



# **Open NAND Flash Interface Specification**

Revision 2.3a  
19-October-2011

**Hynix Semiconductor**  
**Intel Corporation**  
**Micron Technology, Inc.**  
**Phison Electronics Corp.**  
**SanDisk Corporation**  
**Sony Corporation**  
**Spansion**

This 2.3a revision of the Open NAND Flash Interface specification ("Final Specification") is available for download at [www.onfi.org](http://www.onfi.org).

#### SPECIFICATION DISCLAIMER

THIS SPECIFICATION IS PROVIDED TO YOU "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE AUTHORS DO NOT WARRANT OR REPRESENT THAT SUCH USE WILL NOT INFRINGE SUCH RIGHTS. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

Copyright 2005-2011, Hynix Semiconductor, Intel Corporation, Micron Technology, Inc., Phison Electronics Corp., SanDisk Corporation, Sony Corporation, Spansion. All rights reserved.

For more information about ONFI, refer to the ONFI Workgroup website at [www.onfi.org](http://www.onfi.org).

All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

ONFI Workgroup Technical Editor:

Amber Huffman  
Intel Corporation  
2111 NE 25th Ave M/S JF2-53  
Hillsboro, OR 97124 USA  
Tel: (503) 264-7929  
Email: [amber.huffman@intel.com](mailto:amber.huffman@intel.com)

## Table of Contents

1. Introduction.....	1
1.1. Goals and Objectives .....	1
1.2. EZ NAND Overview .....	1
1.3. References .....	1
1.4. Definitions, abbreviations, and conventions.....	2
1.4.1. Definitions and Abbreviations .....	2
1.4.2. Conventions .....	4
2. Physical Interface .....	7
2.1. TSOP-48 and WSOP-48 Pin Assignments .....	7
2.2. LGA-52 Pad Assignments.....	10
2.3. BGA-63 Ball Assignments.....	12
2.4. BGA-100 Ball Assignments.....	16
2.5. Signal Descriptions .....	20
2.6. CE_n Signal Requirements.....	26
2.6.1. Source Synchronous Data Interface Requirements .....	26
2.7. Absolute Maximum DC Ratings .....	26
2.8. Recommended DC Operating Conditions.....	27
2.8.1. I/O Power (VccQ) and I/O Ground (VssQ).....	27
2.9. AC Overshoot/Undershoot Requirements .....	28
2.10. DC and Operating Characteristics.....	28
2.11. Calculating Pin Capacitance .....	31
2.12. Staggered Power-up.....	32
2.13. Power Cycle Requirements.....	32
2.14. Independent Data Buses .....	32
2.15. Bus Width Requirements.....	32
2.16. Ready/Busy (R/B_n) Requirements .....	33
2.16.1. Power-On Requirements.....	33
2.16.2. R/B_n and SR[6] Relationship .....	34
2.17. Write Protect.....	34
3. Memory Organization .....	35
3.1. Addressing .....	36
3.1.1. Multi-plane Addressing .....	37
3.1.2. Logical Unit Selection .....	38
3.1.3. Multiple LUN Operation Restrictions.....	38
3.2. Factory Defect Mapping .....	39
3.2.1. Device Requirements.....	39
3.2.2. Host Requirements .....	40
3.3. Extended ECC Information Reporting.....	40
3.3.1. Byte 0: Number of bits ECC correctability .....	41
3.3.2. Byte 1: Codeword size .....	41
3.3.3. Byte 2-3: Bad blocks maximum per LUN.....	41
3.3.4. Byte 4-5: Block endurance.....	41
3.4. Discovery and Initialization.....	42
3.4.1. CE_n Discovery .....	42
3.4.2. Target Initialization.....	42
4. Data Interface and Timing .....	44
4.1. Data Interface Types.....	44
4.1.1. Signal Function Reassignment .....	44
4.1.2. Bus State .....	45
4.1.3. Source Synchronous and Repeat Bytes.....	46
4.1.4. Data Interface / Timing Mode Transitions.....	47
4.2. Timing Parameters.....	48
4.2.1. General Timings.....	48
4.2.2. Asynchronous .....	49

4.2.3.	Source Synchronous.....	56
	<b>Number of LUNs per x8 data bus .....</b>	<b>64</b>
4.3.	Timing Diagrams .....	69
4.3.1.	Asynchronous .....	69
4.3.2.	Source Synchronous.....	76
4.4.	Command Examples.....	88
4.4.1.	Asynchronous .....	88
4.4.2.	Source Synchronous.....	91
5.	Command Definition.....	96
5.1.	Command Set .....	96
5.2.	Command Descriptions.....	99
5.3.	Reset Definition .....	102
5.4.	Synchronous Reset Definition.....	102
5.5.	Reset LUN Definition.....	103
5.6.	Read ID Definition .....	104
5.7.	Read Parameter Page Definition .....	106
5.7.1.	Parameter Page Data Structure Definition .....	108
5.7.2.	Extended Parameter Page Data Structure Definition .....	122
5.8.	Read Unique ID Definition.....	125
5.9.	Block Erase Definition .....	127
5.10.	Read Status Definition .....	127
5.11.	Read Status Enhanced Definition .....	130
5.12.	Read Status and Read Status Enhanced required usage .....	131
5.13.	Status Field Definition.....	131
5.14.	Read Definition .....	132
5.15.	Read Cache Definition.....	135
5.16.	Page Program Definition .....	139
5.17.	Page Cache Program Definition .....	141
5.18.	Copyback Definition.....	144
5.19.	Small Data Move .....	149
5.20.	Change Read Column Definition.....	152
5.21.	Change Read Column Enhanced Definition .....	152
5.22.	Change Write Column Definition .....	154
5.23.	Change Row Address Definition .....	154
5.24.	Set Features Definition .....	156
5.25.	Get Features Definition.....	158
5.26.	Feature Parameter Definitions .....	158
5.26.1.	Timing Mode.....	159
5.26.2.	I/O Drive Strength.....	160
5.26.3.	EZ NAND control.....	160
6.	Multi-plane Operations .....	161
6.1.	Requirements .....	161
6.2.	Status Register Behavior .....	162
6.3.	Multi-plane Page Program .....	162
6.4.	Multi-plane Copyback Read and Program .....	165
6.5.	Multi-plane Block Erase .....	168
6.6.	Multi-plane Read .....	169
7.	Behavioral Flows .....	174
7.1.	Target behavioral flows .....	174
7.1.1.	Variables .....	174
7.1.2.	Idle states.....	174
7.1.3.	Idle Read states .....	176
7.1.4.	Reset command states .....	178
7.1.5.	Read ID command states .....	180
7.1.6.	Read Parameter Page command states.....	181
7.1.7.	Read Unique ID command states.....	182

7.1.8.	Page Program and Page Cache Program command states .....	183
7.1.9.	Block Erase command states .....	186
7.1.10.	Read command states .....	188
7.1.11.	Set Features command states .....	190
7.1.12.	Get Features command states .....	191
7.1.13.	Read Status command states .....	191
7.1.14.	Read Status Enhanced command states .....	192
7.2.	LUN behavioral flows .....	193
7.2.1.	Variables .....	193
7.2.2.	Idle command states .....	193
7.2.3.	Idle Read states .....	195
7.2.4.	Status states .....	196
7.2.5.	Reset states .....	197
7.2.6.	Block Erase command states .....	197
7.2.7.	Read command states .....	199
7.2.8.	Page Program and Page Cache Program command states .....	201
A.	Sample Code for CRC-16 (Informative) .....	205
B.	Spare Size Recommendations (Informative) .....	207
C.	Device Self-Initialization with PSL (Informative) .....	208
D.	ICC Measurement Methodology .....	209
E.	Measuring Timing Parameters to/From Tri-State .....	217
F.	EZ NAND: End to End Data Path Protection (INFORMATIVE) .....	218

# 1. Introduction

## 1.1. Goals and Objectives

This specification defines a standardized NAND Flash device interface that provides the means for a system to be designed that supports a range of NAND Flash devices without direct design pre-association. The solution also provides the means for a system to seamlessly make use of new NAND devices that may not have existed at the time that the system was designed.

Some of the goals and requirements for the specification include:

- Support range of device capabilities and new unforeseen innovation
- Consistent with existing NAND Flash designs providing orderly transition to ONFI
- Capabilities and features are self-described in a parameter page such that hard-coded chip ID tables in the host are not necessary
- Flash devices are interoperable and do not require host changes to support a new Flash device
- Define a higher speed NAND interface that is compatible with existing NAND Flash interface
- Allow for separate core (Vcc) and I/O (VccQ) power rails
- Support for offloading NAND lithography specific functionality to a controller stacked in the NAND package (EZ NAND)

## 1.2. EZ NAND Overview

EZ NAND includes the control logic packaged together with NAND to perform the NAND management functionality that is lithography specific (e.g. ECC), while retaining the NAND protocol infrastructure. EZ NAND delivers an ECC offloaded solution with minimal command and/or protocol changes. The device parameter page will specify if EZ NAND is supported.

## 1.3. References

This specification is developed in part based on existing common NAND Flash device behaviors, including the behaviors defined in the following datasheets:

- Micron MT29F8G08ABC data sheet available at <http://www.micron.com/products/partdetail?part=MT29F8G08ABCBBH1-12>
- Numonyx NAND04GB2D data sheet available at [http://www.numonyx.com/Documents/Datasheets/NAND04G-B2D\\_NAND08G-BxC.pdf](http://www.numonyx.com/Documents/Datasheets/NAND04G-B2D_NAND08G-BxC.pdf)

The specification also makes reference to the following specifications and standards:

- ONFI Block Abstracted NAND revision 1.1. Specification is available at <http://www.onfi.org>.

## **1.4. Definitions, abbreviations, and conventions**

### **1.4.1. Definitions and Abbreviations**

The terminology used in this specification is intended to be self-sufficient and does not rely on overloaded meanings defined in other specifications. Terms with specific meaning not directly clear from the context are clarified in the following sections.

#### **1.4.1.1. address**

The address is comprised of a row address and a column address. The row address identifies the page, block, and LUN to be accessed. The column address identifies the byte or word within a page to access. The least significant bit of the column address shall always be zero in the source synchronous data interface.

#### **1.4.1.2. asynchronous**

Asynchronous is when data is latched with the WE\_n signal for writes and RE\_n signal for reads.

#### **1.4.1.3. block**

Consists of multiple pages and is the smallest addressable unit for erase operations.

#### **1.4.1.4. column**

The byte (x8 devices) or word (x16 devices) location within the page register.

#### **1.4.1.5. defect area**

The defect area is where factory defects are marked by the manufacturer. Refer to section 3.2.

#### **1.4.1.6. device**

The packaged NAND unit. A device consists of one or more targets.

#### **1.4.1.7. DDR**

Acronym for double data rate.

#### **1.4.1.8. Dword**

A Dword is thirty-two (32) bits of data. A Dword may be represented as 32 bits, as two adjacent words, or as four adjacent bytes. When shown as bits the least significant bit is bit 0 and most significant bit is bit 31. The most significant bit is shown on the left. When shown as words the least significant word (lower) is word 0 and the most significant (upper) word is word 1. When shown as bytes the least significant byte is byte 0 and the most significant byte is byte 3. See Figure 1 for a description of the relationship between bytes, words, and Dwords.

#### **1.4.1.9. latching edge**

The latching edge describes the edge of the CLK or the DQS signal that the contents of the data bus are latched on for the source synchronous data interface. For data cycles the latching edge is both the rising and falling edges of the DQS signal. For command and address cycles the latching edge is the rising edge of the CLK signal.

#### **1.4.1.10. LUN (logical unit number)**

The minimum unit that can independently execute commands and report status. There are one or more LUNs per target.

#### **1.4.1.11. na**

na stands for “not applicable”. Fields marked as “na” are not used.

#### **1.4.1.12. O/M**

O/M stands for Optional/Mandatory requirement. When the entry is set to “M”, the item is mandatory. When the entry is set to “O”, the item is optional.

#### **1.4.1.13. page**

The smallest addressable unit for read and program operations.

#### **1.4.1.14. page register**

Register used to read data from that was transferred from the Flash array. For program operations, the data is placed in this register prior to transferring the data to the Flash array. If EZ NAND is supported a buffer exists in the EZ NAND controller that may be used to facilitate Copyback operations. Refer to section 5.18 for information on EZ NAND Copyback operations.

#### **1.4.1.15. partial page**

A portion of the page, referred to as a partial page, may be programmed if the target supports more than one program per page as indicated in the parameter page. The host may choose to read only a portion of the data from the page register in a read operation; this portion may also be referred to as a partial page.

#### **1.4.1.16. read request**

A read request is a data output cycle request from the host that results in a data transfer from the device to the host. Refer to section 4.1.2 for information on data output cycles.

#### **1.4.1.17. row**

Refers to the block and page to be accessed.

#### **1.4.1.18. source synchronous**

Source synchronous is when the strobe (DQS) is forwarded with the data to indicate when the data should be latched. The strobe signal, DQS, can be thought of as an additional data bus bit.

#### **1.4.1.19. SR[ ]**

SR refers to the status register contained within a particular LUN. SR[x] refers to bit x in the status register for the associated LUN. Refer to section 5.13 for the definition of bit meanings within the status register.

#### **1.4.1.20. target**

An independent Flash component with its own CE\_n signal.

#### **1.4.1.21. Uncorrectable Bit Error Rate, or ratio (UBER)**

A metric for the rate of occurrence of data errors, equal to the number of data errors per bits read. Mathematically, it may be represented as:

$$UBER = \text{cumulative number of data errors} / \text{cumulative number of bits read}$$

Note: The cumulative number of bits read is the sum of all bits of data read back from the device, with multiple reads of the same memory bit as multiple bits read. For example, if a 100GB device is read ten times then there would be about 1TB ( $8 \times 10^{12}$  bits) read. The cumulative number of data errors is the count of the physical pages for which the device fails to return correct data.

Note: This metric only applies to devices that support EZ NAND. EZ NAND delivers an ECC offloaded solution, and thus this metric applies. For raw NAND solutions where the host provides the ECC solution, the UBER is dependent on the host controller capability and UBER for that solution is not within the scope of this specification.

#### **1.4.1.22. Vtt**

Termination voltage.

#### **1.4.1.23. word**

A word is sixteen (16) bits of data. A word may be represented as 16 bits or as two adjacent bytes. When shown as bits the least significant bit is bit 0 and most significant bit is bit 15. The most significant bit is shown on the left. When shown as bytes the least significant byte (lower) is byte 0 and the most significant byte (upper) is byte 1. See Figure 1 for a description of the relationship between bytes, words and Dwords.

### **1.4.2. Conventions**

The names of abbreviations and acronyms used as signal names are in all uppercase (e.g., CE\_n). Fields containing only one bit are usually referred to as the "name" bit instead of the "name" field. Numerical fields are unsigned unless otherwise indicated.

#### **1.4.2.1. Precedence**

If there is a conflict between text, figures, state machines, timing diagrams, and tables, the precedence shall be state machines and timing diagrams, tables, figures, and then text.

#### **1.4.2.2. Keywords**

Several keywords are used to differentiate between different levels of requirements.

##### **1.4.2.2.1. mandatory**

A keyword indicating items to be implemented as defined by this specification.

##### **1.4.2.2.2. may**

A keyword that indicates flexibility of choice with no implied preference.

##### **1.4.2.2.3. optional**

A keyword that describes features that are not required by this specification. However, if any optional feature defined by the specification is implemented, the feature shall be implemented in the way defined by the specification.

**1.4.2.2.4. reserved**

A keyword indicating reserved bits, bytes, words, fields, and opcode values that are set-aside for future standardization. Their use and interpretation may be specified by future extensions to this or other specifications. A reserved bit, byte, word, or field shall be cleared to zero, or in accordance with a future extension to this specification. The recipient shall not check reserved bits, bytes, words, or fields.

