

无锡泰连芯科技有限公司

TLX29LV512M型
并行FLASH存储器电路

2024年06月

目录

1 产品概述	3
1.1 产品简介	3
1.2 特性	3
1.3 产品引脚定义与排布	4
1.4 原理框图	6
2 软件接口	7
2.1 地址空间	7
2.2 数据保护	10
2.3 算法控制器	12
2.4 命令编码表	18
3 电特性	21
3.1 绝对最大额定值	21
3.2 推荐工作条件	21
3.3 DC 参数	22
测试条件	22
3.4 时序参数	22
3.5 上电和掉电	26
4 硬件接口	28
4.1 接口定义	28
4.2 接口状态	28
5 说明事项	29
5.1 运输与储存	29
5.2 开箱与检查	29
5.3 使用操作规程及注意事项	29
5.4 质量保证	29
6 封装尺寸	30
6.1 外形尺寸图	30
6.2 订购信息	31
6.3 推荐焊装工艺	31

1 产品概述

1.1 产品简介

TLX29LV512M 型并行存储器是一款工作于 3.3V 电压，容量为 512Mb 的 NOR Flash 存储器。器件支持 16 位 的数据总线，每次读操作可读出 16 位数据，随机读访问可在 110ns 内完成。芯片支持页模式读访问，在每次随机读访问过程中，芯片内部控制器都会将一页 32 字节的数据全部读出，在后续的读访问中，如果页地址保持不变仅改变页内的字地址，那么芯片可以在 20ns 内完成读操作。

擦除操作后 Flash 单元的状态为逻辑“1”，编程操作可将 Flash 单元的状态从“1”变为“0”，只有擦除操作才能将 Flash 单元的状态从“0”变为“1”，扇区擦除命令的最小目标单元为 128K 字节的扇区，为了提高效率，用户可使用芯片擦除命令将整个芯片擦除。字编程命令的最小目标单元为 2 字节的字，为了提高效率，用户也可使用行编程命令一次性将 512 字节数据写入芯片。

为了保护 Flash 中存储的数据不被有意或无意修改，芯片内部实现了一系列数据保护方法，包括上电/掉电数据保护、硬件写保护、软件写保护和高级扇区保护。

芯片内部控制电路由接口控制器和算法控制器构成，接口控制器负责监测输入管脚的状态和检测用户输入的有效命令，检测到有效命令后，接口控制器将命令及数据交给算法控制器，然后算法控制器根据具体算法的要求产生复杂的控制信号来改变非易失存储器中数据，从而实现用户命令要求的操作。

1.2 特性

- 单电压 (VCC) 读/写/擦操作 (2.7V-3.6V)
- 16 位数据总线
- 32 字节页模式读操作
- 512 字节行模式写操作
- 统一大小的扇区结构 (128KByte)
- 支持编程/擦除中断和恢复
- 支持高级扇区保护
- 可通过状态寄存器、数据查询和专用管脚查询器件状态
- 提供额外 1024 字节可写保护的OTP 存储区
- 支持 CFI 接口
- 擦/写循环次数：10 万次
- 数据保存时间：20 年

- BGA64 封装
- 工作温度范围: -55°C~+125°C;
- ESD 等级: 2 级(2000V)
- 湿度敏感等级: 3 级
- 质量等级: 军温级&N1级

1.3 产品引脚定义与排布

引出端排列见图 1-1。

TOP VIEW

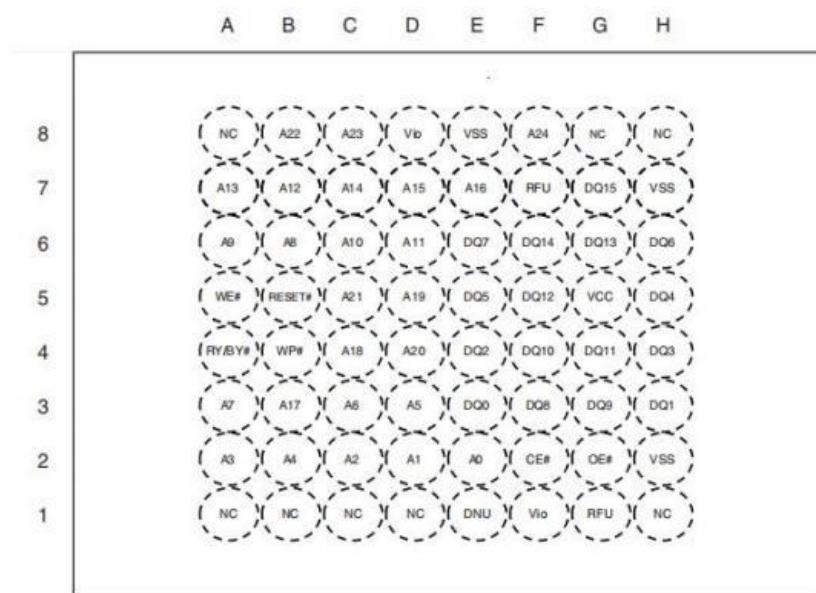


图 1-1 引出端排列

TLX29LV512M 引出端功能见表 1-1。

表 1-1 引出端功能表

引出端序号	符号	功能
A2	A3	地址信号输入
A3	A7	地址信号输入
A4	RY/BY#	芯片状态信号输出
A5	WE#	写使能信号输入
A6	A9	地址信号输入
A7	A13	地址信号输入

引出端序号	符号	功能
B2	A4	地址信号输入
B3	A17	地址信号输入
B4	WP#	写保护信号输入
B5	RESET#	硬件复位信号输入
B6	A8	地址信号输入
B7	A12	地址信号输入
B8	A22	地址信号输入
C2	A2	地址信号输入
C3	A6	地址信号输入
C4	A18	地址信号输入
C5	A21	地址信号输入
C6	A10	地址信号输入
C7	A14	地址信号输入
C8	A23	地址信号输入
D2	A1	地址信号输入
D3	A5	地址信号输入
D4	A20	地址信号输入
D5	A19	地址信号输入
D6	A11	地址信号输入
D7	A15	地址信号输入
D8	VIO	IO 电源
E2	A0	地址信号输入
E3	DQ0	数据信号输入输出
E4	DQ2	数据信号输入输出
E5	DQ5	数据信号输入输出
E6	DQ7	数据信号输入输出
E7	A16	地址信号输入
E8	VSS	地
F1	VIO	IO 电源
F2	CE#	片选信号输入
F3	DQ8	数据信号输入输出

1.4 原理框图

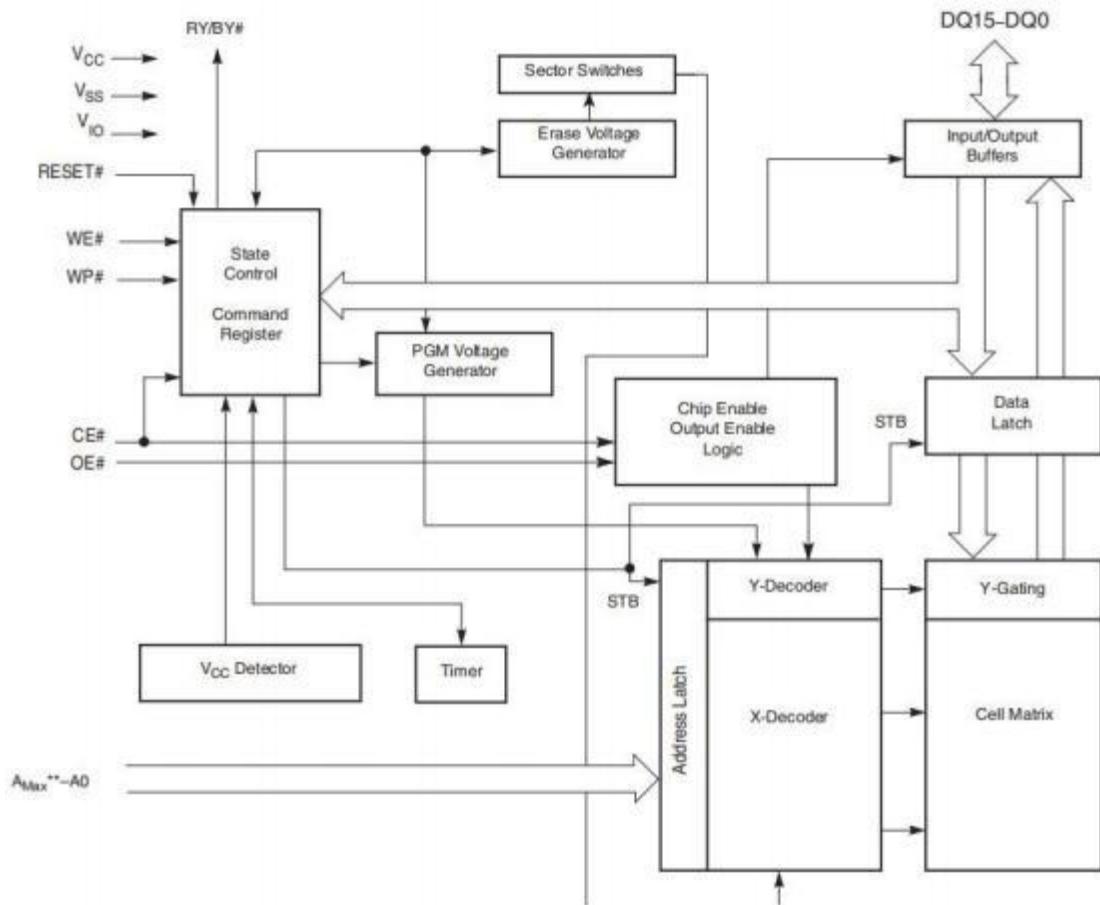


图 1-2 原理框图

2 软件接口

2.1 地址空间

芯片内部存在若干个相互独立的逻辑地址空间，在任意时刻有且仅有一个地址空间可供用户访问，称之为“当前地址空间”。上电复位或硬件复位后，内部算法控制器将“主存储空间”设置为“当前地址空间”，读写操作都被映射到用于存储用户数据的主存储区。用户可以使用命令将其它地址空间切换为“当前地址空间”以实现对其它数据或存储区的访问。

