到写 (CE#V_{IL}) 操作时序图

图 7-16 读到写 (CE#翻转) 操作时序图

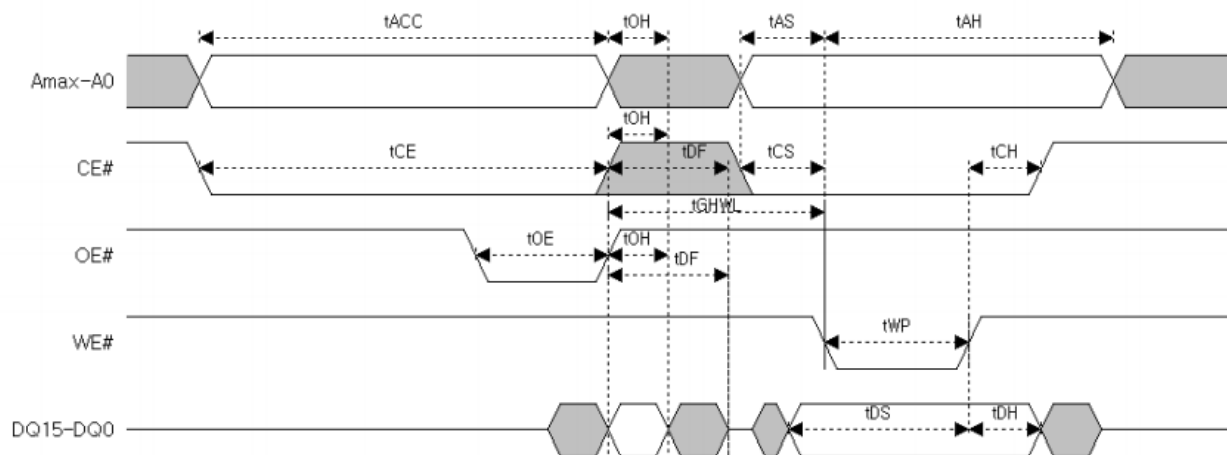


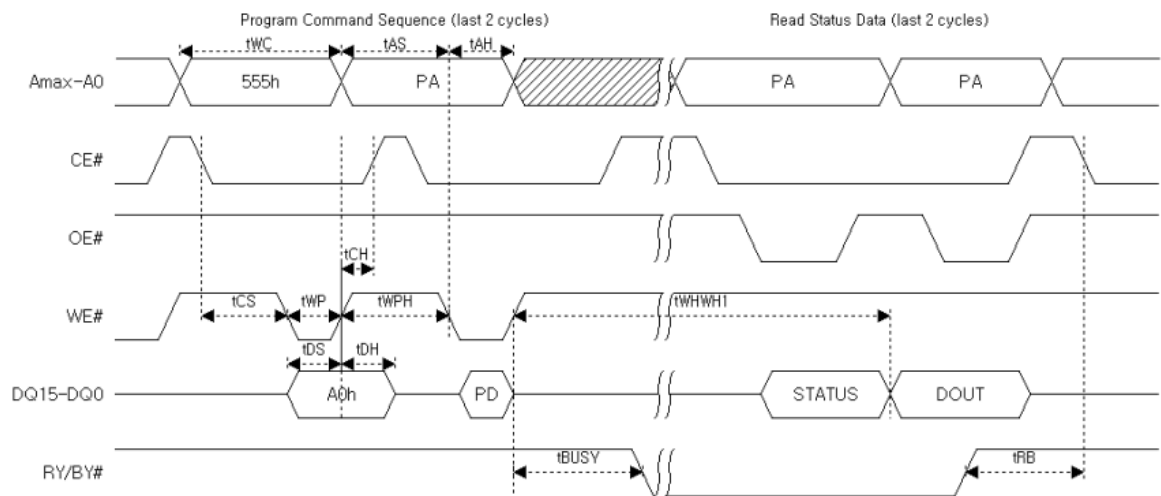
表 7-9 擦除和编程操作

序号	JEDEC 参数	Std 参数	描述		$V_{I0}=2.7V\sim V_{CC}$	$V_{I0}=1.65V\sim V_{CC}$	单位
1	t_{WHWH1}	t_{WHWH1}	写缓冲区编程操作	典型			μs
2			有效的写缓冲区每字编程操作	典型			μs
3			每字或页编程操作	典型			μs
4	t_{WHWH2}	t_{WHWH2}	扇区擦除操作	典型			ms
5		t_{BUSY}	擦除/编程有效到 RY/BY#延迟	最大	90		ns
6		$t_{SR/W}$	读和写操作之间的延	最小	10		ns
7		t_{ESL}	擦除挂起延迟	最大			μs
8		t_{PSL}	编程挂起延迟	最大			μs
9		t_{RB}	RY/BY#恢复时间	最小	0		μs

注 1：WE#上升沿，在切换到另一个地址前必须等 $t_{SR/W}$ 间

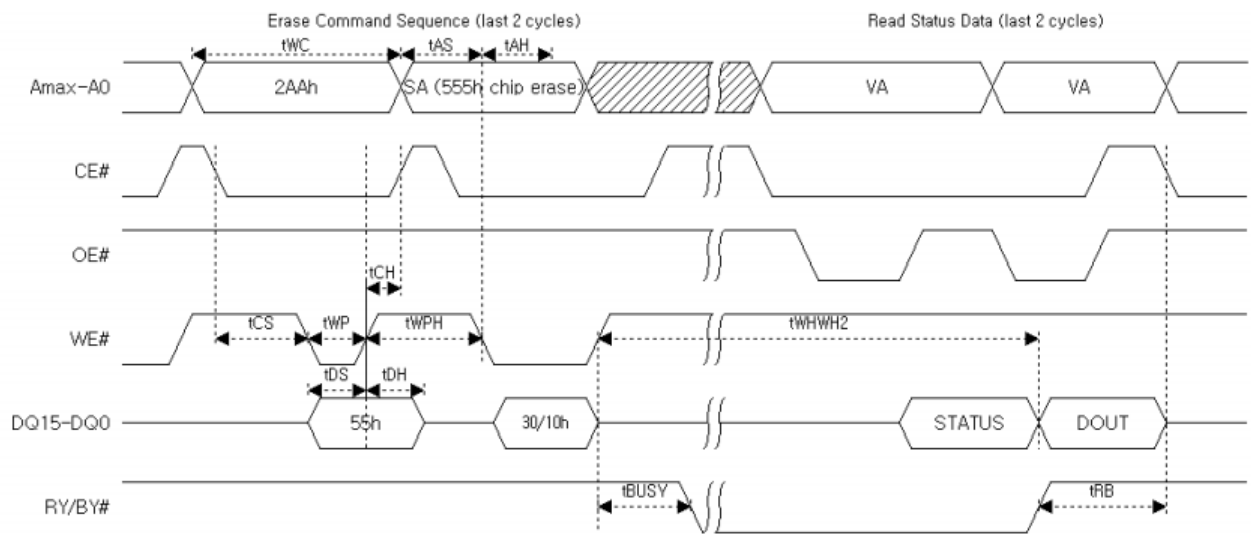
注 2：特性值参见表 7-37

图 7-17 编程操作时序



注 1：PA = 编程地址，PD = 编程数据，DOUT 是编程地址存储的真实数据。

图 7-18 全片/扇区擦除操作时序



注 1: PA = 编程地址, VA=读状态数据的有效地址。

图 7-19 加速编程时序图



表 7-10 加速编程交流时序

参数	描述
t_{VHVPP}	VHH 在 WP#/ACC (最小 250ns) 上升和下降时间

图 7-20 深度断电模式交流时序

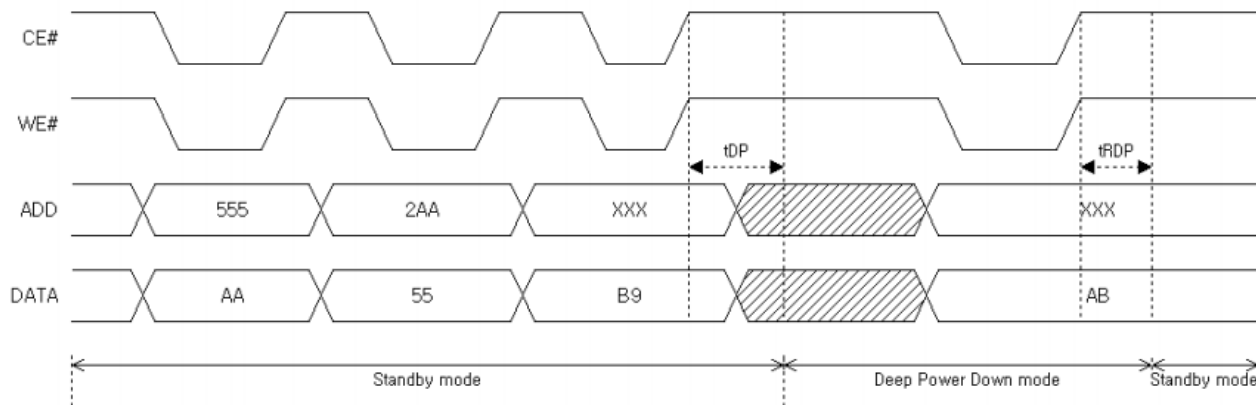


表 7-11 深度断电模式交流时序

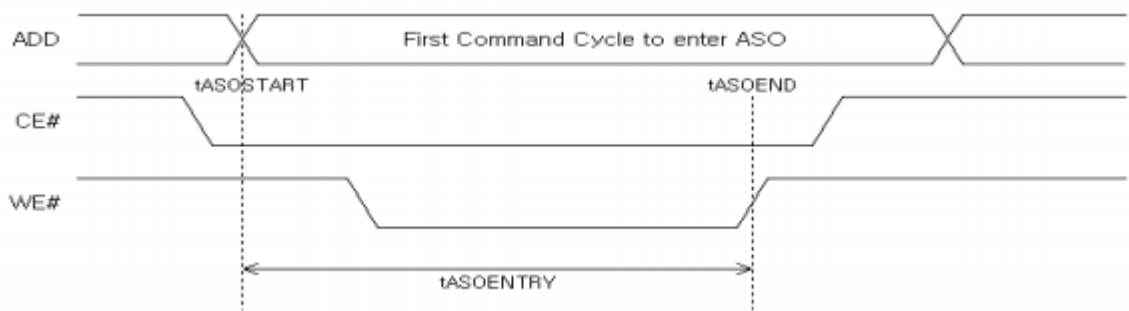
参数	描述	典型值	最大值	单位
TDP	WE#高到深度断电模式	100	200	μs
TRDP	WE#高到释放深度断电模式	10	20	μs

表 7-12 AS0 进入时序

参数	描述
$t_{ASO\,START}$	CE#下降沿或地址，任一个最后一次改变
$t_{ASO\,END}$	CE#上升沿或 WE#上升沿，任一个最后一次改变
$t_{ASO\,ENTRY}$	$25ns < t_{ASO\,ENTRY} < 50ns$ 或 $t_{ASO\,ENTRY} > 150ns$

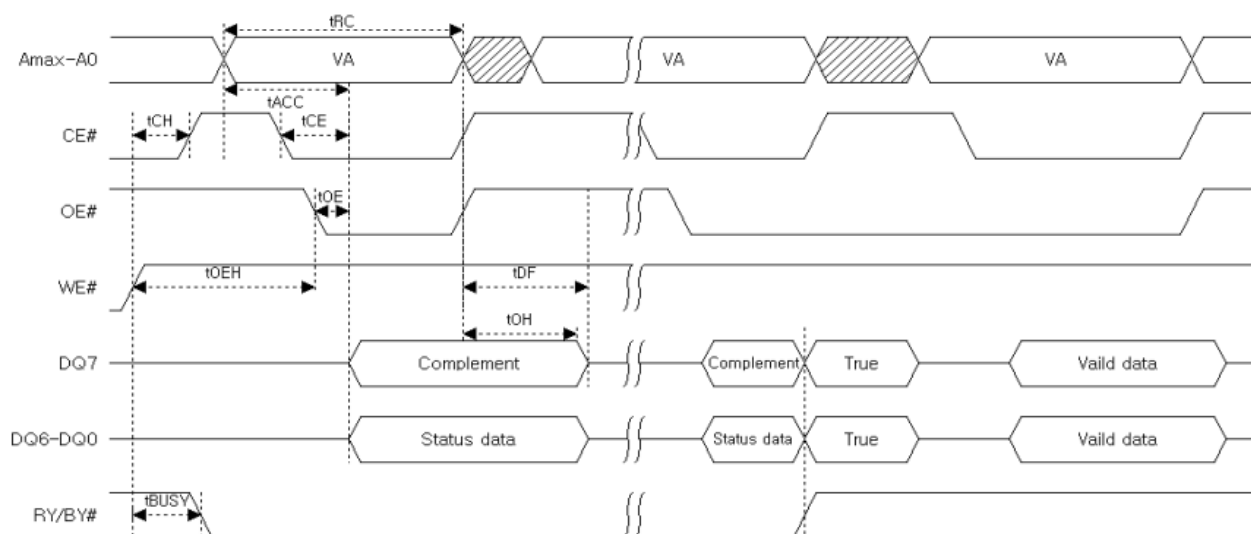
注 1：如果此时间达不到，在 AS0 退出后并且恢复正常过程前，立即执行下面步骤：从 32 字节排列页面中每 64 读一个字

图 7-21 AS0 进入时序图



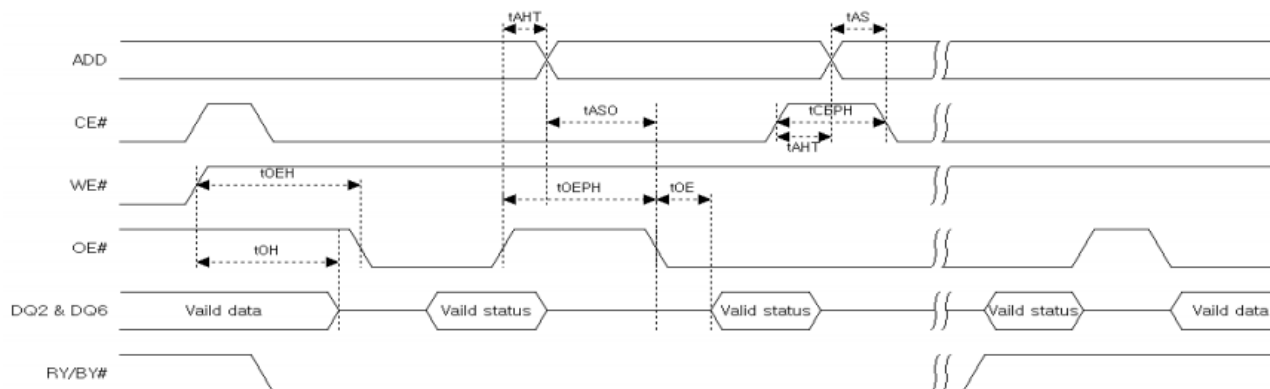
注 1：应用于任何 AS0 进入命令

图 7-22 Data#轮询时序图（嵌入式算法期间）



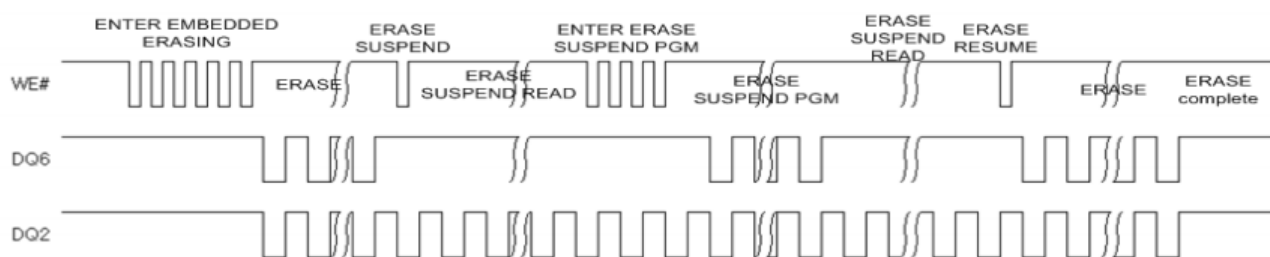
注 1: VA = 有效地址。插图显示了指令序列后的第一个状态周期、最后一个状态读取周期和阵列数据读取周期。

图 7-23 翻转位时序图（嵌入式算法期间）



注 1: 当设备忙时, 任何读地址 DQ6 将翻转。如果地址在活跃于擦除扇区, DQ2 将翻转

图 7-24 DQ2 与 DQ6 关系图



注 1: 只有在擦除挂起扇区内的地址读取时, DQ2 才会翻转。系统可以使用 OE#或 CE#来翻转 DQ2 和 DQ6。

7.4.3 轮流 CE#控制的写操作

表 7-13 轮流 CE#控制的写操作

序号	JEDEC 参数	Std 参数	描述		$V_{IO}=2.7V \sim V_{CC}$	$V_{IO}=1.65V \sim V_{CC}$	单位
1	tAVAV	tWC	写周期时间	最小	60		ns
2	tAVWL	tAS	地址建立时间	最小	0		ns
3		tASO	地址相对于 OE#低电平的建立时间（翻转位轮询期间）	最小	15		ns
4	tWLAX	tAH	地址保持时间	最小	45		ns
5		tAHT	地址相对于 CE#或 OE#高电平的保持时间（翻转位轮询期间）	最小	0		ns
6	tDVWH	tDS	数据建立时间	最小	30		ns
7	tWHDX	tDH	数据保持时间	最小	0		ns
8		tCEPH	CE#高电平脉冲宽度（翻转位轮询期间）	最小	20		ns
9		tOEPH	OE#高电平脉冲宽度（翻转位轮询期间）	最小	20		ns
10	tGHEK	tGHEL	写之前的读恢复时间（OE#高到 CE#低）	最小	0		ns
11	tWLEL	tWS	WE#建立时间	最小	0		ns
12	tELWH	tWH	WE#保持时间	最小	0		ns
13	tELWH	tCP	CE#脉冲宽度	最小	35		ns
14	tEHEL	tCPH	CE#脉冲高电平宽度	最小	30		ns

图 7-25 背对背（CE#）写操作时序图

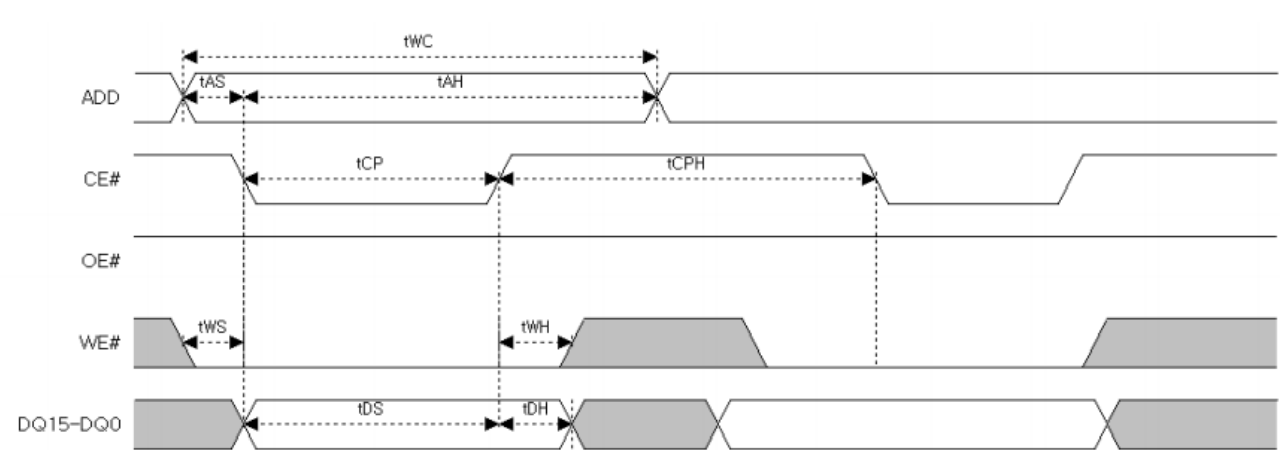
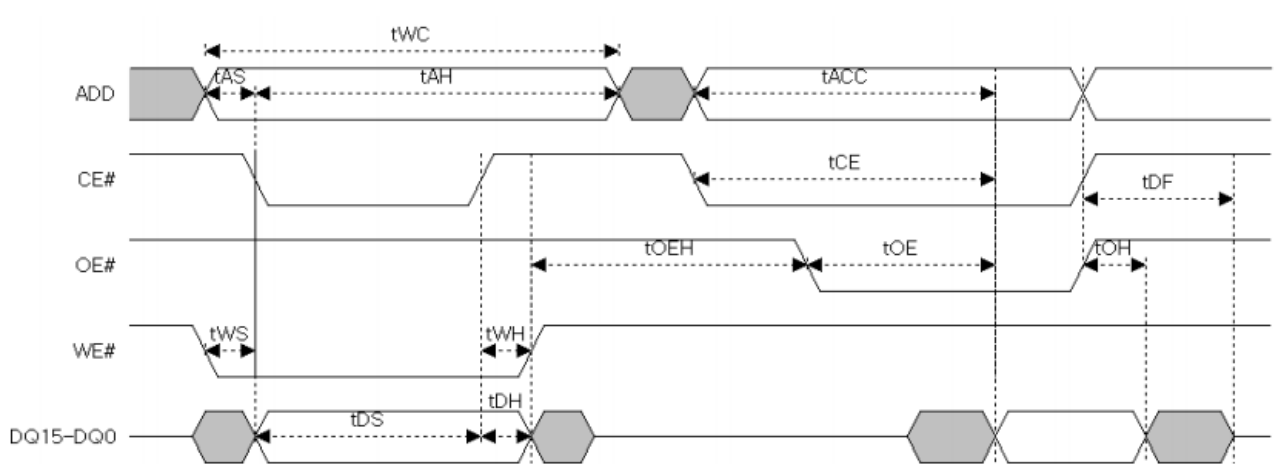


图 7-26 (CE#) 写到读操作时序图



7.4.4 电容特性

表 7-14 FBGA 封装连接电容

参数符号	参数描述	测试设备	典型值	最大值	单位
C_{IN}	输入电容	VIN=0	8	9	pF
C_{OUT}	输出电容	VOUT=0	5	7	pF
C_{IN2}	控制引脚电容	VIN=0	4	8	pF
RY/BY#	输出电容	VOUT=0	3	4	pF