**1.4.2.2.5. shall**

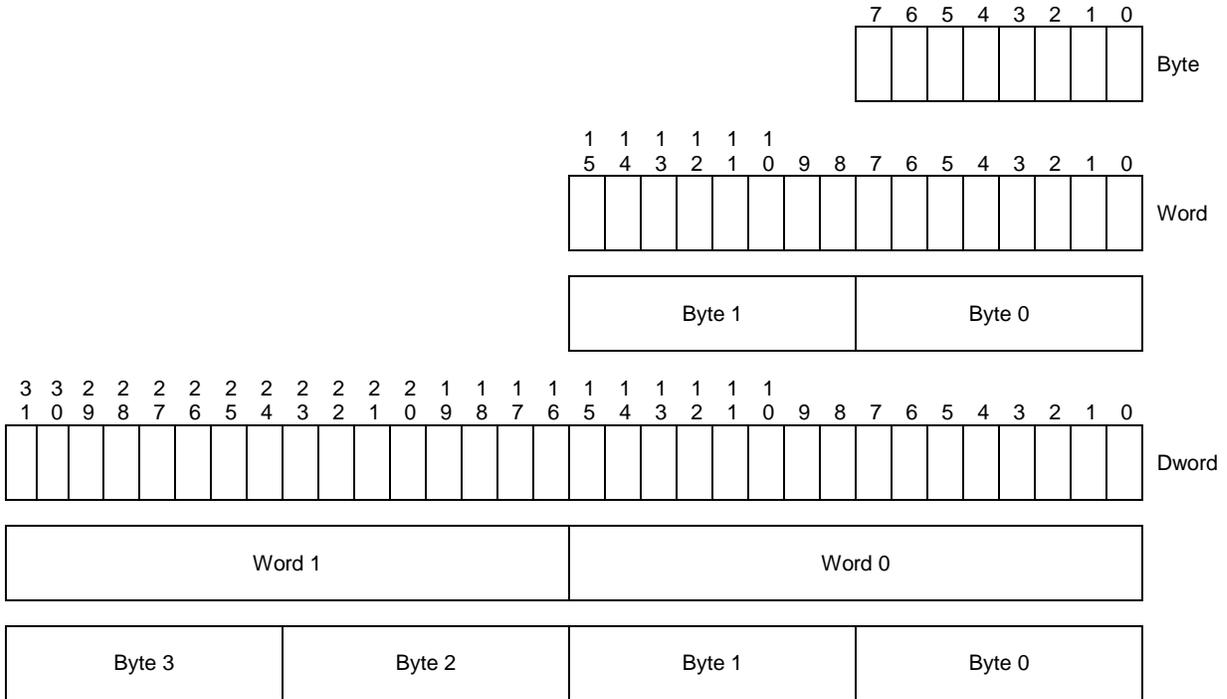
A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to the specification.

**1.4.2.2.6. should**

A keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “it is recommended”.

**1.4.2.3. Byte, word and Dword Relationships**

Figure 1 illustrates the relationship between bytes, words and Dwords.



**Figure 1 Byte, word and Dword relationships**

**1.4.2.4. Behavioral Flow Diagrams**

For each function to be completed a state machine approach is used to describe the sequence and externally visible behavior requirements. Each function is composed of several states to accomplish a set goal. Each state of the set is described by an individual state table. Table 1

below shows the general layout for each of the state tables that comprise the set of states for the function.

State name	Action list		
	Transition condition 0	→	<b>Next state 0</b>
	Transition condition 1	→	<b>Next state 1</b>

**Table 1 State Table Cell Description**

Each state is identified by a unique state name. The state name is a brief description of the primary action taken during the state. Actions to take while in the state are described in the action list.

Each transition is identified by a transition label and a transition condition. The transition label consists of the state designator of the state from which the transition is being made followed by the state designator of the state to which the transition is being made. The transition condition is a brief description of the event or condition that causes the transition to occur and may include a transition action that is taken when the transition occurs. This action is described fully in the transition description text. Transition conditions are listed in priority order and are not required to be mutually exclusive. The first transition condition that evaluates to be true shall be taken.

Upon entry to a state, all actions to be executed in that state are executed. If a state is re-entered from itself, all actions to be executed in the state are executed again.

It is assumed that all actions are executed within a state and that transitions from state to state are instantaneous.

## 2. Physical Interface

### 2.1. TSOP-48 and WSOP-48 Pin Assignments

Figure 2 defines the pin assignments for devices using 48-pin TSOP or 48-pin WSOP packaging for 8-bit data access. Figure 3 defines the pin assignments for devices using 48-pin TSOP or 48-pin WSOP packaging for 16-bit data access. The package with 16-bit data access does not support the source synchronous data interface. The physical dimensions of the TSOP package is defined in the JEDEC document MO-142 variation DD. The physical dimensions of the WSOP package is defined in the JEDEC document MO-259.



**Figure 2** 48-pin TSOP/WSOP pinout for 8-bit data access

**NOTE:** For a source synchronous capable part, pin 35 is not used when configured in the asynchronous data interface. Specifically, VSP2 is present for asynchronous only parts.

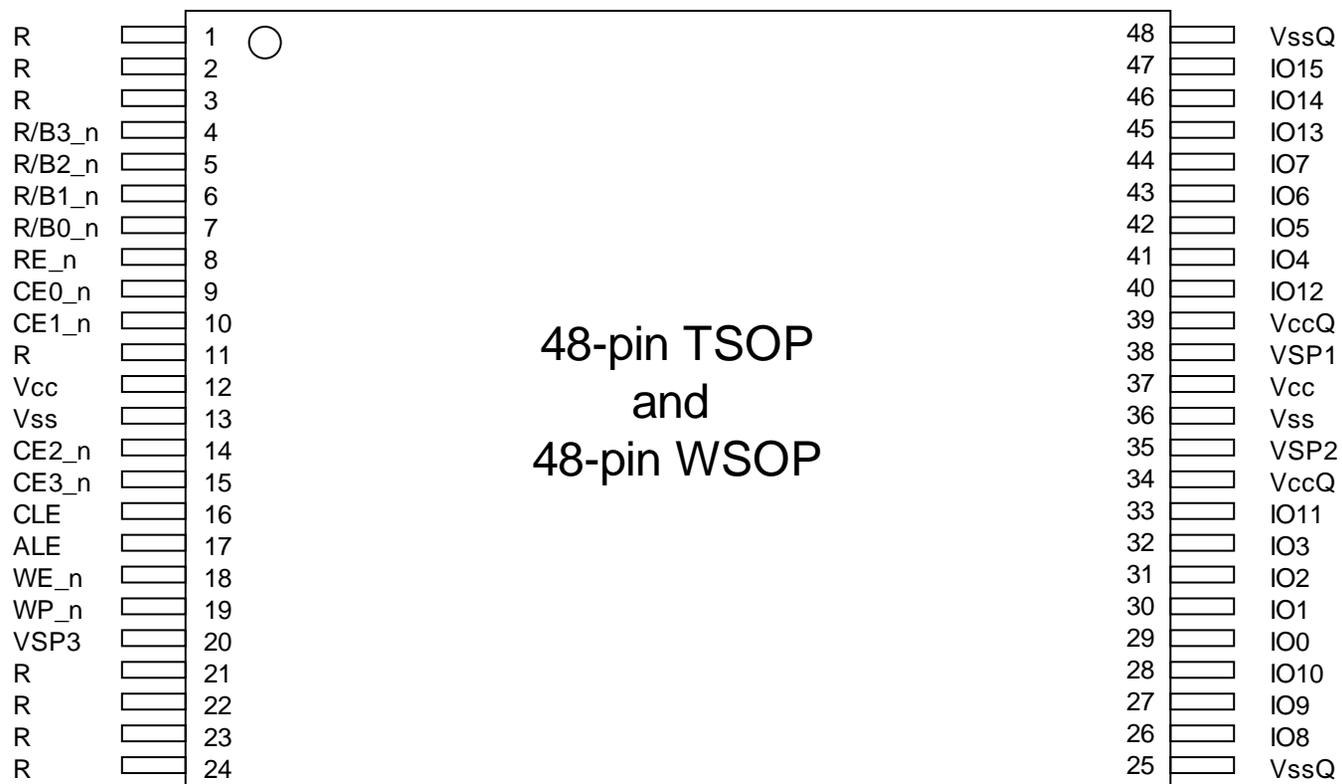


Figure 3 48-pin TSOP/WSOP pinout for 16-bit data access

## 2.2. LGA-52 Pad Assignments

Figure 4 defines the pad assignments for devices using 52-pad LGA packaging with 8-bit data access. An option is specified for two independent 8-bit data buses. Figure 5 defines the pad assignments for devices using 52-pad LGA packaging with 16-bit data access. The physical dimensions of the package are 12mmx17mm or 14mmx18mm. Figure 6 defines the pad spacing requirements for the 52-pad LGA package for both package dimensions. These LGA packages do not support the source synchronous data interface.

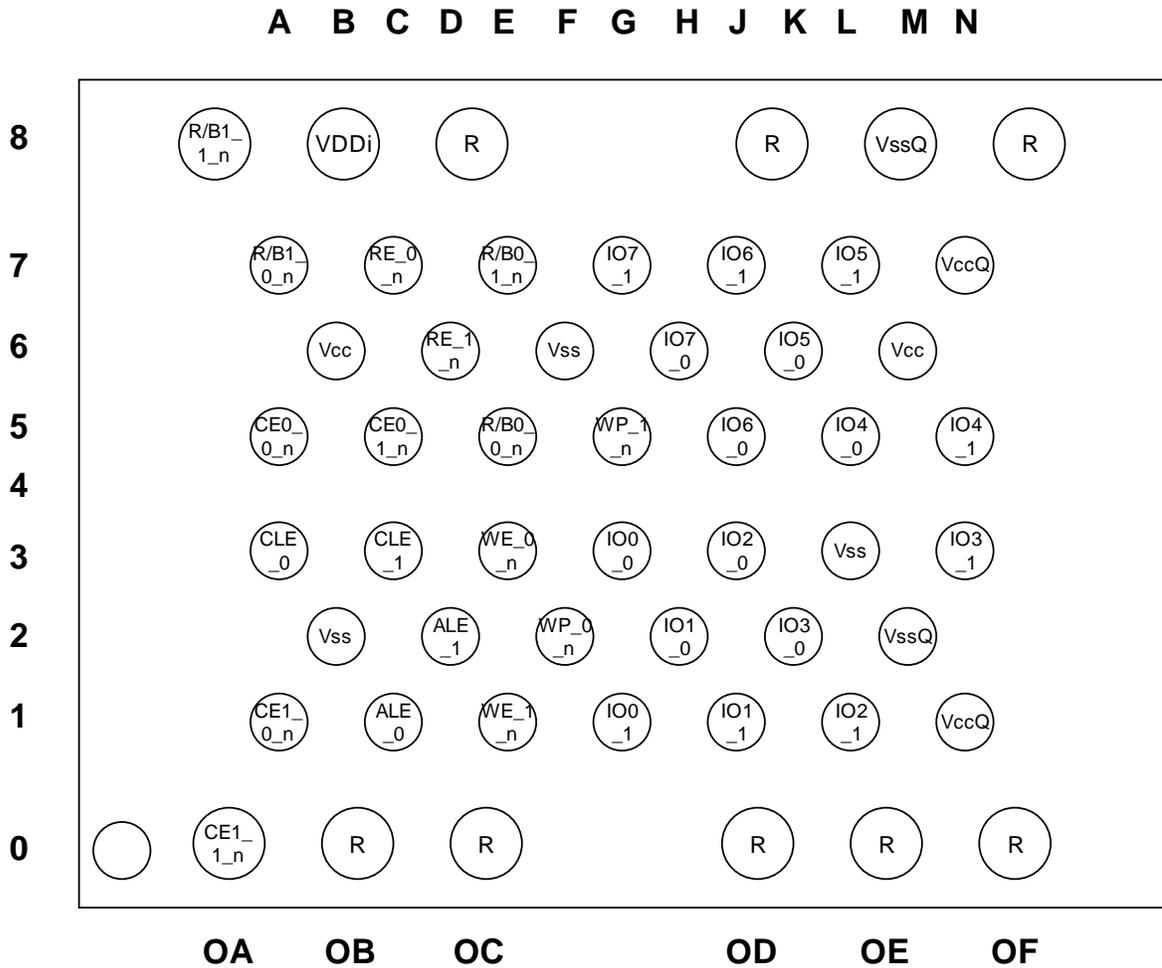


Figure 4 LGA pinout for 8-bit data access

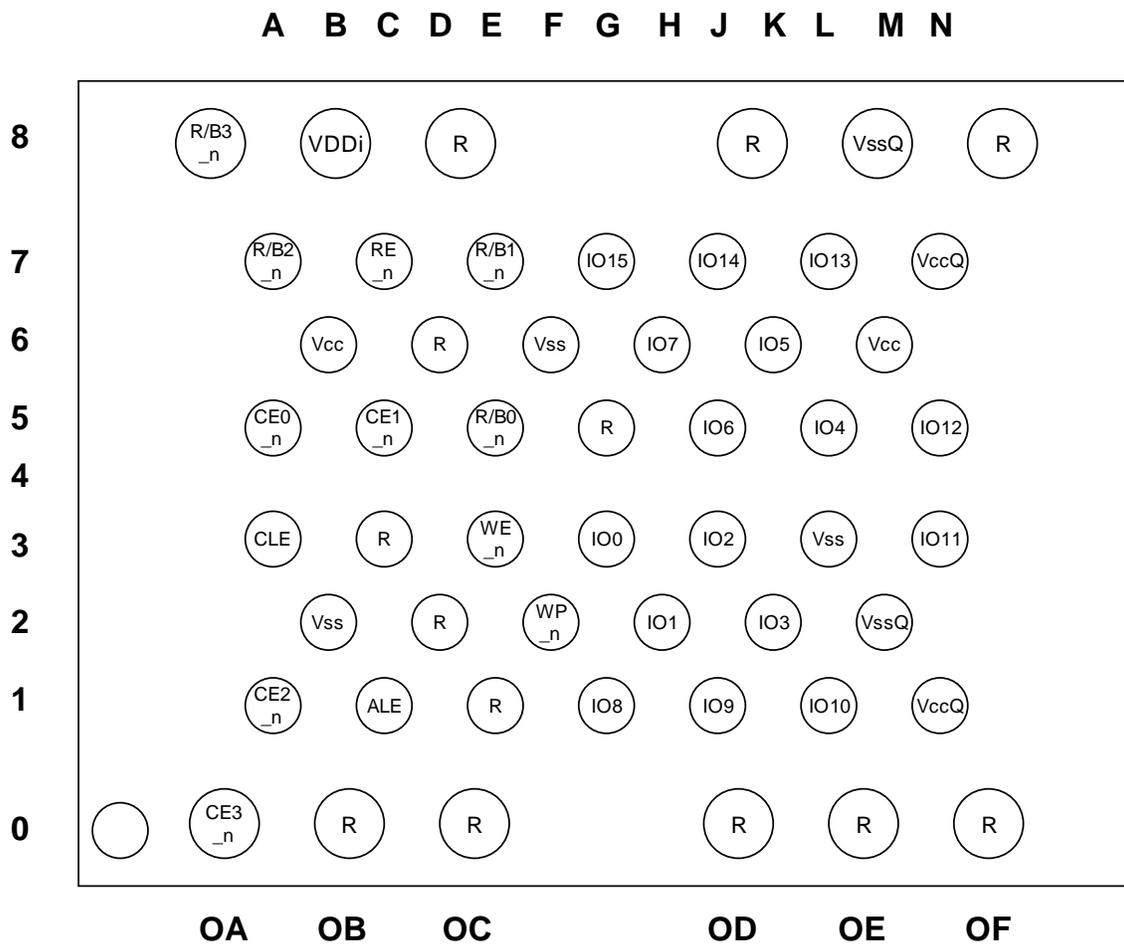


Figure 5 LGA pinout for 16-bit data access



	1	2	3	4	5	6	7	8	9	10
A	R	R							R	R
B	R								R	R
C			WP_n	ALE	VSS	CE0_n	WE_n	R/B0_n		
D			VCC	RE_n	CLE	CE1_n	CE2_n	R/B1_n		
E			R	R	R	R	CE3_n	R/B2_n		
F			R	R	R	R	VSS	R/B3_n		
G			VSP3	VCC	VSP1	R	R	VSP2		
H			R	IO0	R	R	R	VCCQ		
J			R	IO1	R	VCCQ	IO5	IO7		
K			VSSQ	IO2	IO3	IO4	IO6	VSSQ		
L	R	R							R	R
M	R	R							R	R