2.1.1 ID/CFI 地址空间

用户可以从“ID/CFI 地址空间”读取芯片的类型和参数信息，当“主存储空间”为“当前地址空间”且算法控制器在空闲状态时，用户可以写入命令将“ID/CFI 地址空间”切换为“当前地址空间”，具体的命令及编码参见“命令编码表”。需要注意的是，在命令中需要指定一个扇区，完成地址空间切换后，ID/CFI 存储区将被映射到指定扇区，其它未被指定的扇区的状态未定义。ID/CFI 存储区在扇区中占据的地址范围如下表所示。

地址范围	内容	读/写
(SA) + 0000h to 000Fh	ID 存储区	只读
(SA) + 0010h to 0079h	CFI 存储区	只读
(SA) + 007Ah to FFFFh	未定义	只读

地址	数据	含义
(SA) + 0000h	0001h	厂商 ID
(SA) + 0001h	227Eh	器件 ID
(SA) + 0002h	0000h/0001h	0001h:SA 被写保护，0000h: SA 未写保护
(SA) + 00003	0010h	--
(SA) + 0004h	xxxxh	保留
(SA) + 0005h	xxxxh	保留
(SA) + 0006h	xxxxh	保留
(SA) + 0007h	xxxxh	保留
(SA) + 0008h	xxxxh	保留
(SA) + 0009h	xxxxh	保留
(SA) + 000Ah	xxxxh	保留
(SA) + 000Bh	xxxxh	保留
(SA) + 000Ch	0003h	--
(SA) + 000Dh	xxxxh	保留
(SA) + 000Eh	2228h	--
(SA) + 000Fh	2201h	--

2.1.2 状态寄存器地址空间

用户可以从“状态寄存器地址空间”读取芯片内部状态寄存器的值以确定相应算法的执行状态，从该地址空间任意地址处读数据都可以读出状态寄存器的值。在用户写入的“读状态寄存器”命令 WE#上升沿，内部算法的执行状态被捕获到状态寄存器中，同时将“状态寄存器地址空间”切换为“当前地址空间”，此后用户的任何写操作都会被芯片忽略，读操作在读出状态寄存器的同时使芯片退出“状态寄存器地址空间”。

2.1.3 数据查询地址空间

“数据查询”地址空间用于供用户通过读数据来查询内部算法执行状态，从该地址空间任意地址处读数据都会读出 16 位的状态数据，用户通过查询状态数据不同位的值即可确定内部算法执行状态。任何一个可以触发内部算法执行的命令的最后一个周期 WE#上升沿后，“数据查询”地址空间被切换为“当前地址空间”，能够触发内部算法的命令包括：

- 字/行编程命令；
- 芯片/扇区擦除命令；
- 编程/擦除恢复命令；
- 空检查命令；
- 模式寄存器/密码编程命令
- PPB 编程/PPB 擦除命令；

2.1.4 OTP 地址空间

用户可以从“OTP 地址空间”读取芯片内部 1024 字节的 OTP 存储区，其中 512 字节可被厂商写保护，另外 512 字节可被用户写保护。当“主存储空间”为“当前地址空间”且芯片在空闲状态时，用户可以写入命令将“OTP 地址空间”切换为“当前地址空间”，具体的命令及编码参见“命令编码表”。需要注意的是，在地址空间切换命令中需要指定一个扇区，完成地址空间切换后，OTP 存储区将被映射到指定扇区，其它未被指定扇区的状态未定义。一个扇区的大小为 128K 字节，OTP 存储区占据扇区的低 1024 字节地址，扇区剩余地址的状态未定义。

地址范围	内容	大小
(SA) + 0000h to 00FFh	厂商锁定区	512 字节
(SA) + 0100h to 01FFh	用户锁定区	512 字节
(SA) + 0200h to FFFFh	未定义	127K 字节

在 OTP 地址空间，用户可以进行以下操作：

- 使用读操作读 OTP 存储区中存储的数据
- 使用字编程/行编程命令向OTP 存储区中写数据
- 使用退出命令退出“OTP 地址空间”

2.1.5 Lock Register 地址空间

用户可以从“Lock Register 地址空间”访问芯片内部的模式寄存器，模式寄存器是一个 16 位的 OTP 存储器，可以通过将其设置为不同的值来控制芯片的工作模式。当“主存储空间”为“当前地址空间”且芯片在空闲状态时，用户可以写入命令将“Lock Register 地址空间”切换为“当前地址空间”，具体的命令及编码参见“命令编码表”。在“Lock Register 地址空间”用户可以进行以下操作：

使用读操作从任意地址处读取模式寄存器的值

使用简化的字编程命令向模式寄存器中写数据

使用退出命令退出“Lock Register 地址空间”

2.1.6 Password 地址空间

用户可以从“Password 地址空间”访问芯片内部的密码存储区，密码存储区是一个 4x16 的 OTP 存储区，可在其中存入密码用于对 PPB 进行密码保护。当“主存储空间”为“当前地址空间”且芯片在空闲状态时，用户可以写入命令将“Password 地址空间”切换为“当前地址空间”，具体的命令及编码参见“命令编码表”。在“Password 地址空间”用户可以进行以下操作：

使用读操作从地址 0-3 处读取密码存储区中存储的数据

使用简化的字编程命令向密码存储区中写数据

使用解锁命令将 PPB Lock 寄存器设置为 1

使用退出命令退出“Password 地址空间”

2.1.7 PPB 地址空间

用户可以从“PPB 地址空间”访问芯片内部的 PPB 存储区，PPB 存储区是一个非易失存储区，用于存储和每个扇区对应的 PPB 信息。当“主存储空间”为“当前地址空间”且芯片在空闲状态时，用户可以写入命令将“PPB 地址空间”切换为“当前地址空间”，具体的命令及编码参见“命令编码表”。

在“PPB 地址空间”用户可以进行以下操作：

使用读操作从扇区的任意地址处读取和扇区对应的 PPB 信息（其中第 0 位表示扇区是否被写保护）

使用简化的字编程命令向 PPB 存储区中写数据

使用简化的 PPB 擦除命令将所有扇区的 PPB 信息擦除

使用退出命令退出“PPB 地址空间”

2.1.8 DYB 地址空间

用户可以从“DYB 地址空间”访问芯片内部的 DYB 存储区，DYB 存储区是一个易失存储区，用于存储和每个扇区对应的 DYB 信息。当“主存储空间”为“当前地址空间”且芯片在

空闲状态时，用户可以写入命令将“DYB 地址空间”切换为“当前地址空间”，具体的命令及编码参见“命令编码表”。在“DYB 地址空间”用户可以进行以下操作：

使用读操作从扇区的任意地址处读取和扇区对应的 DYB 信息（其中第 0 位表示扇区是否被写保护）

使用简化的字编程命令将 DYB 设置为 0 或 1

使用退出命令退出“DYB 地址空间”

2.1.9 PPB Lock 地址空间

用户可以从“PPB Lock 地址空间”访问芯片内部的 PPB Lock 寄存器，PPB Lock 寄存器是一个易失存储器，用于保存 PPB 的写保护信息。当“主存储空间”为“当前地址空间”且芯片在空闲状态时，用户可以写入命令将“PPB Lock 地址空间”切换为“当前地址空间”，具体的命令及编码参见“命令编码表”。在“PPB Lock 地址空间”用户可以进行以下操作：

使用读操作从任意地址处读取 PPB Lock 寄存器的值（其中第 0 位表示 PPB 是否被写保护）

使用简化的字编程命令将 PPB Lock 寄存器设置为 0

使用退出命令退出“PPB Lock 地址空间”

2.2 数据保护

Flash 存储器是非易失存储器，掉电后存储的数据不会丢失，但在实际工作中有许多因素可导致存储的数据被有意或无意修改，为了避免存储器中存储的数据被意外修改，芯片内部实现了若干数据保护技术以保证数据的安全。

2.2.1 上电写保护

芯片上电复位期时，内部控制电路对芯片进行初始化，在这期间 RESET#、CE#、WE#和OE#管脚的信号将被芯片忽略，WE#信号的上升沿不会将命令写入芯片，芯片的输出为高阻，芯片内部的接口控制器和算法控制器被复位，以保证上电期间不会意外改写芯片中存储的数据。

2.2.2 VCC 低压写保护

当芯片的核电压 VCC 小于 VLKO 时，接口控制器拒绝接受任何写入的命令，算法控制器被复位，以保证数据在芯片上电/掉电时不会被意外修改。当核电压 VCC 大于 VLKO 时，用户必须正确控制芯片以防止数据被意外修改。

2.2.3 软件写保护

用户通过写入命令或命令序列来触发芯片内部的算法控制器执行相应的算法，接口控制器会检查每个写周期的地址信号和数据信号以确定其是否是有效的命令或命令序列的组

成部分，当检测到有效 的命令或完整的命令序列时，接口控制器通过内部接口通知算法控制器执行相应的算法。如果用户写入的是无效命令，或者命令序列没有按照规定的顺序写入，那么命令或命令序列将被丢弃，以保护存储的数据不会被意外修改。需要注意的是，写入的无效命令或命令序列有可能使芯片进入到无效状态，此时必须通过软件复位命令或硬件复位操作才能将芯片恢复到正常工作状态。