注 1：测试条件：TA=25℃，f=1.0MHz

7.5 功能描述

7.5.1 地址空间映射

在该存储设备的地址范围中有几个隔开的地址：

■ 闪存阵列：主非易失性存储阵列用来存储可以被异步读操作随机访问的数据。

■ ID/CFI：一种存储空间用于厂家写入设备特性信息。这个区域包含设备地址 (ID) 和常用闪存接口 (CFI) 信息表。

■ 安全硅区域 (SSR)：一种一次性编程 (OTP) 非易失性存储阵列，用于厂家编写永久性数据，并且客户也可编写自己的永久性数据。

■ 锁寄存器：一种 OTP 非易失性字，用于确认 ASP 功能和锁 SSR。

■ 固定保护位 (PPB)：一种有一位用于每一个扇区的非易失性闪存阵列。当编程时，每一位保护相应的扇区相关擦除和编程。

■ PPB 锁：一种易失性的寄存器，用于失能或禁用 PPB 位的编程和擦除。

■ 密码：一种 OTP 非易失性阵列，用于存储 64 位密码。当使用密码保护模式进行扇区保护时，使能改变 PPB 锁位的状态。

■ 动态保护位 (DYB)：一种易失性阵列，其中一位用于每一个扇区。当设置时，每一位用于在编程和擦除时保护相应扇区。

■ 状态寄存器：一种易失性寄存器，用于展示嵌入式算法状态。

■ 数据轮询状态：一种易失性寄存器，像一种轮翻的、传统的软件兼容的方法来展示嵌入式算法的状态。

主闪存阵列是主要的也是默认的地址空间，但它可能随时会被其它地址空间覆盖。每一种替换的地址空间被称为地址空间覆盖 (ASO, Address Space Overlay)。

每一个 ASO 代替 (覆盖) 整个闪存设置地址范围。任何地址范围并不会被某一特定的 ASO 地址映射定义，是保留为将来所用。所有 ASO 地址映射之外的读访问返回非易失性 (未定义的) 数据。这些位置将响应驱动数据，但不管是 1 还是 0 来响应都没有定义。

在任何时间，本设备的闪存地址空间有四种操作模式，

- 读模式
- 数据轮询模式
- 状态寄存器模式 (SR)
- 地址空间代替模式 (ASO)

在读模式，整个闪存阵列可以被主机系统存储控制器直接访问。上电上设置设备到读模式，硬件复位后，命令复位之后，或嵌入式算法 (EA) 挂起之后，存储设置嵌入式算法控制器 (EAC)。在读模式可以接收读访问和命令写。当 EA 被挂起时读模式可以接受命令集。

在任何模式，状态寄存器读命令可能被执行来状态寄存器在设备地址空间的每一个字地址 ASO 显现。状态寄存器 ASO 模式，设备接口等待读访问，且任何写访问将被忽略。状态寄存器内容的下一个读访问，退出状态寄存器 ASO，并返回当前模式，这里状态寄存器读命令被接收。

在 EA 模式，仅编程/擦除挂起命令或状态寄存器读命令能用。所有其它命令都被忽略。因此，没有 ASO 能进入 EA 模式。

当 EA 被挂起，数据轮询 ASO 可见，直到设备已经将 EA 挂起。当 EA 被挂起时，数据轮询 ASO 是存在的，且闪存阵列数据可得到。当被挂起的 EA 恢复时，数据轮询 ASO 重来，直到 EA 再次被挂起或完成。当 EA 被完成时，数据轮询 ASO 存在，且设备继续当前模式 (从 EA 开始处)。

在 ASO 模式，进入了保留覆盖地址空间 (覆盖主闪存阵列地址映射)。在任何时间仅一个 ASO 可以被进入。发到设备的命令影响当前进入 ASO。每一个 ASO，仅固定命令是有效的。

下面的 ASOs 有非易失性数据，可以使 1 被编程到 0：

- 安全硅区域
- 锁寄存器
- 永久保护位 (PPB)
- 密码
- 仅 PPB ASO 有非易失性数据 0 可以被擦除到 1。

当一个非易失性 ASOs 进入，一个编程或擦除命令执行时，EA 在 ASO 上操作。当 EA 活跃时，ASO 不可读。当 EA 完成后，ASO 保持进入并可读。任何 ASOs 在 EA 操作期间，挂起和恢复命令被忽略。

7.5.1.1 闪存阵列

7.5.1.1.1 扇区地址

- 扇区架构
 - 总共 1024 个扇区
 - 每个扇区大小 128Kbyte

表 7-15 扇区地址划分

扇区 (64Kword/128Kbyte)	地址范围 (16bit)	备注
SA0	0000000h-000FFFFh	起启扇区地址
...	...	...
扇区 3	0030000h-003FFFFh	...
...	...	...
扇区 1020	3FC0000h-3FCFFFFh	...
...	...	...
扇区 1023	3FF0000h-3FFFFFFh	扇区地址结尾

表 7-16 页-缓冲区-扇区地址位

类型	X16		X8	
	数量	地址	数量	地址
一页地址	16	A3-A0	32	A3-AM1
一个写缓冲区地址	256	A7-A0	256	A6-AM1
所有页	4096	A15-A4	4096	A16-A4
所有写缓冲区	256	A15-A8	512	A15-A7
所有扇区	1027 (1Gb)	Amax-A16	1024 (1Gb)	Amax-A16

7.5.1.2 设备 ID 和 CFI (ID-CFI) AS0

系统有两种传统的方法来识别安装在系统中的闪存类型。一种传统上被称为自动选择，现在被称为设备标识 (ID)。另一种方法称为通用 Flash 接口 (CFI)。

对于 ID，一个命令用于启用地址空间覆盖，其中最多可以读取 16 个字(world)的位置，以获得 JEDEC 制造商标识 (ID)、设备 ID，以及一些闪存中的配置和保护状态信息。系统可以使用制造商和设备 ID 来选择与闪存设备一起使用的适当的驱动程序软件。

CFI 还使用一个命令来启用地址空间覆盖，其中可以读取关于闪存如何组织和操作的可扩展的标准信息表。用这种方法，驱动软件不需要考虑到每个可能的内存设备的细节。相反，驱动程序软件是以一种更通用的方式编写的，来处理许多不同的设备，但可以根据 CFI 表中的信息来调整驱动程序的行为。

传统上，这两个地址空间使用单独的命令，并且是单独的覆盖层。然而，这两个地址空间的映射是不重叠的，因此可以组合成一个单个的地址空间，并一起出现在一个覆盖。用于访问 (输入) 自动选择 (ID) 或 CFI 覆盖的传统命令都会导致当前合并的 ID-CFI 地址映射出现。

ID-CFI 地址映射显示在 ID-CFI 命令中使用的地址选择的扇区中，并覆盖 Flash 阵列数据。而 ID-CFI AS0 则输入所有其他扇区的内容尚未定义。

ID-CFI 地址映射从所选扇区的位置 0 开始。在 ID-CFI AS0 的最大定义地址以上到所选扇区的最大地址的位置具有未定义的数据。ID-CFI 输入命令使用与上一代内存相同的地址和数据值来访问 JEDEC 制造商 ID (自动选择) 和通用闪存接口 (CFI) 信息。AS0 进入时间要求见图 7-21 AS0 进入时序图。

表 7-17 ID-CFI 地址映射概览

字地址	字节地址	描述	读/写
(SA) + 0000h to 000Fh	(SA) + 0000h to 001Fh	设备 ID (传统自动选择值)	只读
(SA) + 0010h to 0079h	(SA) + 0020h to 00F2h	CFI 数据结构	只读
(SA) + 0080h to FFFFh	(SA) + 0100h to 1FFFFh	未定义	只读

注：查看完整值请参见表 7-19 ID (自动选择) 地址映射

7.5.1.3 设备 ID 和通用 Flash 接口 (ID-CFI) AS0 映射-自动选择模式

表 7-18 设备 ID 和通用 Flash 接口 (ID-CFI) AS0 映射-自动选择模式

字地址	字节地址	数据字段	#字节数	数据格式	实际数据示例	十六进制读出示例数据
(SA) + 0080h	(SA) + 100h	电子标记尺寸	1	十六进制	19	0013 h
(SA) + 0081h	(SA) + 102h	电子标记修订	1	十六进制	1	0001 h
(SA) + 0082h	(SA) + 104h	Fab Lot #	7	文本	EP87890	0045h, 0050h, 0038h, 0037 h, 0038h, 0039h, 0030h
(SA) + 0089h	(SA) + 112h	晶片#	1	十六进制	23	0017 h
(SA) + 008Ah	(SA) + 114h	晶圆X坐标	1	十六进制	10	000Ah
(SA) + 008Bh	(SA) + 116h	晶圆Y坐标	1	十六进制	15	000Fh
(SA) + 008Ch	(SA) + 118h	类别标号#	7	文本	EP87890	0045h, 0050h, 0038h, 0037 h, 0038h, 0039h, 0030h
(SA) + 0093h	(SA) + 126h	未来预留	13	N/A	N/A	未定义的
Fab Lot # + Wafer # + Die X坐标+ Die Y坐标为每个设备提供唯一的ID。						

7.5.1.3.1 设备 ID

联合电子器件工程委员会 (JEDEC) 标准 JEP106T 定义了兼容内存的制造商 ID。常见的行业用法定义了读取制造商的方法和格式 ID 和来自内存设备的设备特定 ID。制造商和设备 ID 信息主要用于编程设备，使设备与对应的设备自动匹配相应编程算法。华微在这个 32 字节的地址空间中添加了其他字段。

原始的行业格式的结构适用于任何内存数据总线宽度，例如×8、×16、×32。ID 代码值通常是字节宽，但位于总线宽度地址边界 增加设备地址输入将读取连续的字节、字或双字位置，其 ID 码总是位于数据总线的最不重要的字节位置。因为设备数据总线是字（16 位）宽的，所以每个代码字节都位于每个字位置的低半部分。原始的行业格式使高阶字节总是 0。华微已经修改了格式使在地址空间的某些单词中同时使用这两个字节。有关设备 ID 地址映射的详细描述，请参见表 7-19。

表 7-19 ID（自动选择）地址映射

描述	地址 (x16)	地址 (x8)	读数据
厂家 ID	(SA)+000h	(SA)+000h	0001h
设备 ID	(SA)+001h	(SA)+002h	227Eh
保护验证	(SA)+002h	(SA)+004h	扇区保护状态 (1=扇区被保护, 0=扇区未被保护)。这种保护状态仅展示当 ID-CFI ASO 时, 此扇区地址被选择。读其它扇区会返回不确定数据。为了读一个不同的扇区, 必须使用保护状态 ASO 退出命令, 然后再次用新扇区进入 ID-CFI ASO
指示位	(SA)+003h	(SA)+006h	DQ15-DQ08=1 (保留) DA7-厂家锁安全硅区域 1=锁 0=未锁 DQ6-客户锁安全硅区域 1=锁 0=未锁 DQ5=1 (保留) DQ4-WP#保护 0=最低地址扇区 1=最高地址扇区 DQ3-DQ0=1 (保留)
RFU	(SA)+004h	(SA)+008h	保留
	(SA)+005h	(SA)+00Ah	保留
	(SA)+006h	(SA)+00Ch	保留
	(SA)+007h	(SA)+00Eh	保留
	(SA)+008h	(SA)+0010h	保留
	(SA)+009h	(SA)+0012h	保留
	(SA)+00Ah	(SA)+0014h	保留
	(SA)+00Bh	(SA)+0016h	保留
较低软件位	(SA)+00Ch	(SA)+0018h	位 0: 状态寄存器支持 1=状态寄存器支持 0=状态寄存器不支持 位 1: DQ 翻转支持 1=DQ 位翻转支持

			0=DQ 位翻转不支持 位 3—2: 命令设置支持 11=保留 10=保留 01=压缩命令设置 00=经典命令设置
较高软件位	(SA)+00Dh	(SA)+001Ah	保留
设备 ID	(SA)+00Eh	(SA)+001Ch	2228h = 1Gb 2248h = 2Gb
设备 ID	(SA)+00Fh	(SA)+001Eh	2201h

7.5.1.3.2 通用闪存接口 CFI

JEDEC 公共闪存接口 (CFI) 规范 (JESD68.01) 定义了一个标准化的数据结构, 可以从闪存设备中读取, 这允许供应商指定的软件算法用于整个设备系列。数据结构包含系统配置的信息, 如各种电气和定时参数, 以及设备支持的特殊功能。然后, 软件支持可以独立于设备, 并对整个 Flash 设备系列进行前向和向后兼容。

系统可以在选定扇区内的地址读取 CFI 信息, 如设备 ID 和公共 Flash 接口 (ID-CFI) AS0 映射中显示的 CFI 信息。

与设备 ID 信息一样, CFI 信息的结构化适合于任何内存数据总线宽度, 例如×8、×16、×32。代码值总是字节宽 (8bit) 的, 但位于数据总线宽度的边界, 这样增加设备地址读取连续的字节、字, 或双字位置, 代码总是位于数据总线的最少的字节位置。华微因为数据总线是字宽的, 所以每个代码字节都位于每个字位置的低半部分, 并且高阶字节始终为 0。

AS0 的设备 ID 部分 (字位置 0h 至 0Fh) 提供设备的制造商 ID、设备 ID、扇区保护状态和基本功能集信息。

ID-CFI 位于 02h, 显示由 ID-CFI 进入命令中使用的扇区地址 (SA) 选择的扇区的扇区保护状态。读取多个扇区的保护状态必须退出 ID AS0 并使用新 SA 进入 ID AS0。读取位置 02h 的访问时间总是 t_{acc} , 并且该位置的读取需要 CE# 在读取之前变为高电平, 并且返回低电平以启动读取 (异步读取访问)。不支持在位置 02h 和其他 ID 位置之间读取页面模式。支持 02h 以外的 ID 位置之间的页面模式读取。

表 7-20 CFI 查询标识字符

字地址 (x16)	字节地址 (x8)	数据	描述
(SA)+0010h	(SA)+0020h	0051h	查询唯一 ASCII 字符” DRY”
(SA)+0011h	(SA)+0022h	0052h	
(SA)+0012h	(SA)+0024h	0059h	
(SA)+0013h	(SA)+0026h	0002h	主 OEM 命令设置
(SA)+0014h	(SA)+0028h	0000h	
(SA)+0015h	(SA)+002Ah	0040h	主扩展表地址
(SA)+0016h	(SA)+002Ch	0000h	
(SA)+0017h	(SA)+002Eh	0000h	替换 OEM 命令设置 (00h=不存在)
(SA)+0018h	(SA)+0030h	0000h	
(SA)+0019h	(SA)+0032h	0000h	替代 OEM 扩展表地址 (00h=不存在)
(SA)+001Ah	(SA)+0034h	0000h	

表 7-21 CFI 系统接口字符

字地址 (x16)	字节地址 (x8)	数据	描述
(SA)+001Bh	(SA)+0036h	0027h	V _{cc} 最小 (擦除/编程) (D7-D4:volts, D3-D0:100mV)
(SA)+001Ch	(SA)+0038h	0036h	V _{cc} 最大 (擦除/编程) (D7-D4:volts, D3-D0:100mV)
(SA)+001Dh	(SA)+003Ah	0085h	ACC 最小电压 (00h=无 ACC 引脚接出)
(SA)+001Eh	(SA)+003Ch	0095h	ACC 最大电压 (00h=无 ACC 引脚接出)
(SA)+001Fh	(SA)+003Eh	0008h	每单字写 2 ^N μs 典型超时
(SA)+0020h	(SA)+0040h	0009h	最大多字节编程典型超时, 2 ^N μs (00h=不支持)
(SA)+0021h	(SA)+0042h	0008h	每间隔块擦除 2 ^N ms 典型超时
(SA)+0022h	(SA)+0044h	0012h (1Gb)	全芯片 2 ^N ms 典型超时 (00h=不支持)
(SA)+0023h	(SA)+0046h	0001h	单字写 2 ^N 次典型最大超时
(SA)+0024h	(SA)+0048h	0002h	缓冲区写 2 ^N 次典型最大超时
(SA)+0025h	(SA)+004Ah	0003h	间隔块擦除 2 ^N 次典型最大超时
(SA)+0026h	(SA)+004Ch	0003h	全芯片擦除 2 ^N 次典型最大超时 (00h=不支持)

表 7-22 CFI 设备几何定义

字地址(x16)	字节地址(x8)	数据	描述
(SA)+0027h	(SA)+004Eh	001Bh (1Gb) 001 Ch (2 Gb)	设备容量=2 ⁿ 字节
(SA)+0028h	(SA)+0050h	0002h	闪存设备接口描述 0=仅 x8, 1=仅 x16, 2=x8/x16 均可
(SA)+0029h	(SA)+0052h	0000h	
(SA)+002Ah	(SA)+0054h	0009h	多字节写的字节最大值=2 ⁿ (00=不支持)
(SA)+002Bh	(SA)+0056h	0000h	
(SA)+002Ch	(SA)+0058h	0001h	设备里擦除块区域编号, 1=统一设备, 2=启动设备
(SA)+002Dh	(SA)+005Ah	00XXh	擦除块区域 1 信息 (参见 JEDEC JESD68-01 或 JEP137 特性) 00FFh, 0003h, 0000h, 0002h=1Gb
(SA)+002Eh	(SA)+005Ch	000Xh	
(SA)+002Fh	(SA)+005Eh	0000h	
(SA)+0030h	(SA)+0060h	000Xh	
(SA)+0031h	(SA)+0062h	0000h	擦除块区域 2 信号 (参见 CFI 版本 100)
(SA)+0032h	(SA)+0064h	0000h	
(SA)+0033h	(SA)+0066h	0000h	
(SA)+0034h	(SA)+0068h	0000h	
(SA)+0035h	(SA)+006Ah	0000h	擦除块区域 3 信号 (参见 CFI 版本 100)
(SA)+0036h	(SA)+006Ch	0000h	
(SA)+0037h	(SA)+006Eh	0000h	
(SA)+0038h	(SA)+0070h	0000h	
(SA)+0039h	(SA)+0072h	0000h	擦除块区域 4 信号 (参见 CFI 版本 100)
(SA)+003Ah	(SA)+0074h	0000h	
(SA)+003Bh	(SA)+0076h	0000h	
(SA)+003Ch	(SA)+0078h	0000h	
(SA)+003Dh	(SA)+007Ah	FFFFh	保留
(SA)+003Eh	(SA)+007Ch	FFFFh	保留
(SA)+003Fh	(SA)+007Eh	FFFFh	保留

表 7-23 CFI 主要应商特别的扩展查询

字地址(x16)	字节地址(x8)	数据	描述
(SA)+0040h	(SA)+0080h	0050h	查询唯一的 ASCII 字符“PRI”
(SA)+0041h	(SA)+0082h	0052h	
(SA)+0042h	(SA)+0084h	0049h	
(SA)+0043h	(SA)+0086h	0031h	主版本号, ASCII
(SA)+0044h	(SA)+0088h	0035h	副版本号, ASCII
(SA)+0045h	(SA)+008Ah	001Ch	地址敏感解锁 (位 1-0) 00b=需要 01b=不需要 过程技术 (位 5-2) 待定
(SA)+0046h	(SA)+008Ch	0002h	擦除挂起 0=未挂起 1=只读 2=读和写
(SA)+0047h	(SA)+008Eh	0001h	扇区保护 00=不支持 X=最小组里扇区数
(SA)+0048h	(SA)+0090h	0000h	暂时的扇区未保护 00=不支持 01=支持
(SA)+0049h	(SA)+0092h	0008h	扇区保护/未保护组合 04=高电压方法 05=软件命令锁方法 08=高级扇区保护方法
(SA)+004Ah	(SA)+0094h	0000h	同时操作 00=不支持 X=空白数
(SA)+004Bh	(SA)+0096h	0000h	突发模式类型 00=不支持 01=支持
(SA)+004Ch	(SA)+0098h	0003h	页模式类型 00=不支持 01=4 字页 02=8 字页

			03=16 字页
(SA)+004Dh	(SA)+009Ah	0085h	ACC（加速）支持最下限 00=不支持 D7-D4: Volt D3-D0: 100mV
(SA)+004Eh	(SA)+009Ch	0095h	ACC（加速）支持最上限 00=不支持 D7-D4: Volt D3-D0: 100mV
(SA)+004Fh	(SA)+009Eh	0004h(下) 0005h(上)	WP#保护 00h=无 WP 保护闪存设备（无启动） 01h=在顶部 8 个 8Kb 扇区,底部 WP 保护(双启动) 02h=用 WP 保护下面启动设备（下启动） 03h=用 WP 保护上启动设备（上启动） 04h=统一, 下 WP 保护（统一下启动） 05h=统一, 上 WP 保护（统一上启动） 06h=WP 保护所有扇区 07h=统一, 上下 WP 保护
(SA)+0050h	(SA)+00A0h	0001h	编程挂起 00=不支持 01=支持
(SA)+0051h	(SA)+00A2h	0001h	解锁旁路 00=不支持 01=支持
(SA)+0052h	(SA)+00A4h	0009h	安全硅扇区（客户 OTP 区域）大小 2^N （字节）
(SA)+0053h	(SA)+00A6h	0008h	软件功能 bit0:状态寄存器翻转（1=支持, 0=不支持） bit1:DQ 翻转（1=支持, 0=不支持） bit2:新编程挂起/恢复命令（1=支持, 0=不支持） bit3:字编程（1=支持, 0=不支持） bit4:位域编程（1=支持, 0=不支持） bit5:自动探编程（1=支持, 0=不支持） bit6:RFU bit7:每线多次写（1=支持, 0=不支持）

(SA)+0054h	(SA)+00A8h	0005h	页大小=2 ^N 字节
(SA)+0055h	(SA)+00AAh	0006h	擦除挂起超时最大<2 ^N (μ s)
(SA)+0056h	(SA)+00ACh	0006h	编程挂起超时最大<2 ^N (μ s)
(SA)+0057h to (SA)+0077h	(SA)+00AEh SA)+00D6h	FFFFh	保留
(SA)+0078h	(SA)+00F0h	0006h	嵌入式硬件复位超时最大<2 ^N (μ s)，用复位引脚复位
(SA)+0079h	(SA)+00F2h	0009h	非嵌入式硬件复位超时最大<2 ^N (μ s)，上电复位