**Figure 7** BGA-63 ball assignments for 8-bit data access, asynchronous only data interface

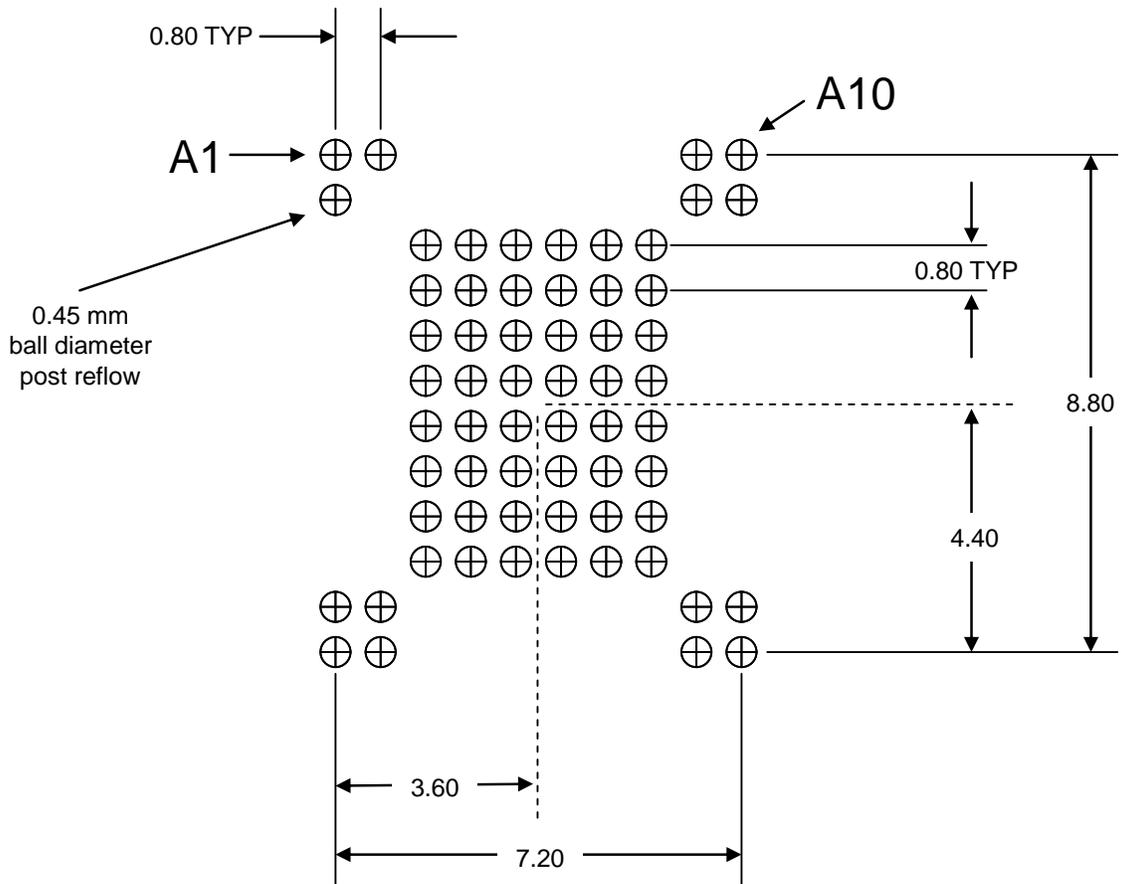
Note that WE\_n is located at ball H7 when a source synchronous capable part is used in asynchronous mode.

	1	2	3	4	5	6	7	8	9	10
A	R	R							R	R
B	R								R	R
C			WP_n	ALE	VSS	CE0_n	R	R/B0_n		
D			VCC	W/R_n	CLE	CE1_n	CE2_n	R/B1_n		
E			R	R	R	R	CE3_n	R/B2_n		
F			R	R	VREFQ	R	VSS	R/B3_n		
G			VSP3	VCC	VSP1	R	R	VSP2		
H			R	DQ0	DQS_c	CLK_c	CLK_t	VCCQ		
J			R	DQ1	DQS_t	VCCQ	DQ5	DQ7		
K			VSSQ	DQ2	DQ3	DQ4	DQ6	VSSQ		
L	R	R							R	R
M	R	R							R	R

**Figure 8 BGA-63 ball assignments for 8-bit data access, source synchronous data interface**

	1	2	3	4	5	6	7	8	9	10
A	R	R							R	R
B	R								R	R
C			WP_n	ALE	VSS	CE0_n	WE_n	R/B0_n		
D			VCC	RE_n	CLE	CE1_n	CE2_n	R/B1_n		
E			R	R	R	R	CE3_n	R/B2_n		
F			R	R	R	R	VSS	R/B3_n		
G			VSP3	VCC	VSP1	IO13	IO15	VSP2		
H			IO8	IO0	IO10	IO12	IO14	VCCQ		
J			IO9	IO1	IO11	VCCQ	IO5	IO7		
K			VSSQ	IO2	IO3	IO4	IO6	VSSQ		
L	R	R							R	R
M	R	R							R	R

**Figure 9 BGA-63 ball assignments for 16-bit, asynchronous only data access**



**Figure 10 BGA-63 ball spacing requirements (top view, dimensions in millimeters)**

## 2.4. BGA-100 Ball Assignments

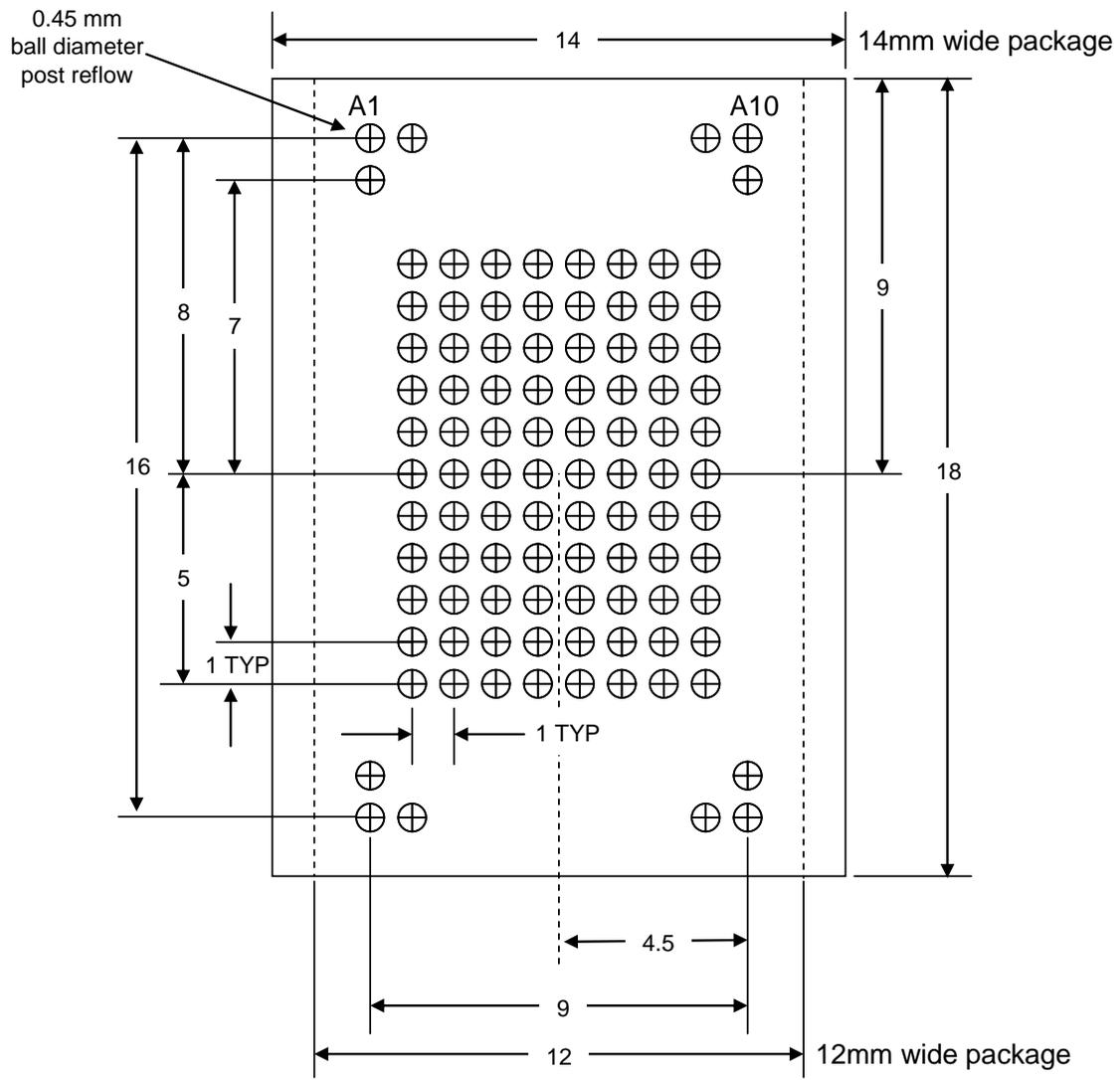
Figure 11 defines the ball assignments for devices using 100-ball BGA packaging with dual 8-bit data access for the asynchronous data interface. Figure 12 defines the ball assignments for devices using 100-ball BGA packaging with dual 8-bit data access for the source synchronous data interface. Figure 13 defines the ball spacing requirements for the 100-ball BGA package. The solder ball diameter is 0.45 mm post reflow. The 100-ball BGA has two package sizes: 12mm x 18mm and 14mm x 18mm.

	1	2	3	4	5	6	7	8	9	10
A	R	R							R	R
B	R									R
C										
D		R	RFT	VSP3_1	WP_1_n	VSP2_1	VSP1_1	RFT	R	
E		R	RFT	VSP3_0	WP_0_n	VSP2_0	VSP1_0	RFT	VDDi	
F		VCC	VCC	VCC	VCC	VCC	VCC	VCC	VCC	
G		VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	
H		VSSQ	VCCQ	R	R	R/B0_1_n	R/B1_1_n	VCCQ	VSSQ	
J		IO0_1	IO2_1	ALE_1	CE1_1_n	R/B0_0_n	R/B1_0_n	IO5_1	IO7_1	
K		IO0_0	IO2_0	ALE_0	CE1_0_n	CE0_1_n	CE0_0_n	IO5_0	IO7_0	
L		VCCQ	VSSQ	VCCQ	CLE_1	RE_1_n	VCCQ	VSSQ	VCCQ	
M		IO1_1	IO3_1	VSSQ	CLE_0	RE_0_n	VSSQ	IO4_1	IO6_1	
N		IO1_0	IO3_0	NC	NC	NC	WE_1_n	IO4_0	IO6_0	
P		VSSQ	VCCQ	NC	NC	NC	WE_0_n	VCCQ	VSSQ	
R										
T	R									R
U	R	R							R	R

**Figure 11 BGA-100 ball assignments for dual 8-bit data access, asynchronous data interface**

	1	2	3	4	5	6	7	8	9	10
A	R	R							R	R
B	R									R
C										
D		R	RFT	VSP3_1	WP_1_n	VSP2_1	VSP1_1	RFT	R	
E		R	RFT	VSP3_0	WP_0_n	VSP2_0	VSP1_0	RFT	VDDi	
F		VCC	VCC	VCC	VCC	VCC	VCC	VCC	VCC	
G		VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	
H		VSSQ	VCCQ	VREFQ_1	VREFQ_0	R/B0_1_n	R/B1_1_n	VCCQ	VSSQ	
J		DQ0_1	DQ2_1	ALE_1	CE1_1_n	R/B0_0_n	R/B1_0_n	DQ5_1	DQ7_1	
K		DQ0_0	DQ2_0	ALE_0	CE1_0_n	CE0_1_n	CE0_0_n	DQ5_0	DQ7_0	
L		VCCQ	VSSQ	VCCQ	CLE_1	W/R_1_n	VCCQ	VSSQ	VCCQ	
M		DQ1_1	DQ3_1	VSSQ	CLE_0	W/R_0_n	VSSQ	DQ4_1	DQ6_1	
N		DQ1_0	DQ3_0	DQS_1_c	DQS_1_t	CLK_1_c	CLK_1_t	DQ4_0	DQ6_0	
P		VSSQ	VCCQ	DQS_0_c	DQS_0_t	CLK_0_c	CLK_0_t	VCCQ	VSSQ	
R										
T	R									R
U	R	R							R	R

Figure 12 BGA-100 ball assignments for dual 8-bit data access, source synchronous data interface



**Figure 13 BGA-100 ball spacing requirements (top view, dimensions in millimeters)**

## 2.5. Signal Descriptions

Table 2 provides the signal descriptions.

Where the pin function has an inverted logic sense, that is, the function is true or invoked for a low signal, the overbar, trailing slash (/), #, or \_n symbol is appended to the symbol. The \_t and \_c symbols represent true and complementary logic states for differential pairs.

Signal Name	Input / Output	Description
R/B_x_n	O	<b>Ready/Busy</b> The Ready/Busy signal indicates the target status. When low, the signal indicates that one or more LUN operations are in progress. This signal is an open drain output and requires an external pull-up. See section 2.16 for requirements.
RE_x_n	I	<b>Read Enable</b> The Read Enable signal enables serial data output. This signal shares the same pin as W/R_x_n in the source synchronous data interface.
W/R_x_n	I	<b>Write/Read Direction</b> The Write/Read Direction signal indicates the owner of the DQ bus and DQS signal in the source synchronous data interface. This signal shares the same pin as RE_x_n in the asynchronous data interface.
CE_x_n	I	<b>Chip Enable</b> The Chip Enable signal selects the target. When Chip Enable is high and the target is in the ready state, the target goes into a low-power standby state. When Chip Enable is low, the target is selected. See section 2.6 for additional requirements.
Vcc	I	<b>Power</b> The Vcc signal is the power supply to the device.
VccQ	I	<b>I/O Power</b> The VccQ signal is the power supply for input and/or output signals. Refer to section 2.8.1.
Vss	I	<b>Ground</b> The Vss signal is the power supply ground.
VssQ	I	<b>I/O Ground</b> The VssQ signal is the ground for input and/or output signals. Refer to section 2.8.1.
VREFQ_x	I	<b>Voltage Reference</b> This signal is reserved for future use.
VDDi	na	<b>ASIC Voltage Control</b> This signal is used to assist in stabilizing the internal power supply to a NAND controller ASIC (e.g. EZ NAND) by connecting to an external capacitor.
CLE_x	I	<b>Command Latch Enable</b> The Command Latch Enable signal is one of the signals used by the host to indicate the type of bus cycle (command, address, data). Refer to section 4.1.2.
ALE_x	I	<b>Address Latch Enable</b> The Address Latch Enable signal is one of the signals used by the host to indicate the type of bus cycle (command, address, data). Refer to section 4.1.2.

Signal Name	Input / Output	Description
WE_x_n	I	<b>Write Enable</b> The Write Enable signal controls the latching of input data in the asynchronous data interface. Data, commands, and addresses are latched on the rising edge of WE_x_n. This signal shares the same pin as CLK_x in the source synchronous data interface.
CLK_x_t	I	<b>Clock</b> The Clock signal is used as the clock in the source synchronous data interface. This signal shares the same pin as WE_x_n in the asynchronous data interface.
CLK_x_c	I	<b>Clock Complement</b> This signal is reserved for future use.
WP_x_n	I	<b>Write Protect</b> The Write Protect signal disables Flash array program and erase operations. See section 2.17 for requirements.
IO0_0 – IO7_0 (DQ0_0 – DQ7_0)	I/O	<b>I/O Port 0, bits 0-7</b> The I/O port is an 8-bit wide bidirectional port for transferring address, command, and data to and from the device. Also known as DQ0_0 – DQ7_0 for the source synchronous data interface.
DQS_x_t	I/O	<b>Data Strobe</b> The data strobe signal that indicates the data valid window for the source synchronous data interface.
DQS_x_c	I/O	<b>Data Strobe Complement</b> This signal is reserved for future use.
IO8 – IO15	I/O	<b>I/O Port 0, bits 8-15</b> These signals are used in a 16-bit wide target configuration. The signals are the upper 8 bits for the 16-bit wide bidirectional port used to transfer data to and from the device.
IO0_1 – IO7_1 (DQ0_1 – DQ7_1)	I/O	<b>I/O Port 1, bits 0-7</b> The I/O port is an 8-bit wide bidirectional port for transferring address, command, and data to and from the device. These pins may be used as an additional 8-bit wide bidirectional port for devices that support two independent data buses. Also known as DQ0_1 – DQ7_1 for the source synchronous data interface.
VSP_x		<b>Vendor Specific</b> The function of these signals is defined and specified by the NAND vendor. Devices shall have an internal pull-up or pull-down resistor on these signals to yield ONFI compliant behavior when a signal is not connected by the host. Any VSP signal not used by the NAND vendor shall not be connected internal to the device.
R		<b>Reserved</b> These pins shall not be connected by the host.
RFT		<b>Reserved for Test</b> These pins shall not be connected by the host.