2.2.4 硬件写保护

芯片提供一个专用的管脚 WP#用于对最高地址扇区进行写保护，如果 WP#信号为低电平，最高 地址扇区将被写保护，如果 WP#信号为高电平，最高地址扇区未被 WP#写保护，WP#管脚内部包含 上拉电阻，如果该管脚悬空，那么WP#信号为高电平。

2.2.5 高级扇区保护

高级扇区保护（Advanced Sector Protection）是一组独立的用于对扇区进行写保护的硬件和软件方法。主存储区中的每一个扇区都有一个易失的 DYB 位和一个非易失的 PPB 位与之对应，如果 DYB 或 PPB 为 0，那么与之对应的扇区被写保护。

DYB 是易失的保护位，掉电后信息将丢失。上电复位或硬件复位后，所有扇区的 DYB 都被复位 为 1，扇区未被 DYB 写保护。如果用户需要对扇区进行写保护，通过写入相应命令将扇区的 DYB 设置为 0 即可。当用户需要对被 DYB 保护的扇区解除写保护时，写入相应命令将对应的 DYB 设置为 1 即可，通过 DYB 在实现写保护的同时兼顾了灵活性。

PPB 是非易失的保护位，掉电后信息不会丢失，如果用户需要对扇区进行写保护，通过编程命 令将扇区的 PPB 编程为 0 即可，如果用户需要对被 PPB 保护的扇区解除写保护，通过擦除命令将扇区 的 PPB 信息擦除为 1 即可。需要注意的是，擦除命令会将所有扇区的 PPB 信息擦除为 1.

芯片也支持对 PPB 的写保护，芯片内部包含一个易失的 PPB Lock 寄存器，如果 PPB Lock 为 0，那么 PPB 被写保护，所有对 PPB 的编程/擦除操作都将被芯片忽略，只有当 PPB Lock 寄存器为 1 时才能 对 PPB 进行编程/擦除操作。芯片向用户提供了两种方法来管理 PPB Lock 寄存器，通过将 Lock Register 编程为不同的值可选择不同的方法来管理 PPB Lock 寄存器。

如果用户将 Lock Register 的第 2 比特编程为 0，芯片将使用默认方法来管理 PPB Lock 寄存器。上电 复位或硬件复位后 PPB Lock 寄存器被复位为 1，PPB 未被写保护。如果需要激活 PPB 写保护，通过命 令将 PPB Lock 寄存器设置为 0 即可。PPB Lock 寄存器被设置为 0 后，没有任何命令可以将其设置为 1，只有通过硬件复位才能将其复位为 1。

如果用户将 Lock Register 的第 3 比特编程为 0，芯片将使用密码来保护 PPB 。上电复位或硬件复位 后 PPB Lock 寄存器被复位为 0，PPB 被写保护。如果需要改写 PPB，用

户必须通过一个解锁命令输入 64 比特的密码，芯片将用户输入的密码和芯片内部存储的密码进行比较，如果用户输入的密码正确，PPB Lock 寄存器被设置为 1，然后用户就可以对 PPB 进行编程/擦除操作，如果输入的密码不正确，PPB Lock 寄存器的值将保持为 0 不变。

2.3 算法控制器

芯片内部的接口控制器负责从用户输入的数据中检测有效命令，检测到有效命令后再交由算法控制器实现命令要求的操作，算法控制器最主要的功能就是实现非易失存储器的编程和擦除控制，除此之外算法控制器还实现了地址空间切换和高级扇区保护功能。

芯片在逻辑上存在着多个地址空间，算法控制器可以根据用户写入的命令实现“默认地址空间”和“其它地址空间”之间的切换。

改变非易失存储器中的内容需要一系列复杂的操作才能实现，这一系列复杂的操作被称为算法，芯片的所有算法都由算法控制器实现，最主要的算法就是编程和擦除算法，用户写入的命令由接口控制器检测到后会触发算法控制器执行相应的算法，算法控制器执行完算法要求的所有步骤后返回空闲状态，在算法执行过程中，算法控制器还负责向用户提供状态信息。

2.3.1 读数据算法

用户按照读操作的要求设置好控制信号 (CE#=0, OE#=0, WE#=1) 和地址信号后芯片内部控制器即开始读取数据，当读访问时序参数 (tACC, tCE 和 tOE) 都满足后数据即从 DQ 管脚输出芯片。

在每次读操作过程中内部控制器都会一次性将整页 32 字节的数据读出，页地址由 Amax-A4 确定，页内字地址由 A3-A0 确定。如果两次读操作之间 CE# 和 OE# 保持低电平且页地址保持不变，仅改变页内字地址，那么仅需要等待 tPACC 即可从 DQ 获得所需数据。需要注意的是，如果在两次读操作之间 CE# 信号或页地址发生了变化，那么需要重新等待 tACC 或 tCE 才能从 DQ 获取数据。

2.3.2 字编程算法

用户可以使用“字编程”命令将一个字的数据 (16Bit) 编程到主存储区。“字编程”命令序列由 4 个写周期组成，首先是 2 个周期的 SDP 解锁周期，然后是 1 个周期的编程命令，最后是 1 个周期的编程地址和数据。写入“字编程”命令后，用户无需提供任何额外的控制信息，芯片内部的算法控制器会自动产生编程操作所需的所有控制信号和时序，编程操作结束后，算法控制器返回到空闲状态。用户可以使用 3 种方法来确定编程操作是否完成：传统的“数据查询”方法、读状态寄存器方法或检测专用管脚 RY/BY# 方法。

编程操作正在进行时，用户写入的除“编程中断”命令之外的任何命令都将被芯片丢

弃。需要注意的是：硬件复位可以立刻终止正在进行的编程操作并使芯片在tRPH后返回读数据状态，为了保持数据完整性，用户应该在硬件复位结束后重新写入之前被终止的编程命令。

2.3.3 行编程算法

用户可以使用“行编程”命令一次性将最多512字节数据写入主存储区或OTP存储区，“行编程”命令包含6-261个写周期，写入行编程命令后，用户无需提供其它的控制信号，芯片内部自动产生编程操作所需的所有控制信号，内部编程操作结束后，算法控制器返回到空闲状态。用户可以使用3种方法来判断编程操作是否完成：传统状态查询方法（Data Polling）、读状态寄存器和监控专用管脚RY/BY#。

编程操作正在进行时，用户写入的除“编程挂起”命令之外的任何命令都将被芯片丢弃。需要注意的是：硬件复位可以立刻终止正在进行的编程操作并使芯片在tRPH后返回读数据状态。为了保持数据完整性，用户应该在硬件复位结束后重新写入之前被终止的编程命令。

2.3.4 编程中断与恢复

用户可以使用“编程中断”命令来暂停正在进行的编程操作，芯片在用户写入“编程中断”命令后最多不超过tPSL即可暂停编程操作，编程操作被暂停后，用户可以从被编程中断的行之外的任意地址读取数据。

芯片支持两种不同的“编程中断”命令，为了和之前的产品兼容，芯片支持传统的“编程/擦除中断”命令（命令码为B0h），芯片也支持专用的“编程中断”命令（命令码为51h）。同理芯片也支持两种“编程恢复”命令：传统的“编程/擦除恢复”命令（命令码为30h）和专用的“编程恢复”命令（命令码为50h）。建议使用专用的中断和恢复命令来暂停编程操作，使用传统的中断和恢复命令来暂停擦除操作。

用户可以使用“编程恢复”命令来恢复被中断的编程操作，编程操作恢复后可根据需要再次被中断，但需要注意的是：“编程恢复”和下一次“编程中断”命令之间的间隔时间应该大于或等于tPRS。

2.3.5 扇区擦除算法

用户可以使用“扇区擦除”命令来擦除1个扇区，在写“扇区擦除”命令之前用户无需进行预编程，内部算法控制器会自动将所有需要擦除的地址预编程为0x0000状态并验证。用户在写“扇区擦除”命令之后无需提供任何额外的控制信息，内部算法控制器会自动产生擦除操作所需的所有信号和时序。“扇区擦除”命令序列由6个写周期组成，首先是2个周期的SDP解锁周期，然后是1周期的“擦除准备”命令，然后又是2个周期的SDP解锁周期，最后是1个周期的扇区擦除命令。芯片接受“扇区擦除”命令后，用户可以使用3种方法来判断擦除操作是否完成：传统的“数据查询”方法、读状态寄存器方法或检测专用管脚

RY/BY#方法。

在擦除操作过程中，用户写入的除“读状态寄存器”和“擦除挂起”命令之外的任何命令都将被芯片丢弃，需要注意的是：硬件复位可以立刻终止正在进行的擦除操作并使芯片在tRPH后返回读数据状态。为了保持数据完整性，用户应该在硬件复位结束后重新写入之前被终止的扇区擦除命令。擦除操作正确完成后，如果扇区未被写保护，扇区所有地址处的数据变为0xFFFF，否则扇区中的数据保持不变。

2.3.6 芯片擦除算法

用户可以使用“芯片擦除”命令将整个主存储区擦除，在写“芯片擦除”命令之前用户无需进行预编程，内部算法会自动将所有需要擦除的地址预编程为0x0000状态并验证。用户在写“芯片擦除”命令之后无需提供任何额外的控制，内部算法会自动产生擦除操作所需的所有信号。“芯片擦除”命令包含6个写周期，首先是2个周期的解锁周期，然后是1周期的擦除准备命令，然后又是2个周期的解锁周期，最后是1个周期的芯片擦除命令。在最后一个周期WE#上升沿后，RY/BY#信号变为低电平。

