Open NAND Flash Interface Specification

Reversion 4.0
2014.04.02

注：本文是“Open NAND Flash Interface Specification Reversion4.0”的中文完整翻译。因为本人是做验证工作，只关心基本功能，因此文中所有涉及到电气部分的描述都忽略了。

本人也是第一次接触 ONFI 规范，因此翻译过程中有些术语或者内容可能理解不正确。有发现错误的朋友欢迎指出，共同学习。(本文档翻译过程比较耗时，基本每天最多只能翻译 20 页左右，另外限于水平有限，可能翻译质量不是很高，还请多多谅解。如果有想补充修改的朋友，请发送邮件给我来索取原 WORD 文档：li.wsh@163.com)

—风语者
2016 西安

1. 介绍

1.1 目的

该规范定义了标准 NAND 闪存接口，对于要设计的系统提供了一种方法，以支持广泛的 NAND 闪存设备。该规范也为兼容在系统设计时尚不存在的新的 NAND 设备提供了一种解决方案。

规范的目标和要求包括：

- 支持多种容量器件及未来新的发展
- 兼容现有 NAND 闪存设计，提供向 ONFI 的有序过渡
- 容量和特性在参数 page 中被自我描述，因此不需要在 Host 中有硬编码(hard-coded)的 chip ID 表。
- 闪存器件的互操作性，不需要 Host 更改，以支持新的 Flash 设备
- 定义了一个兼容现有 NAND 闪存接口的更高速的 NAND 接口
- 允许独立的核心（VCC）和 I / O（VCCQ）电源轨
- 支持将 NAND lithography 确定的功能放在控制器中，并将控制器放在 NAND 封装中(EZ NAND)。

1.2 EZ NAND 概览

EZ NAND 包含和 NAND 封装在一起用于执行 lithography 确定(如 ECC)的 NAND 管理功能的控制逻辑，同时保留 NAND 协议结构。EZ NAND 提供了最小的命令和/或协议变化的卸载(offload)解决方案。是否支持 EZ NAND 由器件参数 page 决定。

1.3 参考

本规范参考以下规范和标准：

- JEDEC SL_18 标准。标准可在 http://www.jedec.org 获得。

1.4 定义，缩写和约定

1.4.1 定义和缩写

在本规范中使用的术语旨在满足本规范的说明，不依赖于在其他规范中的定义。

1.4.1.1 地址(address)

地址包括行地址和列地址。行地址用来识别和访问 page，块(block)以及 LUN。列地址用来识别和访问 page 中的字节(byte)或字(word)。在 NV-DDR,NV-DDR2 和 NV-DDR3 数据接口中，列地址的最低位应始终为 0。

1.4.1.2 异步

异步的意思是，在写操作中数据被 WE_n 信号锁存，在读操作中数据被 RE_n 信号锁存。

1.4.1.3 块(block)

由多个页面组成，并且是擦除操作的最小可寻址单元。

1.4.1.4 列

在 page 寄存器中字节(x8)或字(x16)的位置。

1.4.1.5 数据突发(data burst)

数据突发是一组连续的没有停顿的数据输入或数据输出。一般来说，数据序列中的停顿时间不大于一个数据周期的时间。

1.4.1.5.1 数据突发结束

Host 在退出突发传输后发出一个新的命令。这将退出 NAND 读模式和结束数据突发。

1.4.1.5.2 数据突发退出

Host 在数据突发期间会将 CE_n，ALE 或 CLE 拉高。当在退出状态(exit state)时 ODT 被关闭(如果使能)，如果在退出后数据突发继续进行，则重新发送 warmup 周期(如果使能)。

1.4.1.5.3 数据突发暂停

Host 在数据突发期间停止 DQS(输入脉冲)或 RE(输出脉冲)。ODT(如果使能)在整个暂停期间保持使能，当数据突发从暂停状态继续时，warmup 周期不会重新发送。

1.4.1.6 DDR

Double data rate 的首字母缩写。

1.4.1.7 缺陷区域

缺陷区域是由制造商标识出的制造缺陷。参见 3.3。

1.4.1.8 取消选择(ODT 状态)

当使用 on-die termination(ODT)时，LUN 可能处于取消选择，选定或 Sniff 状态，并执行每个相关联的操作。参见 4.16。

1.4.1.9 器件(device)

封装的 NAND 单元。一个器件由一个或多个 NAND 对象组成。

1.4.1.10 差分信号

差分信号由两个互补信号组成用于信息传输。与此相对的技术是单端信号。每个 RE_n 和 DQS 信号都有互补信号，以提高抗噪声能力。参见 4.10.2。

1.4.1.11 Dword

Dword 是 32bit 的数据，可以被表示为 32bits 的两个相邻的字(word)，或者四个相邻的字节(byte)。当表示为位(bit)时，最低位是 bit0，最高位是 bit31，最高位在左边。当表示为字时，word 0 为最低，word 1 为最高。当表示为字节时，最低字节为 byte 0，最高字节为 byte 3。见图 1 byte,word,Dword 之间关系的描述。

1.4.1.12 主机对象(Host target)

一组 NAND 对象共用相同的 host CE_n 信号。如果不使用 CE_n reduction，则 Host 对象等效于 NAND 对象。

1.4.1.13 锁存边沿(latching edge)

锁存边沿描述了用来锁存数据总线内容的 CLK,RE_n，WE_n 或 DQS 信号的沿。

对 NV-DDR,数据周期的的锁存沿是 DQS 信号的上升沿和下降沿,命令和地址周期的锁存沿是 CLK 信号的上升沿。

对 NV-DDR2 和 NV-DDR3，数据周期的锁存沿是 DQS 信号的上升沿和下降沿，命令和地址周期的锁存沿是 WE_n 信号的上升沿。

1.4.1.14 LUN(logical unit number)

可独立执行命令和报告状态的最小单元。每个 NAND 对象都有一个或多个 LUNs。

1.4.1.15 na

“not applicable”。标为 na 的字段表示不可用。

1.4.1.16 NAND 对象(NAND Target)

在一个 NAND 封装中共用一个 CE_n 信号的一组 LUNs。

1.4.1.17 O/M

Optional/Mandatory-可选/强制。

1.4.1.18 on-die termination(ODT)

由 NAND 器件提供的一种电路终端。参见 4.16。

1.4.1.19 页(Page)

用于读操作和编程操作的最小可寻址单元。

1.4.1.20 页寄存器

用于读取从闪存阵列传输的数据的寄存器。对于编程操作，数据在传输到闪存阵列之前存放在该寄存器中。如果支持 EZ NAND，则 EZ NAND 控制器中会有一个缓存用于回拷(copyback)操作。参见 5.18 EZ NAND 信息的回拷操作。

1.4.1.21 部分页(partial page,已过时)

Page 的一部分，如果参数页表明 NAND 对象中每页支持多于一个编程操作，则可被编程。在读操作中，host 可以选择从页寄存器中只读取部分数据，这部分也称为部分页。

1.4.1.22 读请求

读请求是 host 请求的一个数据输出周期，由此，数据从 device 传输到 host。参见 4.3 数据输出周期。

1.4.1.23 行

要访问的块和页。

1.4.1.24 选定（ODT 状态）

当使用 on-die termination(ODT)时，LUN 可能处于取消选择，选定或 Sniff 状态，并执行每个相关联的操作。参见 4.16。

1.4.1.25 单端信号(Single-ended signaling,ODT 状态)

单端信号是使用一根信号来传输信息。与此相对应的技术是差分信号。

1.4.1.26 Sniff(ODT 状态)

当使用 on-die termination(ODT)时，LUN 可能处于取消选择，选定或 Sniff 状态，并执行每个相关联的操作。参见 4.16。

1.4.1.27 源同步（Source synchronous）

源同步是指，选通脉冲(DQS)和数据一起被发送，用来指示数据何时被锁存。选通脉冲信号 DQS 可被认为是附加的数据总线位。

1.4.1.28 SR[]

SR 是指 LUN 中的状态寄存器。SR[x]指相关 LUN 中状态寄存器的第 x 位。参见 5.13 状态寄存器中的位定义。

1.4.1.29 对象(target)

该术语等效于 NAND 对象。当 Host target 和 NAND target 不会混淆时，使用“target”。

1.4.1.30 不可纠正的误码率(UBER-Uncorrectable Bit Error Rate,or Ratio)

数据错误发生率，等于错误数据的数量除以读比特数(bits)。用以下公式表示：

UBER = 累积的错误数据数量/累积读取的比特数

注意：读取的累积比特数，即通过多次读取相同的存储位(和多比特读取相同的位置)，从 device 读回的所有数据比特的总和。例如，一个 100GB 的 device 被读取 10 次，则总共有 1TB($8x10^{12}$bits)数据被读取。累积的错误数据量是 device 无法返回正确数据的物理 pages 的总数。

注意：该度量方式仅用于支持 EZ NAND 的 device。EZ NAND 提供了一种 ECC offload 的解决方案。对于 host 提供 ECC 方法的行 NAND 方案，UBER 取决于 host 控制器的功能，这种情况下的 UBER 不在本规范中讲述。

1.4.1.31 Volume

Volume 是一个指向 NAND 对象的地址。Volume 做为 volume 编址的一部分来使用，参见 2.20。

1.4.1.32 VREFQ

输入参考电压。

1.4.1.33 Vtt

Termination 电压

1.4.1.34 字(word)

一个字是 16 位的数据(16 bits)。一个字可以使用 16 bits 或者两个相邻的字节(bytes)来表述。当做为 bits 表述时最低位是 bit0,最高位是 bit15，最高位在左边。当用 byte 表述时，最低字节是 byte0，最高字节是 byte1。参见图 1 字节，字和双字的关系描述。

1.4.2 约定

名字的首字母或缩写用于信号名时，全部为大写(例如，CE_n)。“_n”表示低有效信号。使用上划线，反斜杠(\)或#号而不是“_n”来表示低有效信号也是可以接受的。“_t”表示逻辑真信号，“_c”表示一对差分信号的互补信号(例如 RE_n 或 DQS)。

只有 1bit 的字段通常称为“name”位而不是“name”字段。

1.4.2.1 优先级

如果文字、图、状态机、时序图以及表之间有冲突，那么优先级应该是状态机，时序图，表，图，最后是文字。

1.4.2.2 关键字

有几个关键字用来区分不同级别的要求。

1.4.2.2.1 强制(mandatory)

表示应该按本规范的定义开发的条目。

1.4.2.2.2 可能(may)

表示选择的灵活性，没有隐含的偏好。

1.4.2.2.3 可选(optional)

不在本规范中描述的特性。

1.4.2.2.4 保留(reserved)

表示保留的位，字节、字、字段、以及用于未来标准的操作码(命令)。这些保留内容使用和解释可能会在本规范或其他规范的未来扩展中说明。一个保留的位，字节，字，字段应被清 0。接收端不应该检查这些保留的位、字节、字或字段。

1.4.2.2.5 应该(shall)

表示一个强制要求。要求设计者要开发所有这种强制要求，以保证其他符合本规范的产品的互用性。

1.4.2.2.6 应该(should)

被推荐的要求。

1.4.2.3 字节(Byte)，字(word)和双字(Dword)的关系

图 1 展示了字节，字和双字的关系。

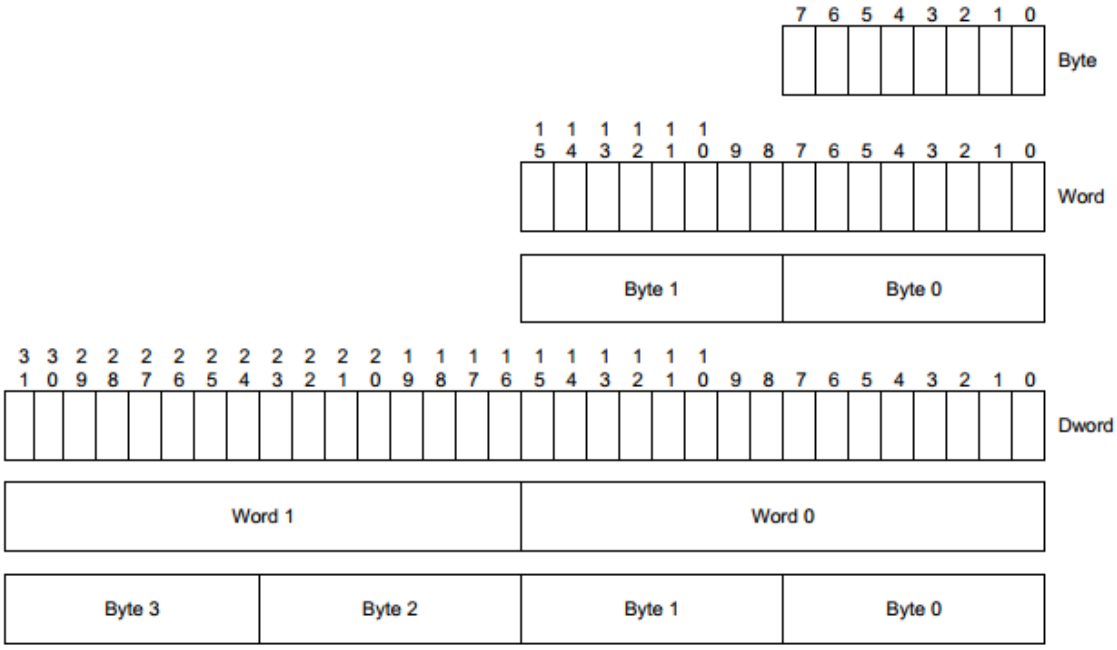


Figure 1 Byte, word and Dword relationships

1.4.2.4 行为流程图

每个功能都有一个状态机来描述其序列。每个功能有几种状态来完成既定目标。每个状态由一个独立的状态表来描述。下表 1 展示了每个状态表的总体布局，状态表包含对应功能的一组状态。

State name	Action list		
	Transition condition 0	→	**Next state 0**
	Transition condition 1	→	**Next state 1**

Table 1 State Table Cell Description

每个状态由唯一的状态名标识。状态名是该状态期间主要操作的简要说明。Action list 里面描述了该状态期间的主要操作。

每个转换由一个转换标签和一个转换条件标识。转换条件是对引起状态转换的事件或条件的简短描述。转换条件按优先级顺序排列，不要求各个条件之间相互独立。第一个为真的条件会被执行。

当进入一个状态时，该状态所有的操作都会被执行，当从该状态本身重新再次进入该状态时，所有操作将再次被执行。

假定一个状态内的所有操作都被执行，并且状态间的转换是瞬间完成的。

2. 物理接口

本章 2.1-2.6 内容关于封装形式及引脚分配，请自行参考原文档。

2.7 信号描述

信号名	方向	描述
R/B_x_n	O	**Ready/Busy** 表示对象状态：信号为低，表示一个或多个 LUN 操作正在处理中(busy)；信号为高，表示 ready。该信号是漏级开漏输出，因此要加额外的上拉，参见 2.18
RE_x_n (RE_x_t)	I	**Read Enable(True)** 该信号用于使能串行数据输出；在 NV-DDR 接口中，该信号与 W/R_x_n 复用相同引脚。
RE_x_c	I	**Read Enable Complement** 该信号与 RE_x_n(RE_x_t)为互补信号，在 NV-DDR2 或 NV-DDR3 接口中为可选信号。当 CE_n 为低时，该信号的值与 RE_x_n(RE_x_t)的值相反，例如，如果 RE_x_t 为高，则 RE_x_c 为低；如果 RE_x_t 为低，则 RE_x_c 为高。
W/R_x_n	I	**Write/Read Direction** 该信号表示 NV-DDR 接口中 DQ 总线和 DQS 信号的使用者。在 SDR，NV-DDR2 和 NV-DDR3 接口中，该信号和 RE_x_n 信号复用相同的引脚。
CE_x_n	I	**Chip Enable** 该信号用于选择对象。当该信号为高，且对象为 ready 状态时，对象会进入低功耗待机状态(low-power standby)。当该信号为低时，对象被选中。参见 2.8。
Vcc	I	**Power** Device 的电源信号
VccQ	I	**I/O Power** 输入/输出信号的电源供电信号，参见 2.10.1
Vss	I	**Ground** 电源地信号
VssQ	I	**I/O Ground** 输入/输出信号的地，参见 2.10.1
VREFQ_x	I	**Voltage Reference** 当 NV-DDR2 或 NV-DDR3 接口被选中时，该信号用作输入和 I/O 信号的外部电压参考。当 SDR 或 NV-DDR 接口被选中时，该信号不使用。
VDDi	Na	**ASIC Voltage Control** 该信号通过连接外部电容来辅助稳定内部电源；内部电源用于向 NAND 控制器 ASIC(例如 EZ NAND)供电。
Vpp	I	**High Voltage Power** 该信号是可选的用于 device 供电的外部高压电源信号。该高压电源可用于增强擦除和编程操作(例如，提高电源效率)
CLE_x	I	**Command Latch Enable** Host 信号之一，用以表明总线周期的类型(命令，数据，地址)。参见 4.3
ALE_x	I	**Address Latch Enable** Host 信号之一，用以表明总线周期的类型(命令，数据，地址)。参见 4.3
WE_x_n	I	**Write Enable** 写使能信号，用于在 SDR 接口中控制命令，地址和输入数据的锁存；或在 NV-DDR2 或 NV-DDR3 接口中控制命令和地址锁存。数据，命令和地址在 WE_x_n 上升沿被锁存。该信号在 NV-DDR 接口中和 CLK_x 信号复用相同引脚。
CLK_x	I	**Clock** NV-DDR 接口中的时钟信号。该信号在 SDR，NV-DDR2 和 NV-DDR3 接口中和 WE_x_n 信号复用相同引脚。
WP_x_n	I	**Write Protect** 写保护信号，disable 闪存阵列的编程和擦除操作，参见 2.19。

IO0_0-IO7_0 (DQ0_0-DQ7_0)	I/O	**I/O Port0,bits 0-7** I/O 端口是一个 8 位宽的双向端口，用于向 device 和从 device 传送地址，命令和数据。在 NV-DDR，NV-DDR2 和 NV-DDR3 接口中为 DQ0_0-DQ7_0。
DQS (DQS_x_t)	I/O	**Data Strobe(True)** 数据选通信号，表示 NV-DDR 和 NV-DDR2 接口中数据有效窗口。
DQS_x_c	I/O	**Data Strobe Complement** 数据选通互补信号，在 NV-DDR2 或 NV-DDR3 接口中可选。当 CE_n 为低的情况下，该信号具有和 DQS(DQS_x_c)信号相反的值，例如，如果 DQS_x_t 为高，则 DQS_x_c 为低；如果 DQS_x_t 为低，则 DQS_x_c 为高。
IO8-IO15	I/O	**I/O Port0,bits 8-15** 用于 16 位宽的目标配置。该信号为 16 位双向端口的高 8 位，该 16 位双向端口用于从 device 或向 device 传送数据。该信号仅用在 SDR 接口中。
IO0_1-IO7_1 (DQ0_1-DQ7_1)	I/O	**I/O Port1,bits 0-7** 用于从 device 或向 device 传送地址，命令和数据的 8 位宽的双向端口。该引脚可以用在支持独立双数据总线的 device 上，作为额外的 8 位宽的双向端口。在 NV-DDR，NV-DDR2 和 NV-DDR3 接口中为 DQ0_1-DQ7_1。
ENo	O	**Enumeration output** 可选的输出信号，用于 CE_n 减少枚举，参见 2.20
Eni	I	**Enumeration input** 可选的输入信号，用于 CE_n 减少枚举，参见 2.20
VSP_x		**Vendor Specific** 该组信号功能由 NAND 制造商来定义。Device 应该在这些信号连接内部上拉或者下拉电阻，以便该信号没有被 Host 连接时能产生符合 ONFI 的行为。NAND 制造商没有使用的任何 VSP 信号都不应该被连到 device。使用者应将 VSP 信号当作 NU 信号对待。
R		**Reserved** 该引脚不能被 host 连接
RFT		**Reserved for Test** 该引脚不能被 Host 连接
NU		**Not Usable** 在正常应用中没有被用到的引脚。此类引脚可以或可以没有内部连接。
NC		**No(internal)connection** 没有内部连接的引脚。
ZQ_x	na	**Reference pin for ZQ calibration**

表 2 信号描述

表 3-表 5 描述了不同类型封装中引脚的分布。如果需要请自行察看原文档。

2.8 CE_n 信号要求

如果一个或多个 LUN 是 active 的并且 host 设置 CE_n 为 1，则其操作继续执行直到 NAND 对象进入待机状态。在 CE_n 变为 1 之后，host 会驱动另一个不同的 CE_n 信号到 0，并启动另一个 NAND 对象的操作。注意，对于双数 x8 的封装(如 BGA-100),如果两个不同的 CE_n 信号被连接到不同的 8 位数据总线上，那么操作会在 2 个 CE_n 信号上并行执行。

当一个 LUN 的 SR[6]被清零，并且相应的 NAND 对象的 CE_n 信号被拉低，则 Host 可能只会发送 Reset,同步 Reset,Reset LUN,读状态，增强的读状态或者 Volume 选择命令到这个 LUN。

2.8.1 CLK 要求(NV-DDR)

当使用 NV-DDR 接口时，如果在数据输入期间 device 不支持停止 CLK，则应符合以下要求：

1. 仅当 CE_n 为高时，CLK 才可被停止或启动。

当使用 NV-DDR 接口时，如果在数据输入期间 device 支持停止 CLK，则应符合以下要求：

1. CLK 仅在以下两种情况下才能被停止或启动：
   a. CE_n 为高，或者
   b. CE_n 为低并且总线状态为数据输入

当使用 NV-DDR 接口时，以下要求应该始终被满足：

1. CLK 应当仅在 CE_n 为高时才能变换频率
2. 当 CE_n 为低时，CLK 应保持在相同的频率
3. CE_n 应该仅在 CLK 已稳定，且具有有效的周期时才能从 1 变为 0(CLK 有效周期基于时序模式的选择)。
4. 当 CE_n 的值改变时，接口应改处于 idle 状态(参见 4.3)。CE_n 应当仅在以下条件为真时转变：
   a. ALE 和 CLE 都被清 0，并且
   b. 当前时钟周期内，DQ/DQS 信号上没有数据传输。

2.9 DC 等级的绝对最大值

如果强度大于表 6 所列的值，则会引起 device 的永久损坏。该表只是强度等级。不推荐执行的操作超出表 7 所列的条件，也不推荐操作超出表 11 和表 12 所列的 DC 和操作特性。除 2.11 中的定义外，超出以上条件的操作会影响 device 的可靠性。

表 6 定义了 device 上和 Vss/VssQ 有关的基于 Vcc 和 VccQ 典型电压的引脚电压。

Parameter	Symbol	Rating	Units
VPP Supply Voltage	V_{PP}	-0.6 to +16	V
Vcc = 3.3V and VccQ = 3.3V nominal			
Vcc Supply Voltage	V_{CC}	-0.6 to +4.6	V
Voltage Input	V_{IN}	-0.6 to +4.6	
VccQ Supply Voltage	V_{CCQ}	-0.6 to +4.6	
Vcc = 3.3V and VccQ = 1.8V nominal			
Vcc Supply Voltage	V_{CC}	-0.6 to +4.6	V
Voltage Input	V_{IN}	-0.45 to +2.4	
VccQ Supply Voltage	V_{CCQ}	-0.45 to +2.4	
Vcc = 3.3V and VccQ = 1.2V nominal			
Vcc Supply Voltage	V_{CC}	-0.6 to +4.6	V
Voltage Input	V_{IN}	-0.4 to +1.5	
VccQ Supply Voltage	V_{CCQ}	-0.4 to +1.5	
Vcc = 1.8V and VccQ = 1.8V nominal			
Vcc Supply Voltage	V_{CC}	-0.45 to +2.4	V
Voltage Input	V_{IN}	-0.45 to +2.4	
VccQ Supply Voltage	V_{CCQ}	-0.45 to +2.4	

Table 6 Absolute maximum DC ratings

2.10 推荐的 DC 操作条件

3.3V 或 1.8V VccQ 的操作条件可用于 SDR 或 NV-DDR 接口。1.8V VccQ 的操作条件应被用于 NV-DDR2 接口。1.2V VccQ 的操作条件应被用于 NV-DDR3 接口。

Parameter	Symbol	Min	Typ	Max	Units
Supply voltage for 3.3V devices	V_{CC}	2.7	3.3	3.6	V
Supply voltage for 1.8V devices	V_{CC}	1.7	1.8	1.95	V
Supply voltage for 3.3V I/O signaling[1]	V_{CCQ}	2.7	3.3	3.6	V
Supply voltage for 1.8V I/O signaling	V_{CCQ}	1.7	1.8	1.95	V
Supply voltage for 1.2V I/O signaling	V_{CCQ}	1.14	1.2	1.26	V
Ground voltage supply	V_{SS}	0	0	0	V
Ground voltage supply for I/O signaling	V_{SSQ}	0	0	0	V
External voltage supply[2]	V_{PP}	10.8	12.0	13.2	V

NOTE:
1. 3.3V VccQ is not supported for NV-DDR2.
2. 3.3V and 1.8V VccQ are not supported for NV-DDR3.
3. The maximum external voltage supply current (I_{PP}) is 5 mA per LUN.

Table 7 Recommended DC operating conditions

2.10.1 I/O 电源(VccQ)和 I/O 地(VssQ)

VccQ 和 Vcc 电压是唯一值。VccQ 应当小于等于 Vcc，包括在 power-on ramp 期间。Device 应该支持以下一种 VccQ/Vcc。

- Vcc = 3.3V， VccQ = 3.3V
- Vcc = 3.3V， VccQ = 1.8V
- Vcc = 3.3V， VccQ = 1.2V
- Vcc = 1.8V， VccQ = 1.8V

所有参数、时序模式、以及其他特性都和支持的电压组合有关。

如果一个 device 具有相同的 Vcc 和 VccQ 电压，则 VccQ 和 VssQ 不要求被内部连接到 device。Device 可能会使用唯一的 Vcc 和 Vss 作为 I/O 和 core 的电源。

2.11 AC 上冲(Overshoot)/下冲(Undershoot)要求

略

2.12 DC 和操作特性

略

2.13 引脚电容计算

略

2.14

略

2.15 电源周期要求

略

2.16 独立的数据总线

在一些 ONFI 封装中可能有两种独立的 8 位数据总线(例如 LGA 和 100-ball 的 BGA)。对于既支持两个独立数据总线，也支持单个数据总线的封装，CE0_n 和 CE2_n 应使用相同的引脚作为第一个数据总线的 CE_n 引脚(标记为 CE0_0_n 和 CE1_0_n)，CE1_n 和 CE3_n 应使用相同的引脚作为第二个数据总线的 CE_n 引脚(标为 CE0_1_n 和 CE1_1_n)。

注意，如果 device 不支持独立的数据总线，那么 CE0_n，CE1_n,CE2_n 和 CE3_n 可以都使用第一个数据总线和第一组控制信号(RE0_n,CLE0_n,ALE0_n,WE0_n 以及 WP0_n)。

在 152-ball BGA 中，有 8 个 CE_n 信号，只有 4 个 R/B_n 信号。表 18 描述了这种情况下每个 CE_n 使用 R/B_n 信号的情况。

Signal Name	CE_n
R/B0_0_n	CE0_0_n, CE2_0_n
R/B0_1_n	CE0_1_n, CE2_1_n
R/B1_0_n	CE1_0_n, CE3_0_n
R/B1_1_n	CE1_1_n, CE3_1_n

Table 18　　R/B_n Signal Use Per CE_n

2.17 总线宽度要求

每个 device 的所有 NAND 对象都应该使用相同宽度的数据总线。所有对象都应该既可以有一个 8 位宽的总线，也可以有一个 16 位宽的总线。注意，支持 NV-DDR，NV-DDR2 或 NV-DDR3 接口的 device 应该具有 8 位宽的总线。

当 Host 支持 16 位宽的总线时，只有数据可以在 16 位宽的总线上传输。所有地址和命令的传输只能使用数据总线的低 8 位。在命令传输期间，host 可能会给数据总线的高 8 位赋予一个值。在地址传输期间，host 应该设置书记总线的高 8 位为 00h。

2.18 Ready/Busy(R/B_n)要求

2.18.1 上电要求

一旦 Vcc 和 VccQ 达到表 7 所列的最小值，并且电源稳定后，R/B_n 信号应该在 RB_valid_Vcc 之后有效，并且在 RB_device_ready 期间应该被设为 1(Ready)，如表 19 所示。R/B_n 直到 Vcc 开始 ramp 之后 50us 才被定义。R/B_n 信号直到两个条件都满足后才有效。

Parameter	Raw NAND	EZ NAND
RB_valid_Vcc	10 µs	250 µs
RB_device_ready	1 ms	2 ms

Table 19　　R/B_n Power-on Requirements

在上电期间，任何时候 VccQ 都应小于或等于 Vcc。图 26 展现了 Vcc 之后 VccQ 的 ramping,但实际上，这两信号应该同时 ramp。

Ready/Busy 是一个漏级开漏电路，因此在终端需要使用一个上拉电阻。上拉电阻和 R/B_n 电路电容负载的组合决定了 R/B_n 的上升时间。

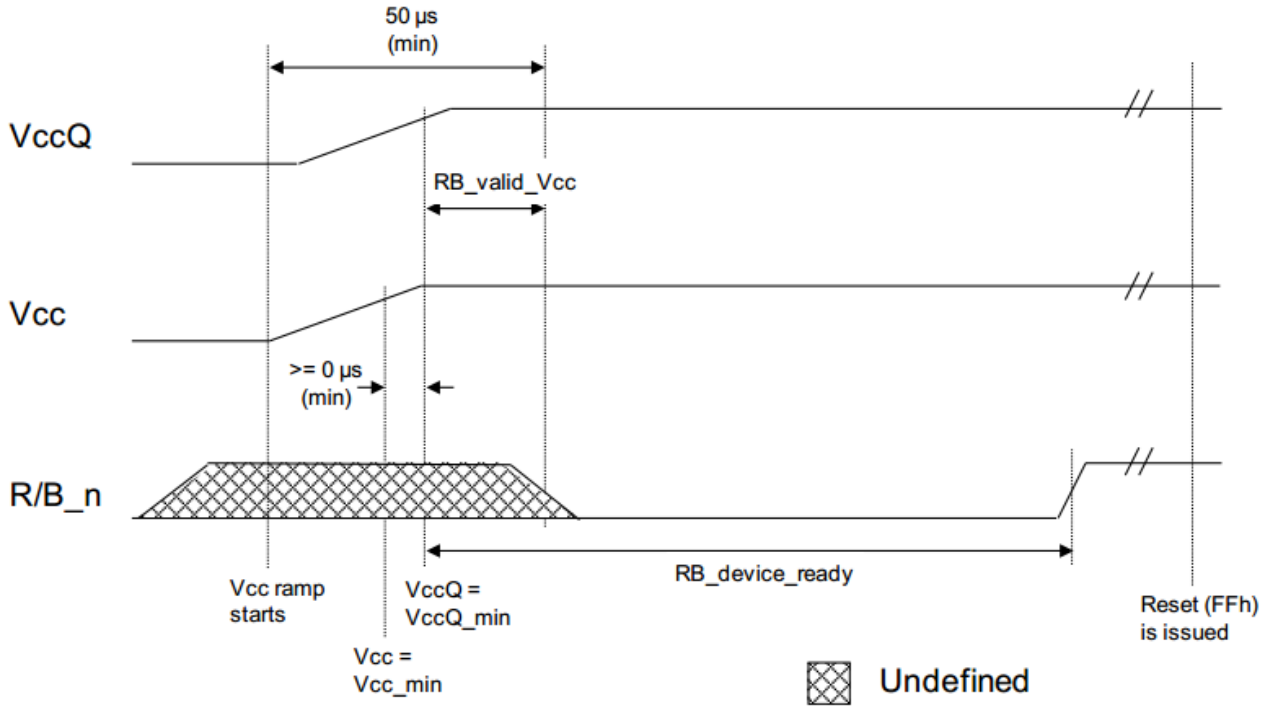


Figure 26　　R/B_n Power-On Behavior

2.18.2　R/B_n 和 SR[6]的关系

R/B_n 应该是对应 NAND 对象或 Volume 上所有 LUNs 的 SR[6](状态寄存器 bit6)的值的逻辑与。例如，R/B3_n 是所有 CE3_n 上 LUNs 的 SR[6]值的逻辑与。因此,R/B_n 反映的是特定 NAND 对象上的任何 LUN 是否处于 busy 状态。

2.19 写保护

当 WP_n 信号为 0 时，会禁止闪存阵列的编程和擦除操作。该信号仅可在 device 上没有命令执行时才能转变。在 WP_n 的值转变后，host 应改至少在 tWW 延时后才能向 device 发送新的命令。

图 27 以编程命令为例，描述了 tWW 时间要求。在 NV-DDR 接口中，WP_n 信号的转变对任何 CLK 都是异步的，和任何 CLK 都没有关系。在 WP_n 信号从 0 变为 1 后，Host 发送新的命令之前(包括编程)，总线应保持 tWW 时间的 idle 状态。WP_n 信号从 1 变为 0 后，host 发送新命令之前，总线应保持 tWW 时间的 idle。

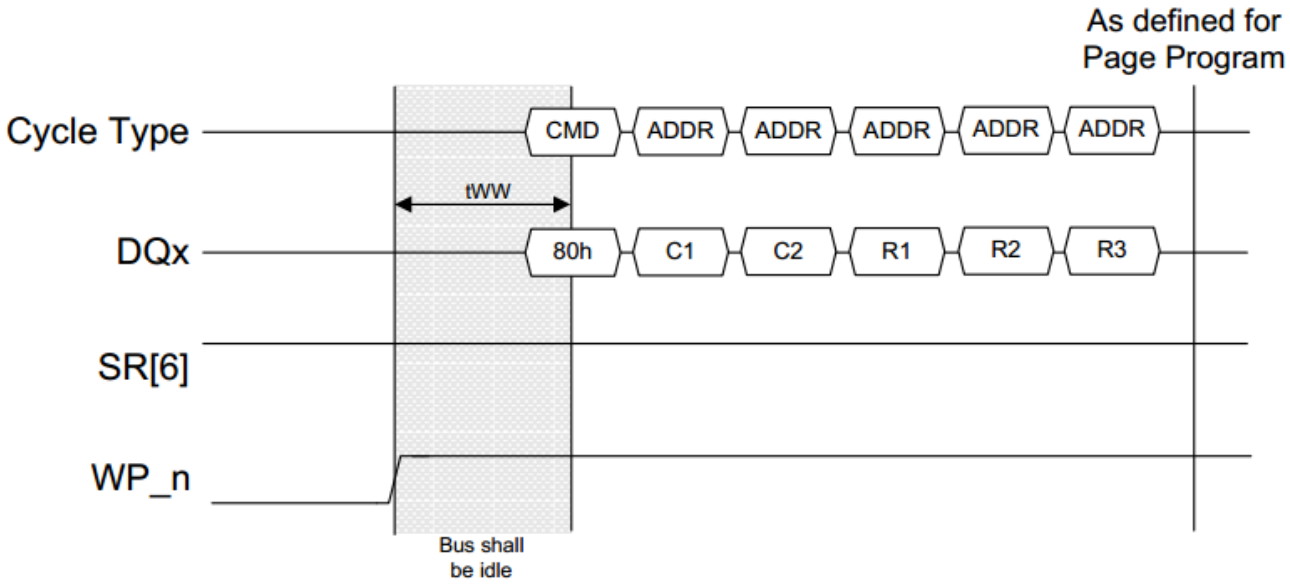


Figure 27　　Write Protect timing requirements, example

2.20 CE_n 引脚 Reduction 机制

在高容量的设计中可能会有很多数量的 CE_n 引脚，这种设计会包含有很多 NAND 封装，每个封装有 2 到 8 个 CE_n 引脚。CE_n 引脚 reduction 机制可以让 host 的单个 CE_n 引脚被多个 NAND 对象复用，因此可以使 Host 需要的 CE_n 引脚数目大大减少。CE_n 引脚 reduction 机制可以使用在数据接口(SDR,NV-DDR,NV-DDR2 或 NV-DDR3)中。

CE_n 引脚 reduction 机制在初始化过程中，为每个 NAND 对象指定一个 Volume 地址。初始化完成后，host 可以通过 Volume 选择命令来寻址特定的 Volume(例如，NAND 对象)。

图 28 展示了一个使用 CE_n 引脚 reduction 机制的拓扑的例子。每个 NAND 封装都增加了 ENi 和 ENo 引脚，而在 NAND 封装之间有一个 daisy chain。Chain 中第一个 NAND 封装的 ENi 没有被连接，而其它所有 NAND 封装的 Eni 引脚都连到前一个封装的 ENo 引脚。

图 29 展示了一个更复杂的拓扑。该拓扑中，多个 NAND 对象可以被连接到一个单个的 host 对象(例如，CE_n 信号)。

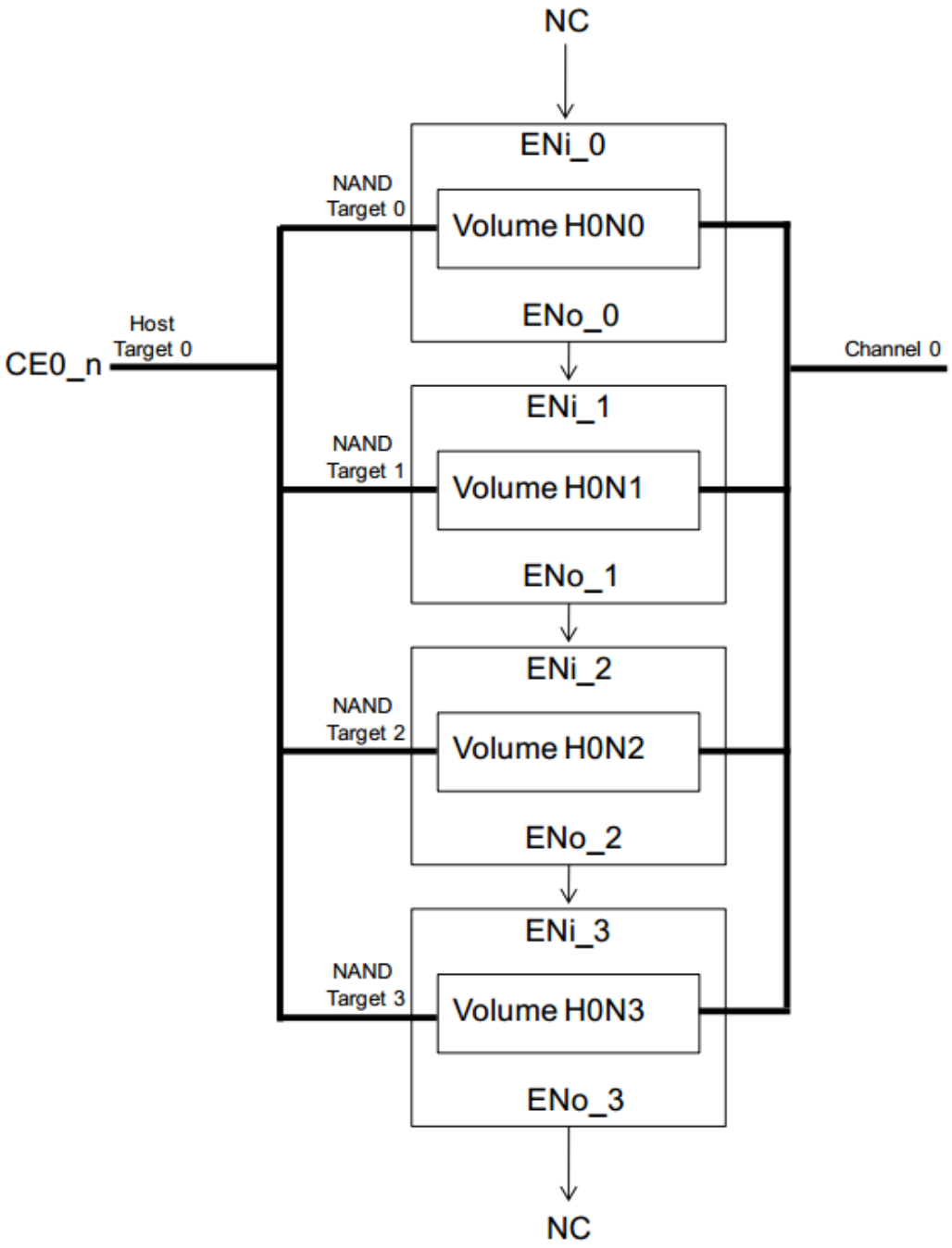


Figure 28　　CE_n Pin Reduction Topology, example 1

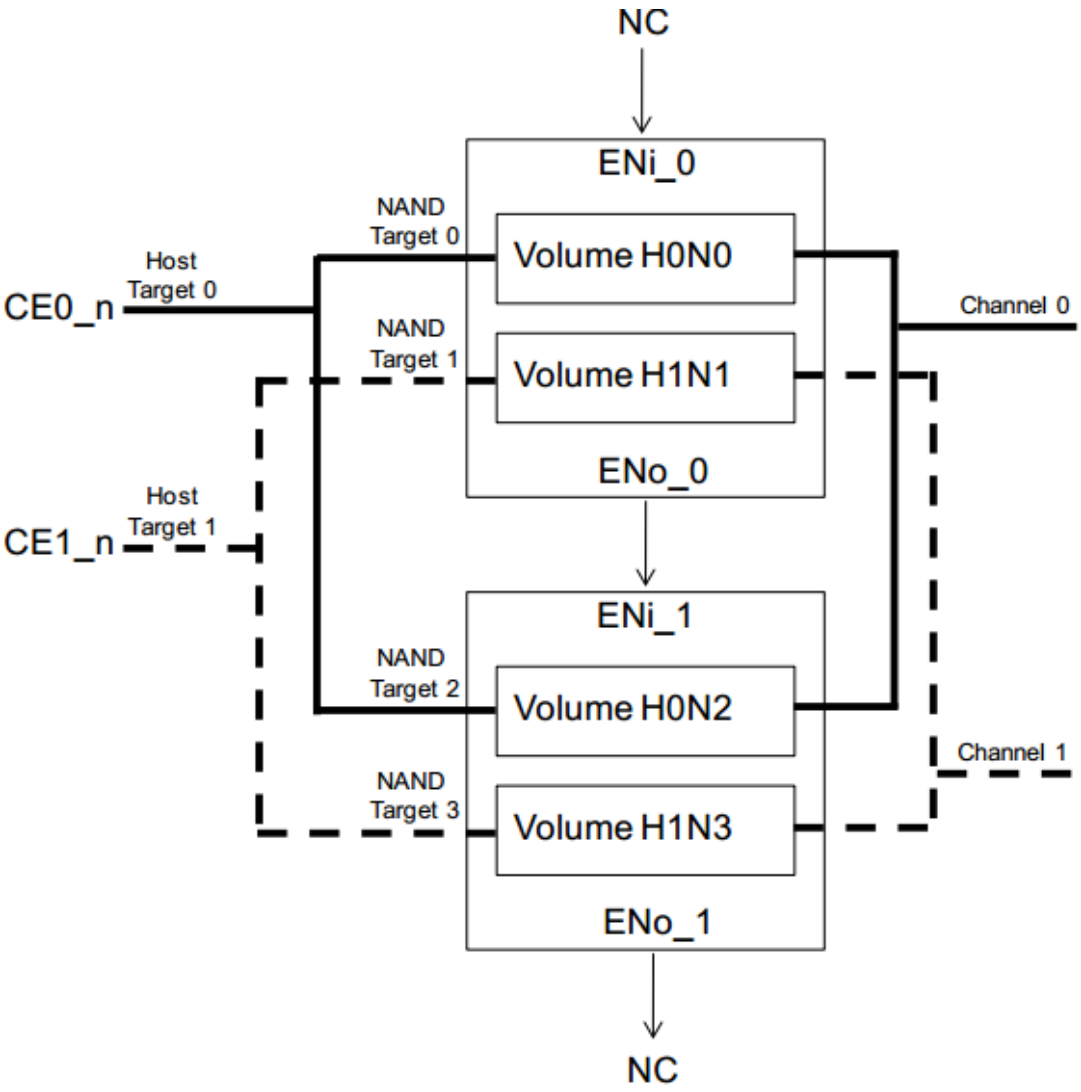


Figure 29　　CE_n Pin Reduction Topology, example 2

不是所有拓扑都支持 CE_n 引脚 reduction 机制。如果一个 NAND 封装中有两个 NAND 对象共享了同一个数据总线，那么每个 NAND 对象在 NAND 封装外都应该有自己独立的 CE_n 引脚。这种情况下，host 应该对每个 NAND 对象使用独立的 host 对象(CE_n 信号)。

ENi 决定了 NAND 封装是否可以接受命令。ENi 由 NAND 封装内部拉高。如果 NAND 对象的 ENi 引脚为高，并且 CE_n 为低，则 NAND 对象可以接受命令；如果 NAND 对象的 ENi 为低，则 NAND 对象不能接受命令。注意：上电后发送的第一个命令是特殊情况，参见 3.5.2 初始化序列。

当 CE_n 为低并且 NAND 对象没有被指定 Volume 地址时，ENo 被 device 驱动为低。当与 NAND 对象关联的 CE_n 为低并且该 NAND 对象被指定了一个 Volume 地址时，ENo 被 device 驱动为三态。当共享一个 ENo 信号的所有目标 NAND 的 CE_n 信号为高时，ENo 被 device 驱动为三态。注意，在 Volume 地址被指定后，如果后一个封装的 ENi 或 ENo 悬空(没有连到其下一个封装)，则 ENo 被后一个封装悬空的 ENi 或 ENo 拉高。

当一个 Volume 地址被指定给一个 NAND 对象后，该 Volume 地址会变为取消选择状态(deselected),并忽略 ENi 引脚直到下一个电源周期。

为了被选定来处理命令，一个 Volume 选择命令应该被发送到 host 对象，该 Volume 选择命令使用之前被指定的 Volume 地址。在 CE_n 信号被拉高 tCEH 时间后，Volume 上所有 LUNs 返回之前的状态(参见 3.2.4)。

2.20.1　　当不支持 CE_n Reduction 时 Volume 的指定

图 30 展现了一个不支持 CE_n reduction 的拓扑。如果 CE_n reduction 不被使用(例如 ENi 和 ENo 没有被连接)，并且 host 期望在封装上有一个终端(terminator)，该封装不和选定的 NAND 对象共享 CE_n，那么每个 NAND 对象都可作为终端(terminator),每个 NAND 对象都应该在初始化时被指定一个 Volume,该 Volume 的指定是通过使用 Volume 配置特性的 Set Features 命令来完成的。

每个 CE_n 都应该各自被拉低，并被指定一个唯一的 Volume 地址。一旦所有 NAND 对象都被指定了 Volume 地址，则 Volume 地址可被用于中止选择(termination selection)。

在操作期间，对于选定的 Volume，以及对选定的 Volume 中任何被分配作为 terminator 的 NAND 对象，其 CE_n 信号需要被拉低。当未被选中的 Volume 的 CE_n 信号被拉低，那么对于选定的 Volume，所有没有被分配作为 terminator 的 LUNs 都会被解除选择(deselected)。当 Volume 地址被指定时，应使用 Volume 选择命令。

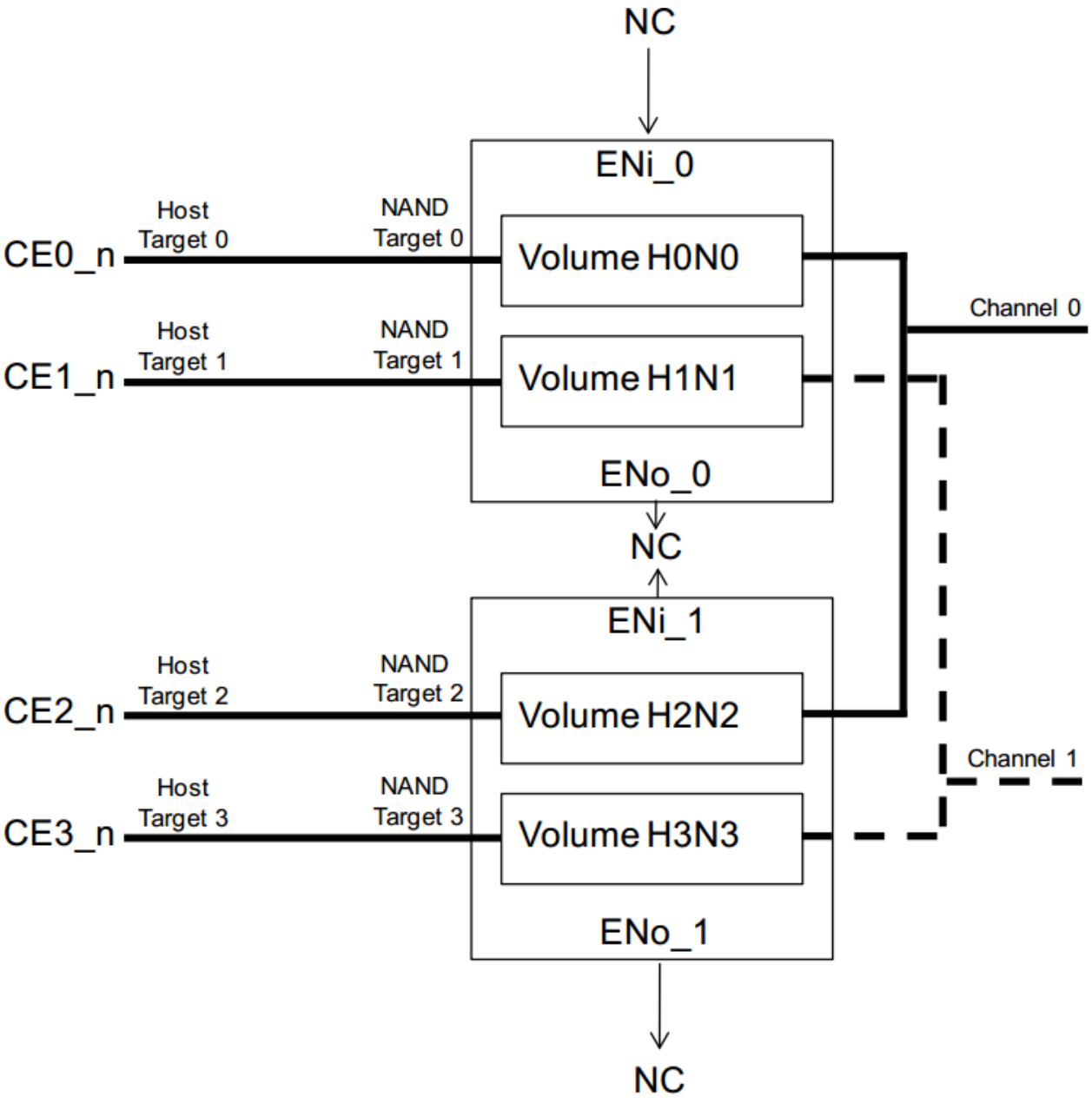


Figure 30 Discrete CE_n per package topology

3. 存储结构(Memory Organization)

图 31 为一个对象 memory 结构的例子。该例中，有两个逻辑单元，每个逻辑单元有两层(plane-NAND 中存储阵列，每个阵列包含若干个 Block)。

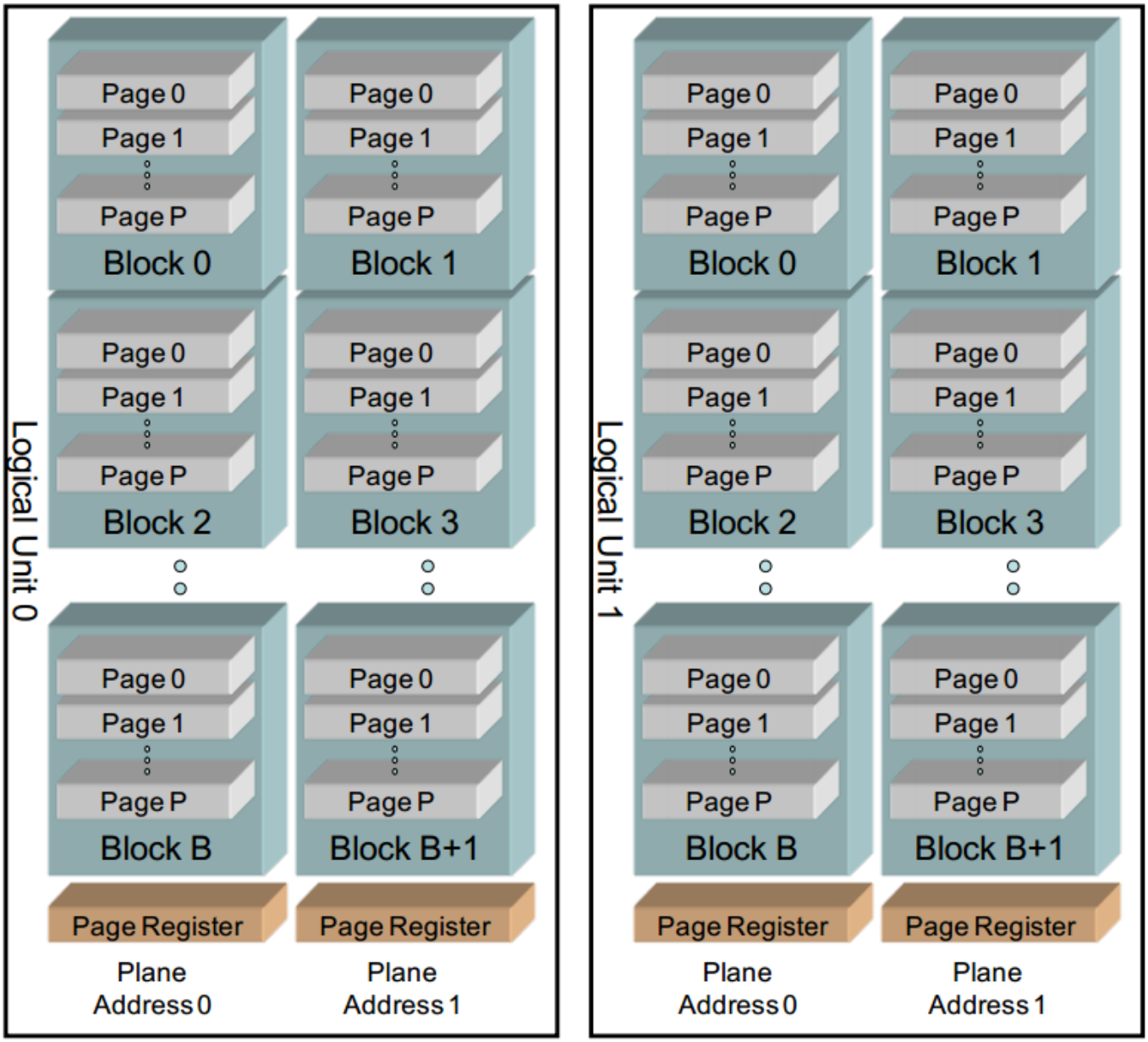


Figure 31　　Target Memory Organization

一个 device 包含一个或多个对象(target)。一个对象由一个 CE_n 信号控制。一个对象位于一个或多个逻辑单元内(LUN-Logical Unit)。

一个逻辑单元(LUN)是可独立执行命令并报告状态的最小单元。特别是，独立的 LUN 可以并行运行任意的命令序列。例如，允许在 LUN 0 上开始一个 Page 编程操作，然后在该操作完成前，可以在 LUN 1 上开始执行一个读命令。参见 3.1.3 多 LUN 操作规则。一个 LUN 至少包含一个 page 寄存器和一个闪存阵列。页寄存器的数量取决于 LUN 支持的多层(multi-plane)操作的数量。闪存阵列包含若干 blocks。

一个 block 是 LUN 的闪存阵列中可擦除的最小数据单元。LUN 中 block 的数量没有明确的限制。一个 block 包含若干个 pages。

一个 page 是执行读和编程操作的最小可编址单元。一个 page 由若干个字节或字组成。每个 page 中用户数据字节的数量，不含括备用数据区(spare data area)，应该是 2 的次幂。每个 block 的 page 数量应该是 32 的整数倍。

每个 LUN 应该至少有一个 page 寄存器。Page 寄存器在数据被转移到闪存阵列的一个 page 之前，或数据被从

闪存阵列的一个 page 转移出来之后，用来零时存放数据。如果支持 EZ NAND，那么 EZ NAND 的控制器中会有一个缓存(buffer)，用来临时存储从每个 LUN 中的 page 寄存器取出或要存入的数据。

Page 寄存器中的字节或字的位置被称为列。

对这种结构，由两种机制可以达到并行操作的目的。同一时间可以有多个命令发送到不同的 LUNs。为了在一个 LUN 中达到更进一步的并行操作，可以使用多层(multi-plane)操作来执行并行的额外 dependent 操作。

3.1 寻址(Addressing)

有两种地址类型：列地址和行地址。列地址用来访问一个 page 中的字节或字。在 NV-DDR，NV-DDR2 和 NV-DDR3 接口中，列地址的最低位应始终为 0。行地址用于寻址 page,block 或 LUN。

当列地址和行地址都被请求时，列地址始终首先在一个或多个 8 位地址周期中被发送，行地址在接下来的一个或多个 8 位地址周期中被发送。一些功能可能只需要行地址，像块擦除(Block Erase)，这种情况下不用发送列地址。

对于列寻址和行寻址，第一个地址周期总是包含最低地址位，而最后一个地址周期总是包含最高地址位。如果行地址和列地址的最高位没有用，则要求最高位清除为 0。

行地址的结构如图 32，最低地址位在右，而最高地址位在左。

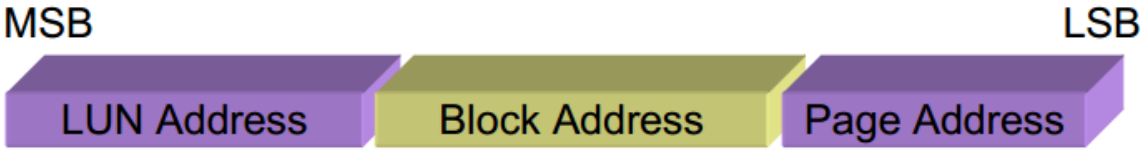


Figure 32 Row Address Layout

Block 的数量以及每个 block 中 page 的数量不要求是 2 的次幂。如果其中一种数量不是 2 的次幂，那么对应的地址应该被舍入到一个整数位(integral number of bits)，由此，其寻址范围可以达到下一个 2 次幂的值。Host 不应该访问不支持范围内的地址。例如，如果每个 block 有 96(1100000b,7bits)个 page，那么 page 的地址应该被舍入到 7bits，这样，其可在 0-127 的范围内寻址 page。在这种情况下，host 不能访问范围 96-127，因为此范围内并没有 pages。

Page 地址始终使用行地址的低位。Block 地址使用行地址的中间位，而 LUN 地址使用行地址的高位。

3.1.1 多层寻址(multi-plane addressing)

多层地址包含图 33 所示的 block 地址的最低位。当在 LUN 上执行一个多层命令序列时，以下规则应适用于多层地址：

- 层地址位(plane address bit(s))应区别于多层命令序列中的其他任何多层操作。
- Page 地址应该和多层命令序列中的其他任何多层操作相同。

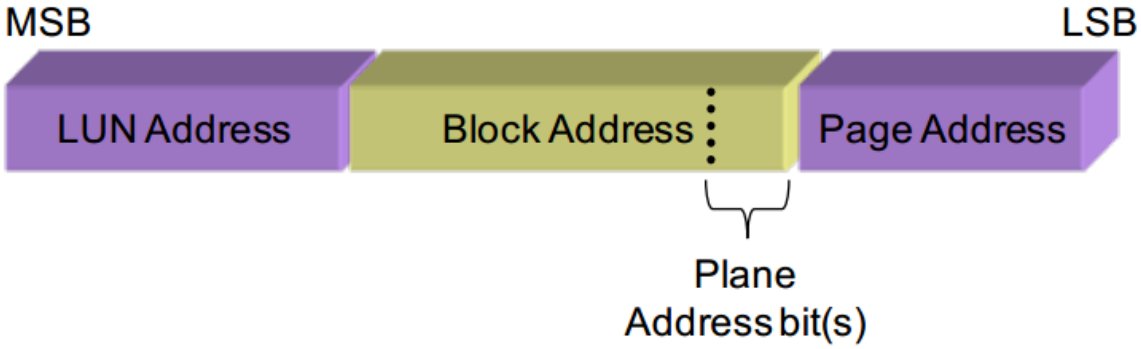


Figure 33 Plane Address Location

3.1.1.1 多层 Block 地址约束

Device 可表明多层 block 地址的约束。详细情况如下：

- 无约束：两层地址之间所有的 block 地址位都不相同
- 完整约束：两层地址之间所有 block 地址位(除了 plane 地址位)都相同
- 低位异或非(XNOR)约束：如果两层地址的最低层地址位(bit0)的 XNOR 为 1，那么这两层地址间有一个

完整约束；如果两层地址的最低层地址位的 XNOR 为 0，那么这两层地址间无约束。

表 20 展示了一个四层操作的 3 中约束类型。

Restriction Type	Plane Address 0	Plane Address 1	Plane Address 2	Plane Address 3
No restriction	Block A	Block B	Block C	Block D
XNOR restriction	Block A	Block B	Block A+2	Block B+2
Full restriction	Block A	Block A+1	Block A+2	Block A+3

Table 20　　Four plane address restriction

表 21 描述了两层地址 A 和 B 之间是否有低位 XNOR 约束。如果有低位 XNOR 约束，那么两层地址 A 和 B 的 block 地址(除了层地址位)应该相同。

Multi-plane Address bits A	Multi-plane Address bits B	Lower Bit XNOR	XNOR Restriction Between A and B
00b	01b	0 XNOR 1 = 0	No
00b	10b	0 XNOR 0 = 1	Yes
00b	11b	0 XNOR 1 = 0	No
01b	10b	1 XNOR 0 = 0	No
01b	11b	1 XNOR 1 = 1	Yes
10b	11b	0 XNOR 1 = 0	No

Table 21　　4-way lower bit XNOR restriction

3.1.2　逻辑单元选择

逻辑单元(LUN)属于 NAND 对象结构的一部分，与 host 共享了一个单独的数据总线。向 host 输出数据时，host 应保证在任何时间只有一个 LUN 被选中，以避免总线冲突。

Host 通过向一个 LUN 发送读状态增强命令(Read Status Enhanced command)来选中该 LUN，以便稍后输出数据。读状态增强命令应该取消选择没有被命令寻址的所有 LUN 的输出路径。LUN 内用于输出的 page 寄存器由之前发送的 Read(Cache)命令选定，并且不受该读状态增强命令的影响。

3.1.3　多逻辑单元(multiple LUN)操作约束

LUN 是独立的实体。多 LUN 操作时指两个以上 LUN 并行处理命令。多 LUN 操作期间，单个的 LUN 可能会处于 busy 或 ready 的任何组合状态。

当一个 page 编程命令(80h)发送到任何一个 LUN，且该 LUN 之前没有接收 11h 命令时，如果 program page register clear enhancement 不被支持或没有被使能，则所有 idle 的 LUN 会清除自己的 page 寄存器。因此，当一个 LUN 中读 page 操作正在进行或者已经完成，但数据还没有从另一个 LUN 被读取，那么 host 不应该在该 LUN 上开始 page 编程命令，否则 page 寄存器中的读操作的内容可能会丢失。当 page 编程正在第二个 LUN 中进行且没有任何约束，则一个读 page 命令可以发送到第一个 LUN 中。如果 page register clear enhancement 被使能，则不使用本约束。

当发送一个 page 编程命令(80h)时，host 在所有数据都被输入并且一个 10h 或 15h 命令被发送前，不能选择同一个 Volume 中的另一个 LUN。多层操作中，应该在所有多层地址的数据输入都完成之后，再选择另一个 LUN。

当发送读命令到多个 LUN 时，host 应设法避免因列地址 corruption 而引发的问题。Host 应该在开始从新选定的 LUN 读出数据之前发送一个 Change Read Column 命令。

如果已经发送了一个多 LUN 操作命令，那么下一个发送的状态命令应该是 Read Status Enhanced 命令。Read Status Enhanced 命令会将没有被选中的 LUN 的输出缓存关闭，以保证只有被 Read Status Enhanced 命令选中的 LUN 会响应下一个数据输出周期。在 Read Status Enhanced 命令结束后，会发送 Read Status 命令直到下一个多 LUN 操作命令被发出。

当 host 同时向多个 LUN 发送了读 page 命令(Read Page)后，host 应在从任何 LUN 读数据之前发送 Read Status Enhanced 命令，以保证通过 00h 命令转换为数据输出模式之后，只有被 Read Status Enhanced 命令选中的 LUN 才

能响应数据输出周期，从而避免总线冲突。对于任何属于多 LUN 读序列的的 LUN，如果之前有读 page(Read Page)命令发向该 LUN 以从该 LUN 传送数据，那么该 LUN 应该请求一个 Change Read Column(Enhanced)命令 。以下是一个序列的例子：

1） 读 page(Read Page)命令发送到 LUN0。
2） 读 page(Read Page)命令发送到 LUN1。
3） Read Status Enhanced 命令选中 LUN0。
4） Change Read Column(Enhanced)命令发送到 LUN0。
5） 数据从 LUN0 传输出来。
6） Read Status Enhanced 命令选中 LUN1。
7） Change Read Column(Enhanced)命令发送到 LUN1。
8） 数据从 LUN1 传输出来。

当多个命令被混合发送到多个 LUN(如，读一个 LUN，编程另一个 LUN)，那么在发送 Read Status Enhanced 命令到选定的 LUN 之后，应发送一个 Change Read Column 或者 Change Read Column Enhanced 命令，然后再从选定的 LUN 输出数据。

在所有 scenarios 中，如果所有的 LUN 都不是 busy,则 host 可以用 Change Read Column Enhanced 命令代替 Read Status Enhanced/Change Read Column 命令。

3.2 Volume 寻址(Volume Addressing)

3.2.1 指定 Volume 地址

为了指定一个 Volume 地址,需要发送一个 Set Feature 命令,该命令带有 Volume 配置(Volume Configuration)的特征地址(Feature Address)，参见 5.30.5。Volume 地址在电源周期后不会被保持，如果要使用 Volume 地址，则每次上电之后都需要重新指定该地址，之后才能访问 NAND device。

3.2.2 选择一个 Volume

在 Volume 地址被指定之后，当 CE_n 被拉低后，其对应的 NAND 对象(以及相应的 LUN)被选中。Host 发送一个 Volume Select 命令给需要执行接下来命令的 Volume(如 NAND 对象)，参见 5.24。

3.2.3 多个 Volume 操作的约束

Volume 是独立的实体。一个多 Volume 操作是指两个及以上的 Volume 并行地处理命令。在向一个没有被选中的 Volume 发送命令之前,CE_n 应该被拉高并保持最小的 tCEH 时间,之后应发送 Volume Select 命令来选择 Volume。当命令(包括多 LUN 操作)正在选定的 Volume 上执行时不能请求 Volume Select 命令。

不支持同时向多个 Volume 发送相同命令。

对于 LUN 级的命令(如读，编程)，host 可以在数据输入或数据输出操作期间选择一个其他的 Volume 执行 LUN 级命令，然后在一段时间之后(LUN 命令完成后？)再恢复数据传输操作。当重新选择一个 Volume 及相关的 LUN 来完成数据输入或数据输出操作时，需要以下操作：