7.5.1.4 数据轮询状态 AS0

数据轮询状态 AS0 包含一个易失性存储器的字，指示 EA 的进展。数据轮询状态 AS0 将在任何命令序列的最后一个写周期之后立即输入启动初始化 EA。启动初始化 EA 的命令包括：

- Word 程序
- 程序缓冲区到闪存
- 芯片擦除
- 扇区擦除
- 擦除恢复/编程恢复
- 编程恢复增强方法
- 空白检查
- 锁定寄存器编程
- 密码编程
- PPB 编程
- 所有 PPB 擦除

工程说明：重置和写缓冲区中止重置命令需要非常短的时间来执行，因此这些命令不支持数据轮询。将出现数据轮询状态字在设备地址空间中的字位置。当 EA 完成时，数据轮询状态 AS0 将退出，设备地址空间返回到 EA 启动时的地址映射模式。

7.5.1.5 安全硅区域 AS0

安全硅区域（SSR）提供了一个额外的闪存区域，可以一次编程和永久保护免受进一步的变化，即它是一个一次性程序（OTP）区域。SSR 为 长度为 512 字节。

在安全硅入口命令期间提供的扇区地址选择由安全硅区域地址映射覆盖的闪存阵列扇区。AS0 进入时序需求见图 7-21 明确要求。SSR 从选定扇区中的位置 0 开始被覆盖。建议使用扇区 0 地址。当输入 SSR AS0 的内容时 所有其他扇区都未定义。在 SSR AS0 的最大定义地址以上到所选扇区的最大地址的位置具有未定义的数据。

表 7-24 安全硅区域

字地址范围	字节地址范围	内容	大小
(SA) + 0000h ~ 0007h	(SA) + 0000h ~ 000Fh	电子 序列号 (ESN) 或由客户决定	16 字节
(SA) + 0008h ~ 00FFh	(SA) + 0010h ~ 01FFh	由客户决定	496 字节
(SA) + 0100h ~ 01FFh	(SA) + 0200h ~ 03FFh	未定义的	512 字节
(SA) + 0200h ~ FFFFh	(SA) + 0400h ~ 1FFFFh	未定义的	127 k 字节

7.5.1.6 ECC 状态 AS0

系统可以通过在读取模式期间发出 ECC 状态输入命令序列来访问 ECC 状态 AS0。ECC 状态 AS0 在读取状态选中的页面时提供了单个位错误修正。AS0 进入时序要求见图 7-21。

ECC 状态 AS0 允许进行以下活动：

- 请读取所选页面的 ECC 状态信息。
- AS0 退出。

7.5.1.6.1 ECC 状态

ECC 状态 AS0 的内容表明，对于选定的 ECC 页面，ECC 保护是否已经纠正了 8 位错误纠正码或 ECC 页面中的 16 个字数据中的错误。

重要提示：此设备不支持 ECC 模式。但命令被接受，输出被 FFF9h 固定。

表 7-25 ECC 状态字-高字节位

位	15	14		12	11	10	9	8
名字	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
值	X	X	X	X	X	X	X	X

表 7-26 ECC 状态字-低字节位

位	7	6	5	4	3	2	1	0
名字	RFU	RFU	RFU	RFU	RFU	在 8 位中纠正了单比特错误纠错码	已纠正单比特错误 16 字数据	RFU
值	X	X	X	X	X	0 = 未纠正错误	0 = 未纠正错误	X

7.5.1.7 扇区保护控制

7.5.1.7.1 锁寄存器 AS0

锁寄存器 AS0 包含一个 OTP 内存的一个字。当输入 AS0 时，锁寄存器将出现在设备地址空间中的所有字的位置。AS0 进入的时序要求见图 7-21。但是，建议仅在设备地址空间的位置 0 处读取或编程锁寄存器，以便将来兼容。

7.5.1.7.2 永久性保护位（PPB）ASO

PPB ASO 为设备中的每个扇区包含一个位的闪存阵列。当输入 PPB ASO 时，一个扇区的 PPB 位会出现在扇区每个地址的最低位（LSB）部份。ASO 进入时序要求见图 7-21。读取扇区中的任何地址都将显示数据，其中 LSB 指示该扇区的非挥发性保护状态。然而，我们建议只在扇区的地址 0 处读取或编程 PPB，以便将来兼容。如果值为 0 的位，则扇区禁止编程和擦除操作。如果该位数为 1 该位不受 PPB 的保护。该扇区可能会受到 ASP 的其他特性的保护

系统可以通过在读取模式期间发出 PPB 进入命令序列来访问 PPB ASO。该进入命令不使用来自进入命令的扇区地址。出现扇区的 PPB 位在扇区中所有字位置的位 0 中。PPB ASO 允许以下活动：

- 读取扇区中任何字的位 0 中扇区的 PPB 保护状态。
- 使用修改的字编程命令对 PPB 位进行编程。
- 用 PPB 擦除命令擦除所有 PPB 位。
- ASO 退出使用传统命令集退出命令来实现向后软件兼容性。
- ASO 退出使用所有 ASO 的通用退出命令——一致退出方法的替代方案。

7.5.1.7.3 PPB 锁 ASO

PPB 锁定 ASO 包含一次易失性存储器。比特控制 PPB ASO 中的比特是否可以被编程或擦除。如果位为 0，则 PPB ASO 禁止编程和擦除操作。如果该位为 1，则 PPB ASO 不受保护。当输入 PPB 锁定 ASO 时，PPB 锁定位将出现在设备地址中的每个地址的最低位（LSB）。ASO 进入时序要求见图 7-21。但是，建议只在设备的地址 0 处读取或编程 PPB 锁定，以便将来兼容。

7.5.1.7.4 密码 ASO

密码 ASO 包含四个字的 OTP 内存。输入 ASO 后，密码将出现在设备地址空间中的地址 0 开始。ASO 进入时序要求见图 7-21。第四个字之前的所有位置都是未定义的。

7.5.1.7.5 动态保护位（DYB）ASO

DYB (Dynamic Protection Bits) ASO 为设备中的每个扇区包含一个易失性存储器阵列。当输入 DYB ASO 时，一个扇区的 DYB 位将出现在每地址的最低位（LSB）中 在这个行业。ASO 进入时序要求见图 7-21。读取扇区中的任何地址都将显示数据，其中 LSB 指示该扇区的非易失性保护状态。然而，建议只在扇区的地址 0 处读取、设置或清除 DYB，以便将来兼容。如果位为 0，则扇区禁止编程和擦除操作。如果位为 1，该扇区不受 DYB 的保护。该扇区可能会受到 ASP 的其他特性的保护。

7.5.2 数据保护

该设备提供了几个功能，以防止通过硬件手段恶意或意外地修改任何扇区。

7.5.2.1 设备保护方法

7.5.2.1.1 上电写入禁止

在开机重置（POR）期间，重置#、CE#、WE#和 OE#将被忽略。在 POR 期间，设备无法被选择，将不接受 WE#上升沿的命令，也不驱动输出。在 POR 期间，主机接口控制器（HIC）和嵌入式算法控制器（EAC）被

重置为备用状态，准备读取阵列数据。CE#或 OE#必须在 POR (t_{VCS}) 结束之前转到 V_{IH} 。

在 POR 结束时，设备条件为：

- 加载所有内部配置信息，
- 该设备正处于读取模式，
- 状态寄存器处于默认值，
- DYB ASO 中的所有位都被设置为不保护所有扇区，
- 写入缓冲区中加载了所有的 1，
- EAC 处于旁路状态。

7.5.2.1.2 低 VCC 写抑制器

当 V_{CC} 小于 V_{LKO} 时，HIC 不接受任何写入周期，EAC 会重置。这可以保护 V_{CC} 通电和断电期间的数据。系统必须向计算机提供适当的信号控制引脚，以防止当 V_{CC} 大于 V_{LKO} 时无意写入。

7.5.2.1.3 带硬件数据保护的断电

当核心电源 (V_{CC}) 下降到锁定电压 (V_{LKO}) 以下时，存储器被认为是断电的。当 V_{CC} 低于 V_{LKO} 时，保护整个存储器阵列免受编程或擦除操作。这确保在功率转换期间不会发生存储器内容的虚假改变。在电源切换到断电期间， V_{IO} 应保持小于或等于 V_{CC} 。

如果 V_{CC} 低于 V_{RST} (Min)，然后返回高于 V_{RST} (Min) 至 V_{CC} 最小值，则进入上电复位接口状态，并且 EAC 启动冷复位嵌入式算法。

7.5.2.2 命令保护

嵌入式算法通过将命令序列写入 EAC 命令内存来启动的。命令内存数组对主机系统不可读，并且没有 ASO。每个主机接口的写都是一个命令或命令序列的一部分写到设备。EAC 会检查每个写入传输中的地址和数据，以确定该写入是否是规定命令序列的一部分。当一个规定命令序列完成后，EAC 将启动适当的 EA。

写入不正确的地址或数据值，或以不正确的顺序写入它们，通常会导致 EAC 返回到其待机状态。然而，这种不正确的命令序列可能会失效。如果设备处于未知状态，在这种情况下，系统必须写入重位命令，或者可能通过驱动 RESET# 信号低来提供硬件复位，以将 EAC 返回到旁路待机状态，准备好进行随机阅读。

在每个写入中提供的地址可能包含一个位搭配，用于帮助识别写入作为对设备的命令。该地址的上部也可以选择将会执行该命令操作所在的扇区地址。扇区地址 (SA) 包括 AMAX 到 A16 闪存地址位 (系统字节地址信号 AMAX 到 a17)。一个命令位模式位于 A10 中 到 A0 闪存地址位 (系统字节地址信号 a11 到 a1)。

每个写入中的数据可以是：用于帮助标识写入为命令的位搭配，标识要执行的命令操作的代码，或提供执行操作所需的信息。

7.5.2.3 安全硅区 (OTP)

安全硅区域 (SSR) 提供了一个额外的闪存区域，可以一次编程和永久保护免受进一步的变化，即它是一个一次性程序 (OTP) 区域。SSR 长度为 512 字节。

7.5.2.4 扇区保护方法

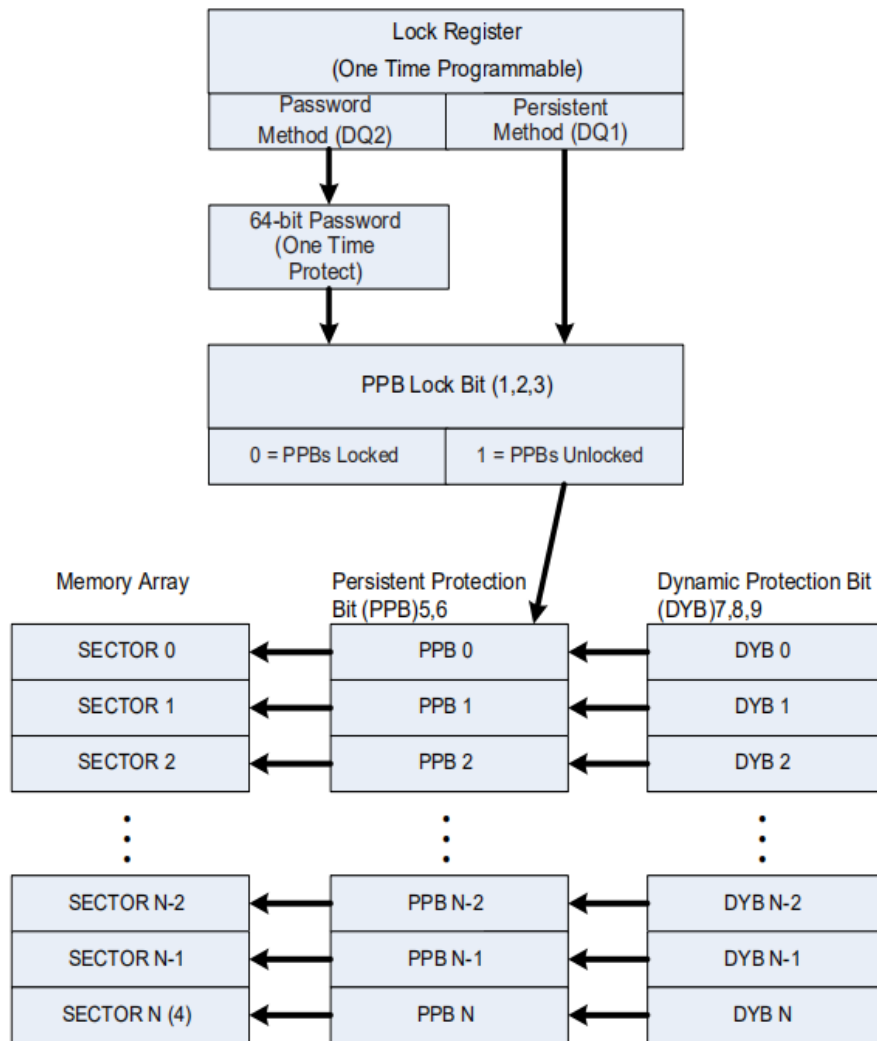
7.5.2.4.1 写保护信号

如果 $WP\# = V_{IL}$ ，最低或最高地址扇区受保护从编程或擦除操作独立于任何其他 ASP 配置。它是最低还是最高扇区取决于所选的设备订购选项（型号）。如果 $WP\# = V_{IH}$ ，最低或最高的地址扇区不受 $WP\#$ 信号的保护，但它可能是受 ASP 配置的其他方面的保护。 $WP\#$ 有一个内部上拉按钮；当未连接时， $WP\#$ 位于 V_{IH} 。

7.5.2.4.2 ASP

高级扇区保护（ASP）是一套独立的硬件和软件方法，用于在任何或所有扇区单独禁用或启用编程或擦除操作。本节描述了保护存储在内存阵列中的数据的各种方法。这些方法的概述如图 7-27 所示。

图 7-27 高级扇区保护概述



注：

1. 位是易失性的，重置时默认为“1”（如果处于密码模式，则默认为“0”）。
2. 编程为“0”会将所有 ppb 锁定到其当前状态。
3. 一旦被编程为“0”，就需要硬件重置才能解锁或应用密码。
4. N=最高地址扇区。
5. 0 个=扇区受到保护，1 个=扇区未受保护。
6. ppb 单独编程，但集体清除
7. 0 个=扇区受到保护，1 个=扇区未受保护。
8. 仅对应的 ppb=“1”保护有效（未受保护）。

9. 易失位：默认为用户在通电时选择（参见订购选项）。

每个主闪存阵列扇区都有一个非易失性（PPB）和与之相关的易失性（DYB）保护位。当任一位为 0 时，扇区不受程序和擦除操作的影响。

当 PPB 锁定位为 0 时，PPB 位不受程序保护和擦除的影响。有两种方法来管理 PPB 锁位的状态，持久性保护和密码保护。

持久性保护方法在 POR 或硬件重置期间将 PPB 锁设置为 1，以便 PPB 位不受设备复位的保护。有一个命令来清除 PPB 锁定位到 0 来保护 PPB 位。在持久保护方法中没有任何命令来设置 PPB 锁定位，因此 PPB 锁定位将保持在 0，直到下一次关机或硬件复位。永久保护方法允许引导代码选择通过编程或擦除 PPB 来更改扇区保护，然后保护 PPB 不受其余正常系统操作数的进一步更改，通过清除 PPB 锁定位。这有时被称为引导代码控制的扇区保护。

密码方法在 POR 或硬件重置期间清除 PPB 锁定位为 0，以保护 PPB。64 位密码可以为永久编程和隐藏的密码方法。命令可以是用于提供一个密码，以进行与隐藏密码的比较。如果密码与 PPB 匹配，锁定位设置为 1 以取消 PPB 保护。可以使用一个命令将 PPB 锁定位清除为 0。

PPB 锁管理方法的选择是通过在锁寄存器中编程 OTP 位，从而永久地选择所使用的方法。

锁寄存器还包含 OTP 位，用于保护 SSR。

PPB 位被移除，以便所有主闪存阵列扇区在从华微发货时不受保护。根据订单，安全硅区域可以被工厂保护或不受保护选项（型号）订购。

7.5.2.4.3 PPB 锁

永久保护位锁是一个用于保护所有 PPB 位的易失性位。当清除为 0 时，它锁定所有 PPB，当设置为 1 时，它允许更改 PPB。只有一个 PPB 锁定每个设备的位。

PPB 锁定命令用于将该位清除为 0。只有在所有 PPB 配置为所需设置后，PPB 锁定位才能清除为 0。

在永久保护模式下，在 POR 或硬件重置期间，PPB 锁被设置为 1。当清除时，没有软件命令序列可以设置 PPB 锁定，只能设置另一个硬件重位或上电等 PPB 锁止位。

在密码保护模式下，在 POR 或硬件重置期间，PPB 锁被清除为 0。PPB 锁只能通过密码解锁命令顺序设置为 1。PPB 锁可以通过他的 PPB 锁定位清除命令进行清除。

7.5.2.4.4 永久性保护位（PPB）

永久性保护位（PPB）位于一个单独的非易失性闪存阵列中。其中一个 PPB 位被分配给每个扇区。当 PPB 为 0 时，其相关扇区不受编程和擦除操作的保护。PPB 是单独编程的，但必须作为一个组擦除，类似于在主阵列中编程的方式，但整个扇区必须在擦除同一时间擦除前的预编程和验证工作由 EAC 负责处理。

编程一个 PPB 位需要典型的字编程时间。在 PPB 位编程操作或 PPB 位擦除期间，数据轮询状态 DQ6 切换位 1 将翻转，直到操作完。清除所有的 PPBs 要求典型的扇区擦除时间。

如果 PPB 锁定为 0，则 PPB 编程或擦除命令不会 PPB 超时而编程或擦除。

当进入 PPB AS0 时，给扇区的一个 PPB 保护状态能被通过 PPB 状态读命令验证。

7.5.2.4.5 动态保护位（DYB）

动态保护位对于每个扇区都是不稳定的和唯一的，可以单独修改。DYBs 只控制那些 PPB 擦除的部分的保护。通过发布 DYB 设置或清除命令序列，DYB 设置为 0 或清除为 1，如果扇区的 PPB 为 1，则每个扇区分别处于保护或非保护状态。这个特性允许软件使用对容易地保护部门不受无意的更改，但并不妨碍在需要更改时容易删除保护。

DYB 可以根据需要经常设置为 0 或清除为 1。

7.5.2.4.6 扇区保护状态概述

每个扇区都可以处于以下保护状态之一：

- 解锁-该扇区是不受保护的，并且可以通过一个简单的命令来改变保护。在电源循环或硬件重置后，保护状态默认为无保护。
- 动态锁定-可以保护一个扇区，并且可以通过一个简单的命令来更改保护。保护状态不会在电源循环或硬件重置期间进行保存。
- 持续锁定-一个扇区受到保护，只有在 PPB 锁定位设置为 1 时才能更改保护。保护状态为非易失性，并在电源循环或硬件重置期间保存。更改保护状态需要编程或擦除 PPB 位。

表 7-27 扇区保护状态

保护位值			扇区的状态
PPB锁	PPB	DYB	
1	1	1	未保护- PPB和DYB是可变的
1	1	0	保护- PPB和DYB是可变的
1	0	1	保护- PPB和DYB是可变的
1	0	0	保护- PPB和DYB是可变的
0	1	1	无保护- PPB不变，DYB可变
0	1	0	保护- PPB不可变，DYB可变
0	0	1	保护- PPB不可变，DYB可变
0	0	0	保护- PPB不可变，DYB可变

7.5.2.4.7 锁寄存器

锁寄存器保存非易失性 OTP 位，用于控制 SSR 的保护，并确定 PPB 锁位管理方法（保护模式）。

表 7-28 锁寄存器

位	默认值	名字
15-9	1	保留
8	0	保留
7	X	保留
6	1	保留
5	1	保留
4	1	保留
3	1	保留
2	1	密码保护模式锁定位
1	1	持久保护模式锁定位
0	0	SSR区域锁定位

安全硅区域（SSR）保护位必须谨慎使用，因为一旦锁定，就没有任何程序可用于解锁安全硅区域的受保护部分，也没有任何受保护的安全硅区存储器空间中的位可以以任何方式进行修改。一旦安全硅区受到保护，任何在该地区编程的尝试都将失败，指示被编程的区域受到保护的状态。SSR 区域指示器位位于锁定寄存器的 bit0 处。

从工厂出厂时，所有设备默认为持久保护方法，所有扇区都不受保护。然后，设备程序员或主机系统可以选择哪个扇区保护方法。编程以下两个一次性可编程的非易失性位，将此部份永久锁定在该模式下：

- 永久保护模式锁定位 (DQ1)
- 密码保护模式锁定位 (DQ2)

如果选择两个锁定位同时进行编程，则该操作将中止。一旦密码模式锁定位被编程，持久模式锁定位将被永久禁用，不再允许更改保护方案。同样地，如果持久模式锁定位被编程，则密码模式将被永久禁用。

如果要选择密码模式，则必须在设置相应的锁存寄存器位之前先对密码进行编程。设置密码保护模式锁定位是可编程的，这是一个上电循环、硬件复位，或 PPB 锁定位设置命令需要设置 PPB 锁定位为 0，以保护 PPB 阵列。

锁寄存器的编程时间与典型的字编程时间相同。在锁定寄存器编程 EA，数据轮询状态 DQ6 切换位，我将切换直到编程已完成。系统还可以通过读取状态寄存器来确定锁寄存器编程的状态。有关这些状态位的信息，请参阅状态寄存器。

用户不需要同时编写 DQ2 或 DQ1 程序。这允许用户在选择设备保护方案之前或之后锁定 SSR。当编程锁定位时，保留位必须为 1（已被屏蔽）。

7.5.2.4.8 永久保护模式

永久性保护方法在 POR 或硬件重置期间将 PPB 锁设置为 1，以便 PPB 位不受设备复位的保护。有一个命令来清除 PPB 锁定位到 0 来保护 PPB。在永久保护方法中没有命令将 PPB 锁位设置为 1，因此 PPB 锁位将保持为 0，直到下一次关机或硬件复位。