在擦除操作过程中，用户写入的除“读状态寄存器”命令之外的任何命令都将被芯片丢弃，需要注意的是：硬件复位可以立刻终止正在进行的擦除操作并使芯片在tRPH后返回读数据状态。为了保持数据完整性，用户应该在硬件复位结束后重新写入之前被终止的芯片擦除命令。用户可以使用3种方法来判断擦除操作是否完成：传统状态查询方法（Data Polling）、读状态寄存器和监控专用管脚RY/BY#。

擦除操作正确完成后，未写保护扇区的所有地址处的数据变为0xFFFF，被写保护的扇区中的数据保持不变。由于扇区写保护而导致扇区没有被擦除不会导致状态寄存器中的“擦除状态位”和“扇区锁定位”被置1。

2.3.7 擦除中断与恢复

用户可以使用“擦除中断”命令来暂停正在进行的扇区擦除操作，用户写入“擦除中断”命令后，芯片需要最多tESL来暂停扇区擦除操作，用户可以通过“数据查询”或者“读状态寄存器”来确定命令是否执行完毕。在执行芯片擦除操作时写入的“擦除中断”命令将被芯片丢弃。

擦除操作被暂停后，用户可以从被擦除中断扇区之外的扇区读数据，从被擦除中断的扇区中读数据将读出无效数据。用户也可以向被擦除中断扇区之外的扇区中写数据，在写数据的过程中，用户可以像标准编程操作一样，通过“数据查询”或“读状态寄存器”来确定编程操作是否完成，如果编程操作正确完成，那么芯片返回“擦除中断状态”，如果编程出现错误，那么必须使用复位命令将芯片恢复至“擦除中断状态”。

当完成所需的操作后，用户可以使用“擦除恢复”命令来恢复被中断的扇区擦除操作，擦除操作恢复后可根据需要再次被中断，但需要注意的是：“擦除恢复”和下一次“擦除中

“断”命令之间的间隔时间应该大于或等于 tERS。

2.3.8 空检查算法

用户可以使用“空检查”命令来确定主存储区的某一个扇区是否被完全擦除，当算法控制器正在进行编程/擦除操作或者编程/擦除操作中断时不能够进行空检查，只有当算法控制器处于空闲状态时才可以写入“空检查”命令。

当“空检查”命令正在执行时，用户通过读操作读出的数据为无效数据。用户可以通过读状态寄存器来确定空检查是否完成，如果状态寄存器的第 7 位为 0，表示空检查命令正在执行，否则表示空检查已经完成。状态寄存器的第 5 位表示空检查的结果，如果为 0 表示扇区被完全擦除，如果为 1 表示扇区未被完全擦除。算法控制器在检查过程中如果发现被检查扇区中任意比特为 0 就会停止检查并更新状态寄存器，空检查操作结束后，算法控制器返回空闲状态。

2.3.9 状态寄存器

用户可以通过读取内部状态寄存器的值来确定编程/擦除操作是否完成以及编程/擦除操作是否成功。首先用户写入一周期的“读状态寄存器”命令，在命令周期 WE#上升沿后，内部状态被捕获进状态寄存器，同时“状态寄存器地址空间”被切换为当前地址空间。然后用户通过读操作即可读出状态寄存器的值，读操作从“状态寄存器地址空间”的所有地址处都可以读出状态寄存器的值，读操作完成后（CE#或 OE#上升沿），芯片自动退出“状态寄存器地址空间”。

状态寄存器中包含两类信息，一类表示最近执行的编程/擦除操作是否成功（第 1、3、4、5 位），另一类表示算法控制器的当前状态（第 2、6、7 位）。状态寄存器的第 0 位和高 8 位为无效数据，虽然在不同的读操作中读出的数据可能会发生变化，但用户不能将其作为有意义的值进行使用。当第 3 位的值为 0 时，使用软件复位命令可将第 5、4 和 1 位复位为 0，但是不会影响第 2、6 和 7 位。使用“清除状态寄存器”命令可将第 5、4、3 和 1 位复位为 0，但不会影响第 2、6 和 7 位。

比特位	复位值	忙状态	闲状态
15-8	X	无效	X
7	1	0	1
6	0	无效	0：擦除未中断；1：擦除中断
5	0	无效	0：擦除成功；1：擦除失败
4	0	无效	0：编程成功；1：编程失败
3	0	无效	0：行编程未终止；1：行编程终止
2	0	无效	0：编程未中断；1：编程中断
1	0	无效	0：无写锁定扇区错误；1：写锁定扇区错误
0	0	无效	X

2.3.10 数据查询

如果用户写入的命令会触发内部算法控制器执行相应算法，那么在命令最后一个周期 WE#上升沿后芯片被自动切换到“数据查询”地址空间，此时从任何地址读数据都将读出 16 比特的状态信息，用户可以根据读出的状态信息确定命令是否执行完毕。芯片当前使用第 1、2、3、5、6、7 位表示状态，其它位留作未来使用，用户应将其忽略。

DQ7: 用户可以通过查询 DQ7 来确定编程/擦除操作是否完，成当用户写入编程命令后，芯片内部算法控制器开始执行相应算法，算法执行期间读出的 DQ7 是编程数据 DQ7 的非，算法执行完毕后读出的 DQ7 是编程数据 DQ7，因此用户可以根据 DQ7 的状态判断编程算法是否完成。需要注意的是，为了保证上述判断规则有效，当进行字编程时，用户应该从编程目标地址处读取 DQ7，当进行行编程时，用户应该从最后一个写入的数据地址处读 DQ7。

当用户写入擦除/空检查命令后，芯片内部算法控制器开始执行相应算法，在算法执行期间读出的 DQ7 为 0，算法执行完毕或擦除算法被中断后读出的 DQ7 为 1，因此用户可以根据 DQ7 的状态判断擦除/空检查算法是否完成。需要注意的是，为了保证上述判断规则有效，用户应该从正在进行擦除/空检查的扇区读 DQ7。

需要注意的是，当用户从某个读周期读出的 DQ7 表明内部算法已经完成时，同一周期读出的 DQ6- DQ0 并不一定是有效数据，因为 DQ7 的变化相对于其它位是异步的，用户只有在下一个读周期才能读出有效数据。

DQ6: 用户可以通过查询 DQ6 来确定编程/擦除操作是否完成，和 DQ7 使用高低电平表示状态不同，DQ6 使用状态是否翻转来表示操作是否完成。用户从任意地址进行两次连续的读操作，如果两次读操作读出的 DQ6 状态不同，表示内部算法控制器正在执行相应算法，反之如果两次读操作读出的 DQ6 状态相同，表示内部算法已经完成。

DQ5: DQ5 表示芯片内部编程/擦除算法是否超时，编程/擦除算法正常进行时 DQ5 为 0，算法超时后 DQ5 变为 1，这是一个错误状态，表示算法控制器未能在允许的最大时间内将非易失存储器的存储单元改变为期望的状态。当出现算法超时后，用户可以使用“软件复位”命令将算法控制器从错误状态恢复至空闲状态，需要注意的是，写完“软件复位”命令后，算法控制器大约需要 2us 才能恢复到空闲状态。

DQ3: 当写入“扇区擦除”命令后，用户可以通过查询 DQ3 的状态来确定扇区擦除算法是否开始执行。写完“扇区擦除”命令后，用户应该首先通过查询 DQ7 或 DQ6 来确定芯片已经正确接受命令，然后再查询 DQ3 来确定算法是否开始执行，如果读出的 DQ3 为 0，表示扇区擦除算法还未开始执行，否则表示算法已经开始执行。扇区擦除算法开始执行后，用户写入的除“擦除中断”和“读状态寄存器”命令之外的任何命令都将被芯片丢弃。

DQ2: 当芯片正在执行擦除命令或者擦除命令被中断后，用户可以通过查询 DQ2 来确定某个扇区是否是擦除命令的目标扇区，和 DQ6 一样，DQ2 也是通过状态的翻转与否来表

示信息，向某个地址进行两次连续的读操作，如果DQ2的状态发生翻转则表示读操作地址所在的扇区是擦除操作的目标扇区，否则表示读操作地址所在的扇区不是擦除操作的目标扇区。

DQ1: 用户写“行编程”命令时必须遵守一系列规则，如果违反任何一条规则都将导致芯片进入错误状态，芯片通过将DQ1设置为1来将错误信息报告给用户，此时用户必须写入“行编程错误复位”命令将算法控制器从错误状态恢复至空闲状态，同时也将状态寄存器中相应的错误信息清除。

操作	DQ7	DQ6	DQ5	DQ3	DQ2	DQ1	RY/BY#
编程算法执行时读数据	DQ7#	翻转	0	无效	不变	0	0
编程算法超时时读数据	DQ7#	翻转	1	无效	不变	0	0
编程中断时读被中断的行	无效	无效	无效	无效	无效	无效	1
编程中断时读其它行	数据	数据	数据	数据	数据	数据	1
擦除算法执行时读擦除目标扇区	0	翻转	0	1	翻转	无效	0
擦除算法执行时读其它扇区	0	翻转	0	1	不变	无效	0
擦除算法中断时读擦除目标扇区	1	不变	0	无效	翻转	无效	1
擦除算法中断时读其它扇区	数据	数据	数据	数据	数据	数据	1
擦除中断状态下编程时读数据	DQ7#	翻转	0	无效	无效	无效	0
行编程错误时读数据	DQ7#	翻转	0	无效	无效	1	0