- 数据输入：host 在恢复数据输入前应发送 Change Row Address 命令。如果不支持该命令，则所有数据应该在选择新的 Volume 之前被传输。
- 数据输出：host 在恢复数据输出之前应发送 Change Read Column Enhanced 或者 Random Data Out 命令。如果这两个命令都不支持，那么所有数据应该在选择新的 Volume 之前被传输。

对于对象(Target)级命令(如 Get Features，Set Features)，host 应该在选择新的 Volume 之前，完成该命令对应的所有数据输入或数据输出操作。

在以下读，编程，擦除以及回拷(Copyback)操作期间不能发送 Volume Select 命令：

- 读(包括回拷读-Copyback Read)
  - <CMD:00h> <ADDR:Column & Row> <CMD:30h>
  - <CMD:00h> <ADDR:Column & Row> <CMD:31h>
  - <CMD:00h> <ADDR:Column & Row> <CMD:32h>
  - <CMD:00h> <ADDR:Column & Row> <CMD:35h>

- 编程(包括回拷编程-Copyback Program) **注意**：如果 Volume 的下一个命令是 Change Row Address，则要先发 Volume Select 命令，之后再发 10h,11h,或 15h 命令。
  - <CMD:80h><ADDR:Column & Row><DIN:Data Input><CMD:10h>
  - <CMD:80h><ADDR:Column & Row><DIN:Data Input><CMD:11h>
  - <CMD:80h><ADDR:Column & Row><DIN:Data Input><CMD:15h>
  - <CMD:81h><ADDR:Column & Row><DIN:Data Input><CMD:10h>
  - <CMD:81h><ADDR:Column & Row><DIN:Data Input><CMD:11h>
  - <CMD:81h><ADDR:Column & Row><DIN:Data Input><CMD:15h>
  - <CMD:85h><ADDR:Column & Row><DIN:Data Input><CMD:10h>
  - <CMD:85h><ADDR:Column & Row><DIN:Data Input><CMD:11h>
  - <CMD:85h><ADDR:Column & Row><DIN:Data Input><CMD:15h>

- 擦除
  - <CMD:60h><ADDR:Row><CMD:D0h>
  - <CMD:60h><ADDR:Row><CMD:D1h>
  - <CMD:60h><ADDR:Row><CMD:60h><ADDR:Row><CMD:D1h>

3.2.4 Volume 恢复(Volume Reversion)

在使用 Volume 寻址时，LUN 应该支持 Volume 的恢复。特别是，如果 CE_n 由高变为低，并且 Volume Select 不是第一个命令，那么 LUN 应返回最后一个指定的 Volume 地址的(Selected)，Sniff，以及取消选择(Deselected)状态（在表 71 中定义）。如果使用的是 NV-DDR2 或者 NV-DDR3 接口并且 on-die termination 被使能，那么会有另外的行为，在 4.16 中描述。

图 34 定义了当 CE_n 由高变低时 Volume 恢复的要求。

注意：

1. 当 CE_n 从低变高时，状态的进入是异步的。

2. 对一个选定的 Volume 上 LUN 的 ODT 功能在图 49 中定义。对一个没有被选中的 Volume 上 LUN 的 ODT 功能在图 50 中定义。

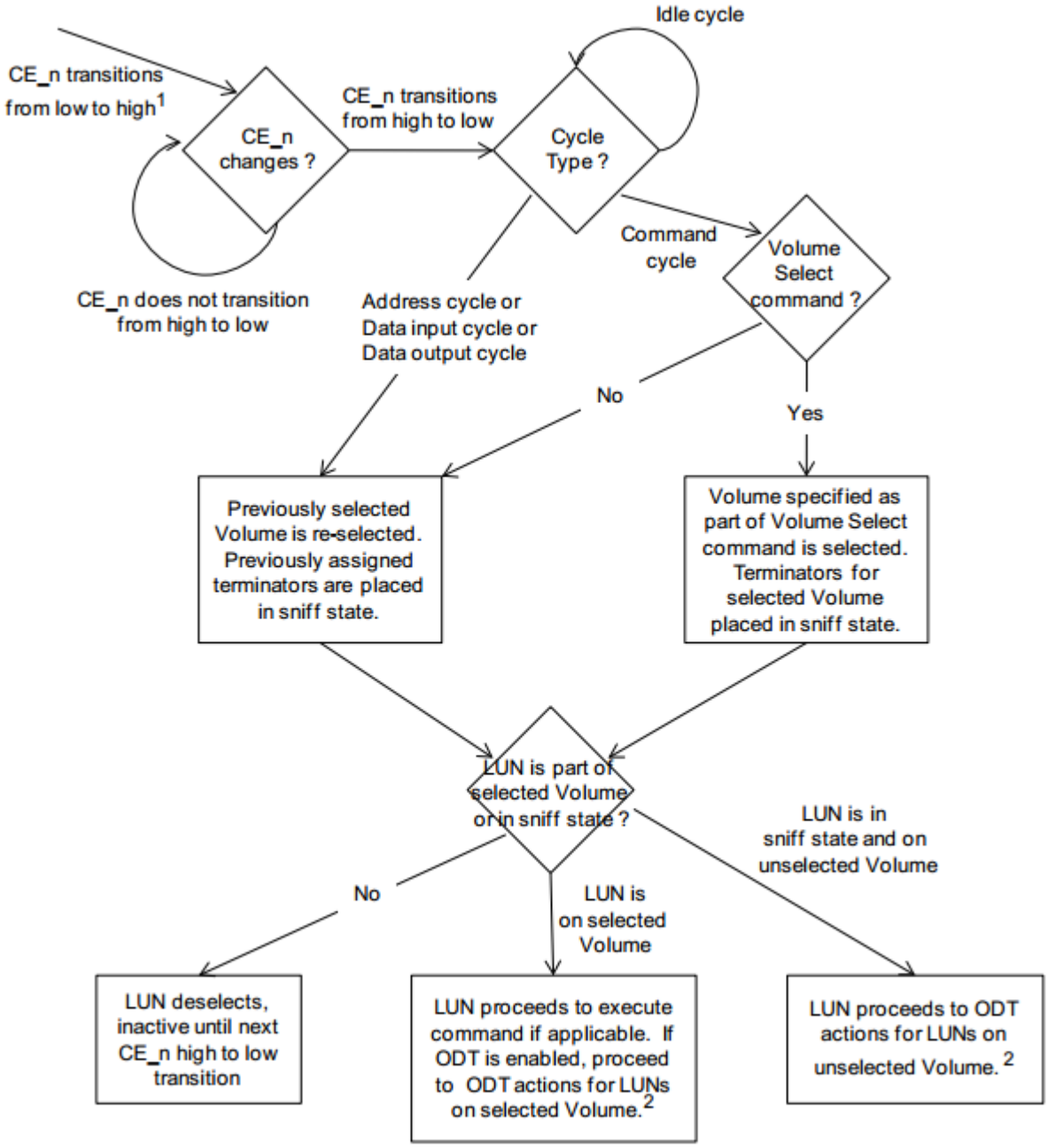


Figure 34 Volume Reversion Behavioral Flow

3.3 工厂缺陷映射

通常假设闪存阵列不是完好的，会引入若干个缺陷，这些缺陷可能会造成某些 block 不可用。块粒度(Block granularity)被用来映射工厂缺陷，因为这些缺陷可能会破坏块擦除功能。

3.3.1 Device 要求

如果一个块是有缺陷的，并且使用 8-bit 数据访问方式，则制造商应该通过设置有缺陷的块的第一个或者最后一个 page 中，缺陷区域的第一个 Byte 为 00h，来标记有缺陷的块，如图 35。如果一个块是有缺陷的并且使用 16-bit 数据访问方式，则通过设置缺陷块的第一个或最后一个 page 中，缺陷区域的第一个 word 为 0000h，来标记有缺陷的块。

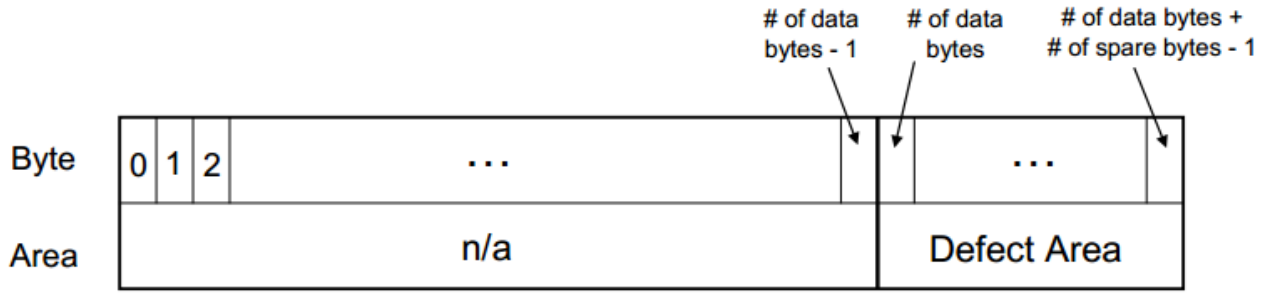


Figure 35　　Area marked in factory defect mapping

3.3.2　Host 要求

Host 不能读或者编程标记为缺陷的块，任何这样的尝试会产生不确定的结果。

图 36 大概描述了遍历工厂映射缺陷的算法。该算法应由 host 执行，以便在对象上执行任何擦除或者编程操作之前，创建一个原始的 bad block table。在一个没有缺陷的块中，所有 page 的初始状态是 FFh(或 FFFFh-16bit 访问)，尽管某些可以通过 ECC 更正的位错误会呈现出来。一个有缺陷的块会按 3.3.1 的方法被标记出来。Host 应该执行擦出或编程操作前，检查每个块中第一个和最后一个 page 的缺陷区域中的第一个 byte/word 的值，来确认块是好的。

注意：在一个 NAND device 的使用期内，缺陷块的缺陷区域可能会发生读干扰，造成值的变化。制造商缺陷标记可能会在 device 使用期内改变值，并期望 host 读取该值来在初次使用时创建 bad block table。

```
for (i=0; i<NumLUNs; i++)
{
    for (j=0; j<BlocksPerLUN; j++)
    {
        Defective=FALSE;

        ReadPage(lun=i; block=j; page=0; DestBuff=Buff);
        if (Buff[PageSize] == 00h)  // Value checked for is 0000h for 16-bit access
            Defective=TRUE;

        ReadPage(lun=i; block=j; page=PagesPerBlock-1; DestBuff=Buff);
        if (Buff[PageSize] == 00h)  // Value checked for is 0000h for 16-bit access
            Defective=TRUE;

        if (Defective)
            MarkBlockDefective(lun=i; block=j);
    }
}
```

Figure 36　　Factory defect scanning algorithm

3.4　扩展的 ECC 信息报告(ECC Information Reporting)

Device 可以在扩展参数页(extended parameter page)中报告扩展的 ECC 信息。要求的 ECC 可修正性与其他 device 参数紧密相关，如有效块(无缺陷的)的数量，以及支持的编程/擦除周期的次数。扩展的 ECC 信息允许 device 确定多种有效方法来使用 device。

表 22 定义了扩展 ECC 信息块。

Byte	Definition
0	Number of bits ECC correctability
1	Codeword size
2-3	Bad blocks maximum per LUN
4-5	Block endurance
6-7	Reserved

Table 22 Extended ECC Information Block Definition

每个字段的定义在接下来章节描述。

3.4.1 Byte0：ECC 可修正的 bit 数(Number of bits ECC correctability)

该字段表示每码字(codeword)中 host 可以修正的 bit 数。码字的 size 在 byte1 中报告。Host 确定了错误修正的数量，对象应达到由 bytes4-5 确定的块耐久性(Block endurance)。当确定的错误修正数量被 host 使用并且接下来块耐久性被使用时，device 中有缺陷的块数量不能超过由 byte 2-3 确定的 bad block 的最大数量。Page 中所有被使用的 byte 都应该被 host 控制器保护，包括 spare byte(如果 ECC 报告的 byte0 值大于 0)。

当该值(byte0)被清除为 0 时，如果 ECC 信息块是有效的(码字 size 非 0)，则对象应返回有效数据。

3.4.2 Byte1：码字 size(Codeword size)

由 byte0 确定的 ECC 可修正的 Bit 数是基于特定的 ECC 码字 size。ECC 码字 size 在本字段中定义，值为 2 的次幂。应该被报告的最小值是 512 bytes(值为 9)。

如果值为 0 被报告，那么该 ECC 信息块是非法的，不能被使用。

3.4.3 Byte2-3：每个 LUN 的最大坏块(bad block)数

该字段包含制造商的和 device 使用期内的每个 LUN 中可能有缺陷的块的最大数量。对最大值得评估是假设 Host 遵循在这个扩展 ECC 信息块中报告的的块耐久性要求和 ECC 要求。

3.4.4 Byte4-5：块耐久性(Block endurance)

该字段表示每个可寻址的 page/block 中编程/擦除周期的最大次数。该值假设 host 使用 byte0 报告的 ECC 可修正数。

块耐久性有两部分：一个数值和一个乘数，按以下公式的计算被报告：

$$\text{数值} \times 10^{\text{乘数}}$$

Byte4 包含数值，Byte5 包含乘数。例如，一个块耐久性为 75,000 周期，会被报告为一个数值 75 和一个乘数 3(75×10^3)。数值字段应该尽可能最小，比如，100,000 应该被报告为一个数值 1 和一个乘数 5(1×10^5)。

3.5 Discovery 和初始化(Discovery and Initialization)

3.5.1 没有 CE_n 引脚 reduction 的 Discovery

本章节讲述不使用 CE_n 引脚减少技术(2.20 中讲述)的 CE_n discovery。如果使用 CE_n 引脚 reduction，则初始化序列应遵循 3.5.2 的描述。

一个封装上可能有多个 chip enable(CE_n)信号，每个都是可独立寻址对象的。为了决定已连接的对象，每个不同的 CE_n 信号都应遵循本章节描述的步骤。CE_n 信号在 device 上应按顺序使用；CE0_n 始终被连接，CE_n 信号应该按数字递增的顺序连接。Host 应该尝试列举连接到所有 host CE_n 信号的对象。

支持独立双数据总线的封装，其 discovery 过程包含额外的步骤来决定对象连接到哪个数据总线上。ONFI 中有双数据总线的封装有：8-bit 数据访问的 LGA，100-ball BGA 以及 152-ball BGA 封装。8-bit 数据访问的 BGA-316 和 BGA-272 是 ONFI 中具有四数据总线的封装。

3.5.1.1 单数据总线 Discovery

要测试的 CE_n 是第一个被 host 拉低的，以使能对象(如果被连接)，在此期间，所有其他的 CE_n 信号都被拉高。随后 host 应该向对象发送 Reset(FFh)命令。Reset 之后，host 应向对象发送一个读 ID 命令。如果 ONFI 签名被地址为 20h 的读 ID 命令返回，则相应的对象已被连接。如果 ONFI 签名没有被返回，或者过程中任何步骤发生错误或 timeout，则 CE_n 没有被连接，使用该 CE_n 信号的其他操作不会被完成。

3.5.1.2 双/四数据总线 Discovery

要测试的 CE_n 第一个被 host 拉低，以使能对象(如果被连接)，在此期间其他 CE_n 信号都被拉高。随后 host 向对象发送 Reset(FFh)命令。Reset 之后，host 应向对象发送一个地址为 20h 的读 ID 命令。如果 ONFI 签名被读 ID 命令返回，则相应的对象已被连接。

如果 ONFI 签名没有被返回(或过程中任何步骤发生错误或 timeout),那么第二个 8-bit 数据总线应该被选通。Host 应该使用第二个 8-bit 向对象发送 Reset(FFh)命令。Reset 之后，host 应使用第二个 8-bit 数据总线向对象发送地址为 20h 的读 ID 命令。如果 ONFI 签名被读 ID 命令返回，则相应的对象已被连接，并且该对象使用第二个 8-bit 数据总线。Discovering 之后，对象是用第二个 8-bit 数据总线，所有随后发送到该对象的命令，包括读参数 page(Read Parameter Page)都应使用第二根 8-bit 数据总线。如果 ONFI 签名没有在第二个 8-bit 数据总线上被返回，则以上描述的第二个 8-bit 数据总线的 discovery 过程应该在第三个和第四个 8-bit 数据总线上被重复。

如果经过以上步骤后，有效地 ONFI 签名仍然没有被发现，或者发生其他错误，那么 CE_n 没有被连接，并且所有使用该 CE_n 信号的操作都不会被完成。

3.5.2 具有 CE_n reduction 的 Discovery

在上电之后，host 可以并行发送 Reset(FFh)给被选中的 Host 对象上的所有 NAND 对象，或者 host 可以顺序发送 Reset(FFh)给连接到特定 Host 对象的每个 NAND 对象。以上方法的选择取决于 host 要求的最大电流。为了并行 reset 所有 NAND 对象，host 发送 Reset(FFh)作为第一个发送给 NAND device 的命令。为了顺序 reset NAND 对象，host 发送给被选中的 Host 对象上所有 NAND 对象的第一个命令是 Reas Status(70h)。

如果一个封装中有多个 NAND 对象，则这些 NAND 对象共享同一个 ENo 信号，这种情况下，host 不能交错发送用来指定 Volume 地址的 Set Feature 命令(意思就是要同时并行发送)。如果 Set Feature 命令没有被同时发送，则 host 应该等待前一个 NAND 对象的 Volume 指定操作完成，之后再发送下一个 Set Feature 命令，来指定封装内共享 ENo 信号的下一个 NAND 对象的 Volume 地址。

在发送完指定 Volume 地址的 Set Feature 命令后，host 不得向关联的 host 对象上的任何 NAND 对象发送其他命令(包括状态命令)，直到经过 tFEAT 时间之后，以便确保合适的 NAND 对象响应下一个命令，允许合适的 ENo/ENi 信号电平被反射(to be reflected)。

当使用 CE_n 引脚 reduction 机制时初始化序列如下：

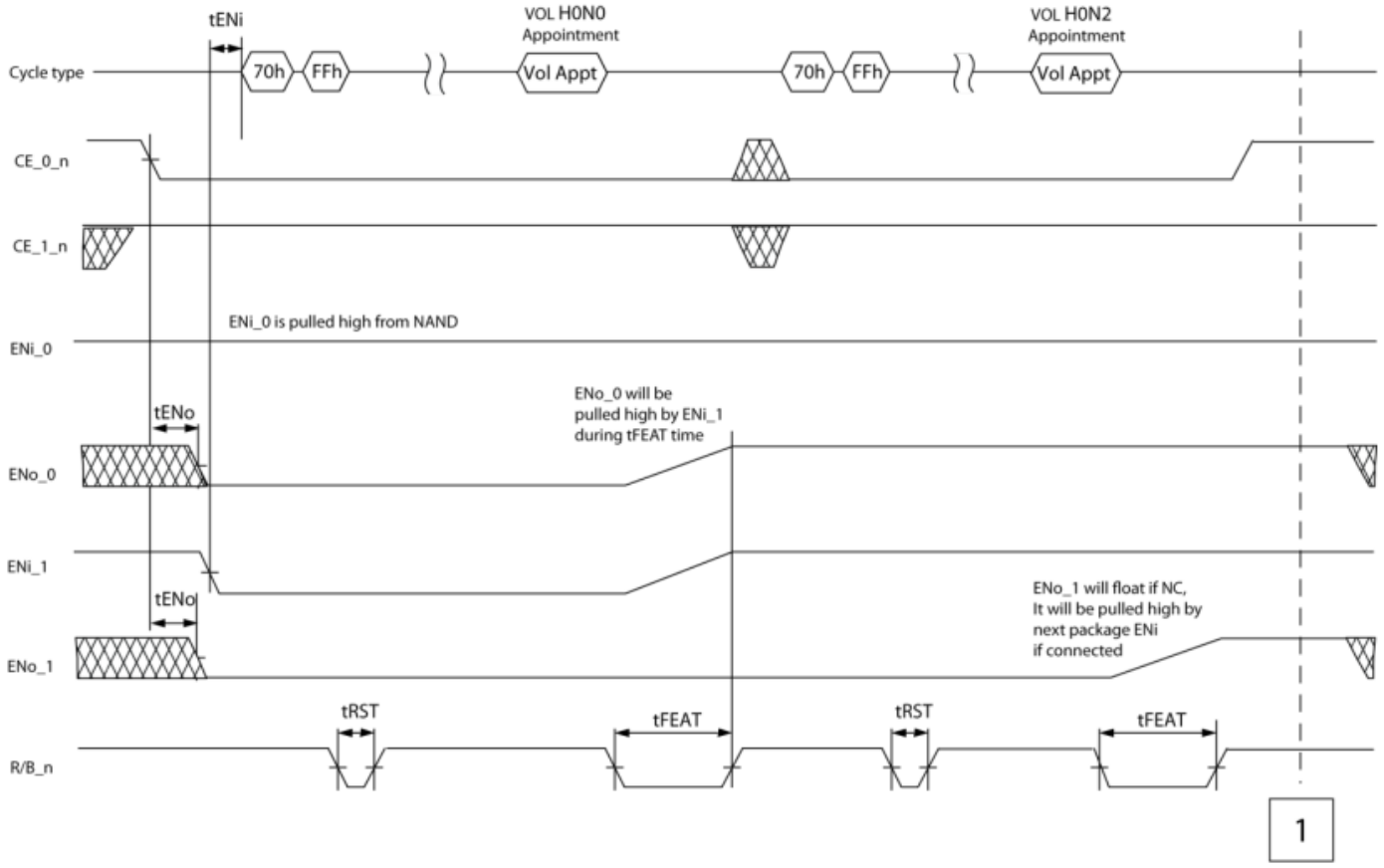
1. 开始向 NAND device 供电。
2. CE_n(Host 对象)被拉低。
3. 如果要并行 reset 所有的 NAND 对象，则 host 发送 Reset(FFh)命令。该命令被连接到 CE_n(Host 对象)的所有 NAND 对象接受。
4. 如果要顺序 reset 每个 NAND 对象，则：
   a. host 发送读状态(Read Status-70h)命令。在任何其他命令之前发送读状态(70h)命令表示顺序 Reset(FFh)每个 NAND 对象。
   b. host 发送 Reset(FFh)命令。当 ENi 信号为高时，该命令只 reset 与 CE_n(Host 对象)连接的 NAND 对象。
5. Host 发送读状态(Read Status-70h)命令，之后等待，直到 SR[6]被设为 1。
6. Host 配置 NAND 对象。host 发送读 ID(Read ID),读参数 page(Read Parameter Page),或其他需要的命令，来配置 NAND 对象。
7. 发送 Set Feature 命令，该命令具有 Volume 配置的特征地址(Feature Address),用来指定 ENi 信号为高的 NAND 对象的 Volume 地址。在所有 NAND 对象中,每个被指定的 Volume 地址都应该是唯一的。Set Feature 命令结束后，ENo 信号被拉高，Volume 被取消选择，直到发送下个 Volume Select 命令。Host 不能向关联的 host 对象上的 NAND 对象发送其他命令，直到经过 tFEAT 时间之后。
8. 对于连接到 Host 对象的每个 NAND 对象，在顺序初始化序列中，会重复步骤 4-7，而在并行初始化序列中，会重复步骤 5-7。
9. 当没有再发现其他 NAND 对象被连接到 Host 对象上时，会为下一个 Host 对象重复步骤 2-8(例如，host CE_n 信号)。
10. 为了完成初始化过程，在 CE_n 信号从高变为低后，需要发送一个 Volume Select 命令，来选择下一个要执行命令的 Volume。

Volume 地址被指定之后,host 可能要完成另一个额外的初始化任务(例如,为 NV-DDR2 或 NV-DDR3 配置 on-die

termination)，之后开始常规操作。

在步骤 2-7 中，host 的 CE_n 信号应该保持为低。如果在步骤 2-7 中被拉低的 host CE_n 信号，在步骤 7 之后，初始化过程完成之前的任何时间被拉高，则应该使用 SDR timing mode0 的 tCS(如，CE_n setup 时间)。

图 37 展示了一个基于图 29 拓扑的顺序 Reset 初始化的时序图。



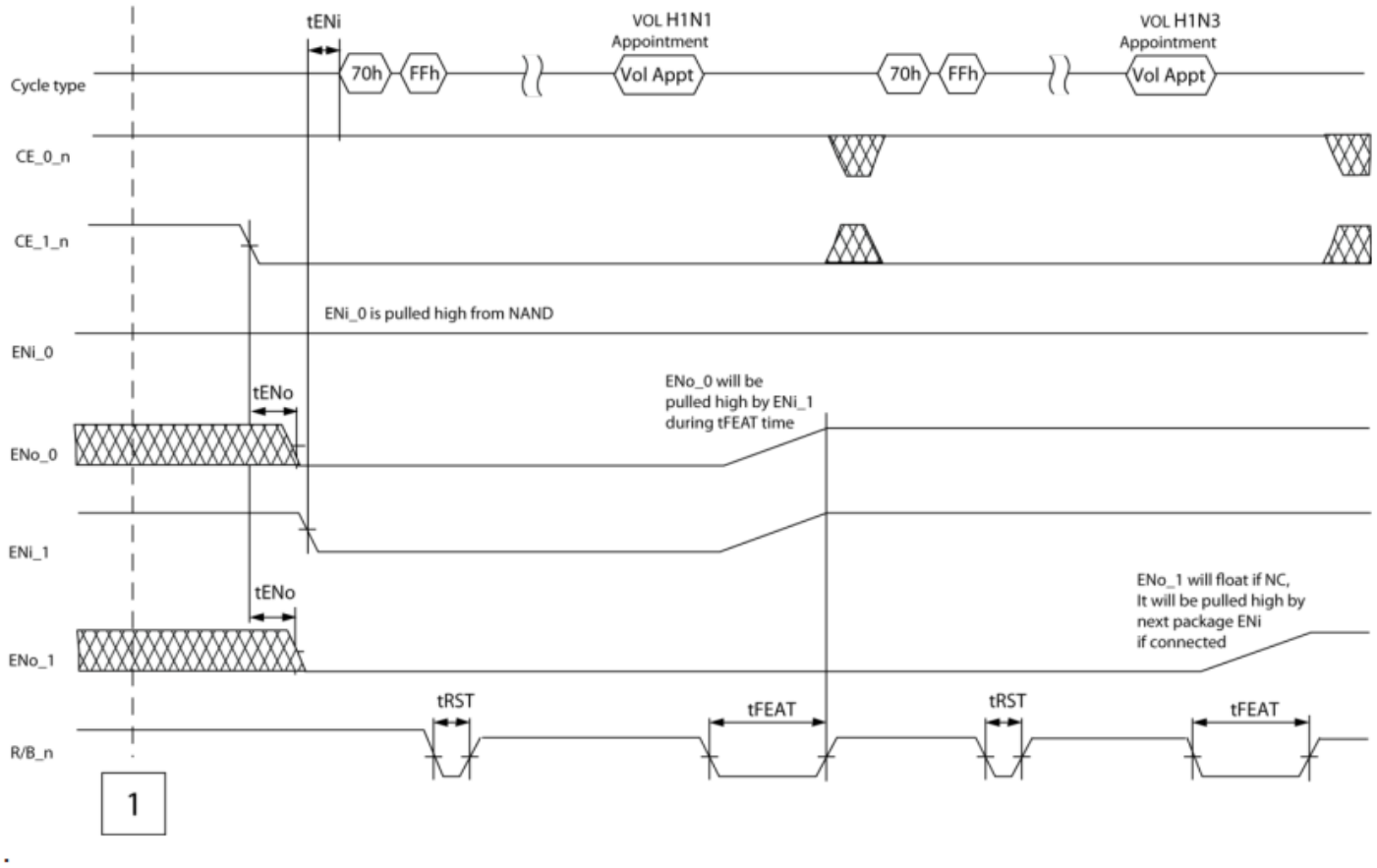


Figure 37 Example of Sequential Reset Initialization of Figure 29 Topology

3.5.3 对象初始化

为了初始化一个已被 discovered 的对象，以下步骤应该被执行。每个被连接的 CE_n 信号都应该执行初始化过程，包括每一个对象的读参数页(Read Parameter Page-ECh)命令。每个 chip enable 对应一个唯一的对象。

Host 应发送 Read Parameter Page(ECh)命令。该命令返回代表 device 容量、特征以及操作参数的信息，当该信息被从 device 读取后，host 应该在对信息数据执行操作之前，检查其 CRC，以确保接收的数据是正确的、没有错误的。

如果读取的第一个参数页（parameter page）的 CRC 是无效的(参见 5.7.1.26)，host 应读取冗余的参数页备份(redundant parameter page copies)。Host 可以通过检查第一个 4 bytes 是否包含至少 2 bytes 的参数页签名，来决定是否采用(present)这个冗余的参数页。如果参数页的签名被采用(present)，接下来 host 应读取这个完整的冗余参数页，之后 host 应检查这个冗余参数页的 CRC。如果 CRC 是正确的，则 host 可以采用这个冗余参数页的内容；如果 CRC 是错误的，则 host 应按以上相同的步骤，尝试读取下一个冗余的参数页。

Host 应该持续读取冗余参数页，直到 host 能够正确地获得参数页的内容。对象返回的所有参数页可能都有无效的 CRC 值；但是，bit-wise majority 或其他 ECC 技术可以用来修复参数页的内容。Host 可以使用 bit-wise majority 或其他 ECC 技术从参数页备份中修复参数页的内容。如果 host 判定某个参数页的签名不被采用，则应读取所有参数页。

读 ID(Read ID)和 Read Parameter Page 命令只适用数据总线的低 8 位。Host 在通过参数页判定 device 支持 16-bit 数据总线宽度前，不能发送在 x16 device 上使用一个字宽度的命令。

在成功获得参数页之后，host 具备了与该对象成功通信所需的所有信息。如果 host 没有预先映射该对象的缺陷块的信息，则接下来，host 应映射出对象中的所有缺陷块。之后，host 就可以开始使用对象了，包括擦除和编程等操作。

4. 数据接口和时序

4.1 数据接口类型概览

ONFI 支持 4 中不同的数据接口类型：SDR，NV-DDR，NV-DDR2 和 NV-DDR3。SDR 是传统的 NAND 接口，使用 RE_n 锁存读数据，WE_n 锁存写数据，没有时钟。NV-DDR 是双数据率(Double Data Rate-DDR)接口，含有用来锁存命令和地址的时钟，和一个用来锁存数据的数据选通信号。NV-DDR2 是双数据率接口，含有额外的扩展速度(scaling speed)的功能,像 on-die termination 以及差分信号。NV-DDR3 接口包含所有 NV-DDR2 的特性,但操作在 VccQ=1.2V。数据接口包含的特性如表 23 所示。

Feature	Data Interface			
	SDR	NV-DDR	NV-DDR2	NV-DDR3
Protocol	Single data rate (SDR)	Double data rate (DDR)	Double data rate (DDR)	Double data rate (DDR)
Maximum Speed		200 MT/s	800 MT/s	800 MT/s
CE_n Pin Reduction support	Yes	Yes	Yes	Yes
Volume Addressing support	Yes	Yes	Yes	Yes
On-die termination support	No	No	Yes	Yes
Differential signaling	No	No	Yes, optional for DQS and/or RE_n	Yes, optional for DQS and/or RE_n
VccQ support	3.3 V or 1.8 V	3.3 V or 1.8 V	1.8 V	1.2 V
External Vpp support	Yes	Yes	Yes	Yes
External VREFQ support	No	No	Yes	Yes
ZQ Calibration	No	No	Yes, optional	Yes, optional
Previous name in older ONFI specifications	Asynchronous	Source Synchronous	n/a	n/a

Table 23 Data Interface Comparison

如果上电时 VccQ=1.8V 或 3.3V，则 device 应该操作在 SDR 接口 timing mode 0。如果 Host 在参数 page 中判定 NV-DDR 和 NV-DDR2 都被支持，则 host 可以通过 Feature Address 为 01h 的 Set Feature 命令，来选择其中一个接口以及支持的 timing mode。参见 5.30.1。

如果上电时 VccQ=1.2V，则 device 应操作在 NV-DDR3 接口 timing mode 0。如果 host 在参数 page 中判明了支持的 NV-DDR3 timing mode，则 host 可以通过将 CE_n 转为高来使能支持的 timing mode，并将接口速度改变为期望的 timing mode。当 host 将 CE_n 拉低后，新的 timing mode 会生效。参见 5.30.1。

NV-DDR，NV-DDR2 和 NV-DDR3 接口使用 DDR 协议。因此，传输的数据总是偶数 byte 的。当使用 DDR 协议时，列地址的最低位应该始终为 0。如果使用 DDR 协议时，列地址的最低位被设为了 1，则结果是不确定的。

4.2 信号功能分配

某些信号在不同的接口中其功能也是不同的；这些不同将在本章节讲述。

对于 NV-DDR，NV-DDR2 或 NV-DDR3 接口，与 SDR 接口相比，其共同的变化有：

- I/O 总线重命名为 DQ 总线
- 新加了一个名为 DQS(DQ strobe)的 DQ 数据总线选通信号。DQS 是双向信号，用于数据传输。DQS 不能用于命令或地址周期。对于从 host 到 device 的数据传输(写)，DQS 的锁存沿对齐到有效数据窗口的中间；对于从 device 到 host 的数据传输(读)，DQS 的锁存沿对齐到 DQ 总线的转换沿。当操作在 SDR 接口时，DQS 应该被 host 拉高，被 device 忽略。

对于 NV-DDR 接口，与 SDR 接口相比，主要变化有：

- WE_n 变成时钟信号(CLK)。CLK 应该被使能并且具有有效的时钟周期，不论命令周期，地址周期和数据周期什么时候发生。在 CE_n 为低期间，CLK 应保持相同的频率。参见 2.8.1。
- RE_n 变成写/读方向信号(W/R_n)。该信号表示谁拥有 DQ 总线和 DQS 信号。Host 应该仅在 ALE 和 CLE 被锁存到 0 时才能转换 W/R_n。参见 4.19.2.6 W/R_n 要求。

对于 NV-DDR2 和 NV-DDR3 接口，与 SDR 接口相比，主要变化有：

- RE_n 可作为单端信号(single-ended)或者作为一个互补信号对(RE_t,RE_c)使用
- 增加了名为 DQS(DQ strobe)的 DQ 数据总线选通信号。DQS 可作为单端信号或者作为一个互补信号对(DQS_t,DQS_c)使用。

表 24 描述了基于所选接口类型的信号功能。

Symbol			Type	Description
SDR	NV-DDR	NV-DDR2 / NV-DDR3		
ALE	ALE	ALE	Input	Address latch enable
CE_n	CE_n	CE_n	Input	Chip enable
CLE	CLE	CLE	Input	Command latch enable
I/O[7:0]	DQ[7:0]	DQ[7:0]	I/O	Data inputs/outputs
—	DQS	DQS / DQS_t	I/O	Data strobe
—	—	DQS_c	I/O	Data strobe complement
RE_n	W/R_n	RE_n / RE_t	Input	Read enable / (Write / Read_n direction)
—	—	RE_c	Input	Read enable complement
WE_n	CLK	WE_n	Input	Write enable / Clock
WP_n	WP_n	WP_n	Input	Write protect
R/B_n	R/B_n	R/B_n	Output	Ready / Busy_n
—	—	ZQ	NA	ZQ calibration

Table 24 Signal Assignment based on Data Interface Type

4.3 总线状态

在所有数据接口中，ALE 和 CLE 被用来判别当前总线的状态。

4.3.1 SDR

表 25 描述了 SDR 的总线状态。注意，在 SDR 中，值为 11b 的 ALE/CLE 是没有定义的。

CE_n	ALE	CLE	WE_n	RE_n	SDR Bus State
1	X	X	X	X	Standby
0	0	0	1	1	Idle
0	0	1	0	1	Command cycle
0	1	0	0	1	Address cycle
0	0	0	0	1	Data input cycle
0	0	0	1	0	Data output cycle
0	1	1	X	X	Undefined

Table 25 Asynchronous Bus State

4.3.2 NV-DDR

表 26 描述了 NV-DDR 操作的总线状态。值为 11b 的 ALE/CLE 用于数据传输。总线状态在 CLK 的上升沿开始，持续一个完整的 CLK 周期。因此，对于数据周期，每个总线状态会有两个数据输入周期或两个数据输出周期。Idle 总线状态用于在一个命令周期、地址周期或一个数据流(stream of data)之后中止 DQ 总线上的操作。

CE_n 的值仅可在总线状态为 idle(例如，ALE 和 CLE 都被清为 0)且该时钟周期内没有数据传输时被改变。

CE_n	ALE	CLE	W/R_n	CLK	NV-DDR Bus State
1	X	X	X	X	Standby
0	0	0	1	Rising edge to rising edge	Idle[1]
0	0	0	0	Rising edge to rising edge	Bus Driving[1]
0	0	1	1	Rising edge to rising edge	Command cycle
0	1	0	1	Rising edge to rising edge	Address cycle
0	1	1	1	Rising edge to rising edge	Data input cycle[2]
0	1	1	0	Rising edge to rising edge	Data output cycle[2]
0	0	1	0	Rising edge to rising edge	Reserved
0	1	0	0	Rising edge to rising edge	Reserved

NOTE:
1. When W/R_n is cleared to '0', the device is driving the DQ bus and DQS signal. When W/R_n is set to '1' then the DQ and DQS signals are not driven by the device.
2. There are two data input/output cycles from the rising edge of CLK to the next rising edge of CLK.

Table 26 NV-DDR Bus State

4.3.3 NV-DDR2 和 NV-DDR3

表 27 描述了 NV-DDR2 和 NV-DDR3 操作的总线状态。

CE_n	ALE	CLE	RE_n (RE_t)	DQS (DQS_t)	WE_n	Data Input or Output[1]	Measurement Point	Bus State
1	X	X	X	X	X	X	X	Standby
0	0	0	1	1	1	None	X	Idle
0	0	1	1	–	–	None	WE_n rising edge to rising edge	Command cycle
0	1	0	1	–	–	None	WE_n rising edge to rising edge	Address cycle
0	0	0	1	–	1	Input	DQS rising edge to rising edge	Data input cycle[2,3]
0	0	0	–	–	1	Output	RE_n rising edge to rising edge	Data output cycle[2,3,4]

NOTE:
1. The current state of the device is data input, data output, or neither based on the commands issued.
2. There are two data input/output cycles from the rising edge of DQS/RE_n to the next rising edge of DQS/RE_n.
3. ODT may be enabled as part of the data input and data output cycles.
4. At the beginning of a data output burst, DQS shall be held high for tDQSRH after RE_n transitions low to begin data output.

Table 27 NV-DDR2 and NV-DDR3 Bus State

4.3.4 暂停数据输入/输出

在任何接口中，Host 可以通过进入 idle 状态来暂停数据输入或数据输出。

在 SDR 接口中，通过保持 WE_n 或 RE_n 为 1 来分别暂停数据输入或数据输出。

在 NV-DDR 接口中，通过将 ALE 和 CLE 都清为 0 来暂停数据输入或数据输出。经过合适的 tCAD 时间后，host 可以通过将 ALE 和 CLE 都设为 1 来继续数据传输。

在 NV-DDR2 或 NV-DDR3 接口中，通过将总线置为 idle 状态来暂停数据输入或数据输出。暂停数据输出也可以通过暂停 RE_n(RE_t/RE_c)并将该信号保持为高或低直到数据突发(data burst)被恢复来实现。暂停数据输入也可以通过暂停 DQS(DQS_t/DQS_c)并保持该信号为高或低直到数据突发被恢复来实现。WE_n 在数据输入或数据输出突发暂停期间应该被置为高。ODT(如果使能)在整个暂停期间保持为 ON，当数据突发从暂停状态重新开始时，不会重新发送 warmup（如果使能）周期。当重新开始时，如果要 disable ODT，或重新发送 warmup 周期，则应请求 host 退出数据突发。当退出并重新开始数据突发时，warmup 周期被请求，参考 4.15 重新发送 warmup 周期。如果 host 希望结束数据突发，则在退出数据突发后，新的命令可以被发送。

4.4 NV-DDR/NV-DDR2/NV-DDR3 和重复字节(Repeat Bytes)

NV-DDR,NV-DDR2 以及 NV-DDR3 接口使用 DDR 数据传输技术来达到一个比较高的数据传输率。然而，一些不常用的配置和设置命令并不要求高数据传输率。另外，这些命令也不是通过用于数据传输的流水线(pipeline)执行的。

对于这些命令，在开发中为了避免不必要的复杂和要求，使用单一的数据率来进行数据传输。特别是，相同的数据 byte 被重复两次，而且应该符合 NV-DDR、NV-DDR2 以及 NV-DDR3 接口的时序要求。这种情况下，数据模式(data pattern)是 D_0 D_0 D_1 D_1 D_2 D_2等。接收端(host 或 device)应只锁存每个数据 byte 的一个 copy。该命令期间数据输入或数据输出不能被暂停。接收端在开始内部操作之前，不需要等待重复的数据 byte。

在 NV-DDR、NV-DDR2 和 NV-DDR3 接口中，重复每个数据 byte 两次的命令有：Set Feature,Read ID,Get Feature,Read Status,Read Status Enhanced,以及 ODT Configure。SDR 命令可以使用 device 支持的最高数据传输率。

4.5 数据接口/时序模式(Timing mode)转换

以下是支持的数据接口间的转换：

- SDR 到 NV-DDR
- SDR 到 NV-DDR2
- NV-DDR 到 SDR
- NV-DDR2 到 SDR

不支持从 NV-DDR 直接转到 NV-DDR2(反之亦然)。这种情况，host 应先转到 SDR 接口，然后再选择期望的 NV-DDR 或 NV-DDR2 接口。

在任何接口中，时序模式(timing mode)之间的转换都是支持的。

为了转变接口为 NV-DDR 或 NV-DDR2，或者转变任何时序模式，会使用带有 Timing Mode feature 的 Set Feature 命令。Set Feature 命令(EFh),Feature Address，及四个参数使用之前选定的接口中，由之前选中的时序模式来执行。当发送 Set Feature 命令时，host 应在整个命令执行期间(包括参数)驱动 DQS 信号为高(如果接口支持，则发送 Set Feature 命令)。在第四个参数 P4 被执行后，直到经过 tITC 时间之前，host 不能向 device 发送任何命令。在发送 Set Feature 命令后，CE_n 信号转变为高之前，host 应在一个 idle 状态周期内保持信号，并且 DQS 应被设为 1。另外，当使用 NV-DDR 接口时，CLK 只能在 CE_n 为高时改变。

不支持从 NV-DDR3 转到其它接口(SDR,NV-DDR 或 NV-DDR2)。如果 VccQ=1.2V，则只支持 NV-DDR3 接口。为了改变 NV-DDR3 的时序模式，host 应将 CE_n 变为高，然后将接口速度转换为期望的时序模式。当 host 将 CE_n 拉低后，新的时序模式生效。当转变 NV-DDR3 的时序模式时，CE_n 变为高之前，host 应在一个 idle 总线状态中保持信号，并且 DQS 应被设为 1。

向 device 发送任何新命令之前，host 应将 CE_n 转为高。当 host 将 CE_n 拉低后，新的接口或时序模式生效。

4.5.1 NV-DDR 或 NV-DDR2 转到 SDR

为了从 NV-DDR 或 NV-DDR2 转变为 SDR，host 应使用 SDR 时序模式 0 下的 Reset 命令(FFh)。出于任何时序模式的一个 device 被要求能够识别从 SDR 时序模式 0 下发送的 Reset 命令(FFh)。Reset 发送后，host 在持续 tITC 时间前不能向 device 发送任何命令。注意，经过 tITC 时间之后，在 Reset 结束前，host 只能发送状态命令。在发送 Reset(FFh)之后，CE_n 转为高之前，host 应在一个 idle 周期状态中保持信号，并且 DQS 应被设为 1。

在 CE_n 已经被拉高，然后再次变低之后，host 应发送 Set Feature 命令来选择合适的 SDR 时序模式。

4.5.2 NV-DDR2 的推荐

在选择 NV-DDR2 接口之前，有一些 NV-DDR2 的推荐设置应该被配置，特别是：

- 应使用 Set Feature 命令来配置 NV-DDR2 的 Configuration feature。
- 如果 on-die termination 被使用在一个更高级的拓扑中，则应发送合适的 ODT Configure 命令。

这些操作应该在选择 NV-DDR2 接口前完成。如果 NV-DDR2 接口已经被选中后，这些设置被改变了，则 host 应保证用一种合适的方法来配置合适的设置，以避免信号完整性的问题。

4.5.3 NV-DDR3 的推荐

在为 NV-DDR3 接口使能期望的时序模式之前，有些为 NV-DDR3 推荐的设置应该在时序模式 0 下被配置，特别是：

- 应使用 Set Feature 命令来配置 NV-DDR3 的 Configuration feature。
- 如果 on-die termination 被使用在一个更高级的拓扑中，则应发送合适的 ODT Configure 命令。

这些操作应该在使能除时序模式 0 以外的其它 NV-DDR3 时序模式之前被完成。

4.6 测试条件

略

4.7 ZQ 校准(ZQ Calibration)

ZQ 校准是可选的，但对超过 400MT/s 速度的 NV-DDR3 接口是推荐的。ZQ 校准对 NV-DDR2 接口是可选的。

对于 ZQCL(ZQ long calibration)，ZQ 校准是通过发送 F9h 命令来执行的，而对于 ZQCS(ZQ short calibration)，ZQ 校准是通过发送 D9h 命令执行的。ZQ 校准用来校正 NAND Ron&ODT 的值。在初始化阶段需要比较长的时间来校正输出驱动和 on-die termination 电路，相对地，执行定期校正的时间会较短。

ZQCL 用来在上电初始化之后执行初始校验。ZQCL 的命令可以由控制器在任何时候发送，取决于系统环境。ZQCL 触发 NAND 内部的校验引擎，一旦校验完成，经过校验的值从校验引擎传送到 NAND IO，用来更新输出驱动和 on-die termination 的值。

ZQCL 允许一个 tZQCL 的时间周期来执行完整的校验并传输校验值。

ZQCS 命令用于执行定期校验以应对电压和温度的变化。由时间参数 tZQCS 定义的一个较短的时间窗口用来执行校验并传输校验值。一个 ZQCS 命令能够在 tZQCS 时间内有效修正一个最小为 1.5%(ZQ Correction)的 R_{ON} 和 Rtt 阻抗错误，该过程可针对所有 speed bins，假设最大敏感度为表 54 中——输出驱动敏感度定义，以及表 70 中——ODT 电压和温度敏感度。ZQCS 命令之间合适的时间间隔可以由这些表以及其它应用-特定参数确定。给定了 NAND 在应用中遵从的温度(Tdriftrate)和电压(Vdriftrate)漂移率，则可以构建一个计算 ZQCS 命令间隔时间的方法。可以通过以下公式来定义时间间隔：

$$ZQCorrection/[(TSens\ X\ Tdriftrate) + (VSens\ X\ Vdriftrate)]$$

TSens=max(dRTTdT, dRONdTM)，VSens=max(dRTTdV, dRONdVM)定义了 NAND 的温度和电压敏感度。

例如，如果 TSens = 0.5%/oC,VSens = 0.2%/mV,Tdriftrate = 1oC/sec,Vdriftrate = 15mV/sec，那么 ZQCS 命令的时间间隔计算如下：

$$1.5/[(0.5\ X\ 1) + (0.2\ X\ 15)] = 0.429 = 429ms$$

在 tZQCL 或 tZQCS 期间，控制器不能在 NAND 通道上(例如，数据总线)执行读状态(read status)等操作。NAND 通道上的 quiet time 可以精确校正输出驱动和 on-die termination 的值。对于多通道的封装，在 ZQ 校准期间所有通道上都不能有数据传输，即使 device 不和执行 ZQ 校准的 LUN 共享同一个通道。一旦 NAND 校准完成，NAND 应及时关闭 ZQ 电流消耗的路径以减少功耗。NAND 阵列的操作可能不会发生在正执行 ZQCS 或 ZQCL 的 device 上，也可能会发生在任何与正在执行 ZQCS 或 ZQCL 的 device 共用 ZQ 电阻的 device 上。在校正过程中，所有连接到 DQ 总线的 device 都应该是高阻抗状态,信号 R/B#会被 device 拉低,但是如果其他 device 正在驱动一个共用的 R/B#为低，那么 host 应该在发送任何命令到数据总线之前等待最大 tWB+tZQCS 的时间(见图 113)。

如果在 ZQ 校正期间发送了一个 RESET 命令，则 NAND device 的输出驱动强度和 ODT 的状态是不确定的，host 应重新执行校正操作。如果在 ZQCL 操作期间发送了一个 RESET 命令(FFh,Fah,FCh)，则 RESET 操作会被执行，NAND device 的输出驱动强度和 ODT 的值会返回出厂设置(就像执行 ZQ 校正前一样)。如果在 ZQCS 期间发送了一个 RESET 命令(FFh,Fah,FCh),则 RESET 操作会被执行，NAND device 的输出驱动强度和 ODT 的值会恢复供应商指定设置(Vendor specific settings)。当 ZQCL 或 ZQCS 被 RESET 命令中断后，reset 时间会小于 10us(tRST < 10us)。

在 NAND device 间共用 ZQ 电阻的系统中，控制器不得允许 device 间的 tZQCL 或 tZQCS 有重叠。

4.7.1 ZQ 外部电阻值，误差以及容性负载

略

4.8 I/O 驱动强度

略

4.9 输出压摆率(电压转换速率-Slew Rate)

略

4.10 电容

略

4.11 略

4.12 略

4.13 略

4.14 差分信号(NV-DDR2/NV-DDR3)

一种用于高速操作的方法，使使用 RE_n 和 DQS 的差分信号。一个互补的 RE_n 和互补的 DQS 信号可以用来构建差分信号对(RE_t/RE_c 和 DQS_t/DQS_c)。当使用差分信号时,RE_n 作为 RE_t,DQS 作为 DQS_t,例如信号的“真”。差分信号可用于通过增强抗扰度(enhanced noise immunity)来提高信号完整性。只有 NV-DDR2 和 NV-DDR3 接口支持差分信号。

一个 device 可能支持差分 RE_n 和/或差分 DQS 信号，是否支持该差分信号报告在参数 page 中。默认情况下，差分信号是被 disable 的。Host 可以通过 NV-DDR2/NV-DDR3 的配置特性(Configuration feature)来配置 device 以使用差分信号，参见 5.30.2。互补的 RE_n(如 RE_c)和互补的 DQS(如 DQS_c)信号是单独被配置/使能的。

当选定的接口是 NV-DDR2 或 NV-DDR3 时，差分信号有效；在 NV-DDR2/NV-DDR3 的配置特征(Configuration feature)中使能差分信号。在使能 NV-DDR2 接口前，推荐使用 SDR 接口来配置 NV-DDR2/NV-DDR3 的配置特征(Configuration feature)。NV-DDR2/NV-DDR3 Configuration feature 中的差分信号设置状态被改变后，host 应在发送接下来的命令之前，将 CE_n 转为高，以避免任何信号完整性的问题。

如果执行了 RESET(FFh)操作，则差分信号会被 disable。同步 Reset(FCh)和 Reset LUN(FAh)对差分信号是无效的。

差分 AC 输入参数定义在表 63 中。差分 AC 输出参数定义在表 64 中。

Parameter	Symbol	Min	Max	Unit
AC differential input cross-point voltage relative to VccQ / 2	VIX(AC)	NV-DDR2: 0.50 x VccQ - 175	NV-DDR2: 0.50 x VccQ + 175	mV
		NV-DDR3: 0.50 x VccQ – 120	NV-DDR3: 0.50 x VccQ + 120	

Table 63 Differential AC Input Parameters

Parameter	Symbol	Min	Max	Unit
AC differential output cross-point voltage	VOX(AC)	Without ZQ cal: 0.50 x VccQ - 200	Without ZQ cal: 0.50 x VccQ + 200	mV
		With ZQ cal: 0.50 x VccQ - 150	With ZQ cal: 0.50 x VccQ + 150	

Table 64 Differential AC Output Parameters

4.15 Warmup 周期(NV-DDR2/NV-DDR3)

为了支持高速操作，在数据输出和数据输入时引入了 warmup 周期。Warmup 周期仅在 N-DDR2 和 NV-DDR3 接口中支持。

数据输出的 warmup 周期在一个数据输出 burst 开始时提供了 RE_n 以及对应的 DQS 的额外(extra)转变。这个额外的 RE_n/DQS 转变期间不会有任何数据传输。额外周期的数目通过 NV-DDR2/NV-DDR3 的配置特征地址(Configuration feature address)来设置，参见 5.30.2。周期数的定义包含一个完整数据输出周期(RE_n 和 DQS 的上升沿及下降沿)。

数据输入的 warmup 周期在一个数据输入 burst 开始提供了额外的 DQS 转变。这个额外的 DQS 转变期间不会有任何数据传输。额外周期的数目通过 NV-DDR2/NV-DDR3 的配置特征地址(Configuration feature address)来设置，参见 5.30.2。周期数的定义包含一个完整数据输入周期(DQS 的上升沿及下降沿)。

对数据输入和数据输出来说，warmup 周期都是可选的，如果使用了 warmup 周期，则不需要配置为相同值。warmup 周期适用于所有命令，包括 SDR 命令(参见 4.4)。当某种数据传输类型使能了 warmup 周期，则 warmup 周期应该在该类型中每个数据突发开始时开始。如果 host 暂停了数据传输，之后有重新恢复了传输，在此期间没有退出并重新进入数据突发，在这种情况下,host 不能再次发送 warmup 周期。退出和重新进入数据突发的操作通过拉高 ALE、CLE 或 CE_n 来执行(不使用 WE_n 来锁存)。如果暂停数据传输一段时间后又恢复传输时没有重新发送 warmup 周期，则 host 要小心避免信号完整性问题。

Warmup 周期在选中 NV-DDR2 或 NV-DDR3 接口后有效，并通过 NV-DDR2/NV-DDR3 的 Configuration feature 来使能。对于 NV-DDR2,推荐使用 SDR 接口来设置 NV-DDR2/NV-DDR3 Configuration feature。如果在 NV-DDR2 或 NV-DDR3 有效时使能了 warmup 周期，则在 Set Feature 命令完成后，warmup 周期可以用于接下来的所有命令。

图 47 展示了一个数据输出时 warmup 周期的例子，该例中 warmup 周期数为 2。如图所示，数据的第一个 byte 在 DQS 的第三个上升沿开始传向 host。

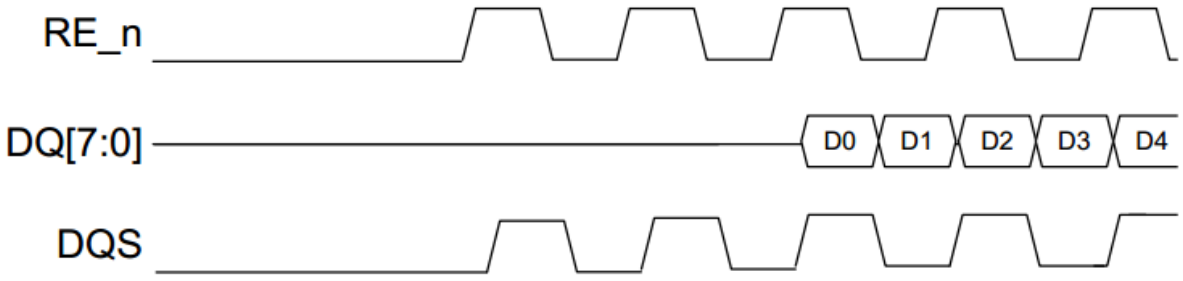


Figure 47 Warmup Cycles for Data Output

4.16 On-die Termination(NV-DDR2/NV-DDR3)

决定于系统拓扑的更高速度的要求需要 on-die termination(ODT)。本章节讲述 DQ[7:0],DQS_t，RE_t 以及 RE_c 信号上的 on-die termination 机制。On-die termination 是一种在典型的拓扑中用来满足更高速度需求的可选功能。如果需要在典型条件下优化功耗，则可能需要 disable on-die termination，而拓扑可能要运行在较低的速度下。只有 NV-DDR2 和 NV-DDR3 接口才能支持 on-die termination。

On-die termination 在初始化期间配置。Host 可能会将 on-die termination 配置为 self-termination，也可能利用矩阵 termination 将 on-die termination 配置成更灵活的组合。

…

…

以下内容略，参见英文原文档。

4.17 时序参数(Timing Parameters)

当要求的最小和最大时序不满足时，device 的行为是不确定的。

4.17.1 一般参数

本章节讲述用于所有接口类型的时序参数。

对于第一个 Read Parameter Page 命令的执行，在初始化完成前，应该使用一个值为 200ms 的 tR 和一个值为 500ns 的 tCCS。对于 page 的读操作，包括在初始化完成后执行的 Read Parameter Page 命令，应使用参数 page 中的 tR 和 tCCS 的值。

在 SDR、NV-DDR、NV-DDR2 以及 NV-DDR3 接口中，有三种最大 tRST。在编程和擦除操作中，对象允许一个更长的最大 reset 时间。最大值符合：

- 对象当前没有执行擦除或编程操作
- 对象正在执行编程操作
- 对象正在执行擦除操作

表 75 定义了 SDR、NV-DDR、NV-DDR2 以及 NV-DDR3 共用的时序参数。

Parameter	Description
tADL[2]	Address cycle to data loading time
tCCS[2]	Change Column setup time, value specified in parameter page
tCEH	CE_n high hold time
tCH	CE_n hold time
tCS	CE_n setup time
tDH	Data hold time
tDS	Data setup time
tFEAT[1]	Busy time for Set Features and Get Features
tITC[1]	Interface and Timing Mode Change time
tRR	Ready to data output cycle (data only)
tRST	Device reset time, measured from the falling edge of R/B_n to the rising edge of R/B_n.
tWB[3,4]	(WE_n high or CLK rising edge) to SR[6] low
tWHR[2]	Command, address, or data input cycle to data output cycle
tWW	WP_n transition to command cycle

NOTE:
1. Measured from the falling edge of SR[6] to the rising edge of SR[6].
2. tADL is used for Program operations. tWHR is used for Read ID, Read Status, and Read Status Enhanced commands. tCCS is used for commands that modify the column address and thus impact the data pipeline; these commands include Change Read Column and Change Write Column.
3. For Set Features when using NV-DDR2 or NV-DDR3, tWB starts on the rising edge of DQS for parameter P4.
4. Commands (including Read Status / Read Status Enhanced) shall not be issued until after tWB is complete.

Table 75 General Timing Parameters

表 76 定义了阵列时序参数(array timing parameters)。阵列时序参数值既可以从参数 page 返回(tR,tPROG,tBERS,和 tCCS)，也可以是表 77 定义的固定值。

Parameter	Description
tBERS[1]	Block erase time
tCCS	Change Column setup time
tPLEBSY[1]	Busy time for multi-plane erase operation
tPLPBSY[1]	Busy time for multi-plane program operation
tPLRBSY[1]	Busy time for multi-plane read operation
tPCBSY	Program cache busy time
tPROG[1]	Page program time
tR[1]	Page read time
tRCBSY[1]	Read cache busy time
NOTE: 1. Measured from the falling edge of SR[6] to the rising edge of SR[6].	

Table 76 Array Timing Parameter Descriptions

有几个“short”busy 时间与 cache 操作(tRCBSY,tPCBSY)以及多层操作(tPLEBSY，tPLPBSY，和 tPLRBSY)关联。这些 busy 时间的典型值和最大值列在表 77 中。

Parameter	Typical	Maximum
tPLEBSY	500 ns	tBERS
tPLPBSY	500 ns	tPROG
tPLRBSY	500 ns	tR
tPCBSY	3 µs	tPROG
tRCBSY	3 µs	tR
NOTE: 1. Typical times for tPCBSY and tRCBSY are the recommended interval at which the host should consider polling status. Device busy time may be longer than the typical value.		

Table 77 Cache and Multi-plane Short Busy Times

CE_n 引脚减少机制可能被用于任何接口中。如果 VccQ=3.3V 或 1.8V，则 CE_n 引脚 reduction 的配置应该使用 SDR 接口完成。如果 VccQ=1.2V,CE_n 引脚 reduction 的配置应该使用 NV-DDR3 接口完成。在封装间的 daisy chain 使用的 enumeration 信号时序以及 Volume 寻址的时序参数如表 78 所示。

Parameter	Description	Minimum	Maximum
tVDLY	Delay prior to issuing the next command after a new Volume is selected using the Volume Select command.	50 ns	-
tCEVDLY	Delay prior to bringing CE_n high after a new Volume is selected using the Volume Select command.	50 ns	-
tENi	ENi low until any issued command is ignored (ENo driving low from previous package in daisy chain)	-	15 ns
tENo	CE_n low until ENo low	-	50 ns

Table 78 CE_n Pin Reduction Enumeration and Volume Addressing Times

Parameter	Description	Maximum
tZQCL	Normal operation Long calibration time	1us
tZQCS	Normal operation Short calibration time	0.3us
NOTE: Increased tZQCL and tZQCS values beyond minimum specified value may result when greater than 8 LUNs share a ZQ resistor		

Table 79 ZQ Calibration Timing parameters

4.17.2 SDR

表 80 定义了 SDR 接口中使用的所有时序参数。表 83 和表 84 定义了时序模式 0，1，2，3，4 和 5 的要求。时序模式 0 应始终被支持，且在上电后 device 操作在该模式。对于更高的时序模式，只有在参数 page 中判定 device 支持该模式后 host 才能开始使用。

当host 以小于 30ns 的 tRC 运行时,应使用 EDO 数据输出周期时序(EDO data output cycle timings),如 4.19.1.5

所述。

Parameter	Description
tALH	ALE hold time
tALS	ALE setup time
tAR	ALE to RE_n delay
tCEA	CE_n access time
tCHZ[1]	CE_n high to output hi-Z
tCLH	CLE hold time
tCLR	CLE to RE_n delay
tCLS	CLE setup time
tCOH	CE_n high to output hold
tCR	CE_n to RE_n low
tCR2	CE_n to RE_n low after CE_n has been high for greater than 1us
tIR[1]	Output hi-Z to RE_n low
tRC	RE_n cycle time
tREA	RE_n access time
tREH	RE_n high hold time
tRHOH	RE_n high to output hold
tRHW	RE_n high to WE_n low
tRHZ[1]	RE_n high to output hi-Z
tRLOH	RE_n low to output hold
tRP	RE_n pulse width
tWC	WE_n cycle time
tWH	WE_n high hold time
tWP	WE_n pulse width
NOTE: 1. Refer to Appendix E for measurement technique.	

Table 80 SDR Timing Parameter Descriptions

4.17.3 NV-DDR

所有 NV-DDR 接口的时序参数都是以 CLK 的上升沿或者 DQS 的锁存沿位基准。注意，R/B_n 和 WP_n 始终是异步信号。

对于按时钟测量的参数(如 tDSH)，参数分别从 CLK 或 DQS 的锁存沿开始测量。

Parameter	Description
tAC	Access window of DQ[7:0] from CLK
tCADf, tCADs	Command, Address, Data delay (command to command, address to address, command to address, address to command, command/address to start of data)
tCAH	Command/address DQ hold time
tCALH	W/R_n, CLE and ALE hold time
tCALS	W/R_n, CLE and ALE setup time
tCAS	Command/address DQ setup time
tCK(avg)[1]	Average clock cycle time, also known as tCK
tCK(abs)	Absolute clock period, measured from rising edge to the next consecutive rising edge
tCKH(abs)[2]	Clock cycle high
tCKL(abs)[2]	Clock cycle low
tCKWR	Data output end to W/R_n high
tDPZ	Data input pause setup time
tDQSCK	Access window of DQS from CLK
tDQSD[3]	W/R_n low to DQS/DQ driven by device
tDQSH	DQS input high pulse width
tDQSHZ[3]	W/R_n high to DQS/DQ tri-state by device
tDQSL	DQS input low pulse width
tDQSQ	DQS-DQ skew, DQS to last DQ valid, per access
tDQSS	Data input to first DQS latching transition
tDSC	DQS cycle time
tDSH	DQS falling edge to CLK rising – hold time
tDSS	DQS falling edge to CLK rising – setup time
tDVW	Output data valid window
tHP	Half-clock period
tJIT(per)	The deviation of a given tCK(abs) from tCK(avg)
tQH	DQ-DQS hold, DQS to first DQ to go non-valid, per access
tQHS	Data hold skew factor
tRHW	Data output cycle to command, address, or data input cycle
tWPRE	DQS write preamble
tWPST	DQS write postamble
tWRCK	W/R_n low to data output cycle

NOTE:
1. tCK(avg) is the average clock period over any consecutive 200 cycle window.
2. tCKH(abs) and tCKL(abs) include static offset and duty cycle jitter.
3. Refer to Appendix E for measurement technique.

Table 81 NV-DDR Timing Parameter Descriptions

4.17.4 NV-DDR2/NV-DDR3

表 82 定义了 NV-DDR2 和 NV-DDR3 接口的时序参数。

Parameter	Description
tAC	Access window of DQ[7:0] from RE_n (RE_t/RE_c crosspoint)
tAR	ALE to (RE_n low or RE_t/RE_c crosspoint)
tCAH	Command/address DQ hold time
tCALH	CLE and ALE hold time
tCALS	CLE and ALE setup time
tCALS2	CLE and ALE setup time when ODT is enabled
tCAS	Command/address DQ setup time
tCHZ[1]	CE_n high to output Hi-Z
tCLHZ[1]	CLE high to output Hi-Z
tCLR	CLE to (RE_n low or RE_t/RE_c crosspoint)
tCR	CE_n to (RE_n low or RE_t/RE_c crosspoint)
tCR2	CE_n to (RE_n low or RE_t/RE_c crosspoint) after CE_n has been high for greater than 1us
tCS1	CE_n setup time for data burst with ODT disabled
tCS2	CE_n setup time with DQS/DQ[7:0] ODT enabled
tCSD	ALE, CLE, WE_n hold time from CE_n high
tCDQSS	DQS setup time for data input start
tDBS	DQS (DQS_t) high and RE_n (RE_t) high setup to ALE, CLE and CE_n low during data burst
tDH_relaxed	Data DQ hold time relaxed timing
tDH_tight	Data DQ hold time tight timing
tDIPW	DQ input pulse width
tDQSD	(RE_n low or RE_t/RE_c crosspoint) to DQS/DQ driven by device
tDQSH	DQS high level width
tDQSL	DQS low level width
tDQSQ	DQS-DQ skew, DQS to last DQ valid, per access
tDQSRE	Access window of DQS from RE_n (RE_t/RE_c)
tDQSRH	DQS hold time after (RE_n low or RE_t/RE_c crosspoint)
tDS_relaxed	Data DQ setup time relaxed timing
tDS_tight	Data DQ setup time tight timing
tDSC(avg)	Average DQS cycle time
tDSC(abs)	Absolute write cycle period, measured from rising edge to the next consecutive rising edge
tDVW	Output data valid window
tJITper	The deviation of a given tRC(abs)/tDSC(abs) from tRC(avg)/tDSC(avg)
tJITcc	Cycle-to-cycle jitter
tQH	DQ-DQS hold, DQS to first DQ to go non-valid, per access
tQSH	DQS output high time (if differential, DQS_t is high)
tQSL	DQS output low time (if differential, DQS_t is low)
tRC(avg)	Average read cycle time, also known as tRC
tRC(abs)	Absolute read cycle period, measured from rising edge to the next consecutive rising edge
tREH(abs)	Absolute RE_n/RE_t high level width
tREH(avg)	Average RE_n/RE_t high level width
tRHW	RE_n high to WE_n low
tRP(abs)	Absolute RE_n/RE_t low level width
tRP(avg)	Average RE_n/RE_t low level width
tRPRE	Read preamble
tRPRE2	Read preamble with ODT enabled
tRPST	Read postamble
tRPSTH	Read postamble hold time
tWC	Write cycle time
tWH	WE_n high pulse width
tWP	WE_n low pulse width
tWPRE	DQS write preamble
tWPRE2	DQS write preamble when ODT enabled
tWPST	DQS write postamble
tWPSTH	DQS write postamble hold time
NOTE: 1. Refer to Appendix E for measurement technique.	