**Table 2 Signal descriptions**

Table 3 provides the signal mapping to pin/pad/ball for each package type listed within the ONFI specification. These signal mappings are required if the packages listed in this specification are implemented. The “Async Only” signal mappings apply to packages where the device is not source synchronous capable. When the device is source synchronous capable, the “Src Sync” signal mappings shall be used. If a signal is marked as “na” then the corresponding package

does not implement that signal. Any signal that does not have an associated number is implicitly numbered "0". For example, WP\_n is equivalent to WP0\_n.

Devices may be implemented with other package types and be ONFI compliant if all other ONFI requirements within this specification are satisfied.

Signal Name	M/O/R	TSOP / WSOP Async only x8	TSOP / WSOP Src Sync x8	TSOP / WSOP Async only x16	LGA Async only x8	LGA Async only x16	BGA-63 Async only x8	BGA-63 Src Sync x8	BGA-63 Async only x16	BGA-100 Async only x8	BGA-100 Src Sync x8
R/B0_n	M	7	7	7	na	E5	C8	C8	C8	na	na
R/B1_n	O	6	6	6	na	E7	D8	D8	D8	na	na
R/B2_n	O	5	5	5	na	A7	E8	E8	E8	na	na
R/B3_n	O	4	4	4	na	OA8	F8	F8	F8	na	na
R/B0_0_n	M	na	na	na	E5	na	na	na	na	J6	J6
R/B0_1_n	O	na	na	na	E7	na	na	na	na	H6	H6
R/B1_0_n	O	na	na	na	A7	na	na	na	na	J7	J7
R/B1_1_n	O	na	na	na	OA8	na	na	na	na	H7	H7
RE_0_n	M	8	8	8	C7	C7	D4	D4	D4	M6	M6
RE_1_n	O	na	na	na	D6	na	na	na	na	L6	L6
W/R_0_n	M	na	8	na	na	na	na	D4	na	na	M6
W/R_1_n	O	na	na	na	na	na	na	na	na	na	L6
CE0_n	M	9	9	9	na	A5	C6	C6	C6	na	na
CE1_n	O	10	10	10	na	C5	D6	D6	D6	na	na
CE2_n	O	14	14	14	na	A1	D7	D7	D7	na	na
CE3_n	O	15	15	15	na	OA0	E7	E7	E7	na	na
CE0_0_n	M	na	na	na	A5	na	na	na	na	K7	K7
CE0_1_n	O	na	na	na	C5	na	na	na	na	K6	K6
CE1_0_n	O	na	na	na	A1	na	na	na	na	K5	K5
CE1_1_n	O	na	na	na	OA0	na	na	na	na	J5	J5
Vcc	M	12 37	12 37	12 37	B6 M6	B6 M6	D3 G4	D3 G4	D3 G4	F2 F3 F4 F5 F6 F7 F8 F9	F2 F3 F4 F5 F6 F7 F8 F9
VccQ	M	34 39	34 39	34 39	N1 N7	N1 N7	H8 J6	H8 J6	H8 J6	H3 H8 L2 L4 L7 L9 P3	H3 H8 L2 L4 L7 L9 P3

Signal Name	M/O/R	TSOP / WSOP Async only x8	TSOP / WSOP Src Sync x8	TSOP / WSOP Async only x16	LGA Async only x8	LGA Async only x16	BGA-63 Async only x8	BGA-63 Src Sync x8	BGA-63 Async only x16	BGA-100 Async only x8	BGA-100 Src Sync x8
										P8	P8
Vss	M	13 36	13 36	13 36	B2 F6 L3	B2 F6 L3	C5 F7	C5 F7	C5 F7	G2 G3 G4 G5 G6 G7 G8 G9	G2 G3 G4 G5 G6 G7 G8 G9
VssQ	M	25 48	25 48	25 48	M2 OE8	M2 OE8	K8 K3	K8 K3	K8 K3	H2 H9 L3 L8 M4 M7 P2 P9	H2 H9 L3 L8 M4 M7 P2 P9
VREFQ_0	R	na	na	na	na	na	na	F5	na	na	H5
VREFQ_1	R	na	na	na	na	na	na	na	na	na	H4
VDDi	O	na	na	na	OB8	OB8	na	na	na	E9	E9
CLE_0	M	16	16	16	A3	A3	D5	D5	D5	M5	M5
CLE_1	O	na	na	na	C3	na	na	na	na	L5	L5
ALE_0	M	17	17	17	C1	C1	C4	C4	C4	K4	K4
ALE_1	O	na	na	na	D2	na	na	na	na	J4	J4
WE_0_n	M	18	18	18	E3	E3	C7	H7	C7	P7	P7
WE_1_n	O	na	na	na	E1	na	na	na	na	N7	N7
CLK_0_t	M	na	18	na	na	na	na	H7	na	na	P7
CLK_1_t	O	na	na	na	na	na	na	na	na	na	N7
CLK_0_c	R	na	na	na	na	na	na	H6	na	na	P6
CLK_1_c	R	na	na	na	na	na	na	na	na	na	N6
WP_0_n	M	19	19	19	F2	F2	C3	C3	C3	E5	E5
WP_1_n	O	na	na	na	G5	na	na	na	na	D5	D5

Signal Name	M/O/R	TSOP / WSOP Async only x8	TSOP / WSOP Src Sync x8	TSOP / WSOP Async only x16	LGA Async only x8	LGA Async only x16	BGA-63 Async only x8	BGA-63 Src Sync x8	BGA-63 Async only x16	BGA-100 Async only x8	BGA-100 Src Sync x8
IO0_0 / DQ0_0	M	29	29	29	G3	G3	H4	H4	H4	K2	K2
IO1_0 / DQ1_0	M	30	30	30	H2	H2	J4	J4	J4	N2	N2
IO2_0 / DQ2_0	M	31	31	31	J3	J3	K4	K4	K4	K3	K3
IO3_0 / DQ3_0	M	32	32	32	K2	K2	K5	K5	K5	N3	N3
IO4_0 / DQ4_0	M	41	41	41	L5	L5	K6	K6	K6	N8	N8
IO5_0 / DQ5_0	M	42	42	42	K6	K6	J7	J7	J7	K8	K8
IO6_0 / DQ6_0	M	43	43	43	J5	J5	K7	K7	K7	N9	N9
IO7_0 / DQ7_0	M	44	44	44	H6	H6	J8	J8	J8	K9	K9
IO8	M	na	na	26	na	G1	na	na	H3	na	na
IO9	M	na	na	27	na	J1	na	na	J3	na	na
IO10	M	na	na	28	na	L1	na	na	H5	na	na
IO11	M	na	na	33	na	N3	na	na	J5	na	na
IO12	M	na	na	40	na	N5	na	na	H6	na	na
IO13	M	na	na	45	na	L7	na	na	G6	na	na
IO14	M	na	na	46	na	J7	na	na	H7	na	na
IO15	M	na	na	47	na	G7	na	na	G7	na	na
IO0_1 / DQ0_1	O	na	na	na	G1	na	na	na	na	J2	J2
IO1_1 / DQ1_1	O	na	na	na	J1	na	na	na	na	M2	M2
IO2_1 / DQ2_1	O	na	na	na	L1	na	na	na	na	J3	J3
IO3_1 / DQ3_1	O	na	na	na	N3	na	na	na	na	M3	M3
IO4_1 / DQ4_1	O	na	na	na	N5	na	na	na	na	M8	M8
IO5_1 / DQ5_1	O	na	na	na	L7	na	na	na	na	J8	J8
IO6_1 / DQ6_1	O	na	na	na	J7	na	na	na	na	M9	M9
IO7_1 / DQ7_1	O	na	na	na	G7	na	na	na	na	J9	J9
DQS_0_t	M	na	35	na	na	na	na	J5	na	na	P5
DQS_1_t	O	na	na	na	na	na	na	na	na	na	N5
DQS_0_c	R	na	na	na	na	na	na	H5	na	na	P4
DQS_1_c	R	na	na	na	na	na	na	na	na	na	N4
VSP0_0	O	38	38	38	na	na	G5	G5	G5	E7	E7
VSP1_0	O	35	na	35	na	na	G8	G8	G8	E6	E6
VSP2_0	O	20	20	20	na	na	G3	G3	G3	E4	E4
VSP0_1	O	na	na	na	na	na	na	na	na	D7	D7
VSP1_1	O	na	na	na	na	na	na	na	na	D6	D6
VSP2_1	O	na	na	na	na	na	na	na	na	D4	D4

**Table 3 Signal mappings: TSOP, LGA, BGA packages**

## 2.6. CE\_n Signal Requirements

If one or more LUNs are active and the host sets CE\_n to one, then those operations continue executing until completion at which point the target enters standby. After the CE\_n signal is transitioned to one, the host may drive a different CE\_n signal to zero and begin operations on another target. Note that if using a dual x8 package (e.g. BGA-100), then operations may execute in parallel on two different CE\_n's if they are connected to different 8-bit data buses.

When SR[6] for a particular LUN is cleared to zero and the CE\_n signal for the corresponding target is cleared to zero, the host may only issue the Reset, Synchronous Reset, Read Status, or Read Status Enhanced commands to that LUN.

### 2.6.1. Source Synchronous Data Interface Requirements

When using the source synchronous data interface, the following requirements shall be met if the device does not support CLK being stopped during data input:

1. CLK shall only stop or start when CE\_n is high.

When using the source synchronous data interface, the following requirements shall be met if the device supports CLK being stopped during data input:

1. CLK shall only stop or start when either:
  - a. CE\_n is high, or
  - b. CE\_n is low and the bus state is data input

When using the source synchronous data interface, the following requirements shall always be met:

1. CLK shall only change frequency when CE\_n is high.
2. When CE\_n is low, CLK shall maintain the same frequency.
3. CE\_n shall only transition from one to zero when the CLK is stable and has a valid period based on the timing mode selected.
4. The interface shall be in an idle state (see section 4.1.2) when CE\_n changes value.  
CE\_n shall only transition when the following are true:
  - a. ALE and CLE are both cleared to zero, and
  - b. There is no data transfer on the DQ/DQS signals during the current clock period.

## 2.7. Absolute Maximum DC Ratings

Stresses greater than those listed in Table 4 may cause permanent damage to the device. This is a stress rating only. Operation beyond the recommended operating conditions specified in Table 5 and the DC and operating characteristics listed in Table 9 and Table 10 is not recommended. Except as defined in section 2.9, extended exposure beyond these conditions may affect device reliability.

Table 4 defines the voltage on any pin relative to Vss and/or VssQ for devices based on their Vcc and VccQ typical voltages.

Parameter	Symbol	Rating	Units
<i>V<sub>CC</sub> = 3.3V and V<sub>CCQ</sub> = 3.3V nominal</i>			
V <sub>CC</sub> Supply Voltage	V <sub>CC</sub>	-0.6 to +4.6	V
Voltage Input	V <sub>IN</sub>	-0.6 to +4.6	
V <sub>CCQ</sub> Supply Voltage	V <sub>CCQ</sub>	-0.6 to +4.6	
<i>V<sub>CC</sub> = 3.3V and V<sub>CCQ</sub> = 1.8V nominal</i>			
V <sub>CC</sub> Supply Voltage	V <sub>CC</sub>	-0.6 to +4.6	V
Voltage Input	V <sub>IN</sub>	-0.2 to +2.4	
V <sub>CCQ</sub> Supply Voltage	V <sub>CCQ</sub>	-0.2 to +2.4	
<i>V<sub>CC</sub> = 1.8V and V<sub>CCQ</sub> = 1.8V nominal</i>			
V <sub>CC</sub> Supply Voltage	V <sub>CC</sub>	-0.2 to +2.4	V
Voltage Input	V <sub>IN</sub>	-0.2 to +2.4	
V <sub>CCQ</sub> Supply Voltage	V <sub>CCQ</sub>	-0.2 to +2.4	

**Table 4 Absolute maximum DC ratings**

## 2.8. Recommended DC Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Supply voltage for 3.3V devices	V <sub>CC</sub>	2.7	3.3	3.6	V
Supply voltage for 1.8V devices	V <sub>CC</sub>	1.7	1.8	1.95	V
Supply voltage for 3.3V I/O signaling	V <sub>CCQ</sub> (V <sub>CCQH</sub> )	2.7	3.3	3.6	V
Supply voltage for 1.8V I/O signaling	V <sub>CCQ</sub> (V <sub>CCQL</sub> )	1.7	1.8	1.95	V
Ground voltage supply	V <sub>SS</sub>	0	0	0	V
Ground voltage supply for I/O signaling	V <sub>SSQ</sub>	0	0	0	V

**Table 5 Recommended DC operating conditions**

### 2.8.1. I/O Power (V<sub>CCQ</sub>) and I/O Ground (V<sub>SSQ</sub>)

V<sub>CCQ</sub> and V<sub>CC</sub> may be distinct and unique voltages. V<sub>CCQ</sub> shall be less than or equal to V<sub>CC</sub>, including during power-on ramp. The device shall support one of the following V<sub>CCQ</sub>/V<sub>CC</sub> combinations:

- V<sub>CC</sub> = 3.3V, V<sub>CCQ</sub> = 3.3V
- V<sub>CC</sub> = 3.3V, V<sub>CCQ</sub> = 1.8V
- V<sub>CC</sub> = 1.8V, V<sub>CCQ</sub> = 1.8V

All parameters, timing modes, and other characteristics are relative to the supported voltage combination.

If a device has the same V<sub>CC</sub> and V<sub>CCQ</sub> voltage levels, then V<sub>CCQ</sub> and V<sub>SSQ</sub> are not required to be connected internal to the device. Specifically, the device may use V<sub>CC</sub> and V<sub>SS</sub> exclusively as the I/O and core voltage supply.

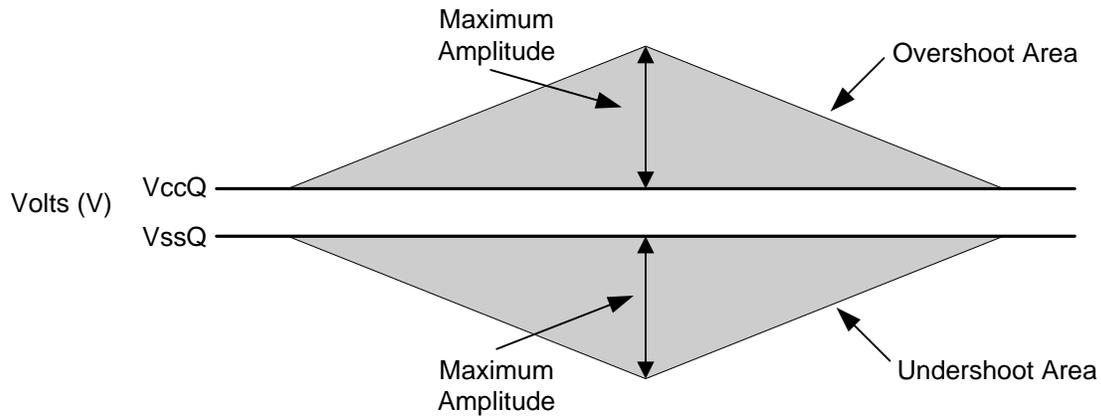
## 2.9. AC Overshoot/Undershoot Requirements

The device may have AC overshoot or undershoot from VccQ and VssQ levels. Table 6 defines the maximum values that the AC overshoot or undershoot may attain. These values apply for both 3.3V and 1.8V VccQ levels.

Parameter	Maximum Value				Unit
	$\leq 100$ MT/s	$> 100$ MT/s and $\leq 133$ MT/s	$> 133$ MT/s and $\leq 166$ MT/s	$> 166$ MT/s and $\leq 200$ MT/s	
Peak amplitude allowed for overshoot area	1	1	1	1	V
Peak amplitude allowed for undershoot area	1	1	1	1	V
Maximum Overshoot area above VccQ	3	2.25	1.8	1.5	V-ns
Maximum Undershoot area below VssQ	3	2.25	1.8	1.5	V-ns

**Table 6 AC Overshoot/Undershoot Maximum Values**

Figure 14 displays pictorially the parameters described in Table 6.