2.3.11 算法性能

参数	最小值	典型值	最大值	单位
扇区擦除时间	--	275	1100	ms
字编程时间	--	125	400	us
512字节页编程时间	--	340	750	us
512字节页编程有效字节编程时间	--	1.33	--	us
扇区编程时间	--	108	192	ms
擦除中断/恢复时间	--	--	40	us
编程中断/恢复时间	--	--	40	us
擦除恢复/中断间隔时间	--	100	--	us
编程恢复/中断间隔时间	--	100	--	us
空检查时间	--	6.2	8.5	ms

2.4 命令编码表

2.4.1 主地址空间访问命令

命令功能	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期
读数据	RA/RD	--	--	--	--	--
软件复位	XXX/F0	--	--	--	--	--
读状态寄存器	555/70	XXX/RD	--	--	--	--
清除状态寄存器	555/71	--	--	--	--	--
字编程	555/AA	2AA/55	555/A0	PA/PD	--	--
行编程	555/AA	2AA/55	SA/25	SA/WC	WBL/PD	SA/29
行编程错误复位	555/AA	2AA/55	555/F0	--	--	--
芯片擦除	555/AA	2AA/55	555/80	555/AA	2AA/55	555/10
扇区擦除	555/AA	2AA/55	555/80	555/AA	2AA/55	SA/30
编程/擦除中断	XXX/B0	--	--	--	--	--
编程/擦除恢复	XXX/30	--	--	--	--	--
编程中断	XXX/51	--	--	--	--	--
编程恢复	XXX/50	--	--	--	--	--
空检查	(SA)555/33	--	--	--	--	--

1. AAA/DD: AAA 表示地址的低 11 位, DD 表示数据的低 8 位
2. RA = 读操作地址输入, RD = 读操作数据输出, PA = 编程地址输入, PD = 编程数据输入
3. XXX = 任意值, SA = 扇区地址, WBL = 行编程字地址, WC = 行编程字总数-1

2.4.2 ID/CFI 地址空间访问命令

命令功能	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期
进入地址空间	555/AA	2AA/55	(SA)555/90	--	--	--
进入地址空间	(SA)55/88	--	--	--	--	--
读数据	RA/RD	--	--	--	--	--
退出地址空间	XXX/F0	--	--	--	--	--

2.4.3 OTP 地址空间访问命令

命令	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期
进入地址空间	555/AA	2AA/55	(SA)555/88	--	--	--
读数据	RA/RD	--	--	--	--	--
字编程	555/AA	2AA/55	555/A0	PA/PD	--	--
行编程	555/AA	2AA/55	SA/25	SA/WC	WBL/PD	SA/29
行编程错误恢复	555/AA	2AA/55	555/F0	--	--	--
退出地址空间	555/AA	2AA/55	555/90	XXX/00	--	--
退出地址空间	XXX/F0	--	--	--	--	--

2.4.4 Lock Register 地址空间访问命令

命令	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期
进入地址空间	555/AA	2AA/55	555/40	--	--	--
读数据	0/RD	--	--	--	--	--
字编程	XXX/A 0	XXX/PD	--	--	--	--
退出地址空间	XXX/90	XXX/00	--	--	--	--
退出地址空间	XXX/F0	--	--	--	--	--

2.4.5 Password 地址空间访问命令

命令	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期	7 周期
进入地址空间	555/AA	2AA/55	555/60	--	--	--	--
字编程	XXX/A 0	PA/PD	--	--	--	--	--
读数据	0/PWD0	1/PWD1	2/PWD2	3/PWD3	--	--	--
解锁	0/25	0/3	0/PWD0	1/PWD1	2/PWD2	3/PWD3	0/29
退出地址空间	XXX/90	XXX/00	--	--	--	--	--
退出地址空间	XXX/F0	--	--	--	--	--	--

1. PWD = 密码

2.4.6 PPB 地址空间访问命令

命令	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期
进入地址空间	555/AA	2AA/55	555/C0	--	--	--
读数据	SA/RD	--	--	--	--	--
PPB 编程	XXX/A 0	XXX/0	--	--	--	--
PPB 擦除	XXX/80	0/30	--	--	--	--
退出地址空间	XXX/90	XXX/00	--	--	--	--
退出地址空间	XXX/F0	--	--	--	--	--

2.4.7 PPB Lock 地址空间访问命令

命令	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期
进入地址空间	555/AA	2AA/55	555/50			
读数据	XXX/RD					
PPB Lock 使能	XXX/A 0	XXX/0				
退出地址空间	XXX/90	XXX/00				
退出地址空间	XXX/F0					

2.4.8 DYB 地址空间访问命令

命令	1 周期	2 周期	3 周期	4 周期	5 周期	6 周期
进入地址空间	555/AA	2AA/55	555/E0	--	--	--
读数据	SA/RD	--	--	--	--	--
DYB 置 0	XXX/A0	SA/0	--	--	--	--
DYB 置 1	XXX/A0	SA/1	--	--	--	--
退出地址空间	XXX/90	XXX/00	--	--	--	--
退出地址空间	XXX/F0	--	--	--	--	--

3 电特性

3.1 绝对最大额定值

绝对最大额定值如下：

RESET#管脚电压	-0.5V~ V_{CC} +0.5V
其它管脚电压	-0.5V ~ V_{IO} + 0.5V
输出短路电流	10 0 mA
V_{CC} 电压	-0.5~4.0 V
V_{IO} 电压	-0.5 ~ V_{CC} +0.5V
存储温度范围(T_{stg})	-65°C ~ 150°C
结温(T_J)	150 °C

3.2 推荐工作条件

推荐工作条件如下：

环境温度 (T_A)	-55~125 °C
V_{CC} 电压	2.7~3.6 V
V_{IO} 电压	1.65 ~ V_{CC} +0.2V

3.3 DC 参数

符号	描述	测试条件	最小值	典型值	最大值	单位
I_{L1}	输入负载电流	$V_{IN} = V_{SS} \sim V_{CC}$	--	0.02	1.0	uA
I_{L0}	输出漏电流	$V_{OUT} = V_{SS} \sim V_{CC}$	--	0.02	1.0	uA
I_{CC1}	随机读电流	$CE\#=L, OE\#=H, f=5\text{MHz}$	--	55	60	mA
I_{CC2}	页内读电流	$CE\#=L, OE\#=H, f=33\text{MHz}$	--	9	25	mA
I_{CC3}	编程擦除电流	$CE\#=L, OE\#=H$	--	45	100	mA
I_{CC4}	待机电流	$CE\#, OE\#, RESET\#=H$	--	70	100	uA
I_{CC5}	硬件复位电流	$CE\#=H, RESET\#=L$	--	10	20	mA
I_{CC6}	休眠电流	$CE\#=L, 地址稳定 8\mu\text{s} 后$	--	100	150	uA
I_{CC7}	上电复位电流	$RESET\#, CE\#, OE\#=V_{IO}$	--	53	80	mA
V_{IL}	输入低电平	$V_{CC} = \text{Min}$	-0.5	--	$0.3*V_{IO}$	V
V_{IH}	输入高电平	$V_{CC} = \text{Min}$	$0.7*V_{IO}$	--	$V_{IO}+0.4$	V
V_{OL}	输出低电平	$I_{OL} = 100\mu\text{A}, V_{CC} = \text{Min}$	--	--	$0.15*V_{IO}$	V
V_{OH}	输出高电平	$I_{OH} = 100\mu\text{A}, V_{CC} = \text{Min}$	$0.85*V_{IO}$	--	--	V
V_{LKO}	写保护电压	--	2.25	--	2.5	V
V_{RST}	复位电压	--	--	1.0	--	V

1. 所有电流参数测试时, $V_{CC} = V_{CC\ MAX}$

3.4 时序参数

3.4.1 测试条件

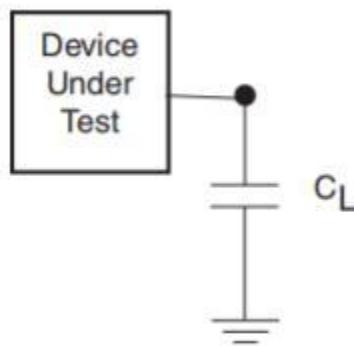


图 3-1 测试设置

参数名	参数值	单位
输出负载电容	30	pF
输入上升/下降时间	1.5	ns
输入电平范围	0-V _{IO}	V
输入信号测试参考点	$V_{IO}/2$	V
输出信号测试参考点	$V_{IO}/2$	V