表 82 NV-DDR2/NV-DDR3 时序参数描述

4.18 时序模式

4.18.1 SDR

Parameter	Mode 0		Mode 1		Mode 2		Unit
	100		50		35		ns
	Min	Max	Min	Max	Min	Max	
tADL	400	—	400	—	400	—	ns
tALH	20	—	10	—	10	—	ns
tALS	50	—	25	—	15	—	ns
tAR	25	—	10	—	10	—	ns
tCEA	—	100	—	45	—	30	ns
tCEH	20	—	20	—	20	—	ns
tCH	20	—	10	—	10	—	ns
tCHZ	—	100	—	50	—	50	ns
tCLH	20	—	10	—	10	—	ns
tCLR	20	—	10	—	10	—	ns
tCLS	50	—	25	—	15	—	ns
tCOH	0	—	15	—	15	—	ns
tCR	10	—	10	—	10	—	ns
tCR2	100	—	100	—	100	—	ns
tCS	70	—	35	—	25	—	ns
tDH	20	—	10	—	5	—	ns
tDS	40	—	20	—	15	—	ns
tFEAT	—	1	—	1	—	1	µs
tIR	10	—	0	—	0	—	ns
tITC	—	1	—	1	—	1	µs
tRC	100	—	50	—	35	—	ns
tREA	—	40	—	30	—	25	ns
tREH	30	—	15	—	15	—	ns
tRHOH	0	—	15	—	15	—	ns
tRHW	200	—	100	—	100	—	ns
tRHZ	—	200	—	100	—	100	ns
tRLOH	0	—	0	—	0	—	ns
tRP	50	—	25	—	17	—	ns
tRR	40	—	20	—	20	—	ns
tRST (raw NAND)	—	5000	—	10/30/500	—	10/30/500	µs
tRST[2] (EZ NAND)	—	250000	—	150/150/500	—	150/150/500	µs
tWB	—	200	—	100	—	100	ns
tWC	100	—	45	—	35	—	ns
tWH	30	—	15	—	15	—	ns
tWHR	120	—	80	—	80	—	ns
tWP	50	—	25	—	17	—	ns
tWW	100	—	100	—	100	—	ns

NOTE:
1. To easily support EDO capable devices, tCHZ and tRHZ maximums are higher in modes 1, 2, and 3 than typically necessary for a non-EDO capable device.
2. If the reset is invoked using a Reset (FFh) command then the EZ NAND device has 250 ms to complete the reset operation regardless of the timing mode. If the reset is invoked using Reset LUN (FAh) command then the values are as shown.

表 83 SDR 时序模式 0，1，2

Parameter	Mode 3		Mode 4 (EDO capable)		Mode 5 (EDO capable)		Unit
	30		25		20		ns
	Min	Max	Min	Max	Min	Max	
tADL	400	—	400	—	400	—	ns
tALH	5	—	5	—	5	—	ns
tALS	10	—	10	—	10	—	ns
tAR	10	—	10	—	10	—	ns
tCEA	—	25	—	25	—	25	ns
tCEH	20	—	20	—	20	—	ns
tCH	5	—	5	—	5	—	ns
tCHZ	—	50	—	30	—	30	ns
tCLH	5	—	5	—	5	—	ns
tCLR	10	—	10	—	10	—	ns
tCLS	10	—	10	—	10	—	ns
tCOH	15	—	15	—	15	—	ns
tCR	10	—	10	—	10	—	ns
tCR2	100	—	100	—	100	—	ns
tCS	25	—	20	—	15	—	ns
tDH	5	—	5	—	5	—	ns
tDS	10	—	10	—	7	—	ns
tFEAT	—	1	—	1	—	1	µs
tIR	0	—	0	—	0	—	ns
tITC	—	1	—	1	—	1	µs
tRC	30	—	25	—	20	—	ns
tREA	—	20	—	20	—	16	ns
tREH	10	—	10	—	7	—	ns
tRHOH	15	—	15	—	15	—	ns
tRHW	100	—	100	—	100	—	ns
tRHZ	—	100	—	100	—	100	ns
tRLOH	0	—	5	—	5	—	ns
tRP	15	—	12	—	10	—	ns
tRR	20	—	20	—	20	—	ns
tRST (raw NAND)	—	10/30/500	—	10/30/500	—	10/30/500	µs
tRST[2] (EZ NAND)	—	150/150/500	—	150/150/500	—	150/150/500	µs
tWB	—	100	—	100	—	100	ns
tWC	30	—	25	—	20	—	ns
tWH	10	—	10	—	7	—	ns
tWHR	80	—	80	—	80	—	ns
tWP	15	—	12	—	10	—	ns
tWW	100	—	100	—	100	—	ns

NOTE:
1. To easily support EDO capable devices, tCHZ and tRHZ maximums are higher in modes 1, 2, and 3 than typically necessary for a non-EDO capable device.
2. If the reset is invoked using a Reset (FFh) command then the EZ NAND device has 250 ms to complete the reset operation regardless of the timing mode. If the reset is invoked using Reset LUN (FAh) command then the values are as shown.

Table 84 SDR Timing Modes 3, 4, and 5

4.18.2 NV-DDR

表 85 描述了标准的 NV-DDR 接口时序模式。不要求 host 的时钟周期与标准时序模式的时钟周期精确匹配。Host 应遵守选定时序模式的 setup 和 hold time。如果 host 通过 Set Feature 命令选定了时序模式 n，那么 host 的时钟应该比时序模式 n-1 的时钟快，而比时序模式 n 的时钟慢或者相等。例如，如果 host 选定了时序模式 2，那么应符合以下公式：

30ns > host 时钟周期 >= 20ns

如果时序模式 0 被选中，则 host 的时钟周期应不慢于 100ns(即小于 100ns)。该要求唯一的例外是当 host 以 SDR 时序模式 0 发送了一个 Reset(FFh)，参见 4.5。

表示数据输入、数据输出、地址或命令周期之前的延迟的时序参数，在延迟(ns 级别)过后应符合时钟上升沿。可以使用以下公式来计算可能发生转变的第一个沿：

=Roundup{[tParam + tCK]/tCK}

时序模式应符合表 29 定义的测试条件。

Parameter	Mode 0		Mode 1		Mode 2		Mode 3		Mode 4		Mode 5		Unit
	50		30		20		15		12		10		ns
	20		~33		50		~66		~83		100		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
tAC	3	25	3	25	3	25	3	25	3	25	3	25	ns
tADL	400	—	400	—	400	—	400	—	400	—	400	—	ns
tCADf	25	—	25	—	25	—	25	—	25	—	25	—	ns
tCADs	45	—	45	—	45	—	45	—	45	—	45	—	ns
tCAH	10	—	5	—	4	—	3	—	2.5	—	2	—	ns
tCALH	10	—	5	—	4	—	3	—	2.5	—	2	—	ns
tCALS	10	—	5	—	4	—	3	—	2.5	—	2	—	ns
tCAS	10	—	5	—	4	—	3	—	2.5	—	2	—	ns
tCEH	20	—	20	—	20	—	20	—	20	—	20	—	ns
tCH	10	—	5	—	4	—	3	—	2.5	—	2	—	ns
tCK(avg) or tCK	50	—	30	—	20	—	15	—	12	—	10	—	ns
tCK(abs)	Minimum: tCK(avg) + tJIT(per) min Maximum: tCK(avg) + tJIT(per) max												ns
tCKH(abs)	0.43	0.57	0.43	0.57	0.43	0.57	0.43	0.57	0.43	0.57	0.43	0.57	tCK
tCKL(abs)	0.43	0.57	0.43	0.57	0.43	0.57	0.43	0.57	0.43	0.57	0.43	0.57	tCK
tCKWR	Minimum: RoundUp{[tDQSCK(max) + tCK] / tCK} Maximum: —												tCK
tCS	35	—	25	—	15	—	15	—	15	—	15	—	ns
tDH	5	—	2.5	—	1.7	—	1.3	—	1.1	—	0.9	—	ns
tDPZ	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	tDSC
tDQSCK	3	25	3	25	3	25	3	25	3	25	3	25	ns
tDQSD	0	18	0	18	0	18	0	18	0	18	0	18	ns
tDQSH	0.4	0.6	0.4	0.6	0.4	0.6	0.4	0.6	0.4	0.6	0.4	0.6	tCK or tDSC[4]
tDQSHZ	—	20	—	20	—	20	—	20	—	20	—	20	ns
tDQSL	0.4	0.6	0.4	0.6	0.4	0.6	0.4	0.6	0.4	0.6	0.4	0.6	tCK or tDSC[4]
tDQSQ	—	5	—	2.5	—	1.7	—	1.3	—	1.0	—	0.85	ns
tDQSS	0.75	1.25	0.75	1.25	0.75	1.25	0.75	1.25	0.75	1.25	0.75	1.25	tCK
tDS	5	—	3	—	2	—	1.5	—	1.1	—	0.9	—	ns
tDSC	50	—	30	—	20	—	15	—	12	—	10	—	ns
tDSH	0.2	—	0.2	—	0.2	—	0.2	—	0.2	—	0.2	—	tCK
tDSS	0.2	—	0.2	—	0.2	—	0.2	—	0.2	—	0.2	—	tCK

tDVW	tDVW = tQH – tDQSQ												ns
tFEAT	—	1	—	1	—	1	—	1	—	1	—	1	µs
tHP	tHP = min(tCKL, tCKH)												ns
tITC	—	1	—	1	—	1	—	1	—	1	—	1	µs
tJIT(per)	-0.7	0.7	-0.7	0.7	-0.7	0.7	-0.6	0.6	-0.6	0.6	-0.5	0.5	ns
tQH	tQH = tHP – tQHS												ns
tQHS	—	6	—	3	—	2	—	1.5	—	1.2	—	1.0	ns
tRHW	100	—	100	—	100	—	100	—	100	—	100	—	ns
tRR	20	—	20	—	20	—	20	—	20	—	20	—	ns
tRST (raw NAND)	—	10/30/ 500	—	10/30/ 500	—	10/30/ 500	—	10/30/ 500	—	10/30/ 500	—	10/30/ 500	µs
tRST[2] (EZ NAND)	—	150/ 150/ 500	—	150/ 150/ 500	—	150/ 150/ 500	—	150/ 150/ 500	—	150/ 150/ 500	—	150/ 150/ 500	µs
tWB	—	100	—	100	—	100	—	100	—	100	—	100	ns
tWHR	80	—	80	—	80	—	80	—	80	—	80	—	ns
tWPRE	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	tCK
tWPST	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	1.5	—	tCK
tWRCK	20	—	20	—	20	—	20	—	20	—	20	—	ns
tWW	100	—	100	—	100	—	100	—	100	—	100	—	ns

NOTE:

1. tDQSHZ is not referenced to a specific voltage level, but specifies when the device output is no longer driving.
2. tCK(avg) is the average clock period over any consecutive 200 cycle window.
3. tCKH(abs) and tCKL(abs) include static offset and duty cycle jitter.
4. tDQSL and tDQSH are relative to tCK when CLK is running. If CLK is stopped during data input, then tDQSL and tDQSH are relative to tDSC.
5. If the reset is invoked using a Reset (FFh) command then the EZ NAND device has 250 ms to complete the reset operation regardless of the timing mode. If the reset is invoked using Synchronous Reset (FCh) or a Reset LUN (FAh) command then the values are as shown.

Table 85 NV-DDR Timing Modes

4.18.3 NV-DDR2/NV-DDR3

表 86，表 87，表 88 和表 89 定义了 NV-DDR2 和 NV-DDR3 接口的时序模式。不要求 host 的时钟周期与标准时序模式中的 tDSC 或 tRC 值精确匹配。Host 应满足选定的时序模式的 setup 和 hold time。如果 host 通过 Set Feature 命令选中了时序模式 n，则 host 的 tDSC 和 tRC 值应比时序模式 n-1 的 tDSC 和 tRC 值要快(即小于)，而比时序模式 n 的 tDSC 和 tRC 值要慢(即大于)或者相等。

时序模式应符合表 29 定义的测试条件。对高于 200MT/s 的速度，使用 25 Ohm 的驱动强度，差分信号 RE_n(RE_t/RE_c)和 DQS(DQS_t/DQS_c),以及外部 VREFQ 来验证时序参数。对高于 533MT/s 的速度，通过执行 ZQ 校正来验证时序参数。对高于 200MT/s 的速度，开发时应使用可变的驱动强度设置，单端信号，和/或内部 VREFQ；这种情况下时序参数的验证由开发者自己定义。如果时序模式 0 被选中，则 tRC 和 tDSC 周期应不慢于 100ns(即小于或等于 100ns)。

由于 NAND 应用及相关拓扑的多样性，NAND device 可以支持不同的 tDS 和 tDH 值并且仍能达到相同的数据输入频率。为了满足不同使用类型的 NAND device，NV-DDR2 和 NV-DDR3 接口在时序模式 4-10 中，每种模式都支持两个 tDS 和 tDH 值，这两类值被定义为 tDS_tight,tDH_tight,tDS_relaxed 以及 tDH_relaxed,如表 88 所示。

时序参数值应满足以下要求(以下部分翻译不是很准确，以英文版为准)：

1. tCHZ 和 tCLHZ 并不是由一个明确的电压确定，而是当 device 输出不再有驱动时确定。
2. 参数 tRC(avg)和 tDSC(avg)是任意的连续 200 个周期的平均值，tRC(avg)/tDSC(avg)是允许的最小比率，其中没有计算由 tJITper 引起的误差。
3. 输入抖动是被允许的，但不能超过规定值。
4. tREH(avg)和 PR(avg)是任意连续 200 个时钟周期平均值的一半，并且是允许半周期的一半，可以有时钟抖动引起的误差。时钟输入抖动是允许的，但不能超过规定值。
5. 周期抖动(tJITper)是 tRC 或 tDSC 周期的最大误差，误差允许为正误差或者负误差。
6. 相邻周期抖动(cycle-to-cycle jitter—tJITcc)时钟从一个周期到下一个周期的偏差。
7. 占空比抖动(duty cycle jitter)用于高电平脉冲和低电平脉冲，但两个的累积值不能超过 tJITper。绝对最小半周期 tRP(abd),tREH(abs),tDQSH 或 tDQSL 不小于平均 cycle 的 43%。
8. 当 device 的操作存在输入 RE_n(RE_t/RE_c)抖动时，tQSL,tQSH 和 tQH 需要通过实际大于 0.45*tRC(avg)但小于 0.43*tRC(avg)的输入占空比抖动来减免。
9. 将占空比抖动降低到大于平均周期的 45%会导致更大的 tQH，进而形成一个更大的数据有效窗口。device 的占空比每改善 1%，会至少提高 0.5%的数据有效窗口。例如，host 的 tREH(abs)为 0.49*tRC(avg),那么 device 达到的 tQH 至少为 0.369*tRC(avg)。
10. 表中所列的 tDS 和 tDH 基于输入压摆率(slew rate)1V/ns。如果输入压摆率不是 1V/ns，则应使用减免方法(derating methodology)。
11. 参数 tDIPW 定义为输入信号的脉冲宽度,这个输入信号介于第一个 VREFQ(DC)和接下来的 VREFQ(DC)。

Constant Timing Parameter Values			
	Min	Max	Unit
tAR	10	—	ns
tCAH	5	—	ns
tCAS	5	—	ns
tCALH	5	—	ns
tCALS	15	—	ns
tCALS2	25	—	ns
tCEH	20	—	ns
tCH	5	—	ns
tCS	20	—	ns
tCS1	30	—	ns
tCS2	40	—	ns
tCSD	10	—	ns
tCHZ	—	30	ns
tCLHZ	—	30	ns
tCLR	10	—	ns
tCR	10	—	ns
tCR2	100	—	ns
tDBS	5	—	ns
tRHW	100	—	ns
tWC	25	—	ns
tWH	11	—	ns
tWP	11	—	ns
tWHR	80	—	ns
tWW	100	—	ns
tFEAT	—	1	µs
tITC	—	1	µs
tRST	—	10/30/500	µs
tRR	20	—	ns
tWB	—	100	ns

Timing Mode Specific Values (Modes 0-3)									
	Mode 0		Mode 1		Mode 2		Mode 3		Unit
	30		25		15		12		ns
	~ 33		40		~ 66		~ 83		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tADL	400	—	400	—	400	—	400	—	ns

Timing Mode Specific Values (Modes 4-7)									
	Mode 4		Mode 5		Mode 6		Mode 7		Unit
	10		7.5		6		5		ns
	100		~133		~166		200		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tADL	400	—	400	—	400	—	400	—	ns

Timing Mode Specific Values (Modes 8-10)									
	Mode 8		Mode 9		Mode 10				Unit
	3.75		3		2.5				ns
	~267		~333		400				MHz
	Min	Max	Min	Max	Min	Max			
tADL	400	—	400	—	400	—			ns

Table 86 NV-DDR2 / NV-DDR3 Timing Parameter Values: Command and Address

Timing Mode Specific Values (Modes 0-3)									
	Mode 0		Mode 1		Mode 2		Mode 3		Unit
	30		25		15		12		ns
	~ 33		40		~ 66		~ 83		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tJITper (DQS)	-2.4	2.4	-2.0	2.0	-1.2	1.2	-1.0	1.0	ns
tJITper (RE_n)	-1.8	1.8	-1.5	1.5	-0.9	0.9	-0.75	0.75	ns
tJITcc (DQS)	4.8	—	4.0	—	2.4	—	2.0	—	ns
tJITcc (RE_n)	3.6	—	3.0	—	1.8	—	1.5	—	ns
Timing Mode Specific Values (Modes 4-7)									
	Mode 4		**Mode 5**		**Mode 6**		**Mode 7**		**Unit**
	10		7.5		6		5		ns
	100		~133		~166		200		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tJITper(DQS)	-0.80	0.80	-0.60	0.60	-0.48	0.48	-0.40	0.40	ns
tJITper(RE_n)	-0.60	0.60	-0.45	0.45	-0.36	0.36	-0.30	0.30	ns
tJITcc(DQS)	1.6	—	1.2	—	0.96	—	0.80	—	ns
tJITcc(RE_n)	1.2	—	0.90	—	0.72	—	0.60	—	ns
Timing Mode Specific Values (Modes 8-10)									
	Mode 8		**Mode 9**		**Mode 10**				**Unit**
	3.75		3		2.5				ns
	~267		~333		400				MHz
	Min	Max	Min	Max	Min	Max			
tJITper(DQS)	-0.30	0.30	-0.24	0.24	-0.2	0.2			ns
tJITper(RE_n)	-0.225	0.225	-0.18	0.18	-0.15	0.15			ns
tJITcc(DQS)	0.6	—	0.48	—	0.4	—			ns
tJITcc(RE_n)	0.45	—	0.36	—	0.3	—			ns

Table 87 NV-DDR2 / NV-DDR3 Timing Parameter Values: Jitter

Constant Timing Parameter Values			
	Min	Max	Unit
tCDQSS	30	—	ns
tDIPW	0.31	—	tDSC(avg)
tDQSH	0.43	—	tDSC(avg)
tDQSL	0.43	—	tDSC(avg)
tDSC(abs)	tDSC(avg) + tJITper(DQS) min	tDSC(avg) + tJITper(DQS) max	ns
tWPRE	15	—	ns
tWPRE2	25	—	ns
tWPST	6.5	—	ns
tWPSTH	15	—	ns

Timing Mode Specific Values (Modes 0-3)									
	Mode 0		Mode 1		Mode 2		Mode 3		Unit
	30		25		15		12		ns
	~ 33		40		~ 66		~ 83		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tDH	4	—	3.3	—	2.0	—	1.1	—	ns
tDS	4	—	3.3	—	2.0	—	1.1	—	ns
tDSC(avg) or tDSC	30	—	25	—	15	—	12	—	ns
Timing Mode Specific Values (Modes 4-7)									
	Mode 4		**Mode 5**		**Mode 6**		**Mode 7**		**Unit**
	10		7.5		6		5		ns
	100		~133		~166		200		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tDH_tight	0.7	—	0.5	—	0.4	—	0.35	—	ns
tDS_tight	0.7	—	0.5	—	0.4	—	0.35	—	ns
tDH_relaxed	0.9	—	0.75	—	0.55	—	0.40	—	ns
tDS_relaxed	0.9	—	0.75	—	0.55	—	0.40	—	ns
tDSC(avg) or tDSC	10	—	7.5	—	6	—	5	—	ns
Timing Mode Specific Values (Modes 8-10)									
	Mode 8		**Mode 9**		**Mode 10**				**Unit**
	3.75		3		2.5				ns
	~267		~333		400				MHz
	Min	Max	Min	Max	Min	Max			
tDH_tight	0.3	—	0.24	—	0.2	—			ns
tDS_tight	0.3	—	0.24	—	0.2	—			ns
tDH_relaxed	0.35	—	0.31	—	0.26	—			ns
tDS_relaxed	0.35	—	0.31	—	0.26	—			ns
tDSC(avg) or tDSC	3.75	—	3	—	2.5	—			ns

Table 88 NV-DDR2 / NV-DDR3 Timing Parameter Values: Data Input

Constant Timing Parameter Values									
	Min				Max				Unit
tAC	3				25				ns
tDQSD	5				18				ns
tDQSRE	3				25				ns
tDVW	tDVW = tQH – tDQSQ								ns
tQH	0.37				—				tRC(avg)
tQSH	0.37				—				tRC(avg)
tQSL	0.37				—				tRC(avg)
tRC(abs)	tRC(avg) + tJITper(RE_n) min				tRC(avg) + tJITper(RE_n) max				ns
tREH(abs)	0.43				—				tRC(avg)
tRP(abs)	0.43				—				tRC(avg)
tREH(avg)	0.45				0.55				tRC(avg)
tRP(avg)	0.45				0.55				tRC(avg)
tRPRE	15				—				ns
tRPRE2	25				—				ns
tRPST	tDQSRE + 0.5*tRC				—				ns
tRPSTH	15				—				ns
tDQSRH	5				—				ns
Timing Mode Specific Values (Modes 0-3)									
	Mode 0		**Mode 1**		**Mode 2**		**Mode 3**		**Unit**
	30		25		15		12		ns
	~ 33		40		~ 66		~ 83		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tDQSQ	—	2.5	—	2.0	—	1.4	—	1.0	ns
tRC(avg) or tRC	30	—	25	—	15	—	12	—	ns
Timing Mode Specific Values (Modes 4-7)									
	Mode 4		**Mode 5**		**Mode 6**		**Mode 7**		**Unit**
	10		7.5		6		5		ns
	100		~133		~166		200		MHz
	Min	Max	Min	Max	Min	Max	Min	Max	
tDQSQ	—	0.8	—	0.6	—	0.5	—	0.4	ns
tRC(avg) or tRC	10	—	7.5	—	6	—	5	—	ns
Timing Mode Specific Values (Modes 8-10)									
	Mode 8		**Mode 9**		**Mode 10**				**Unit**
	3.75		3		2.5				ns
	~267		~333		400				MHz
	Min	Max	Min	Max	Min	Max			
tDQSQ	—	0.350	—	0.3	—	0.25			ns
tRC(avg) or tRC	3.75	—	3	—	2.5	—			ns

Table 89 NV-DDR2 / NV-DDR3 Timing Parameter Values: Data Output

4.19 时序图

本章节定义了每种数据接口中每个操作阶段(命令，地址，数据输入和数据输出周期)的时序图。每种接口中 Read ID 命令的时序图如图 82，图 83 和图 84 所示。每种接口的读状态命令(Read Status)的时序图如图 89，图 90 和图 91 所示。

4.19.1 SDR

4.19.1.1 命令锁存(Latch)时序

R/B_n 信号的要求仅适用于一种命令：当命令发送后 R/B_n 被清为 0，如命令定义中所规定。

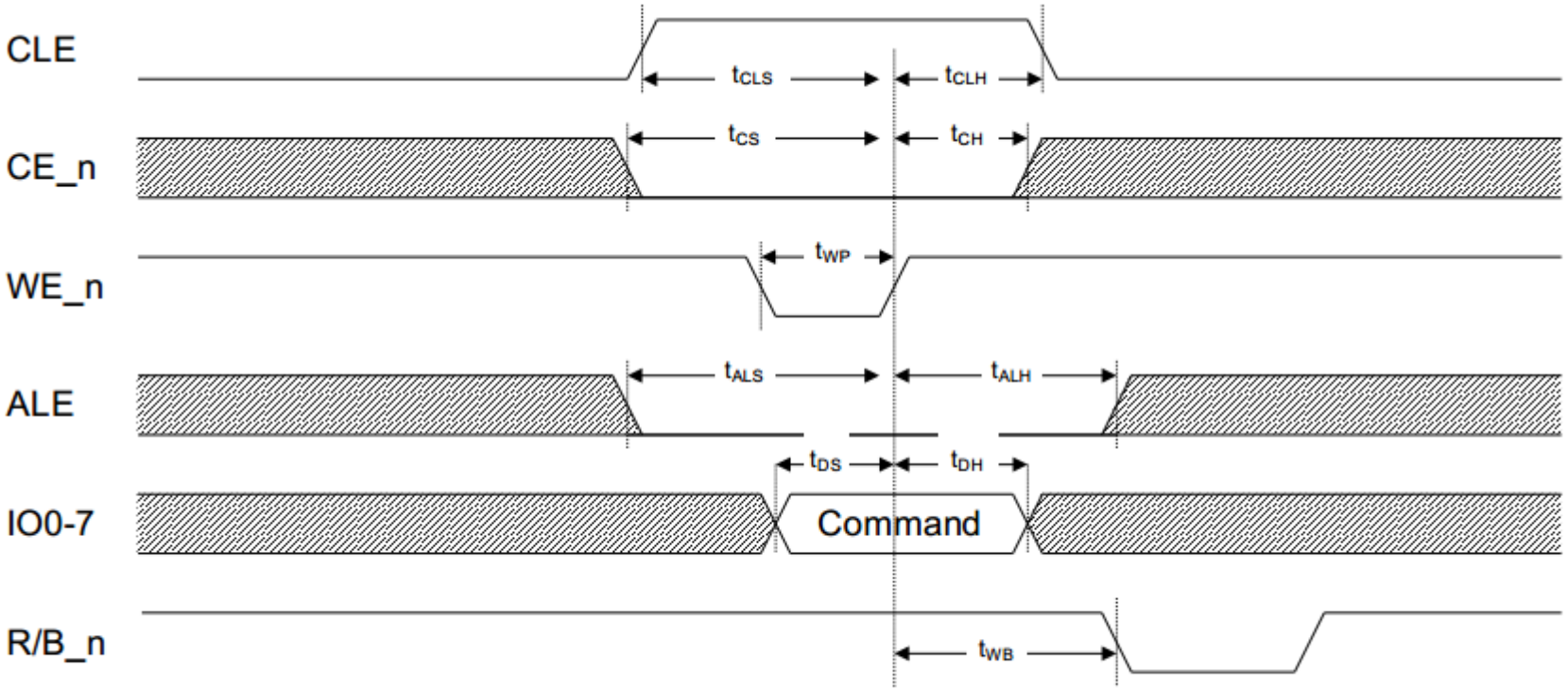


Figure 53 Command latch timings

4.19.1.2 地址锁存时序

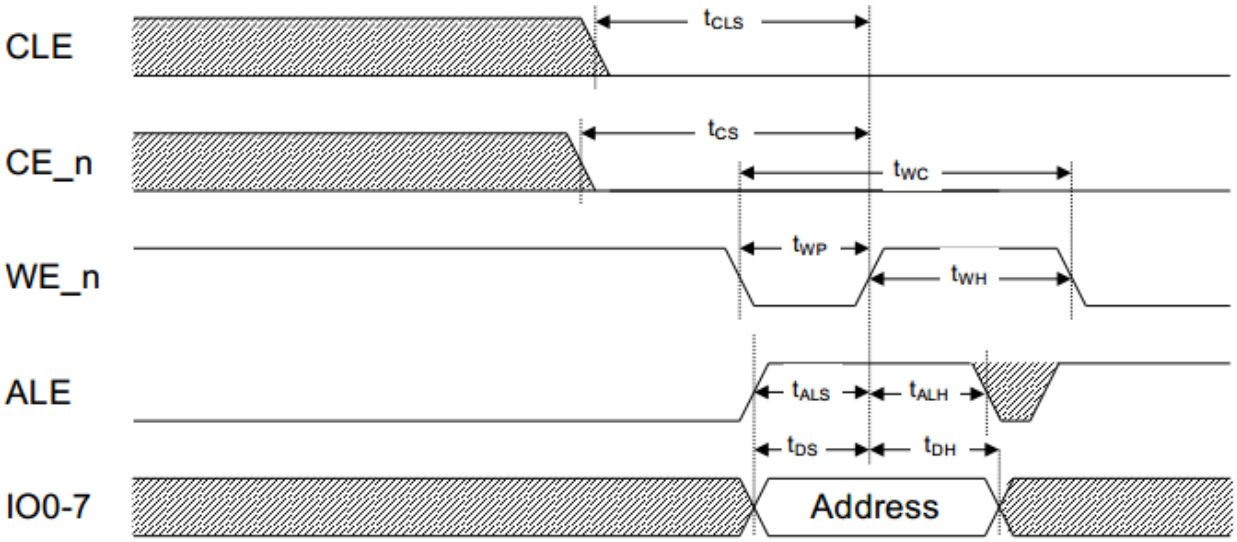


Figure 54　　Address latch timings

4.19.1.3　数据输入周期时序

数据输入可能不需要关心 CE_n。如果不需要关心 CE_n，则 host 应符合 tCS 和 tCH 时序要求，并且 WE_n 下降沿应该始终出现在 CE_n 下降沿之后(例如，tCS>tWP)。

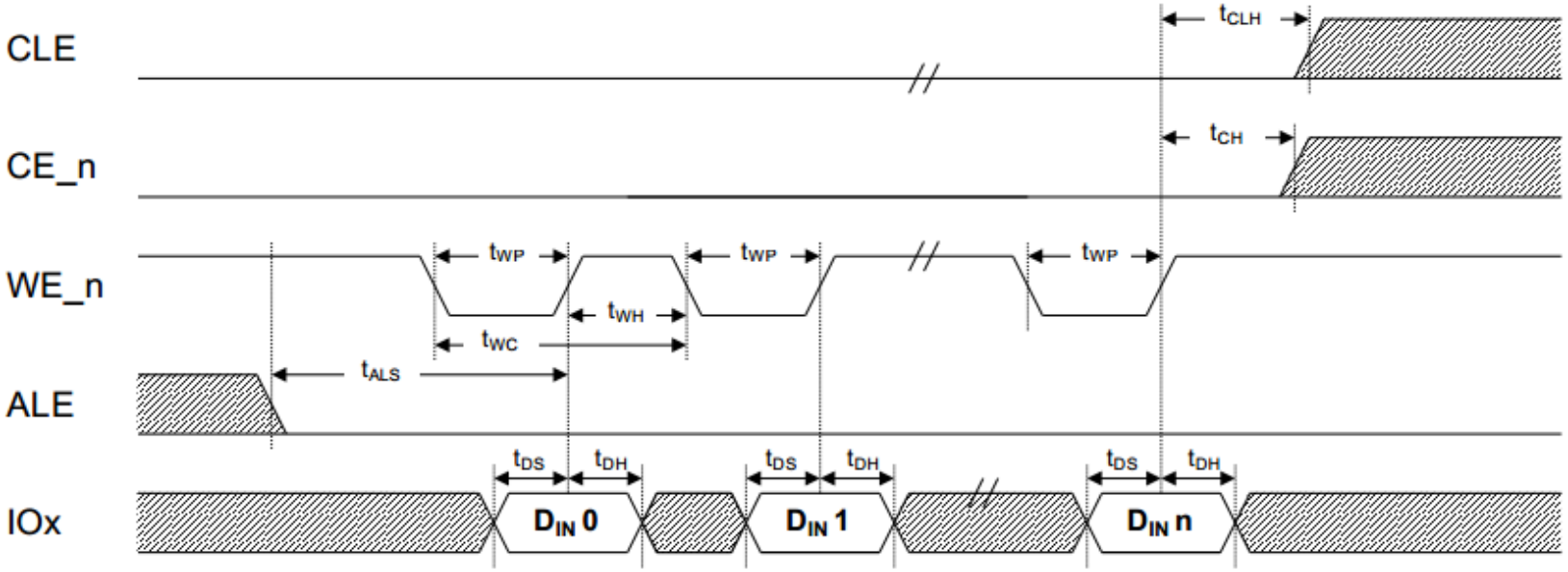


Figure 55　　Data input cycle timings

4.19.1.4 数据输出周期时序

数据输出可能不需要关心 CE_n。如果不用关心 CE_n，则 host 应符合 tCEA 和 tCOH 时序要求，并且 RE_n 应该或者保持为低，或者下降沿出现在 CE_n 下降沿之后。

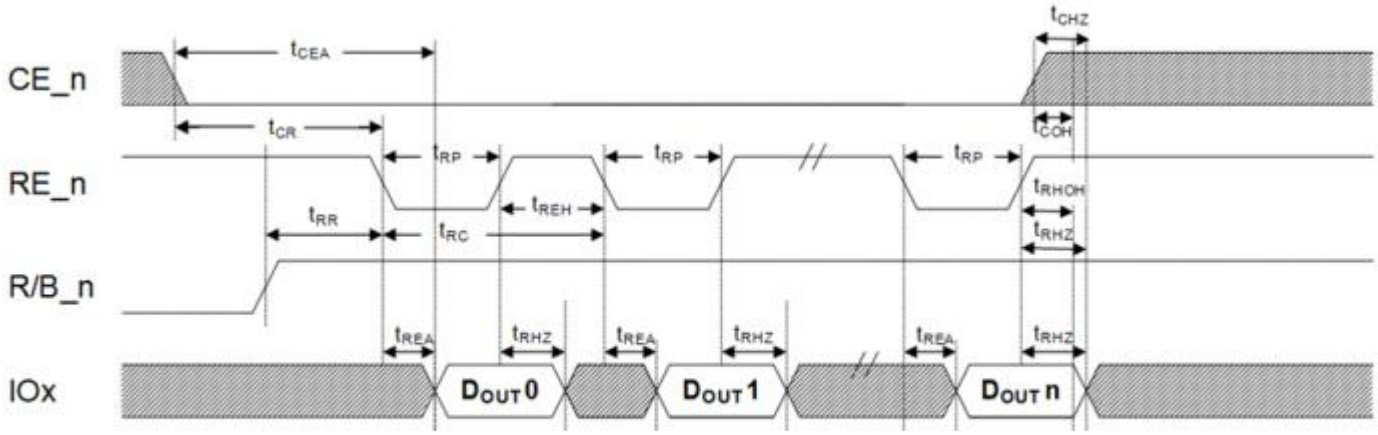


Figure 56 Data output cycle timings

4.19.1.5 数据输出周期时序(EDO)

EDO 数据输出周期时序应该被用于当 host 驱动 tRC 小于 30ns 时。数据输出可能不需要关心 CE_n。如果不用关心 CE_n，则 host 应满足 tCEA 和 tCOH 时序要求。

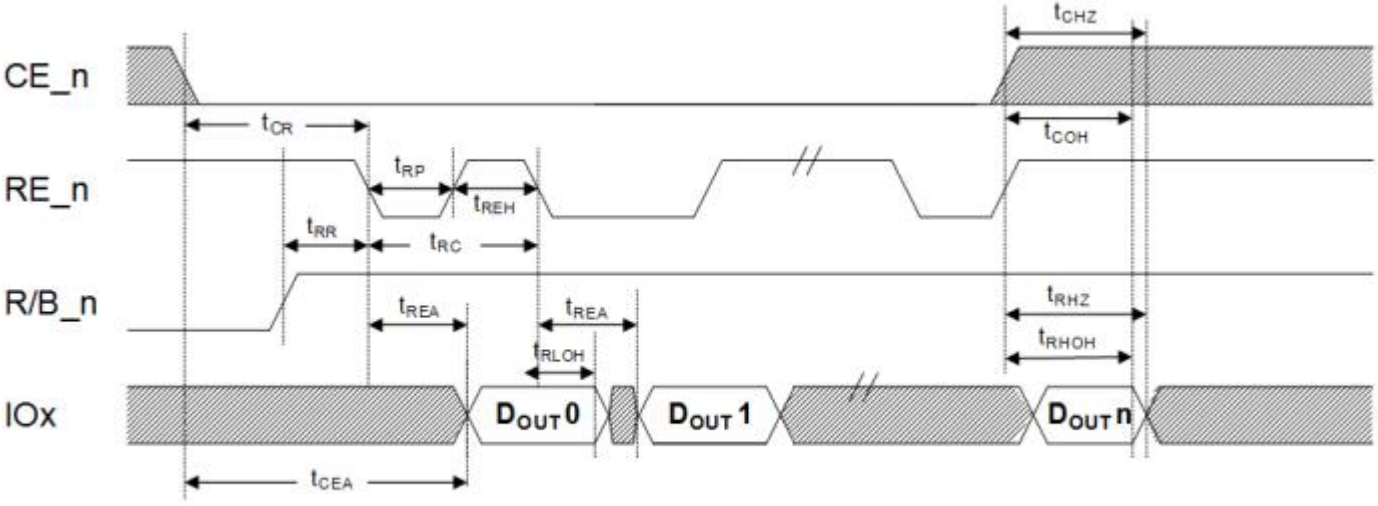


Figure 57 EDO data output cycle timings

4.19.1.6　读状态(Read Status)时序

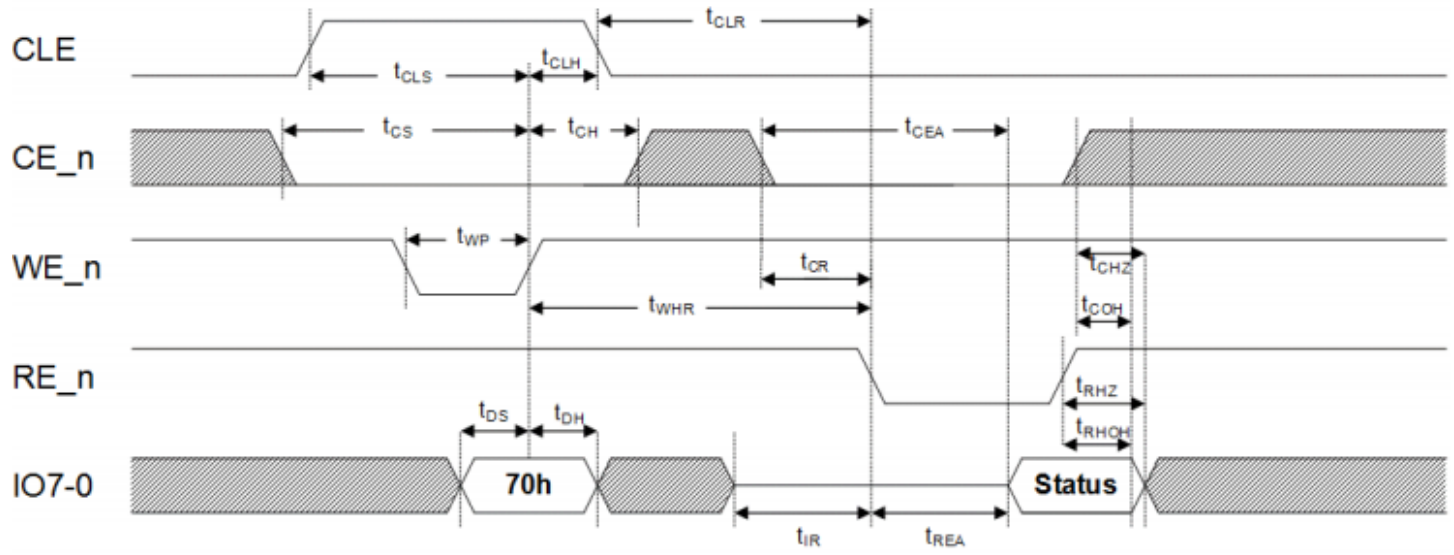


Figure 58　　Read Status timings

4.19.1.7　Read Status Enhanced 时序

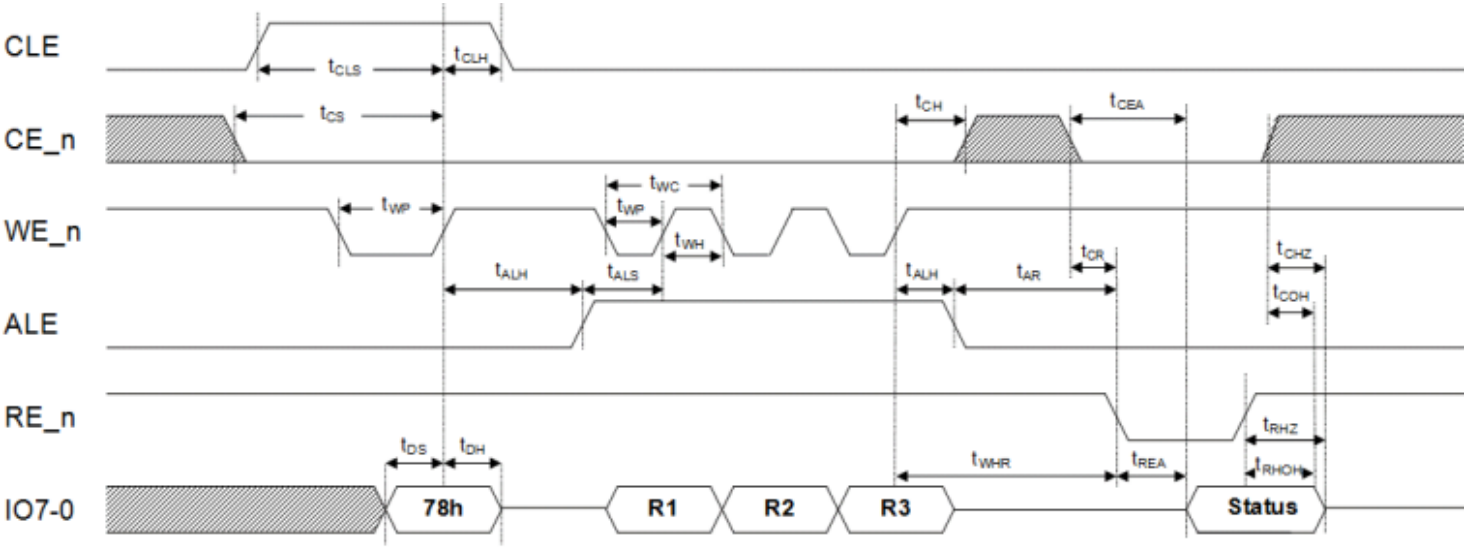


Figure 59　　Read Status Enhanced timings

4.19.2 NV-DDR

对于命令，地址，数据输入和数据输出时序图，tCS 时序参数可能会消耗多个时钟周期。host 应满足 tCS(基于 CLK 的上升沿)，如图所示，因此 host 应该在此之前将 CE_n 拉低并保持足够时间，以便符合这个要求(可能会跨多个时钟周期)。

4.19.2.1 命令周期时序

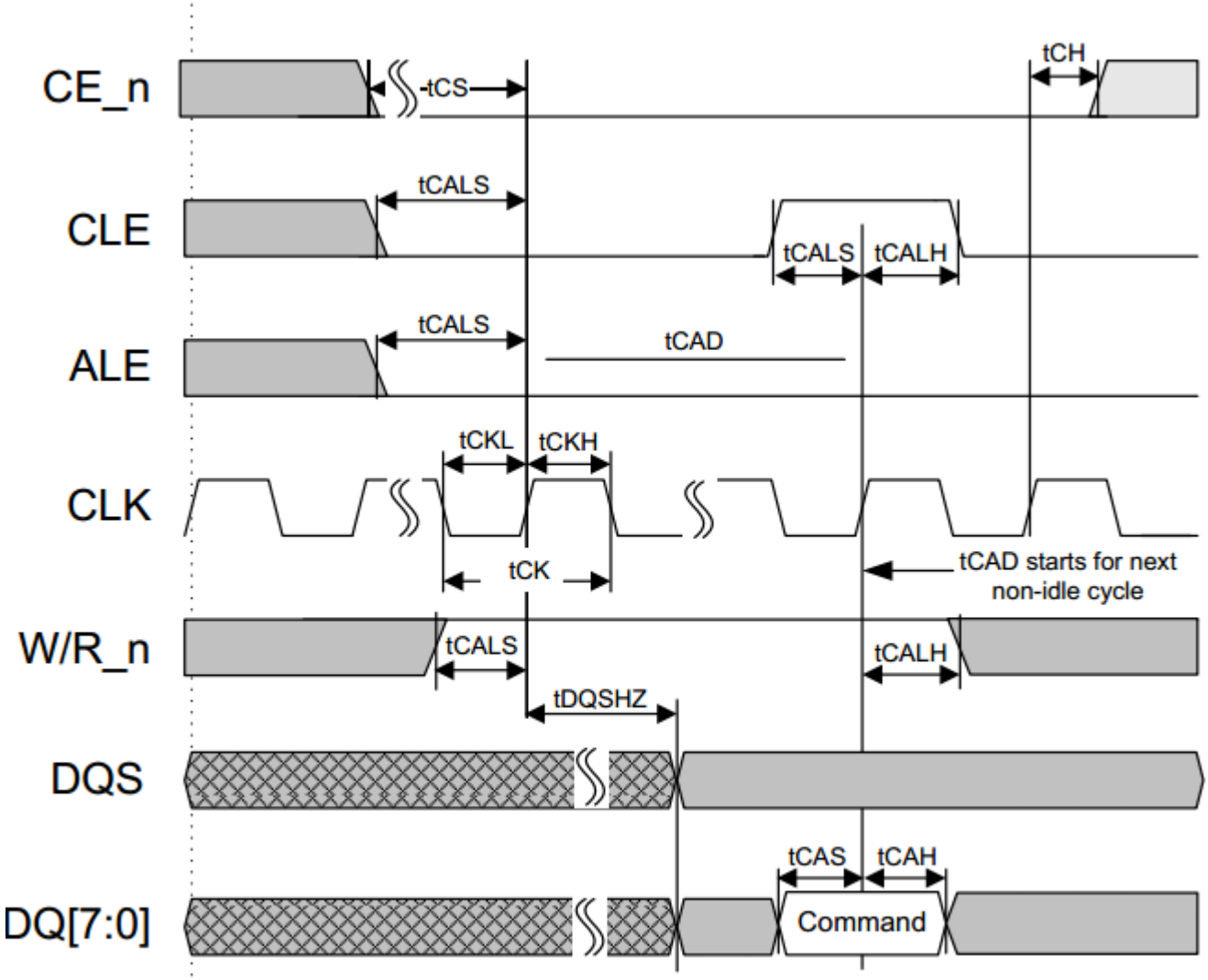


Figure 60 Command cycle timings

注意：

1. tCAD 测量的起始周期可能是 idle 周期(如图所示)、另一个命令周期、一个地址周期，或者一个数据周期。为简化说明，本图所示为 idle 周期。
2. 当 CE_n 从 1 变为 0 时，ALE 和 CLE 处于有效状态，在图中，这个有效状态等效于 tCS 和 tCALS 表示的时间。但是 tCS 和 tCALS 的值是不同的，这两个时序参数值定义在表 85 中。

4.19.2.2 地址周期时序

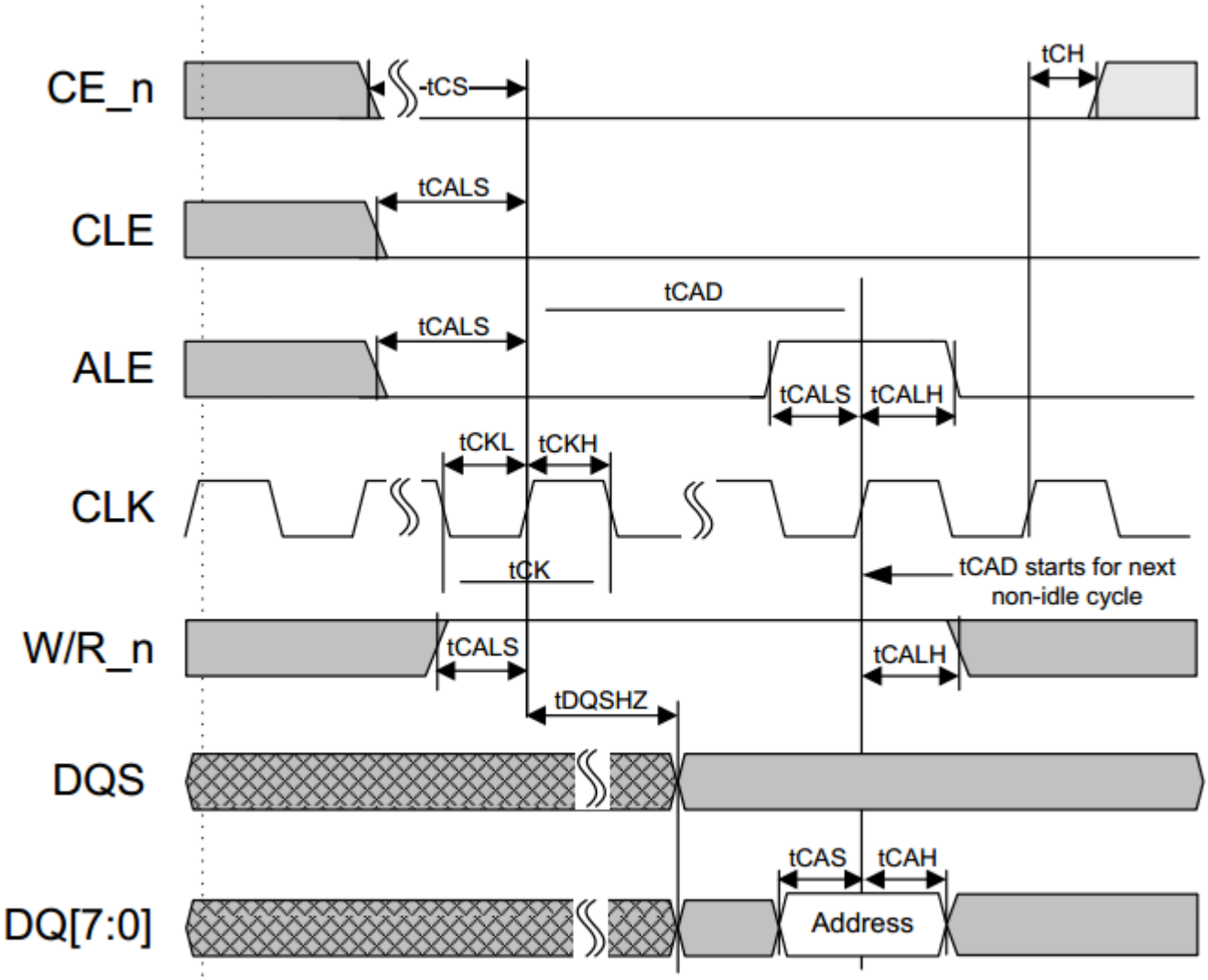


Figure 61 Address cycle timings

注意：

1. 当 CE_n 从 1 变为 0 时，ALE 和 CLE 才处于有效状态。图中，这个有效状态等效于 tCS 和 tCALS 表示的时间，但是 tCS 和 tCALS 的值是不同的，这两个时序参数值定义在表 85 中。

4.19.2.3 数据输入周期时序

数据输入周期时序描述了数据从 host 传输到 device 的时序(例如，数据写)。

对于 Set Feature 命令，相同的数据 byte 会重复两次。在这种情况下，数据模式(pattern)为 D_0 D_0 D_1 D_1 D_2 D_2 等。device 只能锁存每个数据 byte 的一个备份(copy)。CLK 在 Set Feature 命令的数据输入过程中不能被停止。不要求 device 在开始内部操作之前等待重复的数据 byte。

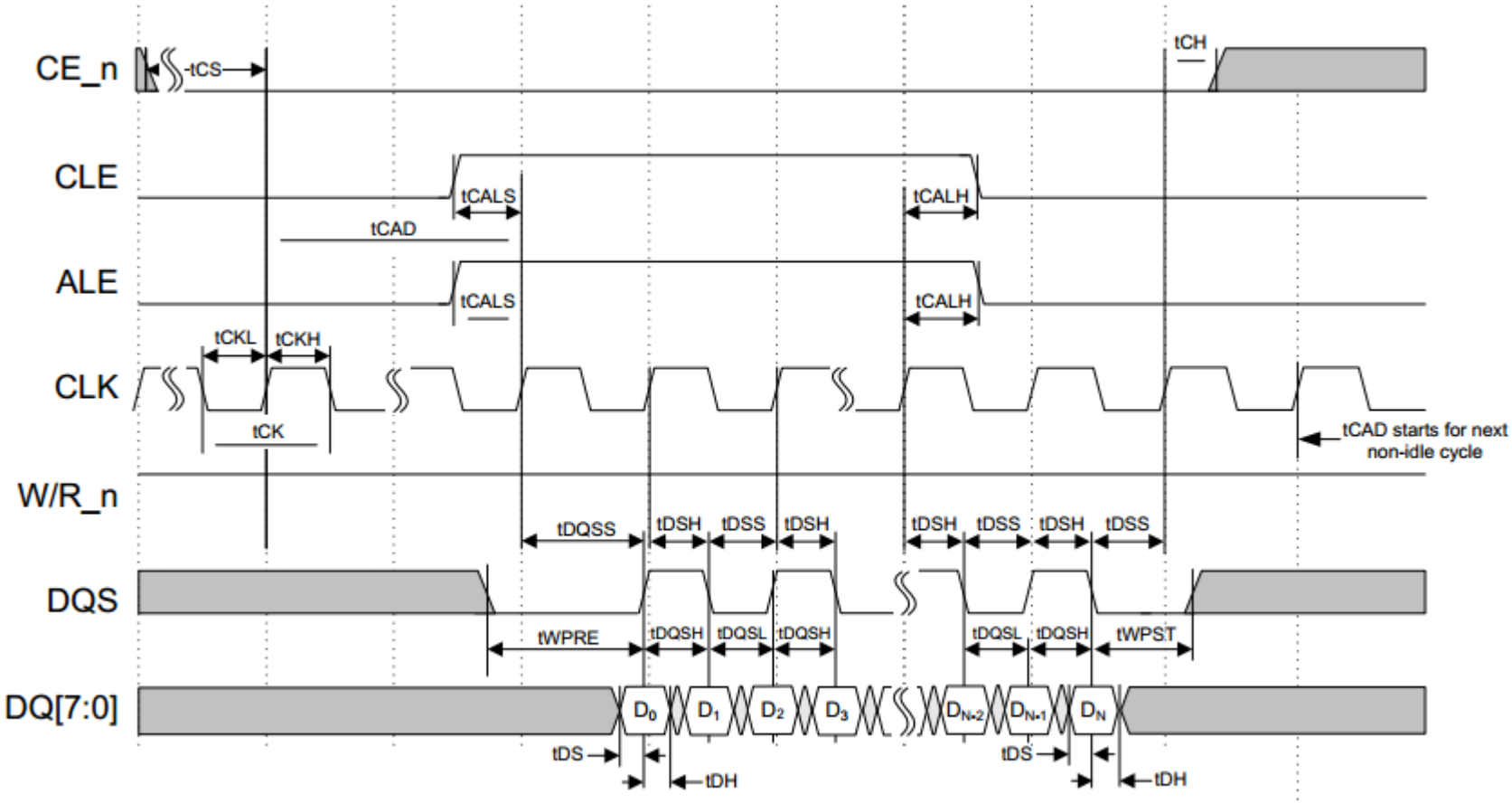


Figure 62　　Data input cycle timing

4.19.2.4　数据输入周期时序，CLK 被停止

在数据输入周期期间，host 为了省电会将 CLK 信号保持为高(停止 CLK)。只在参数 page 中表明 device 支持该特性，host 才能停止 CLK。数据输入周期时序描述了数据从 host 传输到 device 的时序(例如，数据写)。图 63 描述了 CLK 信号被停止的情况下数据输入周期的时序。ALE、CLE 和 W/R_n 信号的值在 CLK 的上升沿锁存，因此当 CLK 保持在高时不关心这些信号。

图 64 展现了当 CLK 被停止的情况下数据输入周期，此时 Host 可随意暂停数据输入。当 host 检测到 tDPZ 时可以暂停数据输入。tDPZ 时序参数用来重新启动到 device 的数据输入操作。当 CLK 重新开始时，host 应能够检测到图 63 和图 64 所示的时序参数，包括 tDSS 和 tDSH。

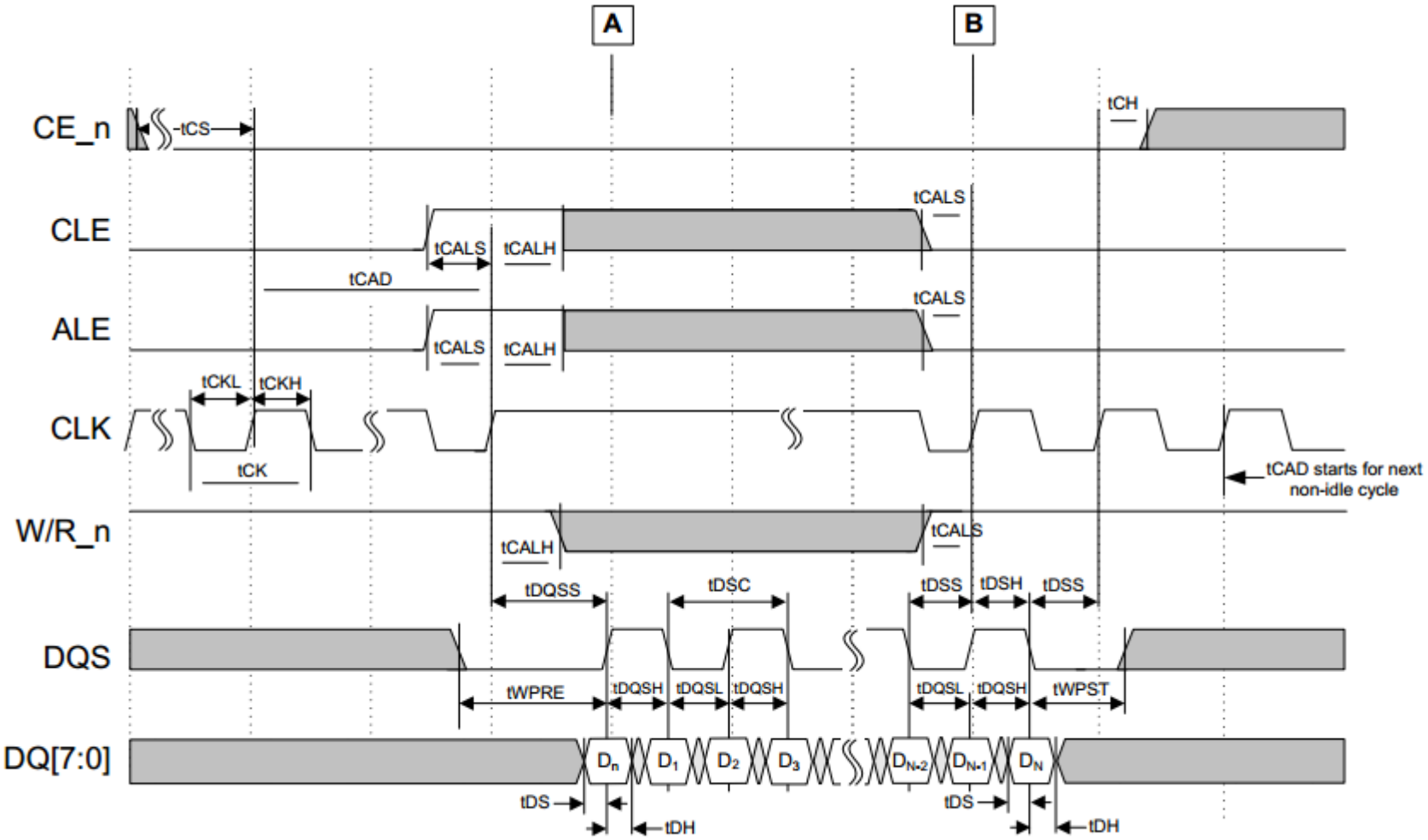


Figure 63 Data input cycle timing, CLK stopped

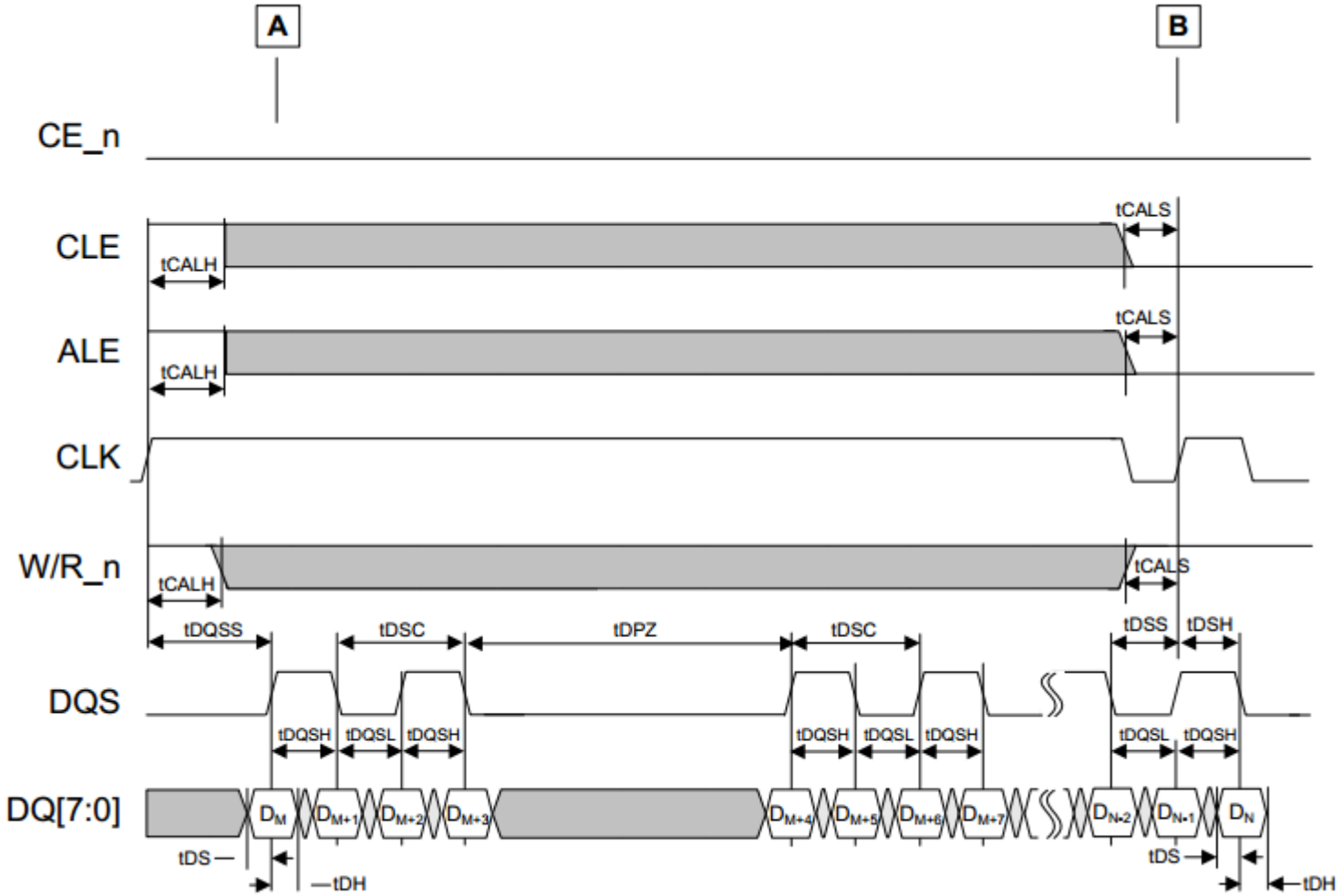


Figure 64　　Data input cycle timing, CLK stopped with data pause

4.19.2.5　数据输出周期时序

数据输出周期时序描述了数据从 device 传输到 host 的时序(例如，数据读)。host 不得开始数据输出(如，ALE/CLE 转变为 11b)，直到经过 tDQSD 时间。

对于 Read ID、Get Features、Read Status 以及 Read Status Enhanced 命令，相同的数据 byte 会重复两次，数据模式(pattern)为 D_0 D_0 D_1 D_1 D_2 D_2 等。host 只能锁存每个数据 byte 的一个备份(copy)。

tCKWR 是一个计算得出的参数，表示 W/R_n 可能从 0 变为 1 的时间。这个参数计算如下：

- tCKWR(min) = RoundUp{[tDQSCK(max) + tCK]/tCK}

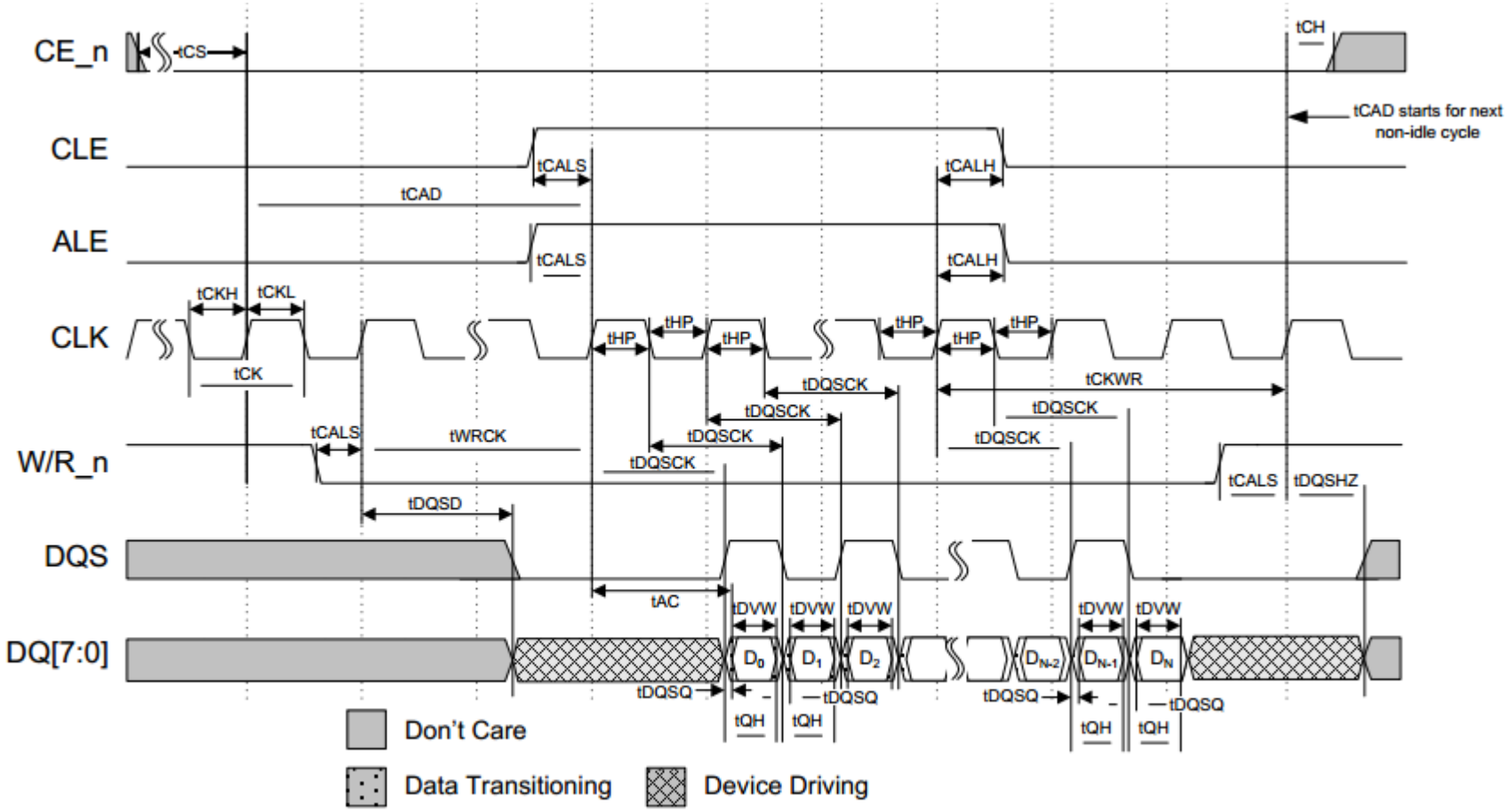


Figure 65　　Data output cycle timing

4.19.2.6　W/R_n 行为时序

图 66 描述了 DQ 总线和 DQS 信号的控制权的转换。当 W/R_n 为 1 时 host 拥有 DQ 总线和 DQS 信号的控制权。当 W/R_n 为 0 时 device 拥有 DQ 总线和 DQS 信号的控制权。当 W/R_n 为 0 时，host 应将 DQ 总线和 DQS 信号驱动为三态。