7.5.2.4.9 密码保护模式

PPB 密码保护模式允许比永久性扇区保护模式更高的安全级别，因为它需要一个 64 位的密码来设置 PPB 锁。除了此密码之外，要求上电复位后，PPB 锁清除为 0，以确保上电时的保护。通过输入整个密码设置 PPB 锁定到 1，成功执行密码解锁命令，允许对扇区 PPB 进行修改。

密码保护注意：

- 密码程序命令只能编程 0。
- 从华微出库时，密码全部为 1。它位于自己的内存空间中，可以通过使用密码程序和密码读取命令进行访问。必须输入确切的密码，才能进入解锁功能。
- 所有 64 位密码组合均作为密码有效。
- 一旦密码被编程和验证，就必须设置密码模式锁定位，以防止读取或修改密码。
- 密码模式锁定位，一旦被编程好，就可以防止在数据总线上读取 64 位密码和进一步的密码编程。所有进一步的编程和读取命令的密码区域是禁用（数据读取为 1），这些命令将被忽略。在密码保护模式锁定位被编程后，无法验证密码是什么。密码验证仅限在选择密码保护模式之前允许。
- 密码模式锁定位不可擦除。

- 必须输入确切的密码，才能进入解锁功能。
- 地址可以以任何顺序加载，但成功匹配需要所有 4 个字。
- 在加载密码地址/数据时，将比较扇区地址和 Word 行地址。如果扇区地址不匹配，则将在该写周期结束时报告该错误。状态寄存器将返回到就绪状态，编程状态位设置为 1，写缓冲区中止状态位设置为 1，表示编程失败操作。它无法更改 PPB 锁位的状态，因为它仍然受到缺乏有效密码的保护。数据轮询状态将保持有效状态，DQ7 设置为在密码解锁命令的最后一个字/字节中的 DQ7 位的补码，然后 DQ6 翻转，RY/BY#将保持较低。
- 在发出编程缓冲区到 Flash 命令后，将比较特定的地址和数据。如果它们与内部设置的值不匹配，则状态寄存器将返回到已就绪，编程状态位设置为 1，将程序状态寄存器位设置为 1，表示编程操作失败。更改 PPB 锁位的状态是失败的，因为它是缺乏有效密码的保护。数据轮询状态将保持活动状态，将 DQ7 设置为密码解锁命令的最后一个字/字节中的 DQ7 位的补充，以及 DQ6 切换 RY/BY#将保持较低。
- 在向该设备提供有效的 64 位密码后，该设备需要大约 100 μ s 来设置 PPB 锁定。
- 密码解锁命令不能超过每 100 μ s $\pm$ 20 μ s 接受一次。这使得一个黑客需要花费不合理的很长时间（5800 万年）来运行所有的试图正确匹配密码 64 位组合。EA 状态检查方法可用于确定 EAC 何时准备好接受一个新的密码命令。
- 64 位密码应该首先输入高位字节数据，以解锁 x8 模式下的密码模式保护。
- 如果在设置密码模式锁定位后密码丢失，则无法清除 PPB 锁定。

7.5.3 读取操作

7.5.3.1 禁用输出的读取

当 CE#信号被断言为低时，主机系统存储器控制器开始读取或写入数据传输。通常在数据传输开始时有一个周期，此时 CE#为低，地址为有效，OE#高，WE#高。在该状态期间，假设读取访问，并且在数据输出保持在高阻抗时开始随机读取过程。如果 OE#信号变为低电平，则接口转换到随机读取状态，数据输出被主动驱动。如果 WE#信号被断言为低电平，则接口转换到写入状态。注意，OE#和 WE#永远不应该同时低电平，以确保主机系统和内存之间没有数据总线争用。

7.5.3.2 随机（异步）读取

每个读取都可以访问存储器中的任意位置（随机访问）。每个随机访问都与 CE#或地址到有效数据（ t_{ACC} 或 t_{CE} ）的延迟相同。

当主机系统接口通过将 CE#驱动为低电平来选择存储器设备时，设备接口离开旁路状态。如果当 CE#变为低时 WE#为高，则开始随机读取访问。数据输出取决于地址映射模式和在读取访问开始时提供的地址。

当 CE#低、OE#低、WE#保持高、地址保持稳定并且满足异步访问时序时，数据出现在 DQ15-DQ0 上。地址访问时间（ t_{ACC} ）等于延迟从稳定的地址到有效的输出数据。芯片使能访问时间（ t_{CE} ）是输出端从稳定 CE#到有效数据的延迟。以便将读取的数据驱动到数据输出端在有效数据可用之前，OE#信号必须至少在输出使能时间（ t_{OE} ）为低电平。

在完成来自 CE#活动（ t_{CE} ）、地址稳定（ t_{ACC} ）或 OE#活动（ t_{OE} ）的随机访问时间时，无论哪一个发生在最近，数据输出将提供来自当前地址映射模式的有效读取数据。如果 CE#保持低，并且 AMAX 至 A4 地址信号中的任

何一个改变为新值，则开始新的随机读取访问。如果 CE#保持低且 OE#变为高，则接口到输出禁用状态的读取。如果 CE#保持低，OE#变为高，WE#变为低，则接口转换到写入状态。如果 CE#返回高电平，则接口转到旁路状态。背靠背访问，其中 CE#在访问之间保持低，需要地址改变来发起第二次访问。请参见 异步读取操作。

7.5.3.3 页读取

在随机读取访问完成之后，如果 CE#保持低，OE#保持低，则同样页内的新的访问开始。每个随机读取都会并行访问整个 32byte 的页面。同一页面内的后续读取具有更快的读取访问速度，比随机读取访问快得多。该页面是由高位的地址位 (AMAX-A4) 选择的，而该页面的特定字由低位的地址位 A3-A0 选择。A_{MAX} 至 A4 地址信号保持不变，只有 A3-A0 改变，在同一页面中选择一个不同的单字。当 OE#保持低时，这是一种异步访问，数据出现在 DQ15-DQ0 上。且当 CE#保持低，满足异步页面访问时间 (t_{PACC}) 时，如果 CE#变高，后续访问返回低，执行随机读取访问，需要时间 (t_{ACC} 或 t_{CE})。

7.5.4 嵌入式操作

7.5.4.1 嵌入式算法控制器 (EAC)

EAC 从主机系统接收用于编程和擦除闪存阵列的命令，并执行改变非易失性存储器状态所需的所有复杂操作。这释放了主机系统从任何需要管理编程和擦除进程。

有四个 EAC 操作类别：

- 待机 (读取模式)
- 地址空间切换
- 嵌入式算法
- 高级扇区保护 (ASP) 管理

7.5.4.1.1 EAC 旁路

在待机模式下，电流消耗大大降低。当没有命令被处理并且没有嵌入算法正在进行时，EAC 进入其待机模式。如果取消选择设备 (CE# = 高) 在嵌入式算法期间，器件仍然汲取有效电流，直到操作完成 (I_{CC3})。当主机接口和 EAC 处于旁路状态时，直流特性中的 I_{CC4} 表示旁路时的电流规格。

7.5.4.1.2 地址空间切换

写入特定地址和数据序列 (命令序列) 将存储器设备地址空间从主闪存阵列切换到地址空间覆盖 (AS0) 之一。

嵌入式算法对当前活动 (输入) AS0 中可见的信息进行操作。系统继续具有对 AS0 的访问权限，直到系统发出 AS0 退出命令，执行硬件复位，或直到设备断电。AS0 退出命令从 AS0 切换回主闪存阵列地址空间。输入特定 AS0 时接受的命令在命令定义表中的 AS0 进入和退出命令之间列出。有关所有命令序列的地址和数据要求，请参见命令摘要。

7.5.4.1.3 嵌入式算法 (EA)

改变存储器阵列中的非易失性数据需要称为嵌入式算法 (EA) 的复杂操作序列。算法完全由设备内部嵌入式算法控制器 (EAC) 管理。主要算法执行主阵列数据和 AS0 的编程和擦除。主机系统将命令代码写入闪

存设备地址空间。EAC 接收命令，执行完成命令的所有必要步骤，并在 EA 的进展期间提供状态信息。

7.5.4.2 编程和擦除总结

闪存数据位在称为扇区的大组中并行擦除。擦除操作将扇区中的每个数据位置于逻辑 1 状态(高)。闪存数据位可以是单独的从擦除 1 状态到编程逻辑 0 (低) 状态。数据位 0 不能编程回 1。随后的读取显示数据仍为 0。擦除操作只能将 0 转换为 1。用不同的 0 位不止一次地编程相同的字位置将导致先前数据和新数据的逻辑 AND 被编程。

编程和擦除操作的持续时间如嵌入式算法性能表所示。可以暂停编程和擦除操作。

- 可以暂停擦除操作以允许对另一扇区(不在擦除扇区中)进行编程或读取。在擦除挂起期间，不能启动其他擦除操作。
- 可以暂停编程操作以允许读取另一位置(不在被编程的行中)。
- 在挂起的编程操作期间，不得启动其他编程或擦除操作——在挂起的编程操作期间，编程或擦除命令将被忽略。
- 在中间编程操作或读取访问完成之后，可以恢复暂停的擦除或编程操作。恢复可以在挂起后的任何时间发生，假设设备是不是在执行另一个命令的过程中。
- 编程和擦除操作可以根据需要经常中断，但是为了使编程或擦除操作进行到完成，在恢复和下次挂起之间必须有一些时间段来暂停嵌入式算法性能，时间要大于或等于 t_{PRS} 或 t_{ERS} 的命令。
- 当嵌入式算法(EA)完成时，EAC 返回到启动 EA 的操作状态和地址空间(擦除挂起或 EAC 旁路)。

系统可以通过读取状态寄存器或使用数据轮询状态来确定程序或擦除操作的状态。有关这些状态位的信息，请参阅状态寄存器。参考到数据轮询状态以获取更多信息。

除了程序暂停和状态读取命令之外，在嵌入式程序算法期间写入设备的任何命令都被忽略。

除了擦除挂起和状态读取命令之外，在嵌入式擦除算法期间写入设备的任何命令都将被忽略。

硬件复位立即终止任何正在进行的编程/擦除操作，并在 t_{RPH} 时间之后返回到读取模式。一旦设备返回，应重新启动终止的操作到空闲状态，以确保数据完整性。

出于性能和可靠性原因，读取和编程是在完整的 32 字节页面内部完成的。

DC 特性中的 I_{CC3} 表示写入（嵌入式算法）操作的有效电流规格。

7.5.4.2.1 编程粒度

HWD29S01GPBGA64M3 支持两种编程方法，字或写缓冲区编程。每个页面都可以通过任何一种方法进行编程。由不同编程的页面对于工业温度版本（-40° C 至+85° C），方法可以在一条线内混合。对于 N1 级版本（-55° C 至+125° C），两个擦除操作之间设备将仅支持每 32 字节页面一次编程操作，不支单字编程命令。

字编程检查命令提供的数据字，并对寻址存储器阵列字中的 0 进行编程，以匹配命令数据字中的 0。

写缓冲区编程检查写缓冲区并对寻址内存阵列页中的 0 进行编程，以匹配写缓冲区中的 0。写入缓冲器不需要被完全填充数据。允许将少至单个位、几个位、单个字、几个字、一页、多页或整个缓冲器作为一个编程操作进行编程。用写入缓冲区的方法减少了写入程序命令中的主机系统开销，并减少了编程操作中的存储器设备内部开销，以使写入缓冲器编程更有效更快，从而不用单词编程命令对单个单词进行编程。

7.5.4.2.2 增量编程

可以通过字或写缓冲器编程方法对相同的字位置进行多次编程，以增量地将 1 更改为 0。

7.5.4.3 自动 ECC

7.5.4.3.1 ECC 概述

自动 ECC 功能显性是在正常编程、擦除和读操作中工作。当设备从写缓冲区传输每一页数据到内存阵列中，内部 ECC 逻辑为每一页执行 ECC 编码为内存阵列的一部分，主机系统不可见。设备评估页数据，并且初始页访问时执行 ECC 编码，在初始内部访问时，ECC 逻辑会更正一位的错误。

编程超过一页会使为特定页而执行 ECC 失效。特定页 ECC 功能失效会保持不变，直到下次主机系统擦除包含此页的扇区。主机系统可以在多次编程操作后读存储在此页的数据，然而，ECC 失效，不能探测到页的错误并更新它。

7.5.4.3.2 编程和擦除 ECC 总节

因性能和可靠性原因，设备执行读和编程是满 32-byte 页并行模式。设备给每一页在编程时每一页通过增加一个 ECC 编码方式提供 ECC。ECC 编码是自动的和显性的。

7.5.4.3.3 ECC 执行

主 FLASH 阵列中每 32-byte 页和 OTP 区域有相应的 ECC 编码功能。ECC 编码包含在 ECC 逻辑中，能探测到并更正读访问页中任何单个位的错误。

第一次写缓冲区编程操作应用一页编程当页的 ECC 编码。然后当一个特定页 ECC 功能失效超过一次时编程操作。允许位或字编程；然而，注意多次编程操作到同样的页会使当页的 ECC 功能失效。包含 ECC 失效页的扇区擦除，会使当页 ECC 功能重生效。

ECC 功能对用户是自动和显性的。ECC 功能的数据显性增强了一次写数据到每一页典型编程操作的数据完整型。ECC 功能也通过允许同一页单个字编程和同一页位行走或字编程超过一次，促进软件兼容性。当一页有自动 ECC 失效时，ECC 功能不能探测和更正此页读数据的错误。

7.5.4.3.4 字编程相关 ECC

字编程编程主闪存阵列中单个字。同一页 32-byte 页中编程多字，使此页的 ECC 保护失效。在此页的字编程后，包含此页的扇区的擦除使自动 ECC 重新生效，

7.5.4.3.5 写缓冲区编程相关 ECC

每一次写缓冲区操作允许 1 位到 512 bytes 的编程。一个 32-byte 是自动 ECC 保护的最小编程粒度。编程同一页超过一次会使此页的自动 ECC 失效。华微推荐一次写缓冲区编程操作编程多页，并且写每一页仅一次。这样可以保持此页自动 ECC 保护生效。最好的，编程排列于 512-byte 同边界的满线 512 bytes

7.5.4.4 编程命令集

7.5.4.4.1 字/字节编程

字/字节编程模式是对 Flash 进行编程的一种方法。在这种模式下，需要使用四个 Flash 指令写入周期用于对单个 Flash 地址进行编程。此编程操作的数据可以是 8 位或 16 位宽。通过写入两个解锁写入周期来启动程序命令序列，随后是编程设置命令。接下来写入地址和数据，这又启动嵌入式字/字节程序算法。系统不需要提供进一步的控制或计时。设备自动生成程序脉冲并在内部验证编程的单元余量。当嵌入式字/字节编程算法完成时，EAC 然后返回到其待机模式。

系统可以通过使用数据轮询状态、读取状态寄存器或监视 RY/BY#输出来确定程序操作的状态。有关这些信息，请参见状态寄存器状态位。有关这些状态位的信息，请参见数据轮询状态。编程操作示意图见图

7-28。

在嵌入式编程算法期间写入到设备的除程序挂起之外的任何命令都被忽略。注意，硬件复位 (RESET#=V_{IL}) 立即终止编程操作并在 t_{RPH} 时间后将设备返回到读取模式。为确保数据完整性，一旦设备完成硬件复位操作，应重新启动程序命令序列。

当输入到锁定寄存器、密码和 PPB ASOs 时，没有解锁写入周期的字/字节编程命令的修改版本用于编程。相同的命令用于进入 PPB 锁和 DYB ASOs 时，更改易失性位。程序命令序列见表 7-29。

单字编程操作期间，系统在发送地址、指令和数据时，必须将 CE# 和 WE# 驱动到 V_{IL}，将 OE# 驱动到 V_{IH}。地址在 WE# 或 CE# 的最后一个下降沿锁存，而数据在 WE# 或 CE# 的第一个上升沿锁存。CE # 可以保持在低电平，或者在每个周期都拉高。

解锁旁路功能允许主机系统向闪存设备发送编程指令，而无需先在指令序列中写入解锁周期。

内部编程完成后，器件将返回读取模式，地址不再被锁存。

注意：

1. 在编程期间，任何指令（暂停编程指令除外）都将被忽略。
2. 系统可以通过读取 DQ 状态位来确定编程操作的状态。。
3. 通常在发送完编程数据后可检测到 BUSY 状态。
4. 编程操作正在进行时，安全硅扇区、自动选择和 CFI 功能不可用。
5. 硬件复位会立即终止编程操作。一旦设备返回读取模式，应重新启动编程指令序列，以确保数据完整性。
6. 内部编程完成后，设备返回读取模式。
7. “0” 不能被编程回 “1”。成功读取表明数据仍为 “0”。
8. 只有擦除操作才能将 “0” 转换为 “1”。
9. 在内部编程期间写入设备的任何指令被忽略，除了挂起指令。
10. 硬件重置和/或断电会立即终止编程操作，一旦设备返回读取模式，应重新启动指令序列以确保数据完整性。
11. 允许多次编程到同一个字地址而不插入擦除。

图 7-28 单字/字节编程

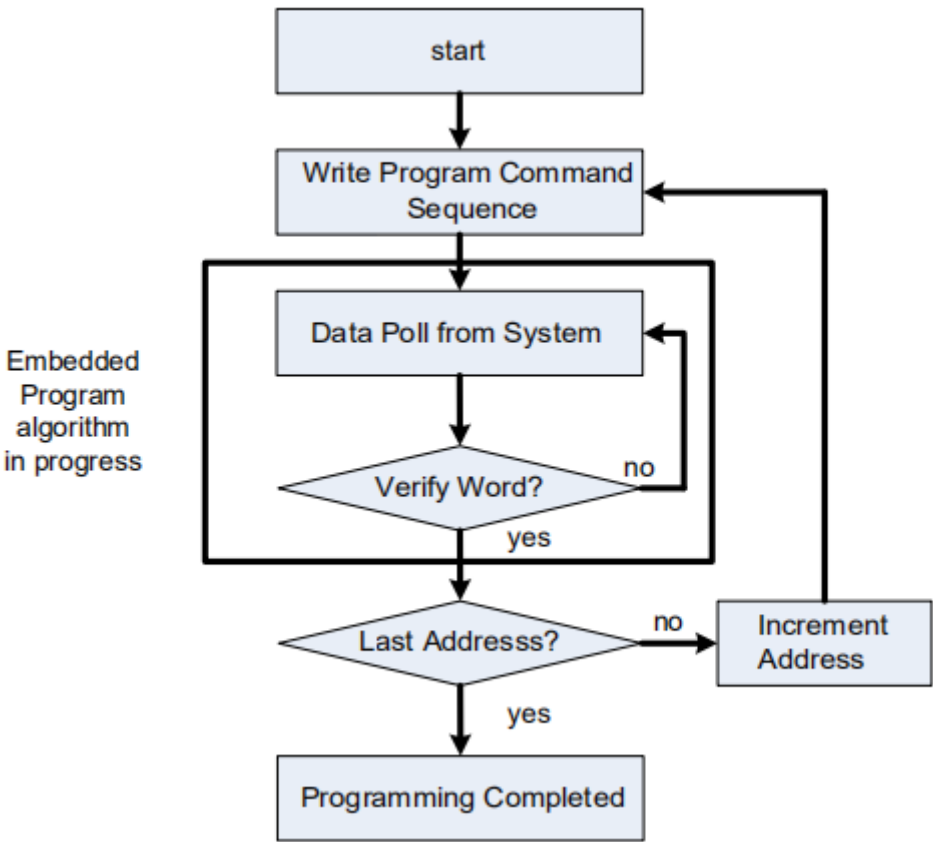


表 7-29 单字/字节编程

Cycle	Operation	Byte Address (x8)	Word Address (x16)	Data
Unlock Cycle 1	Write	AAAh	555h	00AAh
Unlock Cycle 2	Write	555h	2AAh	0055h
Program Setup	Write	AAAh	555h	00A0h
Program	Write	Byte Address	Word Address	Data

图 7-29 单字/字节编程 (x16 模式)

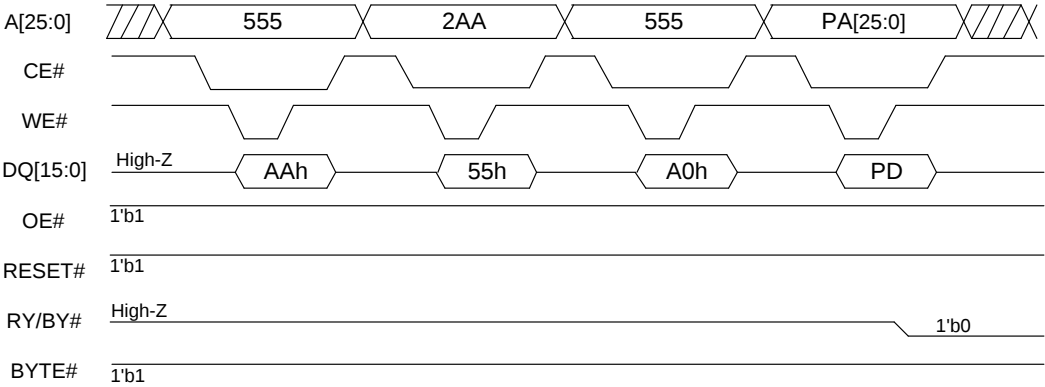
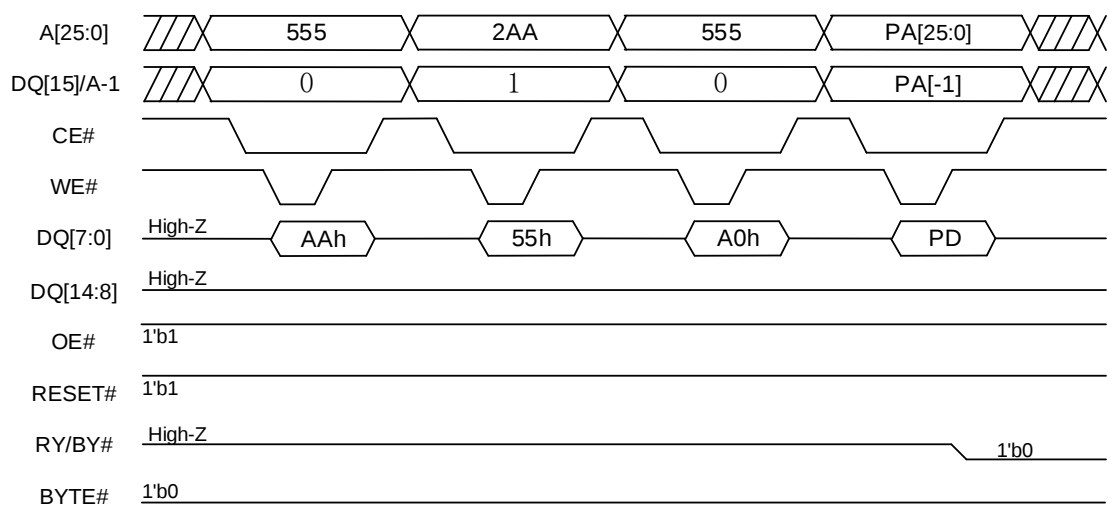