3.4.2 上电复位时序参数

符号	描述	最小值	典型值	最大值	单位
tvcs	VCC 建立时间	300	--	--	us
tvios	VIO 建立时间	300	--	--	us

1. 从 VCC 和 VIO 达到最低工作电压开始测量

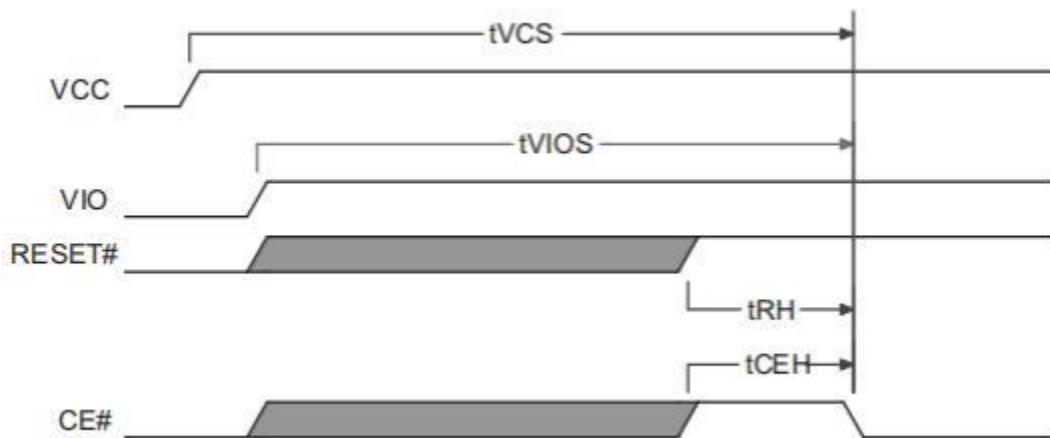


图 3-2 上电复位时序图

3.4.3 硬件复位时序参数

符号	描述	最小值	典型值	最大值	单位
tRP	复位脉冲宽度	200	--	--	ns
tRH	RESET# 上升沿到 CE# 下降沿等待时间	50	--	--	ns
tRPH	RESET# 下降沿到 CE# 下降沿等待时间	35	--	--	us

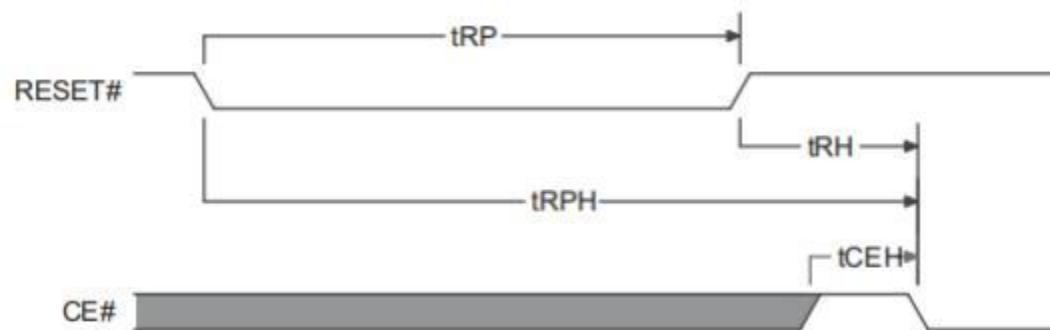


图 3-3 硬件复位时序图

3.4.4 读操作时序参数

符号	描述	最小值	典型值	最大值	单位
tRC	读周期时间	120	--	--	ns
tACC	地址访问时间	--	--	120	ns
tCE	CE# 访问时间	--	--	120	ns
toE	OE# 访问时间	--	--	25	ns

符号	描述	最小值	典型值	最大值	单位
t_{PACC}	页访问时间	--	--	20	ns
t_{OH}	CE#、OE#或地址变化后数据保持时间	0	--	--	ns
t_{DF}	CE#或 OE#无效到输出高阻时间	--	--	15	ns

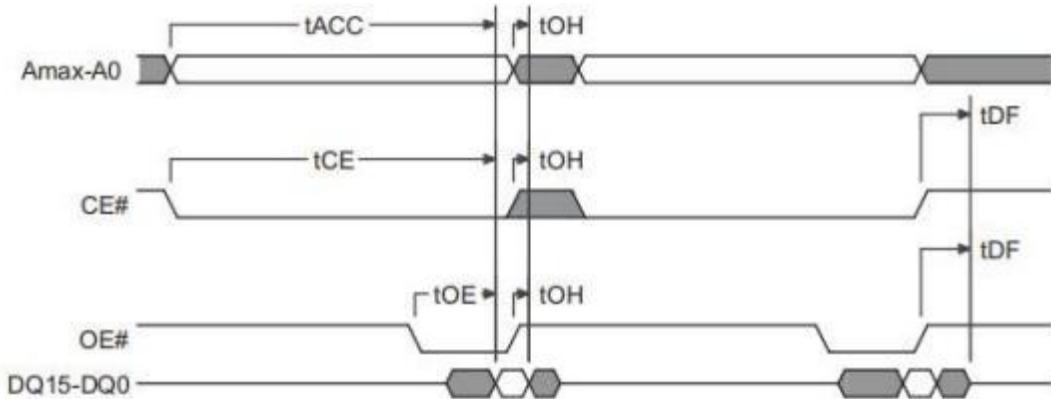


图 3-4 CE#控制读时序图

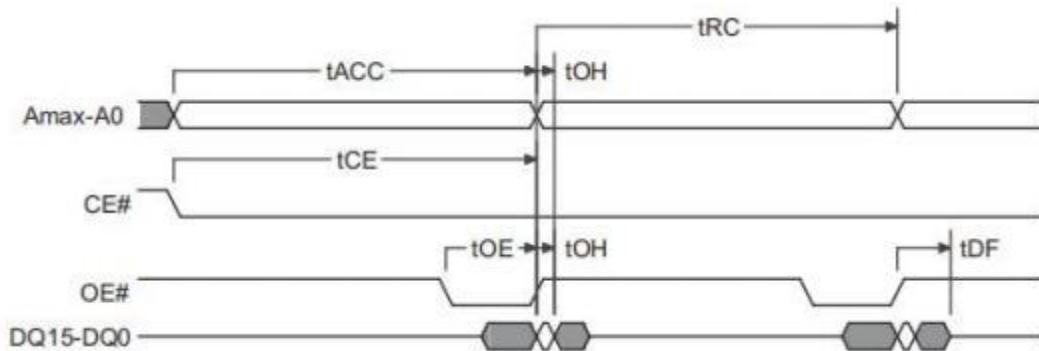


图 3-5 地址控制读时序图

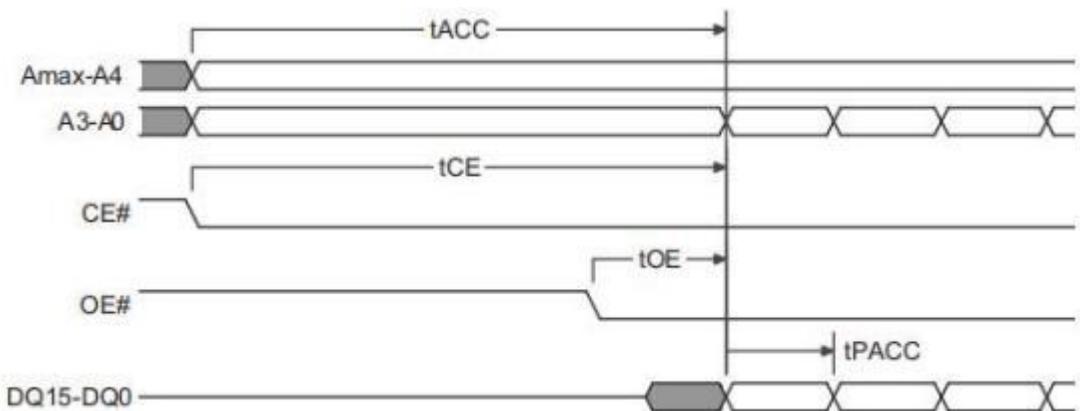


图 3-6 页模式读时序图

3.4.5 WE#控制写操作时序参数

符号	描述	最小值	典型值	最大值	单位
t_{WC}	写周期时间	60	--	--	ns
t_{CS}	CE#对 WE#建立时间	0	--	--	ns

符号	描述	最小值	典型值	最大值	单位
t _{CH}	CE#对 WE#保持时间	0	--	--	ns
t _{GHWL}	OE#对 WE#建立时间	0	--	--	ns
t _{OEH}	OE#对 WE#保持时间	10	--	--	ns
t _{AS}	地址对 WE#建立时间	0	--	--	ns
t _{AH}	地址对 WE#保持时间	45	--	--	ns
t _{DS}	数据对 WE#建立时间	30	--	--	ns
t _{DH}	数据对 WE#保持时间	0	--	--	ns
t _{ASO}	地址对 CE#/OE#建立时间 (数据查询)	15	--	--	ns
t _{AHT}	地址对 CE#/OE#保持时间 (数据查询)	0	--	--	ns
t _{OEPh}	OE#高电平时间 (数据查询)	20	--	--	ns
t _{WP}	WE#低电平脉宽	25	--	--	ns
t _{WPH}	WE#高电平脉宽	20	--	--	ns