当 W/R_n 从 1 变为 0 时，总线被 device 控制，host 应将 DQ 总线和 DQS 信号驱动为三态，device 在 tDQSD 时间内应将 DQS 信号驱动为低；当 W/R_n 从 0 变为 1 时，总线被 host 控制，device 应在 tDQSHZ 时间内将 DQ 总线和 DQS 信号驱动为三态。在 idle 状态期间，当没有数据操作且 W/R_n 为 1 时，host 应将 DQS 和 DQ 总线驱动为高。每当 W/R_n 改变值时，会有一个 turn-around 时间，此时 DQS 信号为三态(host 和 device 都不驱动信号)，见 4.19.2.6。

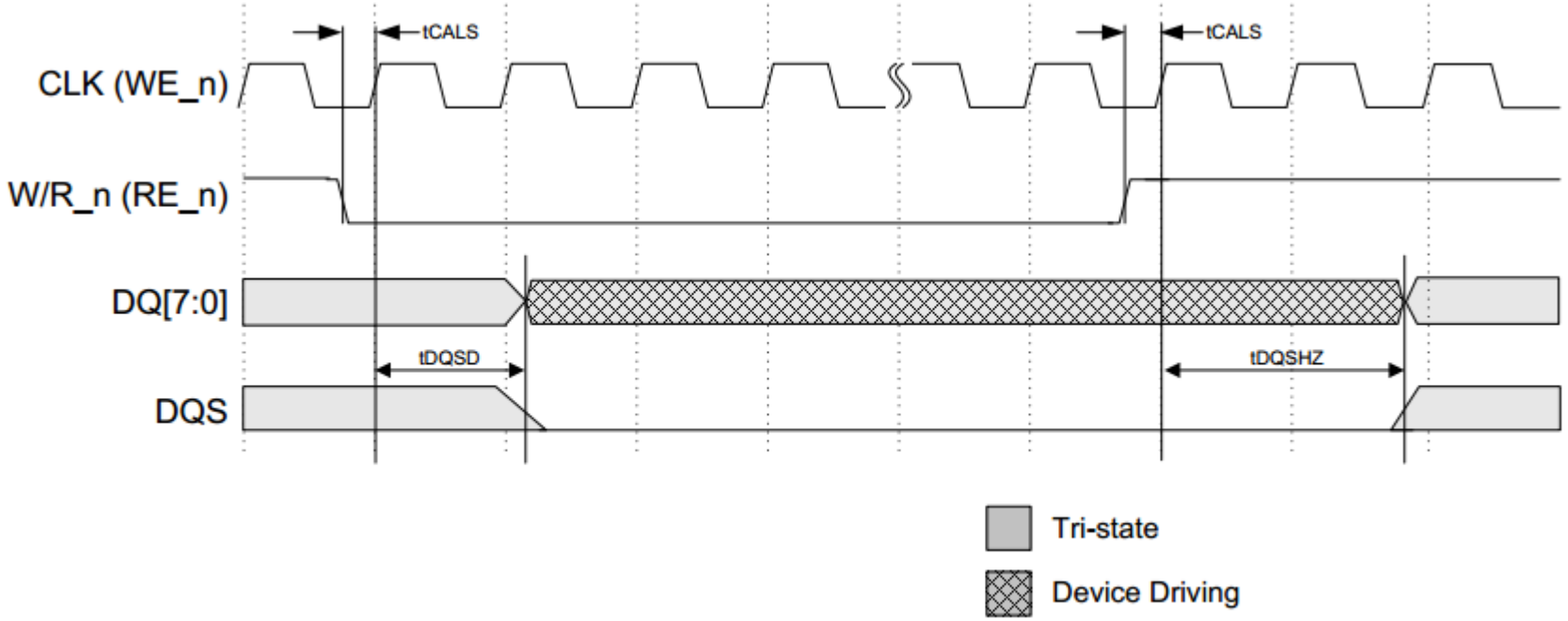


Figure 66 W/R_n timing

4.19.2.7 满足时序要求

某些情况下，在下一个命令操作之前，有多个时序参数应该被满足。例如，在 Change Write Column 命令的数据输出开始之前，tDQSD 和 tCCS 都应该被满足。host 和 device 应保证所有的时序参数都要被满足。如果 tADL、tCSS、tRHW 或 tWHR 被要求，则这些参数是控制参数(governing parameters)(即，这些参数都是最长时间值)。

图 67 和图 68 展现了一个 Read Status 命令的例子，该例子包含了命令和数据操作的所有时序参数。可以看到 tWHR 是数据传输开始之前的控制参数(governing parameter)。图中可以看到，Read Status 命令的数据中，相同的数据 byte 被传输了两次(DO,DO)。

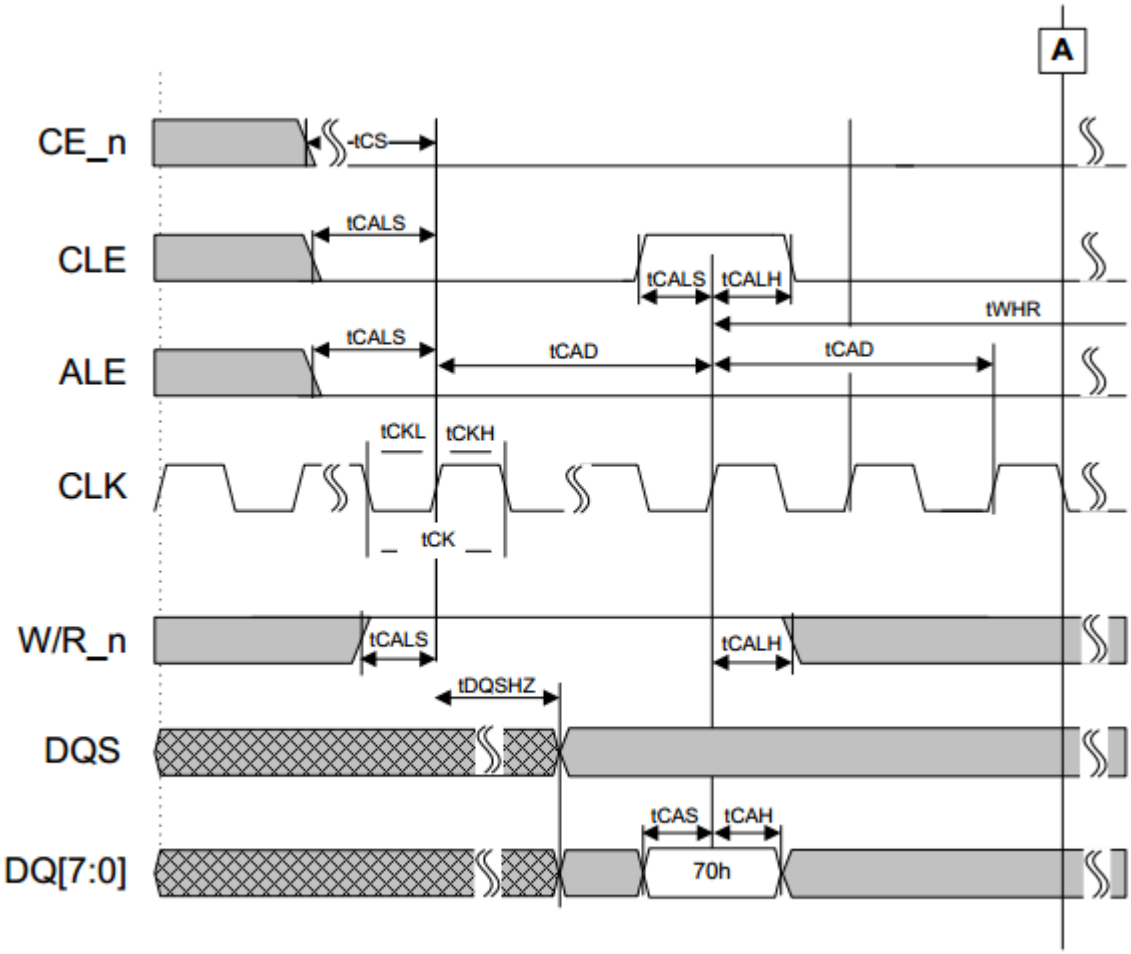


Figure 67　　Read Status including tWHR and tCAD timing requirements

注意：

1. 当 CE_n 从 1 变为 0 时，ALE 和 CLE 应处于有效状态。图中，这个有效状态等效于 tCS 和 tCALS 表示的时间。但是 tCS 和 tCALS 时间是不同的，时序参数值定义在表 85 中。

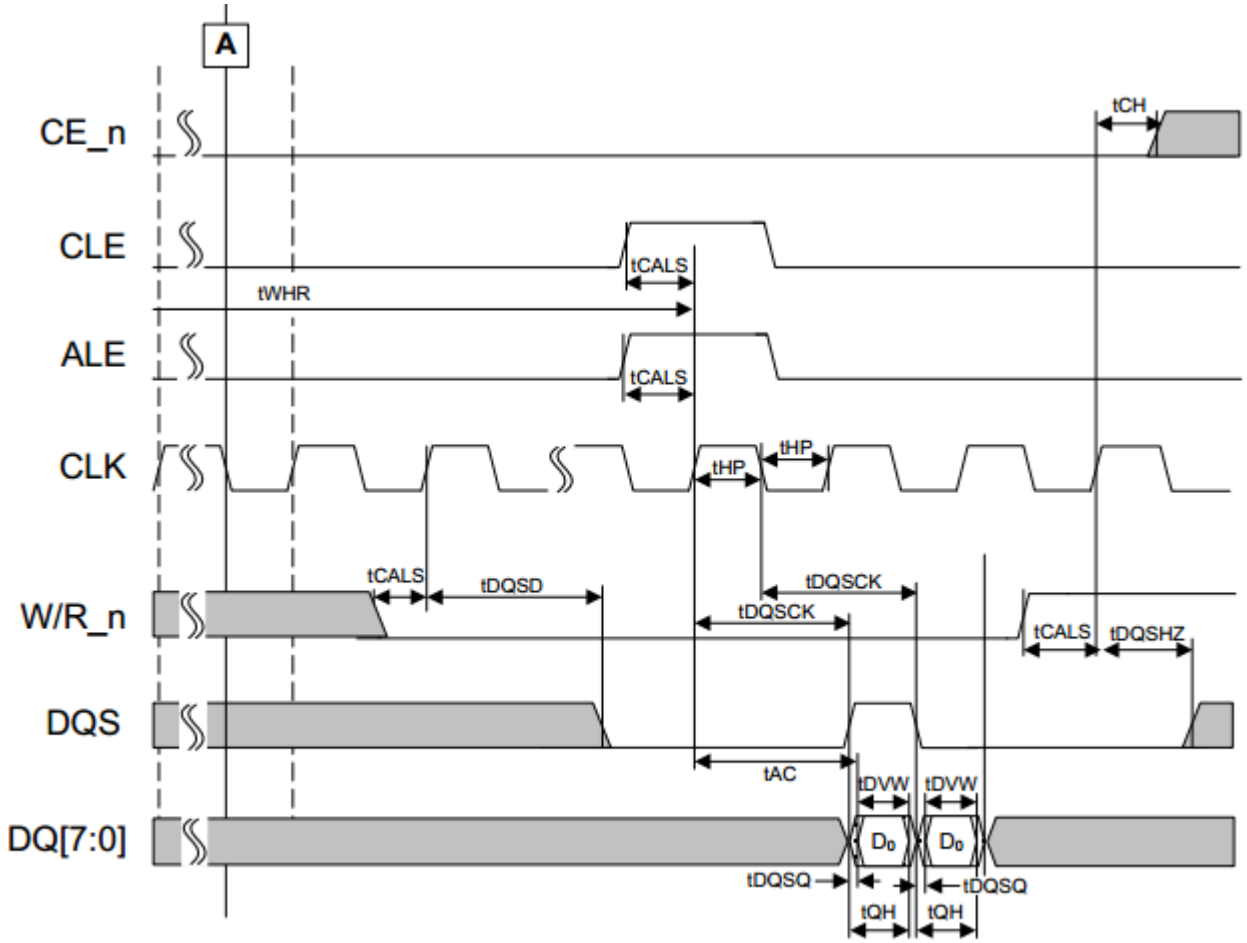


Figure 68　　Read Status including tWHR and tCAD timing requirements, continued

4.19.3　NV-DDR2 和 NV-DDR3

NV-DDR2 和 NV-DDR3 时序图中展示了差分(互补)信号的使用(RE_t/RE_c 和 DQS_t/DQS_c)。差分信号是可选的。RE_n 和 RE_t 是同一个信号;当差分信号被 disable 时不使用 RE_c。DQS 和 DQS_t 是同一个信号;当差分信号被 disable 时不使用 DQS_c。

4.19.3.1　命令周期时序

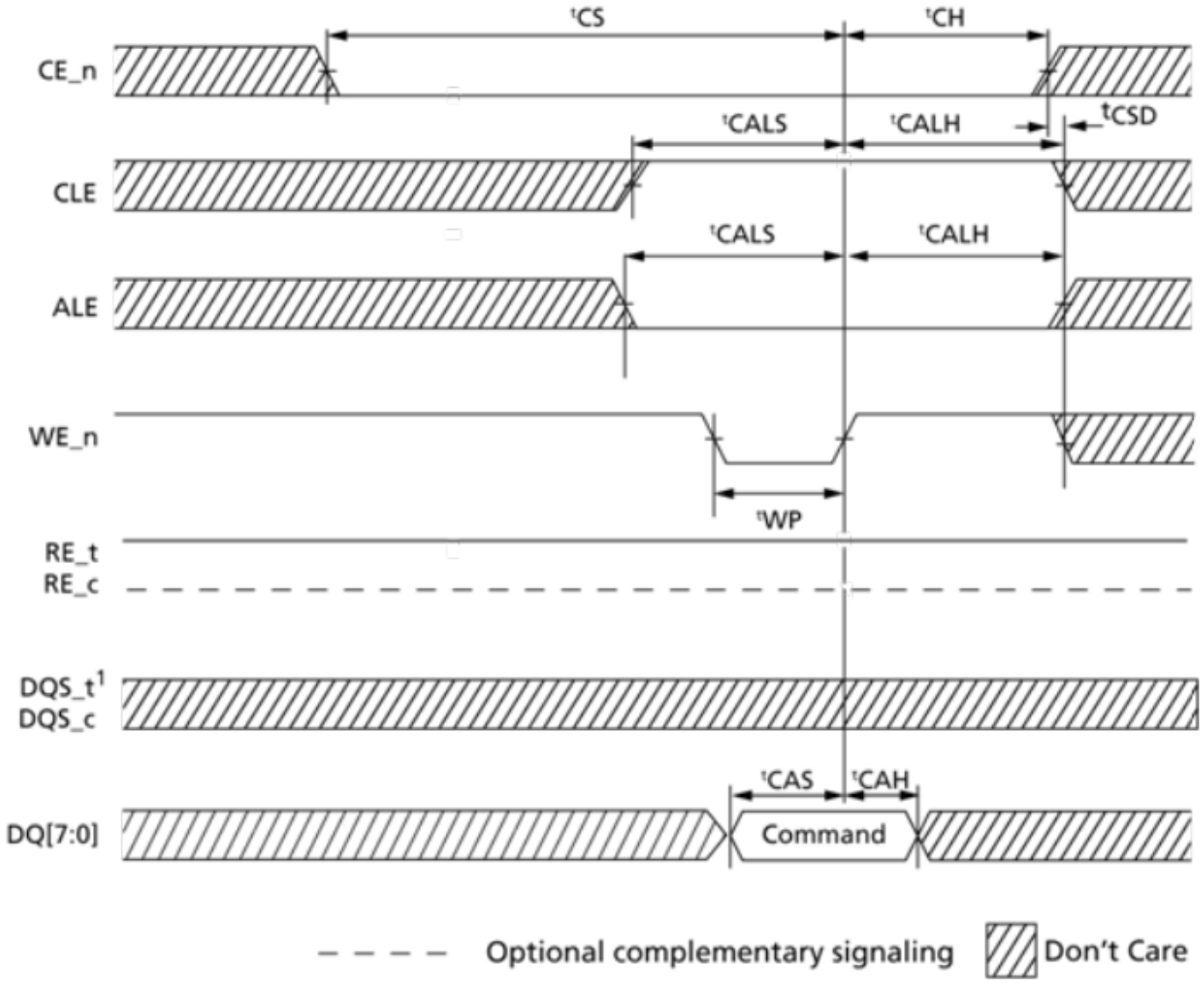


Figure 69 Command cycle timings

注意：当总线状态不是数据输入或数据输出时，如果 ALE，CLE 和 CE_n 都是低(如 idle 状态)，则 host 应保持 DQS(DQS_t)为高，以防止 device 使能 ODT。如果 ODT 被 disable，则在 idle 状态期间不用关心 DQS。

4.19.3.2 地址周期时序

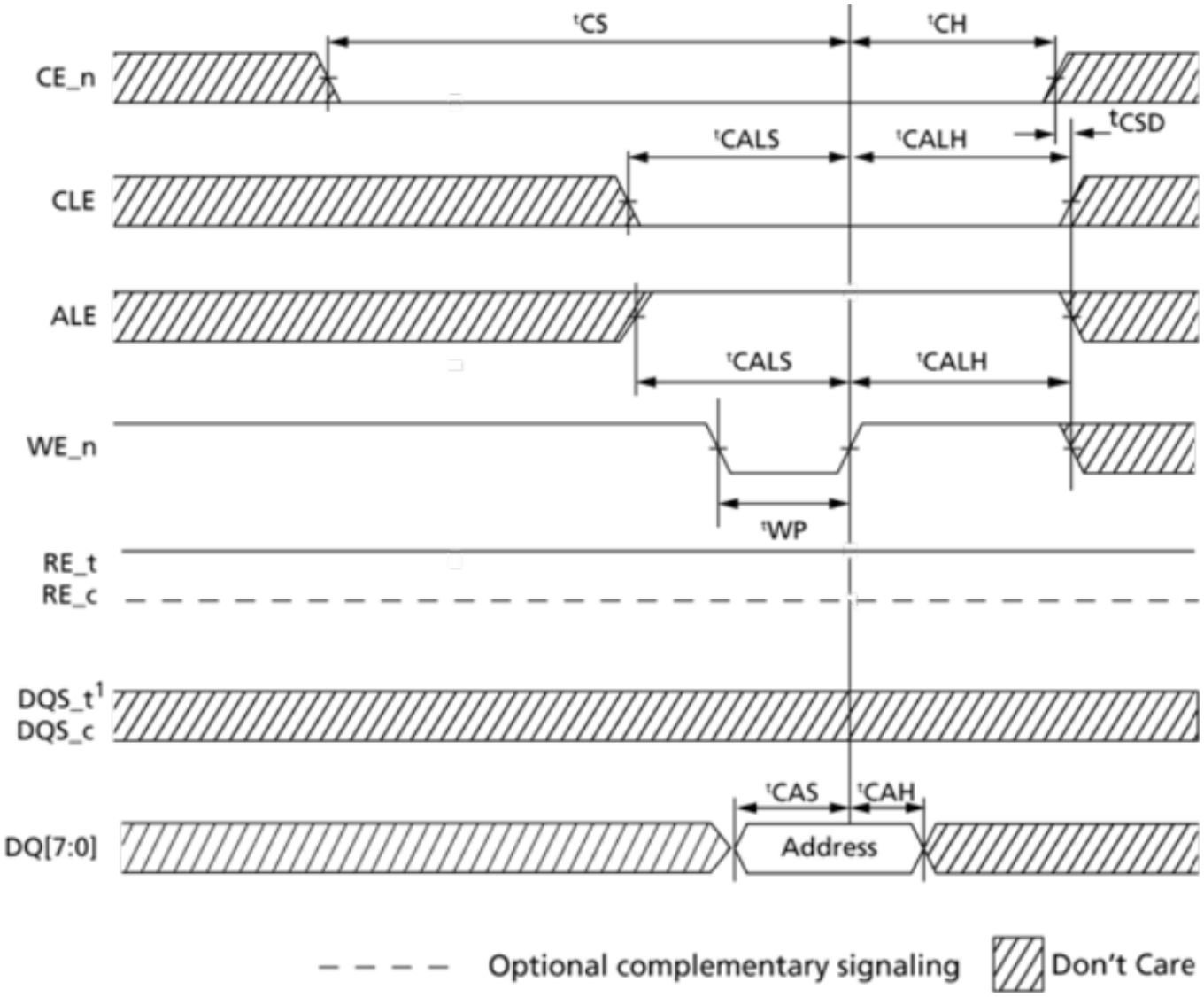


Figure 70　　Address cycle timings

注意：当总线状态不是数据输入或数据输出时，如果 ALE、CLE 和 CE_n 都为低(如，idle 状态)，则 host 应保持 DQS(DQS_t)为高，以防止 device 使能 ODT。如果 ODT 被 disable，则在 idle 状态期间不用关心 DQS。

4.19.3.3　数据输入周期时序

数据输入周期时序描述了数据从 host 传输到 device(如，数据写)的时序。通过停止 DQS(DQS_t/DQS_c)转变来暂停数据输入。

对于 Set Feature 和 ODT 配置命令能够，相同的数据 byte 被重复两次，这种情况下，数据模式(pattern)为 D0 D0 D1 D1 D2 D2 等。device 只能锁存每个数据 byte 的一个备份(copy)。

数据输入不要求使用 ODT。如果通过 Set Feature 选择使用 ODT，则 ODT 在图 71 所示的位置之间被使能和 disable。

注意：

1. tDBS 参考 CLE、ALE 或 CE_n 的最后一个下降沿。
2. host 设置 CE_n、ALE 或 CLE 为 1 来退出数据突发(data burst)。

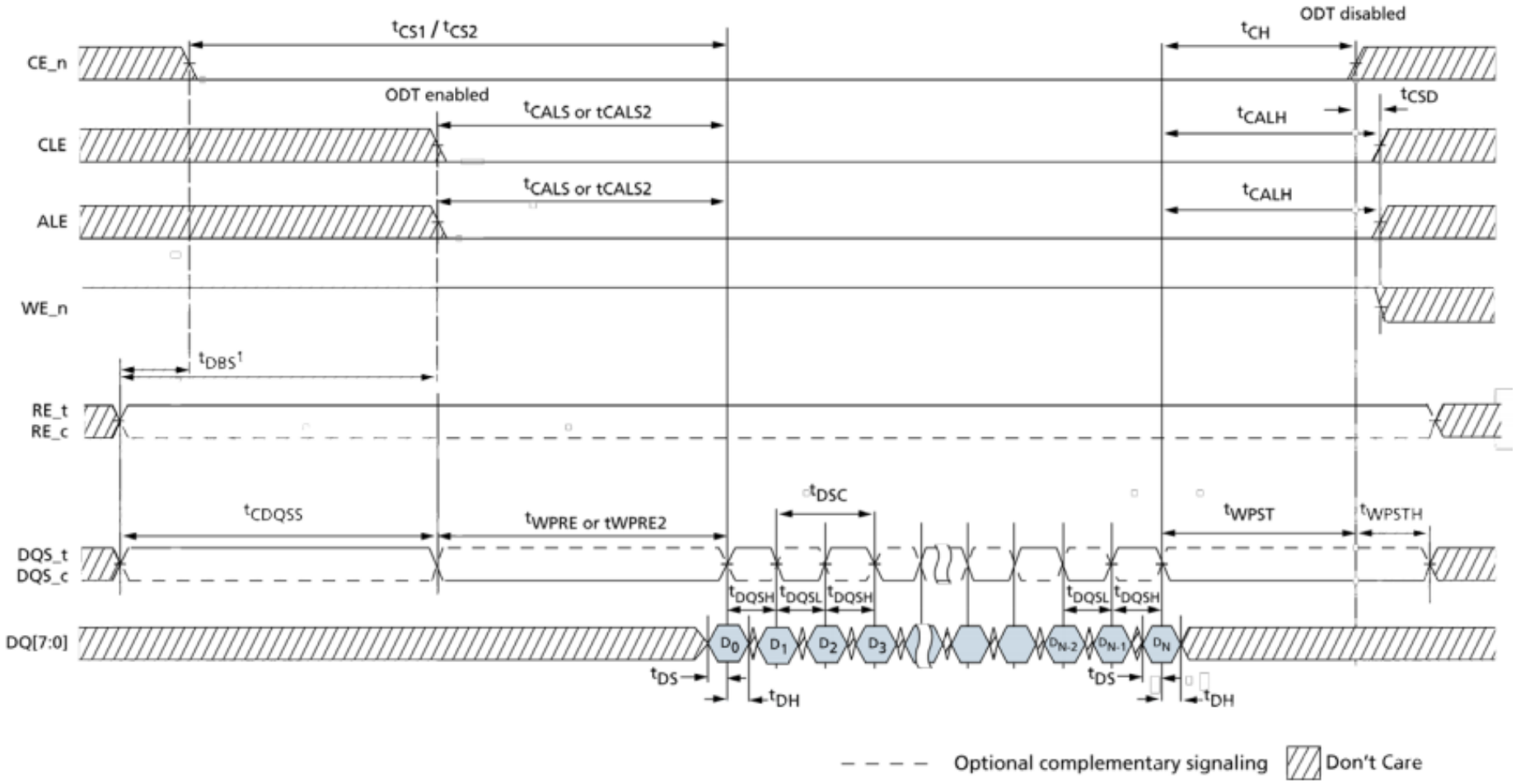


Figure 71　　Data input cycle timing

4.19.3.4　数据输出周期时序

数据输出周期时序描述了数据从 device 传输到 host(如，数据读)的时序。通过停止 RE_n(RE_t/RE_c)的转变可以暂停数据传输。

对于 Read ID，Get Feature，Read Status 和 Read Status Enhanced 命令，相同的数据 byte 被重复两次。这种情况下，数据模式(pattern)为 D_0 D_0 D_1 D_1 D_2 D_2等。host 只能锁存每个数据 byte 的一个备份(copy)。

数据输出不要求使用 ODT。如果通过 Set Feature 选择了使用 ODT，则 ODT 在图 72 所示的位置期间被使能和 disable。

注意：　1. tDBS 参考 CLE、ALE 或 CE_n 的最后一个下降沿。

　　2. host 设置 CE_n、ALE 或 CLE 为 1 来退出数据突发。仅当使用 CE_n 时，tCHZ 才被使用，该参数用来结束数据突发。

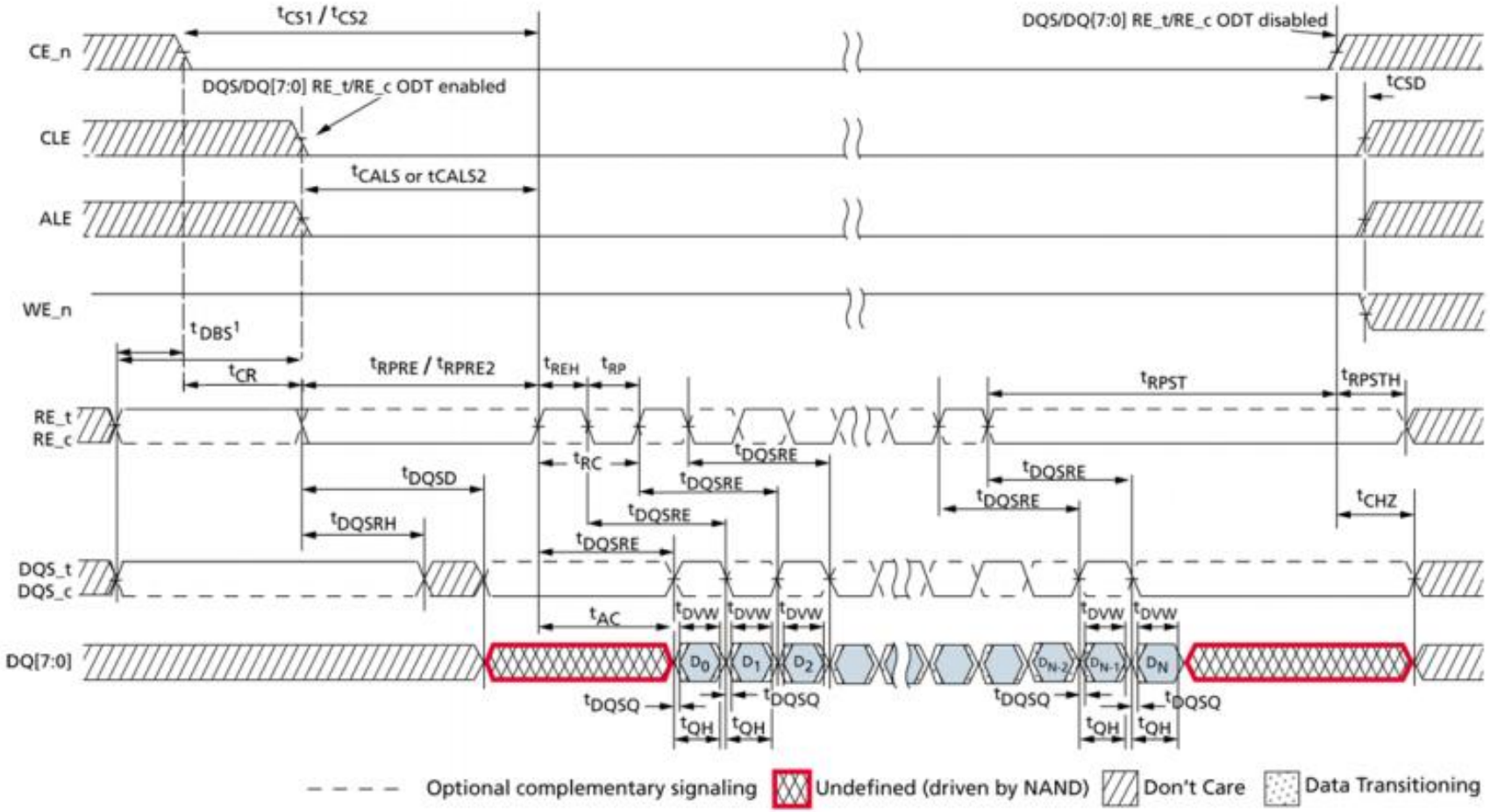


Figure 72 Data output cycle timing

5. 命令定义

5.1 命令设置

表 90 展示了 ONFI 命令设置。

第一个命令周期规定的值表示要执行的命令。有些命令有第二个命令周期，如表 90 所示。具有第二个命令周期的命令一般包含地址。

Command	O/M	1st Cycle	2nd Cycle	Acceptable while Accessed LUN is Busy	Acceptable while Other LUNs are Busy	Target level commands
Read	M	00h	30h		Y	
Multi-plane	O	00h	32h		Y	
Copyback Read	O	00h	35h		Y	
Change Read Column	M	05h	E0h		Y	
Change Read Column Enhanced	O	06h	E0h		Y	
Read Cache Random	O	00h	31h		Y	
Read Cache Sequential	O	31h			Y	
Read Cache End	O	3Fh			Y	
Block Erase	M	60h	D0h		Y	
Multi-plane	O	60h	D1h		Y	
Read Status	M	70h		Y	Y	
Read Status Enhanced	O	78h		Y	Y	
Page Program	M	80h	10h		Y	
Multi-plane	O	80h	11h		Y	
Page Cache Program	O	80h	15h		Y	
Copyback Program	O	85h	10h		Y	
Multi-plane	O	85h	11h		Y	
Small Data Move[2]	O	85h	11h		Y	
Change Write Column[1]	M	85h			Y	
Change Row Address[1]	O	85h			Y	
Read ID	M	90h				Y
Volume Select[3]	O	E1h		Y	Y	
ODT Configure[3]	O	E2h				Y
Read Parameter Page	M	ECh				Y
Read Unique ID	O	EDh				Y
Get Features	O	EEh				Y
Set Features	O	EFh				Y
LUN Get Features	O	D4h			Y	
LUN Set Features	O	D5h			Y	
ZQ Calibration Short	O	D9h			Y	
ZQ Calibration Long	O	F9h			Y	
Reset LUN	O	FAh		Y	Y	
Synchronous Reset	O	FCh		Y	Y	Y
Reset	M	FFh		Y	Y	Y

注意：

1. Change Write Column 只能确定列地址。Change Row Address 可确定行地址和列地址。参见具体命令定义。
2. 如果操作只是编程并且没有数据输出，则 Small Data Move 的第一个操作码(opcode)可以是 80h。Small Data Move 的倒数第二个周期是命令 10h，用来确认编程或回拷(Copyback)操作。
3. 如果 device 即支持 CE_n 引脚 reduction，又支持 matrix termination，则应支持 Volume Select。如果 device 支持 matrix termination，则应支持 ODT 配置(ODT Configure)。

表 90 命令设置

device 不能使用保留的操作码，因为这些保留的操作码可能用于 ONFI 未来版本的规范中。制造商定义操作码不能在 ONFI 中被定义使用。

Type	Opcode
Standard Command Set	00h, 05h – 06h, 10h – 11h, 15h, 30h – 32h, 35h, 3Fh, 60h, 70h, 78h, 80h – 81h, 85h, 90h, D0h – D1h, D4h – D5h, D9h, E0h – E2h, ECh – EFh, F1h – F2h, F9h, FAh, FCh, FFh
Vendor Specific	01h – 04h, 07h – 0Ah, 0Ch – 0Fh, 13h, 16h – 17h, 19h – 1Ah, 1Dh – 2Fh, 33h – 34h, 36h – 3Eh, 40h – 5Fh, 61h, 65h – 6Fh, 71h – 75h, 77h, 79h – 7Fh, 84h, 87h – 8Dh, 8Fh, 91h – CFh, D2h – D3h, D6h – D8h, DAh – DFh, E3h – EBh, F0h, F3h – F8h, FBh, FD – FEh
Reserved	0Bh, 12h, 14h, 18h, 1Bh – 1Ch, 62h – 64h, 76h, 82h – 83h, 86h, 8Eh

Table 91 Opcode Reservations

5.2 命令描述

第 5 章中的命令描述以一种忽略所使用的接口类型(当命令可被用于任何接口中时)的方式来呈现。图 73 为一个例子，展现了以该方式描述的 Change Write Column 命令。图 73 命令描述的例子可以被转为一个特定接口的命令描述,图 74 展示了在 SDR 接口中的 Change Write Column 命令描述。图 75 展示了在 NV-DDR 接口中的 Change Write Column 命令描述。图 76 展示了在 NV-DDR2/3 接口中 Change Write Column 命令描述。注意，第 4 章中定义的时序参数可以在每个命令的描述图中标示出来(例如，NV-DDR 接口中的 tCAD 时序参数)。

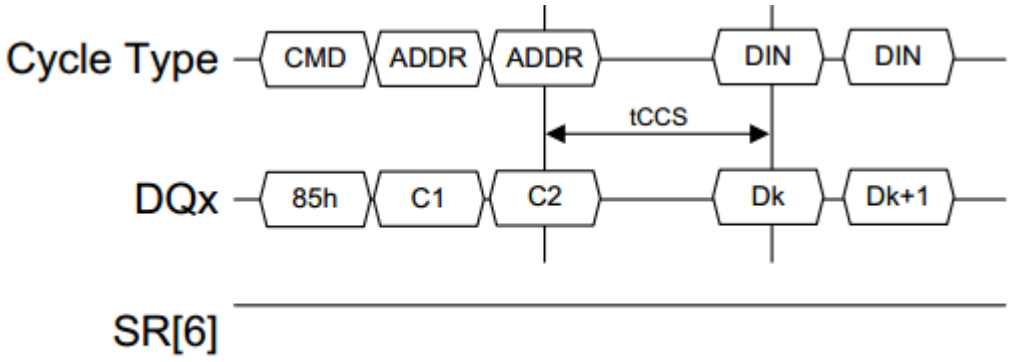


Figure 73 Agnostic command description

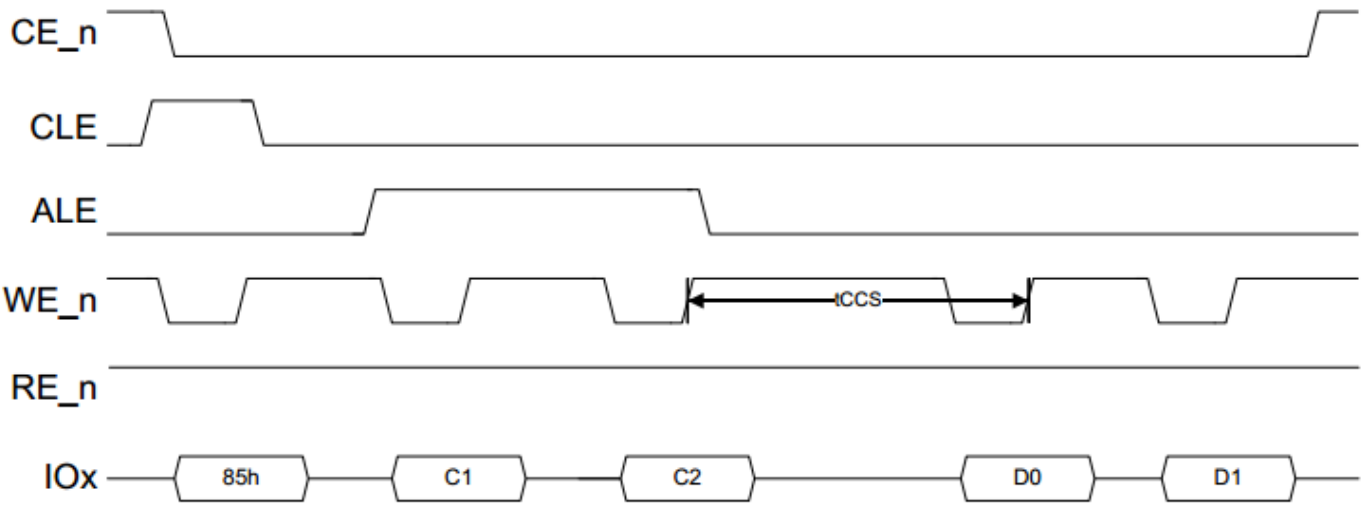


Figure 74 SDR data interface command description

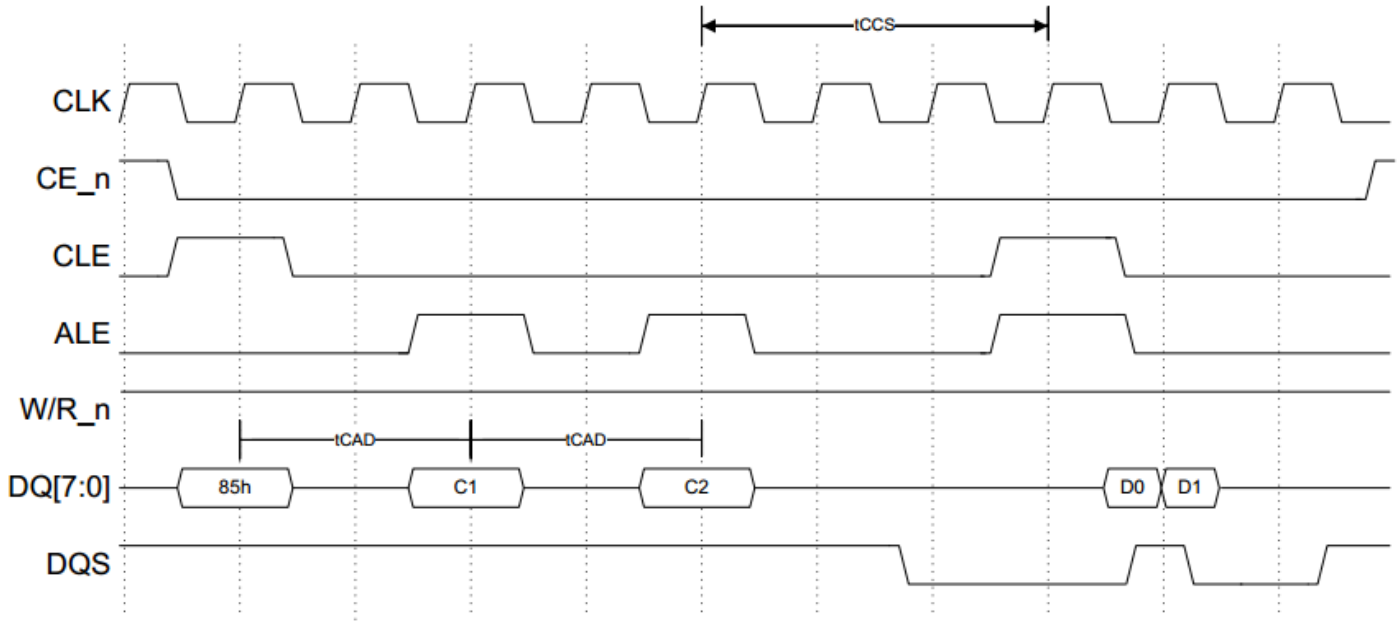


Figure 75 NV-DDR data interface command description

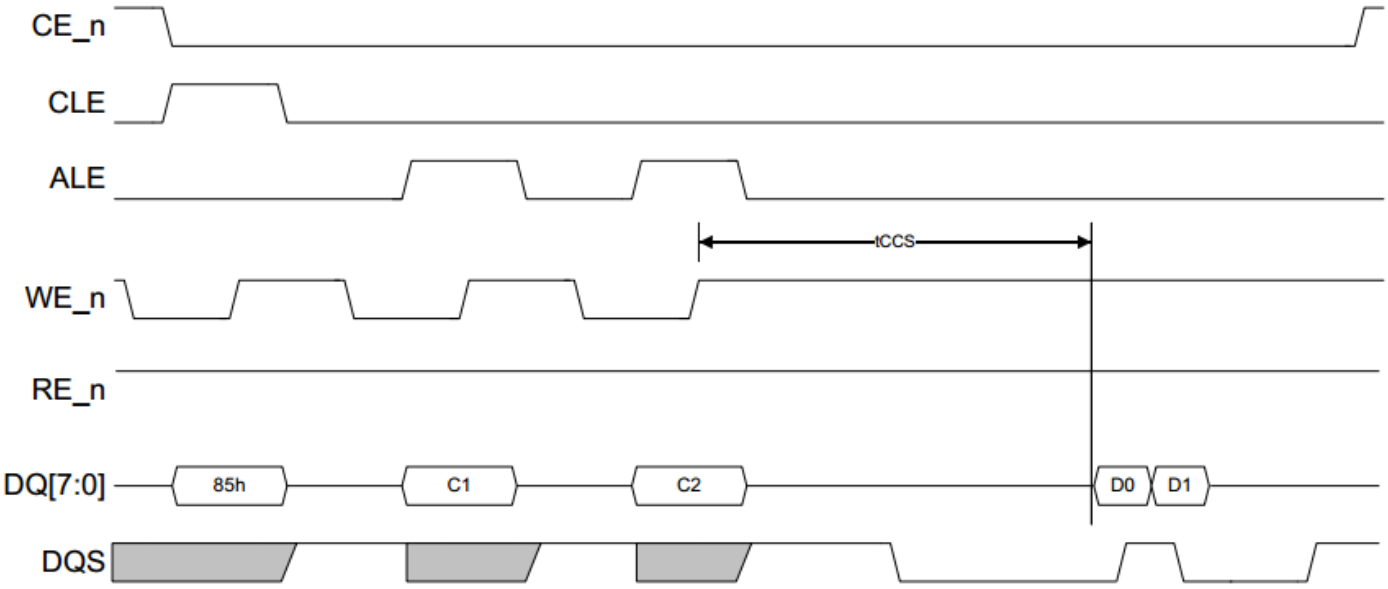


Figure 76 NV-DDR2 and NV-DDR3 data interface command description

注意：当总线状态不是数据输入或数据输出时，如果 ALE、CLE 和 CE_n 都为低(如，idle 状态)，则 host 应保持 DQS(DQS_t)为高，以防止 device 使能 ODT。如果 ODT 被 disable，则 idle 状态期间不用关心 DQS。

5.3 Reset 描述

Reset 功能将对象置为默认的上电状态，如果对象被配置成 NV-DDR 或 NV-DDR2 接口模式，则 reset 之后对象的接口模式会变为 SDR 接口。Reset 之后，R/B_n 的值是未知的；Reset 之后经过 tWB 时间后，R/B_n 被置为低。

注意，Reset 前后，某些特性设置(feature settings)会保持不变(5.30 中描述)。所有 LUN 都回被 Reset 命令复位。Reset 命令可在对象的任何状态执行，除了上电期间，R/B_n 变为 1 之后才能发送 Reset。图 77 定义了 Reset 的行为和时序。

当对象被配置为 NV-DDR3 接口时发送了 Reset，对象应维持 NV-DDR3 接口模式。

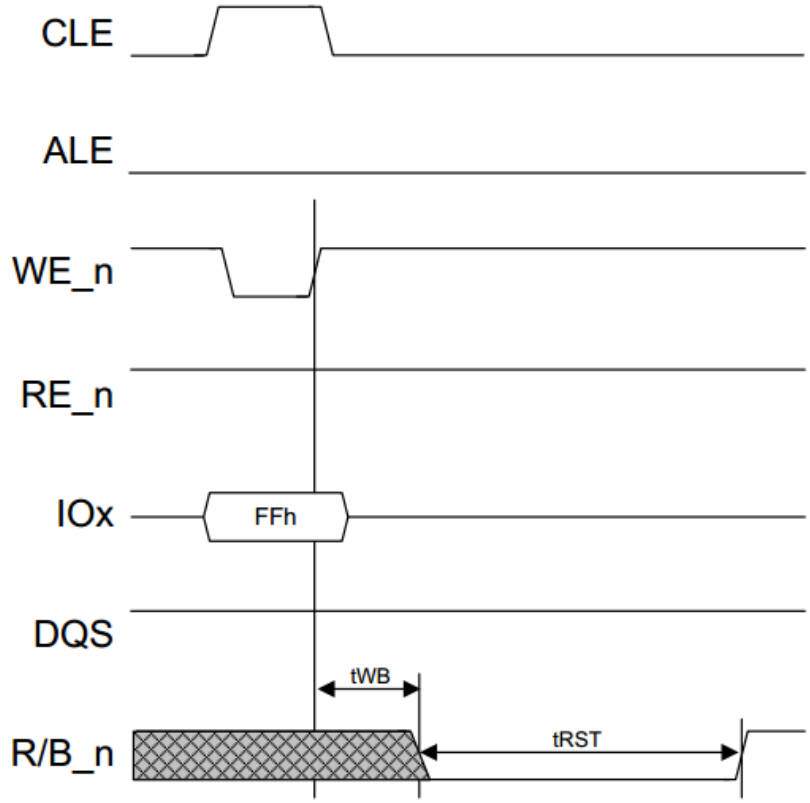


Figure 77　　Reset timing diagram

5.4 同步 Reset 定义

同步 Reset 命令会复位对象及所有的 LUN。该命令可以在对象的任何状态发送。图 78 定义了同步 Reset 命令的行为和时序。当同步 Reset 发送后，R/B_n 的值是未知的；同步 Reset 发送后经过 tWB 时间，R/B_n 被置为低。

支持 NV-DDR、NV-DDR2 或 NV-DDR3 接口的 device 都应支持同步 Reset 命令。该命令仅当使用 NV-DDR、NV-DDR2 或 NV-DDR3 接口时才会被接受。当 device 被配置为 SDR 接口时 host 不能发送该命令。该命令之后，对象应维持在 NV-DDR、NV-DDR2 或 NV-DDR3 接口模式。

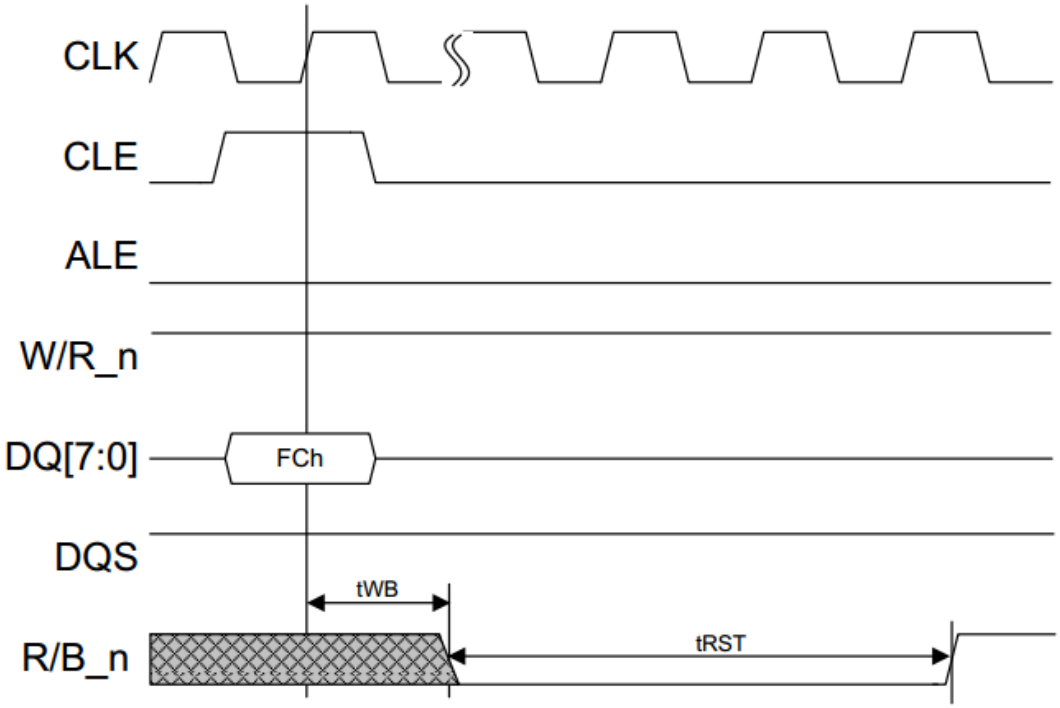


Figure 78　　Synchronous Reset timing diagram

5.5 Reset LUN 定义

Reset LUN 命令用于复位一个特定的 LUN。该命令只能被命令中寻址的 LUN 接受。命令可在 LUN 的任何状态下执行。图 79 定义了 Reset LUN 命令的行为和时序。当 Reset LUN 命令发送后，SR[6]的值是未知的；Reset LUN 命令发送后经过 tWB 时间之后，SR[6]被置为低。该命令不会影响对象的接口设置。

Reset LUN 可用来取消正在执行的命令操作。当对象出现问题时，例如，挂起状态，应使用 Reset(FFh)或同步 Reset(FCh)命令。

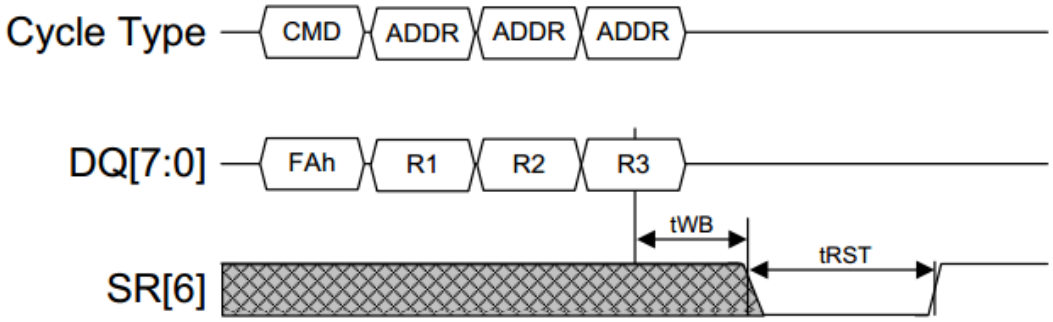


Figure 79 Reset LUN timing diagram

5.6 Read ID 定义

Read ID 命令用来识别支持 ONFI 规范的对象。如果对象支持 ONFI 规范，则应返回 ONFI 签名。ONFI 签名是一个 ASCII 编码的‘ONFI’，其中‘O’=4Fh，‘N’=4Eh，‘F’=46h，‘I’=49h。对于 ONFI 4.0 版本之前的 device，读取超过 4 bytes 会读出不确定的值。图 80 定义了 Read ID 命令的行为和时序。

ONFI4.0 中新定义了 Read ID 命令的第 5 和第 6 byte。第 5 byte 用来识别 device 是否在 NV-DDR3 接口中上电。

在 NV-DDR、NV-DDR2 或 NV-DDR3 接口中发送 Read ID 命令时，每个数据 byte 会被接收两次。host 只锁存每个数据 byte 的一个备份(copy)，见 4.4 章。

对于 Read ID 命令，只有地址 00h 和 20h 是有效的。为了检查 ONFI 签名，应进入地址 20h(例如，进入地址 00h 并读取 36 bytes 来获得 ONFI 签名是非法的)。

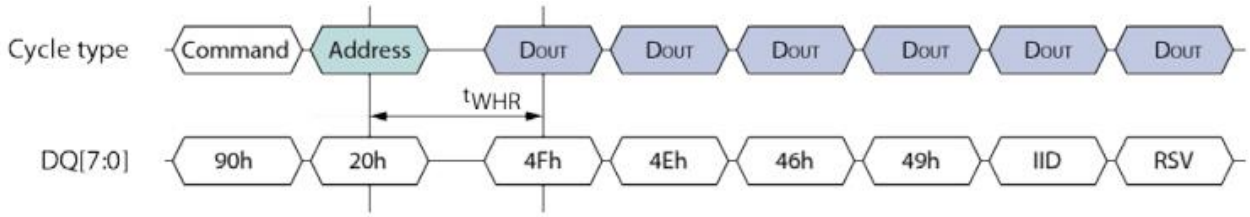


Figure 80 Read ID timing diagram for ONFI signature

IID	Power on Interface ID, 01h for NV-DDR3, 00h for SDR
RSV	6th byte is reserved for future use

上电接口 ID 在每个电源周期仅被设置一次并且不会再发生变化。如果 device 在 SDR 或 NV-DDR3 接口执行 Reset(FFh)，那么 Reset 之后发送 Set Feature(EFh)命令来改变接口模式时并不会改变 IID 值。

Read ID 命令也可以通过地址 00h 来识别特定 NAND 的 JEDEC 制造商 ID 和 device ID。图 81 定义了 Read ID 检索 JEDEC 制造商 ID 和 device ID 的行为和时序。读取超过第一个 2 bytes 会读出制造商定义的值。

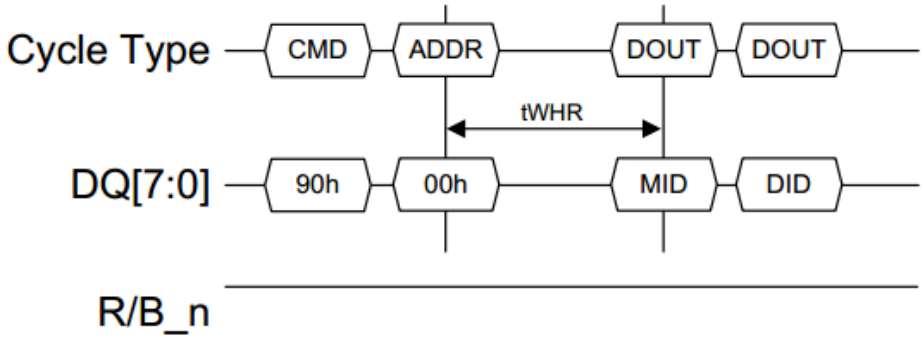


Figure 81 Read ID timing diagram for manufacturer ID

MID	Manufacturer ID for manufacturer of the part, assigned by JEDEC.
DID	Device ID for the part, assigned by the manufacturer.

Read ID 命令可以使用 SDR、NV-DDR、NV-DDR2 或者 NV-DDR3 接口来发送。每种接口的时序参数如图 82，图 83 和图 84 所示。

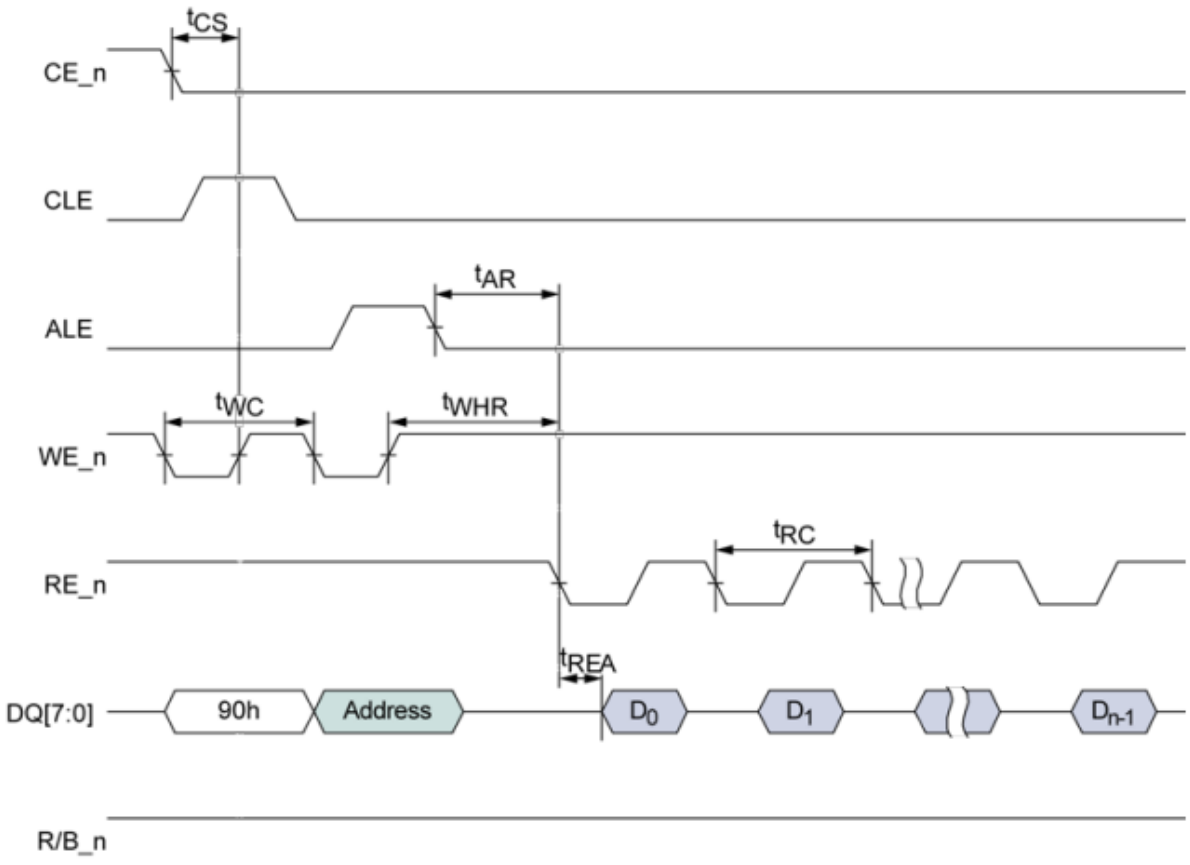


Figure 82 Read ID command using SDR data interface

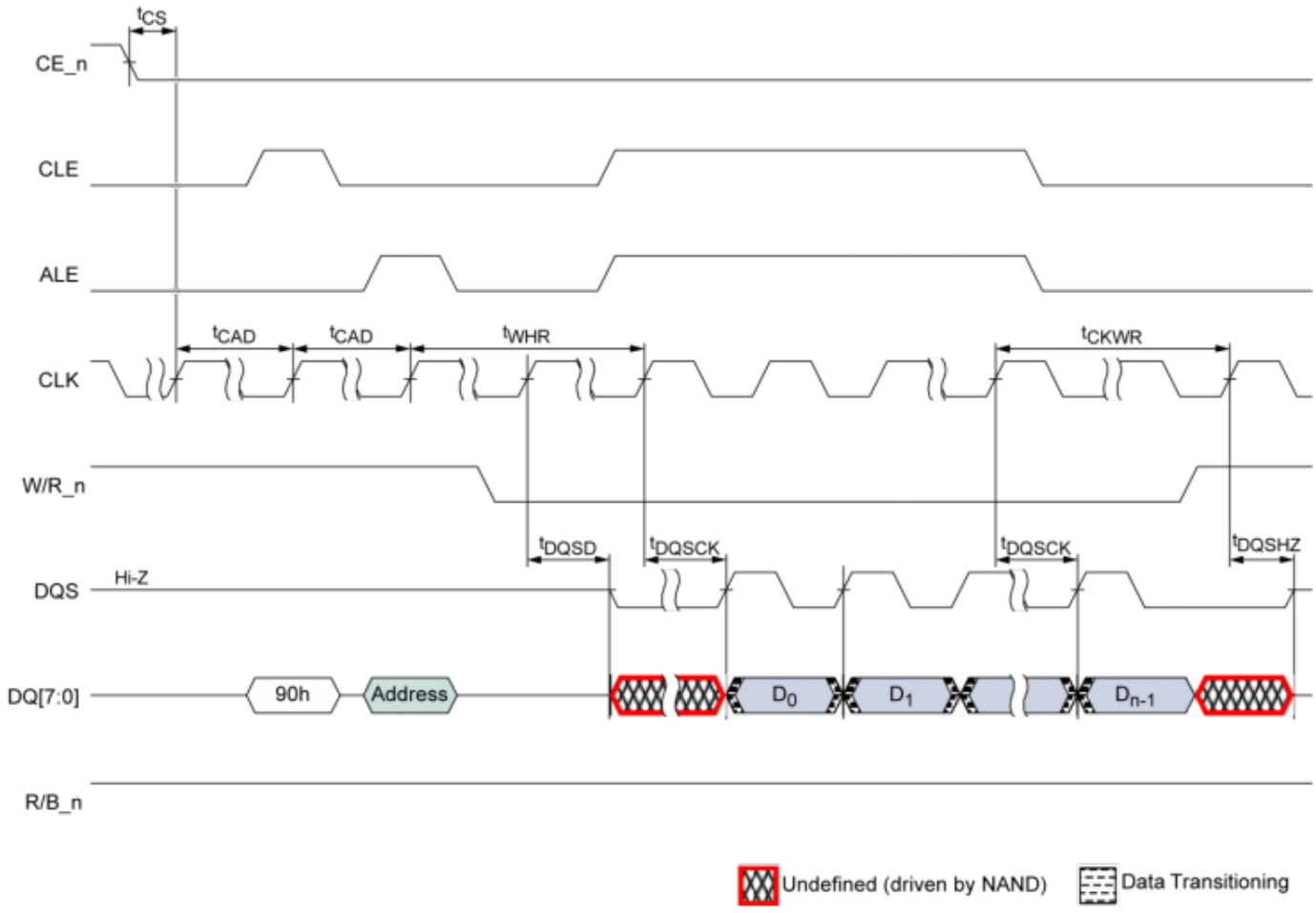


Figure 83 Read ID command using NV-DDR data interface

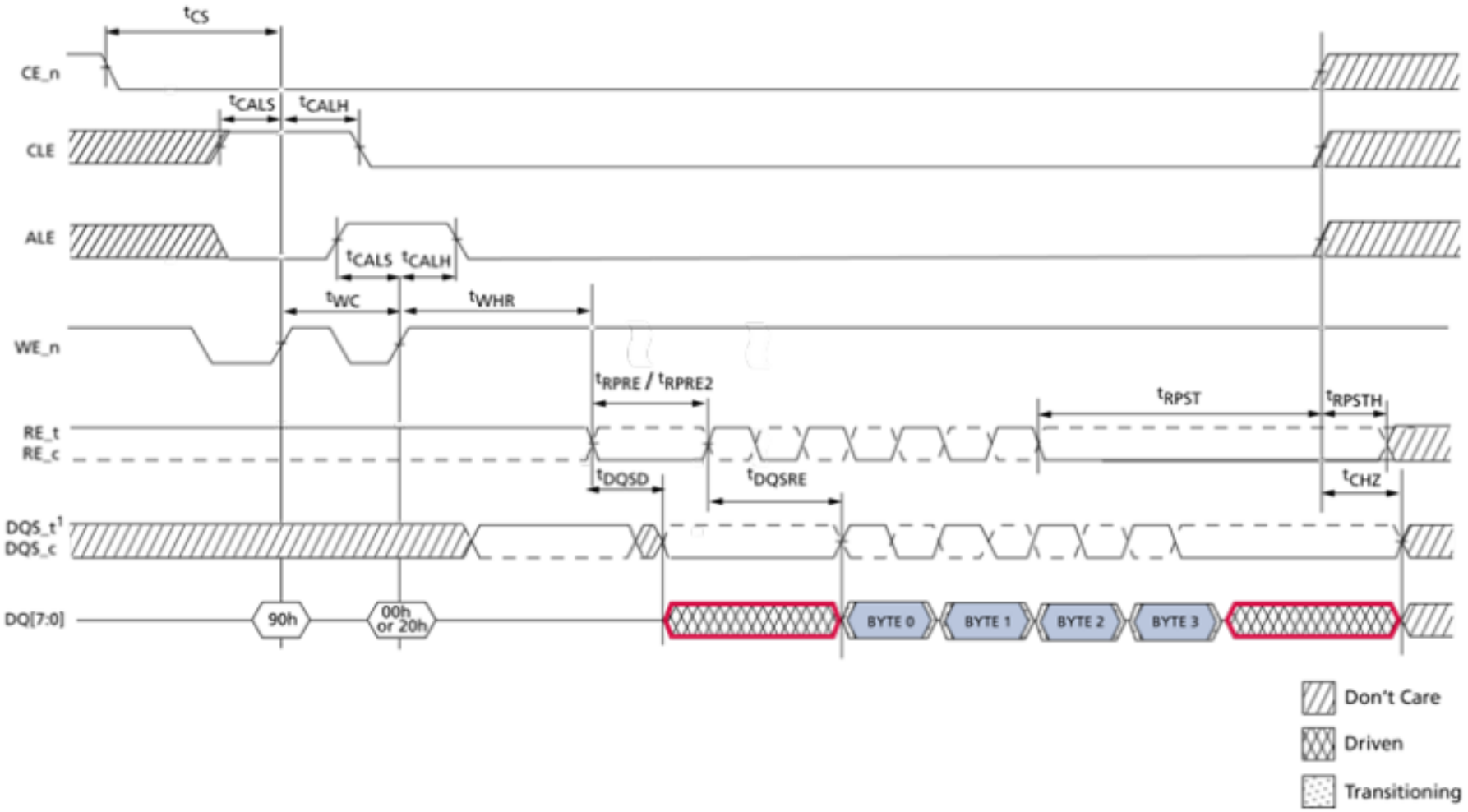


Figure 84 Read ID command using NV-DDR2 or NV-DDR3 data interface

注 1：图 83 和图 84 中数据 byte 重复了两次(DQS 的上升沿和下降沿)。图 84 中，当总线状态不是数据输入和数据输出时，如果 ALE、CLE 和 CE_n 都为低(如，idle 状态)，则 host 应保持 DQS(DQS_t)为高，以防止 device 使能 ODT。如果 ODT 被 disable，则在 idle 状态期间不用关心 DQS。