图 7-30 单字/字节编程 (x8 模式)



7.5.4.4.2 写缓冲编程

写入缓冲区用于对齐在 512 字节边界（行）上的 512 字节地址范围内的数据进行编程。这比标准“字”编程更快。因此，全写缓冲器编程操作必须在行边界上对齐。程序小于完整 512 字节的编程操作可以在任何字边界上开始，但可以不跨越线边界。在写缓冲器编程操作开始时，缓冲器中的所有位都是 1 (FFFFh 字)，因此任何未加载的位置都将保留现有数据。有关地址映射的信息，请参见产品概述。

写缓冲区编程允许在一次操作中编程多达 512 个字节。在每个写缓冲器编程操作中，可以从 1 位编程到 512 字节。建议要写入多个页面，并且每个页面只写入一次。为了获得最佳性能，编程应在 512 字节边界上对齐的 512 字节的整行中完成。写缓冲器仅在主闪存阵列或 SSR AS0 中支持编程。

通过首先写入两个解锁周期来启动写入缓冲器编程操作。这之后第三周期是写具有扇区地址 (SA) 的写缓冲器命令，编程即将发生。接下来，系统写入字位置的数量减去 1。这告诉设备有多少写缓冲器地址加载了数据，并因此告诉设备何时期望程序缓冲区到闪存确认命令。扇区地址必须在写入缓冲区命令和写入字/字节计数命令中匹配。要编程的位置数量不能超过写入缓冲区的大小，否则操作将中止。（加载的数量 = 要编程的位置数减 1。例如，如果系统编程了 6 个地址，则应将 05h 写入设备。）要编程的扇区必须解锁（不受保护）。

然后，系统写入起始地址/数据组合。该起始地址是要编程的第一个地址/数据对，并选择写入缓冲区线地址。扇区地址必须与写入缓冲区扇区地址匹配，否则操作将中止并返回到启动状态。所有后续地址/数据对必须按顺序排列。所有写缓冲区地址必须在同一线地址内。如果系统试图加载超出此范围的数据，操作将中止并返回到启动状态。

计数器对于每个数据加载操作递减。注意，在对数据写入进行计数时，每次写入都被认为是数据被加载到写入缓冲器中。没有命令是可能的在写入缓冲区加载期间。停止加载写缓冲器的唯一方法是使用编程操作线地址之外的地址进行写入。此无效地址将立刻自动中止写入缓冲区命令。

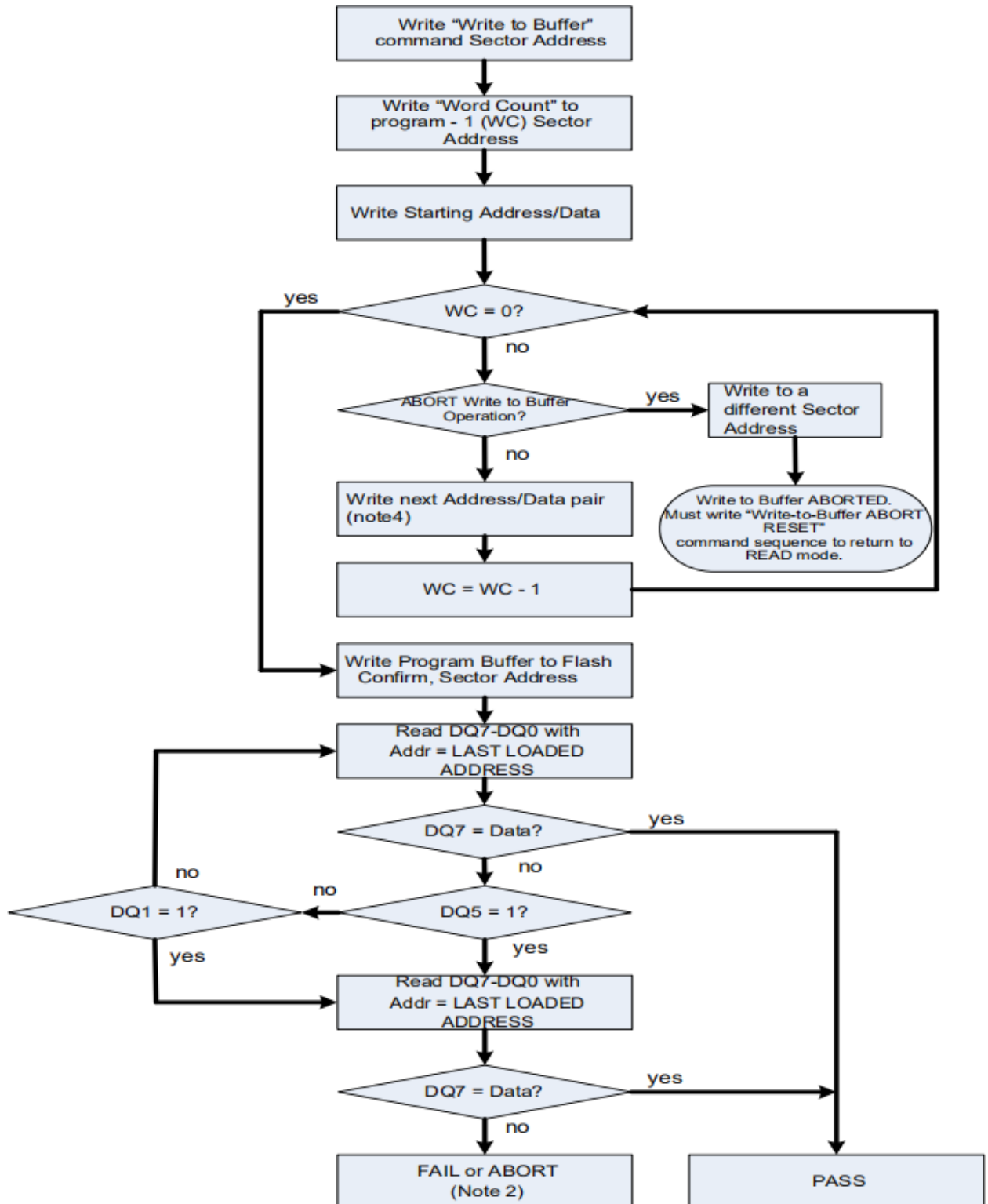
一旦加载了指定数量的写缓冲区位置，系统就必须将编程缓冲区写入扇区地址的闪存命令。然后设备状态为忙碌。嵌入式编程算法自动编程并验证数据的正确数据模式。在这些操作期间，系统不需要提供任何控制或定时。如果序号不正确写入缓冲器位置已被加载到操作序列，操作将中止并返回到起始状态。当写入到闪存程序缓冲区以外任何内容时，就会发生中止命令出现在字/字节计数的末尾。

可以使用编程挂起命令暂停写入缓冲器嵌入式编程操作。当嵌入式编程算法完成时，EAC 返回到 EAC 待机或擦除暂停启动编程操作的待机状态。

系统可以通过使用数据轮询状态、读取状态寄存器或监视 RY/BY# 输出来确定程序操作的状态。有关这些信息，请参见状态寄存器状态位。有关这些状态位的信息，请参见数据轮询状态。编程操作示意图见图

7-31 具有数据轮询状态的写缓冲区编程操作图 7-31。

图 7-31 具有数据轮询状态的写缓冲区编程操作



注 1：即使 DQ5=1，也应重新检查 DQ7，因为 DQ7 可能与 DQ5 同时变化。

注 2：如果由于 DQ5=1 而到达该流程图位置，则设备失败。如果这个流程图因为 DQ1=1，所以到达位置，然后中止写缓冲区操作。在任一情况下，必须将适当的复位命令写入设备，以使设备返回到读取模式。写入缓冲如果 DQ1=1，则编程中止-复位。如果 DQ5=1，则软件复位或写缓冲区编程中

止-复位。

注 3：写入缓冲区编程所需的命令序列见表 7-30。

注 4：如果指定了扇区地址，则所选扇区中的任何地址都是可接受的。但是，当用数据加载写缓冲区地址位置时，所有地址都必须落在所选写缓冲区内。

注 5：在字节编程时，WC 指 BC。

写缓冲区编程序列将在以下情况下中止：

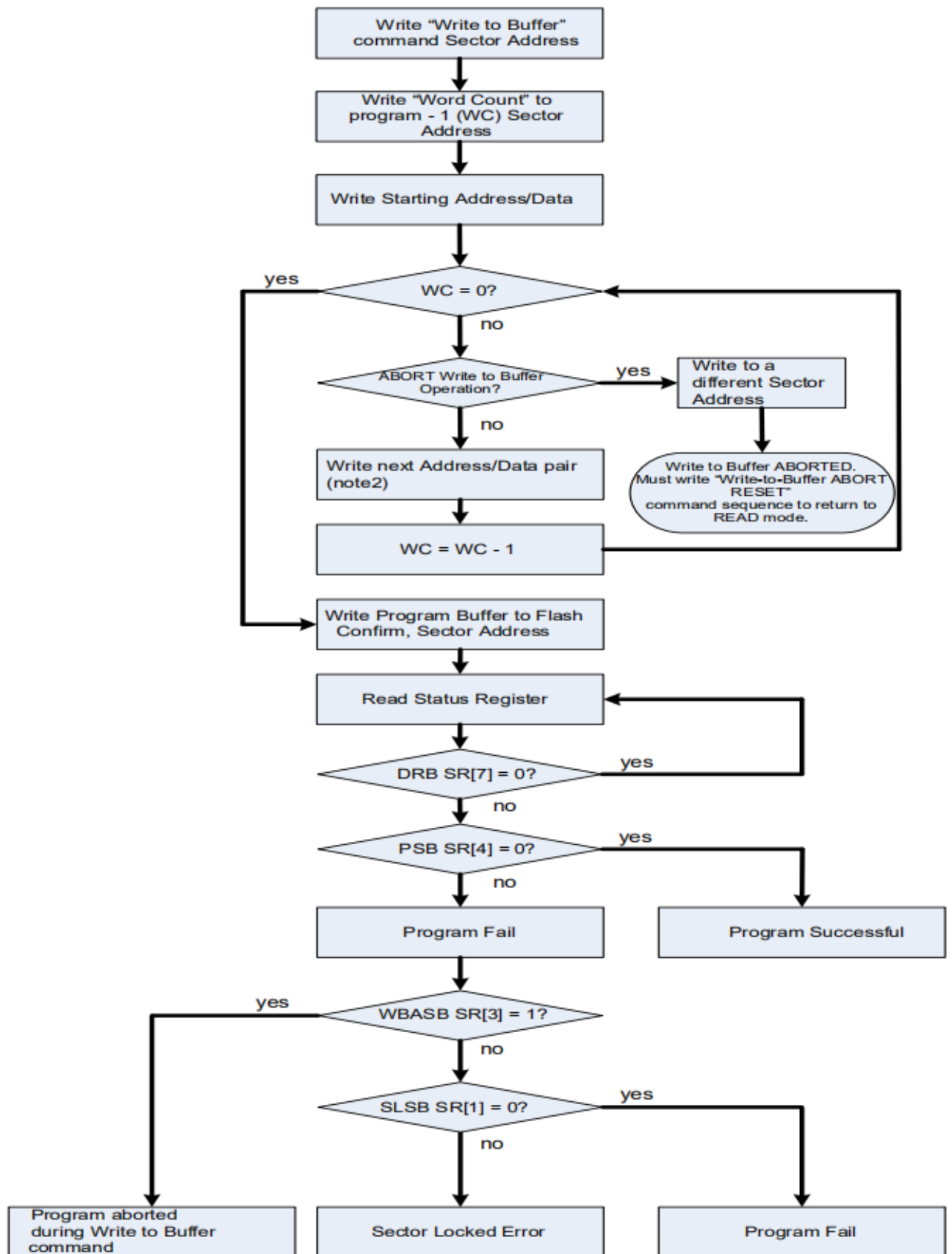
在以下情况下，写入缓冲区编程序列将被中止：

- 加载大于缓冲区大小的字/字节计数值（255）。
- 写一个线外的地址在写入缓冲区命令中标识。
- 程序缓冲区到闪存命令在写入字/字节计数加载数据字的数量后不发出。

当任何导致写缓冲区命令中止的条件发生时，中止将紧接在有问题的条件之后发生，并将指示状态寄存器为编程失败，由于写入缓冲器中止位 bit3(WBASB=1)而在位 bit4(PSB=1)处。下一个成功的编程操作将清除故障状态，或者可以发出清除状态寄存器来清除 PSB 状态位。

写缓冲器编程序列可以通过以下方式停止：硬件复位或电源循环。然而，使用这些方法中的任一种的这些方法可能会将被编程的区域留在中间状态具有无效或不稳定的数据值。在这种情况下，相同的区域将需要用相同的数据重新编程或擦除，以确保数据值被正确地编程或擦除。

图 7-32 使用状态寄存器的写缓冲区编程操作



注 1：写入缓冲区编程所需的命令序列见表 7-30。

注 2：当指定扇区地址时，所选扇区中的任何地址都是可接受的。但是，在加载时具有数据的写缓冲区地址位置，所有地址必须位于选定的写缓冲区页内。

注 3：在字节编程时，WC 指 BC。

表 7-30 写缓冲编程

序列	字节地址	字地址	数据	描述
发送解锁命令 1	AAAh	555h	00AAh	
发送解锁命令 2	555h	2AAh	0055h	
发送写缓冲区命令扇区地址	Sector Address		0025h	
发送 SA 处的数据数量	Sector Address		WC	WC=要编程的字-1 的数值
例如：WC=0：编程 1 个字 WC=1：编程 1 个字				
加载地址/数据对	Starting Address		PD	选择写缓冲区页和加载地址/数据对
加载下一地址/数据对	WBL		PD	所有地址必须在选择的写缓冲 员边界内，且必须顺序加载
.....				
加载最后一地址/数据对	WBL		PD	
发布写缓冲区编程确认扇区地址	Sector Address		0029h	命令必须跟随最后一个写缓冲 区位加载，或终止操作
设备状态忙				

注 1：WBL=Write Buffer Location(必须在开始地址选定的写缓冲区的边界内，必须按顺序加载)。

注 2：PD=Program Data

注 2：写入缓冲区的数据量可以是 1~256 个字 (x16) 或 1~512 个字节 (x8)。

注 3：Last = 写缓冲区编程操作的最后一个周期。

注 4：为获得最大效率，建议在写入缓冲区中加载尽可能多的数据量。

图 7-33 写缓冲区编程 (x16 模式)

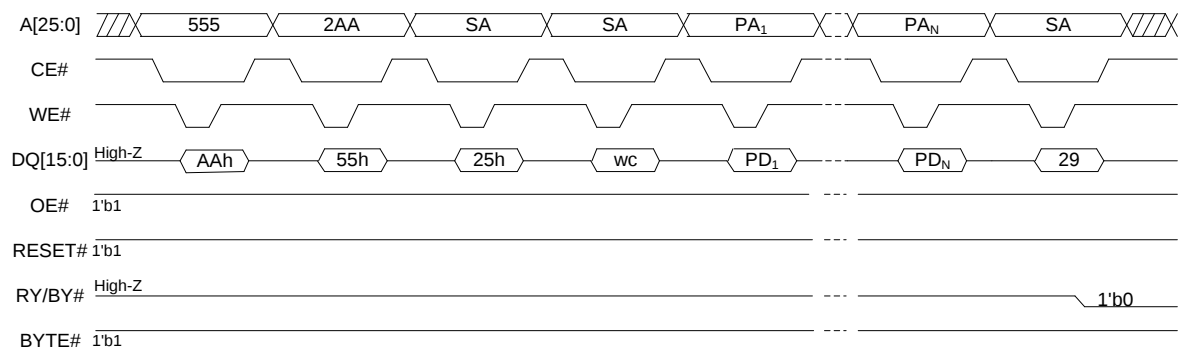
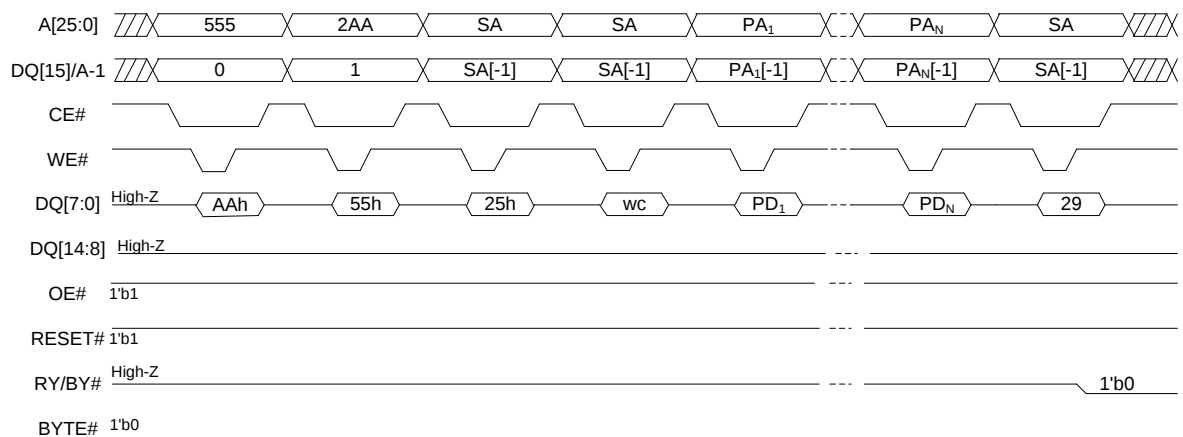


图 7-34 写缓冲区编程 (x8 模式)



7.5.4.5 解锁旁路命令

解锁旁路命令 (20h) 用于将设备置于解锁旁路模式。发出解锁旁路命令需要三次总线写入操作。

当器件进入解锁旁路模式时，不需要标准编程操作所需的两个初始解锁周期，从而实现更快的总编程时间。

解锁旁路命令与解锁旁路程序命令一起使用，以便比使用标准程序命令更快地对设备进行编程。使用这些命令可以节省大量时间当到设备的周期时间较长时。

当处于解锁旁路模式时，只有以下命令有效：解锁旁路编程命令可以发布到设备内的编程地址。可以发出解锁旁路复位命令使设备返回读取模式。在解锁旁路模式下，可以像在读取模式下一样读取设备。

除了 UNLOCK BYPASS 命令外，当 WP#/ACC 升高到 V_{HI} 时，器件自动进入 UNLOCK BYPASS 模式。当 WP#/ACC 返回 V_{IH} 或 V_{IL} 时，设备不再处于解锁旁路状态模式，正常操作恢复。从 V_{IH} 到 V_{HI} 和从 V_{HI} 到 V_{IH} 的转变必须比 t_{VHVP} 慢。(请参阅加速程序，数据轮询/切换 AC 特性。)

注意：华微建议使用 ENTER UNLOCK BYPASS 和 UNLOCK BYPASS RESET 命令进入和退出 UNLOCK BYPASS 模式，而不是将 WP#/ACC 提高到 V_{HI} 。WP#/ACC 不应提高到 V_{HI} 来自除读取模式之外的任何模式；否则，设备可能处于不确定状态。WP#/ACC 在 V_{HI} 累计停留时间不应超过 80 小时。

7.5.4.6 解锁旁路复位命令

解锁旁路复位 (90/00h) 命令用于从解锁旁路模式返回到读取/复位模式。发出解锁旁路复位命令需要两次总线写入操作。读取/重置命令不会退出解锁旁路模式。

7.5.4.7 加速编程操作

加速编程操作提供了比标准程序命令序列更快的性能。在 V_{HI} 电压供应下通过 WP#/ACC 使能操作。当系统断言 V_{HI} 置高输入时，设备自动进入解锁旁路模式，使系统能够使用解锁旁路编程 (A0h) 命令序列。在

完成嵌入式编程操作时从 ACC 中移除 V_{BI} 使设备恢复正常操作。

设备状态	要求/建议
设备块	要求：在将 WP#/ACC 提升至 VHH 之前必须无保护
V_{BI} 应用到 WP#/ACC	要求：最大累积周期为 80 小时。
WP#/ACC	要求：除加速编程外，不得在 V_{BI} 进行操作；否则可能会损坏设备。
	建议：在加速编程运行期间保持稳定到 V_{BI} 。
上电	建议：在 V_{CC}/V_{CCQ} 稳定开启后，在 WP#/ACC 上应用 V_{BI} 。
下电	建议：在 V_{CC}/V_{CCQ} 变低之前，将 WP#/ACC 从 V_{BI} 调整为 V_{IH}/V_{IL} 。

7.5.4.8 编程挂起/编程恢复指令

编程挂起命令允许系统中断嵌入式编程操作，以便可以从任何非挂起行读取数据。当在编程期间写入编程挂起命令时，设备在 t_{PSL} （编程暂停延迟）内暂停编程操作并更新状态位。在编写程序挂起命令时，不关心地址。

有两个命令可用于编程挂起。传统的组合擦除/编程挂起命令 (B0h 命令代码) 和单独的编程挂起命令 (51h 命令代码)。有两个编程恢复命令。传统的组合擦除/编程恢复命令 (30h 命令代码) 和单独的编程恢复命令 (50h 命令代码)。建议使用单独的编程挂起和恢复命令用于编程，并仅将传统组合命令用于擦除、暂停和恢复。

在编程操作被挂起之后，系统可以从任何未挂起的行读取阵列数据。编程挂起命令也可以在编程操作期间发出，同时擦除挂起。在这种情况下，可以从不处于擦除挂起或编程挂起的任何地址读取数据。

写入编程恢复命令后，设备恢复编程，状态位被更新。系统可以通过读取状态寄存器来确定程序操作的状态或使用数据轮询。有关这些状态位的信息，请参阅状态寄存器。有关更多信息，请参阅数据轮询状态。在编程挂起期间有效的访问和命令是：