**Figure 14 Overshoot/Undershoot Diagram**

## 2.10. DC and Operating Characteristics

All operating current ratings in this section are specified per active logical unit (LUN). A LUN is active when there is a command outstanding to it. All other current ratings in this section are specified per LUN (regardless of whether it is active).

For high performance applications it may be desirable to draw increased current for ICC1-ICC4. For these applications, the device may draw up to 100 mA per active LUN in both 3.3V and 1.8V devices. Increased current may be used to improve sustained write performance.

All ICC measurements are measured with each Vcc pin decoupled with a 0.1  $\mu$ F capacitor. The ICC definition assumes outputs change between one and zero every other data cycle (once per CLK period, every other DQS transition) for data signals. The test conditions and measurement methodology for the ICC values is defined in Appendix D.

Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Array read current	ICC1	Refer to Appendix D	-	-	50	mA
Array program current	ICC2		-	-	50	mA
Array erase current	ICC3		-	-	50	mA
I/O burst read current	ICC4 <sub>R</sub> <sup>4</sup>		-	-	50	mA
I/O burst write current	ICC4 <sub>W</sub>		-	-	50	mA
Bus idle current	ICC5		-	-	10	mA
Standby current, CMOS	ISB	CE_n=VccQ-0.2V, WP_n=0V/VccQ	-	-	50	$\mu$ A
Staggered power-up current	IST <sup>1</sup>	CE_n=VccQ-0.2V tRise = 1 ms cLine = 0.1 $\mu$ F	-	-	10	mA

NOTE:

1. Refer to Appendix C for an exception to the IST current requirement.
2. ICC1, ICC2, and ICC3 as listed in this table are active current values. For details on how to calculate the active current from the measured values, refer to Appendix D.
3. During cache operations, increased ICC current is allowed while data is being transferred on the bus and an array operation is ongoing. For a cached read this value is ICC1 + ICC4<sub>R</sub>; for a cached write this value is ICC2(active) + ICC4<sub>W</sub>.
4. For ICC4<sub>R</sub> the test conditions in Appendix D specify IOOUT = 0 mA and requires static outputs with no output switching. When outputs are not static, additional VccQ current will be drawn that is highly dependent on system configuration. IccQ may be calculated for each output pin assuming 50% data switching as  $(I_{ccQ} = 0.5 * C_L * V_{ccQ} * \text{frequency})$ , where C<sub>L</sub> is the capacitive load.

**Table 7 DC and Operating Conditions for raw NAND, measured on Vcc rail**

Parameter	Symbol	Test Conditions		Min	Typ	Max	Units
Array read current	ICC1	Refer to Appendix D	Per LUN	-	-	85	mA
	ICCQ1		Controller	-	-	200	
	ICCQ1		Per LUN	-	-	15	
Array program current	ICC2		Per LUN	-	-	85	mA
	ICCQ2		Controller	-	-	75	
	ICCQ2		Per LUN	-	-	15	
Array erase current	ICC3		Per LUN	-	-	85	mA
	ICCQ3		Controller	-	-	75	
	ICCQ3		Per LUN	-	-	15	
I/O burst read current	ICC4 <sub>R</sub> <sup>4</sup>		LUN	-	-	50	mA
	ICCQ4 <sub>R</sub> <sup>4</sup>		Controller	-	-	50	
I/O burst write current	ICC4 <sub>W</sub>		LUN	-	-	50	mA
	ICCQ4 <sub>W</sub>		Controller	-	-	50	
Bus idle current	ICC5		Per LUN	-	-	45	mA
	ICCQ5		Controller	-	-	10	
	ICCQ5	Per LUN	-	-	15		
Standby current, CMOS	ISB	CE_n=VccQ-0.2V, WP_n=0V/VccQ	Per LUN	-	-	50	µA
	ISBQ		Controller	-	-	1000	
Staggered power-up current	IST <sup>1</sup>	CE_n=VccQ-0.2V tRise = 1 ms cLine = 0.1 µF	Per LUN	-	-	10	mA
	ISTQ		Controller	-	-	20	

**NOTE:**

1. Refer to Appendix C for an exception to the IST current requirement.
2. ICC1, ICC2, and ICC3 as listed in this table are active current values. For details on how to calculate the active current from the measured values, refer to Appendix D.
3. During cache operations, increased ICC current is allowed while data is being transferred on the bus and an array operation is ongoing. For a cached read this value is ICC1 + ICC4<sub>R</sub> on Vcc and ICCQ1 on VccQ; for a cached write this value is ICC2(active) + ICC4<sub>W</sub> on Vcc, and ICCQ2 on VccQ.
4. For ICC4<sub>R</sub> the test conditions in Appendix D specify IO<sub>UT</sub> = 0 mA and requires static outputs with no output switching. When outputs are not static, additional VccQ current will be drawn that is highly dependent on system configuration. IccQ may be calculated for each output pin assuming 50% data switching as (IccQ = 0.5 \* C<sub>L</sub> \* VccQ \* frequency), where C<sub>L</sub> is the capacitive load.

**Table 8 DC and Operating Conditions for EZ NAND, measured on Vcc or VccQ rail**

The maximum leakage current requirements (ILI and ILO) in Table 9 and Table 10 are tested across the entire allowed VccQ range, specified in Table 5.

DC signal specifications apply to the following signals and only when using the source synchronous data interface: CLK, DQ[7:0], DQS, ALE, CLE, and W/R\_n. For all signals in asynchronous and all other signals in source synchronous, the AC signal specification shall be met. For signals where DC signal specifications apply, the transition times are measured between VIL (DC) and VIH (AC) for rising input signals and between VIH (DC) and VIL (AC) for falling input signals. The receiver will effectively switch as a result of the signal crossing the AC input level and remain in that state as long as the signal does not ring back above (below) the DC input LOW (HIGH) level.

The parameters in Table 9 and Table 10 apply to power-on default values in the device. If I/O drive strength settings or other device settings are changed, these values may be modified. The output characteristics for a device that supports driver strength settings (as indicated in the parameter page) are specified in the impedance tables (Table 28 and Table 29).

Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Standby current, CMOS	ISBQ	CE_n=VccQ-0.2V, WP_n=0V/VccQ	-	-	25	μA
Input leakage current	ILI	VIN=0V to VccQ	-	-	+10	μA
Output leakage current	ILO	VOUT=0V to VccQ	-	-	+10	μA
DC Input high voltage	VIH (DC)	-	VccQ * 0.7	-	VccQ + 0.3	V
AC Input high voltage	VIH (AC)	-	VccQ * 0.8	-	VccQ + 0.3	V
DC Input low voltage	VIL (DC)	-	-0.3	-	VccQ * 0.3	V
AC Input low voltage	VIL (AC)	-	-0.3	-	VccQ * 0.2	V
Output high voltage <sup>1</sup>	VOH	IOH=-400 μA	VccQ * 0.67	-	-	V
Output low voltage <sup>1</sup>	VOL	IOL=2.1 mA	-	-	0.4	V
Output low current (R/B_n)	IOL(R/B_n)	VOL=0.4 V	8	10	-	mA
NOTE:						
1. VOH and VOL defined in this table shall only apply to devices in asynchronous mode that do not support driver strength settings. If driver strength settings are supported then Table 28 shall be used to derive the output driver impedance values.						

**Table 9 DC and Operating Conditions for VccQ of 3.3V, measured on VccQ rail**

Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Standby current, CMOS	ISBQ	CE_n=VccQ-0.2V, WP_n=0V/VccQ	-	-	25	μA
Input leakage current	ILI	VIN=0V to VccQ	-	-	+10	μA
Output leakage current	ILO	VOUT=0V to VccQ	-	-	+10	μA
DC Input high voltage	VIH (DC)	-	VccQ * 0.7	-	VccQ+0.3	V
AC Input high voltage	VIH (AC)	-	VccQ * 0.8	-	VccQ+0.3	V
DC Input low voltage	VIL (DC)	-	-0.3	-	VccQ * 0.3	V
AC Input low voltage	VIL (AC)	-	-0.3	-	VccQ * 0.2	V
Output high voltage <sup>1</sup>	VOH	IOH=-100 μA	VccQ - 0.1	-	-	V
Output low voltage <sup>1</sup>	VOL	IOL=100 μA	-	-	0.1	V
Output low current (R/B_n)	IOL(R/B_n)	VOL=0.2 V	3	4	-	mA
NOTE:						
1. VOH and VOL defined in this table shall only apply to devices in asynchronous mode that do not support driver strength settings. If driver strength settings are supported then Table 29 shall be used to derive the output driver impedance values.						

**Table 10 DC and Operating Conditions for VccQ of 1.8V, measured on VccQ rail**

## 2.11. Calculating Pin Capacitance

To calculate the pin capacitance for all loads on the I/O bus, the host should utilize the reported pin capacitance per target in Read Parameter Page (refer to section 5.7). The maximum

capacitance may be used, or the typical capacitance if provided by the device may be used. The algorithm to use is:

```
PinCapacitance = 0;
for (target = 0; target < TotalTargets; target++)
    PinCapacitance += GetCapacitanceFromRPP(target);
```

This methodology will calculate an accurate maximum or typical pin capacitance, respectively, accounting for all targets present.

## 2.12. Staggered Power-up

Subsystems that support multiple Flash devices may experience power system design issues related to the current load presented during the power-on condition. To limit the current load presented to the host at power-on, all devices shall support power-up in a low-power condition.

Until a Reset (FFh) command is received by the target after power-on, the target shall not draw more than IST of current per LUN and ISTQ (if present for devices that support EZ NAND). For example, a target that contains 4 LUNs may draw up to 40 mA of current until a Reset (FFh) command is received after power-on.

This value is measured with a nominal rise time (tRise) of 1 millisecond and a line capacitance (cLine) of 0.1  $\mu$ F. The measurement shall be taken with 1 millisecond averaging intervals and shall begin after Vcc reaches Vcc\_min and VccQ reaches VccQ\_min.

## 2.13. Power Cycle Requirements

As part of a power cycle, the host shall hold both the Vcc and VccQ voltage levels below 100 mV for a minimum time of 100 ns. If these requirements are not met as part of a power cycle operation, the device may enter an indeterminate state.

## 2.14. Independent Data Buses

There may be two independent 8-bit data buses in some ONFI packages (i.e. the LGA and the 100-ball BGA package). For packages that support either two independent data buses or a single data bus (e.g. LGA-52) then CE0\_n and CE2\_n shall use the same pins as the first data bus CE\_n pins (marked as CE0\_0\_n and CE1\_0\_n) and CE1\_n and CE3\_n shall use the same pins as the second data bus CE\_n pins (marked as CE0\_1\_n and CE1\_1\_n). Note that CE0\_n, CE1\_n, CE2\_n, and CE3\_n may all use the first data bus and the first set of control signals (RE0\_n, CLE0\_n, ALE0\_n, WE0\_n, and WPO\_n) if the device does not support independent data buses.

Implementations may tie the data lines and control signals (RE\_n, CLE, ALE, WE\_n, WP\_n, and DQS) together for the two independent 8-bit data buses externally to the device.

## 2.15. Bus Width Requirements

All targets per device shall use the same data bus width. All targets shall either have an 8-bit bus width or a 16-bit bus width. Note that devices that support the source synchronous interface shall have an 8-bit bus width.

When the host supports a 16-bit bus width, only data is transferred at the 16-bit width. All address and command line transfers shall use only the lower 8-bits of the data bus. During

command transfers, the host may place any value on the upper 8-bits of the data bus. During address transfers, the host shall set the upper 8-bits of the data bus to 00h.

## 2.16. Ready/Busy (R/B\_n) Requirements

### 2.16.1. Power-On Requirements

Once  $V_{CC}$  and  $V_{CCQ}$  reach the  $V_{CC}$  minimum and  $V_{CCQ}$  minimum values, respectively, listed in Table 5 and power is stable, the R/B\_n signal shall be valid after RB\_valid\_Vcc and shall be set to one (Ready) within RB\_device\_ready, as listed in Table 11. R/B\_n is undefined until 50  $\mu$ s has elapsed after  $V_{CC}$  has started to ramp. The R/B\_n signal is not valid until both of these conditions are met.

Parameter	Raw NAND	EZ NAND
RB_valid_Vcc	10 $\mu$ s	250 $\mu$ s
RB_device_ready	1 ms	2 ms

Table 11 R/B\_n Power-on Requirements

During power-on,  $V_{CCQ}$  shall be less than or equal to  $V_{CC}$  at all times. Figure 15 shows  $V_{CCQ}$  ramping after  $V_{CC}$ , however, they may ramp at the same time.

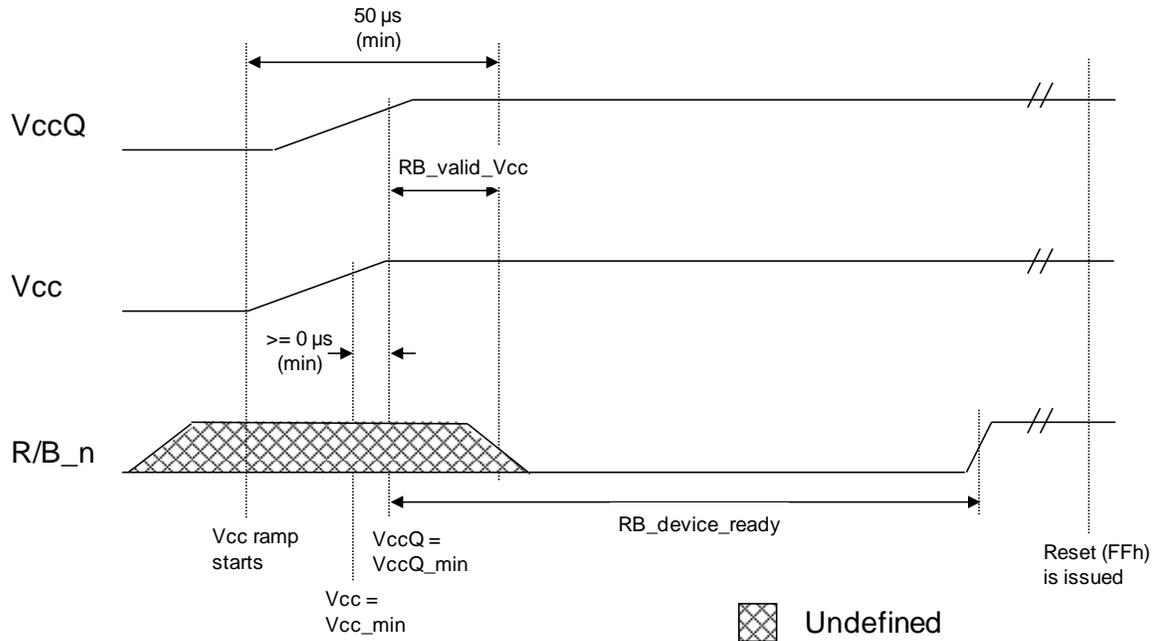


Figure 15 R/B\_n Power-On Behavior

Ready/Busy is implemented as an open drain circuit, thus a pull-up resistor shall be used for termination. The combination of the pull-up resistor and the capacitive loading of the R/B\_n circuit determines the rise time of R/B\_n.

### 2.16.2. R/B\_n and SR[6] Relationship

R/B\_n shall reflect the logical AND of the SR[6] (Status Register bit 6) values for all LUNs on the corresponding target. For example, R/B3\_n is the logical AND of the SR[6] values for all LUNs on CE3\_n. Thus, R/B\_n reflects whether any LUN is busy on a particular target.

### 2.17. Write Protect

When cleared to zero, the WP\_n signal disables Flash array program and erase operations. This signal shall only be transitioned while there are no commands executing on the device. After modifying the value of WP\_n, the host shall not issue a new command to the device for at least t<sub>WW</sub> delay time.

Figure 16 describes the t<sub>WW</sub> timing requirement, shown with the start of a Program command. The transition of the WP\_n signal is asynchronous and unrelated to any CLK transition in the source synchronous data interface. The bus shall be idle for t<sub>WW</sub> time after WP\_n transitions from zero to one before a new command is issued by the host, including Program. The bus shall be idle for t<sub>WW</sub> time after WP\_n transitions from one to zero before a new command is issued by the host.