5.7 读参数 page(Read Parameter Page)定义

读参数 page 命令用于检索数据结构，该数据结构描述了对象的结构、特性、时序以及其他行为参数。有时也会有一个扩展的参数 page 用以描述额外的信息。图 85 定义了 Read Parameter Page 的行为。

参数 page 中的数值都是固定不变的。在电源管理事件后(power management events)不要求 host 读参数 page。

上电后，host 第一次执行 Read Parameter Page 命令应使用时序模式 0。如果 host 检测到对象支持更高级的时序模式，则这些支持的时序模式可用于后面执行的 Read Parameter Page 命令。

在执行完 Read Parameter Page 命令后，可以发送 Change Read Column 命令来读取参数 page 的指定部分。

Read Status 命令可以在 Read Parameter Page 命令执行期间用来检查其状态。在 Read Status 命令完成后，host 应在命令通道上发送 00h 来继续 Read Parameter Page 命令的数据输出流程。

执行 Read Parameter Page 命令期间不能使用 Read Status Enhanced 和 Change Read Column Enhanced 命令。

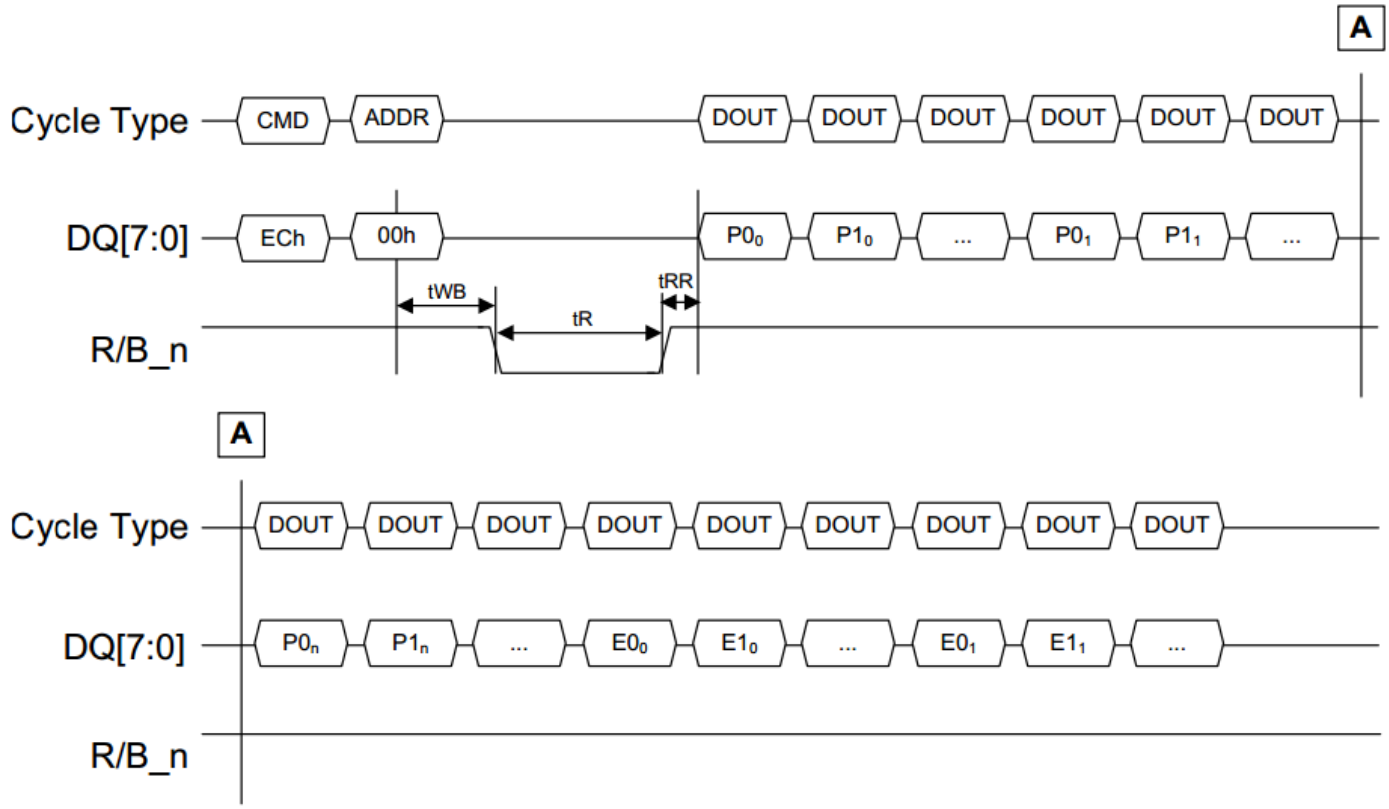


Figure 85　　Read Parameter Page command timing

P0k-Pnk　参数 page 数据结构的第 k 个备份(copy)。参见 5.7.1。读取超过最后一个参数 page 备份末尾的 byte（或者超过最后一个扩展参数 page 备份——如果支持的话）会返回未知数。

E0k-Enk　扩展参数 page 数据结构的第 k 个备份，参见 5.7.2。读取超过最后一个扩展参数 page 备份末尾的 byte 会返回未知值。该字段仅当支持扩展参数 page 时才会出现，是否支持扩展参数 page 在参数 page 的 Features 支持字段中标示。

5.7.1 参数 page 数据结构定义

表 92 定义了参数 page 的数据结构。对于跨多个 bytes 的参数，其最低有效 byte 对应第一个 byte，参见 1.4.2.3。

当涉及到与数据访问 size 相关的(比如，在 8-bit 数据访问的 device 中)item 时，参数 page 中的数值是以 byte 为单位。例如，对象返回 page 中有多少数据 bytes。对于一个支持 16-bit 数据访问的 device，要求 host 将 byte 值转换为 word 值来使用。

未使用的字段应该由对象清为 0。

Byte	O/M	Description
	Revision information and features block	
0-3	M	Parameter page signature Byte 0: 4Fh, “O” Byte 1: 4Eh, “N” Byte 2: 46h, “F” Byte 3: 49h, “I”
4-5	M	Revision number 10-15 Reserved (0) 9 1 = supports ONFI version 4.0 8 1 = supports ONFI version 3.2 7 1 = supports ONFI version 3.1 6 1 = supports ONFI version 3.0 5 1 = supports ONFI version 2.3 4 1 = supports ONFI version 2.2 3 1 = supports ONFI version 2.1 2 1 = supports ONFI version 2.0 1 1 = supports ONFI version 1.0 0 Reserved (0)
6-7	M	Features supported 15 1 = supports Package Electrical Specification 14 1 = supports ZQ calibration 13 1 = supports NV-DDR3 12 1 = supports external Vpp 11 1 = supports Volume addressing 10 1 = supports NV-DDR2 9 1 = supports EZ NAND 8 1 = supports program page register clear enhancement 7 1 = supports extended parameter page 6 1 = supports multi-plane read operations 5 1 = supports NV-DDR 4 1 = supports odd to even page Copyback 3 1 = supports multi-plane program and erase operations 2 1 = supports non-sequential page programming 1 1 = supports multiple LUN operations 0 1 = supports 16-bit data bus width
8-9	M	Optional commands supported 14-15 Reserved (0)

Byte	O/M	Description
		13 1 = supports ZQ calibration (Long and Short) 12 1 = supports LUN Get and LUN Set Features 11 1 = supports ODT Configure 10 1 = supports Volume Select 9 1 = supports Reset LUN 8 1 = supports Small Data Move 7 1 = supports Change Row Address 6 1 = supports Change Read Column Enhanced 5 1 = supports Read Unique ID 4 1 = supports Copyback 3 1 = supports Read Status Enhanced 2 1 = supports Get Features and Set Features 1 1 = supports Read Cache commands 0 1 = supports Page Cache Program command
10	O	ONFI-JEDEC JTG primary advanced command support 4-7 Reserved (0) 3 1 = supports Multi-plane Block Erase 2 1 = supports Multi-plane Copyback Program 1 1 = supports Multi-plane Page Program 0 1 = supports Random Data Out
11		Reserved (0)
12-13	O	Extended parameter page length
14	O	Number of parameter pages
15-31		Reserved (0)
	Manufacturer information block	
32-43	M	Device manufacturer (12 ASCII characters)
44-63	M	Device model (20 ASCII characters)
64	M	JEDEC manufacturer ID
65-66	O	Date code
67-79		Reserved (0)
	Memory organization block	
80-83	M	Number of data bytes per page
84-85	M	Number of spare bytes per page
86-89		Obsolete – Number of data bytes per partial page
90-91		Obsolete – Number of spare bytes per partial page
92-95	M	Number of pages per block
96-99	M	Number of blocks per logical unit (LUN)
100	M	Number of logical units (LUNs)
101	M	Number of address cycles 4-7 Column address cycles 0-3 Row address cycles
102	M	Number of bits per cell
103-104	M	Bad blocks maximum per LUN
105-106	M	Block endurance
107	M	Guaranteed valid blocks at beginning of target
108-109	M	Block endurance for guaranteed valid blocks
110	M	Number of programs per page
111		Obsolete – Partial programming attributes
112	M	Number of bits ECC correctability

Byte	O/M	Description
113	M	Number of plane address bits 4-7 Reserved (0) 0-3 Number of plane address bits
114	O	Multi-plane operation attributes 6-7 Reserved (0) 5 1 = lower bit XNOR block address restriction 4 1 = read cache supported 3 Address restrictions for cache operations 2 1 = program cache supported 1 1 = no block address restrictions 0 Overlapped / concurrent multi-plane support
115	O	EZ NAND support 3-7 Reserved (0) 2 1 = Requires Copyback Adjacency 1 1 = supports Copyback for other planes & LUNs 0 1 = supports enable/disable of automatic retries
116-127		Reserved (0)
	Electrical parameters block	
128	M	I/O pin capacitance, maximum
129-130	M	SDR timing mode support 6-15 Reserved (0) 5 1 = supports timing mode 5 4 1 = supports timing mode 4 3 1 = supports timing mode 3 2 1 = supports timing mode 2 1 1 = supports timing mode 1 0 1 = supports timing mode 0, shall be 1
131-132		Obsolete – SDR program cache timing mode support
133-134	M	t_{PROG} Maximum page program time (µs)
135-136	M	t_{BERS} Maximum block erase time (µs)
137-138	M	t_R Maximum page read time (µs)
139-140	M	t_{CCS} Minimum change column setup time (ns)
141	O	NV-DDR timing mode support 7 Reserved (0) 6 Obsolete 5 1 = supports NV-DDR timing mode 5 4 1 = supports NV-DDR timing mode 4 3 1 = supports NV-DDR timing mode 3 2 1 = supports NV-DDR timing mode 2 1 1 = supports NV-DDR timing mode 1 0 1 = supports NV-DDR timing mode 0
142	O	NV-DDR2 timing mode support 7 1 = supports timing mode 7 6 1 = supports timing mode 6 5 1 = supports timing mode 5 4 1 = supports timing mode 4 3 1 = supports timing mode 3 2 1 = supports timing mode 2 1 1 = supports timing mode 1 0 1 = supports timing mode 0
143	O	NV-DDR / NV-DDR2 features 4-7 Reserved (0)

Byte	O/M	Description
		3 1 = device requires Vpp enablement sequence 2 1 = device supports CLK stopped for data input 1 1 = typical capacitance values present 0 tCAD value to use
144-145	O	CLK input pin capacitance, typical
146-147	O	I/O pin capacitance, typical
148-149	O	Input pin capacitance, typical
150	M	Input pin capacitance, maximum
151	M	Driver strength support 3-7 Reserved (0) 2 1 = supports 18 Ohm drive strength 1 1 = supports 25 Ohm drive strength 0 1 = supports driver strength settings
152-153	O	t_R Maximum multi-plane page read time (µs)
154-155	O	t_{ADL} Program page register clear enhancement tADL value (ns)
156-157	O	t_R Typical page read time for EZ NAND (µs)
158	O	NV-DDR2/3 features 6-7 Reserved (0) 5 1 = external VREFQ required for >= 200 MT/s 4 1 = supports differential signaling for DQS 3 1 = supports differential signaling for RE_n 2 1 = supports ODT value of 30 Ohms 1 1 = supports matrix termination ODT 0 1 = supports self-termination ODT
159		NV-DDR2/3 warmup cycles 4-7 Data Input warmup cycles support 0-3 Data Output warmup cycles support
160-161	O	NV-DDR3 timing mode support 8-15 Reserved (0) 7 1 = supports timing mode 10 6 1 = supports timing mode 9 5 1 = supports timing mode 8 4 1 = supports timing mode 7 3 1 = supports timing mode 6 2 1 = supports timing mode 5 1 1 = supports timing mode 4 0 1 = supports timing modes 0-3
162	O	NV-DDR2 timing mode support 3-7 Reserved (0) 2 1 = supports NV-DDR2 timing mode 10 1 1 = supports NV-DDR2 timing mode 9 0 1 = supports NV-DDR2 timing mode 8
163		Reserved (0)
	Vendor block	
164-165	M	Vendor specific Revision number
166-253		Vendor specific
254-255	M	Integrity CRC

Byte	O/M	Description
	Redundant Parameter Pages	
256-511	M	Value of bytes 0-255
512-767	M	Value of bytes 0-255
768+	O	Additional redundant parameter pages

Table 92 Parameter page definitions

5.7.1.1 Byte 0-3：参数页签名(Parameter page signature)

该字段包含了参数页签名。当签名中两个或两个以上 byte 有效时，表示该参数页有效。

Byte 0 应设为 4Fh
Byte 1 应设为 4Eh
Byte 2 应设为 46h
Byte 3 应设为 49h

5.7.1.2 Byte 4-5：版本号

该字段表示对象支持的 ONFI 规范版本号。对象可能会支持多个 ONFI 规范版本。这是个 bit 字段，其中每个定义的 bit 对应一个对象支持的确定的规范版本。

Bit 0 应被清为 0
Bit 1 为 1 表示对象支持 ONFI V1.0 规范
Bit 2 为 1 表示对象支持 ONFI V2.0 规范
Bit 3 为 1 表示对象支持 ONFI V2.1 规范
Bit 4 为 1 表示对象支持 ONFI V2.2 规范
Bit 5 为 1 表示对象支持 ONFI V2.3 规范
Bit 6 为 1 表示对象支持 ONFI V3.0 规范
Bit 7 为 1 表示对象支持 ONFI V3.1 规范
Bit 8 为 1 表示对象支持 ONFI V3.2 规范
Bit 9 为 1 表示对象支持 ONFI V4.0 规范
Bits 10-15 为保留位，应清为 0。

5.7.1.3 Byte 6-7：支持功能(Feature supported)

该字段表示对象所支持的可选特性。

Bit0 为 1 表示对象的数据总线宽度为 16-bits；Bit0 为 0 表示对象的数据总线宽度为 8-bits。host 使用该 Bit 所表示的数据总线宽度来传输所有 ONFI 定义的命令(x8 或 x16)。注意，某些命令，像 Read ID，始终只能传输 8-bit 数据。如果支持 NV-DDR、NV-DDR2 或 NV-DDR3 接口，则数据总线宽度应该是 8-bits。

Bit1 为 1 表示对象支持多 LUN 操作(见 3.1.3)；如果 Bit1 被清为 0，则 host 不能向一个 LUN 发送命令，直到对象上其他所有 LUN 都为 idle(例如，R/B_n 为 1)。

Bit2 为 1 表示对象支持非序列(non-sequential)page 编程操作，因此 host 可以对一个块内的所有 page 以任意的顺序编程。Bit2 被清为 0 表示对象不支持非序列(non-sequential)page 编程操作。如果 Bit2 被清 0，host 应该对一个块内的所有 page 从 page0 开始按顺序编程。

Bit3 为 1 表示对象支持多层(multi-plane)编程和擦除操作，参见 5.7.1.29。

Bit4 为 1 表示回拷(Copyback)操作没有奇/偶 page 的限制，比如，一个读操作可以访问一个奇数 page，然后使用 Copyback 将其内容编程到一个偶数 page。相反，一个读操作可以访问一个偶数 page，然后使用 Copyback 将其内容编程到一个奇数 page。Bit4 被清为 0 表示 host 应确保 Copyback 的读和编程操作是从奇数 page 到奇数 page，或从偶数 page 到偶数 page。

Bit5 为 1 表示对象支持 NV-DDR 接口。如果 Bit5 被设为 1，对象应将 NV-DDR 时序模式支持字段中所支持的所有 NV-DDR 时序模式标示出来。Bit5 清 0 表示对象不支持 NV-DDR 接口。

Bit6 为 1 表示对象支持多层(multi-plane)读操作，参见 5.7.1.29。

Bit7 为 1 表示对象含有一个扩展参数页，该扩展参数页存储在最后一个参数页备份之后的的数据 bytes 中。如果 Bit7 被清 0，则不支持扩展参数页。对于支持 EZ NAND 的 device，该位应该被清为 0，参见 5.7.2。注意：该位在 BA NAND 规范中被指定用来表示是否支持 BA NAND。如果 device 支持 BA NAND，则参数页中表示 ECC 修正 bit 数的 byte112 应该被清为 0。

Bit8 为 1 表示对象支持只清除被编程(80h)命令寻址的 LUN 中的 page 寄存器。如果 bit8 被清为 0，则编程(80h)命令会清除对象中每个 LUN 的 page 寄存器。在上电时，device 会清除对象中所有 LUN 的 page 寄存器，参见 5.30.1 关于怎么使能该特性。

Bit9 为 1 表示对象支持 EZ NAND。如果 Bit9 被清 0，则对象不支持 EZ NAND，对象被配置为 raw NAND。

Bit10 为 1 表示对象支持 NV-DDR2 接口。如果 Bit10 被设为 1，对象应将 NV-DDR2 时序模式支持字段中所支持的所有 NV-DDR2 时序模式标示出来。Bit10 被清为 0 表示对象不支持 NV-DDR2 接口。

Bit11 为 1 表示 NAND 对象支持 Volume 寻址(Volume addressing)，如 3.2 章定义。如果 Bit11 被清 0，则对象不支持 Volume 寻址。

Bit12 为 1 表示对象支持外部 Vpp。如果 Bit12 被清 0，则对象不支持外部 Vpp。

Bit13 为 1 表示对象支持 NV-DDR3 接口。如果 Bit13 被设为 1，对象应将 NV-DDR3 时序模式支持字段中所支持的所有 NV-DDR3 时序模式标示出来。Bit13 被清 0 表示对象不支持 NV-DDR3 接口。

Bit14 为 1 表示对象支持 ZQ 校正(ZQ calibration)。Bit14 被清 0 表示对象不支持 ZQ 校正。

Bit15 为 1 表示对象支持封装电气规范(Package Electrical Specification)和 Pad 电容(Pad Capacitance)，参见 4.10.3。如果 Bit15 被清 0，则表示对象不支持以上两规范。

5.7.1.4 Byte 8-9：可选命令支持

该字段表示对象支持的可选命令。

Bit0 为 1 表示对象支持 Page Cache Program 命令。如果 Bit0 被清 0，host 不能向对象发送 Page Cache Program 命令。对于支持 EZ NAND 的对象，该位应该被清 0，表示对象不支持 Page Cache Program 命令。

Bit1 为 1 表示对象支持 Read Cache Random，Read Cache Sequential 以及 Read Cache End 命令。如果 Bit1 被清 0，host 不能向对象发送以上 3 个命令。对于支持 EZ NAND 的对象，该位应该被清为 0，表示对象不支持这 3 个命令。

Bit2 为 1 表示对象支持 Get Feature 和 Set Feature 命令，如果 Bit2 被清 0，host 不能向对象发送这 2 个命令。

Bit3 为 1 表示对象支持 Read Status Enhanced 命令。如果 Bit3 被清 0，host 不能向对象发送该命令。如果对象具有多个 LUN 或者支持多层(multi-plane)操作，则应支持 Read Status Enhanced 命令。

Bit4 为 1 表示对象支持回拷编程(Copyback Program)和回拷读(Copyback Read)命令。如果 Bit4 被清 0，则 host 不能向对象发送这两个命令。如果支持多层(multi-plane)操作并且该位被设为 1，则应支持多层回拷操作(multi-plane copyback)。

Bit5 为 1 表示对象支持 Read Unique ID 命令。如果 Bit5 被清 0，host 不能向对象发送该命令。

Bit6 为 1 表示对象支持 Change Read Column Enhanced 命令。如果 Bit6 被清 0，则 host 不能向对象发送该命令。

Bit7 为 1 表示对象支持 Change Row Address 命令。如果 Bit7 被清 0，则 host 不能向对象发送 Change Row Address 命令。

Bit8 为 1 表示对象的编程和回拷操作都支持 Small Data Move 命令。如果 Bit8 被清 0，则对象的编程或回拷操作不支持 Small Data Move 命令。Small Data Move 命令与重叠的多层是互斥的，参见 5.19。当 Bit8 设为 1 时，device 应支持 11h 命令来刷新任何内部的数据管道(flush any internal data pipeline)，不管是否支持多层操作。

Bit9 为 1 表示对象支持 Reset LUN 命令。如果 Bit9 被清 0，host 不能向对象发送该命令。

Bit10 为 1 表示对象支持 Volume Select 命令。如果 Bit10 被清 0，host 不能发送该命令。device 如果支持 CE_n 引脚 reduction 或 matrix termination 功能，则也应支持 Volume Select 命令。

Bit11 为 1 表示对象支持 ODT Configure 命令。如果 Bit11 被清 0，host 不能发送该命令。如果 device 支持 matrix termination 功能，则也应支持该命令。

Bit12 为 1 表示对象支持 LUN Get Feature 和 LUN Set Feature 命令。如果 Bit12 被清 0，host 不能发送这两个命令。

Bit13 为 1 表示对象支持 ZQ Calibration Long(ZQCL)和 ZQ Calibration Short(ZQCS)命令。如果 Bit13 被清 0，host 不能发送这两个命令。

Bit 14-15 为保留位，应被清 0。

5.7.1.5 Byte 10：ONFI-JEDEC JTG primary advanced(主要高级)命令支持

该字段表示对象支持的、由 ONFI-JEDEC 定义的主要高级命令。对于这些命令，高级命令的主要版本并不是基于传统的 ONFI。对基于传统 ONFI 的主要高级命令的支持表示在常规的参数 page 中。

Bit0 为 1 表示对象支持 ONFI-JEDEC JTG primary Random Data Out 命令。如果 Bit0 被清 0，host 不能向对象发送该命令。ONFI-JEDEC JTG primary Random Data Out 命令是序列：00h 5-Addr 05h 2-Addr E0h。

Bit1 为 1 表示对象支持 ONFI-JEDEC JTG primary Multi-plane Page Program 命令。如果 Bit1 被清 0，host 不能向对象发送该命令。ONFI-JEDEC JTG primary Multi-plane Page Program 命令在初始编程(initial program)序列之后的第一个编程序列周期使用 81h 代替 80h。

Bit2 为 1 表示对象支持 ONFI-JEDEC JTG primary Multi-plane Copyback Program 命令。如果 Bit2 被清 0，host 不能向对象发送该命令。ONFI-JEDEC JTG primary Multi-plane Copyback Program 命令在初始编程序列之后的第一个编程序列周期使用 81h 代替 85h。

Bit3 为 1 表示对象支持 ONFI-JEDEC JTG primary Multi-plane Block Erase 命令。如果 Bit3 被清 0，host 不能向对象发送该命令。ONFI-JEDEC JTG primary Multi-plane Block Erase 在块地址之间不使用 D1h 命令。

Bit4-7 为保留位，应清 0。

5.7.1.6 Byte 12-13：扩展参数页长度

如果在对象的功能支持(Feature supported)字段中表明对象支持一个扩展参数页，则该字段用来确定扩展参数页的长度(16 bytes 的整数倍)。因此，数值 2 表示 32bytes，数值 3 表示 48bytes。最小长度为 3，对应 48bytes。

5.7.1.7 Byte 14：参数页的数量

如果在对象的功能支持字段中表明对象支持一个扩展参数页，则该字段用来确定参数页的数量。例如，数值 3 表示有 3 个参数页，因此扩展参数页从 byte768 开始。扩展参数页的数量应该和参数页的数量相同。

5.7.1.8 Byte 32-43：Device 制造商

该字段表示 device 的制造商。该字段的内容为一个 12bytes 的 ASCII 编码的字符串。如果有必要的话，device 应该使用空格(20h)来分隔字符串，以保证字符串为合适的长度。

关于制造商怎么使用 ASCII 来表示自己的名字并没有标准的规定。如果 host 要求使用标准的制造商 ID，则可以使用 JEDEC 制造商 ID，参见 5.7.1.10。

5.7.1.9 Byte 44-63：Device 模型

该字段表示 device 的模型数量。该字段的内容为一个 20bytes 的 ASCII 字符串。如果有必要，device 应使用空格(20h)来分隔字符串，以保证字符串为合适的长度。

5.7.1.10 Byte 64：JEDEC 制造商 ID

该字段表示 device 制造商的 JEDEC 制造商 ID。

5.7.1.11 Byte 65-66：数据码(Data Code)

该字段包含一个 device 制造时间的数据码。Byte65 应包含年的低两位数字(例如，数值 05h 表示 2005 年)。Byte66 应包含工作周，比如，数值 00h 表示 1 月份的第一周。

5.7.1.12 Byte 80-83：每页的数据 byte 数

该字段包含每页的数据 byte 数。该字段的数值应该是 2 的次幂，最小值为 512 bytes。

5.7.1.13 Byte 84-85：每页的 spare byte 数

该字段包含每页的 spare byte 数。数值没有限制。

附录 B 列出了基于 page size 的每页 bytes 数的推荐值，以及 device ECC 可修正 bit 数的推荐值。

5.7.1.14 Byte 86-89：废弃的(Obsolete)——每个局部页中的数据 byte 数

该字段已被废弃。

5.7.1.15 Byte 90-91：废弃的——每个局部页中的 spare byte 数

该字段已被废弃。

5.7.1.16 Byte 92-95：每块中的 page 数

该字段包含每块中的 page 数。数值应该是 32 的整数倍。参见 3.1 寻址要求。

5.7.1.17 Byte 96-99：每个逻辑单元中的块的数量

该字段表示每个逻辑单元中块的数量，数值没有限制，参见 3.1 寻址要求。

5.7.1.18 Byte 100：逻辑单元(LUN)的数量

该字段表示对象支持的逻辑单元的数量。逻辑单元数是连续的、从 LUN 地址 0 开始的。该字段应大于 0。

5.7.1.19 Byte 101：地址周期(Address Cycles)数

该字段表示用于行地址和列地址的地址周期数。该字段表示的地址周期数被 host 使用在要求行和/或列寻址的操作中(如，页编程——Page Program)。

Bits 0-3 表示用于行地址的地址周期数。该位应大于 0。

Bits 4-7 表示用于列地址的地址周期数。该位应大于 0。

注意：贯穿本规范中的例子都使用 2-byte 的列地址和 3-byte 的行地址。但是，host 负责在每个序列中提供基于该字段值的行地址和列地址的周期数。

5.7.1.20 Byte 102：每个 cell 中的 bit 数

该字段表示闪存阵列中的每个 cell 中 bit 数。该字段应大于 0。

对某些 device，会包含支持 EZ NAND 的设计，device 可能有不同类型的闪存阵列构成。数值 FFh 表示每个 cell 中的 bit 数是不确定的。

5.7.1.21 Byte 103-104：每个 LUN 中最大坏块数

该字段包含每个 LUN 中的最大坏块的数量，这些坏块是 device 在制造过程中或者在产品周期内产生的缺陷块。该最大值的确定是基于假设 host 符合参数页中规定的块耐久性要求以及 ECC 要求。

5.7.1.22 Byte 105-106：块耐久性(Block endurance)

该字段表示每个可寻址的页/块中的最大编程/擦除周期数。这个数值假设 host 使用参数页设定的 ECC 最低可修正性。

一个 page 可以在部分操作中被编程，这些操作遵循每页编程数字段中定义的值。但是，如果在同一个 page 内的不同位置编程，则不会记入该值超过一次（个人理解，如果在一个 page 内部分编程，而且位置不同，则不管在该 page 中编程了几次，只要是部分编程——既没有编程整个 page，都只记为一次）。

块耐久性表示为一个数值和一个乘数，这两个数值按照以下公式表示：

$$\text{数值} \times 10^{\text{乘数}}$$

Byte105 是数值，Byte106 是乘数。例如，一个耐久性为 75,000 的对象，应该在该字段中表示为：数值 75 和乘数 3(75×10^3)。对于一次写的 device，对象中该字段应表示为：数值 1 和乘数 0。对于只读的 device，对象中该字段应表示为数值 0 和乘数 0。数值字段应改越小越好，例如 100,000 应该被表示为数值 1 和乘数 5(1×10^5)。

5.7.1.23 Byte 107：对象起始位置开始的保证有效块

该字段表示从对象的块地址 0 开始的有效块的数量。该字段的最小值为 1h。当 host 遵循规定的可修正 bit 数时，该字段规定的块被保证成为有效块，这些有效块符合耐久性要求，参见 5.7.1.24。

5.7.1.24 Byte 108-109：保证有效的块的耐久性

该字段表示在保证有效块区域(参见 5.7.1.23)中，每个可寻址的页/块中的最小编程/擦除周期数。这个数值要求 host 使用参数页规定的 ECC 最低可修正性。该数值不会被编码。如果数值为 0000h，则没有规定最小周期数，尽管块从工厂已经保证是有效的。

5.7.1.25 Byte 110：每页中的编程数

该字段表示如果没有擦除，一个页最多可被编程的次数。在规定次数的编程操作完成后，host 应在受影响的 page 执行进一步的编程操作之前，发送一个擦除命令到该块中。该字段应大于 0。对一个 page 的相同部分进行编程而没有进行擦除操作会导致不确定的 page 内容。

5.7.1.26 Byte 111：废弃的——部分编程特性

该字段已被废弃。

5.7.1.27 Byte 112：ECC 可修正的 bit 数

该字段表示每 512bytes 中 host 可以修复的 bit 数。通过 host 修复规定数量的错误，对象应达到参数页中规定的块耐久性。当规定的错误修正数量被 host 应用块耐久性要求被遵循，则最大的坏块数量不能超过 device 中规定的值。如果最低 ECC 要求的数值大于 0，那么 page 中所有被使用的 bytes，包括 spare bytes，都应该被 host 控制器的 ECC 保护。

如果推荐的 ECC 码字的 size 不是 512bytes，那么该字段应被设为 FFh。host 应该从扩展参数页中读取扩展 ECC 信息，以便检查该 device 的 ECC 要求。

当该字段值被清为 0，则对象应返回有效数据。对于支持 EZ NAND 的对象，该字段应被清为 0，表示对象返回修正过的值给 host。

5.7.1.28 Byte 113：多层(Multi-plane)寻址

该字段描述了多层寻址参数。

Bit0-3 表示多层寻址使用的 bit 数。当支持多层操作时这个值应大于 0h。关于层地址位置信息，参见 3.1.1。

5.7.1.29 Byte 114：多层操作属性

该字段描述了多层操作的属性。当在支持功能字段中表明支持多层操作时，该字段应该是强制的。

Bit0 表示是否支持重叠的多层操作。如果 Bit0 为 1，则支持重叠的多层操作；如果 Bit0 被清 0，则支持并发的多层操作。

Bit1 表示多层操作的块地址没有限制。如果设为 1，表示所有块地址 bit 在多层操作之间可能不同。如果清为 0，表示块地址有限制。参见 bit5 块地址限制的规定。

Bit2 表示多层编程操作时否支持 program cache。如果设为 1，则多层操作支持 program cache。如果清为 0，则多层操作不支持 program cache。注意，在多层回拷编程操作中不能使用 program cache。见 bit3 多层寻址的限制。

Bit3 表示除多层寻址的多层地址位之外的块地址位在以下两种情况中是否会改变：a).一个在命令 15h 之间的 program cache 序列；或 b).一个在命令 31h 之间的 read cache 序列。如果该位设为 1 并且 bit2 被设为 1，则 host 在 program cache 序列中可能会改变多层寻址的数量和块地址位的值(除了多层地址位)。如果该位设为 1 并且 bit4 被设为 1，则 host 在 read cache 序列中可能会改变多层寻址的数量和块地址位的值(除了多层地址位)。如果该位被清 0 并且 bit2 被设为 1，则对于每个 program cache 操作，发送到 LUN 的块地址位(除了层地址位——plane address bits)和多层寻址的数量应该相同(个人理解，是针对每次 program cache 操作之间的地址位相同)。如果该位被清 0 并且 bit4 被设为 1，则对于每个 read cache 操作，发送到 LUN 的块地址位(除了多层地址位)和多层寻址的数量应该相同。

Bit4 表示多层读操作是否支持 read cache。如果该位设为 1，则多层读操作支持 read cache。如果该位清 0，则多层读操作不支持 read cache。注意，多层回拷读操作不能使用 read cache。

Bit5 表示多层操作要求的块地址限制类型。如果该位设为 1，当两个多层地址之间低位多层地址位的 XNOR 为 1 时，所有块地址位(除了多层地址位)都应该相同；如果该位被清 0，不管两个层地址间的多层地址位是什么样，所有的块地址位(除了多层地址位)都应该相同。参见 3.1.1.1 关于交错块地址限址的详细定义。该字段表示的限制适用于所有的多层操作(读，编程，擦除以及回拷编程)。

Bit6-7 为保留位。

5.7.1.1 Byte 115：EZ NAND 支持

该字段描述了对象支持的 EZ NAND 属性。该字段仅当对象支持 EZ NAND 时才被使用。

Bit0 表示对象是否支持通过 host Set Features 命令使能和 disable 自动重试功能。如果该位被设为 1，则 host 可以通过 Set Feature 命令来使能和 disable 自动重试功能；如果该位被设为 0，则 EZ NAND 控制器决定是否执行一个重试，而不需要 host 介入。如果 EZ NAND 控制器执行了自动重试，则典型的 page 读时间(tR)可能会被超过。

Bit1 表示是否支持从不同源到目标层或 LUN 的回拷操作。如果该位为 1，则回拷操作是从不同源到目标层或 LUN。如果该位被清为 0，则回拷操作是从相同的源到目标层或 LUN。

Bit2 表示对象是否要求回拷读和回拷编程命令是相邻的；如果该位被设为 1，则发送的每个回拷读命令都应该紧跟在一个回拷编程命令之后，这两个命令要交替发送，不能有两个相同的命令相连(比如，一个回拷读命令必须和一个回拷编程命令成一对)。如果该位被清为 0，则在发送回拷编程命令之前，可能会有多个回拷读命令被发送。对于 ONFI2.3，要求回拷要相邻，因此该为应该被设为 1。

Bit3-7 是保留位。

5.7.1.2 Byte 128：I/O 引脚最大电容

该字段表示对象的最大 I/O 引脚电容，单位为 pF。该字段可能被 host 用来计算数据总线的负载，参见 2.13。对于支持封装电气规范和 Pad 电容规范(参见 5.7.1.3，参数页 Byte6-7，bit15)的 device，该字段表示最大 pad I/O 电容，参见 4.10.3。

5.7.1.3 Byte 129-130：支持的 SDR 失序模式

该字段表示支持的 SDR 时序模式。对象应始终支持 SDR 时序模式 0。

Bit0 应该被设为 1，表示对象支持 SDR 时序模式 0。

Bit1 为 1 表示对象支持 SDR 时序模式 1。

Bit2 为 1 表示对象支持 SDR 时序模式 2。

Bit3 为 1 表示对象支持 SDR 时序模式 3。

Bit4 为 1 表示对象支持 SDR 时序模式 4。

Bit5 为 1 表示对象支持 SDR 时序模式 5。

Bit6-15 为保留位，应设为 0。

5.7.1.4 Byte 131-132：废弃的——支持的 SDR program cache 时序模式

该字段已被废弃。

5.7.1.5 Byte 133-134：最大 page 编程时间

该字段表示最大 page 编程时间(tPROG)，单位为 ms。

5.7.1.6 Byte 135-136：最大块擦除时间

该字段表示最大块擦除时间(tBERS)，单位为 ms。

5.7.1.7 Byte 137-138：最大 page 读时间

该字段表示最大 page 读时间(tR)，单位 ms。对于支持 EZ NAND 的 device，该值与 device 的不可修复的错误比特率有关。

5.7.1.8 Byte 139-140：最小列改变 setup 时间

该字段表示最小列改变 setup 时间(tCCS)，单位 ns。在发送了一个 Change Read Column 命令之后，host 在一段最小 tCCS 时间之后才能读数据。在发送了一个包含有所有列地址周期的 Change Write Column 命令之后，host 在经过一段最小 tCCS 时间后才能写数据。当支持 NV-DDR、NV-DDR2 或 NV-DDR3 接口时，tCCS 的值应该始终大于或等于 tWHR 和 tADL。

5.7.1.9 Byte 141：支持的 NV-DDR 时序模式

该字段表示支持的 NV-DDR 时序模式。如果对象支持 NV-DDR 接口，则应至少支持一种 NV-DDR 时序模式。对象应支持连续的 NV-DDR 时序模式(例如，如果支持时序模式 n-1 和时序模式 n+1，则也应该支持时序模式 n)。

Bit0 为 1 表示对象支持 NV-DDR 时序模式 0。

Bit1 为 1 表示对象支持 NV-DDR 时序模式 1。
Bit2 为 1 表示对象支持 NV-DDR 时序模式 2。
Bit3 为 1 表示对象支持 NV-DDR 时序模式 3。
Bit4 为 1 表示对象支持 NV-DDR 时序模式 4。
Bit5 为 1 表示对象支持 NV-DDR 时序模式 5。
Bit6-7 是保留位，应被清为 0。

5.7.1.10 Byte 142：支持的 NV-DDR2 时序模式

该字段表示支持的 NV-DDR2 时序模式。如果对象支持 NV-DDR2 接口，则至少应支持一种 NV-DDR2 时序模式。对象应支持连续的 NV-DDR2 时序模式(例如，如果支持时序模式 n-1 和 n+1，则也应支持时序模式 n)。

Bit0 为 1 表示对象支持 NV-DDR2 时序模式 0。
Bit1 为 1 表示对象支持 NV-DDR2 时序模式 1。
Bit2 为 1 表示对象支持 NV-DDR2 时序模式 2。
Bit3 为 1 表示对象支持 NV-DDR2 时序模式 3。
Bit4 为 1 表示对象支持 NV-DDR2 时序模式 4。
Bit5 为 1 表示对象支持 NV-DDR2 时序模式 5。
Bit6 为 1 表示对象支持 NV-DDR2 时序模式 6。
Bit7 为 1 表示对象支持 NV-DDR2 时序模式 7。

5.7.1.11 Byte 143：NV-DDR/NV-DDR2/NV-DDR3 特征

该字段描述了 NV-DDR，NV-DDR2 和 NV-DDR3 的操作特征和属性。当支持 NV-DDR 或 NV-DDR2 接口时该 byte 是强制的。

Bit0 表示 host 使用的 tCAD 的值。如果 Bit0 为 1，则 host 在 NV-DDR 的命令、地址和数据传输中应使用 tCADs(slow)值。如果 Bit0 为 0，则 host 在 NV-DDR 的命令、地址和数据传输中应使用 tCADf(fast)值。该位仅适用于 NV-DDR 接口。

Bit1 表示典型的 CLK 、I/O 和输入引脚电容值是否被报告在参数页中。如果该位被设为 1，则典型的 CLK、I/O 和输入引脚电容值被报告在参数页中。如果该位被清 0，则不使用典型的电容字段。

Bit2 表示 device 是否支持在数据输入期间停止 CLK，如图 63 所示。如果该位被设为 1，则 host 为了省电，可以在数据输入期间停止 CLK，进而暂停数据传输。如果该位被清 0，则 host 在数据输入期间应保持 CLK 正常运行。该位仅适用于 NV-DDR 接口。

Bit3 表示 device 在 Vpp 之前请求上电序列来提供有效的 Vcc，之后在关闭 Vcc 之前请求断电序列来关闭 Vpp。

Bit4-7 为保留位。

5.7.1.12 Byte 144-145：典型的 CLK 输入引脚电容

该字段表示对象典型的 CLK 输入引脚的电容。这个值适用于 CLK 信号。该字段仅适用于 NV-DDR 接口。该字段的单位为 0.1pF。例如，竖直 31 表示 3.1pF。该字段仅当 NV-DDR/NV-DDR2/NV-DDR3 的功能字段中表明支持典型电容值时才有效。CLK 输入引脚电容的额外约束规定在 4.10 章中。

5.7.1.13 Byte 146-147：典型 I/O 引脚电容

该字段表示对象的典型 I/O 引脚电容，单位为 0.1pF,例如，数值 31 表示 3.1pF。这个数值仅当 NV-DDR/NV-DDR2/NV-DDR3 的功能字段中表明支持典型电容值时才有效。I/O 引脚电容的额外约束规定在 4.10 章中。该值适用于所有的 I/O 引脚。如果支持 ODT，则该值也适用于 RE_n(RE_t/RE_c)和 W/R_n。对于支持封装电气规范和 Pad 电容规范（见 5.7.1.3，参数页 Byte6-7，bit15）的 device，该字段表示典型 pad I/O 电容，参见 4.10.3。

5.7.1.14 Byte148-149：典型输入引脚电容

该字段表示对象的典型输入引脚电容。该值适用于所有输入(如，ALE，CLE，WE_n)，除了：CLK，CE_n 和 WP_n 信号。如果不支持 ODT,则该值适用于 RE_n(RE_t/RE_c)和 W/R_n。该字段以 0.1pF 为单位,例如,数值 31 表示 3.1pF。该值仅当 NV-DDR/NV-DDR2/NV-DDR3 的功能字段表明支持典型电容值时才有效。输入引脚电容的额外约束规定在 4.10 章中。

5.7.1.15 Byte 150：最大输入引脚电容

该字段表示对象的最大输入引脚电容，单位为 pF。该值适用于所有输入，包括 CLK，CE_n 和 WR_n。如果支持 ODT，则该值不适用于 RE_n(RE_t/RE_c)和 W/R_n。该字段可被 host 用来计算数据总线的负载，参见 2.13。

5.7.1.16 Byte 151：支持的驱动强度

该字段表示对象是否支持可配置的驱动强度以及相关的功能。

Bit0 为 1 表示对象支持可配置的驱动强度设置，如表 30 所示。如果该位被设为 1，则 device 应支持 35 Ohm 和 50 Ohm 的设置，device 在上电时应使用表 30 定义的驱动强度 35 Ohm。如果该位被清 0，则上电时的驱动强度没有定义。对于支持 NV-DDR、NV-DDR2 或 NV-DDR3 接口的 device，该位应被设为 1。

Bit1 为 1 表示对象的 I/O 驱动强度设置支持表 30 中的 25 Ohm。

Bit2 为 1 表示对象的 I/O 驱动强度设置支持表 30 中的 18 Ohm。

Bit3-7 为保留位。

5.7.1.17 Byte 152-153：最大的多层页读时间

该字段表示多层页的最大 page 读时间(tR)，单位为 ms。多层页读时间可能大于非多层页读时间。如果在支持功能字段中表明对象支持多层读，则该字段应被支持。

5.7.1.18 Byte 154-155：编程页寄存器清除增强 tADL 值

该字段表示当编程页寄存器清除增强(program page register clear enhancement)被使能时，从 ALE 到数据 loading 的时间(tADL)，单位为 ns。如果编程页寄存器清除增强被 diasble，则 tADL 为选定的时许模式定义的值。这个增强的 tADL 值仅适用于编程(80h)命令序列，不能用于 Set Feature、回拷(Copyback)或其他命令。

5.7.1.19 Byte 156-157：EZ NAND 的典型页读时间

该字段表示支持 EZ NAND 的 device 的典型页读时间(tR)，单位为 ms。不支持 EZ NAND 的 device 不使用该字段。对于 NAND 中每个 cell 包含多 bit 的 device，该值是某些 page 的 tR 典型值的平均(比如，底层和上层的 page)。

5.7.1.20 Byte 158：NV-DDR2/3 特征

该字段描述了 NV-DDR2 和 NV-DDR3 的操作特征和属性。当支持 NV-DDR2 或 NV-DDR3 接口时，该字段是强制的。

Bit0 表示是否支持 self-termination ODT。如果该位为 1，则支持 self-termination ODT。如果该位被清 0，则不支持 self-termination ODT，并且 host 不能使能 ODT，参见 4.16。

Bit1 表示是否支持 matrix termination ODT。该位为 1 表示支持 matrix termination ODT。如果该位被清为 0，则不支持 matrix termination ODT，并且 host 不能发送 ODT 配置命令。如果支持 matrix termination ODT(该位为 1)，则 device 也应该支持 self-termination ODT，参见 4.16。

Bit2 表示是否支持可选的值为 30 Ohms 的 on-die termination。如果该位为 1，则支持值为 30 Ohms 的 on-die termination。如果该位被清 0，则不支持 30 Ohms 的 on-die termination，并且 host 不能选择值。

Bit3 表示是否支持可选的差分信号 RE_n。如果该位为 1，则支持差分信号 RE_n。如果该位被清 0，则不支持差分信号 RE_n，参见 4.10.2。

Bit4 表示是否支持可选的差分信号 DQS。如果该位为 1，则表示支持差分信号 DQS。如果该位被清 0，表示不支持差分信号 DQS。参见 4.10.2。

Bit5 表示对于速度>=200MT/s，是否需要外部 VERFQ。如果该位为 1，则需要外部 VREFQ，并且当使用的时序模式的速度>=200MT/s 时，host 应将 VEN 设为 1(参见 5.30.2)。如果该位被清 0，当使用的时序模式其速度>=200MT/s 时，host 可以选择使用 VREFQ(也就是可用可不用)。

Bit6-7 为保留位。

5.7.1.21 Byte 159：NV-DDR2/3 warmup 周期

该字段描述了 NV-DDR2 或 NV-DDR3 操作中对 warmup 周期的支持。warmup 周期定义在 4.15。当支持 NV-DDR2 或 NV-DDR3 接口时要求强制支持该字段。

Bit0-3 表示数据输出操作支持的 warmup 周期数。如果该字段被清为 0h，则对象的数据输出操作不支持 warmup 周期。host 向对象配置的 warmup 周期数不能大于提供的值(谁提供的值？)。参见 5.30.2 关于配置数据输出的 warmup

周期数的详细描述。

Bit4-7 表示数据输入操作支持的 warmup 周期数。如果该字段被清为 0h，则对象的数据输入操作不支持 warmup 周期。host 配置给对象的 warmup 周期数不能大于提供的值。参见 5.30.2 关于配置数据输入的 warmup 周期。

5.7.1.22 Byte 160-161：支持的 NV-DDR3 时序模式

该字段表示支持的 NV-DDR3 时序模式。如果对象支持 NV-DDR3 接口，则至少应支持一种 NV-DDR3 时序模式。对象应该支持连续的 NV-DDR3 时序模式(例如，如果支持时序模式 n-1 和 n+1，则也应支持时序模式 n)。

Bit0 为 1 表示对象支持 NV-DDR3 时序模式 0-3。

Bit1 为 1 表示对象支持 NV-DDR3 失序模式 4。

Bit2 为 1 表示对象支持 NV-DDR3 时序模式 5。

Bit3 为 1 表示对象支持 NV-DDR3 时序模式 6。

Bit4 为 1 表示对象支持 NV-DDR3 时序模式 7。

Bit5 为 1 表示对象支持 NV-DDR3 时序模式 8。

Bit6 为 1 表示对象支持 NV-DDR3 时序模式 9。

Bit7 为 1 表示对象支持 NV-DDR3 时序模式 10。

Bit8-15 为保留位，应被清为 0。

5.7.1.23 Byte 162：支持的 NV-DDR2 时序模式

该字段是 Byte142 的继续，表示支持的 NV-DDR2 时序模式。如果对象支持 NV-DDR2 接口，则至少应支持一种 NV-DDR2 时序模式。对象应支持连续的 NV-DDR2 时序模式(例如，如果支持时序模式 n-1 和 n+1，则也应支持时序模式 n)。

Bit0 为 1 表示对象支持 NV-DDR2 时序模式 8。

Bit1 为 1 表示对象支持 NV-DDR2 时序模式 9。

Bit2 为 1 表示对象支持 NV-DDR2 时序模式 10。

Bit3-7 为保留位，应被清 0。

5.7.1.24 Byte 164-165：制造商规定的版本号

该字段表示一个制造商规定的版本号，应被制造商用来表示支持的制造商规定参数页区域的布局以及制造商规定的特征地址。该字段的格式由制造商规定。

5.7.1.25 Byte 166-253：制造商规定

该字段为制造商使用保留。

5.7.1.26 Byte 254-255：完整性检验(Integrity CRC)

完整性校验 CRC(Cyclic Redundancy Check——循环冗余校验)字段用来验证传送到 host 的参数页内容的正确性。参数页的 CRC 为一个字(16-bit)的字段。CRC 的计算覆盖了参数页中从 byte0 到 byte253 的连续所有的数据。

CRC 应该从参数页中 byte0 开始按字节(byte—8-bit)计算。每个 byte 中的 bit 是按从高位(bit7)到低位(bit0)的顺序计算的。

CRC 应使用以下 16-bit 的生成多项式计算的：

$$G(X) = X_{16} + X_{15} + X_2 + 1$$

这个多项式可以表示为 16 进制的 8005h。

在开始计算 CRC 之前，CRC 值应该使用初始值 4F4Eh。CRC 计算完之后，最终的 CRC 值不用做 XOR。数据 byte 或计算后的 CRC 值不用反转。

5.7.1.27 Byte 256-511：冗余参数页 1

该字段应包含参数页 byte0-255 的值。Byte256 是 byte0 的值。

当 CRC 校验表明 byte0-255 中出现错误时使用冗余参数页。冗余参数页应保存在非易失性的介质中；对象不能通过转发第一个 256bytes 的值来创建该冗余参数页的数据。

5.7.1.28 Byte 512-767：冗余参数页 2

该字段包含参数页 byte0-255 的值。Byte512 是 byte0 的值。

当 CRC 校验表明 byte0-255 和第一个冗余参数页中都出现错误时使用该冗余参数页。冗余参数页应保存在非易失性的介质中；对象不能通过转发第一个 256bytes 的值来创建该冗余参数页的数据。

5.7.1.29 Byte 768+：额外的冗余参数页

768 及以上的 bytes 可以包含参数页数值的额外冗余备份。对象可以创建的冗余参数页的数量是没有限制的。对象可以创建额外的备份来避免所有三个强制的备份(参数页第一个 256bytes+两个冗余参数页数据)都发生 CRC 错误的情况。

host 可以通过检查第一个双字(Dword——哪里的第一个 Dword？)来决定是否需要创建额外的参数页。如果 4bytes 中至少有 2 个 bytes 匹配参数页签名，则需要创建额外参数页。

5.7.2 扩展参数页数据结构定义

扩展参数页如果存在的话，向 host 提供了额外的信息，这些信息因为参数页中没有足够的空间来存储，因此只能存放在扩展的参数页中。额外参数页是以段(in section)构成的，每段的长度是 16bytes 的整数倍。段的类型定义在表 93 中。

Section Type	Section Definition
0	Unused section marker. No section present.
1	Section type and length specifiers.
2	Extended ECC information.
3-255	Reserved

Table 93 Section Type Definitions

段类型(section type)应该按顺序定义在扩展参数页中(除了段类型值 0)。例如，如果段类型 12 和类型 15 都被定义在扩展参数页中，那么类型 12 应该在类型 15 之前定义。每个段类型只能有一种实例。所有未使用的段都应该被标记为一个值为 0 的段类型。当软件遇到一个值为 0 的段类型时，表示有效段结束。

表 94 定义了段类型 1。当扩展参数页中存在超过 8 个段，则段类型 1 定义了额外段。段类型 1 的长度应该为 16bytes 的整数倍。

Byte	O/M	Description
0	M	Section 8 type
1	M	Section 8 length
2	O	Section 9 type
3	O	Section 9 length
4	O	Section 10 type
5	O	Section 10 length
6 – (end)	O	Section 11 – n type & lengths

Table 94 Section Type 1: Additional Section Type and Length Specifiers

表 95 定义了段类型 2 的结构。段类型 2 规定了扩展的 ECC 信息。每个扩展的 ECC 信息块的长度都是 8bytes。如果一个扩展的 ECC 信息块没有被定义，则块中所有值都应该被清为 0。段类型 2 的长度为 16bytes 的整数倍。

Byte	O/M	Description
0-7	M	Extended ECC information block 0
8-15	O	Extended ECC information block 1
16 – (end)	O	Extended ECC information block 2 – n (if present)

Table 95 Section Type 2: Extended ECC Information

扩展的 ECC 信息块定义在 3.4 章中规定。

表 96 定义了扩展参数页的数据结构。对于跨多个 bytes 的参数，参数的最低 byte 对应第一个 byte，参见 1.4.2.3 中关于字和双字的详细描述。

当涉及的对象与数据访问宽度相关时(如，在 8-bit 数据访问的 device 中)，扩展参数页中的数值都是以 byte 为单位。例如，在一个 page 中对象会返回多少 byte 的数据。对于支持 16-bit 数据访问的 device，要求 host 将 byte 值转换为 word 值来使用。

未使用字段应该由对象清为 0h。

Byte	O/M	Description
	Revision information and features block	
0-1	M	Integrity CRC
2-5	M	Extended parameter page signature Byte 0: 45h, “E” Byte 1: 50h, “P” Byte 2: 50h, “P” Byte 3: 53h, “S”
6-15		Reserved (0)
16	M	Section 0 type
17	M	Section 0 length
18	M	Section 1 type
19	M	Section 1 length
20	O	Section 2 type
21	O	Section 2 length
22-31	O	Section 3 – 7 types & lengths
32 – (end)	M	Section information

Table 96 Extended Parameter Page definition

5.7.2.1 Byte 0-1：CRC 完整性校验

完整性校验 CRC 字段用来验证传向 host 的扩展参数页的内容是否正确。扩展参数页的 CRC 是一个 16-bit 的字段。CRC 的计算覆盖了扩展参数页中从 byte2 到最后的所有连续数值。

CRC 从扩展参数页中的 byte2 开始按 byte(8-bit)计算一直到最后。每个 8-bit 中的位是按从高位(bit7)到低位(bit0)的顺序计算的。

CRC 使用以下 16-bit 的生成多项式计算：

$$G(X) = X_{16} + X_{15} + X_2 + 1$$

这个多项式可以表示为 16 进制的 8005h。

在开始计算 CRC 之前，CRC 值应该使用初始值 4F4Eh。最终的 CRC 值不用做 XOR。数据 byte 或 CRC 计算值不用进行反转。

5.7.2.2 Byte 2-5：扩展参数页签名

该字段包含扩展参数页的签名。当签名的两个或两个以上 byte 有效时，表示扩展参数页是有效的。

Byte2 应被设为 45h。

Byte3 应被设为 50h。

Byte4 应被设为 50h。

Byte5 应被设为 53h。

5.7.2.3 Byte 16：段 0 类型

段 0 是扩展参数页的第一个段，从 byte 偏移 32 开始。该字段规定了段 0 的类型。段类型定义在表 93 中。

5.7.2.4 Byte 17：段 0 长度

段 0 是扩展参数页的第一个段，从 byte 偏移 32 开始。该字段规定了段 0 的长度。长度以 16byte 的整数倍定义的，因此，数值 1 表示 16bytes，数值 2 表示 32bytes。

5.7.2.5 Byte 18：段 1 类型

段 1 是扩展参数页的第二个段，紧接着段 0 开始。该字段规定了段 1 的类型。段类型定义在表 93 中。如果没有段 1，则该字段应该被清为 0。

5.7.2.6 Byte 19：段 1 长度

段 1 是扩展参数页的第二个段，紧接着段 0 开始。该字段规定了段 1 的长度。长度以 16bytes 的整数倍定义，因此，数值 1 表示 16byte，数值 2 表示 32bytes。如果没有段 1，则该字段应被清为 0。

5.7.2.7 Byte 20：段 2 类型

段 2 是扩展参数页的第三个段，紧接着段 1 开始。该字段规定了段 2 的类型。段类型定义在表 93 中。如果没有段 2，则该字段应被清为 0。

5.7.2.8 Byte 21：段 2 长度

段 2 是扩展参数页的第二个段，紧接着段 1 开始。该字段规定了段 2 的长度。长度以 16bytes 的整数倍定义，因此，数值 1 表示 16bytes，数值 2 表示 32bytes。如果没有段 2，则该字段应被清为 0。

5.7.2.9 Byte 22-31：段 3-7 类型和长度

Bytes22-31 按照段 0 和段 1 类型和长度定义的格式，按顺序定义了段 3-7 的类型和长度。如果某段不存在，则相应的类型和长度字段应被清 0。

5.7.2.10 Byte 32-(结束)：段信息

段 0 从 byte 偏移 32 开始，长度为 16bytes 的整数倍。如果有额外的段(段 1，2，3 等)，则每段紧接着上一个段开始，其长度也都是 16bytes 的整数倍。

5.8 Read Unique ID 定义

Read Unique ID 命令用来校验 device 的 16byte 唯一 ID(Unique ID——UID)。唯一 ID 代表 device 制造商，是唯一的(该定义是我理解的，不一定准确，英文原文实在读不通)。

UID 数据保存在闪存阵列中。为了让 host 判断 UID 是否没有位错误(bit error)，UID 和其补码(complement)一起被返回，如表 97。如果 UID 的 XOR 以及其逐位(bit-wise)补码的 XOR 都为 1，则 UID 是有效的。

Bytes	值
0-15	UID
16-31	UID 补码(逐位的—bit-wise)

表 97 UID 及补码

为了在 UID 有 bit error 的情况使 UID 的校验更充分，应该在对象中存储一个 16 位的 UID 备份及对应的补码。例如，读 bytes32-63 会向 host 返回一个 UID 和其补码的备份。

在 Read Unique ID 命令执行期间不能发送 Read Status Enhanced 命令。

图 86 定义了 Read Unique ID 的行为。host 可以使用对象支持的任何时序模式来校验 UID 数据。

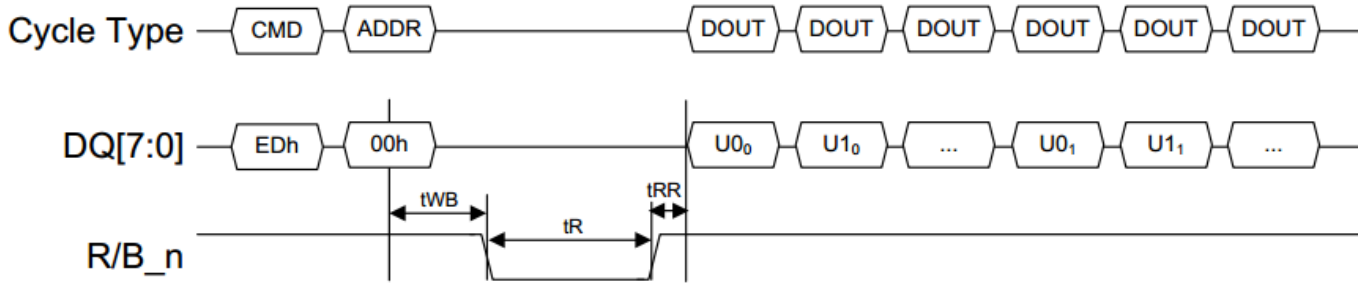


Figure 86　　Read Unique ID command timing

U0k-Unk　UID 及其补码的第 K 个备份。读取超过 512byte 会返回未知值。

5.9 块擦除(Block Erase)定义

块擦除命令用来擦除由 LUN 中块地址参数定义的块数据。一个块擦除操作在结束后，如果 SR[0]返回 0，则认为块擦除操作成功了。当 SR[6]由 0 变为 1 时，该命令的 SR[0]有效，直到下一次 SR[6]变为 0。图 87 定义了块擦除操作的行为和时序。

如果 host 试图擦除一个工厂标记为坏块的块时，device 不会启动该操作，device 会将该操作的 FAIL 位置为 1。

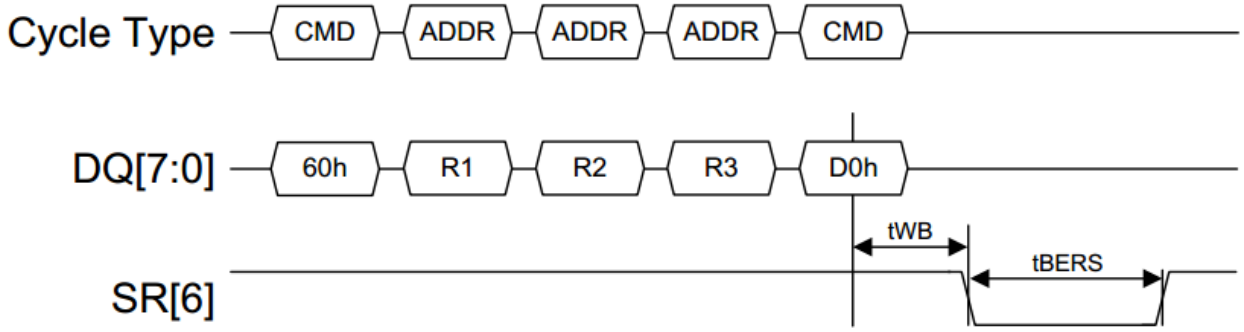


Figure 87　　Block Erase timing

R1-R3　　要擦除的块的行地址。R1 是行地址的最低 byte。

5.10 读状态(Read Status)定义

在非多层操作中，读状态命令用来检查上一个操作的状态值。如果多个多层操作正在一个 single LUN 上进行，则读状态命令会为状态寄存器返回复合的(composite)状态值，而每层地址的状态寄存器都是独立的。也就是说，读状态命令会返回表 98 所示的独立寄存器位的复合状态值。参见 5.13 状态寄存器位定义。

Status Register bit	Composite status value
Bit 0, FAIL	OR
Bit 1, FAILC	OR
Bit 3, CSP	OR

Table 98　　Composite Status Value

当在 NV-DDR、NV-DDR2 或 NV-DDR3 接口中发送读状态命令时，每个数据 byte 都被接收两次。host 只锁存每个数据 byte 的其中一个备份。参见 4.4。