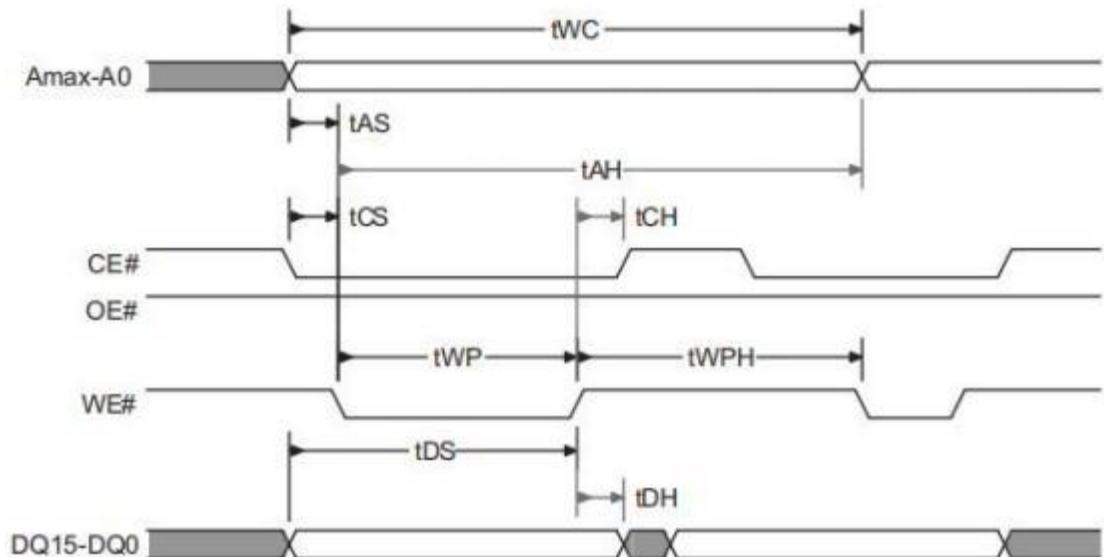


图 3-7 WE#控制的写-写时序图

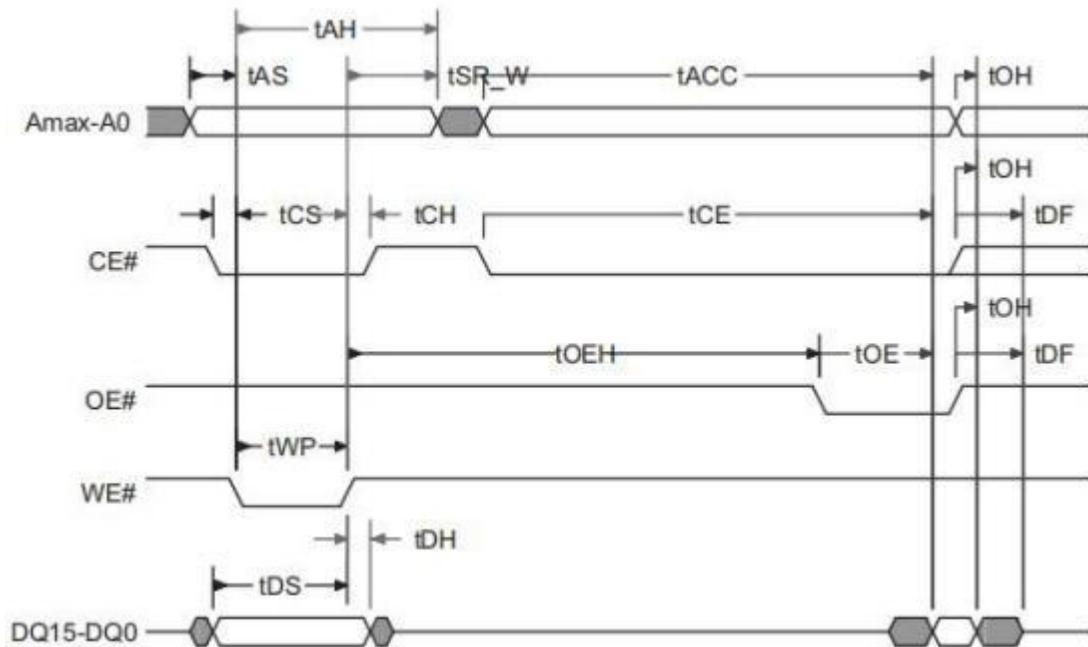


图 3-8 WE#控制的写-读时序图

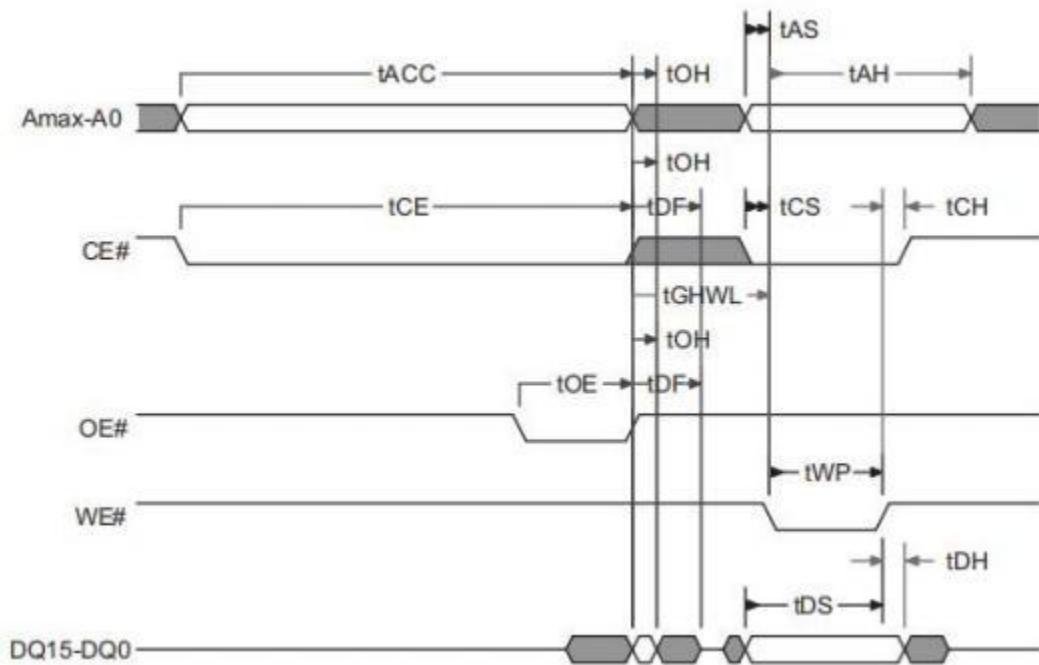


图 3-9 WE#控制的读-写时序图

3.5 上电和掉电

在芯片上电或掉电期间，用户必须保证 VCC 大于或等于 VIO。上电时，只有当 VCC 和 VIO 都超过其最低工作电压 tVCS 时间后用户才可以对芯片进行读写操作，在这之前芯片内部逻辑执行上电复位操作，所有的输入信号都被芯片忽略。

如果 VCC 电压下降到低于 VLKO 最大值，用户必须将 VCC 和 VIO 都降低至低于

VRST 最小值并保持至少 t_{PD} 时间, 然后再次将 VCC 和 VIO 升高至工作电压, 这样才能触发内部上电复位逻辑重新对芯片进行初始化。如果 VCC 下降但没有超过 VLKO 最大值, 那么芯片内部的配置信息不会丢失, 等 VCC 恢复至正常工作电压后芯片即可正常工作。如果芯片上电初始化失败, 用户可以使用硬件复位来对芯片进行重新初始化。

符号	描述	最小值	最大值	单位
V _{CC}	内核电源电压	2.7	3.6	V
V _{LKO}	写保护电压	2.25	2.5	V
V _{RST}	复位电压	1.0	--	V
t _{VCS}	VCC 建立时间	300	--	us
t _{PD}	复位低压持续时间	15	--	us