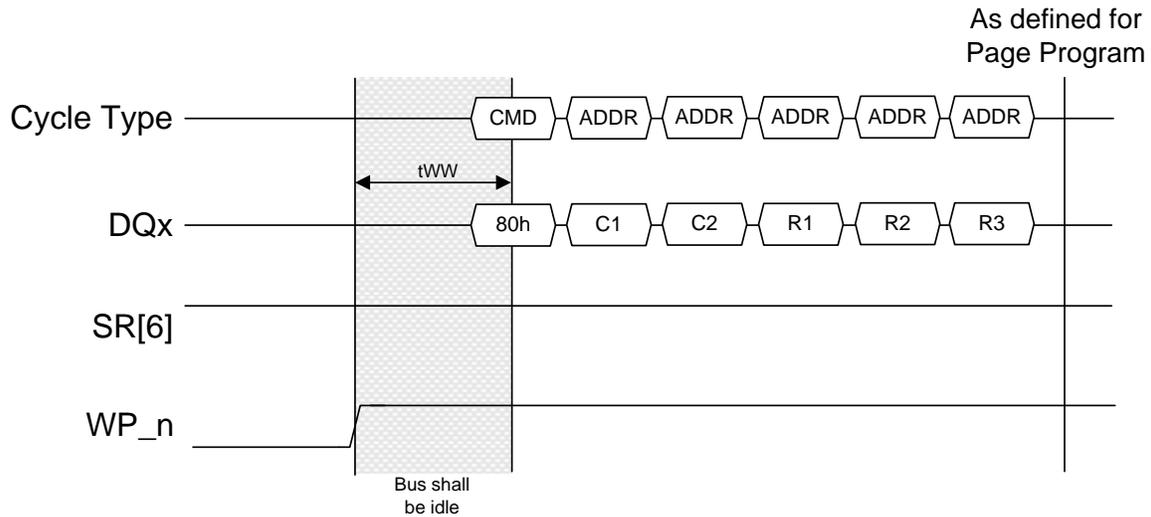
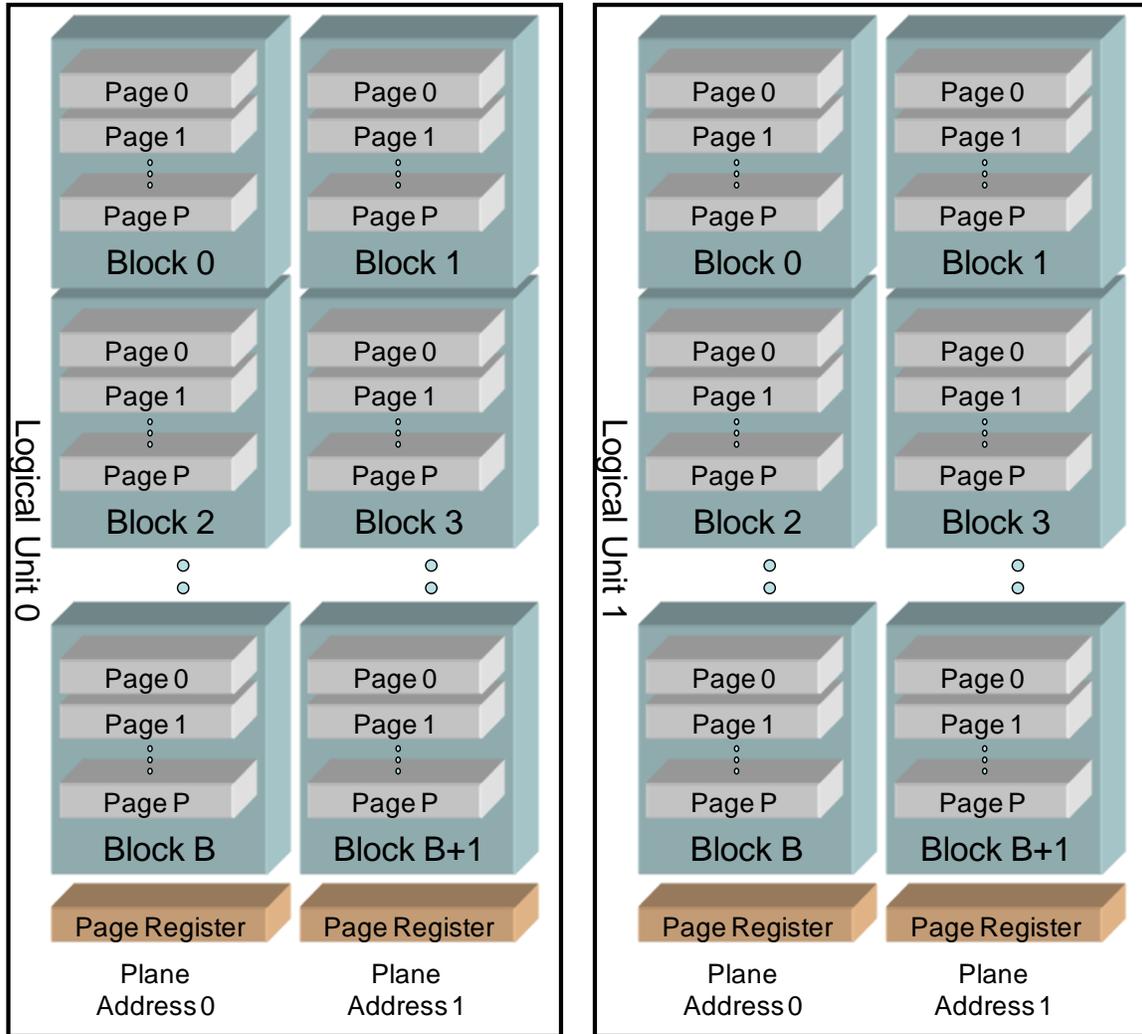


Figure 16 Write Protect timing requirements, example

### 3. Memory Organization

Figure 17 shows an example of a Target memory organization. In this case, there are two logical units where each logical unit has two planes.



**Figure 17 Target memory organization**

A device contains one or more targets. A target is controlled by one CE\_n signal. A target is organized into one or more logical units (LUNs).

A logical unit (LUN) is the minimum unit that can independently execute commands and report status. Specifically, separate LUNs may operate on arbitrary command sequences in parallel. For example, it is permissible to start a Page Program operation on LUN 0 and then prior to the operation's completion to start a Read command on LUN 1. See multiple LUN operation restrictions in section 3.1.3. A LUN contains at least one page register and a Flash array. The number of page registers is dependent on the number of multi-plane operations supported for that LUN. The Flash array contains a number of blocks.

A block is the smallest erasable unit of data within the Flash array of a LUN. There is no restriction on the number of blocks within the LUN. A block contains a number of pages.

A page is the smallest addressable unit for read and program operations. A page consists of a number of bytes or words. The number of user data bytes per page, not including the spare data area, shall be a power of two. The number of pages per block shall be a multiple of 32.

Each LUN shall have at least one page register. A page register is used for the temporary storage of data before it is moved to a page within the Flash array or after it is moved from a page within the Flash array. If EZ NAND is supported then a buffer exists in the EZ NAND controller that provides for temporary storage of data that may then be transferred to or from the page register within each LUN.

The byte or word location within the page register is referred to as the column.

There are two mechanisms to achieve parallelism within this architecture. There may be multiple commands outstanding to different LUNs at the same time. To get further parallelism within a LUN, multi-plane operations may be used to execute additional dependent operations in parallel.

### 3.1. Addressing

There are two address types used: the column address and the row address. The column address is used to access bytes or words within a page, i.e. the column address is the byte/word offset into the page. The least significant bit of the column address shall always be zero in the source synchronous data interface, i.e. an even number of bytes is always transferred. The row address is used to address pages, blocks, and LUNs.

When both the column and row addresses are required to be issued, the column address is always issued first in one or more 8-bit address cycles. The row addresses follow in one or more 8-bit address cycles. There are some functions that may require only row addresses, like Block Erase. In this case the column addresses are not issued.

For both column and row addresses the first address cycle always contains the least significant address bits and the last address cycle always contains the most significant address bits. If there are bits in the most significant cycles of the column and row addresses that are not used then they are required to be cleared to zero.

The row address structure is shown in Figure 18 with the least significant row address bit to the right and the most significant row address bit to the left.



Figure 18 Row Address Layout

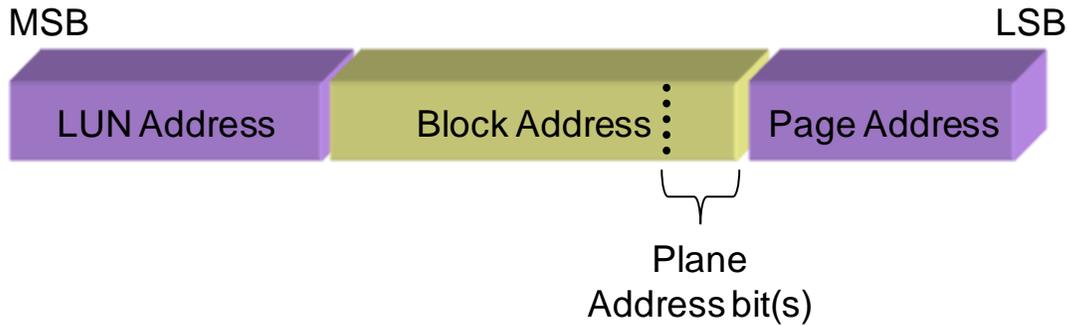
The number of blocks and number of pages per block is not required to be a power of two. In the case where one of these values is not a power of two, the corresponding address shall be rounded to an integral number of bits such that it addresses a range up to the subsequent power of two value. The host shall not access upper addresses in a range that is shown as not supported. For example, if the number of pages per block is 96, then the page address shall be rounded to 7 bits such that it can address pages in the range of 0 to 127. In this case, the host shall not access pages in the range from 96 to 127 as these pages are not supported.

The page address always uses the least significant row address bits. The block address uses the middle row address bits and the LUN address uses the most significant row address bit(s).

### 3.1.1. Multi-plane Addressing

The multi-plane address comprises the lowest order bits of the block address as shown in Figure 19. The following restrictions apply to the multi-plane address when executing a multi-plane command sequence on a particular LUN:

- The plane address bit(s) shall be distinct from any other multi-plane operation in the multi-plane command sequence.
- The page address shall be the same as any other multi-plane operations in the multi-plane command sequence.



**Figure 19** Plane Address Location

#### 3.1.1.1. Multi-plane Block Address Restrictions

The device may indicate multi-plane block address restrictions. The specific cases are:

- No restriction: All block address bits may be different between two plane addresses.
- Full restriction: All block address bits (other than the plane address bits) shall be the same between two plane addresses.
- Lower bit XNOR restriction: If the XNOR of the lowest plane address bits (bit 0) is one between two plane addresses, then there is a full restriction between these two plane addresses. If the XNOR of the lower plane address bits is zero between two plane addresses, then there is no restriction between these two plane addresses.

Table 12 illustrates the three types of restrictions for a four plane operation.

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 12** Four plane address restriction

Table 13 describes whether there is a lower bit XNOR restriction between two plane addresses A and B, based on their plane address bits for a 4 plane implementation. If there is a lower bit XNOR restriction, then the block addresses (other than the plane address bits) shall be the same between multi-plane addresses A and B.

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 13 4-way lower bit XNOR restriction**

### 3.1.2. Logical Unit Selection

Logical units within one target share a single data bus with the host. The host shall ensure that only one LUN is selected for data output to the host at any particular point in time to avoid bus contention.

The host selects a LUN for future data output by issuing a Read Status Enhanced command to that LUN. The Read Status Enhanced command shall deselect the output path for all LUNs that are not addressed by the command. The page register selected for output within the LUN is determined by the previous Read (Cache) commands issued, and is not impacted by Read Status Enhanced.

### 3.1.3. Multiple LUN Operation Restrictions

LUNs are independent entities. A multiple LUN operation is one in which two or more LUNs are simultaneously processing commands. This implies that R/B\_n is cleared to zero when the subsequent LUN operation is issued.

When a Page Program command (80h) is issued on any LUN that is not preceded by an 11h command, all idle LUNs may clear their page registers if the program page register clear enhancement is not supported or enabled. Thus, the host should not begin a Page Program command on a LUN while a Read Page operation is either ongoing or has completed but the data has not been read from another LUN, as the contents of the page register for the Read operation are lost. A Read Page can be issued to one LUN while a Page Program is ongoing within a second LUN without any restriction. If the program page register clear enhancement is enabled, this restriction does not apply.

When issuing a Page Program command (80h), the host should not select another LUN until after all data has been input and a 10h or 15h command has been issued. In the case of multi-plane operations, all data input for all multi-plane addresses should be completed prior to selecting another LUN.

When issuing Reads to multiple LUNs, the host shall take steps to avoid issues due to column address corruption. Specifically, if the column addresses in Reads issued to multiple LUNs are different, then the host shall issue a Change Read Column before starting to read out data from a newly selected LUN. If the column addresses are the same, then no Change Read Column is necessary.

If a multiple LUN operation has been issued, then the next status command issued shall be Read Status Enhanced. Read Status Enhanced causes LUNs that are not selected to turn off their

output buffers. This ensures that only the LUN selected by the Read Status Enhanced command responds to a subsequent data output cycle. After a Read Status Enhanced command has been completed, the Read Status command may be used until the next multiple LUN operation is issued.

When the host has issued Read Page commands to multiple LUNs at the same time, the host shall issue Read Status Enhanced before reading data from either LUN. This ensures that only the LUN selected by the Read Status Enhanced command responds to a data output cycle after being put in data output mode with a 00h command, and thus avoiding bus contention. If the host issues a Change Read Column (Enhanced) for any LUN that Read Page commands are outstanding for, then the host shall issue a Change Read Column (Enhanced) prior to transferring data from any subsequent LUN that is part of the multiple LUN read sequence. An example sequence is shown below:

- 1) Read Page command issued to LUN 0
- 2) Read Page command issued to LUN 1
- 3) Read Status Enhanced selects LUN 0
- 4) Change Read Column (Enhanced) issued to LUN 0 (if needed by host)
- 5) Data transferred from LUN 0
- 6) Read Status Enhanced selects LUN 1
- 7) Change Read Column (Enhanced) issued to LUN 1 (required if action 4 taken)
- 8) Data transferred from LUN 1

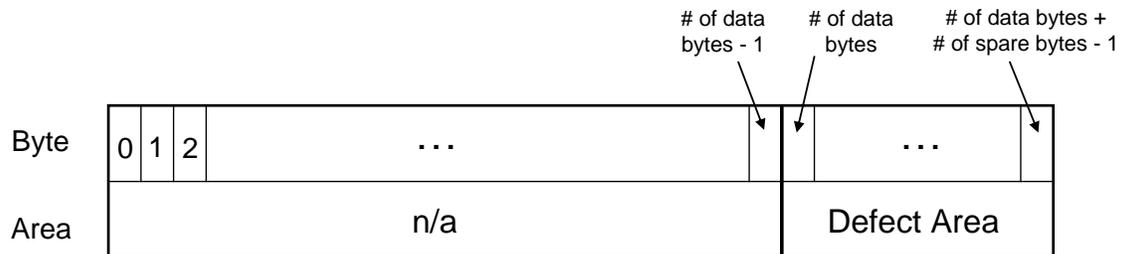
The host may substitute Change Read Column Enhanced for the Read Status Enhanced / Change Read Column sequence if all LUNs are not busy.

### 3.2. Factory Defect Mapping

The Flash array is not presumed to be pristine, and a number of defects may be present that renders some blocks unusable. Block granularity is used for mapping factory defects since those defects may compromise the block erase capability.

#### 3.2.1. Device Requirements

If a block is defective and 8-bit data access is used, the manufacturer shall mark the block as defective by setting the first byte in the defect area, as shown in Figure 20, of the first or last page of the defective block to a value of 00h. If a block is defective and 16-bit data access is used, the manufacturer shall mark the block as defective by setting the first word in the defect area of the first or last page of the defective block to a value of 0000h.



**Figure 20** Area marked in factory defect mapping

### 3.2.2. Host Requirements

The host shall not erase or program blocks marked as defective by the manufacturer, and any attempt to do so yields indeterminate results.

Figure 21 outlines the algorithm to scan for factory mapped defects. This algorithm should be performed by the host to create the initial bad block table prior to performing any erase or programming operations on the target. The initial state of all pages in non-defective blocks is FFh (or FFFFh for 16-bit access) for all page addresses, although some bit errors may be present if they are correctable via the required ECC reported to the host. A defective block is indicated by a byte value equal to 00h for 8-bit access or a word value equal to 0000h for 16-bit access being present at the first byte/word location in the defect area of either the first page or last page of the block. The host shall check the first byte/word of the defect area of both the first and last past page of each block to verify the block is valid prior to any erase or program operations on that block.