图 88 定义了读状态的行为和时序。

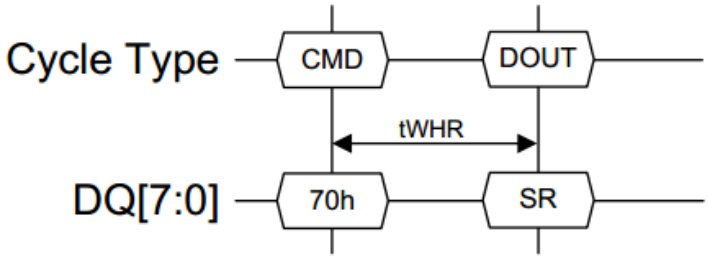


Figure 88　　Read Status timing

SR　　状态值，如 5.13 定义

读状态命令可以使用 SDR、NV-DDR、NV-DDR2 或 NV-DDR3 接口发送。每种接口的时序参数如图 89，图 90，图 91 所示。

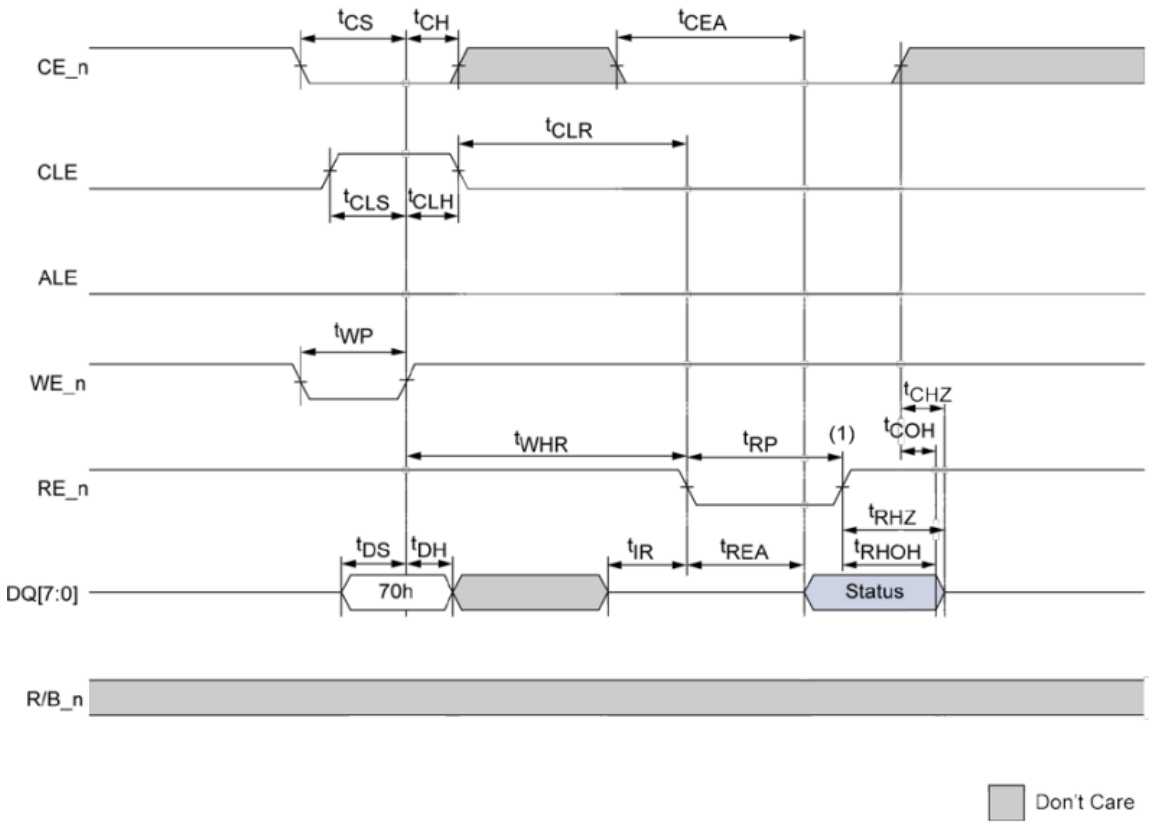


Figure 89 Read Status command using SDR data interface

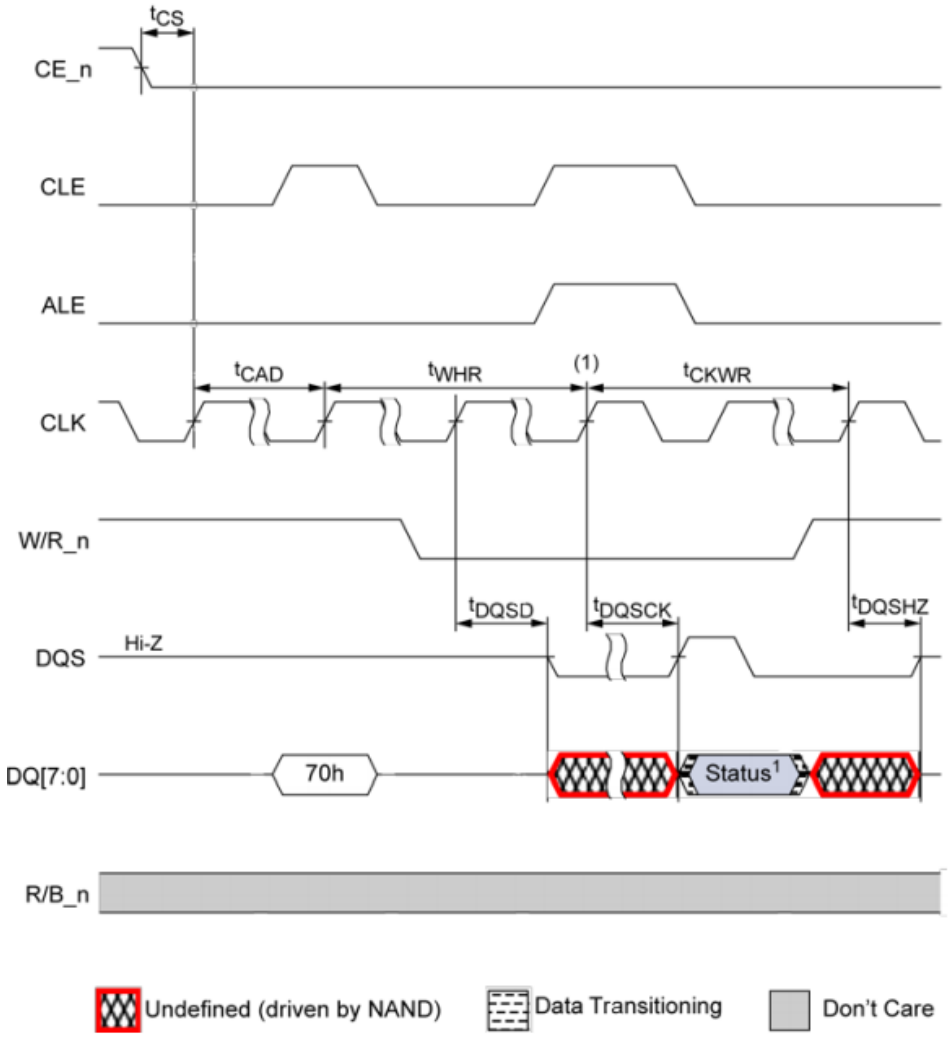


Figure 90 Read Status command using NV-DDR data interface

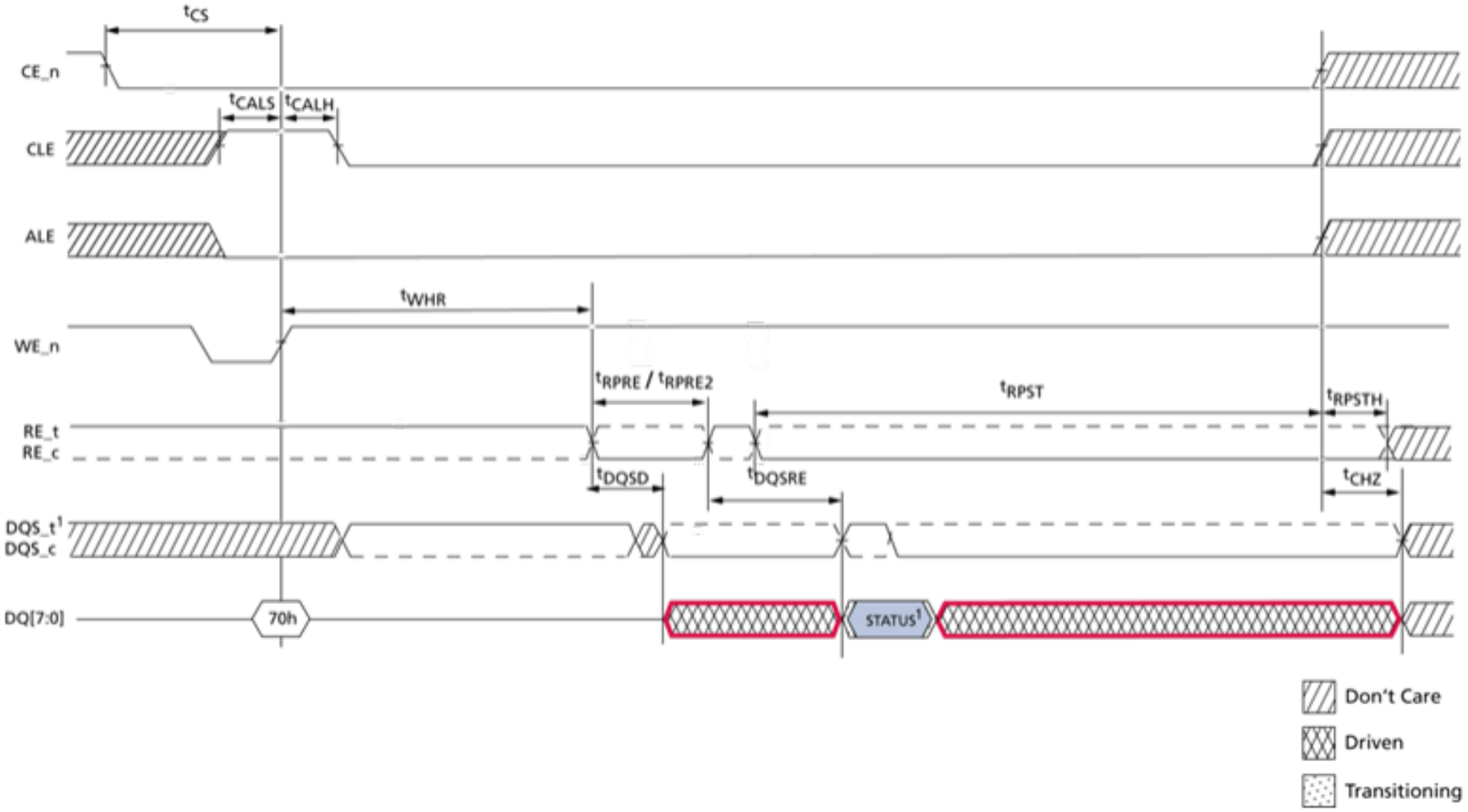


Figure 91 Read Status command using NV-DDR2/3 data interface

注 1:对于 SDR 接口,通过 RE_n 脉冲或者将 RE_n 拉低可以不断地读状态。对于 NV-DDR 接口,可以通过将 ALE/CLE 置为 11b 来不断地读状态。对于 NV-DDR2/3 接口，在 RE_n 保持为低期间，device 可以选择更新状态(可更新可不更新，由功能决定是否支持)。如果 device 支持在 RE_n 保持为低期间更新状态，则 host 可以通过 DQ[7:0]数据值的更新来持续读取更新过的状态。但是，DQS 只能根据 RE_n 的转变来转变。如果 device 不支持在 RE_n 保持为低期间更新状态，则状态根据 RE_n 的转变来更新。在图 91 中，当总线状态不是数据输入或数据输出周期时，如果 ALE、CLE 以及 CE_n 都为低(比如，idle 状态)，则 host 应将 DQS(DQS_t)置为低，以防止 device 使能 ODT。如果 ODT 被 disable，则在 idle 状态期间不用关心 DQS。

5.11 Read Status Enhanced 定义

Read Status Enhanced 命令用来检索(读取)由地址决定的特定 LUN 和层次上的上一个操作的状态值。图 92 定义了 Read Status Enhanced 命令的行为和时序。如果行地址是无效的，则返回的状态值为未知值。host 使用 Read Status Enhanced 命令来选择 LUN(参见 3.1.2)。注意，如果 LUN 中页寄存器被选中用于数据输出，则 Read Status Enhanced 命令没有效果。

当在 NV-DDR、NV-DDR2 或 NV-DDR3 接口中发送 Read Status Enhanced 命令时，每个数据 byte 会被接收两次，host 只会锁存每个数据 byte 的其中一个备份，参见 4.4。

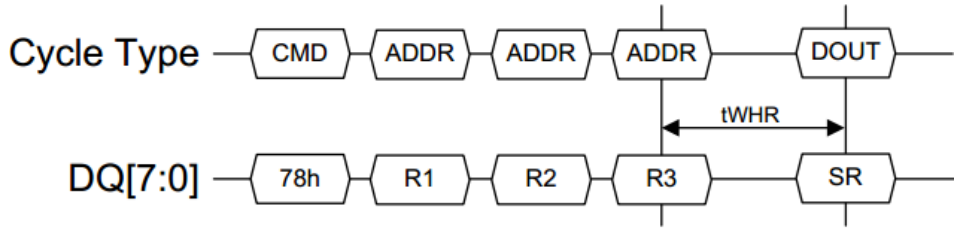


Figure 92 Read Status Enhanced timing

R1-R3 包含 LUN 和层地址的行地址，用来检索状态。与 LUN 和层地址无关的行地址不会被使用。R1 是最低 byte。

SR 状态值，如 5.13 章中定义

5.12 读状态(Read Status)和 Read Status Enhanced 的使用要求

在确定的序列中，host 只能使用一种状态命令。本章节描述了在什么情况下要求使用特定的状态命令。

如果在 R/B_n 为 0 期间向 LUN 发送了一个命令，那么接下来的状态命令应该是 Read Status Enhanced 命令，该命令会使未被选中的 LUN 关闭其输出缓存，以保证只有被 Read Status Enhanced 命令选中的 LUN 才会响应后面 RE_n 输入信号的翻转。

当 host 已经向多个 LUN 同时发送了 Read Page 命令时，在从任何 LUN 读取数据之前，host 应发送 Read Status Enhanced 命令。该命令会使未被选中的 LUN 关闭其输出缓存，以保证在数据输出被 00h 命令选中之后，只有被 Read Status Enhanced 命令选中的 LUN 才会响应后面 RE_n 输入信号的翻转，参见 3.1.3，关于如果在多 LUN 读操作中使用了 Change Read Column(Enhanced)命令时的额外要求。

在对象级命令期间和之后，host 不能发送 Read Status Enhanced 命令。在这种情况下，host 使用读状态命令来见车状态值。该要求唯一的例外是，在 Reset 已经发送的情况下，如果向多个 LUN 发送命令，这种情况下可以使用 Read Status Enhanced 命令来判定每个 active 的 LUN 何时完成 Reset。

5.13 状态字段定义

读状态和 Read Status Enhanced 命令返回的状态寄存器的 byte 值(SR)其格式描述如下。如果 RDY 位被清 0，则状态 byte 中所有其他位(除 WP_n 外)都是无效的，host 应忽略这些值。

Value	7	6	5	4	3	2	1	0
Status Register	WP_n	RDY	ARDY	VSP	CSP	R	FAILC	FAIL

FAIL　如果为 1，则表示上一个命令失败。如果为 0，表示上一个命令成功。对于行 NAND 操作，该位只在编程或擦除操作时有效。对于 EZ NAND 操作，该位在读、编程以及擦除操作中有效。在 program cache 操作期间，该位仅当 ARDY 为 1 时有效。在支持 ZQ 校正的 device 中，如果 ZQ 校正失败，则该位被设为 1。

FAILC　如果为 1，则上一个命令的前一个命令失败(上上个命令)。如果该位为 0，在上一个命令的前一个命令成功。该位仅在 program cache 操作中有效。在 Page Cache Program 序列中，该位直到第二个 15h 命令或 10h 命令被发送后才会有效。如果不支持 program cache 操作，则该位不会使用，应被清 0。对于 EZ NAND 操作，该位不被使用(EZ NAND 不支持 cache 命令)。

CSP　Command Specific：该位具有命令特别含义？。

对于 EZ NAND 操作，如果 CSP(Threshold-临界)位被设为 1，则上一个读操作超出了 ECC 的极限，host 应执行适当的操作(例如，向一个新的空间重新写数据)。当 FAIL 为 1 时不用关心 CSP(临界)位。

对于其他操作，该位是保留位。

ARDY　如果为 1，则表示没有正在进行的阵列操作。如果为 0，表示某个命令正在被执行(RDY 被清为 0)，或正在进行一个阵列操作。如果不支持重叠多层操作(overlapped multi-plane)或 cache 命令，则该位不会被使用。

RDY　如果为 1，表示另一个命令的 LUN 或层地址已经准备好，并且状态值中所有其他位都有效。如果为 0，则表示发送的上一个命令还没有执行完，并且 SR[5:0]位是无效的，应该被 host 忽略。该位会影响 R/B_n 的值，参见 2.18.2。当 cache 操作正在进行时，该位表示另一个命令是否可被接受，而 ARDY 表示上一个操作是否完成。

WP_n　如果为 1，则表示 device 不是写保护的。如果为 0，表示 device 是写保护的。不论 RDY 位是什么值，该位都始终是有效的。

R　保留的

VSP　制造商定义的

5.14 读命令定义

读命令读取一个由 LUN 中的行地址确定的页的数据。页数据可以从页寄存器中从指定的列地址读取。图 93 定义了读命令的行为和时序。读操作超出一个页的结尾会导致向 host 返回未知值。

当通过检测度状态来判定 tR(从闪存阵列传输到页寄存器)什么时候完成时，host 应重新发送一个 00h 命令来开始读数据。发送 00h 命令会导致要被返回的数据从选定的列地址开始传输。

对于支持 EZ NAND 的 device，应该在 tR 结束后发送一个 Read Status(Enhanced)命令，以确保在数据从 device 传输之前读操作是成功的。如果在检查状态之前数据就从 device 被传输出来了，则返回的数据有可能是坏的(corrupt data)。

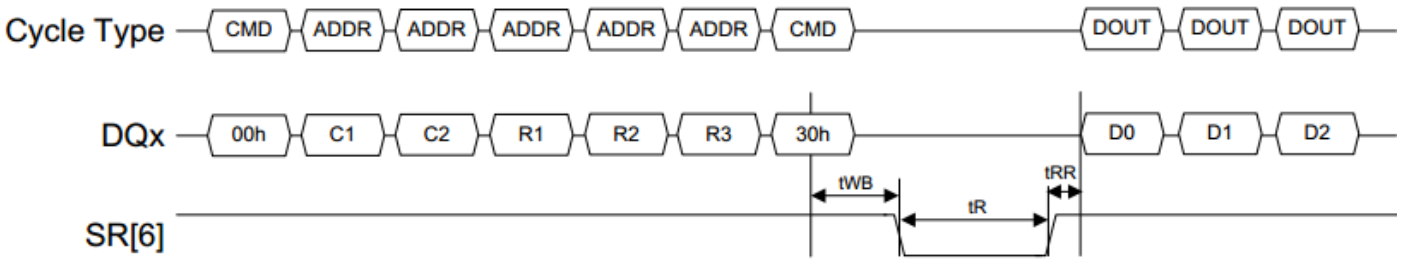


Figure 93　Read timing

C1-C2　要检索的页的列地址。C1 是最低 byte。　——检索即寻址，个人理解
R1-R3　要检索的页的行地址。R1 是最低 byte。
Dn　从寻址的页中读出的数据 byte

5.15 Read Cache 定义

该命令不支持 EZ NAND。

当另一个页正在同时从闪存阵列中选定的 LUN 被读取期间，Read Cache Sequential 和 Read Cache Random 命令可以许可(permit——批准、许可的意思？)一个将要从页寄存器被读取的页。一个如 5.14 章定义的读命令，应该在一个 read cache 序列中 Read Cache Sequential 或 Read Cache Random 命令开始之前被发送。Read Cache Sequential 或 Read Cache Random 命令应该在发送 Read Cache End(3Fh)命令之前被发送。

Read Cache(Sequential 或 Random)命令可以在读命令完成之后(SR[6]为 1)被发送。Host 可以进入闪存阵列中将要被读取的下页地址。数据输出始终从列地址 00h 开始。如果 host 没有进入一个寻址的地址，则下一个相邻的页会被读取。当 Read Cache(Sequential 或 Random)命令被发送后，SR[6]被清为 0(busy)。操作开始后，SR[6]被置 1(ready)，之后 host 可以开始从上一个读或 Read Cache(Sequential 或 Random)操作读取数据。发送一个额外的 Read Cache(Sequential 或 Random)命令会将最近的从阵列读取的数据复制到页寄存器中。当再没有页将被读取时，通过发送 3Fh 命令将最后一个页复制到页寄存器。当 SR[6]为 1 时(ready)，host 可以开始从页寄存器读取数据。当 31h 和 3Fh 命令被发送后，SR[6]应该被清 0(busy)，直到页被从闪存阵列复制完毕。

当一个块的最后一个页被读取后，host 不能再发送 Read Cache Sequential(31h)命令。如果命令被同时发送给多个 LUN，host 在发送一个 Read Cache Sequential(31h)或 Read Cache End(3Fh)命令给 LUN 之前，应先发送一个 Read Status Enhanced(78h)命令来选择该 LUN。

图 94 定义了 Read Cache Sequential 命令的行为和时序，这个命令用来为接下来发送到对象的读命令开始执行 cache 操作(意思就是在读命令之前要先执行 cache 命令来把数据复制到页寄存器——个人理解)。图 95 定义了 Read Cache Random 行为和时序，该命令同样用来为接下来发送到对象的读命令开始执行 cache 操作。每种情况下，SR[6]表示下一个选定的页是否可以从页寄存器被读取。

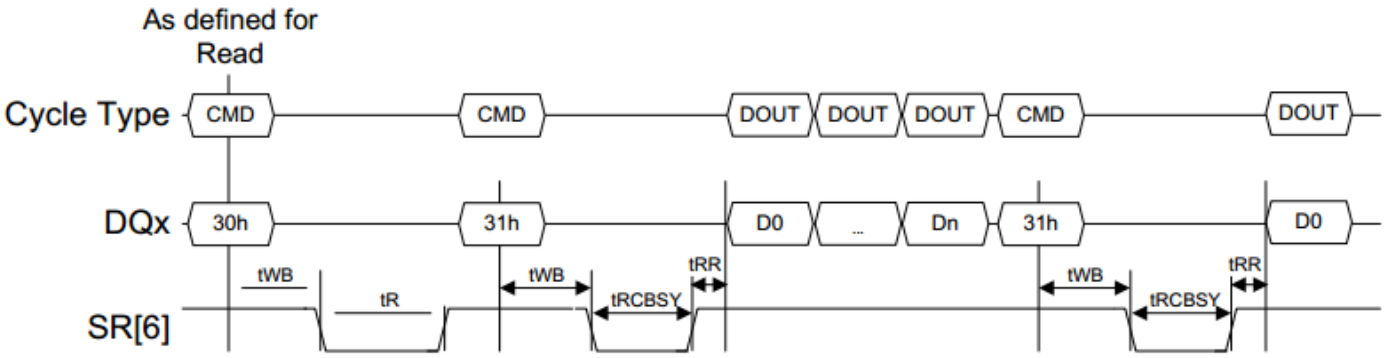


Figure 94　Read Cache Sequential timing, start of cache operations

D0-Dn　由最初的读或之前的 cache 操作从页中读取的数据 byte/words。

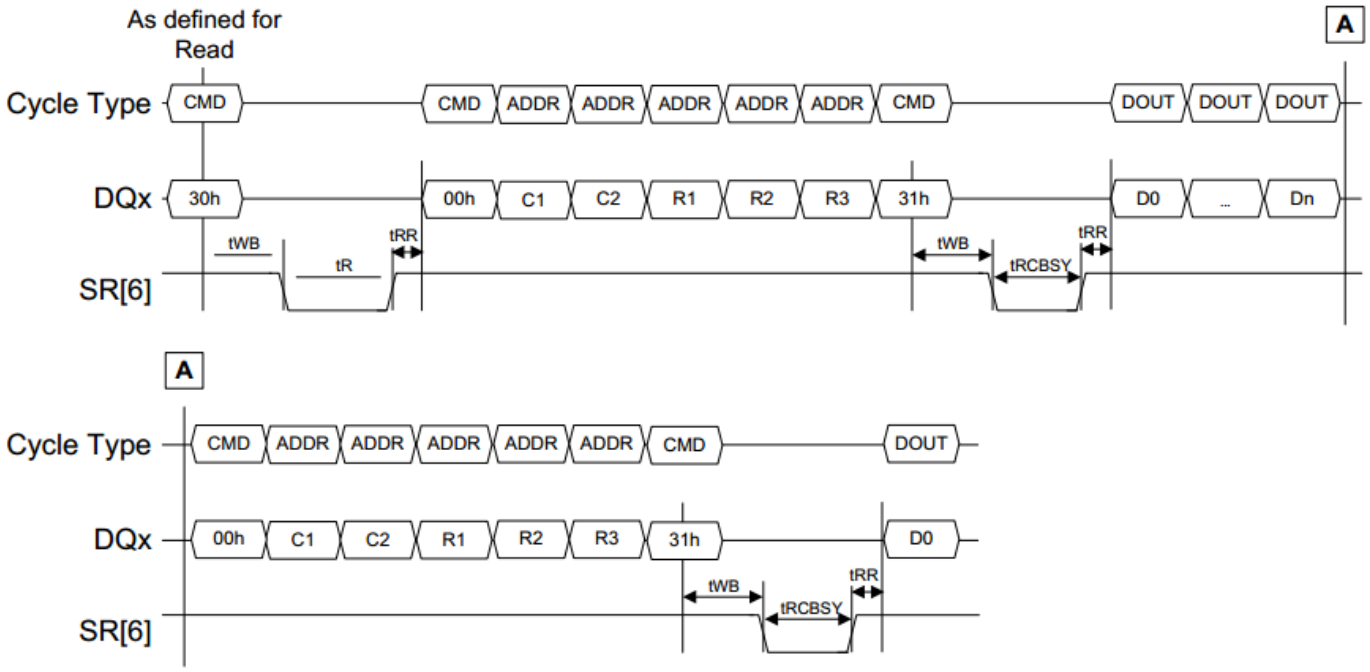


Figure 95　　Read Cache Random timing, start of cache operations

C1-C2　　要检索的列地址。C1 是最低 byte。列地址被忽略。

R1-R3　　要检索的行地址。R1 是最低 byte。

D0-Dn　　由最初的读或者之前的 cache 操作从页中读取的数 bytes/words

图 96 定义了 cache 操作结束时 Read Cache(Sequential 或 Random)命令的行为和时序。该行为对 Read Cache Sequential 和 Read Cache Random 命令都适用。命令码 3Fh 表示传输最后一个选定的页的数据到页寄存器，而不用开始另一个读操作。

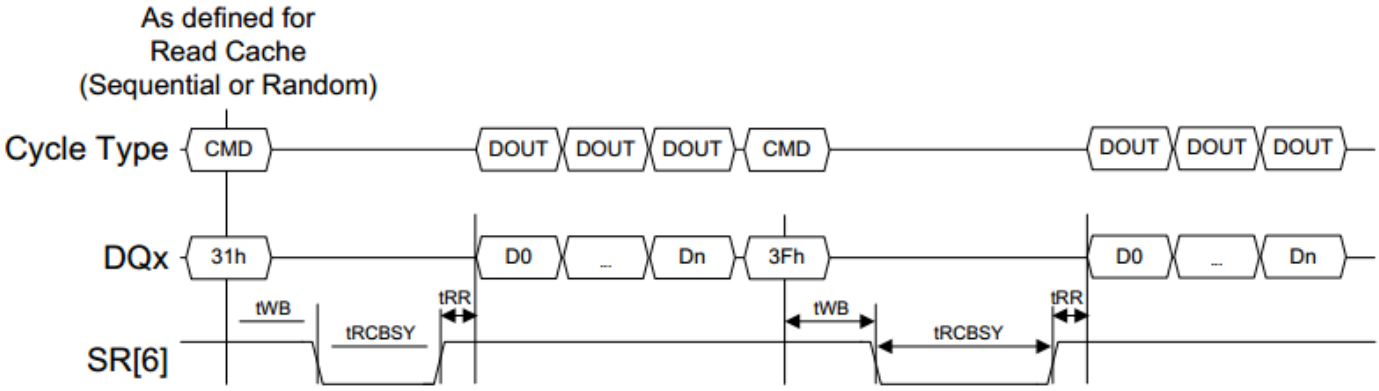


Figure 96　　Read Cache timing, end of cache operations

D0-Dn　　有之前的 cache 操作从页中读取的数据 bytes/words

5.16 页编程(Page Program)定义

页编程命令将列地址指定的一个页或部分页的数据传输到页寄存器。页寄存器的内容随后被编程到行地址指定的闪存阵列中。在 SR[6]从 0 变为 1 之后，该命令的 SR[0]有效，直到 SR[6]下一次变为 0。图 97 定义了页编程命令的行为和时序。对超过页寄存器结尾的写操作没有被定义。

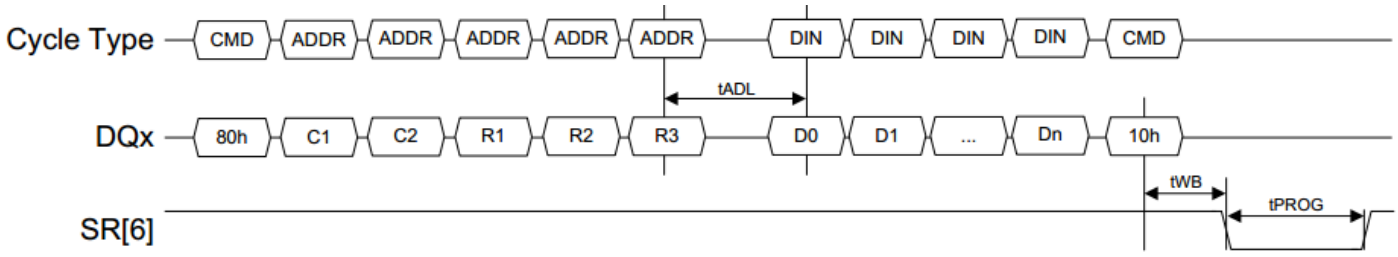


Figure 97　　Page Program timing

C1-C2　要写入数据的缓存开始位置的列地址。C1 是最低 byte。

R1-R3　被编程的页的行地址。R1 是最低 byte。

D0-Dn　要写入被寻址页的数据 bytes/words。

5.17 页缓存编程(Page Cache Program)定义

EZ NAND 不支持该命令。

当要编程的下个页被 host 传输到页寄存器期间，页缓存编程命令可以许可(permit——批准、许可的意思？)将要被写入闪存中指定 LUN 的一个页或者部分页的数据。当命令 10h 被发送后，在 SR[6]被置为 1(ready)之前，所有的数据都被写入到闪存阵列中。当 SR[5]从 0 变为 1 后，该命令的 SR[0]是有效的，直到 SR[5]再次变为 0。当 SR[6]从 0 变为 1 之后，该命令的 SR[1]有效。

图 98 和图 99 定义了页缓存编程命令的行为和时序。注意，在缓存(caching)操作结束时的 tPROG 可能会比典型值长，因为这个时间也用于结束前一个页的编程操作。对超过页寄存器末尾的写操作没有被定义。

如果支持 program page register clear enhancement，当收到编程(80h)命令时，host 可以选择只清除选定的 LUN 或层地址对应的页寄存器。这种情况下，tADL 时间可能会比选定的时序模式中的定义要长，参见 5.7.1.18。关于怎么使能该功能的细节，参见 5.30。

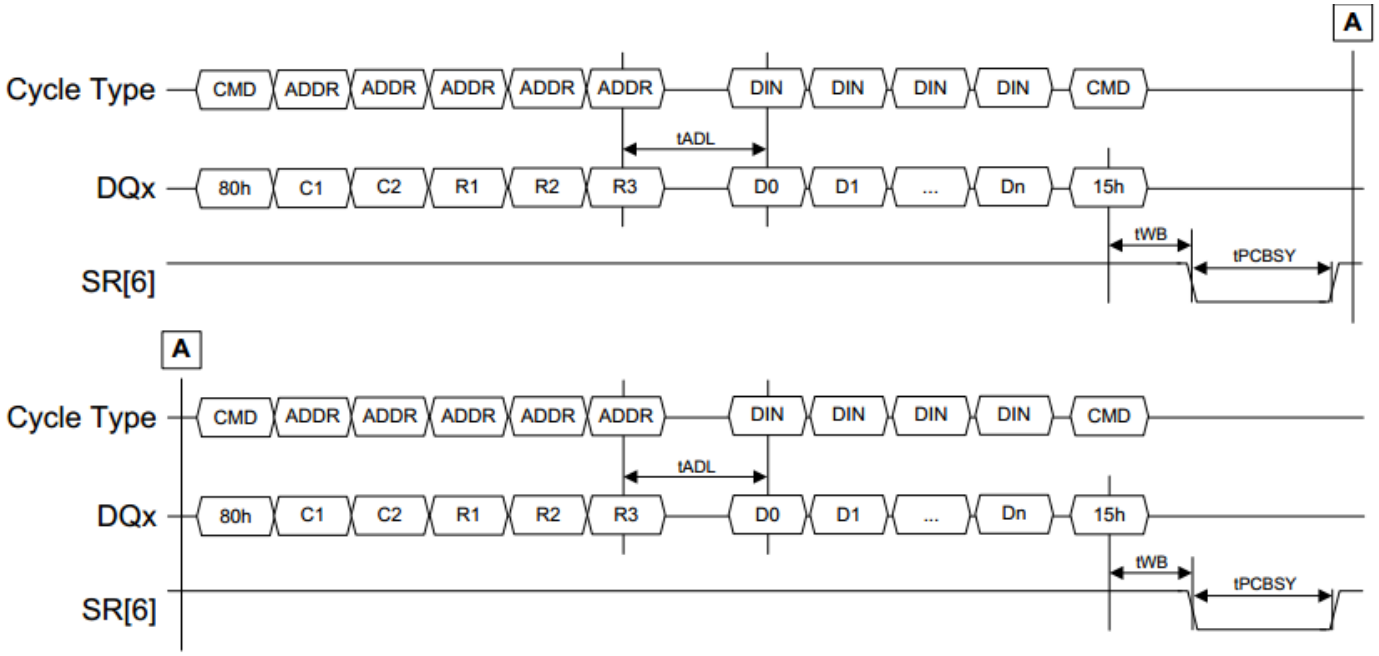


Figure 98　　Page Cache Program timing, start of operations

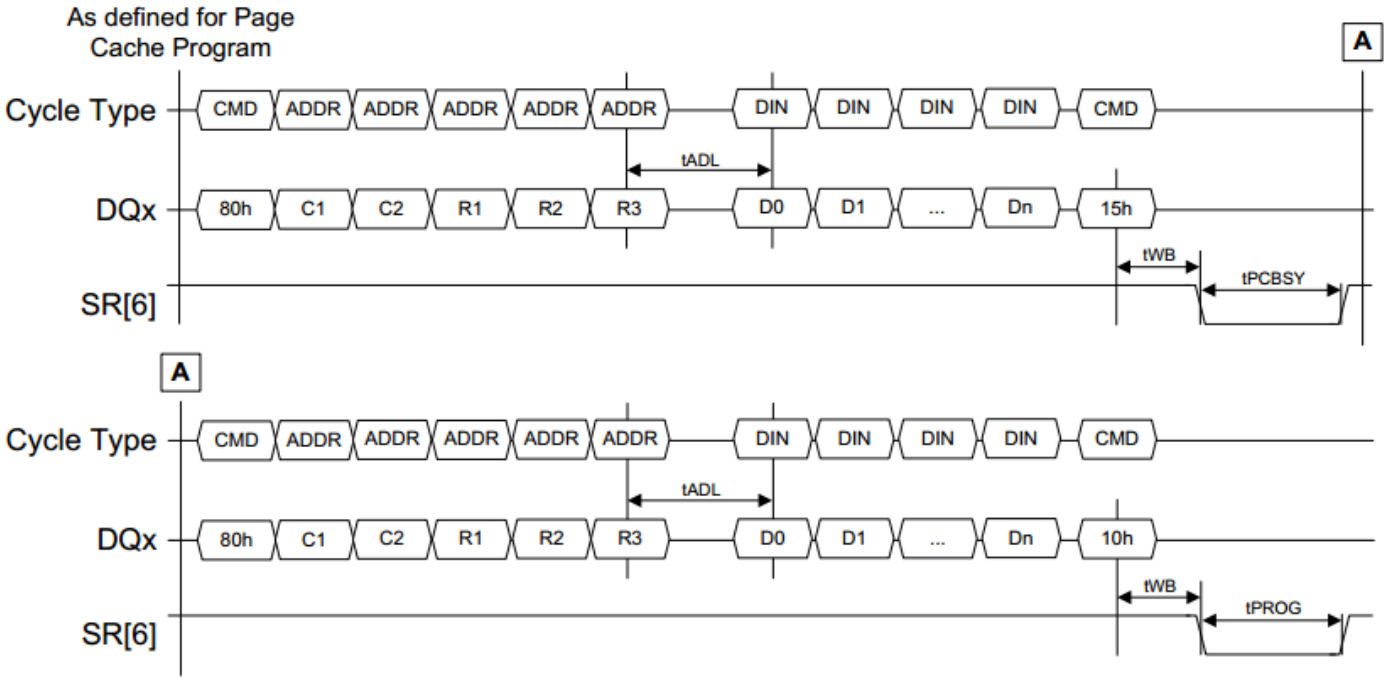


Figure 99　　Page Cache Program timing, end of operations

C1-C2　　将要写入数据的缓存起始位置的列地址。C1 是最低 byte。

R1-R3　　被编程的页的行地址。R1 是最低 byte。

D0-Dn　　要写入被寻址页的数据 bytes/words。

5.18 回拷(Copyback)定义

回拷命令从一个位置读取页数据然后将读取的数据移动到同一个 LUN 中的第二个位置。如果对象支持 EZ NAND，则数据可以被移动到不同层或 LUN 的第二个位置。可以参考参数页来判定支持回拷的目标位置。从第一个位置读取的数据可以被 host 读取，包括使用 Change Read Column。在读出数据结束并发送回拷编程(Copyback Program)后，host 可以根据需要使用 Change Write Column 命令执行数据修正(data modification)。图 100 定义了回拷操作的行为和时序。

回拷操作使用单独的页寄存器来进行读和编程操作。如果对象支持 EZ NAND，则 EZ NAND 控制器中的缓存用来进行读和编程操作，每个 LUN 中的页寄存器不能被访问。

当支持多层寻址时，回拷读和回拷编程的多层地址应该和非多层回拷操作相同。如果支持 EZ NAND，该限制不适用；参见参数页。

回拷同样也有奇/偶页的限制。当从奇数页读取时，内容需要被写入一个奇数页。相反，当从偶数页读取时，读取的内容需要被写入一个偶数页。参见 5.7.1.3。

这个版本的 ONFI 规范要求所有开发都遵循回拷相邻(Copyback Adjacency)的规则。在未来 ONFI 版本中，对于 EZ NAND 的开发，有可能不会要求回拷读和回拷编程要相邻。例如，在一个回拷编程之前，可能会有多个回拷读被发送。这个要求也同样适用于多层回拷操作。

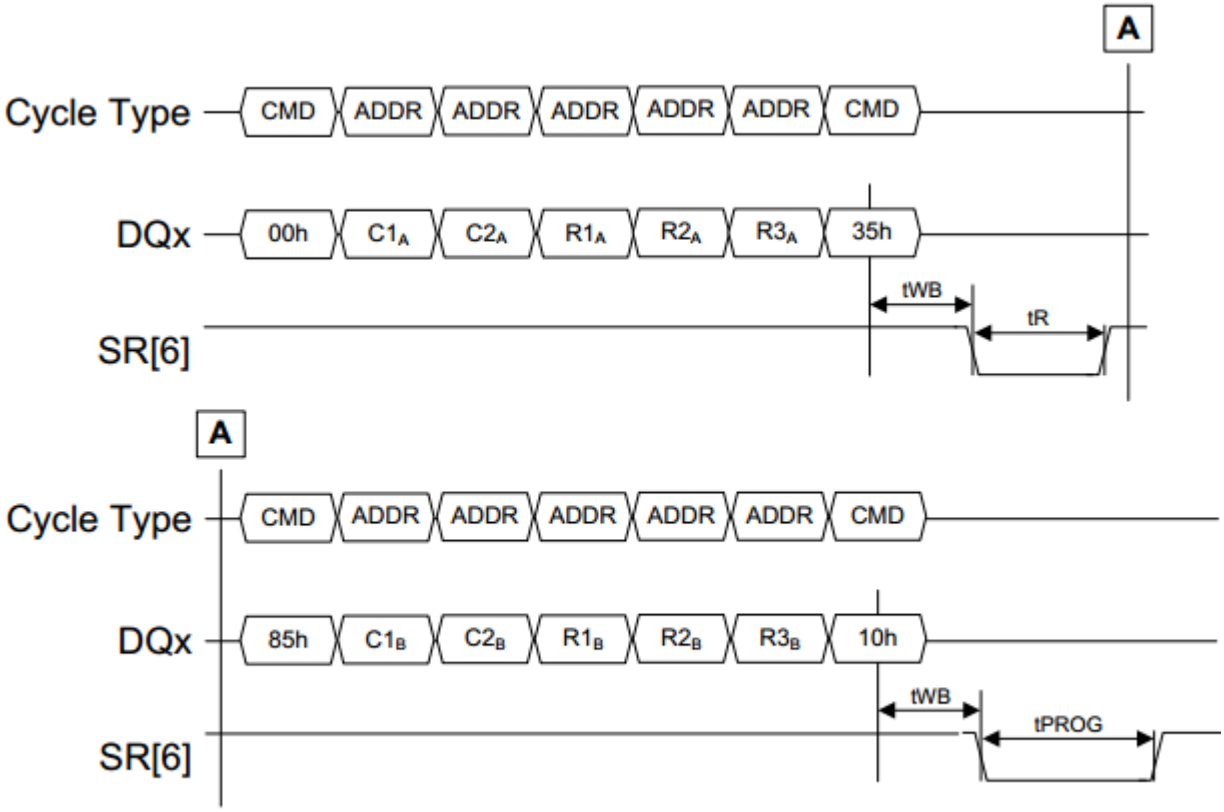


Figure 100 Copyback timing

$C1_A$-$C2_A$ 要检索的页的列地址。$C1_A$是最低 byte。

$R1_A$-$R3_A$ 要检索的页的行地址。$R1_A$是最低 byte。

$C1_B$-$C2_B$ 要编程的页的列地址。$C1_B$是最低 byte。

$R1_B$-$R3_B$ 要编程的页的行地址。$R1_B$是最低 byte。

图 101 和图 102 定义了数据输出和数据修正支持的回拷。

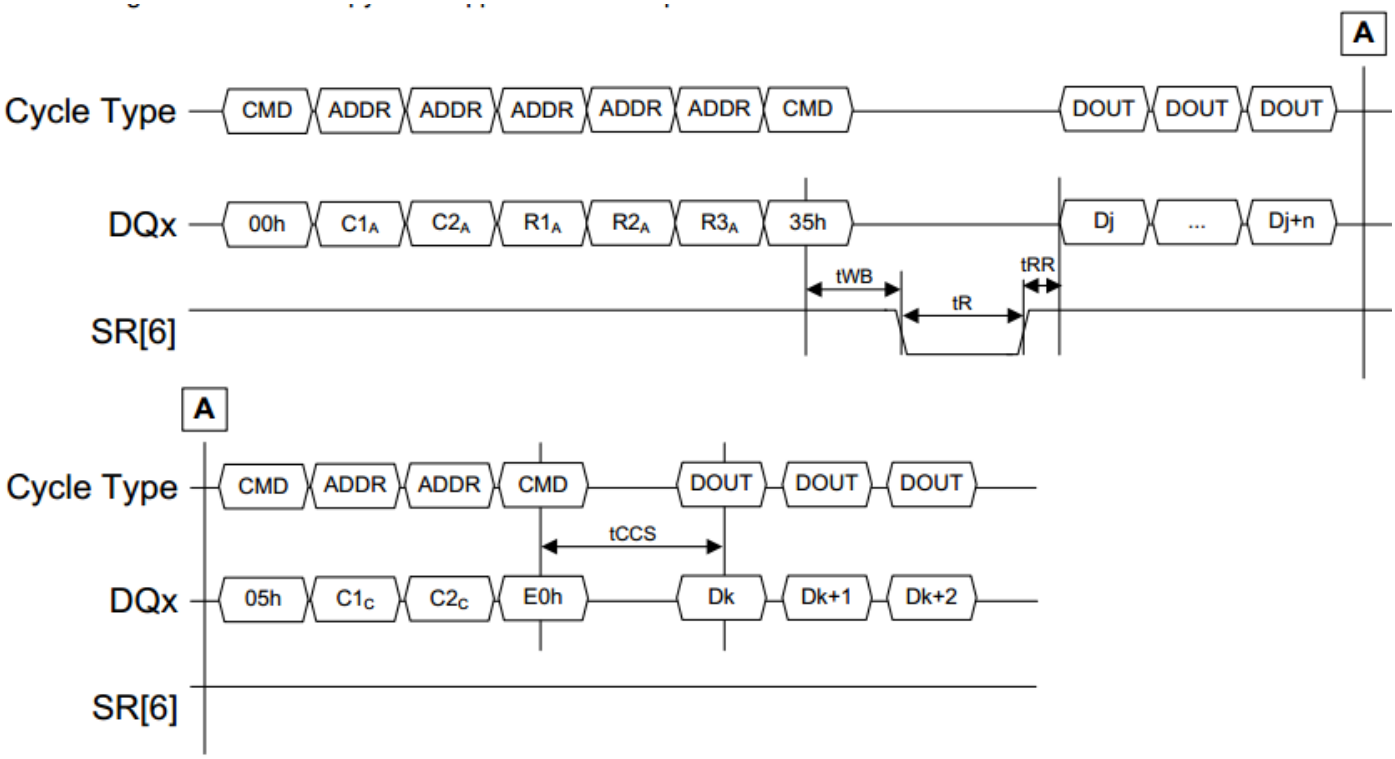


Figure 101 Copyback Read with data output

$C1_A$-$C2_A$ 要检索的页的列地址。$C1_A$ 是最低 byte。

$R1_A$-$R3_A$ 要检索的页的行地址。$R1_A$ 是最低 byte。

Dj-(Dj+n) 从 $C1_A$-$C2_A$ 中指定的列地址开始读取的数据 byte。

C1C-C2C 要从页寄存器读出的新位置(k)的列地址。$C1_C$ 是最低 byte。

Dk-Dk+n 从 $C1_C$-$C2_C$ 中指定的列地址开始读取的数据 byte。

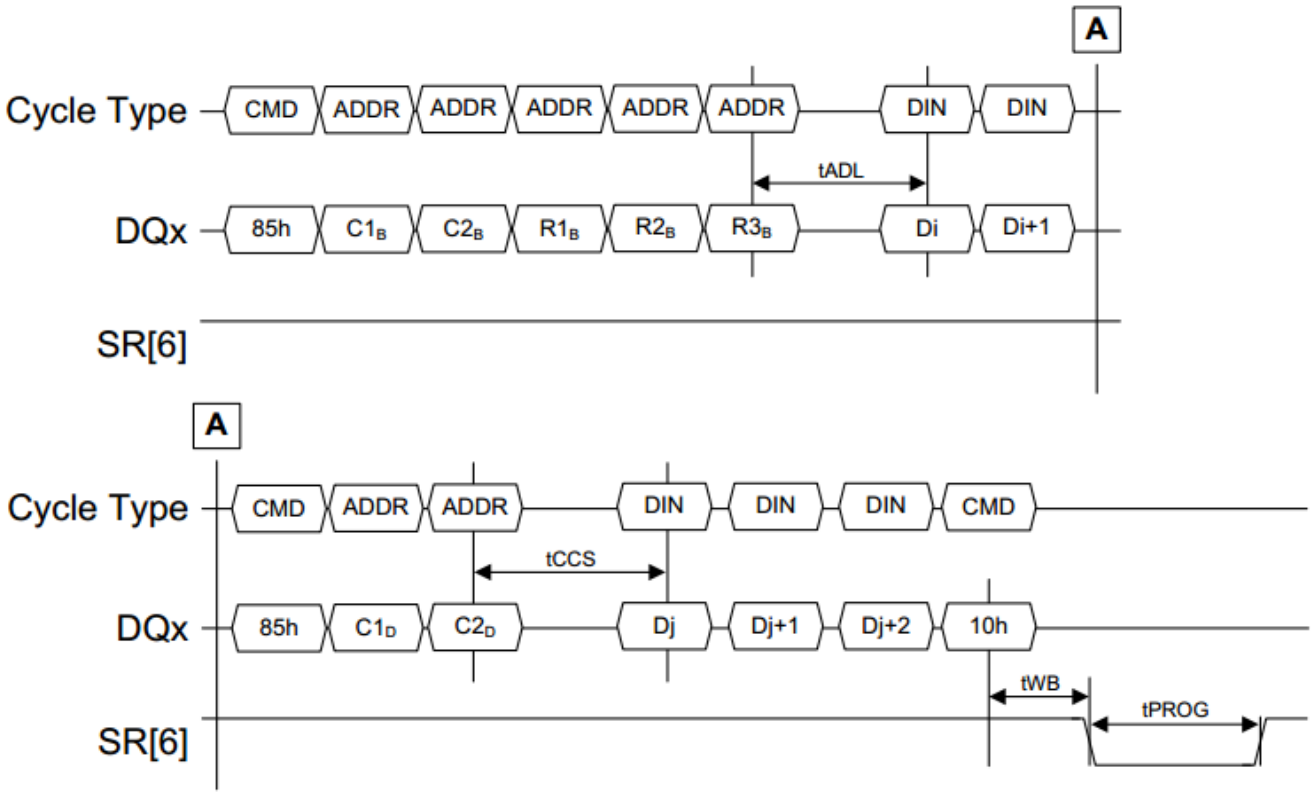


Figure 102　　Copyback Program with data modification

$C1_B$-$C2_B$　　要编程的页的列地址，$C1_B$ 是最低 byte。

$R1_B$-$R3_B$　　要编程的页的行地址，$R1_B$ 是最低 byte。

Di-Di+n　　从 C1-$C2_B$ 中指定的列地址开始的要 overwrite 到页寄存器的数据 byte。

$C1_D$-$C2_D$　　页寄存器中要 overwrite 数据的新位置(j)的列地址。$C1_D$ 是最低 byte。

Dj-Dj+n　　从 C1-$C2_D$ 中指定的列地址开始 overwrite 的数据 byte。

5.19 Small Data Move

如果参数页中表明支持 Small Data Move 命令，那么对于编程和回拷操作(包括多层编程和回拷操作)，host 可以以小于 device 的 page size 的增量将数据传输到页寄存器。该操作中 host 也可以读出数据。如果 Small Data Move 是一个没有数据输出的编程操作，则第一个周期可以使用操作码 80h。对于含有数据输出的回拷和编程操作，在第一个周期中应适用操作码 85h。

图 103 定义了一个具有 Small Data Move 的编程或回拷编程的数据修正过程;该序列可以根据需要来重复进行，以完成数据传输。图 104 定义了最后一个编程操作，该操作用来结束具有 small data move 操作的编程或回拷编程。发往相同层地址的序列中所有的编程操作应该具有相同的行地址(R1B-R3B)。一个 small data move 操作中的命令 11h 在恢复数据输出之前，用来刷新任何内部的数据管道(通道)。

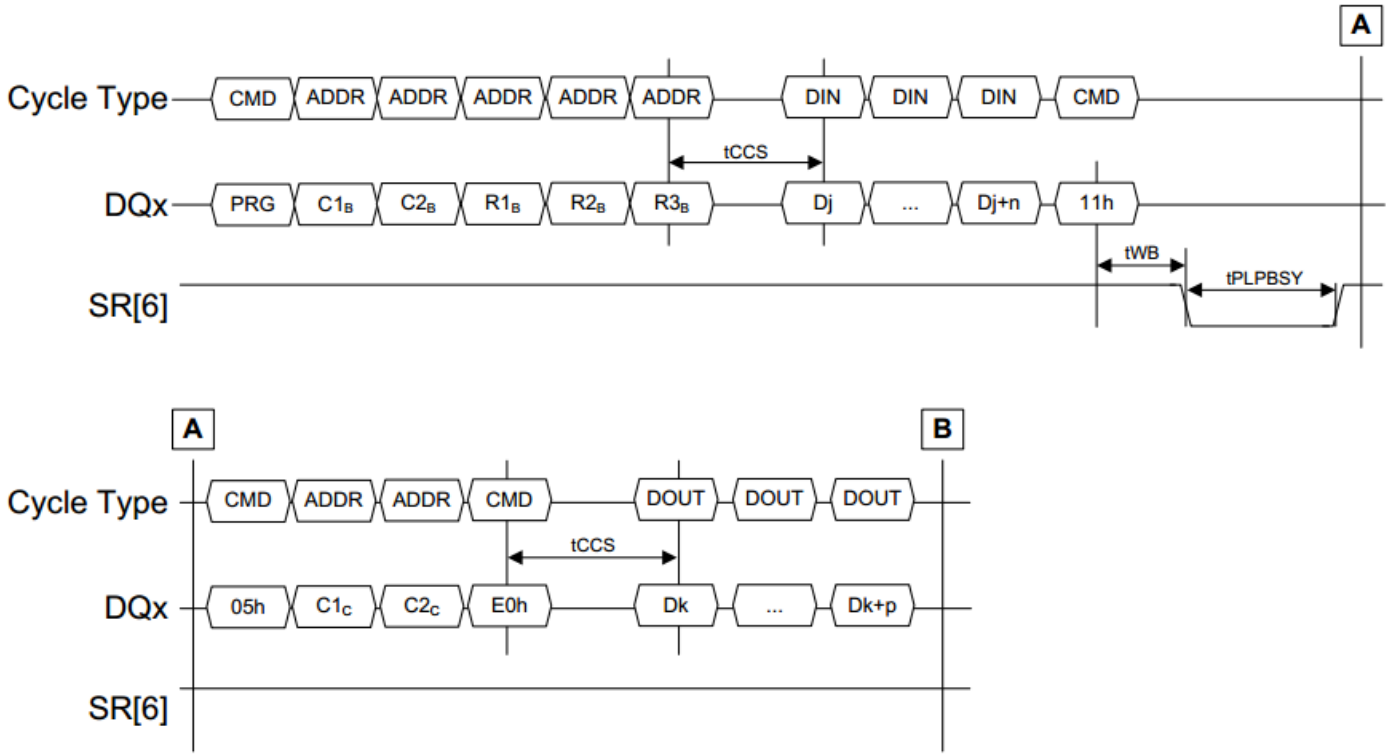


Figure 103　　Small data moves, data modification

PRG　　编程命令，80h 或 85h。紧接着任何数据输出的命令应该是 85h。

C1-C2$_B$　　页寄存器中要被写的列地址。C1$_B$ 是最低 byte。

R1-R3$_B$　　要编程的页的行地址。R1$_B$ 是最低 byte。

Dj-(Dj+n)　　页寄存器中从 C1-C2$_B$ 指定的列地址开始被更新的数据 byte。

C1-C2$_C$　　页寄存器中要检索的 byte/word 的列地址。C1$_C$ 是最低 byte。

Dk-(Dk+p)　　从 C1-C2$_C$ 指定的列地址开始读取的数据 byte。

注意：如果支持 Change Read Column Enhanced 命令，则图 103 中该命令(small data move)可被替换为 Change Read Column 命令。该流程中 Change Read Column(Enhanced)命令的使用以及数据输出是可选的；该流程可用来为编程或回拷编程操作递增地传输数据。

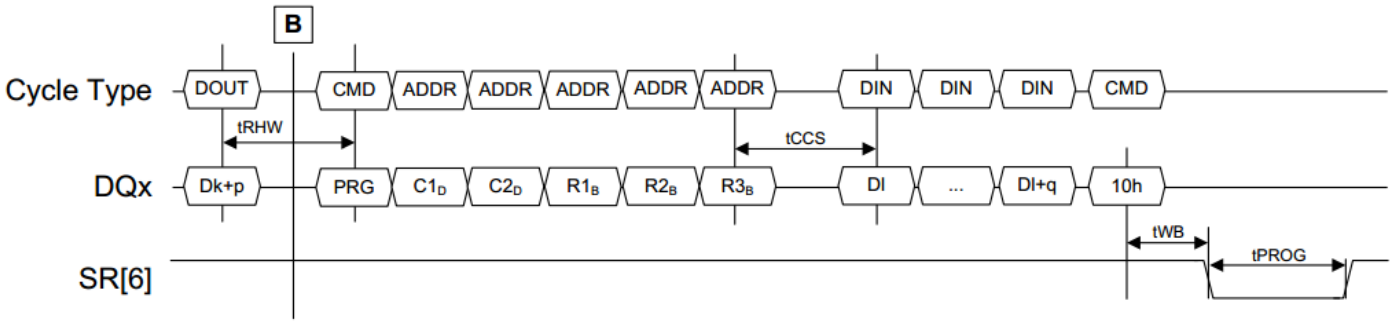


Figure 104　　Small data moves, end

PRG　　编程命令，80h 或 85h。如果命令中有任何数据输出时应使用 85h 命令。

C1-C2D　　页寄存器中要被写的列地址。C1D 是最低 byte。

R1-R3B　　要编程的页的行地址。R1B 是最低 byte。

DI-(DI+q)　　页地址中从 C1-C2D 指定的列地址开始被更新的数据 byte。

5.20 Change Read Column 定义

Change Read Column 命令用来改变列地址，以便在被选定的 LUN 的页寄存器中从该被改变的地址读取数据。只有当 LUN 处于一个读空闲(read idle)状态时，才能发送 Change Read Column 命令。图 105 定义了 Change Read Column 命令的行为和时序。

Host 只能在 E0h 命令被发送到 LUN 后经过 tCCS ns 时间之后从 LUN 读取数据，如图 105 所示。

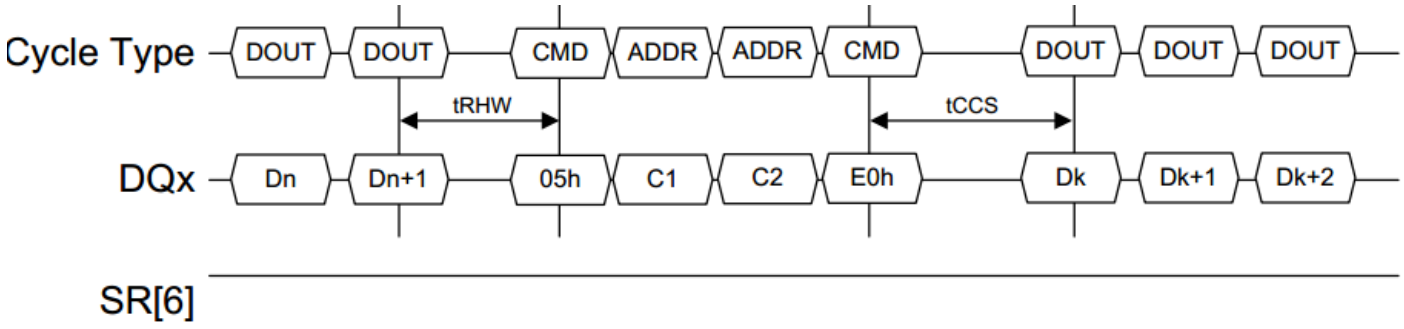


Figure 105 Change Read Column timing

Dn 列地址变化之前读取的数据 byte

C1-C2 要为接下来的数据传输设置的新的列地址。C1 是最低 byte。

Dk 从新的列地址开始读取的数据 byte。

5.21 Change Read Column Enhanced 定义

该命令用来改变 LUN 地址，层地址和列地址，以便让之后的读命令能够从一个页中这些改变过的地址来读取数据。当独立的 LUN 操作或多层 LUN 操作被执行时使用该命令来给定新列的完整地址。图 106 定义了 Change Read Column Enhanced 命令的行为和时序。

除非参数页中表明支持 Change Read Column Enhanced 命令，否则该命令不能被 host 发送。当对象级数据输出命令(Read ID,Read Parameter Page,Read Unique ID,Get Feature)正在执行时不能发送 Change Read Column Enhanced 命令，或者该命令不能紧接着对象级命令立即被发送。

Change Read Column Enhanced 命令可以使没有被选中的空闲的 LUN(SR[6]为 1)关闭其输出缓存，以保证只有被 Change Read Column Enhanced 命令选中的 LUN 才能响应接下来的数据输出。当 Change Read Column Enhanced 命令被发送后，如果未被选中的 LUN 是 active 的(SR[6]为 0)，则 host 应该在接下来的数据输出之前发送一个 Read Status Enhanced(78h)命令来保证所有未被选中的 LUN 都关闭了输出缓存。

ONFI-JEDEC 组织已经定义了一个修正版的 Change Read Column Enhanced 命令，用于随机数据输出(Random Data Output)。在该定义中，一个 00h 命令用来指定要读取数据的行地址(块或页)，随后发送一个正常的 Change Read Column 命令来指定列地址。该定义如图 107 所示。可以检查参数页来判定 device 是否支持该修正版本。

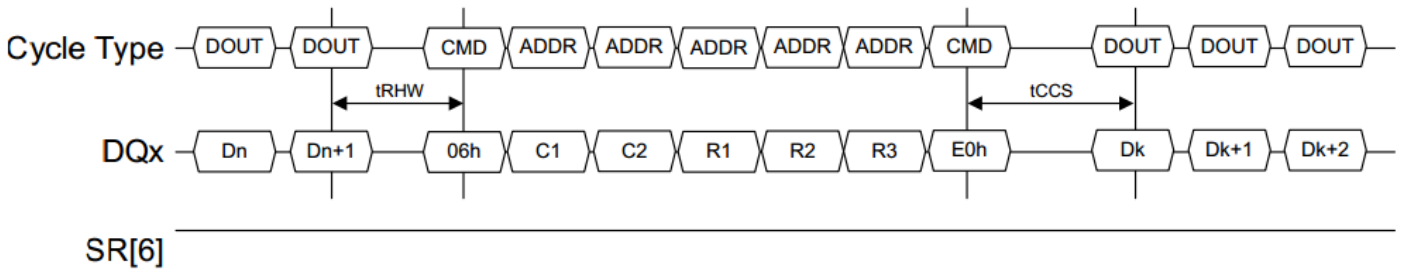


Figure 106 Change Read Column Enhanced timing

Dn 行和列地址改变前读取的数据 byte。

C1-C2 为接下来数据传输设置的新的列地址。C1 是最低 byte。

R1-R3 为接下来数据传输设置的新的行地址。R1 是最低 byte。

Dk 从新的行和列地址开始读取的数据 byte。

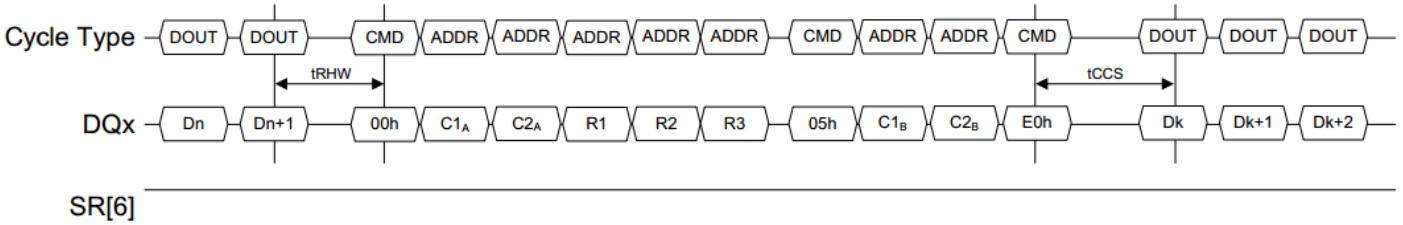


Figure 107 Change Read Column Enhanced timing, ONFI-JEDEC Joint Taskgroup primary definition

Dn 行地址和列地址改变前读取的数据 byte。

C1A-C2A 00h 序列指定的列地址；未被使用。C1A 是最低 byte。

R1-R3 为接下来数据传输设置的新的行地址。R1 是最低 byte。

C1B-C2B 为接下来数据传输设置的新的列地址。C1B 是最低 byte。

Dk 从新的行和列地址开始读取的数据 byte。

5.22 Change Write Column 定义

Change Write Column 命令用来改变被选中的 LUN 的页寄存器中要被写入的列地址。图 108 定义了该命令的行为和时序。

Host 只能在上一个列地址被写入到 LUN 后经过 tCCS ns 之后才能写数据到 LUN，如图 105 所示。

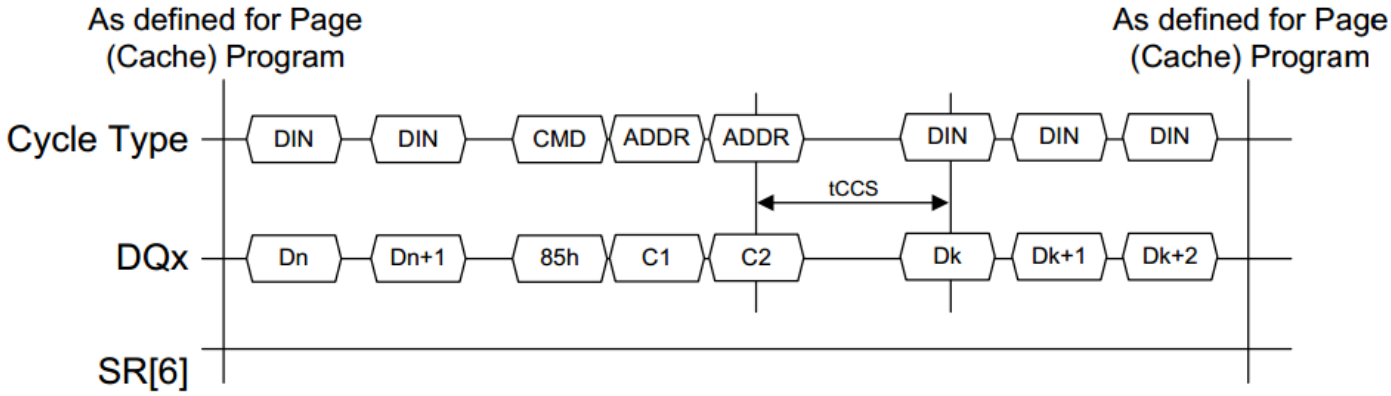


Figure 108 Change Write Column timing

C1-C2 为接下来数据传输设置的新的列地址。C1 是最低 byte。

Dn 被写入到前一个被寻址列的数据 byte。

Dk 被写入到新的列地址的数据 byte。

5.23 Change Row Address 定义

Change Row Address 命令用来为选定的 LUN 改变将被写入的行和列地址。这个机制可用来未正在执行的编程操作调整块地址、页地址以及列地址。正在执行中的编程操作，其 LUN 和层地址应该相同。图 109 定义了 Change Row Address 命令的行为和时序。

Host 只能在上一个行地址被写入到 LUN 后经过 tCCS ns 之后才能向 LUN 写数据，参见图 109。

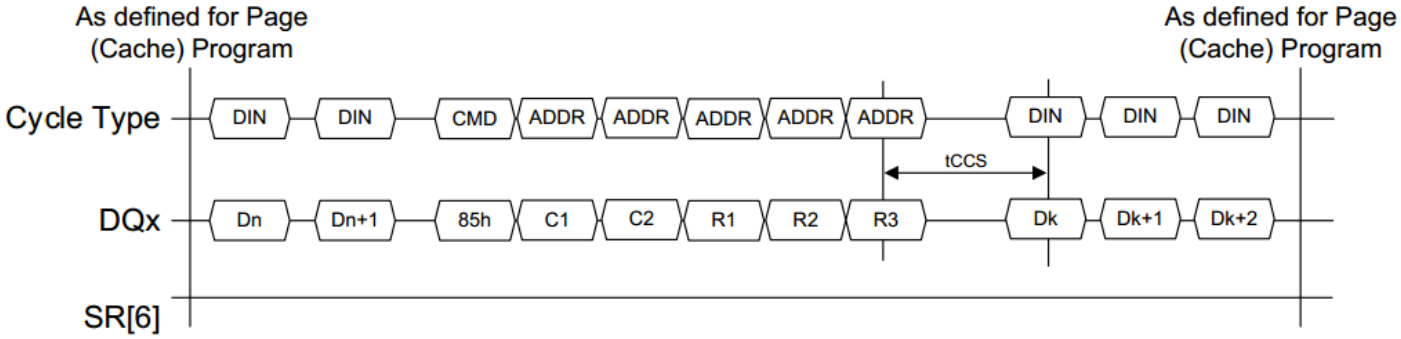


Figure 109　Change Row Address timing

C1-C2	为接下来数据传输设置的新的列地址。C1 是最低 byte。
R1-R3	正在被编程的页的行地址。正在进行中的编程操作的 LUN 地址和层地址应该相同。R1 是最低 byte。
Dn	行地址改变前正在被写的数据 byte；该数据将被写入新的行地址。
Dk	从最新列地址开始被写入新的块和页的数据 byte。

5.24 Volume 选择定义

Volume Select 命令用来选择一个由地址指定的特定的 Volume。当使用 CE_n 引脚 reduction 或 matrix termination 时，要求使用 Volume Select。

该命令被所有连接到特定 Host Target 的 NAND 对象接受(参见 2.20)。该命令可以在任何状态下，在 Volume 中的任何 LUN 中被执行。Volume Select 命令只能在 CE_n 被拉低后作为第一个命令被发送；CE_n 应该被保持为高至少 tCEH，以便 Volumn Select 命令可以被所有连接到 Host Target 的 NAND 对象正确接收。

如果共用同一个 Host Target 的 Volumn 被配置为使用不同的数据接口，则 host 应该使用 SDR 接口来发送 Volume Select 命令。

当 Volume Select 命令被发送之后，Volume 地址和指定地址不匹配的所有 NAND 对象都应该被取消选择(deselected)以达到省电目的(等效于 CE_n 被拉高的行为)。如果未选中的 Volumn 中有一个 LUN 已经被指定为被选中的 Volume 的 terminator，则该 LUN 将进入 Sniff 状态，参见表 71on-die termination 的 LUN 状态描述。

如果被设定的 Volumn 地址不会对应到任何指定的 Volumn 地址，则所有 NAND 对象都应被取消选择，直到发送下一个 Volume Select 命令。如果 Volume Select 命令不是 CE_n 被拉低后的第一个命令，则 NAND 对象会恢复到之前的选择，取消选择，或 Sniff 状态。关于 Volumn 恢复行为，参见 3.2.4。

Volume 地址在经过所有 reset 命令(包括 Reset(FFh))，后仍然会被保持。图 110 定义了 Volumn Select 命令的行为和时序。表 99 定义了 Volume 地址字段，其作为命令的一部分。

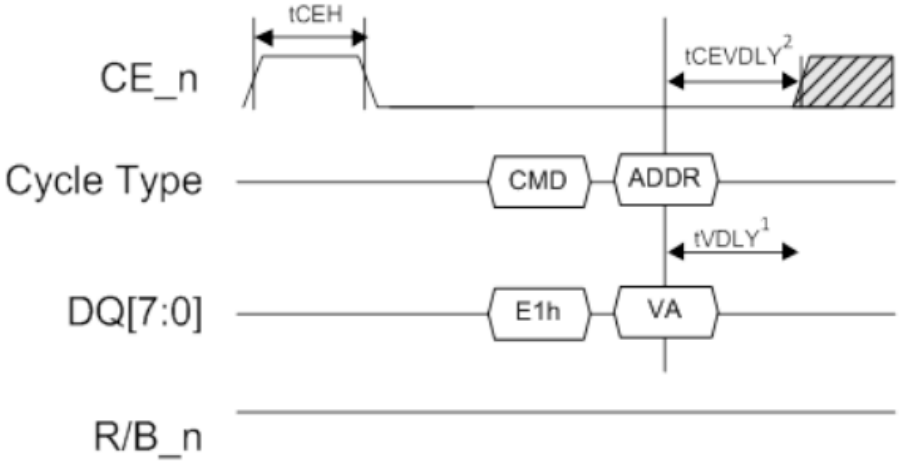


Figure 110　Volume Select timing diagram

注意：

1. 在经过一段 tVDLY 时间之前，host 不能向任何 Volume 的任何 LUN 发送新的命令。该延迟用以确保为下

一个发送的命令选择合适的 Volume。

2. 在经过一段 tCEVDLY 时间之前，host 不能将任何 Volumn 的 CE_n 拉高。这延迟用以确保之前发送的 Volume Select 命令选择了正确的 Volumn。

Volume Address	7	6	5	4	3	2	1	0
VA	Reserved (0)				Volume Address			

Table 99 Volume Address

Volume Address	Specifies the Volume to select.