- 读取到任何其他非擦除挂起扇区
- 读取到任何其他非编程挂起线
- 状态读取命令
- 退出 AS0 或命令集退出
- 编程恢复命令

系统必须写入编程恢复命令，退出编程挂起模式，继续编程操作。编程恢复命令的进一步写入被忽略。另一个编程可以在设备恢复编程后写入挂起命令。

编程操作可以根据需要经常中断，但是为了使编程操作进展到完成，在恢复和下一个暂停通信之间必须有一些时间段并且大于或等于嵌入式算法控制器 (EAC) 中的 t_{PRS} 。

在 AS0 中输入编程时不支持暂停和恢复。在编程中，不支持暂停进入 AS0。

表 7-31 编程挂起期间（tPSL 后）可接受的指令

指令		备注
读		可以从任何未挂起的缓冲区页面读取数据
程序恢复		
重置		
解锁旁路	输入	
	读	可以从任何未挂起的缓冲区页面读取数据
	重置	
安全硅扇区	安全硅扇区入口	
	读	
	安全硅扇区出口	
自动	制造商 ID	
	设备 ID	
	扇区保护验证	
	安全设备验证	
	重置	
CFI	CFI 查询	
	读	
	重置	

图 7-35 编程挂起（x16 模式）

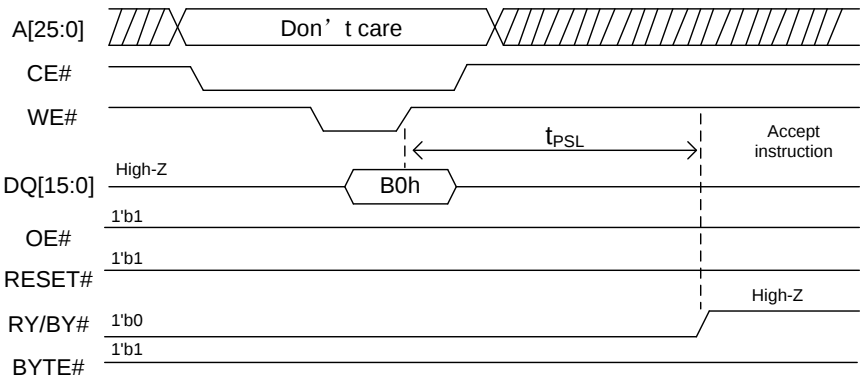


图 7-36 编程挂起（x8 模式）

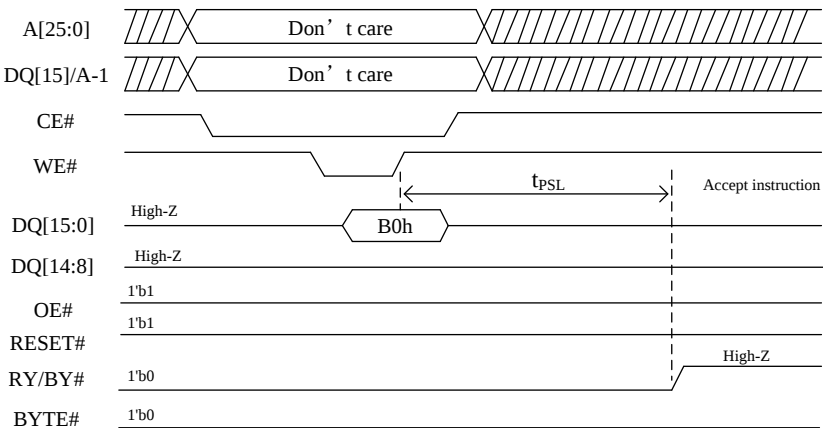


图 7-37 编程恢复（x16 模式）

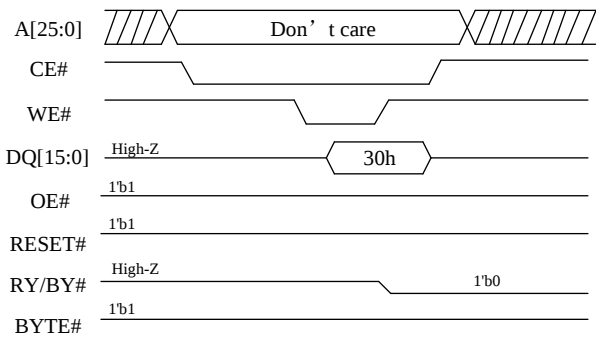
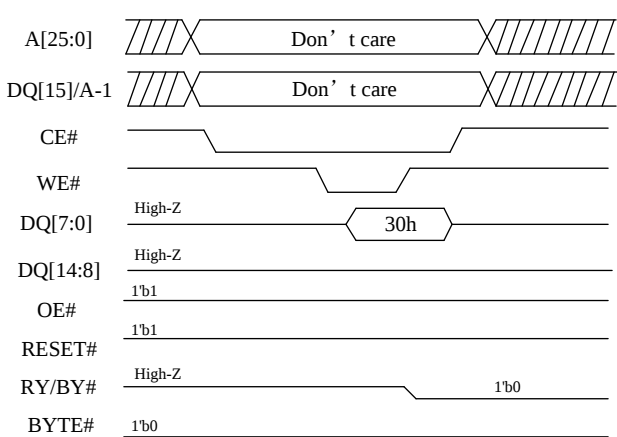


图 7-38 编程恢复（x8 模式）



7.5.4.9 空白检查

空白检查命令将确认所选主闪存阵列扇区是否被擦除。空白检查命令不允许在空白检查期间读取数组。读取数组，而正在执行的此命令将返回未知数据。

为了在扇区上启动空白检查，将 33h 写入扇区中的地址 555h，当 EAC 处于待机状态时，可以不写入空白检查命令或擦除或挂起。

使用状态寄存器读取来确认设备是否仍在忙，以及完成时扇区是否为空。状态寄存器的第 7 位将显示设备是否正在执行空白检查（类似于擦除操作）。如果扇区被擦除，状态寄存器的位 5 将被清零为 0，如果未被擦除，则被设置为 1。

一旦发现任何位未被擦除，设备将停止操作并报告结果。

一旦空白检查完成，EAC 将返回待机状态。

7.5.4.10 擦除命令集

7.5.4.10.1 扇区擦除

扇区擦除功能擦除存储器阵列中的一个扇区。在擦除之前，设备不需要系统进行预编程。嵌入式擦除算法自动编程并验证电擦除之前全 0 数据模式的整个扇区。成功擦除扇区后，被擦除扇区内的所有位置都包含 FFFFh。系统不需要提供这些操作期间的任何控制或定时。扇区擦除命令序列通过写入两个解锁周期来启动，随后是设置命令。然后执行两个附加的解锁写入循环。后面跟着要擦除的扇区的地址和扇区擦除命令。当 WE# 走高时，在第 6 个周期结束时，RY/BY# 走低。

系统可以通过读取状态寄存器或使用数据轮询来确定擦除操作的状态。有关这些状态位的信息，请参阅状态寄存器。请参阅数据轮询更多信息。

一旦扇区擦除操作开始，状态寄存器读取和擦除暂停命令就有效。所有其他命令都被忽略。然而，请注意，硬件复位立即终止测试擦除操作，并在 tRPH 时间后返回读取模式。如果扇区擦除操作终止，则一旦设备将操作重置，则必须重新启动扇区擦除命令序列，以确保数据完整性。

有关参数和时序图，请参见嵌入式算法控制器（EAC）。受 ASP DYB 和 PPB 锁定位保护的扇区将不会被擦除。参见 ASP。

表 7-32 扇区擦除

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Unlock	Write	AAAh	555h	00AAh
2	Unlock	Write	555h	2AAh	0055h
3	Setup Instruction	Write	AAAh	555h	0080h
4	Unlock	Write	AAAh	555h	00AAh
5	Unlock	Write	555h	2AAh	0055h
6	Sector Erase Instruction	Write	SA ₀	SA ₀	0030h
...	...	...	...	...	...
N+6	Sector Erase Instruction	Write	SA _N	SA _N	0030h

注 1：SA = 扇区地址。

注 2：可以选择无限的附加扇区进行擦除；指令必须在 50 μs 内写入。

图 7-39 扇区擦除操作

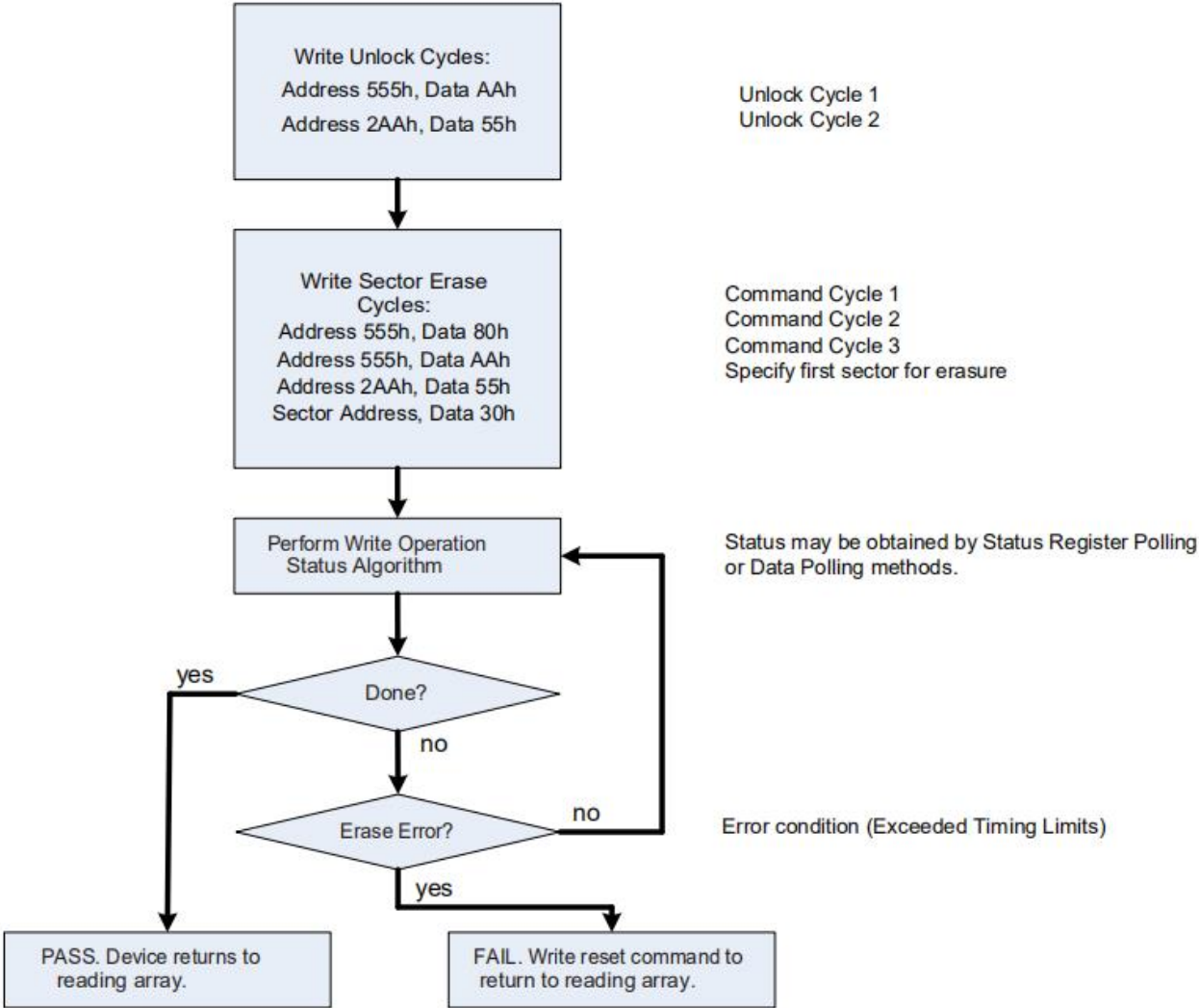


图 7-40 扇区擦除 (x16 模式)

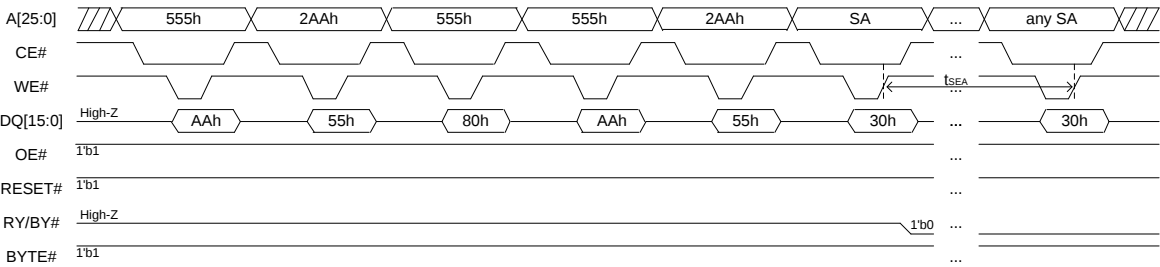
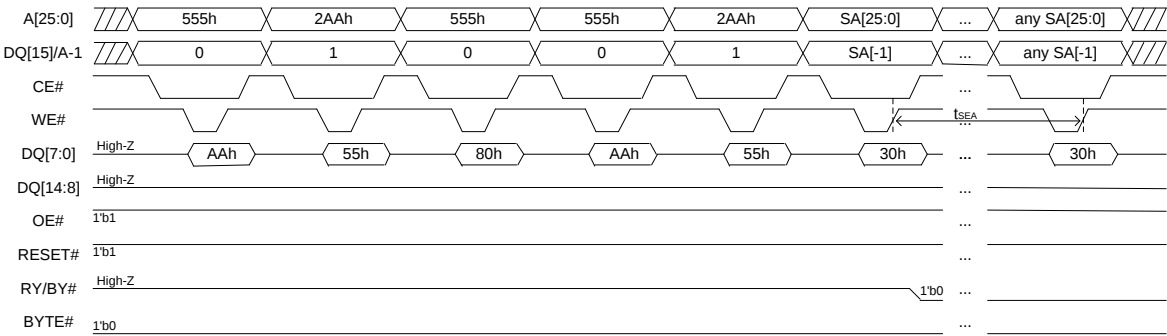


图 7-41 扇区擦除 (x8 模式)



7.5.4.10.2 全片擦除

芯片擦除功能擦除整个主闪存阵列。在擦除之前，设备不需要系统进行预编程。嵌入式擦除算法自动编程和验证在电擦除之前，对全 0 数据模式的整个存储器进行擦除。成功擦除芯片后，设备内的所有位置都包含 FFFFh。这些操作期间系统不需要提供任何控制或定时。芯片擦除命令序列通过写入两个解锁周期来启动，随后是一个设置命令。接着是两个附加的解锁写入循环通过芯片擦除命令，该命令又调用嵌入式擦除算法。当 WE#走高时，在第 6 个周期结束时，RY/BY#走低。

当嵌入式擦除算法完成时，EAC 返回到待机状态。请注意，当嵌入式擦除操作正在进行时，系统无法从设备读取数据。系统可以通过读取 RY/BY#、状态寄存器或使用数据轮询来确定擦除操作的状态。有关这些状态位的信息，请参阅状态寄存器。参考数据轮询状态了解更多信息。

一旦芯片擦除操作已经开始，只有状态读取、硬件复位或电源循环是有效的。所有其他命令都被忽略。然而，硬件复位或电源循环立即终止擦除操作，并在 tRPH 时间后返回读取模式。如果芯片擦除操作终止，则一旦设备返回到空闲状态，则必须重新启动芯片擦除命令序列以确保数据完整性。

有关参数和时序图，请参见表 7-37，异步写入操作和替代 CE#控制的写入操作。

受 ASP DIB 和 PPB 锁定位保护的扇区将不会被擦除。参见 ASP。如果一个扇区在芯片擦除期间受到保护，芯片擦除将跳过受保护的扇区并继续下一个扇区。受保护扇区上的擦除失败不会将状态寄存器擦除状态位和扇区锁定位设置为 1。

图 7-42 全片擦除操作

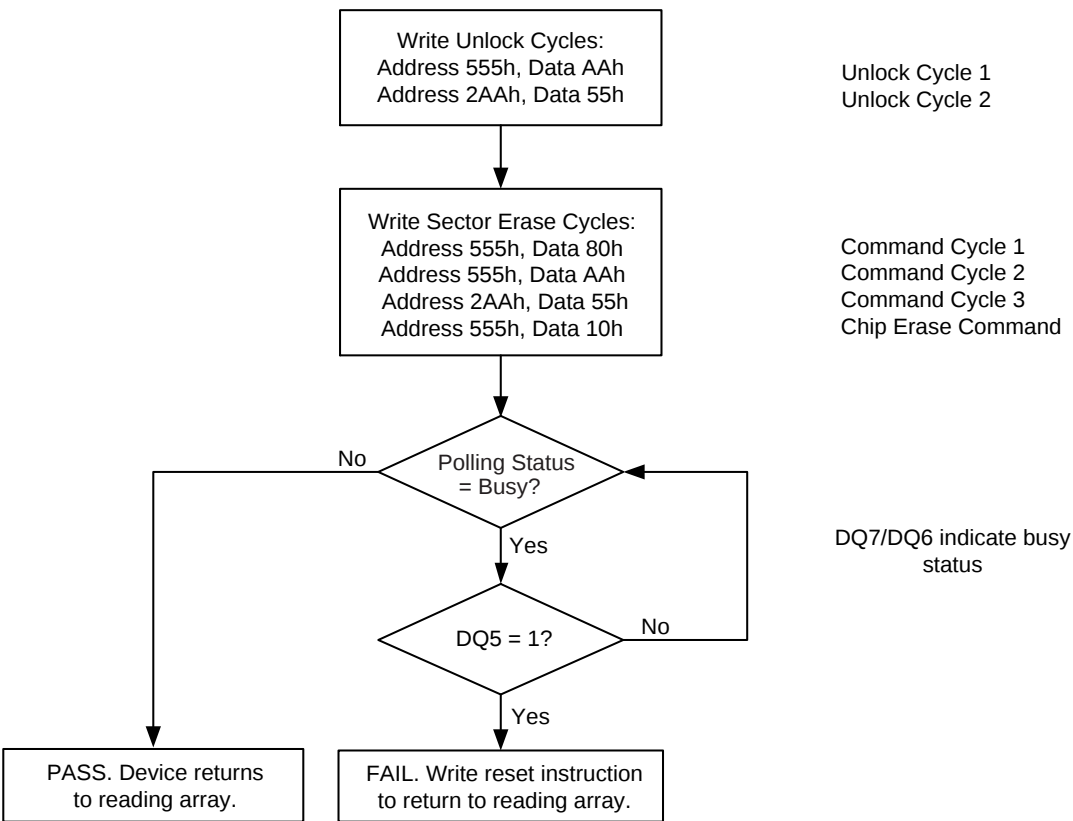


表 7-33 全片擦除

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Unlock	Write	AAAh	555h	00AAh
2	Unlock	Write	555h	2AAh	0055h
3	Setup Instruction	Write	AAAh	555h	0080h
4	Unlock	Write	AAAh	555h	00AAh
5	Unlock	Write	555h	2AAh	0055h
6	Sector Erase Instruction	Write	AAAh	555h	0010h

图 7-43 全片擦除 (x16 模式)

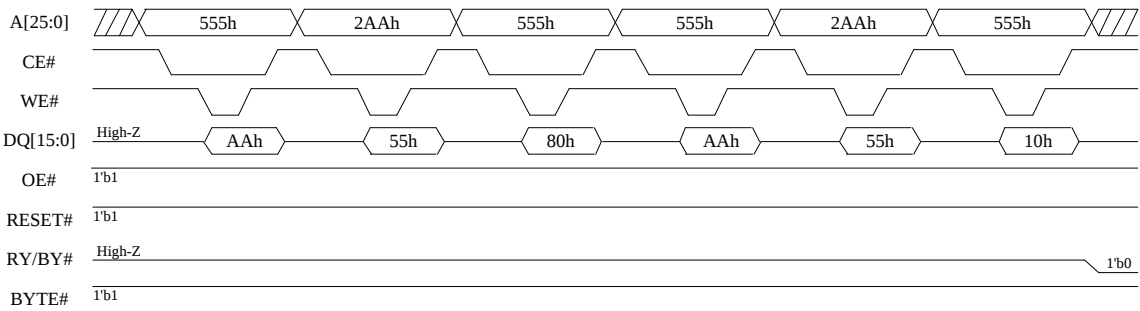
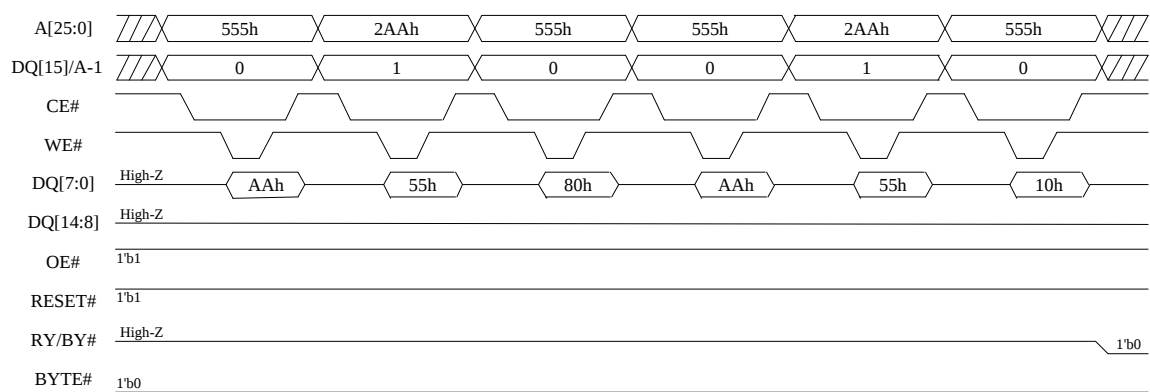