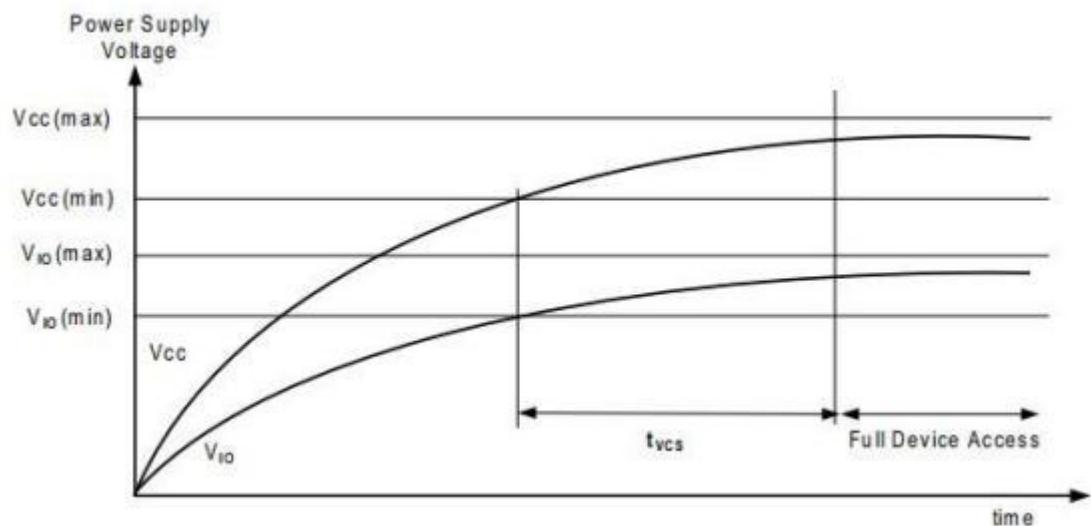


图 3-10 上电时序波形

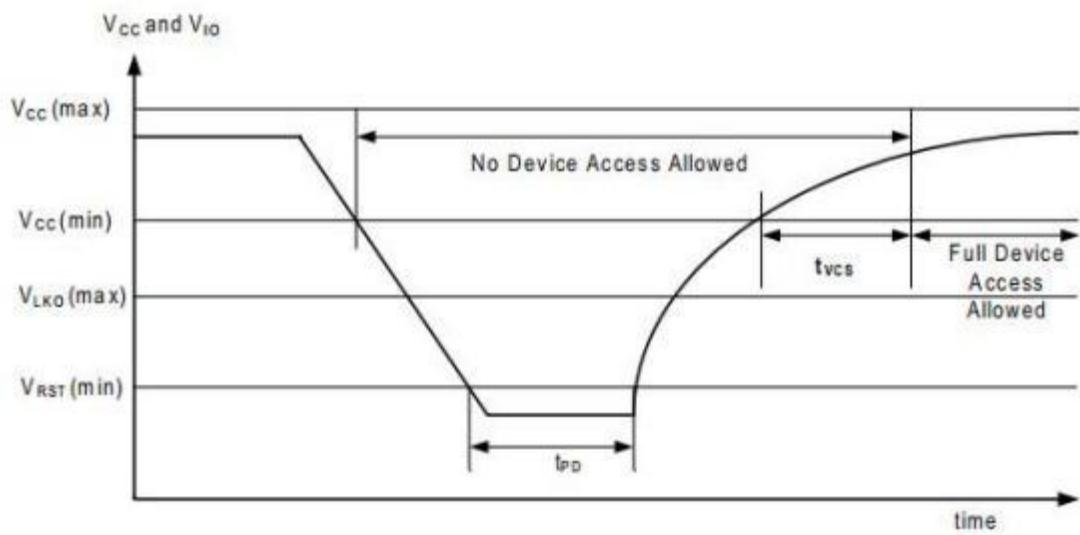


图 3-11 掉电时序波形

4 硬件接口

4.1 接口定义

符号	类型	描述
RESET#	I	硬件复位, 输入低电平可将芯片复位到读数据状态
CE#	I	片选信号
OE#	I	输出使能信号
WE#	I	写使能信号
A	I	地址信号, 512Mb: A[24:0]; 1Gb: A[25:0]
DQ	IO	16 位数据输入输出信号
WP #	I	内部带上拉的写保护输入, 输入低电平时最高地址扇区被写保护
RY/BY#	OD	开漏输出, 芯片忙时输出 0, 芯片闲时输出高阻, 需要外部上拉
VCC	P	核电压
VIO	P	IO 电压
VSS	G	地
NC	-	管脚和内部电路无连接
RFU	-	保留, 管脚和内部电路无连接
DNU	-	测试管脚, 用户禁止使用, 可将其悬空或连接到 VSS

4.2 接口状态

接口状态	VCC	VIO	RESET#	CE#	OE#	WE#	A	DQ
接口掉电	< VLKO	X	X	X	X	X	X	HihgZ
上电复位	$\geq VCCMIN$	$\geq VIO MIN$	X	X	X	X	X	HihgZ
硬件复位	$\geq VCCMIN$	$\geq VIO MIN$	L	X	X	X	X	HihgZ
接口待机	$\geq VCCMIN$	$\geq VIO MIN$	H	H	X	X	X	HihgZ
接口休眠	$\geq VCCMIN$	$\geq VIO MIN$	H	L	X	X	有效	输出
输出关闭	$\geq VCCMIN$	$\geq VIO MIN$	H	L	H	H	有效	HihgZ
读数据	$\geq VCCMIN$	$\geq VIO MIN$	H	L	L	H	有效	输出
写数据	$\geq VCCMIN$	$\geq VIO MIN$	H	L	H	L	有效	输入

5 说明事项

5.1 运输与储存

芯片在适宜环境下储运。

使用指定的防静电包装盒进行产品的包装和运输。在运输过程中，确保芯片不要与外物发生碰撞。

5.2 开箱与检查

开箱使用芯片时，请注意观察产品标识。确定产品标识清晰，无污迹，无擦痕。同时，注意检查无损坏，无伤痕，管脚整齐，无缺失，无变形。

5.3 使用操作规程及注意事项

器件必须采取防静电措施进行操作。取用芯片时应佩戴防静电手套，防止人体电荷对芯片的静电冲击，损坏芯片。将芯片插入电路板上的底座时以及将芯片从电路板上的底座取出时，应注意施力方向以确保芯片管脚均匀受力。不要因为用力过猛，损坏芯片管脚，导致无法使用。

推荐下列操作措施：

- a) 器件应在防静电的工作台上操作，或带指套操作；
- 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（如：集成电路专用盒）；
- e) 生产、测试、使用以及转运过程中应避免使用引起静电的塑料、橡胶或丝织物；
- f) 相对湿度尽可能保持在50%±30%。

5.4 质量保证

公司质量管理体系根据国军标 GJB9001 要求制定了完善的质量管理工作流程，对产品的设计、生产和销售进行日常质量管理。产品制定依据 GJB7400 《合格制造厂认证用半导体集成电路通用规范》裁剪后的标准进行设计和生产，并按照 GJB548 《微电子器件试验方法和程序》的要求进行试验和检验。产品兼容性好、可靠性高。

6 封装尺寸

6.1 外形尺寸图

TLX29LV512M 采用 BGA64 封装，焊球材料为 SAC305，焊球直径为 0.60mm。

具体封装尺寸如图 6-1。

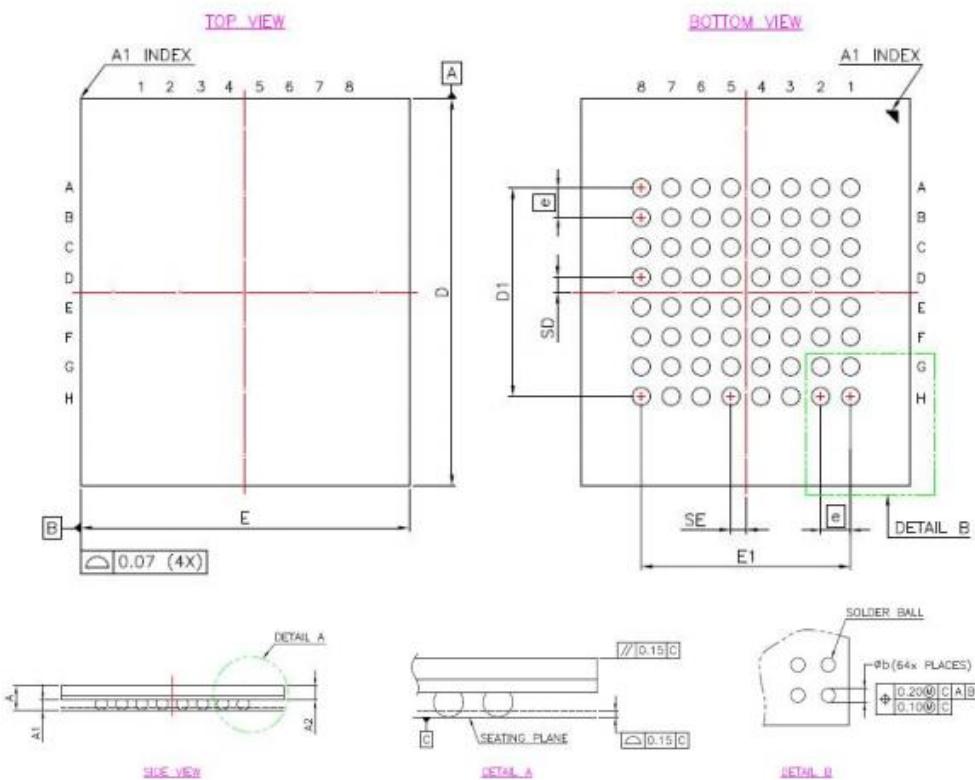


图 6-1 封装外形尺寸图

尺寸符号	最小值	公称值	最大值
A	--	--	1.40
A1	0.40	0.50	0.60
A2	0.65	--	--
øb	0.50	0.60	0.70
D	12.9	13.00	13.10
D1	--	7.00	--
E	10.90	11.00	11.10
E1	--	7.00	--
SE	--	0.50	--
SD	--	0.50	--
e	--	1.00	--

6.2 订购信息

订购型号	温度等级	封装类型	MSL	质量等级
JTLX29LV512M	-55 °C ~ +125 °C	BGA64	MSL1/3	N1/军温级
TLX29LV512M	-40 °C ~ +125 °C	BGA64	MSL1/3	工业级

6.3 推荐焊装工艺

TLX29LV512M 采用无铅回流焊温度，如图 6-2 所示。

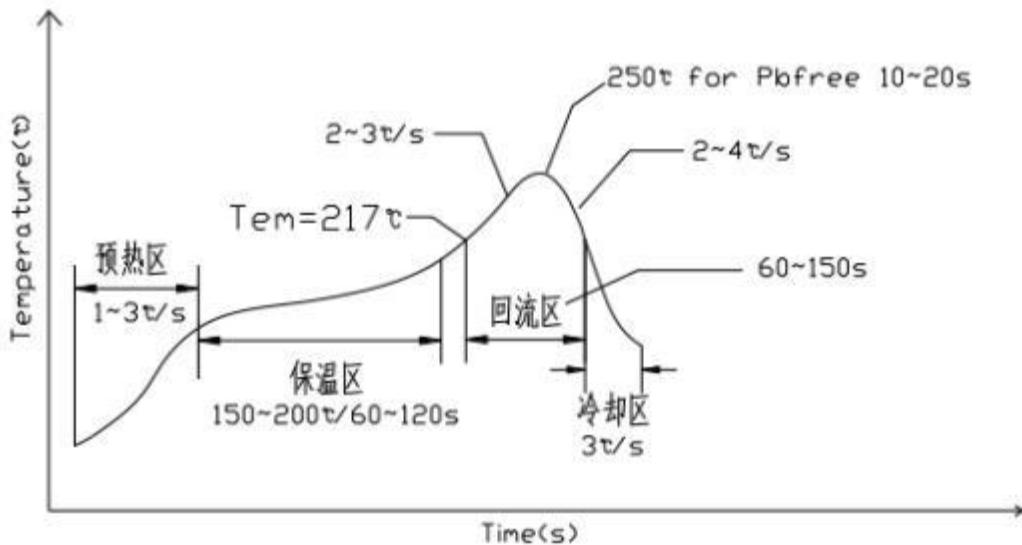


图 6-2 无铅回流焊曲线图