5.25 ODT 配置定义

ODT 配置命令用来在使用 matrix termination 时配置 on-die termination。ODT 配置命令确定一个特定的 LUN 是否是一个 Volume 和 Rtt 设置的 terminator。如果 LUN 被确定作为一个或多个 Volumn 的一个 terminator，当数据输入或数据输出周期在作为 terminator 的 Volumn 上被执行时，LUN 应该使能 on-die termination。

如果 ODT 配置命令被用来为任何 LUN 指定 Rtt 设置，则该命令应该被用来为所有 Volume 的 LUN 确定 Rtt 设置。在这种情况下，ODT 配置命令应该被发送到每个 Volumn 的至少一个 LUN。当 ODT 配置命令被发送到一个 Volumn 上的至少一个 LUN 时，则该 Volumn 应该开始为该 Volumn 上的所有 LUN 使用 ODT Confituration Matrix。ODT Matrix 的默认值是 0000h，比如，termination 被 disable。

当在 NV-DDR2 或 NV-DDR3 接口中发送 ODT 配置命令时，每个数据 byte 被传输两次，device 只会所存每个数据 byte 的其中一个备份。当使用 NV-DDR 接口时不能发送该命令。

当该命令被发送且 NV-DDR2 或 NV-DDR3 接口被使能，则更新过的 termination 设置会立即生效。当改变这些设置时，host 应该小心以避免任何信号完整性问题。当 NV-DDR2 接口被使能后该命令被发送，则推荐转换到 SDR 接口，对 termination 设置进行正确的更改，之后在转换回 NV-DDR2 接口。如果 NV-DDR3 接口被使能后该命令被发送，则推荐转换到一个较慢的时序模式，对 termination 设置进行正确的更新，之后再转换回较快的 NV-DDR3 时序模式。

On-die termination 的设置在所有 reset 命令(包括 Reset(FFh))命令期间都会保持不变。图 111 定义了 ODT 配置命令的行为和时序。

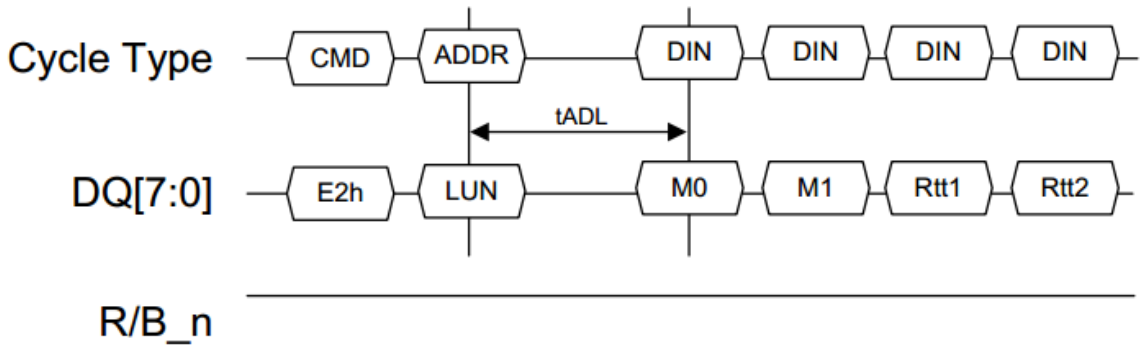


Figure 111 ODT Configure timing diagram

LUN 确定作为 terminator 的 LUN。该字段的格式和含有 LUN 地址的行地址格式相同，参见 3.1 章。

M0 ODT configuration matrix 的低 byte。

M1 ODT configuration matrix 的高 byte。

Rtt1 DQ[7:0]/DQS 的 termination 设置

Rtt2 RE_n 的 termination 设置

R 保留的(0h)

表 100 定义了作为 ODT 配置命令一部分的 ODT Configuration Matrix(ODT 配置矩阵)。如果一个 bit 被设为 1，则 LUN 应作为对应 Volumn(Vn)的 terminator，n 对应于 Volume 地址。

Volume Address	7	6	5	4	3	2	1	0
M0	V7	V6	V5	V4	V3	V2	V1	V0
M1	V15	V14	V13	V12	V11	V10	V9	V8

Table 100 ODT Configuration Matrix

表 101 定义了作为 ODT 配置命令一部分的 on-die termination 设置，包括 DQ[7:0],DQS_t,DQS_c,RE_t 和 RE_c 的 Rtt 值。

Rtt Settings	7	6	5	4	3	2	1	0
Rtt1	DQ[7:0]/DQS Rtt & ODT Enable for Data Output				DQ[7:0]/DQS Rtt & ODT Enable for Data Input			
Rtt2	Reserved				RE_n Rtt & ODT Enable			

Table 101 On-die Termination Settings

DQ[7:0]/DQS Rtt & ODT Enable for Data Input

This field controls the on-die termination settings for the DQ[7:0], DQS_t and DQS_c signals for data input operations (i.e. writes to the device).
The values are:
0h = ODT disabled
1h = ODT enabled with Rtt of 150 Ohms
2h = ODT enabled with Rtt of 100 Ohms
3h = ODT enabled with Rtt of 75 Ohms
4h = ODT enabled with Rtt of 50 Ohms
5h = ODT enabled with Rtt of 30 Ohms (Optional)
6h-Fh Reserved

DQ[7:0]/DQS Rtt & ODT Enable for Data Output

This field controls the on-die termination settings for the DQ[7:0], DQS_t and DQS_c signals for data output operations (i.e. reads from the device).
The values are:
0h = ODT disabled
1h = ODT enabled with Rtt of 150 Ohms
2h = ODT enabled with Rtt of 100 Ohms
3h = ODT enabled with Rtt of 75 Ohms
4h = ODT enabled with Rtt of 50 Ohms
5h = ODT enabled with Rtt of 30 Ohms (Optional)
6h-Fh Reserved

RE_n Rtt & ODT Enable

This field controls the on-die termination settings for the RE_t and RE_c signals.
The values are:
0h = ODT disabled
1h = ODT enabled with Rtt of 150 Ohms
2h = ODT enabled with Rtt of 100 Ohms
3h = ODT enabled with Rtt of 75 Ohms
4h = ODT enabled with Rtt of 50 Ohms
5h = ODT enabled with Rtt of 30 Ohms (Optional)
6h-Fh Reserved

5.26 ZQ Calibration Long(ZQCL)

ZQCL 命令在上点初始化或 reset 过程中用来执行初始校正。向命令寄存器写 F9h，随后跟着一个含有 LUN 地址的行地址周期，该序列用来在选定的 die 上执行 ZQCL。该命令可在任何时间被控制器发送，取决于系统环境。ZQCL 命令会触发并启动 NAND 内部的校正引擎。校正完成之后，校正值被从校正引擎传输到 NAND I/O，反映为更新过的 RON 和 Rtt 值。

在 ZQCL 操作期间，正在执行 ZQCL 操作的 NAND device 上不能有任何阵列操作。与正在执行 ZQCL 操作的 NAND device 共用 ZQ 信号的任何其他 NAND device 是允许执行阵列操作的。

NAND 允许一个由 tZQCL 确定的时间窗口来执行完整的校正并传输校正值。当 ZQCL 被发送后，时序参数 tZQCL 必须被满足。

当 ZQCL 操作完成后，host 应该检查状态。如果 FAIL 位被置起(SR[0]=1)，则表示校正过程失败，用户应该检查 RZQ 电阻的连接。如果 ZQCL 操作失败，device 将恢复到制造商定义的值。如果在 ZQCL 操作期间执行了 RESET 操作，则 NAND device 的输出驱动强度和 ODT 值会恢复出厂设置(就如没有执行 ZQ 校正一样)。

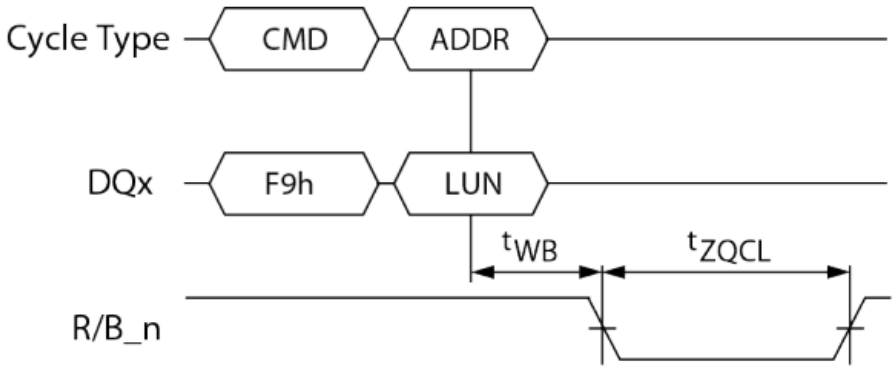


Figure 112 ZQ Calibration Long

5.27 ZQ Calibration Short(ZQCS)

ZQCS 命令用来执行定期的校正以应对电压和温度的微小变化。向命令寄存器写 D9h，随后跟着一个含有 LUN 地址的行地址周期，这个序列用来在选定的 die 上执行 ZQCS。一个由时序参数 tZQCS 确定的较短的时间窗口用来执行简化的校正并传输校正值。一个 ZQCS 命令可以在 tZQCS 时间内有效修正一个最小为 1.5% RON 和 Rtt 的阻抗错误，假设最大敏感度为表 55 和表 70 中规定的值。

在 ZQCS 操作期间，正在执行 ZQCS 操作的 NAND device 上不能有任何阵列操作。与正在执行 ZQCS 操作的 NAND device 共用 ZQ 信号的任何其他 NAND device 都是允许执行阵列操作的。

当 ZQCS 操作完成后，host 应该检查状态。如果 FAIL 位被置起(SR[0]=1)，则表示校正过程失败，用户应该遵循制造商规定的指示。如果 ZQCS 操作失败，则 device 会恢复到制造商定义的值。

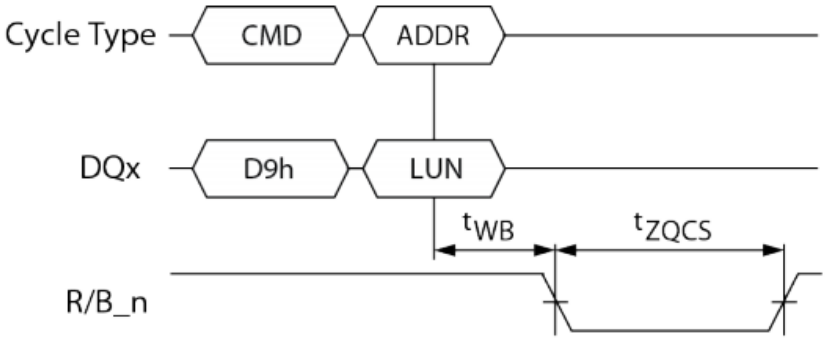


Figure 113 ZQ Calibration Short

5.28 Set Feature 定义

Set Feature 命令用来更改一个特定功能的设置。例如，该命令可用来使能一个在上电阶段被 disable 的特性。命令参数始终在数据总线的低 8-bit 上传输。图 114 定义了 Set Feature 的行为和时序。

当在 NV-DDR、NV-DDR2 或 NV-DDR3 接口中发送 Set Feature 命令时，每个数据 byte 被传输两次。Device 只会锁存每个数据 byte 的其中一个备份，参见 4.4 章。

Set Feature 用来改变时序模式和接口类型。当改变时序模式时，device 的 busy 为 tITC 而不是 tFEAT。在 tITC 时间内，host 不能读状态(poll for status)。

LUN Set Feature(D5h)命令的功能和对象级 Set Feature(EFh)命令相同，除了只有被寻址的 LUN 的设置被改变。假设本文档之前提到的 Set Feature 命令的功能都和 LUN Set Feature 命令相同，除非有另外说明。

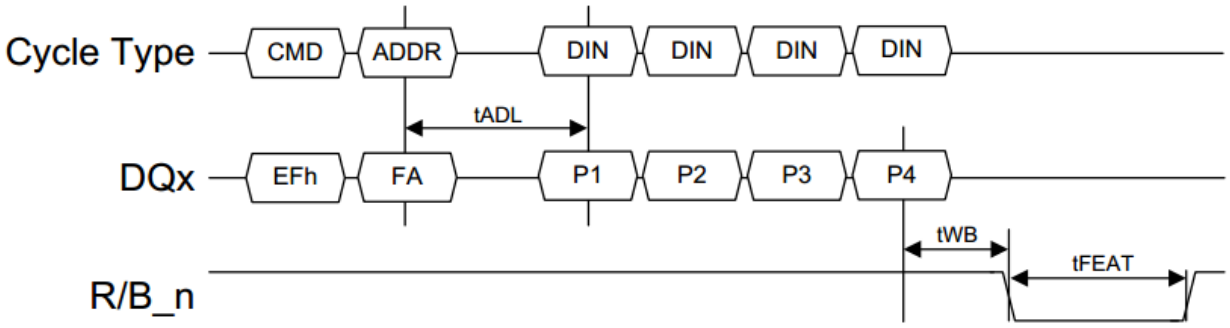


Figure 114 Set Features timing

*** NOTE:** Busy time is tITC when setting the timing mode.

FA Feature Address，用来识别要改变设置的 feature

P1-P4 为指定的 feature 识别新设置的参数

P1 子特性参数 1

P2 子特性参数 2

P3 子特性参数 3

P4 子特性参数 4

参见 5.30 章关于特性和子特性参数的定义。

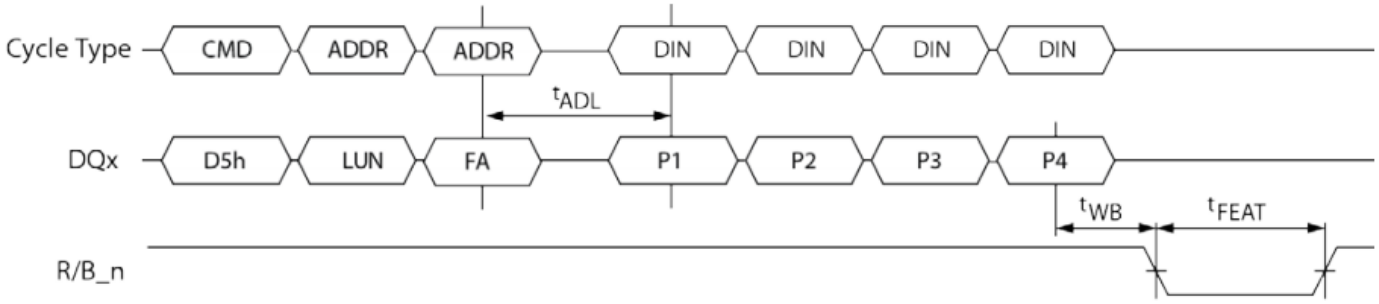


Figure 115 LUN Set Features timing

*** NOTE:** Busy time is tITC when setting the timing mode.

LUN LUN 地址。LA0=bit0,LA1=bit1,LA2=bit2。（例如 LUN 0 = 0oh,LUN 1 = 01h）

FA Feature Address，用来识别要改变设置的 feature

P1-P4 为指定的 feature 识别新设置的参数

P1 子特性参数 1

P2 子特性参数 2

P3 子特性参数 3

P4 子特性参数 4

参见 5.30 关于特性和子特性参数的定义。

5.29 Get Feature 定义

Get Feature 是一种 Host 用来判定特定特性的当前设置的机制。该命令会返回特性的当前设置(包括由之前 Set Feature 命令做的修改)。该命令的参数始终在暑假总线的低 8-bit 上传输。Host 在读取数据的第一个 byte 之后，在发送其他命令之前(包括读状态或 Read Status Enhanced)应完成所有期望数据的读取。图 116 定义了 Get Feature 命令的行为和时序。

当在 NV-DDR、NV-DDR2 或 NV-DDR3 接口中发送 Get Feature 命令时，每个数据 byte 被接收两次，host 只会锁存每个数据 byte 的其中一个备份，参见 4.4 章。

如果使用读状态命令来检查 tFEAT 时间何时结束，则 host 应发送命令 00h 来开始传输特性数据(从参数 P1 开始传输)。

LUN Get Feature(D4h)命令的功能和对象级命令 Get Feature 相同，除了只有被寻址的 LUN 的设置被返回。假设本文档中之前提到的 Get Feature 命令的功能都和 LUN Get Feature 相同，除非特别说明。

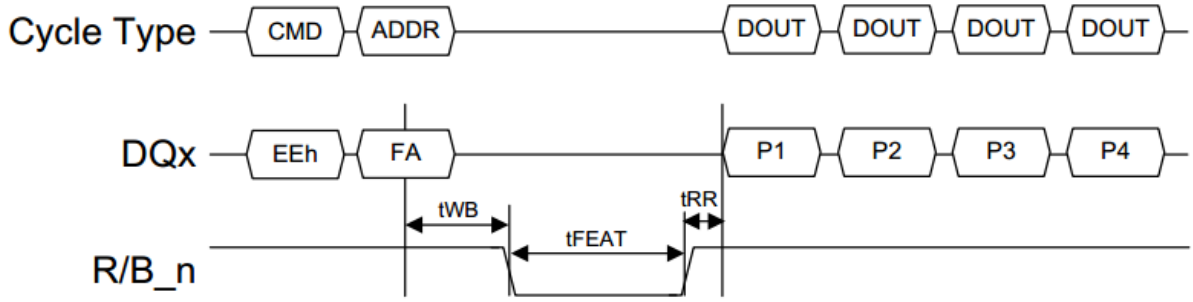


Figure 116 Get Features timing

FA Feature address identifying feature to return parameters for.

P1-P4 Current settings/parameters for the feature identified by argument P1

P1	Sub feature parameter 1 setting
P2	Sub feature parameter 2 setting
P3	Sub feature parameter 3 setting
P4	Sub feature parameter 4 setting

Refer to section 5.30 for the definition of features and sub feature parameters.

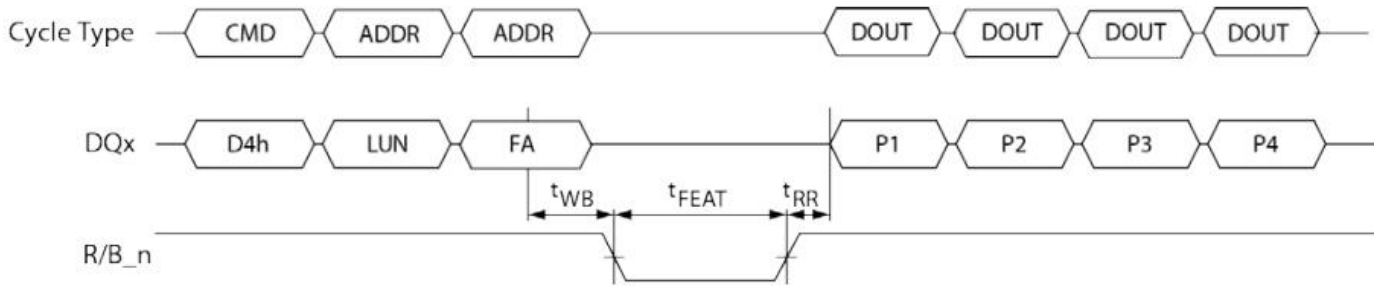


Figure 117 LUN Get Features timing

LUN LUN Address. LA0 = bit 0, LA1 = bit1, LA2 = bit 2. (i.e. LUN 0 = 00h, LUN 1 = 01h, etc.)

FA Feature address identifying feature to return parameters for.

P1-P4 Current settings/parameters for the feature identified by argument P1

P1 Sub feature parameter 1 setting
P2 Sub feature parameter 2 setting
P3 Sub feature parameter 3 setting
P4 Sub feature parameter 4 setting

Refer to section 5.30 for the definition of features and sub feature parameters

5.30 特性参数(Feature Parameter)定义

如果对象不支持 Set Feature 和 Get Feature 命令，那么也不会支持特性参数。另外，对象只支持其遵循的 ONFI 规范版本中定义的特性参数。

特性设置对电源周期是易失的，对于每种特性设置，该特性在复位后能否被保持会有明确的说明。

特性地址 (Feature Address)	描述
00h	保留
01h	时序模式
02h	NV-DDR2/NV-DDR3 配置
03h-0Fh	保留
10h	I/O 驱动强度
11h-2Fh	保留
30h	外部 Vpp 配置
31h-4Fh	保留
50h	EZ NAND 控制
51h-57h	保留
58h	Volume 配置
59h-5Fh	保留
60h	BA NAND：错误信息
61h	BA NAND：配置
62h-7Fh	保留
80h-FFh	制造商定义

5.30.1 时序模式

如果对象符合 ONFI 规范 V1.0，则该设置应该被支持。

当 SDR、NV-DDR 或 NV-DDR2 接口被使能，则数据接口的设置和时序模式的编号在 Reset(FFh)后不会被保持；Reset(FFh)之后，数据接口应该是 SDR，时序模式应该是模式 0。所有其他时序模式的设置在经过 Reset(FFh)、同步 Reset(FCh)以及 Reset LUN(FAh)之后都会被保持。如果当接口被配置为 NV-DDR 或 NV-DDR2 接口时发送了 Reset(FFh)命令，则 host 应使用 SDR 接口及时序模式 0，直到新的接口和/或时序模式被 Set Feature 命令选中。Host 只能设置在参数页中明确表明支持的时序模式。

Host 使用 Set Feature 命令来将接口从 NV-DDR 或 NV-DDR2 转为 SDR，其结果是未知的，为了从这两种接口转换到 SDR 接口，host 应该使用 Reset(FFh)命令。

当 VccQ=1.2V 时，NV-DDR3 接口被默认使能。不能从 NV-DDR3 接口直接转换到 SDR、NV-DDR 或 NV-DDR2 接口。如果子特性参数 P1 中 bit[4:5]“数据接口”被改变，device 将保持在 NV-DDR3 接口模式。如果当接口是 NV-DDR3 时发送了 Reset(FFh)命令，则时序模式的所有设置将会保持不变，而且 host 会继续使用 NV-DDR3 接口。Host 发送 Reset(FFh)命令之后，推荐使用时序模式 0，直到 host 将 device 重新配置为更快的时序模式。

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	R	PC	Data Interface		Timing Mode Number			
P2	Reserved (0)							
P3	Reserved (0)							
P4	Reserved (0)							

Timing Mode Number　设置 host 使用的时序模式编号。默认上电编号为 0h。

Data Interface　00b=SDR(上电默认值)
01b=NV-DDR
10b=NV-DDR2
11b=保留(数据接口不支持 NV-DDR3)

PC　编程清除(Program Clear)位，控制 program page register clear enhancement，定义了当接收到编程(80h)命令时清除页寄存器的行为。如果被清为 0，则当接收到编程(80h)命令时，对象中每个 LUN 的页寄存器都会被清除。如果设为 1，则只有被编程(80h)命令选中的 LUN 和 interleave address 的页寄存器会被清除，并且编程命令的 tADL 时间为参数页中的值。

Reserver/R　保留值，应被 host 清为 0。对象不会对保留字段的值敏感。

5.30.2 NV-DDR2 和 NV-DDR3 配置

如果对象支持 NV-DDR2 或 NV-DDR3 接口，则该设置也应该支持。该设置控制数据输入和输出的差分信号，基本的 on-die termination 配置，以及 warmup 周期。对于 NV-DDR2 接口，以上设置在 Reset(FFh)后不会被保持，并且上电值会恢复为 0。对于 NV-DDR3 接口，这些设置在 Reset(FFh)后会保持不变。对于 NV-DDR2 和 NV-DDR3 接口，这些设置在同步 Reset(FCh)和 Reset LUN(FAh)后会保持不变。上电后所有字段的默认值都为 0h。

NV-DDR2 接口应该在这些设置的时序模式特性中使能生效。建议在使能 NV-DDR2 接口前先配置这个功能(意思是指本章节描述的 NV-DDR2 和 NV-DDR3 的配置)。

当时序模式特性更改后，NV-DDR2 或 NV-DDR3 接口被使能，则更新过的设置会立即生效。在 NV-DDR2 或 NV-DDR3 被使能期间，当 host 更改这些设置时应小心，以避免任何信号完整性问题。如果在 NV-DDR2 被使能期间发生问题，则建议转换到 SDR 接口，更改合适的该配置(指 NV-DDR2/3 配置)，然后再转换回 NV-DDR2 接口。如果在 NV-DDR3 被使能期间发生问题，则建议转换到时序模式 0，更改合适的该配置(指 NV-DDR2/3 配置)，然后再转换回期望的时序模式。如果这些设置被更改，则 host 应小心，以确保正确的设置以适当的方式被应用，以避免信号完整性问题。

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	DQ/DQS/RE_n ODT Enable				R	CMPR	CMPD	VEN
P2	Warmup DQS cycles for Data Input				Warmup RE_n and DQS cycles for Data Output			
P3	Reserved							
P4	Reserved							

VEN　　如果设为 1，则使用外部 VREFQ 作为输入和 I/O 信号的参考。如果被清为 0，则不使用 VREFQ 作为任何信号的输入参考。CE_n 和 WP_n 是 CMOS 信号。设计可以为 WE_n、ALE 或 CLE 使用 CMOS 或 SSTL_18(NV-DDR2)，或 NV-DDR3 输入级别。对于其他所有信号，包括 DQ[7:0],DQS_t 以及 RE_t，使用 SSTL_18(NV-DDR2)或 NV-DDR3 输入级别，不管 VREFQ 怎么配置。如果 VEN=0，则使用 VccQ/2(不是 VREFQ)作为参考。注意：当 CMPD、CMPR 被清 0 时，RE_t 和 DQS_t 仅使用 VREFQ。

CMPD　　如果设为 1，则互补信号 DQS(DQS_c)被使能。如果被清为 0，则不使用互补信号 DQS(DQS_c)。

CMPR　　如果被设为 1，则互补信号 RE_n(RE_c)被使能。如果被清 0，则不使用互补信号 RE_n(RE_c)。

DQ/DQS/RE_n ODT Enable

该字段控制 DQ[7:0],DQS_t,DQS_c,RE_t 以及 RE_c 信号的 on-die termination 设置。值为：

0h = ODT disable
1h = ODT 被使能，Rtt=150 Ohms
2h = ODT 被使能，Rtt=100 Ohms
3h = ODT 被使能，Rtt=75 Ohms
4h = ODT 被使能，Rtt=50 Ohms
5h = ODT 被使能，Rtt=30 Ohms(可选)

注意：Rtt 的设置可以为 DQ[7:0]/DQS 以及 RE_n 信号分别规定。DQ[7:0]/DQS 可以被分别规定为数据输入和数据输出操作，参见 5.25 章 ODT 配置命令定义。如果值被通过 ODT 配置命令确定，则该字段不被使用。Get Feature 会返回该字段之前设置的值，无论使用 ODT 配置命令确定的 Rtt 设置。

数据输出的 Warmup RE_n 和 DQS 周期

该字段表示用于数据输出的 RE_n 和 DQS 的 warmup 周期数。这些值即数据输出开始时初始“dummy”RE_t/RE_c 周期数。Host 应忽烈与“dummy”RE_t/RE_c 周期对应的“dummy”DQS_t/DQS_c 周期。值定义如下：

0h = 0 周期，特性被 disable
1h = 1 warmup 周期
2h = 2 warmup 周期
3h = 4 warmup 周期
4h-FFh = 保留的

数据输入的 warmup DQS 周期

该字段表示用于数据输入的 DQS 的 warmup 周期数。这些值即数据输入操作开始时的初始“dummy”DQS_t/DQS_c 周期数。值定义如下：

0h = 0 周期，特性被 disable
1h = 1 warmup 周期
2h = 2 warmup 周期
3h = 4 warmup 周期
4h-FFh = 保留的

Reserved　　保留值，应该被 host 清 0。对象不会对保留字段值敏感。

5.30.3 I/O 驱动强度

如果对象支持 NV-DDR、NV-DDR2 或 NV-DDR3 接口，则该设置也应该被支持。I/O 驱动强度设置在 Reset(FFh)、同步 Reset(FCh)以及 Reset LUN(FAh)命令后应保持不变。对于 NV-DDR、NV-DDR2 和 NV-DDR3 接口，上电后默认驱动强度值为 35 Ohm(10b)。

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	Reserved (0)						Drive Strength	
P2	Reserved (0)							
P3	Reserved (0)							
P4	Reserved (0)							

Drive strength
00b: 18 Ohm (Optional, not supported for NV-DDR3)
01b: 25 Ohm (Optional)
10b: 35 Ohm (power-on default)
11b: 50 Ohm

Reserved　Reserved values shall be cleared to zero by the host. Targets shall not be sensitive to the value of reserved fields.

5.30.4 外部 Vpp 配置

如果参数页中表明对象支持外部 Vpp，则该设置也应该被支持。该设置用来控制外部 Vpp 的使能，在 Reset(FFh)、同步 Reset(FCh)以及 Reset LUN(FAh)命令后该设置会保持不变。该设置中所有字段的上电默认值都为 0h。

Vpp 必须在 Set Feature 命令使能 Vpp 之前有效。

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	Reserved (0)							Vpp
P2	Reserved (0)							
P3	Reserved (0)							
P4	Reserved (0)							

Vpp　0b = External Vpp is disabled
1b = External Vpp is enabled

Reserved　Reserved values shall be cleared to zero by the host. Targets shall not be sensitive to the value of reserved fields.

5.30.5 Volume 配置

该设置用来配置 Volume 地址。如果参数页中表明对象支持 Volume 寻址，则该设置应该被支持。在 Volume 地址被指定之后，对应的 Volume 的 ENo 引脚被设为 1，ENi 引脚被忽略，直到下一个电源周期后，Volume 被取消选择，直到发送一个 Volume Select 命令来选择相关的 Volume。

Host 对每个 Volume 在每个电源周期只能执行一次 Volume 配置，设定的地址随后被 Select Volume 命令用来访问这个 NAND 对象。该设置在 Reset(FFh)、同步 Reset(FCh)以及 Reset LUN(FAh)命令后会保持不变。该设置没有上电默认值。

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	Reserved				Volume Address			
P2	Reserved							
P3	Reserved							
P4	Reserved							

Volume Address　Specifies the Volume address to appoint

5.30.6 EZ NAND 控制

如果 device 支持 EZ NAND，则也应支持该设置。该功能用来控制 EZ NAND devicd 的设置。

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	Reserved (0)							RD
P2	Reserved (0)							
P3	Reserved (0)							
P4	Reserved (0)							

Retry Disable(RD)　　如果设为 1，则 EZ NAND device 不能自动执行重试。如果被清 0，则在错误条件发生期间 EZ NAND device 可以自动执行重试。如果自动重试被 disable，则 device 可能超出规定的 UBER。只有当参数页标明支持 disable 自动重试时，该功能才能被 disabe。如果一个 EZ NAND 控制器在执行一个自动重试，则典型的页读时间(tR)可能被超过。

Reserved/R　　保留值，应该被 host 清为 0。对象不会对保留字段的值敏感。

6. 多层操作(Multi-plane Operations)

一个 LUN 可以支持多层读、编程以及擦除操作。多层操作是指，相同类型的多个命令被发送到同一个 LUN 上的不同块上。参见 5.7.1.29 关于多层操作的寻址限制。多层操作有两种方法：并发的和重叠的(overlapped)。

当执行多层操作时，执行的操作/功能应该是相同类型。可以在多层操作中执行的功能有：

- 页编程
- 回拷读和编程
- 块擦除
- 读

6.1　要求

如果支持多层操作，则层地址包含如图 33 所示的块地址的最低位。LUN 和页地址要相同。块地址(除了层地址位-plane address bits)也可能要求要相同，参见 5.7.1.29。

对于回拷编程操作，其限制和一个多层编程操作相同。但是，要先发送一个回拷读命令到层地址，该层地址和多层回拷编程操作中的层地址相同。回拷的读可以以多层或非多层的方式发送。如果读是非多层的，则读操作可能有不同的页地址。如果读是多层的，则读应该有相同的页地址。

多层操作允许相同类型的操作被发送到同一个 LUN 上的其他块上。多层操作有两种方法：并发的和重叠的。并发的多层操作在访问闪存阵列之前，会等待，直到所有层地址的所有命令、地址和数据都开始。重叠多层操作在命令、地址和数据都开始后立即开始执行其操作，在此期间可以开始下一个多层操作的命令、地址及数据。

每个地址的层地址组成(plane address component)都应该是不同的。一个单个的多层(cached)编程操作如图 118 所示。在“Multi-plane Op 1”和“Multi-plane Op n”之间的所有层地址都应该是互不相同的。在 10h 或 15h(cached)命令周期被发送后，之前发送的层地址可以被用在后面的多层操作。

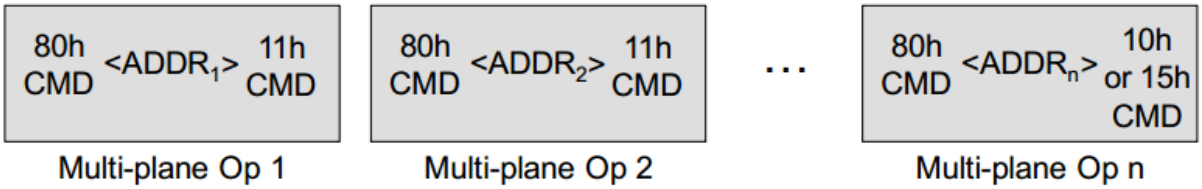


Figure 118　　Multi-plane Program (Cache)

对于多层擦除操作，每个地址的层地址组成都应该是不同的。一个单个的多层擦除操作如图 119 所示。在“Multi-plane Op 1”和“Multi-plane Op n”之间的所有层地址都应该互不相同。在 D0h 命令周期被发送后，之前发送的层地址可以被用在后面的多层操作中。

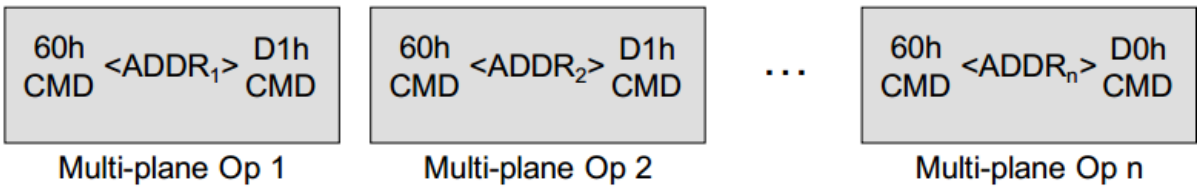


Figure 119 Multi-plane Erase

对于多层读操作，每个地址的层地址组成都应该是不同的。一个单个的多层读(cache)操作如图 120 所示。在“Multi-plane Op 1”和“Multi-plane Op n”之间的所有层地址都应该互不相同。在 30h 或 31h(cached)命令发送后，之前发送的层地址可用于后面的多层操作中。

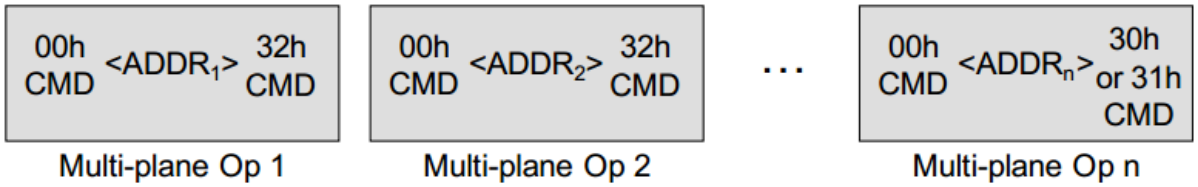


Figure 120 Multi-plane Read (Cache)

6.2 状态寄存器行为

某些状态寄存器位在每个层地址之间是独立的。其他状态寄存器位都是被整个 LUN 共用的。该章节定义了这些在层地址之间独立的寄存器位。对于并发和重叠操作，这些定义都是相同的。

对于多层编程和擦除操作，FAIL/FAILC 位对每个层地址是独立的。表 102 列出了寄存器每个位在多层操作中是独立还是被整个 LUN 中是共用的。

Value	7	6	5	4	3	2	1	0
Status Register	WP_n	RDY	ARDY	VSP	R	R	FAILC	FAIL
Independent	N	N	N	N	N	N	Y	Y

Table 102 Independent Status Register bits

6.3 多层页编程

页编程命令把被列地址指定的一个页或部分页中的数据传输到页寄存器。之后，页寄存器的内容被编程到行地址指定的闪存阵列中。使用多层操作可以将多个编程命令以很小的 busy 时间间隔连续地发送到 LUN 中。图 121 定义了两个多层编程命令的行为和时序。

如果参数页表明对象支持缓存(cache)操作，那当执行多层页编程操作时可以使用缓存(cache)操作。参见 5.7.1.28。

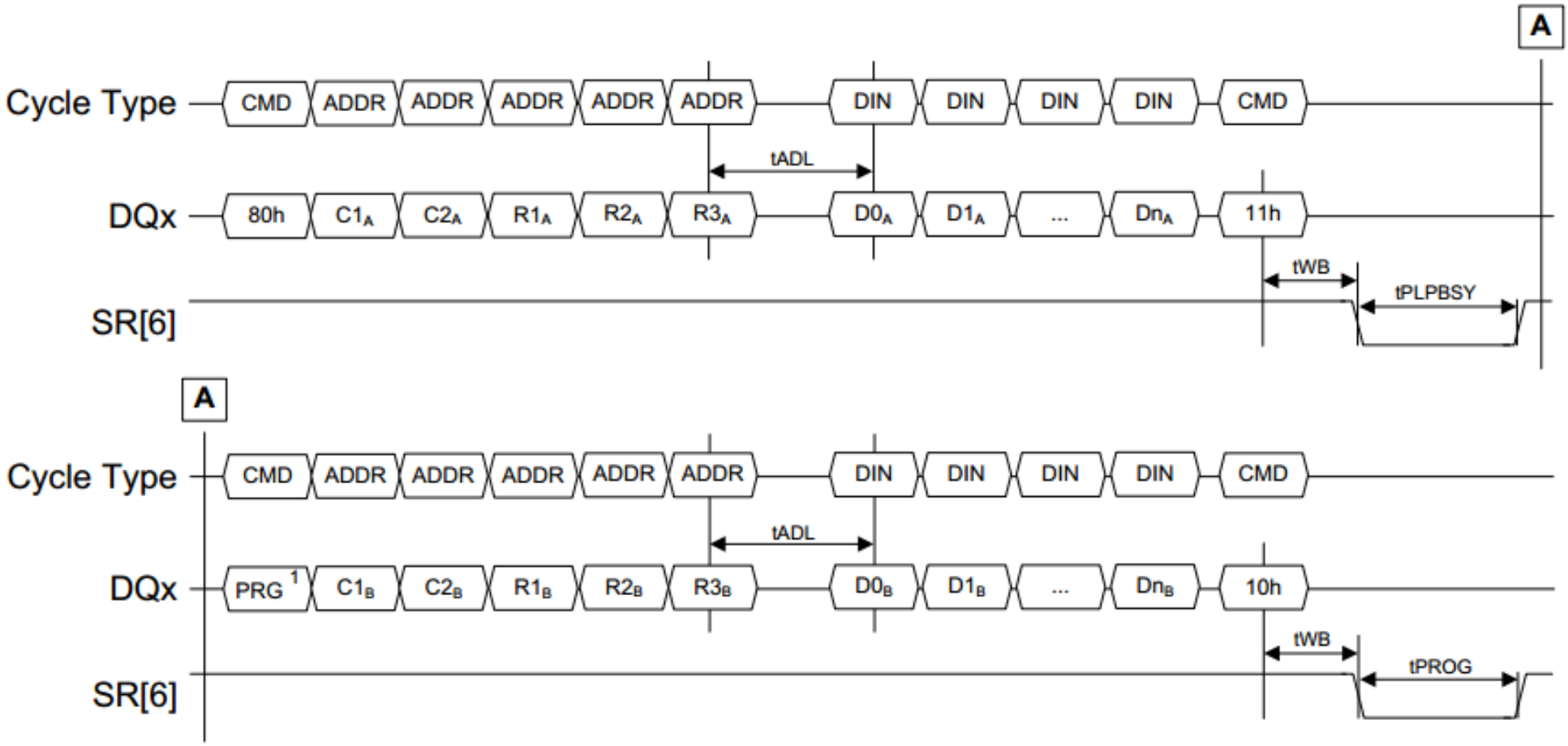


Figure 121 Multi-plane Page Program timing

注意：

1.有两种形式的多层页编程。ONFI 1.X 和 2.X 版本定义了多层页编程中所有编程序列的第一个周期(80h)。ONFI-JEDEC 组织定义了一个多层页编程中初始编程序列之后接下来的第一个周期(81h)。可参考参数页来判定 device 是否支持 81h。

$C1_A$-$C2_A$ 页 A 的列地址。$C1_A$ 是最低 byte。

$R1_A$-$R3_A$	页 A 的行地址。$R1_A$是最低 byte。
DO_A-Dn_A	页 A 中被编程的数据
PRG	80h 或 81h，参见注意 1。
$C1_B$-$C2_B$	页 B 的列地址，$C1_B$是最低 byte。
R1B-$R3_B$	页 B 的行地址，$R1_B$是最低 byte。
DO_B-Dn_B	页 B 中被编程的数据

页 A 和页 B 的行地址在层地址位中应该被区分开(不相同)。

使用 15h(而不是 10h)命令来结束一个多层编程操作，表示这是一个缓存(cache)操作。如果多层编程操作支持编程缓存，则 host 只能发送 15h 命令来结束多层编程操作，参见 5.7.1.28。

6.4 多层回拷读和编程

回拷功能从一个位置读取页数据，然后把该数据移动到第二个位置。使用多层操作可以将回拷编程命令以较短的 busy 时间连续发送到对象。图 122，图 123 以及图 124 定义了两个回拷编程操作的行为和时序。回拷编程的读可以是多层的，也可以不是。图 122 定义了非多层的读序列，而图 123 定义了多层的读序列。

回拷读操作(不管是否多层)使用的层地址应该和接下来的多层回拷编程操作使用的层地址相同。如果支持 EZ NAND，则不适用该限制；参见参数页。

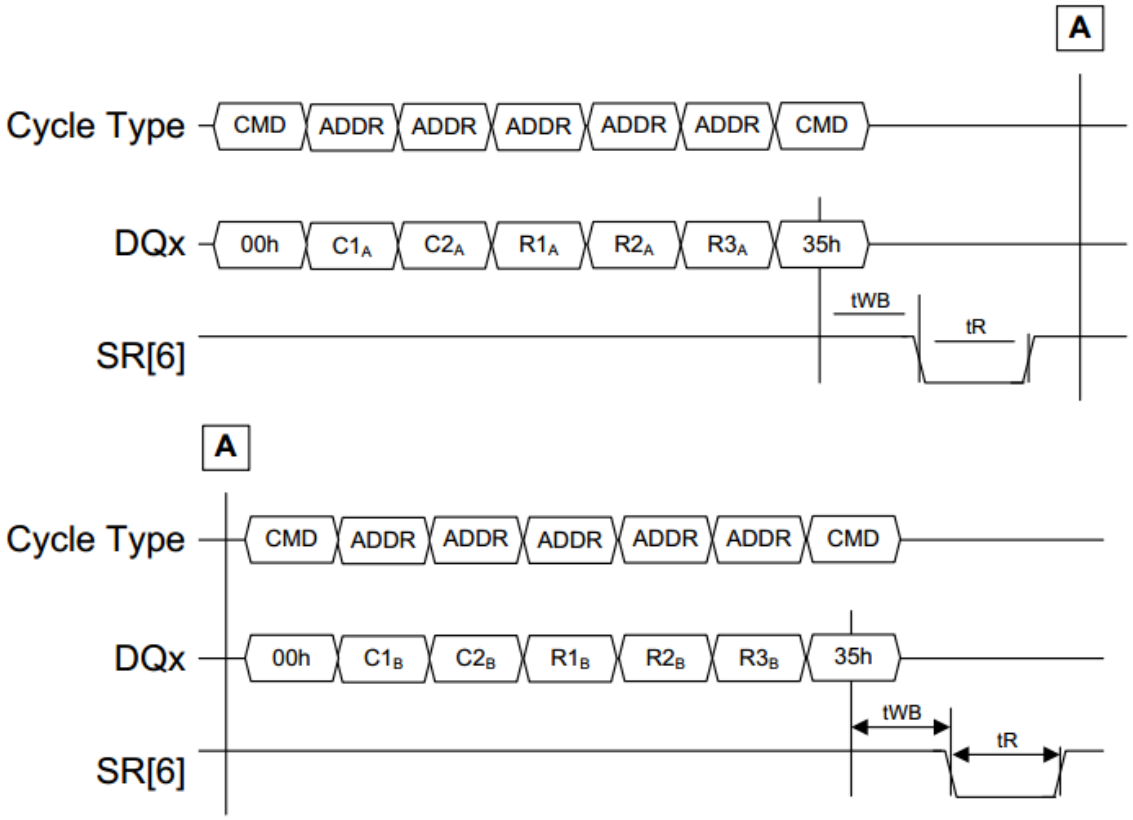


Figure 122　　Non-multi-plane Copyback Read timing for multi-plane Copyback Program

C1A-C2A　　源 pageA 的列地址，C1A 是最低 byte

R1A-R3A　　源 pageA 的行地址，R1A 是最低 byte

C1B-C2B　　源 pageB 的列地址，C1B 是最低 byte

R1B-R3B　　源 pageB 的行地址，R1B 是最低 byte

层地址位中所有的源 page 的行地址都应该不同。

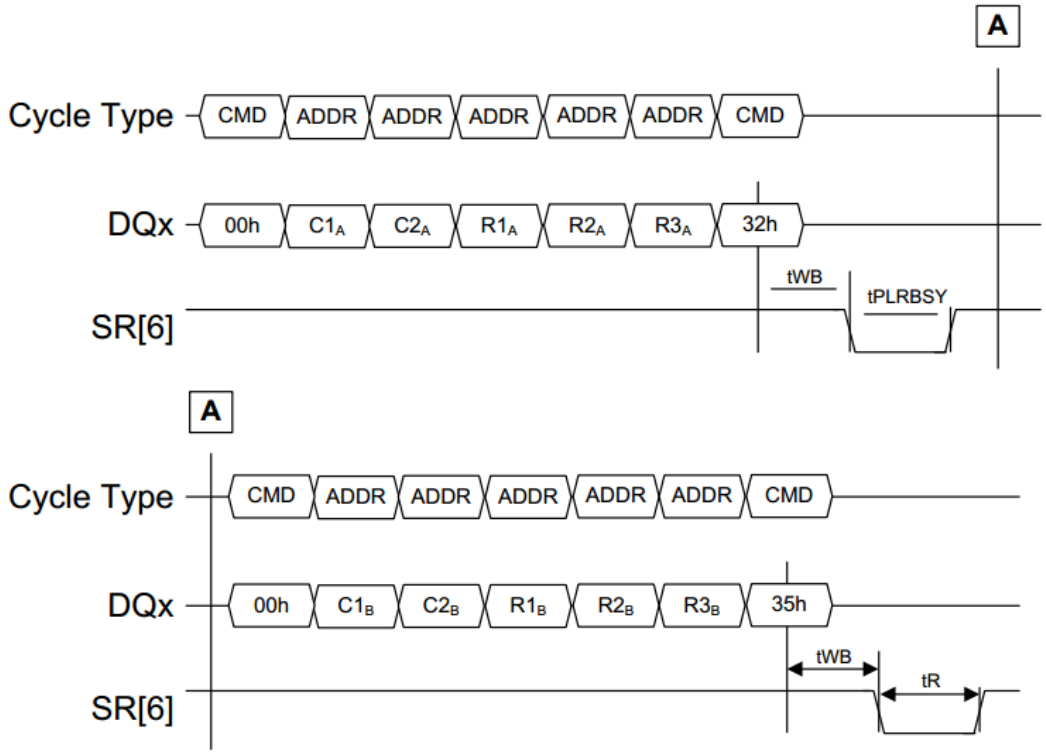


Figure 123 Multi-plane Copyback Read timing for Multi-plane Copyback Program

$C1_A$-$C2_A$	Column address for source page A. $C1_A$ is the least significant byte.
$R1_A$-$R3_A$	Row address for source page A. $R1_A$ is the least significant byte.
$C1_B$-$C2_B$	Column address for source page B. $C1_B$ is the least significant byte.
$R1_B$-$R3_B$	Row address for source page B. $R1_B$ is the least significant byte.

层地址位中所有的源 page 的行地址都应该不同。对于多层读，源 page 的地址应该相同。

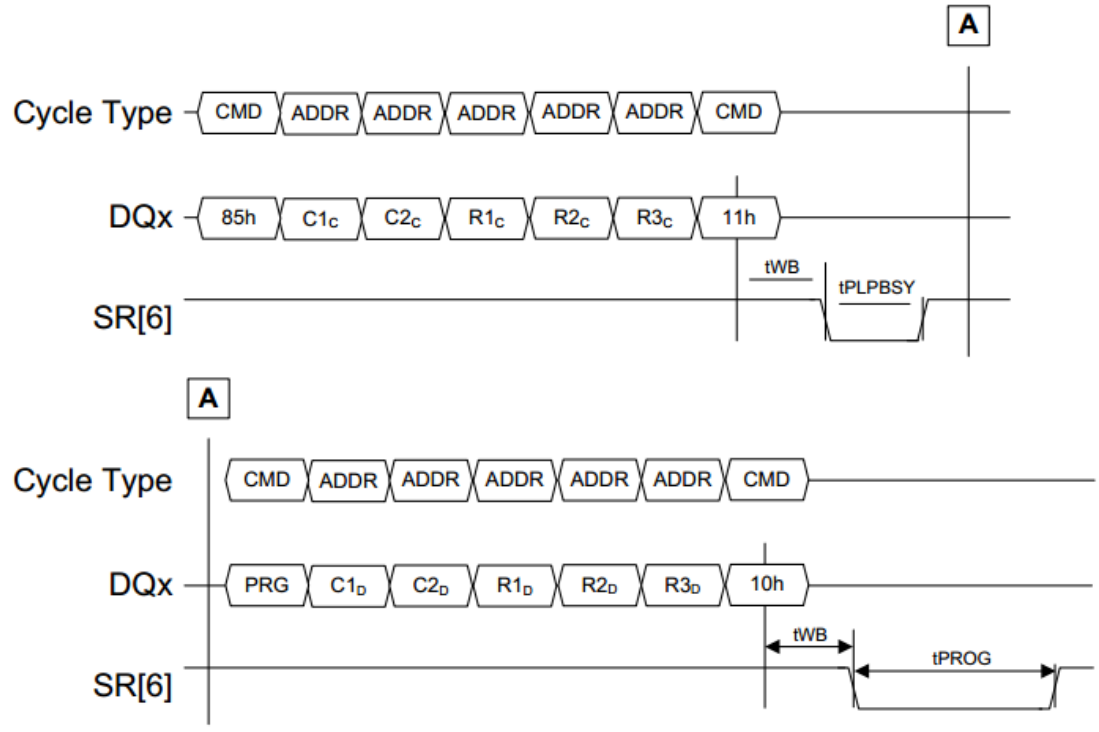


Figure 124 Multi-plane Copyback Program

Notes:

1. There are two forms of Multi-plane Copyback Program. ONFI 1.x and 2.x revisions have defined all first cycles for all program sequences in a Multi-plane Copyback Program as 85h. The ONFI-JEDEC Joint Taskgroup has defined the subsequent first cycles after the initial Copyback Program sequence in a Multi-plane Copyback Program as 81h. Refer to the parameter page to determine if the device supports subsequent first cycles in a Multi-plane Copyback Program sequence as 81h.

$C1_C$-$C2_C$	Column address for destination page C. $C1_C$ is the least significant byte.
$R1_C$-$R3_C$	Row address for destination page C. $R1_C$ is the least significant byte.
PRG	85h or 81h. Refer to Note 1.
$C1_D$-$C2_D$	Column address for destination page D. $C1_D$ is the least significant byte.
$R1_D$-$R3_D$	Row address for destination page D. $R1_D$ is the least significant byte.

The row addresses for all destination pages shall differ in their plane address bits. The page address for all destination addresses for multi-plane copyback operations shall be identical.

6.5 多层块擦除

图 125 定义了一个多层块擦除操作的行为和时序。只展示了两种操作，但是，在最终的 60h/D0h 序列之前可以发送另外的 60h/D1h 序列的擦除操作，这取决于 LUN 支持多少多层操作。

ONFI-JEDEC 组织定义了一个修改的多层块擦除操作，这个修改的操作中，指定另外要擦除的块的下一个行地址没有被 D1h 命令分隔开。该定义展示在图 126 中。可以参考参数页来判定 device 是否支持块地址之间没有 D1h 命令。

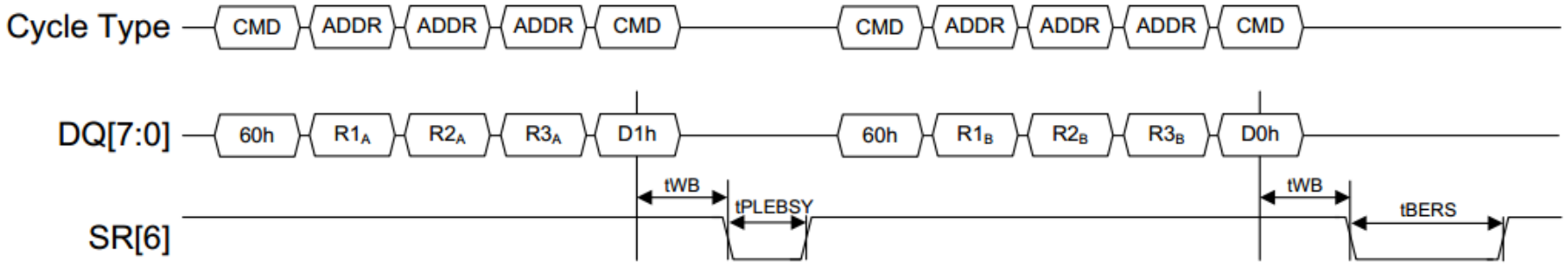


Figure 125 Multi-plane Block Erase timing

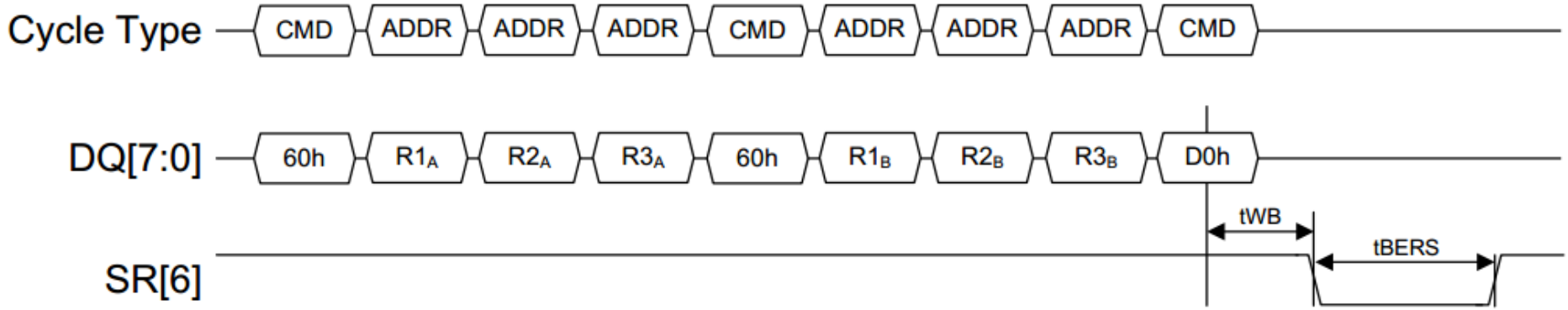


Figure 126 Multi-plane Block Erase timing, ONFI-JEDEC Joint Taskgroup primary definition

$R1_A$-$R3_A$	Row address for erase block A. $R1_A$ is the least significant byte.
$R1_B$-$R3_B$	Row address for erase block B. $R1_B$ is the least significant byte.

6.6 多层读

读命令用来读取指定的 LUN 中由行地址指定的一个页的数据。页数据可以从指定的列地址开始的页寄存器中读取。使用一个多层操作可以将多个读命令以较短的 busy 时间间隔连续发送到 LUN。图 127 定义了两个多层读命令的行为和时序。图 128 定义了在多层读命令准备好返回数据之后读数据的行为和时序。

如果参数页表明对象支持缓存(cache)操作，则当执行多层读操作时可以使用缓存(cache)操作，参见 5.7.1.28。

在从一个 LUN 读取数据之前，应该先发送 Change Read Column Enhanced 命令。如果没有发送 Change Read Column Enhanced 命令就读取了数据，则接收到的输出是不确定的。

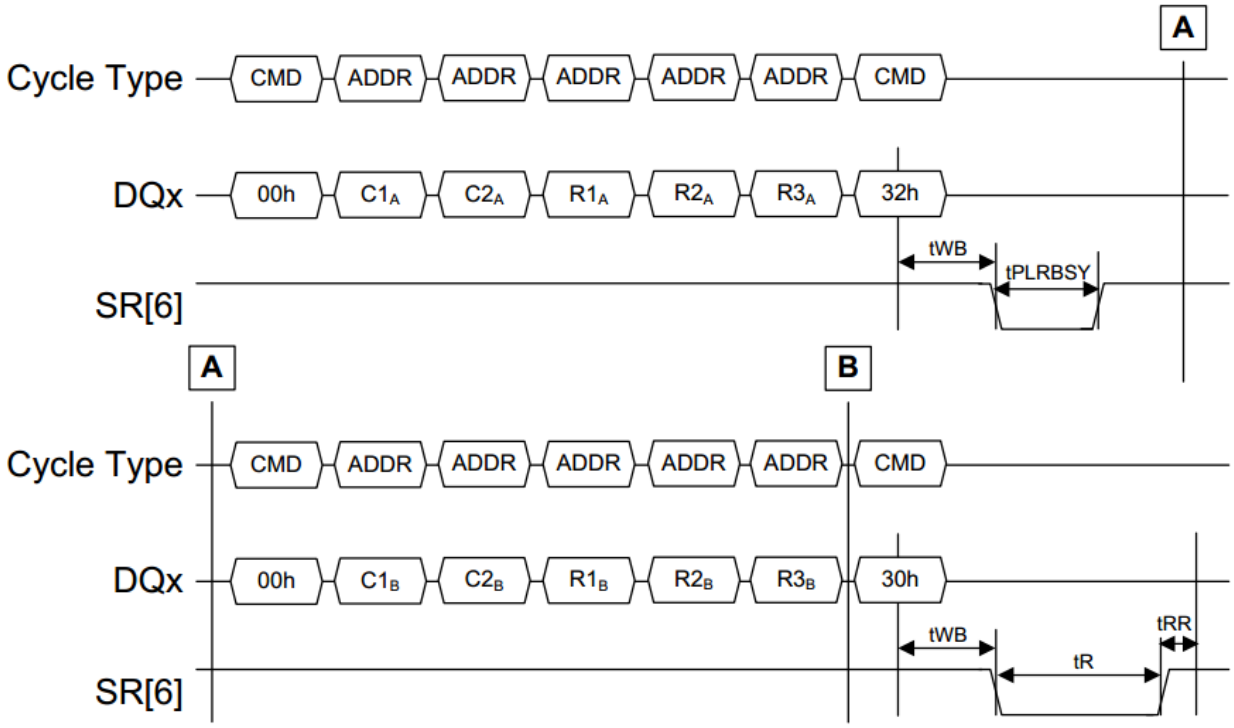


Figure 127 Multi-plane Read command issue timing

Figure 127 Multi-plane Read command issue timing

$C1_A$-$C2_A$	Column address for page A. $C1_A$ is the least significant byte.
$R1_A$-$R3_A$	Row address for page A. $R1_A$ is the least significant byte.
$C1_B$-$C2_B$	Column address for page B. $C1_B$ is the least significant byte.
$R1_B$-$R3_B$	Row address for page B. $R1_B$ is the least significant byte.

The row addresses for page A and B shall differ in the plane address bits.

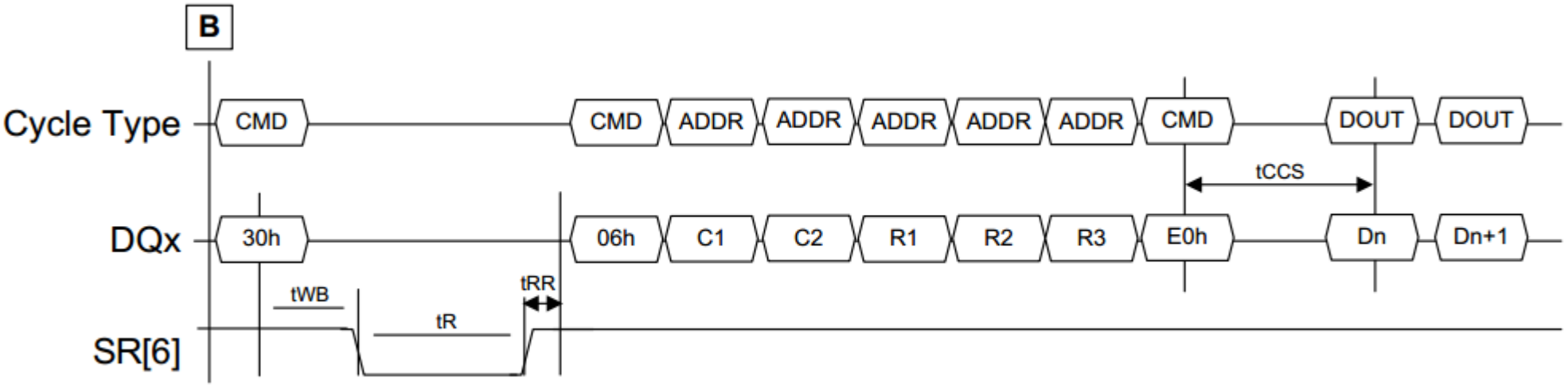


Figure 128 Multi-plane Read data output timing, continued from command issue

C1-C2	要读取的列地址，C1 是最低 byte
R1-R3	要读取的行地址(指定 LUN 和层地址)。R1 是最低 byte。
Dn	从寻址的行和列开始读取的数据 byte

行地址中应该指定具有有效读数据的 LUN 和层地址。

对于多层 Read Cache Sequential 操作，初始的多层读命令之后紧跟着一个 Read Cache 操作码 31h，如图 129 所示。

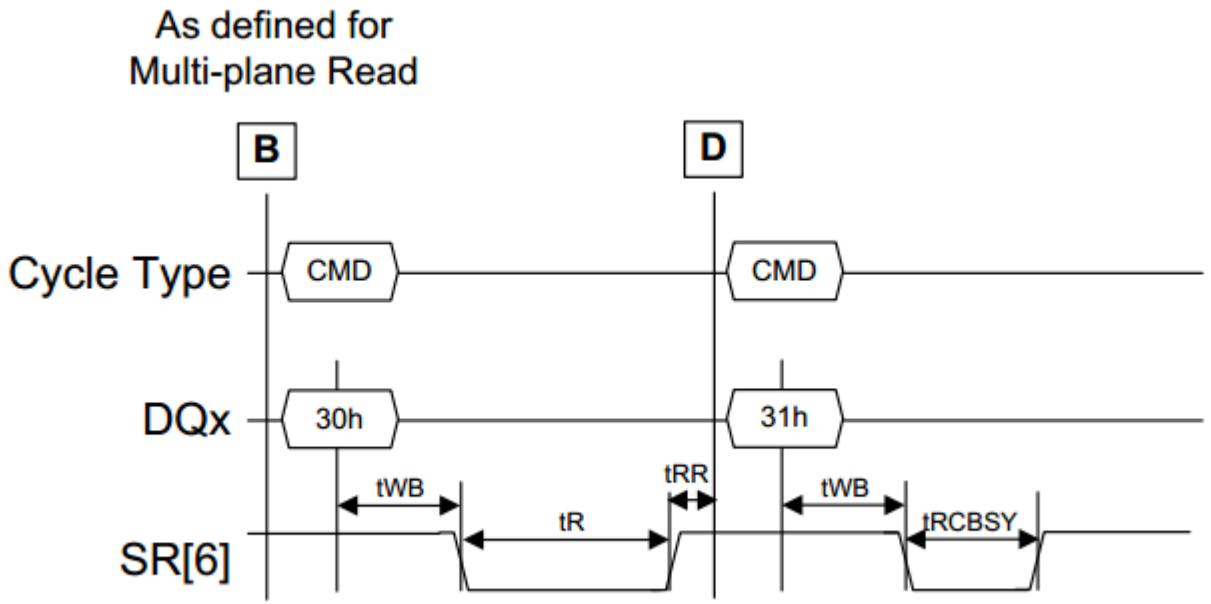


Figure 129 Multi-plane Read Cache Sequential command issue timing

对于多层 Read Cache Random 操作，初始的多层读命令之后紧跟着另一个读多层命令序列，该序列中最后一个操作码是 31h，如图 130 所示。

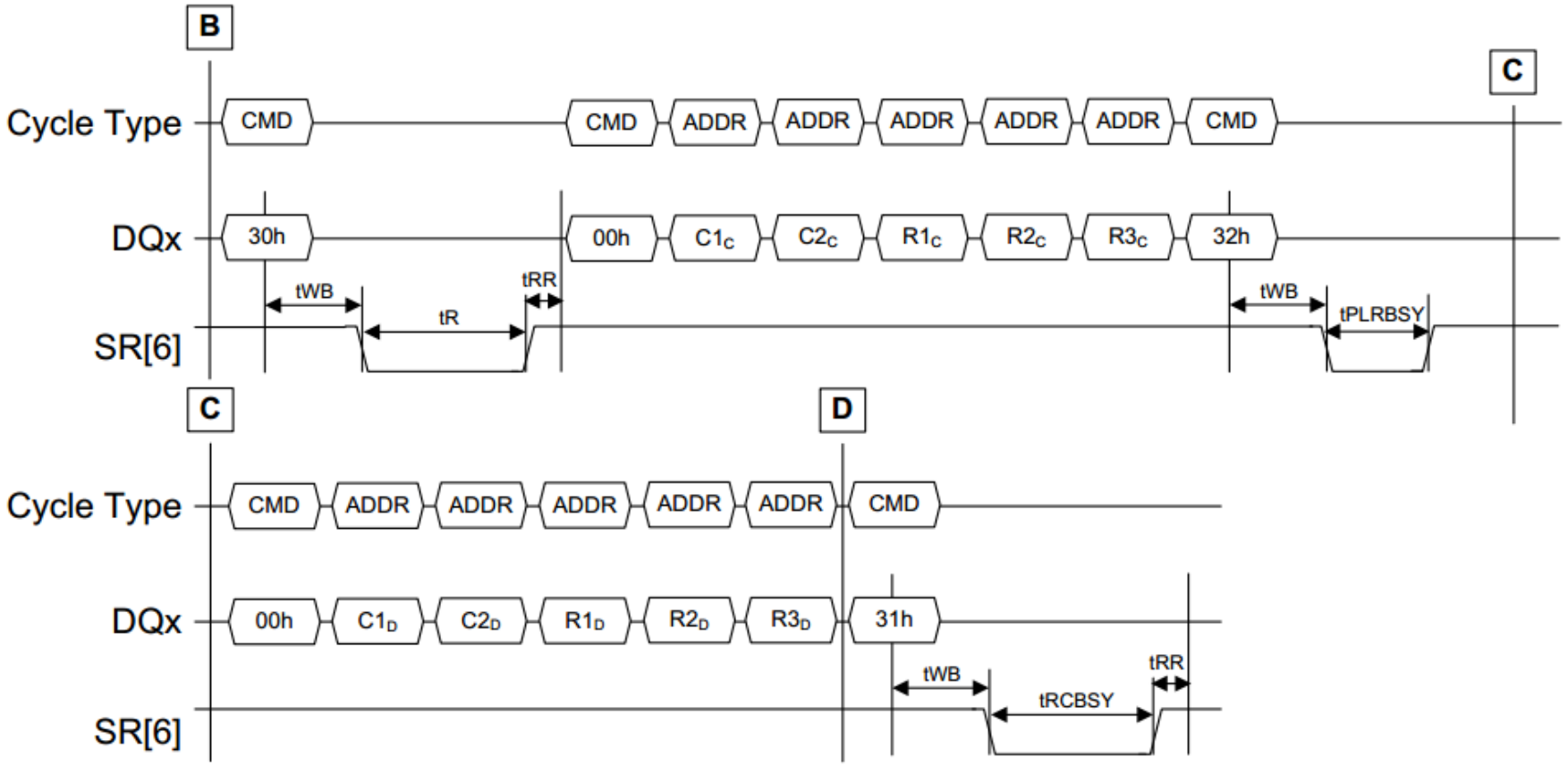


Figure 130 Multi-plane Read Cache Random command issue timing

$C1_C$-$C2_C$	Column address for page C. $C1_C$ is the least significant byte.
$R1_C$-$R3_C$	Row address for page C. $R1_C$ is the least significant byte.
$C1_D$-$C2_D$	Column address for page D. $C1_D$ is the least significant byte.
$R1_D$-$R3_D$	Row address for page D. $R1_D$ is the least significant byte.