图 7-44 全片擦除（x8 模式）



7.5.4.11 擦除挂起/擦除恢复指令

擦除挂起命令允许系统中断扇区擦除操作，然后从主闪存阵列读取数据或对其编程。此命令仅在扇区擦除或程序操作。如果在芯片擦除操作期间写入，则忽略擦除挂起命令。

当在扇区擦除操作期间写入擦除挂起命令时，设备需要最大 tESL（擦除挂起等待时间）来挂起擦除操作并更新状态位。

在擦除操作被挂起之后，部件进入擦除挂起模式。系统可以从主闪存阵列读取数据或将数据编程到主闪存阵列。在擦除挂起的任何地址读取扇区产生未确定的数据。系统可以通过读取状态寄存器或使用数据轮询来确定扇区是主动擦除还是被擦除挂起。参见状态寄存器有关这些状态位的信息。有关更多信息，请参阅数据轮询状态。

在擦除挂起的编程操作完成之后，EAC 返回到擦除挂起状态。系统可以通过读取状态寄存器来确定程序运行的状态，仅和标准编程操作一样。

如果在擦除挂起期间发生编程失败，清除或复位命令将使设备返回到擦除挂起状态。在再次尝试之前，需要恢复并完成擦除对存储器阵列进行编程。

在擦除挂起期间有效的访问和命令有：

- 向任何其他非挂起扇区读取
- 计划向任何其他非挂起扇区
- 状态寄存器读取
- 状态寄存器清除
- 进入 DYB ASO
- DYB 设置
- DYB 清除
- DYB 状态读取
- 退出 ASO 或命令设置退出
- 擦除恢复命令

要恢复扇区擦除操作，系统必须写入擦除恢复命令。设备将恢复到擦除状态，状态位将被更新。恢复命令的进一步写入是被忽略了。在芯片已经恢复擦除之后，可以写入另一个擦除挂起命令。

在 ASO 中输入时，不支持擦除、挂起和恢复。在擦除时，不支持挂起进入 ASO。

表 7-34 擦除挂起期间（tESL 后）可接受的指令

指令		备注
读		从未选择用于扇区擦除的任何大块中读取输入
单字编程		将数据编程到未选择用于扇区擦除的任何大块
写入缓冲区		将数据编程到未选择用于扇区擦除的任何大块
编程缓冲区到闪存（确认）		
写入缓冲区中止复位		
编程暂停		
擦除恢复		
重置		
解锁旁路	进入	
	读	从未选择用于扇区擦除的任何大块中读取输入
	单字编程	将数据编程到未选择用于扇区擦除的任何大块
	写入缓冲区	将数据编程到未选择用于扇区擦除的任何大块
	编程缓冲区到闪存（确认）	
	写入缓冲区中止复位	
	重置	
安全硅扇区	安全硅扇区入口	
	读	
	单字编程	
	写入缓冲区	
	编程缓冲区到闪存（确认）	
	写入缓冲区中止复位	
	安全硅扇区出口	
自动选择	制造商 ID	
	设备 ID	
	扇区保护验证	
	安全设备验证	
	重置	

CFI	CFI 查询	
	读	
	重置	

图 7-45 擦除挂起 (x16 模式)

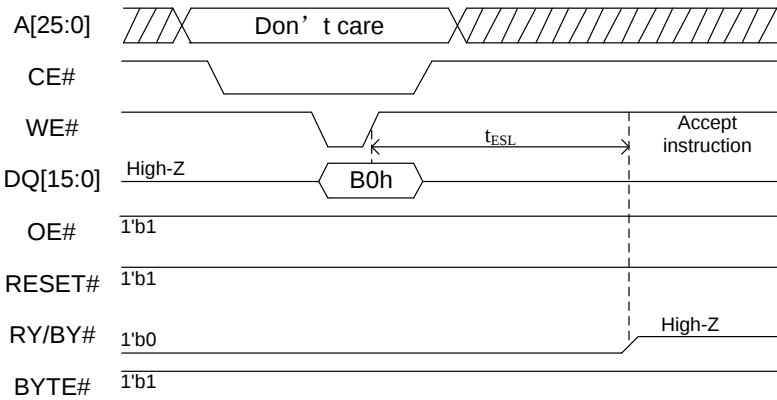


图 7-46 擦除挂起 (x8 模式)

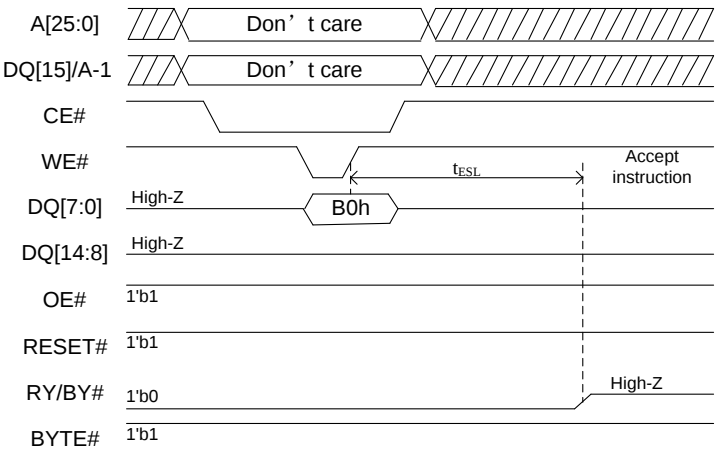


图 7-47 擦除恢复 (x16 模式)

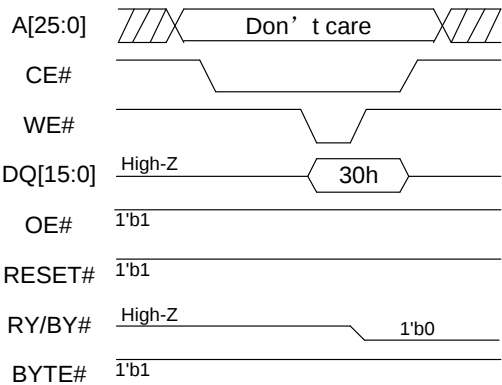
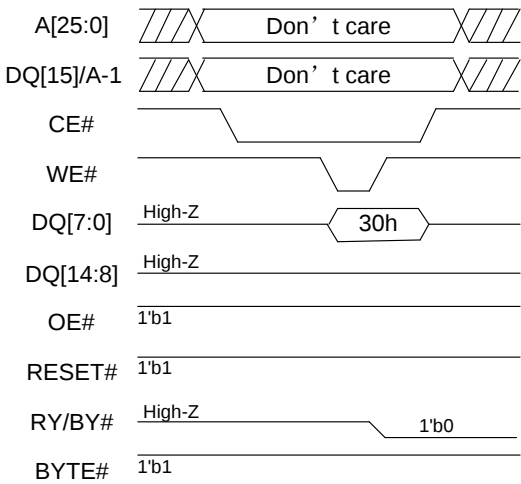


图 7-48 擦除恢复 (x8 模式)



7.5.4.12 AS0 进入和退出

7.5.4.12.1 ID-CFI AS0

系统可以通过在读取模式期间发出 ID-CFI 进入命令序列来访问 ID-CFI AS0。该进入命令使用命令中的扇区地址 (SA) 来确定哪个扇区将被覆盖以及哪个扇区的保护状态在字位置 2h 处报告。详见表 7-19。

ID-CFI AS0 允许以下活动：

- 读取 ID-CFI AS0，使用与 entry 命令中使用的相同的 SA。
- 在扇区地址 (SA) +2h 处读取扇区保护状态。位置 2h 提供关于扇区的当前对所定位的扇区的保护状态的易失性信息。在位置 2h 处的字的位 0 标示出了与寻址扇区相关的 PPB 和 DYB 位的逻辑 NAND，使得如果扇区被任一保护对于所示状态受保护的扇区，PPB=0 或 DYB=0 位。(1=受保护扇区，0=未受保护扇区)。此保护状态仅针对输入 ID-CFI AS0 时选择的 SA 显示。读取其他 SA 会提供未定义的数据。要读取不同的 SA 保护状态，必须使用 AS0 退出命令，然后使用新 SA 再次输入 ID-CFI AS0。

■ AS0 退出。

下面是一个使用 CFI 进入和退出函数的 C 源代码示例。

```
/* Example: CFI Entry command */
*( (UINT16 *)base_addr + 0x555 ) = 0x0098; /* write CFI entry command */
/* Example: CFI Exit command */
*( (UINT16 *)base_addr + 0x000 ) = 0x00F0; /* write cfi exit command */
```

7.5.4.12.2 状态寄存器 ID

状态寄存器 AS0 包含用于嵌入式算法的易失性状态寄存器的单个字。当发出状态寄存器读取命令时，捕获当前状态（通过上升的 WE# 的上升沿）进入寄存器，并输入 AS0。状态寄存器内容出现在所有字位置。第一次读取访问退出状态寄存器 AS0（具有 CE# 或 OE# 的上升沿）并返回到发出状态寄存器读取命令时正在使用的地址空间映射。写命令不会退出状态寄存器 AS0 状态。

7.5.4.12.3 安全硅域 AS0

系统可以通过在读取模式期间发出安全硅区域进入命令序列来访问安全硅区域。该进入命令使用命令

中的扇区地址 (SA) 来确定将覆盖哪个扇区。安全硅区域 AS0 允许以下活动：

- 阅读安全硅区域。
- 允许使用字或写缓冲器编程命令对客户安全硅区域进行编程。使用传统安全硅退出命令的
- AS0 退出软件向后兼容性。
- AS0 退出使用所有 AS0 的通用退出命令——一致退出方法的替代方案。

7.5.4.12.4 锁寄存器 AS0

系统可以通过在读取模式期间发出锁定寄存器进入命令序列来访问锁定寄存器。该进入命令不使用来自进入命令的扇区地址。锁寄存器出现在设备地址空间中的字位置 0 处。设备地址空间中的所有其他位置都是未定义的。

锁定寄存器 AS0 允许以下活动：

- 读取锁定寄存器，使用设备地址位置 0。
- 使用修改后的字编程命令对客户锁寄存器进行编程。
- AS0 使用传统命令退出设置退出命令以实现向后软件兼容性。
- AS0 退出使用所有 AS0 的通用退出命令——一致退出方法的替代方案。

7.5.4.12.5 密码 AS0

密码 AS0 系统可以在读取模式下通过发出密码输入命令序列来访问密码 AS0。该进入命令不使用来自进入命令的扇区地址。密码出现在设备地址空间中的字位置 0 到 3 处。设备地址空间中的所有其他位置都是未定义的。

密码 AS0 允许以下活动：

- 读取密码，使用设备地址位置 0 到 3。
- 使用修改后的字编程命令对密码进行编程。
- 用密码解锁命令解锁 PPB 锁位。
- AS0 出口使用传统命令设置退出命令以实现向后软件兼容性。
- AS0 退出使用所有 AS0 的通用退出命令——一致退出方法的替代方案。

7.5.4.12.6 PPB AS0

系统可以通过在读取模式期间发出 PPB 进入命令序列来访问 PPB AS0。该进入命令不使用来自进入命令的扇区地址。出现扇区的 PPB 位在扇区中所有字位置的位 0 中。PPB AS0 允许以下活动：

- 读取扇区中任何字的位 0 中扇区的 PPB 保护状态。
- 使用修改的字编程命令对 PPB 位进行编程。
- 用 PPB 擦除命令擦除所有 PPB 位。
- AS0 退出使用通用命令集退出命令用于向后软件兼容性。
- AS0 退出使用所有 AS0 的通用退出命令——一致退出方法的替代方案。

7.5.4.12.7 PPB 锁 AS0

系统可以通过在读取模式期间发出 PPB 锁 AS0 进入命令序列来访问 PPB 锁 AS0。该进入命令不使用来自进入命令的扇区地址。全局 PPB 锁位它出现在设备中所有字位置的位 0 中。

PPB 锁 AS0 允许以下活动：

- 读取设备地址空间中任何字的位 0 中的 PPB 锁保护状态。
- 使用修改后的字编程命令设置 PPB 锁定位。
- ASO 退出使用通用命令集退出命令用于向后软件兼容性。
- ASO 退出使用所有 ASO 的通用退出命令——一致退出方法的替代方案。

7.5.4.12.8 DYB ASO

系统可以通过在读取模式期间发出 DYB 进入命令序列来访问 DYB ASO。该进入命令不使用来自进入命令的扇区地址。出现扇区的 DYB 位在扇区中所有字位置的位 0 中。DYB ASO 允许以下活动：

- 读取扇区中任何字的位 0 中扇区的 DYB 保护状态。
- 使用修改后的字编程命令设置 DYB 位。
- 使用修改后的字编程命令清除 DYB 位。
- ASO 退出使用传统命令集退出命令向后软件兼容性。
- ASO 退出使用所有 ASO 的通用退出命令——一致退出方法的替代方案。

7.5.4.12.9 软件（命令）复位/ASO 退出

软件复位是命令集的一部分，它也使 EAC 返回旁路状态，必须用于以下情况：

- 退出 ID/CFI 模式
- 清除超时位 (DQ5)，用于超时时的数据轮询

软件重置不影响 EA 模式。一旦编程或擦除开始，重置命令将被忽略，直到操作完成。软件复位不影响输出；它主要服务于从 ASO 模式或从失败的编程或擦除操作返回到读取模式。

软件复位可能导致从无效命令序列可能导致的未定义状态返回到读取模式。然而，可能需要硬件复位以从 so 返回到正常操作我未定义的状态。

没有软件重置延迟要求。复位命令在 t_{WPH} 期间执行。

7.5.5 状态监测

监控 EA 状态有三种方法。HWD29S01GPBGA64M3 使用称为数据轮询和就绪/忙碌# (RY/BY#) 信号的方法。另一种方法是读取状态寄存器。

7.5.5.1 状态寄存器

编程和擦除操作的状态由单个 16 位状态寄存器提供。通过写入状态寄存器读取命令，然后进行读取访问，状态为接收器。当统计发出状态寄存器读取命令，当前状态（通过 WE# 的上升沿）被捕获到寄存器中并输入 ASO。状态寄存器的内容在完整的内存地址空间。在状态寄存器 ASO 中时的任何有效读取（CE# 和 OE# 低）访问将退出 ASO（对于 t_{CEPH}/t_{OEPH} 时间，CE# 或 OE# 的上升沿）并返回到发出状态寄存器读取命令时正在使用的地址空间映射。

状态寄存器包含与最近完成的嵌入式算法（EA）的结果（成功或失败）相关的位：

- 擦除状态（位 5）、
- 程序状态（位 4）、
- 写缓冲区中止（位 3）、
- 扇区锁定状态（位 1）、
- RFU（位 0）。

以及，与任何进程中 EA 的当前状态相关的位：

- 设备忙（位 7），

- 擦除挂起（位 6），
- 程序挂起（位 2），

当前状态位指示 EA 是进行中、挂起还是完成。

保留高位 8 位(位 15: 8)。这些具有未定义的高值或低值，可以从一个状态读取变为另一个状态。这些位应该被视为不关心并被任何软件忽略正在阅读状态。

如果状态寄存器位 3=0，则软复位命令将清除状态寄存器的 0 位[5, 4, 1, 0]。它不会影响当前状态位。清除状态寄存器命令将清除为 0 是状态寄存器的结果相关位，但不会影响当前状态位。

表 7-35 状态寄存器

位#	15:8	7	6	5	4	3	2	1	0
位描述	保留	设备就绪位	擦除暂停状态位	擦除状态位	编程状态位	写缓冲区中止状态位	程序挂起状态位	扇区锁定状态位	保留
位名字		DRB	ESSB	ESB	PSB	WBASB	PSSB	SLSB	
复位状态	X	1	0	0	0	0	0	0	0
忙碌的状态	无效的	0	无效的	无效的	无效的	无效的	无效的	无效的	无效的
就绪状态	X	1	0 =持起状态下无Erase 1 =在暂停中擦除	0 = Erase 成功 1 = Erase 失败	0 =编程成功 1 =编程失败	0 =编程未中止 1 =编程在执行写入缓冲区命令时中止	0 =否暂停中的程序 1 =编程暂停	0 =运行时扇区未锁定 1 =扇区锁定错误	X

注 1：位 15 到 8 和 0 被保留以供将来使用，并且可以显示为 0 或 1。检查状态时，这些位应该被忽略（屏蔽）。

注 2：当中没有嵌入算法时，位 7 为 1 设备的进展。

注 3：只有当位 7 为 1 时，位 6 至 1 才有效。

注 4：通过冷复位或热复位将所有位置于其复位状态。

注 5：位 5、4、3 和 1 被清除状态寄存器命令或复位命令清除为 0。

注 6：在发出擦除挂起命令时，用户必须继续读取状态，直到 DRB 变为 1。

注 7：ESSB 通过擦除恢复命令清除为 0。

注 8：ESB 反映最近一次擦除操作的成功或失败。

注 9：PSB 反映最近程序操作的成功或失败。

注 10：在擦除挂起时，对挂起的扇区编程将导致程序失败，并将程序状态位设置为 1。

注 11：在发出编程挂起命令时，用户必须继续读取纹身直到 DRB 变成 1。

注 12：通过程序恢复命令将 PSSB 清除为 0。

注 13：SLSB 表示程序或擦除操作失败，因为扇区被锁定。

注 14：SLSB 反映了统计数据最近的程序或擦除操作。

7.5.5.2 数据轮询状态

在活动的嵌入式算法期间，EAC 切换到数据轮询 AS0，以向任何读取访问显示 EA 状态。状态信息的单个字别名了设备地址空间的所有位置。在状态字中，有几个位用于确定 EA 的状态。这些被称为 DQ 位，因为它们在 EA 进行时的读取访问期间出现在数据总线上。DQ 位 15 至 8、DQ4 和 DQ0 被保留并提供未定义的数据。状态监控软件必须屏蔽保留位并将其视为不关心。表 7-30 和以下小节描述剩余位的功能。

该器件提供多个位来确定编程、擦除操作或密码解锁的状态。以下小节介绍了 DQ1、DQ2、DQ3、DQ5、DQ6 和 DQ7 的功能。

7.5.5.2.1 DQ7: Data#轮询

Data#轮询位 DQ7 向主机系统指示嵌入式算法是正在进行中还是已经完成。Data#轮询在编程或擦除命令序列中最后一个 WE#脉冲的上升沿之后有效。请注意，Data #轮询仅对写入缓冲区编程期间写入缓冲区页中的最后一个字/字节有效。读取 Data #轮询状态除写入缓冲区页中要编程的最后一个字/字节之外的任何字上，将返回错误的状态信息。

在嵌入式编程算法期间，器件在 DQ7 上输出编程到 DQ7 的数据位的补码。此 DQ7 状态也适用于擦除挂起期间的编程。当嵌入式编程算法完成后，设备输出编程到最后编程的字/字节的位 7 的数据位。在编程挂起的情况下，设备只允许读取阵列数据。如果一个编程地址落在受保护扇区内，DQ7 上的 Data #轮询活动约 20 μ s，然后设备返回读取阵列数据。

在嵌入式擦除或空白检查算法期间，Data#轮询在 DQ7 上产生 0。当算法完成时，或者如果设备进入擦除挂起模式，Data#轮询在 DQ7 上会产生 1。这类似于针对嵌入式编程算法描述的补码/真数据输出：擦除功能将扇区中的所有位改变为 1；在此之前，设备输出补码或“0”。系统必须在选定用于擦除的扇区内提供地址，以便读取 DQ7 上的有效状态信息。

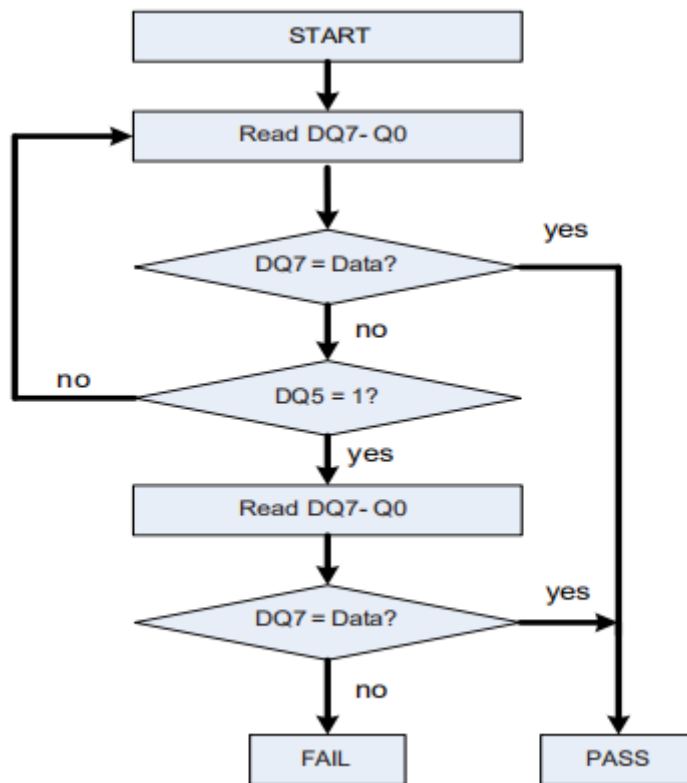
写入擦除命令序列后，如果选择用于擦除的扇区受到保护，则 DQ7 上的 Data#轮询活动约 100 μ s，则设备返回读取阵列数据。

当系统检测到 DQ7 已经从补码变为真数据时，它可以在随后的读取周期中读取 DQ15-DQ0 处的有效数据。这是因为 DQ7 可能与 DQ6-DQ0 异步变化而输出使能(OE#)被断言为低电平。这如图 7-22 所示。表 7-36 显示了 DQ7 上 Data #轮询的输出。图 7-31 显示了写入缓冲区中使用的 Data #轮询算法编程。

有效的 DQ7 数据轮询状态只能从以下位置读取：

- 加载到写入缓冲区中用于写入缓冲区编程操作的最后一个字/字节的地址；
- 单字编程操作的位置；
- 或扇区中的一个位置正被擦除或空白检查；
- 或芯片擦除期间任何扇区的位置。

图 7-49 Data#轮询算法



注 1：即使 DQ5=1，也应重新检查 DQ7，因为 DQ7 可能与 DQ5 同时变化。

7.5.5.2.2 DQ6：切换位 I

DQ6 上的切换位 I 指示嵌入式编程或擦除算法是否正在进行或完成，或者设备是否已进入编程挂起或擦除挂起模式。切换位可以在任何地址读取，并且在命令序列中最后一个 WE# 脉冲的上升沿之后（在编程或擦除操作之前）有效。