**NOTE:** Over the lifetime use of a NAND device, the defect area of defective blocks may encounter read disturbs that cause values to change. The manufacturer defect markings may change value over the lifetime of the device, and are expected to be read by the host and used to create a bad block table during initial use of the part.

```
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 21 Factory defect scanning algorithm

### 3.3. Extended ECC Information Reporting

The device may report extended ECC information in the extended parameter page. The required ECC correctability is closely related to other device parameters, like the number of valid blocks and the number of program/erase cycles supported. Extended ECC information allows the device to specify multiple valid methods for using the device.

Table 14 defines the extended ECC information block.

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 14 Extended ECC Information Block Definition**

The definition of each field follows in the subsequent sections.

### 3.3.1. Byte 0: Number of bits ECC correctability

This field indicates the number of bits that the host should be able to correct per codeword. The codeword size is reported in byte 1. With this specified amount of error correction by the host, the target shall achieve the block endurance specified in bytes 4-5. When the specified amount of error correction is applied by the host and the block endurance is followed, then the maximum number of bad blocks specified in bytes 2-3 shall not be exceeded by the device. All used bytes in the page shall be protected by host controller ECC including the spare bytes if the ECC requirement reported in byte 0 has a value greater than zero.

When this value is cleared to zero, the target shall return valid data if the ECC Information Block is valid (the Codeword size is non-zero).

### 3.3.2. Byte 1: Codeword size

The number of bits of ECC correctability specified in byte 0 is based on a particular ECC codeword size. The ECC codeword size is specified in this field as a power of two. The minimum value that shall be reported is 512 bytes (a value of 9).

If a value of 0 is reported then this ECC Information Block is invalid and should not be used.

### 3.3.3. Byte 2-3: Bad blocks maximum per LUN

This field contains the maximum number of blocks that may be defective at manufacture and over the life of the device per LUN. The maximum rating assumes that the host is following the block endurance requirements and the ECC requirements reported in this extended ECC information block.

### 3.3.4. Byte 4-5: Block endurance

This field indicates the maximum number of program/erase cycles per addressable page/block. This value assumes that the host is using the ECC correctability reported in byte 0.

The block endurance is reported in terms of a value and a multiplier according to the following equation: value x  $10^{\text{multiplier}}$ . Byte 4 comprises the value. Byte 5 comprises the multiplier. For example, a block endurance of 75,000 cycles would be reported as a value of 75 and a multiplier of 3 ( $75 \times 10^3$ ). The value field shall be the smallest possible; for example 100,000 shall be reported as a value of 1 and a multiplier of 5 ( $1 \times 10^5$ ).

## 3.4. Discovery and Initialization

### 3.4.1. CE<sub>n</sub> Discovery

There may be up to four chip enable (CE<sub>n</sub>) signals on a package, one for each separately addressable target. To determine the targets that are connected, the procedure outlined in this section shall be followed for each distinct CE<sub>n</sub> signal. CE<sub>n</sub> signals shall be used sequentially on the device; CE<sub>0</sub><sub>n</sub> is always connected and CE<sub>n</sub> signals shall be connected in a numerically increasing order. The host shall attempt to enumerate targets connected to all host CE<sub>n</sub> signals.

The discovery process for a package that supports independent dual data buses includes additional steps to determine which data bus the target is connected to. The LGA and 100-ball BGA package with 8-bit data access are the packages within ONFI that have a dual data bus option.

#### 3.4.1.1. Single Data Bus Discovery

The CE<sub>n</sub> to test is first pulled low by the host to enable the target if connected, while all other CE<sub>n</sub> signals are pulled high. The host shall then issue the Reset (FFh) command to the target. Following the reset, the host should then issue a Read ID command to the target. If the ONFI signature is returned by the Read ID command with address 20h, then the corresponding target is connected. If the ONFI signature is not returned or any step in the process encountered an error/timeout, then the CE<sub>n</sub> is not connected and no further use of that CE<sub>n</sub> signal shall be done.

#### 3.4.1.2. Dual Data Bus Discovery

The CE<sub>n</sub> to test is first pulled low by the host to enable the target if connected, while all other CE<sub>n</sub> signals are pulled high. The host shall then issue the Reset (FFh) command to the target. Following the reset, the host should then issue a Read ID command with address 20h to the target. If the ONFI signature is returned by the Read ID command, then the corresponding target is connected.

If the ONFI signature is not returned (or any step in the process encountered an error/timeout), then the second 8-bit data bus should be probed. The host shall issue the Reset (FFh) command to the target using the second 8-bit data bus. Following the reset, the host should then issue a Read ID command with address 20h to the target on the second 8-bit data bus. If the ONFI signature is returned by the Read ID command, then the corresponding target is connected and is using the second 8-bit data bus. After discovering that the target is using the second 8-bit data bus, all subsequent commands to that target shall use the second 8-bit data bus including Read Parameter Page.

If after this point a valid ONFI signature is not discovered or further errors were encountered, then the CE<sub>n</sub> is not connected and no further use of that CE<sub>n</sub> signal shall be done.

### 3.4.2. Target Initialization

To initialize a discovered target, the following steps shall be taken. The initialization process should be followed for each connected CE<sub>n</sub> signal, including performing the Read Parameter Page (ECh) command for each target. Each chip enable corresponds to a unique target with its own independent properties that the host shall observe and subsequently use.

The host should issue the Read Parameter Page (ECh) command. This command returns information that includes the capabilities, features, and operating parameters of the device.

When the information is read from the device, the host shall check the CRC to ensure that the data was received correctly and without error prior to taking action on that data.

If the CRC of the first parameter page read is not valid (refer to section 5.7.1.4), the host should read redundant parameter page copies. The host can determine whether a redundant parameter page is present or not by checking if the first four bytes contain at least two bytes of the parameter page signature. If the parameter page signature is present, then the host should read the entirety of that redundant parameter page. The host should then check the CRC of that redundant parameter page. If the CRC is correct, the host may take action based on the contents of that redundant parameter page. If the CRC is incorrect, then the host should attempt to read the next redundant parameter page by the same procedure.

The host should continue reading redundant parameter pages until the host is able to accurately reconstruct the parameter page contents. The host may use bit-wise majority or other ECC techniques to recover the contents of the parameter page from the parameter page copies present. When the host determines that a parameter page signature is not present (refer to section 5.7.1.1), then all parameter pages have been read.

The Read ID and Read Parameter Page commands only use the lower 8-bits of the data bus. The host shall not issue commands that use a word data width on x16 devices until the host determines the device supports a 16-bit data bus width in the parameter page.

After successfully retrieving the parameter page, the host has all information necessary to successfully communicate with that target. If the host has not previously mapped defective block information for this target, the host should next map out all defective blocks in the target. The host may then proceed to utilize the target, including erase and program operations.

## 4. Data Interface and Timing

### 4.1. Data Interface Types

ONFI supports two different data interface types: asynchronous and source synchronous. The asynchronous data interface is the traditional NAND interface that uses RE\_n to latch data read, WE\_n to latch data written, and does not include a clock. The source synchronous data interface includes a clock that indicates where commands and addresses should be latched and a data strobe that indicates where data should be latched.

On power-up, the device shall operate in asynchronous data interface timing mode 0. After the host determines that the source synchronous data interface is supported in the parameter page, the host may select a source synchronous timing mode by using Set Features with a Feature Address of 01h. Refer to section 5.26.1.

The source synchronous data interface uses a DDR protocol. Thus, an even number of bytes is always transferred. The least significant bit of the column address shall always be zero in the source synchronous data interface. If the least significant bit of the column address is set to one in the source synchronous data interface then the results are indeterminate.

#### 4.1.1. Signal Function Reassignment

The function of some signals is different when using the asynchronous data interface versus when using the source synchronous data interface. When source synchronous is selected, the function of RE\_n and WE\_n is modified and DQS is enabled.

WE\_n becomes the clock signal (CLK) when in source synchronous mode. CLK shall be enabled with a valid clock period whenever a command cycle, address cycle, or data cycle is occurring. CLK shall maintain the same frequency while CE\_n is driven to zero. Refer to section 4.2.3.

RE\_n becomes the write/read direction signal (W/R\_n) when in source synchronous mode. This signal indicates the owner of the DQ data bus and the DQS signal. The host shall only transition W/R\_n when ALE and CLE are latched to zero.

The I/O bus is renamed to the DQ bus in the source synchronous interface.

A strobe signal for the DQ data bus is used in source synchronous mode, called DQS (DQ strobe). DQS is bi-directional and is used for all data transfers. DQS is not used for command or address cycles. The latching edge of DQS is center aligned to the valid data window for data transfers from the host to the device (writes). The latching edge of DQS is aligned to the transition of the DQ bus for data transfers from the device to the host (reads). DQS should be pulled high by the host and shall be ignored by the device when operating in the asynchronous data interface.

When W/R\_n changes from one to zero, the host shall tri-state the DQ bus and the DQS signal and then the device shall drive DQS to zero. When W/R\_n changes from zero to one, the device shall tri-state the DQ bus and the DQS signal. DQS and the DQ bus should be driven high by the host during idle when no data operations are outstanding and W/R\_n is set to one. There is a turn-around time whenever W/R\_n changes its value where the DQS signal is tri-stated (as neither the host nor the device is driving the signal), see section 4.3.2.6.

Symbol		Type	Description
Asynchronous	Source synchronous		
ALE	ALE	Input	Address latch enable
CE_n	CE_n	Input	Chip enable
CLE	CLE	Input	Command latch enable
I/O[7:0]	DQ[7:0]	I/O	Data inputs/outputs
—	DQS	I/O	Data strobe
RE_n	W/R_n	Input	Read enable / (Write / Read_n direction)
WE_n	CLK	Input	Write enable / Clock
WP_n	WP_n	Input	Write protect
R/B_n	R/B_n	Output	Ready / Busy_n

**Table 15 Signal Reassignments between Data Interface Types**

#### 4.1.2. Bus State

ALE and CLE are used to determine the current bus state in asynchronous and source synchronous data interfaces. Table 16 describes the bus state for asynchronous. Note that in asynchronous the value 11b for ALE/CLE is undefined.

CE_n	ALE	CLE	WE_n	RE_n	Asynchronous 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 16 ALE/CLE value and asynchronous bus state**

Table 17 describes the bus state for source synchronous. In source synchronous the value 11b for ALE/CLE is used for data transfers. The bus state lasts for an entire CLK period, starting with the rising edge of CLK. Thus, for data cycles there are two data input cycles or two data output cycles per bus state. The idle bus state is used to terminate activity on the DQ bus after a command cycle, an address cycle, or a stream of data.

The value of CE\_n shall only change when the source synchronous bus state is idle (i.e. ALE and CLE are both cleared to zero) and no data is being transmitted during that clock period.

CE_n	ALE	CLE	W/R_n	CLK	Source Synchronous Bus State
1	X	X	X	X	Standby
0	0	0	1	Rising edge to rising edge	Idle <sup>1</sup>
0	0	0	0	Rising edge to rising edge	Bus Driving <sup>1</sup>
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 <sup>2</sup>
0	1	1	0	Rising edge to rising edge	Data output cycle <sup>2</sup>
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 17 ALE/CLE value and source synchronous bus state**

#### 4.1.2.1. Pausing Data Input/Output

The host may pause data input or data output by inserting Idle cycles.

In the asynchronous data interface, pausing data input or data output is done by maintaining WE\_n or RE\_n at a value of one, respectively.

In the source synchronous data interface, pausing data input or data output is done by clearing ALE and CLE both to zero. The host may continue data transfer by setting ALE and CLE both to one after the applicable tCAD time has passed.

#### 4.1.3. Source Synchronous and Repeat Bytes

The source synchronous interface uses DDR to achieve a high data transfer rate. However, certain configuration and settings commands are not often used and do not require a high data transfer rate. Additionally, these commands typically are not serviced by the pipeline used for data transfers.

To avoid adding unnecessary complexity and requirements to implementations for these commands, the data is transferred using single data rate. Specifically, the same data byte is repeated twice and shall conform to the timings required for the source synchronous data interface. The data pattern in these cases is D<sub>0</sub> D<sub>0</sub> D<sub>1</sub> D<sub>1</sub> D<sub>2</sub> D<sub>2</sub> etc. The receiver (host or device) shall only latch one copy of each data byte. CLK should not be stopped during data input for these commands. The receiver is not required to wait for the repeated data byte before beginning internal actions.

The commands that repeat each data byte twice in the source synchronous data interface are: Set Features, Read ID, Get Features, Read Status, and Read Status Enhanced.

#### **4.1.4. Data Interface / Timing Mode Transitions**

##### **4.1.4.1. Asynchronous to Source Synchronous**

To transition from an asynchronous timing mode to a source synchronous timing mode, the procedure described in this section shall be followed. The Set Features command is used to change the data interface and timing mode. The Set Features command (EFh), Feature Address, and the four parameters are entered using the previously selected timing mode in the asynchronous data interface. When issuing the Set Features command, the host shall drive the DQS signal high. After the fourth parameter, P4, is entered until the tITC time has passed the host shall not issue any commands to the device.

Prior to issuing any new commands to the device, the host shall transition CE\_n high. When CE\_n is high, the host selects the new CLK rate. After issuing the Set Features command and prior to transitioning CE\_n high, the host shall observe the following requirements:

- ALE and CLE shall be cleared to zero
- RE\_n / W/R\_n shall be set to one
- WE\_n / CLK shall be set to one
- DQS shall be set to one