在层地址位中 page C 和 page D 的行地址应该不同。

对于多层 Read Cache 操作，每个 Read Cache 操作跟着两个数据输出操作。每个数据输出操作序列如图 128 所示。在最后一组(比如，2)数据输出操作之前，host 应发送一个 Read Cache End 命令(3Fh)。

7. 行为流程(Behavioral Flows)

7.1 对象行为流程(Target behavioral flows)

对象状态机描述了允许的对象运行序列。如果没有条件为真，则对象保持在当前状态。

7.1.1 变量

该章节描述了对象状态机中使用的变量。

tbStatusOut	该变量在当一个数据读周期要返回状态值时为真。该变量上电后值为 FALSE。
tbChgCol	当允许使用 Change Read Column 来改变列时该变量为真。上电后值为 FALSE。
tbChgColEnh	当允许使用 Change Read Column Enhanced 来改变列时该变量为真。上电后值为 FALSE。
tCopyback	当对象发送一个回拷命令时该变量为真。上电后值为 FALSE。
tLunSelected	该变量包含当前被 host 选中的 LUN。上电后值为 0。
tLastCmd	该变量包含对象接收的上一个命令(除 70h/78h 外)的第一个周期。
tReturnState	该变量包含状态操作之后将要返回的状态。
tbStatus78hReq	当下一个状态操作为一个 78h 命令(并且不是一个 70h 命令)时为真。上电后值为 FALSE。

7.1.2 Idle 状态

T_PowerOn[1]	对象执行以下操作： 1. R/B_n 被清 0； 2. Each LUN shall draw less than 10mA of power per staggered power-up requirement		
	1.对象准备接收 FFh(Reset)命令 [2]	→	T_PowerOnReady
	注意： 1. 当 Vcc 达到 Vcc_min 时，上电后进入该状态； 2. 该条件应该在 Vcc 达到 Vcc_min 之后 1ms 内被执行；		

T_PowerOnReady	对象执行以下操作： 1. R/B_n 被设为 1； 2. Each LUN shall draw less than 10mA of power per staggered power-up requirement		
	1.接收到命令周期 FFh(Reset)	→	T_RST_PowerOn

T_Idle	tCopyback 设为 FALSE。tReturnState 设为 T_Idle		
	1.WP_n 信号转变	→	T_Idle_WP_Transition
	2.LUN 标示其转换后的 SR[6]值	→	T_Idle_RB_Transition
	3.接收命令周期	→	T_Cmd_Decode

T_Cmd_Decode[1]	解码接收到的命令。tbStatusOut 被设为 FALSE。如果 R/B_n 被设为 1 并且接收到的命令不是 70h(读状态)，则 tbStatus78hReq 被设为 FALSE。

1.命令 80h(页编程)或命令 60h(块擦除)解码，WP_n 为低	→	T_Idle
2.命令 FFh(Reset)解码	→	T_RST_Execute
3.命令 FCh(同步 Reset)解码	→	T_RST_Execute_Sync
4.命令 FAh(Reset LUN)解码	→	T_RST_Execute_LUN
5.命令 90h(读 ID)解码	→	T_RID_Execute
6.命令 ECh(读参数页)解码	→	T_RPP_Execute
7.命令 EDh(Read Unique ID)解码	→	T_RU_Execute
8.命令 80h(页编程)解码，WP_n 为高	→	T_PP_Execute
9.命令 60h(块擦除)解码，WP_n 为高	→	T_BE_Execute
10.命令 00h(读)解码	→	T_RD_Execute
11.命令 EFh(Set Feature)解码	→	T_SF_Execute
12.命令 EEh(Get Feature)解码	→	T_GF_Execute
13.命令 70h(读状态)解码	→	T_RS_Execute
14.命令 78h(Read Status Enhanced)解码	→	T_RSE_Execute
15.命令 E1h(Volume Select)解码	→	T_VS_Execute
16.命令 E2h(ODT 配置)解码	→	T_ODT_Execute
注意： 1.Host 在发送对象级命令(Reset,读 ID,读参数页,Read Unique ID,Set Feature,Get Feature)之前，应确保 R/B_n 为 1		

T_Idle_WP_Transition	标示所有 LUN 状态机的 WP_n 值

1.状态从 T_Idle_Rd 转入该状态	→	T_Idle_Rd
2.其他(else)	→	T_Idle

T_Idle_RB_Transition	R/B_n 被设为所有 LUN 状态寄存器 SR[6]值[1]的与(AND)

1.无条件的(Unconditional)	→	tReturnState
注意： 1. 当发送的 LUN 级命令正在处理时，在对象重新进入一个 idle 之前，R/B_n 可以转换到一个新的值。		

7.1.3 Idle Read 状态

T_Idle_Rd	等待读请求(数据或状态)或者其他操作。tReturnState 被设为 T_Idle_Rd

1.WP_n 信号转换	→	T_Idle_WP_Transition
2.LUN 标示其转换过的 SR[6]值	→	T_Idle_RB_Transition
3.接收到读请求，并且设置 tbStatusOut 为真	→	T_Idle_Rd_Status
4.接收到读请求，并且(tLastCmd 设为 90h 或 EEh)	→	T_Idle_Rd_XferByte
5.接收到读请求，并且(tLastCmd 设为 ECh 或 EDh)	→	T_Idle_Rd_LunByte
6.接收到读请求，并且 tbStatus78hReq 设为 FALSE[1]	→	T_Idle_Rd_LunData
7.接收到命令周期 05h(Change Read Column)，并且	→	T_CR_Execute[2]

tbChgCol 设为真		
8.接收到命令周期 06h(Change Read Column Enhanced)，并且 tbChgColEnh 设为真	→	T_CRE_Execute[2]
9.接收到命令周期 31h，并且 tbStatus78hReq 设为 FALSE	→	T_Idle_Rd_CacheCmd
10.接收到命令周期 3Fh，并且 tLastCmd 设为 31h，tbStatus78hReq 设为 FALSE	→	T_Idle_Rd_CacheCmd
11.接收到命令周期	→	T_Cmd_Decode
注意： 1. 当 tbStatus78hReq 被设为真时，从特定 LUN 中读取数据之前，host 应该发送一个 Read Status Enhanced(78h)命令，后面跟一个 00h 命令； 2. 如果该对象的其他 LUN 上有正在执行的读操作，则在传输数据之前应该发送一个 Change Read Column(Enhanced)命令。参见 3.1.3 多 LUN 操作限制的描述；		

T_Idle_Rd_CacheCmd	将 tLastCmd 设置为接收到的命令。将接收到的命令传递给 LUN tLunSelected		
	1.无条件的(Unconditional)	→	T_Idle_Rd

T_Idle_Rd_XferByte	返回数据的下一个 byte		
	1.无条件的(Unconditional)	→	T_Idle_Rd

T_Idle_Rd_LunByte	从 LUN tLunSelected 的页寄存器请求数据 byte		
	1.从 LUN tLunSelected 接收到 byte	→	T_Idle_Rd_XferHost

T_Idle_Rd_LunData	从 LUN tLunSelected 请求数据 byte(x8)或数据 word(x16)		
	1.从 LUN tLunSelected 接收到 byte 或 word	→	T_Idle_Rd_XferHost

T_Idle_Rd_XferHost	将从 LUN tLunSelected 接收到的数据 byte 或 word 传送给 host		
	1.tReturnState 设为 T_RD_StatusOff,tCopyback 设为真	→	T_RD_Copyback
	2.tReturnState 设为 T_RD_StatusOff	→	T_Idle_Rd
	3.其他(else)	→	tReturnState

T_Idle_Rd_Status	从 LUN tLunSelected 请求状态		
	1.从 LUN tLunSelected 接收到状态	→	T_Idle_Rd_StatusEnd

T_Idle_Rd_StatusEnd	将从 LUN tLunSelected 接收到的状态 byte 传送给 host		
	1.无条件的(Unconditional)	→	tReturnState

T_CR_Execute	等待一个列地址周期		
	1.接收到列地址周期	→	T_CR_Addr

T_CR_Addr	存储接收到的列地址周期		
	1.更多的列地址周期被请求	→	T_CR_Execute
	2.所有的列地址周期都被接收到	→	T_CR_WaitForCmd

T_CR_WaitForCmd	等待一个命令周期		
	1.接收到命令周期 E0h	→	T_CR_ReturnToData

T_CR_ReturnToData	请求 LUN tLunSelected 使用接收到的列地址在页寄存器中选择列		
	1.tReturnState 为 T_PP_MplWait	→	T_PP_WaitForDataOut
	2. tReturnState 为 T_RD_Status_Off	→	T_Idle_Rd
	3.其他(else)	→	tReturnState

T_CRE_Execute	等待一个列地址周期		
	1.接收到列地址周期	→	T_CRE_ColAddr

T_CRE_ColAddr	存储接收到的列地址周期		
	1.更多的列地址周期被请求	→	T_CRE_Execute
	2.所有的列地址周期都被接收到	→	T_CRE_RowAddrWait

T_CRE_RowAddrWait	等待一个行地址周期		
	1.接收到行地址周期	→	T_CRE_RowAddr

T_CRE_RowAddr	存储接收到的行地址周期		
	1.更多的行地址周期被请求	→	T_CRE_RowAddrWait
	2.所有的行地址周期都被接收到	→	T_CRE_WaitForCmd

T_CRE_WaitForCmd	等待一个命令周期		
	1.接收到命令周期 E0h	→	T_CRE_ReturnToData

T_CRE_ReturnToData	对象执行以下操作： 1. 将 tLunSelected 设为由接收到的行地址选中的 LUN； 2. 请求 LUN tLunSelect 使用接收到的列地址选择页寄存器中的列； 3. 将接收到的层地址标示在 tLunSelected 以在数据输出中使用； 4. 请求所有没有被选中的 idle LUN 关闭各自的输出缓存[1]；		
	1.tReturnState 为 T_PP_MplWait	→	T_PP_WaitForDataOut
	2. tReturnState 为 T_RD_Status_Off	→	T_Idle_Rd
	3.其他(else)	→	tReturnState
	注意： 1. 当接收到 Change Read Column Enhanced 命令时，如果没有被选中的 LUN 处于 idle 状态(SR[6]为 1)，则这些 LUN 只会关闭各自的输出缓存；当发送 Change Read Column Enhanced 命令后，如果 LUN 是 active 的(SR[6]为 0)，则在接下来的数据输出之前，应发送一个 Read Status Enhanced(78h)命令，以确保所有没有被选中的 LUN 都关闭了输出缓存。		

7.1.4 Reset 命令状态

T_RST_PowerOn	对象执行以下操作： 1. tLastCmd 设为 FFh 2. tbStatusOut 设为 FALSE 3. 对象发送一个 Reset 请求给每一个 LUN		
	1. 无条件的(Unconditional)	→	T_RST_PowerOn_Exec

T_RST_PowerOn_Exec	对象执行以下操作： 1. 执行对象级 reset 的操作 2. R/B_n 被设为 0		
	1. 无条件的(Unconditional)	→	T_RST_Perform

T_RST_Execute[1]	对象执行以下操作： 1. tLastCmd 设为 FFh 2. 对象选择 SDR 接口 3. 对象向每一个 LUN 发送一个 Reset 请求 4. tbChgCol 设为 FALSE 5. tbChgColEnh 设为 FALSE 6. 请求所有 LUN 将页寄存器置为无效		
	1. 无条件的(Unconditional)	→	T_RST_Perform
	注意： 1.当在任何其他状态接收到一个 Reset(FFh)命令时进入该状态(如果是上电后第一次 Reset 不适用该情况)		

T_RST_Execute_Sync[1]	对象执行以下操作： 1. tLastCmd 设为 FCh 2. tbStatusOut 被设为 FALSE 3. 对象向每一个 LUN 发送一个 Reset 请求 4. tbChgCol 设为 FALSE 5. tbChgColEnh 设为 FALSE 6. 请求所有 LUN 将页寄存器置为无效		
	1. 无条件的(Unconditional)	→	T_RST_Perform
	注意： 1. 当在任何其他状态接收到一个同步 Reset(FCh)命令时进入该状态		

T_RST_Execute_LUN[1]	对象执行以下操作： 1. tLastCmd 设为 FAh 2. tbStatusOut 被设为 FALSE 3. tbChgCol 设为 FALSE 4. tbChgColEnh 设为 FALSE 5. 等待一个地址周期		
	1. 无条件的(Unconditional)	→	T_RST_LUN_Addr

T_RST_LUN_AddrWait	等待一个地址周期		
	1.接收到地址周期	→	T_RST_LUN_Addr

T_RST_LUN_Addr	存储接收到的地址周期		
	1.更多的地址周期被接收到	→	T_RST_LUN_AddrWait
	2.所有的地址周期都被接收到	→	T_RST_LUN_Perform

T_RST_LUN_Perform	对象执行以下操作： 1. 对象向被寻址的 LUN 发送一个 Reset 请求 2. R/B_n 被清 0 3. 请求被寻址的 LUN 将其页地址变为无效		
	1.被寻址的 LUN 的 reset 操作完成，并且 tbStatusOut 被设为 FALSE	→	T_Idle
	2.被寻址的 LUN 的 reset 操作完成，并且 tbStatusOut 被设为真	→	T_Idle_Rd

T_RST_Perform	对象执行以下操作： 1. 执行对象级 reset 操作 2. R/B_n 被设为 0 3. tReturnState 设为 T_RST_Perform		
1.对象和 LUN 的 reset 操作完成		→	T_RST_End
2.接收到命令周期 70h(读状态)		→	T_RS_Execute
3.接收到读请求，并且 tbStatusOut 被设为真		→	T_Idle_Rd_Status

T_RST_End	对象执行以下操作： 1. R/B_n 被设为 1		
1.tbStatusOut 被设为 FALSE		→	T_Idle
2.tbStatusOut 被设为真		→	T_Idle_Rd

7.1.5 Read ID 命令状态

T_RID_Execute	对象执行以下操作： 1. tLastCmd 设为 90h 2. 等待一个地址周期 3. tbChgCol 设为 FALSE 4. tbChgColEnh 设为 FALSE 5. 请求所有 LUN 设置其页寄存器为无效		
1.接收到地址周期 00h		→	T_RID_Addr_00h
2.接收到地址周期 20h		→	T_RID_Addr_20h

T_RID_Addr_00h	等待一个读请求		
1.接收到读 byte 请求		→	T_RID_ManufacturerID
2.接收到命令周期		→	T_Cmd_Decode

T_RID_ManufacturerID	返回 JEDEC 制造商 ID		
1.接收到读 byte 请求		→	T_RID_DeviceID
2.接收到命令周期		→	T_Cmd_Decode

T_RID_DeviceID	返回 device ID[1]		
1.无条件(Unconditional)		→	T_Idle_Rd
注意： 1. 读取的 byte 超过 device ID 会返回制造商定义值(vendor specific values)			

T_RID_Addr_20h	等待一个读请求		
1.接收到读 byte 请求		→	T_RID_Signature
2.接收到命令周期		→	T_Cmd_Decode

T_RID_Signature	返回下一个 ONFI 签名[1]		
1.上一个 ONFI 签名被返回		→	T_Idle_Rd
2.其他(else)		→	T_RID_Addr_20h
注意： 1. 读取超过 4bytes 将返回未知值			

7.1.6 读参数页命令状态

T_RPP_Execute	对象执行以下操作： 1．tLastCmd 设为 ECh 2．tbChgCol 设为真 3．tbChgColEnh 设为 FALSE 4．等待一个地址周期 5．请求所有 LUN 将其页寄存器设为无效 6．对象选择 LUN 来执行读参数页操作，将 tLunSelected 设为该 LUN 的地址		
1.接收到地址周期 00h		→	T_RPP_ReadParams

T_RPP_ReadParams	对象执行以下操作： 1．请求 LUN tLunSelected 将 SR[6]清为 0 2．R/B_n 被清 0 3．请求 LUN tLunSelected 将页寄存器中的参数页数据设置为可读取 4．tReturnState 设为 T_RPP_ReadParams_Cont		
1.读页完成		→	T_RPP_Complete
2.接收到命令周期 70h(读状态)		→	T_RS_Execute
3.接收到读请求，并且 tbStatusOut 设为真		→	T_Idle_Rd_Status

T_RPP_ReadParams_Cont			
1.读页完成		→	T_RPP_Complete
2.接收到命令周期 70h(读状态)		→	T_RS_Execute
3.接收到读请求，并且 tbStatusOut 设为真		→	T_Idle_Rd_Status

T_RPP_Complete	请求 LUN tLunSelected 将 SR[6]设为 1。R/B_n 被设为 1		
1.无条件的(Unconditional)		→	T_Idle_Rd

7.1.7 Read Unique ID 命令状态

T_RU_Execute	对象执行以下操作： 1．tLastCmd 设为 EDh 2．tbChgCol 设为真 3．tbChgColEnh 设为 FALSE 4．请求所有 LUN 将页寄存器设为无效 5．等待一个地址周期 6．对象选择 LUN 来执行读 unique ID，并将 tLunSelected 设为该 LUN 的地址		
1.接收到地址周期 00h		→	T_RU_ReadUid

T_RU_ReadUid	对象执行以下操作： 1．请求 LUN tLunSelected 将 SR[6]清 0 2．R/B_n 被清 0 3．请求 LUN tLunSelected 将页寄存器中的 Unique ID 数据设为可读 4．tReturnState 设为 T_RU_ReadUid		
1.LUN tLunSelected 表明参数页中数据可读		→	T_RU_Complete
2.接收到命令周期 70h(读状态)		→	T_RS_Execute
3.接收到读请求，并且 tbStatusOut 设为真		→	T_Idle_Rd_Status

T_RU_Complete	请求 LUN tLunSelected 将 SR[6]设为 1。R/B_n 被设为 1。		
	1.无条件的(Unconditional)	→	T_Idle_Rd

7.1.8 页编程和页缓存编程(Page Cache Program)命令状态

T_PP_Execute	对象执行以下操作： 1．tLastCmd 设为 80h 2．如果 R/B_n 被清 0，则 tbStatus78hReq 被设为真 3．如果不支持 program page register clear enhancement 或被 disable，则请求所有 LUN 清除其页寄存器[1]		
	1.无条件的(Unconditional)	→	T_PP_AddrWait
	注意： 1．如果编程命令不是发往 Idle LUN，则该 LUN 可以不用清除其页寄存器		

T_PP_Copyback	如果 R/B_n 被清 0，则 tbStatus78hReq 被设为真		
	1.无条件的(Unconditional)	→	T_PP_AddrWait

T_PP_AddrWait	等待一个地址周期		
	1.接收到地址周期	→	T_PP_Addr

T_PP_Addr	存储接收到的地址周期		
	1.更多的地址周期被接收到	→	T_PP_AddrWait
	2.所有的地址周期都被接收到	→	T_PP_LUN_Execute

T_PP_LUN_Execute	对象执行以下操作： 1．tLunSelected 被设为由接收到的行地址标识的 LUN 2．如果 program page register clear enhancement 被使能，则请求 LUN tLunSelected 清除由层地址指定的页寄存器 3．对象以相关的地址向 LUN tLunSelected 发送编程命令		
	1.无条件的(Unconditional)	→	T_PP_LUN_DataWait

T_PP_LUN_DataWait	等待将要从 host 接收的数据 byte/word 或命令周期		
	1.从 host 接收到数据 byte/word	→	T_PP_LUN_DataPass
	2.接收到命令周期 15h，并且 tCopyback 设为 FALSE	→	T_PP_Cmd_Pass
	3.接收到命令周期 10h 或 11h	→	T_PP_Cmd_Pass
	4.接收到命令周期 85h	→	T_PP_ColChg

T_PP_LUN_DataPass	将从 host 接收到的数据 byte/word 传送到 LUN tLunSelected		
	1.无条件的(Unconditional)	→	T_PP_LUN_DataWait

T_PP_Cmd_Pass	将接收到的命令传送给 LUN tLunSelected		
	1.传送的命令为 11h	→	T_PP_MplWait
	2.传送的命令为 10h 或 15h	→	T_Idle

T_PP_MplWait	等待发送下一个编程命令。tReturnState 设为 T_PP_MplWait		
	1. 接收到命令周期 85h[1]	→	T_PP_AddrWait
	2.接收到命令周期 80h[2]，并且 tCopyback 设为 FALSE	→	T_PP_AddrWait
	3.接收到命令周期 05h	→	T_CR_Execute
	4.接收到命令周期 06h	→	T_CRE_Execute
	5.接收到命令周期 70h	→	T_RS_Execute
	6.接收到命令周期 78h	→	T_RSE_Execute
	7.接收到读请求，并且 tbStatusOut 设为真	→	T_Idle_Rd_Status
	注意： 1. 如果 85h 是回拷、Change Row Address 或 Small Data Move 操作的一部分，则 LUN 地址和层地址应该和前面的编程操作相同。如果 85h 是 Small Data Move 操作的一部分，则页地址也同样应该和前面的编程操作相同。 2. 发送的编程操作的地址周期应该与前面的编程操作具有相同的 LUN 地址和页地址。层地址应该和前一个编程操作中发送的层地址不同。		

T_PP_ColChg	等待列地址周期		
	1.接收到地址周期	→	T_PP_ColChg_Addr

T_PP_ColChg_Addr	存储接收到的地址周期		
	1.更多的地址被接收到	→	T_PP_ColChg
	2.所有的地址周期都被接收到	→	T_PP_ColChg_LUN

T_PP_ColChg_LUN	请求 LUN tLunSelected 更改列地址为接收到的列地址		
	1.无条件的(Unconditional)	→	T_PP_ColChg_Wait

T_PP_ColChg_Wait	等待一个从 host 接收的地址周期、数据 byte/word 或命令周期		
	1. 接收到地址周期	→	T_PP_RowChg_Addr
	2.从 host 接收到数据 byte/word	→	T_PP_LUN_DataPass
	3.接收到命令周期 15h，并且 tCopyback 设为 FALSE	→	T_PP_Cmd_Pass
	4.接收到命令周期 10h 或 11h	→	T_PP_Cmd_Pass
	5.接收到命令周期 85h	→	T_PP_ColChg

T_PP_RowChg	等待行地址周期		
	1.接收到地址周期	→	T_PP_RowChg_Addr

T_PP_RowChg_Addr	存储接收到的地址周期		
	1.更多的地址被接收到	→	T_PP_RowChg
	2.所有的地址周期都被接收到	→	T_PP_RowChg_LUN

T_PP_RowChg_LUN	请求 LUN tLunSelected 更改行地址为接收到的行地址[1]		
	1.无条件的(Unconditional)	→	T_PP_LUN_DataWait
	注意： 1. LUN 地址和层地址应该和之前执行的编程操作相同。		

T_PP_WaitForDataOut	等待读请求(数据或状态)或其他操作。tReturnState 设为 T_PP_WaitForDataOut		
	1. 接收到读请求，并且 tbStatusOut 设为真	→	T_Idle_Rd_Status
	2.接收到读请求，并且 tbStatus78hReq 设为 FALSE[1]	→	T_Idle_Rd_LunData
	3.接收到命令周期 70h	→	T_RS_Execute
	4.接收到命令周期 78h	→	T_RSE_Execute
	5.接收到命令周期 00h	→	T_RD_Execute
	6.接收到命令周期	→	T_PP_MplWait
	注意： 1. 当 tbStatus78hReq 被设为真时，从一个特定的 LUN 读取数据之前,host 应发送一个 Read Status Enhanced(78h)命令，后面紧跟着一个 00h 命令。		

7.1.9 块擦除命令状态

T_BE_Execute	对象执行以下操作： 1. tLastCmd 设为 60h 2. 如果 R/B_n 被清 0，则 tbStatus78hReq 被设为真 3. 等待一个行地址周期		
	1.接收到地址周期	→	T_BE_Addr

T_BE_Addr	存储接收到的行地址周期		
	1.更多的地址周期被接收到	→	T_BE_Execute
	2.所有的地址周期都被接收到	→	T_BE_LUN_Execute

T_BE_LUN_Execute	tLunSelected 被设为由接收到的行地址指定的 LUN。对象以相关的行地址发送擦除命令到 LUN tLunSelected		
	1.无条件的(Unconditional)	→	T_BE_LUN_Confirm

T_BE_LUN_Confirm	等待 D0h 或 D1h 命令周期		
	1.接收到 D0h 或 D1h 命令周期	→	T_BE_Cmd_Pass

T_BE_Cmd_Pass	将接收到的命令传送给 LUN tLunSelected		
	1.传送的命令为 D1h	→	T_BE_MplWait
	2.传送的命令为 D0h	→	T_Idle

T_BE_MplWait	等待下一个要发送的擦除命令。tReturnState 设为 T_BE_MplWait		
	1.接收到命令周期 60h	→	T_BE_Execute
	2.接收到命令周期 70h	→	T_RS_Execute
	3.接收到命令周期 78h	→	T_RSE_Execute
	4.接收到读请求，并且 tbStatusOut 设为真	→	T_Idle_Rd_Status

7.1.10 读命令状态

T_RD_Execute			
	1.tbStatusOut 设为真	→	T_RD_StatusOff
	2.其他(Else)	→	T_RD_AddrWait

T_RD_StatusOff	tbStatusOut 设为 FALSE。tReturnState 设为 T_RD_StatusOff		
	1.接收到地址周期	→	T_RD_Addr
	2.接收到读请求，并且 tLastCmd 设为 80h	→	T_PP_WaitForDataOut
	3.接收到读请求，并且 tLastCmd 设为 EEh	→	T_Idle_Rd_XferHost
	4.接收到读请求	→	T_Idle_Rd_LunData
	5.接收到命令周期 05h	→	T_CR_Execute
	6.接收到命令周期 06h	→	T_CRE_Execute

T_RD_AddrWait	tLastCmd 设为 00h。tbChgCol 设为真。tbChgColEnh 设为真。如果 R/B_n 被清 0，则 tbStatus78hReq 被设为真。等待一个地址周期。		
	1.接收到地址周期	→	T_RD_Addr

T_RD_Addr	存储接收到的地址周期		
	3. 更多的地址周期被接收到	→	T_RD_AddrWait
	4. 所有的地址周期都被接收到	→	T_RD_LUN_Execute

T_RD_LUN_Execute	对象执行以下操作： 1. 将 tLunSelected 设为由接收到的行地址指定的 LUN 2. 发送 Read Page 命令，地址为 LUN tLunSelected 的 3. 请求所有未被选中的 idle LUN 关闭各自的输出缓存 [1]		
	1.无条件(Unconditional)	→	T_RD_LUN_Confirm
	注意： 1. 没有被选中的 LUN 如果处于 idle 状态，则只关闭输出缓存。如果其他 LUN 是 active 的，则 host 应发送一个 Read Status Enhanced(78h)命令，以确保在发送读(00h)命令之前所有未被选中的 LUN 都关闭了输出缓存。		

T_RD_LUN_Confirm	等待接收 30h,31h,32h 或 35h		
	1.接收到命令周期 30h,31h,32h 或 35h	→	T_RD_Cmd_Pass

T_RD_Cmd_Pass	将接收到的命令传送给 LUN tLunSelected		
	1.传送的命令是 35h	→	T_RD_Copyback
	2.传送的命令是 30h,31h 或 32h	→	T_Idle_Rd

T_RD_Copyback	tCopyback 设为真。tReturnState 设为 T_RD_Copyback		
	1.接收到的命令周期为 00h	→	T_RD_Execute
	2.接收到的命令周期为 05h	→	T_CR_Execute
	3.接收到的命令周期为 06h	→	T_CRE_Execute
	4.接收到的命令周期为 85h	→	T_PP_Copyback
	5.接收到的命令周期为 70h	→	T_RS_Execute
	6.接收到的命令周期为 78h	→	T_RSE_Execute
	7.LUN 标示其 SR[6]值转变	→	T_Idle_RB_Transition
	8.接收到读请求，并且 tbStatusOut 设为真	→	T_Idle_Rd_Status
	9.接收到读请求	→	T_Idle_Rd_LunData

7.1.11 Set Feature 命令状态

T_SF_Execute	对象执行以下操作： 1. tLastCmd 设为 EFh 2. 请求所有 LUN 将页寄存器设为无效 3. 等待一个地址周期		
	1.接收到地址周期	→	T_SF_Addr

T_SF_Addr	存储接收到的 feature 地址(feature address)		
	1.无条件(Unconditional)	→	T_SF_WaitForParams

T_SF_WaitForParams	等待接收数据 byte		
	1.数据 byte 被写入对象	→	T_SF_StoreParam

T_SF_StoreParams	存储接收到的参数		
	1.更多的参数被请求	→	T_SF_WaitForParams
	2.所有参数都被接收	→	T_SF_Complete

T_SF_Complete	对象执行以下操作： 1. 请求 LUN tLunSelected 将 SR[6]清 0 2. R/B_n 被清 0 3. Set Feature 命令结束 4. tReturnState 设为 T_SF_Complete		
	1.Set Feature 命令完成	→	T_SF_UpdateStatus
	2.接收到命令周期 70h(读状态)	→	T_RS_Execute
	3.接收到读请求，并且 tbStatusOut 设为真	→	T_Idle_Rd_Status

T_SF_UpdateStatus	对象执行以下操作： 1. 请求 LUN tLunSelected 设置 SR[6]为 1 2. R/B_n 被设为 1		
	1.tbStatusOut 被设为 FALSE	→	T_Idle
	2.tbStatusOut 被设为真	→	T_Idle_Rd

7.1.12 Get Feature 命令状态

T_GF_Execute	对象执行以下操作： 1. tLastCmd 设为 EEh 2. 请求所有 LUN 将页寄存器设为无效 3. tbChgCol 设为 FALSE 4. tbChgColEnh 设为 FALSE 5. 等待一个地址周期		
	1.接收到地址周期	→	T_GF_Addr

T_GF_Addr	存储接收到的 feature 地址(feature address)		
	1.无条件(Unconditional)	→	T_GF_RetrieveParams

T_GF_RetrieveParams	对象执行以下操作： 1. 请求 LUN tLunSelected 将 SR[6]清 0 2. R/B_n 被清 0 3. 检索参数 4. tReturnState 设为 T_GF_RetrieveParams		
	1.参数准备好被传输到 host	→	T_GF_Ready
	2.接收到命令周期 70h(读状态)	→	T_RS_Execute
	3.接收到读请求，并且 tbStatusOut 设为真	→	T_Idle_Rd_Status

T_GF_Ready	请求 LUN tLunSelected 设置 SR[6]为 1。R/B_n 设为 1		
	1.无条件(Unconditional)	→	T_Idle_Rd

7.1.13 读状态命令状态

T_RS_Execute			
	1. tbStatus78hReq 被设为 FALSE[1]	→	T_RS_Perform
	注意： 1. 当 tbStatus78hReq 被设为真时，不能发送读状态(70h)命令(非法的)		

T_RS_Perform	对象执行以下操作： 1. tbStatusOut 被设为真 2. 将接收到的 70h 命令标示在 LUN tLunSelected 中		
	1. tReturnState 设为 T_Idle	→	T_Idle_Rd
	2.其他	→	tReturnState

7.1.14 Read Status Enhanced 命令状态

T_RSE_Execute[1]	tbStatus78hReq 被设为 FALSE。tbStatusOut 被设为真。等待一个行地址周期。		
	1. 接收到行地址周期	→	T_RSE_Addr
	注意： 1. host 不能在紧跟着一个对象级命令(Reset,Read ID,读参数页，Read Unique ID，Set Feature,Get Feature)之后发送 Read Status Enhanced 命令。从被 Read Status Enhanced 命令选中的 LUN 中读出的状态值可能和在对象级命令期间被选中的 LUN 不相关。		

T_RSE_Addr	存储接收到的行地址		
	1. 更多的行地址周期被请求	→	T_RSE_Execute
	2.所有的行地址周期都被接收到	→	T_RSE_Select

T_RSE_Select	对象执行以下操作： 1. 将 tLunSelected 设为被接收到的行地址选中的 LUN 2. 将接收到的 78h 命令和行地址标示到所有 LUN		
	1. tReturnState 设为 T_Idle	→	T_Idle_Rd
	2.其他(else)	→	tReturnState

7.1.15 Volume Select 命令状态

T_VS_Execute	对象执行以下操作： 1. tLastCmd 设为 E1h 2. 等待一个地址周期		
	1.接收到地址周期	→	T_VS_Complete

T_VS_Complete	将接收到的 Volume 地址标示到所有 LUN		
	1.无条件的(Unconditional)	→	T_Idle

7.1.16 ODT 配置命令状态

T_ODTC_Execute	对象执行以下操作： 1. tLastCmd 设为 E2h 2. 等待一个地址周期		
	1.接收到地址周期	→	T_ODTC_Addr

T_ODTC_Addr	存储接收到的 LUN；将接下来的 matrix 和 Rtt 设置应用到被标示的 LUN		
	1.无条件的(Unconditional)	→	T_ODTC_WaitForParam

T_ODTC_WaitForParam	等待接收数据 byte		
	1.数据 byte 被写到对象	→	T_ODTC_StoreParam

T_ODTC_StoreParam	存储接收到的 matrix(byte 0/1)或 Rtt(byte 2/3)参数		
	1.更多的参数被请求	→	T_ODTC_WaitForParam
	2.所有的参数都被接收到	→	T_ODTC_Complete

T_ODTC_Complete	标示 LUN，确定 ODT 配置参数(M0,M1,Rtt1,Rtt2)		
	1.无条件的(Unconditional)	→	T_Idle

7.2 LUN 行为流程(LUN behavioral flows)

LUN 状态机描述了当执行 LUN 操作时允许的序列。如果没有条件为真，则 LUN 保持在当前状态。

7.2.1 变量

该章节描述了 LUN 状态机使用的变量。

lunStatus 该变量包含当前 LUN 状态寄存器的值。上电后值为 00h

lunFail[] 该数组包含每一个交错地址(interleave address)的 FAIL 和 FAILC 位。例如，lunFail[3][1]包含层地址 3 的 FAILC 位。上电后数组中每个变量的值为 00b

lunLastConfirm 该变量包含上一个确定的命令周期(30h,31h,32h,35h,10h,15h,11h,D0h,D1h)。上电后值为 FFh

lunOutputMpl 该变量包含请求用于数据输出的层地址。上电后值为 0h

lunReturnState 该变量包含状态操作之后要返回的状态。上电后值为 L_Idle

lunStatusCmd 该变量包含上一个接收到的状态命令。上电后值为 70h

lunStatusMpl 该变量包含前一个 78h 命令中标示的层地址。上电后值为 0h

lunblnterleave 该变量当 LUN 正在执行一个多层操作时被设为 1。上电后值为 FALSE

lunbMplNextCmd　该变量当 LUN 准备好接收下一个多层命令时被设为真
lunEraseAddr[]　该变量包含被暂停的擦除操作的块地址

7.2.2 Idle 命令状态

L_Idle[1]	lunReturnState 被设为 L_Idle		
	1.接收到对象请求	→	L_Idle_TargetRequest
	注意： 1. 当 Vcc 达到 Vcc_min 时，上电后进入该状态		

L_Idle_TargetRequest	如果对象标示了一个地址，则该地址被 LUN 存储		
	1.对象请求 LUN 执行一个 Reset	→	L_RST_Execute
	2.对象标示 WP_n 值	→	L_WP_Update
	3.对象请求更新 SR 寄存器	→	L_SR_Update
	4.对象请求接收到的状态或状态命令	→	L_Status_Execute
	5.对象标示用于数据输出的层地址	→	L_Idle_Mpl_DataOutAddr
	6.对象标示输出缓存应被关闭	→	L_Idle
	7.对象请求清除夜寄存器	→	L_Idle_ClearPageReg
	8.对象请求将页寄存器设为无效	→	L_Idle_InvalidPageReg
	9.对象标示该 LUN 的编程请求	→	L_PP_Execute
	10.对象标示该 LUN 的擦除请求	→	L_BE_Execute
	11.对象标示该 LUN 的擦除恢复请求	→	L_ER_Execute
	12.对象标示该 LUN 的读页(Read Page)请求	→	L_RD_Addr
	13.对象标示读参数页请求	→	L_Idle_RdPp
	14.对象标示 Read Unique ID 请求	→	L_Idle_RdUid
	15.对象标示接收到的 Volume 地址	→	L_Idle_VolAddr
	16.对象标示接收到的 ODT 配置设置	→	L_Idle_ODTConfig

L_WP_Update	lunStatus[7]设为对象标示的 WP_n 值		
	1.无条件的(Unconditional)	→	lunReturnState

L_SR_Update	将 lunStatus 更新为对象标示的值		
	1.无条件的(Unconditional)	→	lunReturnState

L_Idle_Mpl_DataOutAddr	将 lunOutputMpl 设为对象标示的层地址		
	1.无条件的(Unconditional)	→	lunReturnState

L_Idle_ClearPageReg	将页寄存器值设为全 1		
	1.无条件的(Unconditional)	→	lunReturnState

L_Idle_InvalidPageReg	将页寄存器设为无效		
	1.无条件的(Unconditional)	→	lunReturnState

L_Idle_RdPp	LUN 执行以下操作： 1. LUN 将参数页的数据读入页寄存器 2. lunReturnState 设为 L_Idle_RdPp_Cont		
	1.参数页数据传送到页寄存器	→	L_Idle_RdPp_End
	2.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_Idle_RdPp_Cont			
	1.参数页数据传送到页寄存器	→	L_Idle_RdPp_End
	2.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_Idle_RdPp_End	LUN 向对象表明参数页数据在页寄存器中		
	1. 无条件(Unconditional)	→	L_Idle_Rd

L_Idle_RdUid	LUN 执行以下操作： 1. LUN 将 Unique ID 数据读入到页寄存器 2. lunReturnState 设为 L_Idle_RdUid		
	1.Unique ID 数据传输到页寄存器	→	L_Idle_RdUid_End
	2.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_Idle_RdUid_End	LUN 向对象表明参数页数据在页寄存器中		
	1. 无条件(Unconditional)	→	L_Idle_Rd

7.2.3 Idle Read 状态

L_Idle_Rd	lunReturnState 被设为 L_Idle_Rd		
	1.读操作完成	→	L_Idle_Rd_Finish
	2.对象请求被选中的列地址	→	L_Idle_Rd_ColSelect
	3.从对象接收到读请求	→	L_Idle_Rd_Xfer
	4.接收到命令周期 31h(Read Cache Sequential)	→	L_RD_Cache_Next
	5.接收到命令周期 3Fh(Read Cache End),并且 lunLastConfirm 为 31h	→	L_RD_Cache_Xfer_End
	6.接收到对象请求	→	L_Idle_TargetRequest

L_Idle_Rd_Finish	将 lunStatus[5]设为 1		
	1. 无条件(Unconditional)	→	L_Idle_Rd

L_Idle_Rd_Xfer	将对象请求的下一个数据 byte(x8)或 word(x16)从页寄存器返回给对象。将列地址递增。		
	1. lunReturnState 设为 L_PP_Mpl_Wait	→	L_PP_Mpl_Wait
	2.无条件(Unconditional)	→	L_Idle_Rd

L_Idle_Rd_ColSelect	选择页寄存器中的列，该列由从对象接收到的列地址确定。		
	1. lunReturnState 设为 L_PP_Mpl_Wait	→	L_PP_Mpl_Wait
	2.无条件(Unconditional)	→	L_Idle_Rd

L_Idle_VolAddr			
	1.Volume 地址与该 LUN 所在的 NAND 对象匹配(LUN 保持为被选中状态)	→	L_Idle
	2.LUN 是指定 Volume 地址的一个 terminator	→	L_Idle_VolSniff
	3.Volume 地址与该 LUN 所在的 NAND 对象不匹配	→	L_Idle_VolDeselect
	4.无条件的(Unconditional)	→	L_Idle
	注意： 1. 更多细节请参考 ODT 行为流程图 48，图 49 以及图 50。		

L_Idle_VolSniff	LUN 进入 Sniff 状态		
	1. 无条件(Unconditional)	→	L_Idle

L_Idle_VolDeselect	LUN 被解除选择		
	1. 无条件(Unconditional)	→	L_Idle

L_Idle_ODTConfig	LUN 存储/应用指定的 ODT matrix 和 Rtt 设置		
	1. 无条件(Unconditional)	→	L_Idle

7.2.4 Status 状态

L_Status_Execute			
	1. 对象请求状态值	→	L_Status_Value
	2. 对象表明 78h 已经被接收到	→	L_Status_Enhanced
	3. 对象表明 70h 已经被接收到	→	L_Status_Legacy

L_Status_Value			
	1. lunblnterleave 设为真，并且 lunStatusCmd 设为 70h	→	L_Status_Mpl_Comp
	2. lunblnterleave 设为真，并且 lunStatusCmd 设为 78h	→	L_Status_Mpl_Addr
	3. lunblnterleave 设为 FALSE	→	L_Status_Lun

L_Status_Enhanced			
	1. 行地址中的 LUN 表明与该 LUN 匹配	→	L_Status_Record_78h
	2. 其他(else)	→	L_Status_Output_Off

L_Status_Record_78h	lunStatusCmd 被设为 78h，lunStatusMpl 被设为由对象表明的层地址。LUN 关闭其输出缓存		
	1. lunReturnState 设为 L_Idle(lunLastConfirm 设为 30h,31h,32h 或 35h)	→	L_Idle_Rd
	2. 其他(else)	→	lunReturnState

L_Status_Output_Off	LUN 关闭其输出缓存		
	1. lunReturnState 设为 L_Idle_Rd	→	L_Idle
	2. 其他(else)	→	lunReturnState

L_Status_Legacy	lunStatusCmd 被设为 70h		
	1. 无条件的(Unconditional)	→	lunReturnState

L_Status_Mpl_Comp	LUN 要返回的状态值构成如下： • status[7:2] = lunStatus[7:2] • status[1] = for all x,OR of lunFail[x][1] • status[0] = for all x,OR of lunFail[x][0] 将状态返回给对象		
	1. 无条件的(Unconditional)	→	lunReturnState

L_Status_Mpl_Addr	LUN 要返回的状态值构成如下： • status[7:2] = lunStatus[7:2] • status[1:0] = lunFail[lunStatusMpl][1:0] 将状态返回给对象		
1. 无条件的(Unconditional)		→	lunReturnState

L_Status_Lun	向对象返回 lunStatus		
1. 无条件的(Unconditional)		→	lunReturnState

7.2.5 Reset 状态

L_RST_Execute[1]	LUN 执行以下操作： 1. lunStatus[6]被清 0 2. lunStatus[6]值被标示到对象 3. 执行 LUN reset 4. lunblnterleave 被设为 FALSE 5. lunReturnState 被设为 L_RST_Execute		
1. LUN reset 完成		→	L_RST_Complete
2. 对象请求接收到的状态或状态命令		→	L_Status_Execute
注意： 1. 当在任何其他状态接收到来自对象状态机的标识来执行一个 Reset 时进入该状态			

L_RST_Complete	LUN 执行以下操作： 1. lunStatus[1:0]被清除为 00b 2. 对于所有层地址 x，将 lunFail[x][1:0]清除为 00b 3. lunStatus[6]被设为 1 4. lunStatus[6]的值被标示到对象 5. 向对象状态机表明该 LUN 的 Reset 已完成		
1. 无条件的(Unconditional)		→	L_Idle

7.2.6 块擦除命令状态

L_BE_Execute	lunblnterleave 设为 FALSE		
1. 无条件的(Unconditional)		→	L_BE_WaitForCmd

L_BE_WaitForCmd	等待一个命令周期		
1. 接收到命令周期 D0h		→	L_BE_Erase
2.接收到命令周期 D1h		→	L_BE_Mpl

L_BE_Erase	LUN 执行以下操作： 1. lunStatus[6]被清 0 2. 如果 lunblnterleave 为真，则 lunStatus[5]被清为 0。 3. lunStatus[6]的值被标示到对象 4. lunLastConfirm 设为 D0h 5. 如果 lunblnterleave 被设为真，并且支持并发的 interleaving，则擦除被请求的块以及任何之前被请求的块		
1. 无条件的(Unconditional)		→	L_BE_Erase_Wait

L_BE_Erase_Wait	lunReturnState 设为 L_BE_Erase_Wait		
	1.被请求的块擦除完成，并且 lunblnterleave 设为真	→	L_BE_Mpl_Sts
	2.被请求的块擦除完成	→	L_BE_Sts
	3.对象请求清除页寄存器	→	L_Idle_ClearPageReg
	4.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_BE_Mpl	LUN 以指定的顺序执行以下操作： 1. lunblnterleave 设为真 2. lunLastConfirm 设为 D1h 3. lunStatus[6:5]被清为 00b。lunStatus[6]的值被标示到对象 4. 如果指定块支持重叠(overlapped)，则 LUN 开始擦除该块 5. lunbMplNextCmd 被设为 FALSE 6. LUN 准备接收下一个块擦除操作		
	1.无条件的(Unconditional)	→	L_BE_Mpl_Wait

L_BE_Mpl_Wait	lunReturnState 设为 L_BE_Mpl_Wait		
	1.一个重叠的多层擦除操作完成	→	L_BE_Mpl_Overlap
	2.准备接收下一个擦除命令，并且 lunbMplNextCmd 被设为 FALSE	→	L_BE_Mpl_NextCmd
	3.对象表明该 LUN 擦除请求，并且 lunbMplNextCmd 被设为真	→	L_BE_WaitForCmd
	4.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_BE_Mpl_NextCmd	LUN 以指定的顺序执行以下操作： 1. lunbMplNextCmd 被设为真 2. 如果没有正在执行的阵列操作，则 lunStatus[5]被设为 1 3. lunStatus[6]被设为 1。lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_BE_Mpl_Wait

L_BE_Mpl_Overlap	LUN 为已完成的重叠多层操作以指定的顺序执行以下操作： 1. mplComplete 设为已完成操作的层地址 2. lunFail[mplComplete][0]被设为操作的编程状态 如果所有的阵列操作都完成了，则 lunStatus[5]被设为 1		
	1.无条件的(Unconditional)	→	lunReturnState

L_BE_Sts	LUN 以指定的顺序执行以下操作： 1. lunStatus[0]被设为擦除状态 2. lunStatus[6]被设为 1。lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_Idle

L_BE_Mpl_Sts	LUN 为每个已完成的多层操作以指定的顺序执行以下操作： 1. mplComplete 设为已完成操作的 interleave 地址 2. lunFail[mplComplete][0]被设为擦除状态值 lunStatus[6:5]被设为 11b，并且 lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_Idle

7.2.7 读命令状态

如果不支持缓存(Caching)，则所有对于状态位 5 的操作都被忽略。

L_RD_Addr	LUN 以指定的顺序执行以下操作： 1. 记录从对象接收到的地址 2. 如果支持多层地址，则选择根据层地址选择正确的页寄存器 3. 根据接收到的列地址在页寄存器中选择列		
	1.无条件的(Unconditional)	→	L_RD_WaitForCmd

L_RD_WaitForCmd	lunbInterleave 设为 FALSE。等待一个命令周期		
	1.接收到命令周期 30h 或 35h	→	L_RD_ArrayRead
	2. 接收到命令周期 31h，并且 lunLastConfirm 等于 30h 或 31h	→	L_RD_Cache_Xfer
	3.接收到命令周期 32h	→	L_RD_Mpl_Xfer

L_RD_ArrayRead	LUN 执行以下操作： 1. lunStatus[6:5]被清 0 2. lunStatus[6]的值被标示到对象 3. lunLastConfirm 设为上一个命令周期(30h 或 35h) 4. 从阵列读取被请求的页。如果为并发多层操作，则从阵列读取所有被请求的页。 5. lunReturnState 设为 L_RD_ArrayRead_Cont		
	1.读取请求的页完成	→	L_RD_Complete
	2. 对象请求接收到的状态或状态命令	→	L_Status_Execute

L_RD_ArrayRead_Cont			
	1.读取请求的页完成	→	L_RD_Complete
	2. 对象请求接收到的状态或状态命令	→	L_Status_Execute

L_RD_Complete	lunStatus[6:5]被设为 11b。lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_Idle_Rd

L_RD_Cache_Next	选择上个页读行地址的递增作为下一个行地址(如，上一个行地址+1)		
	1.无条件的(Unconditional)	→	L_RD_Cache_Xfer

L_RD_Cache_Xfer	LUN 执行以下操作： 1. lunStatus[6:5]被清为 00b。lunStatus[6]的值被标示到对象 2. lunLastConfirm 设为 31h 3. 为选中的地址开始读操作 4. lunReturnState 设为 L_RD_Cache_Xfer		
	1.对于前一个读操作，页寄存器中的数据可读	→	L_RD_Cache_Sts
	2.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_RD_Cache_Xfer_End	LUN 执行以下操作： 1. lunStatus[6]被清为 0。 2. lunStatus[6]的值被标示到对象 3. lunLastConfirm 设为 3Fh 4. lunReturnState 设为 L_RD_Cache_Xfer_End		
	1.对于前一个读操作，页寄存器中的数据可读	→	L_RD_Cache_Sts_End
	2.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_RD_Cache_Sts	lunStatus[6]被设为 11b。lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_Idle_Rd

L_RD_Cache_Sts_End	lunStatus[6:5]被设为 1。lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_Idle_Rd

L_RD_Mpl_Xfer	LUN 执行以下操作： 1. lunStatus[6:5]被清为 00b。 2. lunStatus[6]的值被标示到对象 3. lunLastConfirm 设为 32h 4. lunbMplNextCmd 被设为 FALSE 5. 如果支持重叠 interleaving，则开始读取指定的页 6. 准备接收下一个要读取的页 7. lunReturnState 设为 L_RD_Mpl_Xfer		
	1.对象准备接收下一个要读取的页	→	L_RD_Mpl_Wait
	2.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_RD_Mpl_Wait	lunStatus[6]被设为 1。lunStatus[6]的值被标示到对象。lunReturnState 设为 L_RD_Mpl_Wait		
	1.一个重叠多层读操作完成	→	L_RD_Mpl_Overlap
	2.对象标示该 LUN 的读页请求	→	L_RD_Addr
	3.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_RD_Mpl_Overlap	对于已完成的重叠多层操作，LUN 以指定顺序执行以下操作： 1. mplComplete 设为已完成的操作的层地址 如果所有的阵列操作都已完成，则 lunStatus[5]被设为 1		
	1.无条件的(Unconditional)	→	lunReturnState

7.2.8 页编程和页缓存编程(Page Cache Program)命令状态

如果不支持 caching 或重叠(overlapped) interleaving,则所有状态位 5 的操作被忽略。如果不支持 caching，则所有状态位 1 的操作被忽略。

L_PP_Execute	lunbInterleave 设为 FALSE		
	1.无条件的(Unconditional)	→	L_PP_Addr

L_PP_Addr	LUN 以指定的顺序执行以下操作： 1. 记录从对象接收到的地址 2. 如果支持多层寻址，则根据层地址选择正确的页寄存器 3. 根据接收到的列地址选择页寄存器中的列		
	1.无条件的(Unconditional)	→	L_PP_WaitForData

L_PP_WaitForData	等待接收数据。lunReturnState 被设为 L_PP_WaitForData		
	1.对象将数据 byte 或 word 传递给 LUN	→	L_PP_AcceptData
	2.接收到命令周期 10h(执行编程)	→	L_PP_Prog
	3.接收到命令周期 15h(cache program)	→	L_PP_Cache
	4.接收到命令周期 11h(interleave)	→	L_PP_Mpl
	5.对象请求被选中的列地址	→	L_PP_ColSelect
	6.对象请求被选中的行地址	→	L_PP_RowSelect

L_PP_AcceptData	将数据 byte(x8)或 word(x16)写入到页寄存器中选定的列地址。递增列地址。		
	1.无条件的(Unconditional)	→	L_PP_WaitForData

L_PP_Prog	LUN 以指定的顺序执行以下操作： 1. lunStatus[6:5]被清为 00h。lunStatus[6]的值被标示到对象 2. lunLastConfirm 设为 10h 3. 如果仅指定了一个要编程的页，则将 lunblnterleave 清为 FALSE 4. 如果 lunblnterleave 为真并且支持并发 interleaving，则 LUN 开始指定页以及任何之前指定页的编程		
	1.无条件的(Unconditional)	→	L_PP_Prog_Wait

L_PP_Prog_Wait	lunReturnState 设为 L_PP_Prog_Wait		
	1.所有被请求的页的写操作完成，并且 lunblnterleave 被设为真	→	L_PP_Mpl_Sts
	2.被请求的页的写操作完成，并且 lunblnterleave 被清为 FALSE	→	L_PP_Sts
	3.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_PP_Cache	LUN 以指定的顺序执行以下操作： 1. lunStatus[6:5]被清为 00b。lunStatus[6]的值被标示到对象 2. lunLastConfirm 设为 15h 3. 等待数据输入的页寄存器变为可写 4. 开始编程操作		
	1.无条件的(Unconditional)	→	L_PP_Cache_Wait

L_PP_Cache_Wait	lunReturnState 被设为 L_PP_Cache_Wait		
	1.数据输入的页寄存器可写	→	L_PP_CacheRdy
	2. 对象请求接收到的状态或状态命令	→	L_Status_Execute

L_PP_CacheRdy	LUN 执行以下操作： 1. 如果 lunblnterleave 被设为 FALSE，则 lunStatus[1]被设为 lunStatus[0]的值 2. 如果 lunblnterleave 被设为真，则对于所有多层地址 x，lunFail[x][1]被设为 lunFail[x][0]的值。 3. lunStatus[6]被设为 1。lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_PP_CacheRdy_Wait

L_PP_CacheRdy_Wait	lunReturnState 设为 L_PP_CacheRdy_Wait		
	1.前一个缓存(cache)操作完成，并且 lunblnterleave 设为真	→	L_PP_Mpl_Cache_Sts
	2.前一个缓存操作完成	→	L_PP_Cache_Sts
	3.对象表示该 LUN 的编程请求	→	L_PP_Addr
	4.对象请求清除页寄存器	→	L_Idle_ClearPageReg
	5.对象请求接收到的状态或状态命令	→	L_Status_Execute

L_PP_Mpl	LUN 以指定的顺序执行以下操作： 1. lunblnterleave 设为真 2. lunStatus[6:5]被清为 00b。lunStatus[6]的值被标示到对象 3. lunLastConfirm 设为 11h 4. lunbMplNextCmd 被设为 FALSE 5. 如果支持重叠 interleaving，则 LUN 开始指定页的编程		
	1.无条件的(Unconditional)	→	L_PP_Mpl_Wait

L_PP_Mpl_Wait	lunReturnState 设为 L_PP_Mpl_Wait		
	1.一个重叠多层编程操作完成	→	L_PP_Mpl_Overlap
	2.前一个缓存编程(cache Program)结束	→	L_PP_Mpl_Cache_Sts
	3.LUN 准备好接收下一个编程命令，并且 lunbMplNextCmd 被设为 FALSE	→	L_PP_Mpl_NextCmd
	4.对象标示该 LUN 的编程请求，并且 lunbMplNextCmd 被设为真	→	L_PP_Addr
	5.对象请求被选中的列地址	→	L_Idle_Rd_ColSelect
	6.对象标示在数据输出中使用的层地址	→	L_Idle_Mpl_DataOutAddr
	7.对象请求接收到的状态或状态命令	→	L_Status_Execute
	8.从对象接收到读请求	→	L_Idle_Rd_Xfer

L_PP_Mpl_NextCmd	LUN 以指定的顺序执行以下操作： 1. lunbMplNextCmd 被设为真 2. 如果没有正在执行的阵列操作，则 lunStatus[5]被设为 1 3. lunStatus[6]被设为 1。lunStatus[6]被标示到对象		
	1.无条件的(Unconditional)	→	L_PP_Mpl_Wait

L_PP_Sts	LUN 以指定的顺序执行以下操作： 1. lunStatus[1]被设为前一个操作的编程状态 2. lunStatus[0]被设为最终操作的编程状态 3. lunStatus[6:5]被设为 11b 4. lunStatus[6]的值被标示到对象		
	1.无条件的(Unconditional)	→	L_Idle

L_PP_Cache_Sts	LUN 以指定的顺序执行以下操作： 1. lunStatus[0]被设为编程状态 2. lunStatus[5]被设为 1		
	1.无条件的(Unconditional)	→	lunReturnState

L_PP_Mpl_Cache_Sts	对于所有已完成的 cache 操作，LUN 以指定的顺序执行以下操作： 1. mplAddr 设为 cache 操作的 interleave 地址 2. lunFail[mplAddr][0]被设为编程状态 如果所有阵列操作都已完成，则 lunStatus[5]被设为 1		
1.无条件的(Unconditional)		→	lunReturnState

L_PP_Mpl_Overlap	对于所有已完成的重叠多层操作，LUN 以指定的顺序执行以下操作： 1. mplComplete 设为已完成操作的 interleave 地址 2. lunFail[mplComplete][0]被设为操作的编程状态 如果所有阵列操作都已完成，则 lunStatus[5]被设为 1		
1.无条件的(Unconditional)		→	lunReturnState

L_PP_Mpl_Sts	对于每个已完成的多层操作，LUN 以指定的顺序执行以下操作： 1. mplComplete 设为已完成操作的层地址 2. lunFail[mplComplete][1]被设为前一个操作的编程状态 3. lunFail[mplComplete][0]被设为最终操作的编程状态 lunStatus[6:5]被设为 11b，lunStatus[6]的值被标示到对象		
1.无条件的(Unconditional)		→	L_Idle

L_PP_ColSelect	根据被对象请求的、接收到的列地址来选择页寄存器中的列		
1.无条件的(Unconditional)		→	L_PP_WaitForData

L_PP_RowSelect	根据被对象请求的、接收到的行地址来选择要编程的块和页		
1.无条件的(Unconditional)		→	L_PP_WaitForData

A. CRC-16 参考代码

略（参见英文原文档）

B. SPARE SIZE 推荐

本附录描述了基于参数页中 ECC 要求的每页中 spare bytes 的推荐。该推荐用于行 NAND 的开发，但不适用于支持 EC NAND 的 device。表 103 列出了对于 2KB、4KB 以及 8KB page size device 的推荐。

Page Size	Number of bits ECC correctability	Spare Bytes Per Page Recommendation
2048 bytes	<= 8 bits	64 bytes
2048 bytes	> 8 bits	112 bytes
4096 bytes	<= 8 bits	128 bytes
4096 bytes	> 8 bits	218 or 224 bytes
8192 bytes	<= 8 bits	256 bytes
8192 bytes[2]	> 8 bits	448 bytes

NOTE:
1. The number of bits ECC correctability is based on a 512 byte codeword size.
2. If more correction is required than spare area size allows for with a 512 byte codeword size, it is recommended that the host use a larger ECC codeword size (e.g. 1KB, 2KB, etc). The device manufacturer may provide guidance on the ECC codeword size to use in the extended parameter page.

Table 103 Spare Area Size Recommendations for raw NAND

Host 从页寄存器以离散的单元(包括数据，元数据(metadata)以及 ECC 校验 bytes)传输 byte。该离散单元被推荐为偶数 bytes，用于支持 NV-DDR、NV-DDR2 或 NV-DDR3 接口的 device。

作为一个例子,假设 page size 为 8192 bytes,使用的 ECC 码字 size 为 1KB,那么每个离散单元中将有 1024bytes 数据被传输，该页中有 8 个离散单元的数据被传输。该页的 spare bytes 应该被分配，以便有足够的空间来存储元数据(metadata)和校验 bytes。当 Spare bytes 被除以 8 时，spare bytes 也应该是一个偶数 bytes(例如，该页中包含的离散单元的数量)。

C. Device 使用 PSL 自我初始化(Self-Initialization)

有些 device 在闪存阵列中存储了其自身闪存阵列的初始化信息。Device 可以在上电时或者上电后第一次 Reset 期间 load 这些信息。

制造商可以选择支持 PSL 作为其中一个制造商定义的引脚。如果支持 PSL，则应具有以下行为：

- PSL = 0 V：配置信息在上电时 load。IST 电流可以最大达 15mA，R/B_n 变为 1 的时间最大为 5ms。
- PSL = Vcc 或不连接：如果支持配置信息，则配置信息在上电后第一次 Reset 期间被 load。IST 电流的要求没有变化。

如果 device 不支持 PSL，则应该满足 IST 要求。

参见制造商 datasheet 来判定是否支持上电时自我初始化。

D. ICC 测量方法

本章节定义了用来测量 ICC 参数的技术，ICC 参数定义在 2.12 章。

用来测量 DC 和操作条件的一般测试条件定义在表 104 中。对于特定的接口，用来测量 DC 和操作条件的测试条件定义在表 105 中。

Parameter	Testing Condition
General conditions	1. Vcc = Vcc(min) to Vcc(max) 2. VccQ = VccQ(min) to VccQ(max) 3. CE_n = 0 V 4. WP_n = VccQ 5. IOUT = 0 mA 6. Measured across operating temperature range 7. N data input or data output cycles, where N is the number of bytes or words in the page 8. No multi-plane operations. 9. Sample a sufficient number of times to remove measurement variability. 10. Sample an equal ratio of page types that exist in a block. A page type is a group of page addresses and is commonly referred to as upper or lower page (or middle page for 3 bits per cell devices). 11. Choose the first good even/odd block pair beginning at blocks 2-3
Array preconditioning for ICC1 and ICC3	The array is preconditioned to match the data input pattern for ICC2.
Fixed wait time (no R/B_n polling)	ICC1: tR = tR(max) ICC2: tPROG = tPROG(max) ICC3: tBERS = tBERS(max)

Table 104 Common Testing Conditions for ICC

Parameter	SDR	NV-DDR	NV-DDR2/NV-DDR3
AC Timing Parameters	tWC = tWC(min) tRC = tRC(min) tADL = ~tADL(min) tCCS = ~tCCS(min) tRHW = ~tRHW(min)	tCK = tCK(avg) tADL =~tADL(min) tCCS = ~tCCS(min) tRHW = ~tRHW(min)	tWC = tWC(min) tRC = tRC(avg) tDSC = tDSC(avg) tADL = ~tADL(min) tCCS = ~tCCS(min)
Bus idle data pattern	IO[7:0] = FFh IO[15:0] = FFFFh	DQ[7:0] = FFh	DQ[7:0] = FFh
Repeated data pattern (Used for ICC2, ICC4R, ICCQ4R, ICCQ4W and ICC4W)	IO[7:0] = A5h, AAh, 5Ah, 55h IO[15:0] = A5A5h, AAAAh, 5A5Ah, 5555h	DQ[7:0] = A5h, AAh, 5Ah, 55h	DQ[7:0] = A5h, AAh, 5Ah, 55h
NOTES: 1. The value of tCK(avg), tRC(avg), and tDSC(avg) used should be the minimum value of the timing modes supported for the device. The NV-DDR, NV-DDR2, and NV-DDR3 timing modes supported by the device are indicated in the parameter page. 2. ICCQ4R testing is performed with default drive strength setting.			

Table 105 Data Interface Specific Testing Conditions for ICC

下图详细描述了 ICC1、ICC2、ICC3、ICC4R、ICC4W 以及 ICC5 的测试过程。

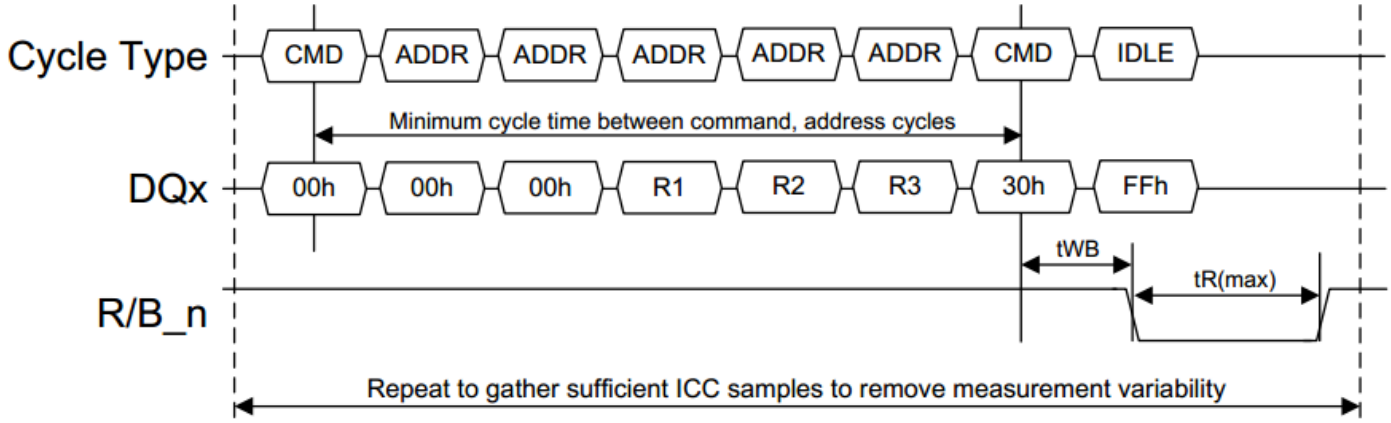


Figure 131 ICC1 measurement procedure

以下内容略