在嵌入式编程或擦除算法操作期间，对任何地址的连续读取周期导致 DQ6 切换。（系统可以使用 OE# 或 CE# 来控制读取周期）。当操作完成，DQ6 停止切换。

写入擦除命令序列后，如果选择用于擦除的扇区受到保护，则 DQ6 切换约 100μs，然后 EAC 返回待机（读取模式）。如果所选扇区不受保护的扇区，则嵌入式擦除算法擦除未受保护的扇区。

系统可以一起使用 DQ6 和 DQ2 来确定扇区是主动擦除还是擦除挂起。当设备正在主动擦除时（即，嵌入式擦除算法正在进行中），DQ6 切换。当设备进入程序暂停模式或擦除暂停模式时，DQ6 停止切换。然而，系统还必须使用 DQ2 来确定哪些扇区正在擦除或擦除待定。或者，系统可以使用 DQ7（参见 DQ7：Data# 轮询）。

DQ6 还在擦除挂起程序模式期间进行切换，并在嵌入式程序算法完成后停止切换。表 7-36 显示了 DQ6 上切换位 I 的输出。图 7-50 显示了 toggle 位算法的流程图，读取切换位 DQ6/DQ2 解释了该算法。图 7-50 显示了切换位时序图。图 7-31 显示了 DQ2 和 D 之间的差异图形形式的 Q6。另请参见 DQ2：切换位 II。

7.5.5.2.3 DQ3：扇区擦除定时器

在写入扇区擦除命令序列之后，系统可以读取 DQ3 以确定擦除是否已经开始。有关更多详细信息，请参见扇区擦除。

在写入扇区擦除命令后，系统应读取 DQ7（Data#轮询）或 DQ6（切换位 I）的状态，以确保设备已接受命令序列，然后读取 DQ3。如果 DQ3 为 1，则嵌入式擦除算法已经开始；在擦除操作完成之前，所有其他命令（擦除挂起除外）都将被忽略。表 7-36 显示了 DQ3 相对于其他状态位。

7.5.5.2.4 DQ2：切换位 II

当与 DQ6 一起使用时，DQ2 上的切换位 II 指示特定扇区是否正在主动擦除（即，嵌入式擦除算法正在进行），或者该扇区是否处于擦除暂停状态。切换位 II 在命令序列中最后一个 WE#脉冲的上升沿之后有效。

当系统读取选定用于擦除的扇区内的地址时，DQ2 切换。（系统可以使用 OE#或 CE#来控制读取周期）。但是 DQ2 不能区分扇区是否正在主动擦除或被擦除挂起。通过比较，DQ6 指示设备是在主动擦除，还是处于擦除挂起状态，但不能区分扇区是否被选择用于擦除。因此，扇区和模式信息都需要状态位。参考表 7-36 比较 DQ2 和 DQ6 的输出。图 7-49 以流程图的形式显示了切换位算法，以及读翻转位 DQ6/DQ2 解释了该算法。图 7-50 显示了切换位时序图。图 7-31 以图形形式显示了 DQ2 和 DQ6 之间的差异。

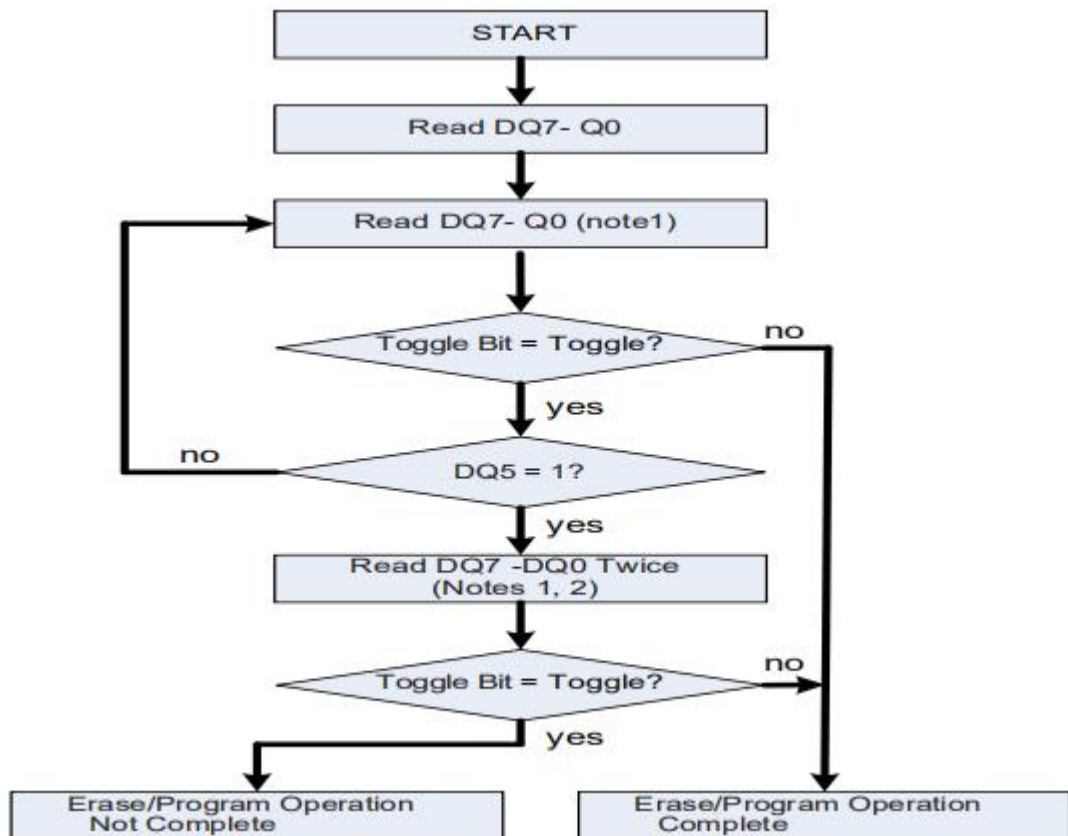
7.5.5.2.5 读取翻转位 DQ6/DQ2

下面的讨论参见图 7-49。每当系统最初开始读取切换位状态时，它必须连续读取 DQ7-DQ0 至少两次，以确定切换位是否正在翻转。通常，系统将在第一次读取之后记录并存储翻转位的值。在第二次读取之后，系统将翻转位的新值与前一个值进行比较值。如果翻转位未切换，则设备已完成编程或擦除操作。系统可以在下一个读取周期读取 DQ15-DQ0 上的阵列数据。

然而，如果在最初的两个读取周期之后，系统确定切换位仍在翻转，则系统还应注意 DQ5 的值是否高（参见 DQ5：超出定限）。如果是，则系统应再次确定翻转位是否正在翻转，因为翻转位可能在 DQ5 变为高电平时已经停止翻转。如果翻转位不再翻转，设备已成功完成编程或擦除操作。如果仍在翻转，则设备未成功完成操作，系统必须写入复位命令返回读取阵列数据。

剩下的场景是，系统最初确定切翻转正在切换，并且 DQ5 没有变高。系统可以通过连续的步骤继续监测翻转位和 DQ5。读取周期，确定上一段所述的状态。或者，它可以选择执行其他系统任务。在这种情况下，系统必须从算法返回时确定操作状态。

图 7-50 翻转位程序



注 1: 读取切换位两次, 以确定它是否在切换。见正文。

注 2: 重新检查切换位, 因为当 DQ5 变为 1 时, 它可能会停止切换。见正文。

7.5.5.2.6 DQ5: 超过定时限值

DQ5 指示编程或擦除时间是否已超过指定的内部脉冲计数限制。在这些条件下, DQ5 产生 1。这是指示编程故障或擦除周期未成功完成。系统必须发出复位命令, 使设备返回读取阵列数据。

发生超时, 软件必须发送复位命令以清除超时位 (DQ5) 并将 EAC 返回到读取阵列模式。在这种情况下, 闪存可能会继续发送复位命令后, 通信忙 2 μ s。

7.5.5.2.7 DQ1: 写入缓冲区中止

DQ1 指示写入缓冲区操作是否中止。在这些条件下, DQ1 产生 1。系统必须发出 Write-to-Buffer-Abort-Reset 命令序列以将 EAC 返回到旁路 (读取模式) 和状态寄存器失败位被清除。有关更多详细信息, 请参见写入缓冲区编程。

表 7-36 数据轮询状态

操作		DQ7 (注2)	DQ6	DQ5 (注1)	DQ3	DQ2 (注2)	DQ1 (注4)	RY / #
标准模式	嵌入式编程算法	DQ7 #	切换	0	N/A	没有切换	0	0
	擦除扇区内读取	0	切换	0	1	切换	N/A	0
	Reading Outside擦除扇区	0	切换	0	1	没有切换	N/A	0
程序暂停模式(注3)	程序暂停扇区内的读取	无效的 (不允许)	无效的 (不允许)	无效的 (不允许)	无效的 (不允许)	无效的 (不允许)	无效的 (不允许)	1
	在非程序暂停扇区内读取	数据	数据	数据	数据	数据	数据	1
Erase Suspend Mode	在Erase Suspend扇区内读取	1	没有切换	0	N/A	切换	N/A	1
	在非Erase Suspend扇区内读取	数据	数据	数据	数据	数据	数据	1
	在非Erase Suspend扇区内编程	DQ7 #	切换	0	N/A	N/A	N/A	0
写入-缓冲(注4, 5)	忙碌的状态	DQ7 #	切换	0	N/A	没有切换	0	0
	超过计时限制	DQ7 #	切换	1	N/A	N/A	0	0
	ABORT状态	DQ7 #	切换	0	N/A	N/A	1	0

注 1：当嵌入式程序或嵌入式擦除操作超过最大定时限制时，DQ5 切换到“1”。有关更多信息，请参见 DQ5：超出计时限制。

注 2：DQ7 和 DQ2 要求读取状态信息时的有效地址。有关更多详细信息，请参见相应的小节。

注 3：数据对于程序挂起行中的地址无效。除编程挂起线之外的所有地址可以读取有效数据。

注 4：DQ1 表示写入缓冲区编程操作期间写入缓冲区中止状态。

注 5：仅适用于编程操作。

注 6：密码解锁操作时，DQ7 设置为在密码解锁命令的最后一个字/字节中的 DQ7 位的补码，此操作成功则 DQ6 翻转，RY/BY#将保持较低

7.5.6 错误类型和清除原则

嵌入式操作状态方法报告的错误有三种类型。根据错误类型的不同，报告的状态和清除错误状态的过程也不同。随之是错误状态的清除：

- 如果在错误发生之前输入了 AS0，则设备仍在 AS0 中输入，等待 AS0 读取或命令写入。
- 如果在错误发生之前挂起擦除，设备将返回到擦除挂起状态等待闪存阵列读取或命令写入命令。
- 否则，设备将处于旁路状态，等待闪存阵列读取或命令写入。

7.5.6.1 嵌入式操作错误

如果在嵌入式操作（编程、擦除、空白检查或密码解锁）期间发生错误，设备（EAC）将保持繁忙状态。RY/BY#输出保持低，数据轮询状态继续为覆盖在所有地址位置上，并且状态寄存器显示就绪，具有有效的

状态位。设备将保持忙碌状态，直到主机系统状态监视和错误状态已清除。

在嵌入算法错误状态期间，数据轮询状态将显示以下内容：

■DQ7 是在密码解锁命令的情况下，加载到写入缓冲区的最后一个字/字节或密码的最后一个字中的 DQ7 位的反转。对于擦除或空白检查，DQ7=0 为故障

■DQ6 继续切换

■DQ5=1；嵌入式操作的失败

■DQ4 是 RFU，应视为不关心（屏蔽）

■DQ3=1，以指示正在进行的嵌入式扇区擦除

■DQ2 继续翻转，独立于用于读取状态

■DQ1=0 的地址；写缓冲区中止错误

■DQ0 是 RFU，应视为不在乎（屏蔽）

在嵌入算法错误状态期间，状态寄存器将显示以下内容：

■SR[7]=1；显示的有效状态

■SR[6]=X；在擦除或空白检查错误时，可能会也可能不会在 EA 错误

■SR[5]=1 擦除或空白检查错误；否则=0

■SR[4]=1 程序或密码解锁错误；否则=0

■SR[3]=0；写缓冲区中止

■SR[2]=0；程序挂起

■SR[1]=0；受保护扇区

■SR[0]=X；RFU，视为不在乎（被屏蔽）

当检测到嵌入式算法错误状态时，有必要清除错误状态，以便返回正常操作，RY/BY#为高，为新的读取或命令写入做好准备。错误状态可以通过写入来清除：

■复位命令

■状态寄存器清除命令

在嵌入式算法错误状态期间接受的命令有：

■状态寄存器读取

■复位命令

■状态寄存器清除命令

7.5.6.2 保护错误

如果嵌入式算法试图改变受保护区域内的数据（编程或擦除受保护扇区或 OTP 区域），则设备(EAC)将忙碌 20 至 100μs 的时间段，然后返回到正常运行。在忙碌期间，RY/BY#输出保持低电平，数据轮询状态继续覆盖在所有地址位置上，并且状态寄存器显示 not ready，其中无效状态位（SR[7]=0）。

在保护错误状态忙碌期间，数据轮询状态将显示以下内容：

■DQ7 是加载到写入缓冲区的最后一个字/字节中的 DQ7 位的反转。对于擦除失败，DQ7=0

■DQ6 继续翻转，与用于读取状态的地址无关

■DQ5=0；指示在忙碌期间嵌入式操作没有失败

■DQ4 是 RFU，应被视为不关心（被屏蔽）

■DQ3=1 指示正在进行的嵌入式扇区擦除

■DQ2 con 继续翻转，独立于用于读取状态

■DQ1=0 的地址；写缓冲区中止错误

■DQ0 是 RFU，应视为不关心（被屏蔽）

在保护错误状态忙期间接受的命令有：

■状态寄存器读取

当忙碌时段结束时，设备恢复正常操作，数据轮询状态不再覆盖，RY/BY#为高，状态寄存器显示就绪，

具有有效的状态位。设备准备好对闪存阵列读取或写入新命令。

在保护错误状态忙碌期之后，状态寄存器将显示以下内容：

- SR[7]=1；显示的有效状态
- SR[6]=X；保护错误忙周期后可能会也可能不会擦除挂起
- SR[5]=1 擦除错误，否则=0
- SR[4]=1 编程错误，否则=0
- SR[3]=0；编程未中止
- SR[2]=0；无编程挂起
- SR[1]=1；由于试图更改受保护位置
- SR[0]=X 而导致的错误；RFU，视为不关心（被屏蔽）

在保护错误状态忙期间之后接受的命令有：

- 任何命令

7.5.6.3 写缓冲区中止

如果在写入缓冲区命令期间发生错误，则设备(EAC)保持忙碌。RY/BY#输出保持低，数据轮询状态继续覆盖在所有地址位置上，并且状态寄存器显示就绪，状态位有效。设备保持忙碌状态，直到主机系统状态监视检测到错误状态并清除错误状态。

在写入缓冲区中止(WBA)错误状态期间，数据轮询状态将显示以下内容：

- DQ7 是加载到写缓冲区的最后一个字/字节中的 DQ7 位的反转
- DQ6 继续翻转，与用于读取状态的地址无关
- DQ5=0；表示没有失败编程操作。WBA 是在编程操作开始之前写入缓冲区命令输入的值中的错误
- DQ4 是 RFU，应视为不在乎（被屏蔽）

■DQ3 在编程操作后不关心，因为没有正在进行擦除。如果写缓冲器编程操作是在擦除操作被挂起之后开始的，则 DQ3=1。如果没有擦除操作正在进行中，然后 DQ3 是一个不在乎的，应该被屏蔽。

■DQ2 在编程操作后不切换，因为没有进行擦除。如果写缓冲器编程操作是在擦除操作已被挂起，则 DQ2 将在擦除操作被挂起的扇区中切换，而不在任何其他扇区中切换。如果没有正在进行的擦除操作，则 DQ2 是不在乎，应该被屏蔽。

- DQ1=1：写缓冲区中止错误
- DQ0 是 RFU，应视为不关心（被屏蔽）

在嵌入算法错误状态期间，状态寄存器将显示以下内容：

- SR[7]=1；显示的有效状态
- SR[6]=X；在 WBA 错误状态
- SR[5]=0 期间，可能会也可能不会被擦除暂停；擦除成功的
- SR[4]=1；编程相关误差
- SR[3]=1；写缓冲区中止
- SR[2]=0；没有编程挂起
- SR[1]=0；操作期间扇区未锁定
- SR[0]=X；RFU，视为不关心（被屏蔽）

当检测到 WBA 错误状态时，有必要清除错误状态以返回正常操作，RY/BY#为高，准备好进行新的读取或命令写入。错误状态可以通过写入：

- 写缓冲区中止复位命令
 - 清除状态寄存器并恢复正常操作
- 状态寄存器清除命令

在嵌入式算法错误状态期间接受的命令有：

- 状态寄存器读

■写缓冲区中止复位命令

■状态寄存器清除命令

7.5.7 嵌入式算法性能表

表 7-37 嵌入式算法特性 (−55℃—125℃)

参数	参数备注	典型值 (注 1)	最大值 (注 2)	单位	备注
扇区擦除时间	128 kbyte	275	1100	ms	包括擦除前预编程
全芯片擦除	1G	285		s	
单字节/字编程时间		30	200	μs	
缓冲区编程 (字节模式)	64-byte	92	460	μs	
	128-byte	117	600		
	256-byte	171	900		
缓冲区编程 (字模式)	64-word	117	600		
	128-word	171	900		
	256-word	285	1500		
每字的有效写缓冲区编程操作	512-byte	1.11	—	μs	
每字的扇区写缓冲区编程操作	全缓冲区编程	72	269	ms	注 5
擦除挂起/擦除恢复	t_{ESL}	—	50	μs	
编程挂起/编程恢复	t_{ESL}	—	50	μs	
擦除恢复到下一次擦除挂起	t_{ERS}	100	—	μs	最小 60ns, 但擦除完成需要 ≥ 典型周期
编程恢复到下一次编程挂起	t_{PRS}	100	—	μs	最小 60ns, 但编程完成需要 ≥ 典型周期
空白检查		18	25	ms	

注 1: 典型的编程和擦除时间假设以下条件: 25℃、3.0 V VCC、10000 次循环和随机数据模式。

注 2: 在 125℃的最坏情况条件下, VCC=2.70 V、100000 个周期和随机数据模式。

注 3: 有效的写入缓冲区规范基于 512 字节的写入缓冲区操作。

注 4：在嵌入式擦除算法的预编程步骤中，扇区和芯片擦除之前所有字都被编程到 0000h。

注 5：系统级开销是为程序命令执行总线周期序列所需的时间。

7.5.8 写

7.5.8.1 异步写入

当 WE# 在 CE 为低电平之后变为低电平时，存在从一种读取状态到写入状态的转变。如果 WE# 在 CE# 变低之前为低，则存在从旁路状态直接到在没有开始读取访问的情况下写入状态的传输。

当 CE# 低、OE# 高且 WE# 低时，开始写入数据传输。注意，OE# 和 WE# 不应同时低，以确保主机系统和内存之间没有数据总线争用。当满足异步写入周期定时要求时，WE# 可以变为高电平以捕获 EAC 命令存储器中的地址和数据值。

地址由 WE# 或 CE# 的下降沿捕获，以较晚出现者为准。数据由 WE# 或 CE# 的上升沿捕获，以较早发生者为准。

当 CE# 在 WE# 变为低电平之前为低电平并且在 WE# 变为高电平之后保持低电平时，该访问被称为 WE# 控制的写入。当 WE# 为高且 CE# 为高时，有到旁路状态的传输。如果 CE# 保持低，WE# 变为高，过渡到输出禁用状态的读取。

当 WE# 在 CE# 变为低电平之前为低电平并且在 CE# 变为高电平之后保持低电平时，该访问被称为 CE# 控制写入。CE# 控制的写入传输到待机状态。

如果 WE# 在 CE# 变为低电平之前为低电平，则由 CE# 变为低电平开始写入传输。如果 CE# 变为高电平后 WE# 为低电平，则地址和数据由 CE# 的上升沿捕获。这种情况参考 CE# 控制写入状态传输。

写后是读访问，其中 CE# 在访问之间保持低电平，需要地址改变来启动下一个读访问。

背靠背访问，其中 CE# 在访问之间保持低，需要地址改变来发起第二次访问。

EAC 命令存储器阵列不能被主机系统读取并且没有 AS0。EAC 检查每个写入传输中的地址和数据，以确定写入是否是合法命令序列的一部分。。当合法命令序列完成时，EAC 将启动适当的 EA。

7.5.8.2 写脉冲“毛刺”保护

WE# 上小于 3 ns（典型值）的噪声脉冲不会启动写入周期。

7.5.8.3 逻辑抑制

通过将 OE# 保持在 V_{IL} ，或将 CE# 保持在 V_{IH} ，或将 WE# 保持在 V_{IH} ，来抑制写入循环。要启动写入周期，CE# 和 WE# 必须为低 (V_{IL})，而 OE# 为高 (V_{IH})。

7.5.9 节能模式

7.5.9.1 接口旁路待机

当主机未选择设备进行数据传输时，旁路待机是接口的默认低功耗状态（CE#=高）。在此状态下，所有输入都被忽略，除 RY/BY# 之外的所有输出都被忽略，是高阻抗。RY/BY# 是 EAC 的直接输出，不受主机接口控制。

8 使用注意事项

器件必须采取防静电措施进行操作。

该器件操作注意事项如下：

- a) 器件应在防静电的工作台上操作；
- b) 试验设备和器具应接地；
- c) 不能触摸器件引线；
- d) 器件应存放在导电材料制成的容器中；
- e) MOS 区域内避免使用能引起静电的塑料、橡胶或丝织物；
- f) 若可行，相对湿度保持在 50%以上。

9 订货信息

表 9-1 产品订货信息

序号	产品型号	封装形式	封装材料	质量保证等级	引脚材料	重量（g）	详细规范
1	HWD29S01GPBGA64M3	PBGA	塑料	N1 级	锡银铜合金	0.29g	Q/HWD 50594-2024

注1：N/N1级器件满足GJB 7400A 《合格制造厂认证用半导体集成电路通用规范》N/N1级的筛选要求

注2：产品订货信息为我司已有产品或确定研制的器件，可根据用户需求研制其他封装形式的器件。