##### **4.1.4.2. Source Synchronous to Source Synchronous**

To transition from a source synchronous timing mode to another source synchronous timing mode, the procedure described in this section shall be followed. The Set Features command is used to change the timing mode. The Set Features command (EFh), Feature Address, and the four parameters are entered using the previously selected timing mode in the source synchronous data interface. After the fourth parameter, P4, is entered until the tITC time has passed the host shall not issue any commands to the device.

Prior to issuing any new commands to the device, the host shall transition CE\_n high. When CE\_n is high, the host selects the new CLK rate. After issuing the Set Features command and prior to transitioning CE\_n high, the host shall observe the following requirements:

- ALE and CLE shall be cleared to zero
- W/R\_n shall be set to one
- CLK shall continue running at the previously selected speed grade

##### **4.1.4.3. Source Synchronous to Asynchronous**

To transition from a source synchronous timing mode to an asynchronous timing mode, the procedure described in this section shall be followed. To transition from the source synchronous data interface to the asynchronous data interface, the Reset (FFh) command shall be used. After the Reset is issued, the host shall not issue any commands to the device until after the tITC time has passed. Note that after the tITC time has passed, only status commands may be issued by the host until the Reset completes.

The host shall transition to the asynchronous data interface. Then the host shall issue the Reset (FFh) command described in the previous paragraph using asynchronous timing mode 0, thus the host transitions to the asynchronous data interface prior to issuing the Reset (FFh). A device in any timing mode is required to recognize a Reset (FFh) command issued in asynchronous timing mode 0. After issuing the Reset (FFh) and prior to transitioning CE\_n high, the host shall observe the following requirements:

- ALE and CLE shall be cleared to zero

- RE\_n / W/R\_n shall be set to one
- WE\_n / CLK shall be set to one

After CE\_n has been pulled high and then transitioned low again, the host should issue a Set Features to select the appropriate asynchronous timing mode.

## 4.2. Timing Parameters

All timing parameters are from a host perspective. For example, the “Minimum WE\_n pulse width” is the minimum allowed WE\_n pulse width that the host is permitted to present to the device while still assuring correct operation of the device. The behavior of the device when the required host minimum and maximum times are not adhered to is undefined. Note that the host needs to account for channel effects in meeting the specified timings with the device.

### 4.2.1. General Timings

This section describes timing parameters that apply regardless of the data interface type being used.

For execution of the first Read Parameter Page command, prior to complete initialization, a tR value of 200 microseconds and tCCS value of 500 ns shall be used. For page reads, including execution of additional Read Parameter Page commands after initialization is complete, the value for tR and tCCS contained in the parameter page shall be used.

There are three maximums listed for tRST in the asynchronous and source synchronous data interfaces. The target is allowed a longer maximum reset time when a program or erase operation is in progress. The maximums correspond to:

1. The target is not performing an erase or program operation.
2. The target is performing a program operation.
3. The target is performing an erase operation.

Table 18 defines the array timing parameters. The array timing parameter values are either returned in the parameter page (tR, tPROG, tBERS, and tCCS) or they are statically defined in Table 19.

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

**Table 18 Array Timing Parameter Descriptions**

There are “short” busy times associated with cache operations (tRCBSY, tPCBSY) and multi-plane operations (tPLEBSY, tPLPBSY, and tPLRBSY). Typical and maximum times for these busy times are listed in Table 19.

Parameter	Typical	Maximum
tPLEBSY	500 ns	tBERS
tPLPBSY	500 ns	tPROG
tPLRBSY	500 ns	tR
tPCBSY	3 $\mu$ s	tPROG
tRCBSY	3 $\mu$ s	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 19 Cache and Multi-plane Short Busy Times**

#### 4.2.2. Asynchronous

Table 20 defines the descriptions of all timing parameters. Table 23 and Table 24 define the requirements for timing modes 0, 1, 2, 3, 4, and 5. Timing mode 0 shall always be supported and the device operates in this mode at power-on. A host shall only begin use of a more advanced timing mode after determining that the device supports that timing mode in Read Parameter Page.

The host shall use EDO data output cycle timings, as defined in section 4.3.1.5, when running with a tRC value less than 30 ns.

Parameter	Description
tADL <sup>3</sup>	ALE to data loading time
tALH	ALE hold time
tALS	ALE setup time
tAR	ALE to RE_n delay
tCCS <sup>3</sup>	WE_n high to RE_n low, value specified in parameter page
tCEA	CE_n access time
tCEH	CE_n high hold time
tCH	CE_n hold time
tCHZ <sup>2</sup>	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
tCS	CE_n setup time
tDH	Data hold time
tDS	Data setup time
tFEAT <sup>1</sup>	Busy time for Set Features and Get Features
tIR <sup>2</sup>	Output hi-Z to RE_n low
tITC <sup>1</sup>	Interface and Timing Mode Change time
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 <sup>2</sup>	RE_n high to output hi-Z
tRLOH	RE_n low to output hold
tRP	RE_n pulse width
tRR	Ready to RE_n low (data only)
tRST	Device reset time, measured from the falling edge of R/B_n to the rising edge of R/B_n.
tWB <sup>4</sup>	WE_n high to SR[6] low
tWC	WE_n cycle time
tWH	WE_n high hold time
tWHR <sup>3</sup>	WE_n high to RE_n low
tWP	WE_n pulse width
tWW	WP_n transition to WE_n low

**NOTE:**

1. Measured from the falling edge of SR[6] to the rising edge of SR[6].
2. Refer to Appendix E for measurement technique.
3. 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.
4. Commands (including Read Status / Read Status Enhanced) shall not be issued until

after  $t_{WB}$  is complete.

**Table 20 Asynchronous Timing Parameter Descriptions**

The testing conditions that shall be used to verify that a device complies with a particular asynchronous timing mode are listed in Table 21 for devices that support the asynchronous data interface only and do not support driver strength settings.

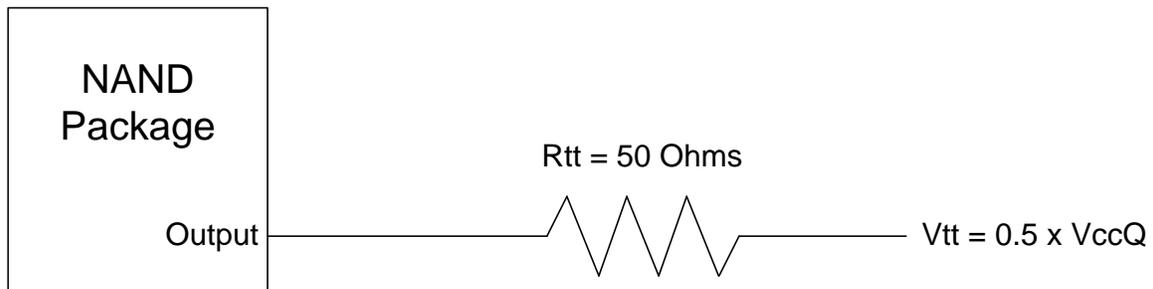
Parameter	Value
Input pulse levels	0.0 V to $V_{ccQ}$
Input rise and fall times	5 ns
Input and output timing levels	$V_{ccQ} / 2$
Output load for $V_{ccQ}$ of 3.3V	CL = 50 pF
Output load for $V_{ccQ}$ of 1.8V	CL = 30 pF

**Table 21 Testing Conditions for Asynchronous Only Devices**

The testing conditions that shall be used to verify compliance with a particular timing mode for devices that support driver strength settings are listed in Table 22. This includes all devices that support the source synchronous data interface. It also includes devices that only support the asynchronous data interface that support driver strength settings.

Parameter	Value
Positive input transition	VIL (DC) to VIH (AC)
Negative input transition	VIH (DC) to VIL (AC)
Minimum input slew rate	$tIS = 1.0 \text{ V/ns}$
Input timing levels	$V_{ccQ} / 2$
Output timing levels	$V_{ccQ} / 2$
Driver strength	Nominal
Output capacitive load <sup>1</sup>	CL = 5 pF
EZ NAND output reference load	50 Ohm to $V_{tt}$
NOTE: 1. Assumes small propagation delay from output to CL.	

**Table 22 Testing Conditions for Devices that Support Driver Strength Settings**



**Figure 22 EZ NAND Output Reference Load**

Parameter	Mode 0		Mode 1		Mode 2		Unit
	100		50		35		
	Min	Max	Min	Max	Min	Max	
tADL	200	—	100	—	100	—	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
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 <sup>2</sup> (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:

- 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.
- 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 LUN Reset (FAh) then the values are as shown.

**Table 23 Asynchronous Timing Modes 0, 1, and 2**

Parameter	Mode 3		Mode 4 (EDO capable)		Mode 5 (EDO capable)		Unit
	30		25		20		
	Min	Max	Min	Max	Min	Max	
tADL	100	—	70	—	70	—	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
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 <sup>2</sup> (EZ NAND)	—	150/15 0/500	—	150/ 150/ 500	—	150/ 150/ 500	μs
tWB	—	100	—	100	—	100	ns
tWC	30	—	25	—	20	—	ns
tWH	10	—	10	—	7	—	ns
tWHR	60	—	60	—	60	—	ns
tWP	15	—	12	—	10	—	ns
tWW	100	—	100	—	100	—	ns

NOTE:

- 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.
- 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 LUN Reset (FAh) then the values are as shown.

**Table 24 Asynchronous Timing Modes 3, 4, and 5**

### **4.2.3. Source Synchronous**

All source synchronous timing parameters are referenced to the rising edge of CLK or the latching edge of DQS. Note that R/B\_n and WP\_n are always asynchronous signals.

If CLK is a different frequency than those described in the source synchronous timing modes, then the host shall meet the setup and hold requirements for the next fastest timing mode.

For parameters measured in clocks (e.g. tDSH), the parameter is measured starting from a latching edge of CLK or DQS, respectively.

Parameter	Description
tAC	Access window of DQ[7:0] from CLK
tADL <sup>5</sup>	Address cycle to data loading time
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
tCCS <sup>5</sup>	Change Column setup time, value specified in parameter page
tCEH	CE_n high hold time
tCH	CE_n hold time
tCK(avg) <sup>2</sup>	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) <sup>3</sup>	Clock cycle high
tCKL(abs) <sup>3</sup>	Clock cycle low
tCKWR	Data output end to W/R_n high
tCS	CE_n setup time
tDH	Data DQ hold time
tDPZ	Data input pause setup time
tDQsCK	Access window of DQS from CLK
tDQSD <sup>4</sup>	W/R_n low to DQS/DQ driven by device
tDQSH	DQS input high pulse width
tDQSHZ <sup>4</sup>	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
tDS	Data DQ setup time
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
tFEAT <sup>1</sup>	Busy time for Set Features and Get Features
tHP	Half-clock period

tITC <sup>1</sup>	Interface and Timing Mode Change time
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
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 <sup>6</sup>	CLK rising edge to SR[6] low
tWHR <sup>5</sup>	Command, address or data input cycle to data output cycle
tWPRE	DQS write preamble
tWPST	DQS write postamble
tWRCK	W/R_n low to data output cycle
tWW	WP_n transition to command cycle
<p>NOTE:</p> <ol style="list-style-type: none"> <li>1. Measured from the falling edge of SR[6] to the rising edge of SR[6].</li> <li>2. tCK(avg) is the average clock period over any consecutive 200 cycle window.</li> <li>3. tCKH(abs) and tCKL(abs) include static offset and duty cycle jitter.</li> <li>4. Refer to Appendix E for measurement technique.</li> <li>5. 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.</li> <li>6. Commands (including Read Status / Read Status Enhanced) shall not be issued until after tWB is complete.</li> </ol>	

**Table 25 Source Synchronous Timing Parameter Descriptions**

The device may be configured with multiple driver strengths with the Set Features command. There is a 50 Ohm, 35 Ohm, 25 Ohm, and 18 Ohm setting that the device may support. Support for all four driver strength settings is required for devices that support the source synchronous data interface. A device that only supports the asynchronous data interface may support all or a subset of driver strength settings. Devices that support driver strength settings shall comply with the output driver requirements in this section.

Setting	Driver Strength	VccQ
18 Ohms	2.0x = 18 Ohms	3.3V
25 Ohms	1.4x = 25 Ohms	
35 Ohms	1.0x = 35 Ohms	
50 Ohms	0.7x = 50 Ohms	
18 Ohms	2.0x = 18 Ohms	1.8V
25 Ohms	1.4x = 25 Ohms	
35 Ohms	1.0x = 35 Ohms	
50 Ohms	0.7x = 50 Ohms	

**Table 26 I/O Drive Strength Settings**

The impedance values correspond to several different VccQ values are defined in Table 28 for 3.3V VccQ and Table 29 for 1.8V VccQ. The test conditions that shall be used to verify the impedance values is specified in Table 27.

Condition	Temperature (TA)	VccQ (3.3V)	VccQ (1.8V)	Process
Minimum Impedance	TOPER (Min) degrees Celsius	3.6V	1.95V	Fast-fast
Nominal Impedance	25 degrees Celsius	3.3V	1.8V	Typical
Maximum impedance	TOPER (Max) degrees Celsius	2.7V	1.7V	Slow-slow

**Table 27      Testing Conditions for Impedance Values**

<b>R<sub>ON</sub> = 18 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	28.7	16.2	7.0	Ohms
	0.5 x V <sub>CCQ</sub>	36.0	18.0	9.0	Ohms
	0.8 x V <sub>CCQ</sub>	50.0	21.0	11.8	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	50.0	21.0	11.8	Ohms
	0.5 x V <sub>CCQ</sub>	36.0	18.0	9.0	Ohms
	0.8 x V <sub>CCQ</sub>	28.7	14.0	7.0	Ohms
<b>R<sub>ON</sub> = 25 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	40.0	22.3	9.3	Ohms
	0.5 x V <sub>CCQ</sub>	50.0	25.0	12.6	Ohms
	0.8 x V <sub>CCQ</sub>	68.0	29.0	16.3	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	68.0	29.0	16.3	Ohms
	0.5 x V <sub>CCQ</sub>	50.0	25.0	12.6	Ohms
	0.8 x V <sub>CCQ</sub>	40.0	19.0	9.3	Ohms
<b>R<sub>ON</sub> = 35 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	58.0	32.0	12.8	Ohms
	0.5 x V <sub>CCQ</sub>	70.0	35.0	18.0	Ohms
	0.8 x V <sub>CCQ</sub>	95.0	40.0	23.0	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	95.0	40.0	23.0	Ohms
	0.5 x V <sub>CCQ</sub>	70.0	35.0	18.0	Ohms
	0.8 x V <sub>CCQ</sub>	58.0	32.0	12.8	Ohms
<b>R<sub>ON</sub> = 50 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	80.0	45.0	18.4	Ohms
	0.5 x V <sub>CCQ</sub>	100.0	50.0	25.0	Ohms
	0.8 x V <sub>CCQ</sub>	136.0	57.0	32.0	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	136.0	57.0	32.0	Ohms
	0.5 x V <sub>CCQ</sub>	100.0	50.0	25.0	Ohms
	0.8 x V <sub>CCQ</sub>	80.0	45.0	18.4	Ohms

**Table 28 Impedance Values for 3.3V V<sub>CCQ</sub>**

<b>R<sub>ON</sub> = 18 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	34.0	13.5	7.5	Ohms
	0.5 x V <sub>CCQ</sub>	31.0	18.0	9.0	Ohms
	0.8 x V <sub>CCQ</sub>	44.0	23.5	11.0	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	44.0	23.5	11.0	Ohms
	0.5 x V <sub>CCQ</sub>	31.0	18.0	9.0	Ohms
	0.8 x V <sub>CCQ</sub>	34.0	13.5	7.5	Ohms
<b>R<sub>ON</sub> = 25 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	47.0	19.0	10.5	Ohms
	0.5 x V <sub>CCQ</sub>	44.0	25.0	13.0	Ohms
	0.8 x V <sub>CCQ</sub>	61.5	32.5	16.0	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	61.5	32.5	16.0	Ohms
	0.5 x V <sub>CCQ</sub>	44.0	25.0	13.0	Ohms
	0.8 x V <sub>CCQ</sub>	47.0	19.0	10.5	Ohms
<b>R<sub>ON</sub> = 35 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	66.5	27.0	15.0	Ohms
	0.5 x V <sub>CCQ</sub>	62.5	35.0	18.0	Ohms
	0.8 x V <sub>CCQ</sub>	88.0	52.0	22.0	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	88.0	52.0	22.0	Ohms
	0.5 x V <sub>CCQ</sub>	62.5	35.0	18.0	Ohms
	0.8 x V <sub>CCQ</sub>	66.5	27.0	15.0	Ohms
<b>R<sub>ON</sub> = 50 Ohms</b>					
<b>Description</b>	<b>VOUT to V<sub>SSQ</sub></b>	<b>Maximum</b>	<b>Nominal</b>	<b>Minimum</b>	<b>Unit</b>
R_pulldown	0.2 x V <sub>CCQ</sub>	95.0	39.0	21.5	Ohms
	0.5 x V <sub>CCQ</sub>	90.0	50.0	26.0	Ohms
	0.8 x V <sub>CCQ</sub>	126.5	66.5	31.5	Ohms
R_pullup	0.2 x V <sub>CCQ</sub>	126.5	66.5	31.5	Ohms
	0.5 x V <sub>CCQ</sub>	90.0	50.0	26.0	Ohms
	0.8 x V <sub>CCQ</sub>	95.0	39.0	21.5	Ohms

**Table 29 Impedance Values for 1.8V V<sub>CCQ</sub>**

The pull-up and pull-down impedance mismatch is defined in Table 30. Impedance mismatch is the absolute value between pull-up and pull-down impedances. Both are measured at the same temperature and voltage. The testing conditions that shall be used to verify the impedance mismatch requirements are: V<sub>CCQ</sub> = V<sub>CCQ</sub>(min), V<sub>OUT</sub> = V<sub>CCQ</sub> x 0.5, and T<sub>A</sub> is across the full operating range.

Output Impedance	Maximum	Minimum	Unit
18 Ohms	6.3	0.0	Ohms
25 Ohms	8.8	0.0	Ohms
35 Ohms	12.3	0.0	Ohms
50 Ohms	17.5	0.0	Ohms

**Table 30 Pull-up and Pull-down Impedance Mismatch**

The input slew rate requirements that the device shall comply with are defined in Table 31. The testing conditions that shall be used to verify the input slew rate are listed in Table 32.

Description	Timing Mode						Unit
	0	1	2	3	4	5	
Input slew rate (min)	0.5	0.5	0.5	0.5	0.5	0.5	V/ns
Input slew rate (max)	4.5	4.5	4.5	4.5	4.5	4.5	V/ns
Derating factor for setup times, address and command	TBD	TBD	TBD	TBD	TBD	TBD	ns per 100 mV
Derating factor for hold times, address and command	TBD	TBD	TBD	TBD	TBD	TBD	ns per 100 mV
Derating factor for setup times, data input and output	TBD	TBD	TBD	TBD	TBD	TBD	ns per 100 mV
Derating factor for hold times, data input and output	TBD	TBD	TBD	TBD	TBD	TBD	ns per 100 mV

**Table 31 Input Slew Rate Requirements**

Parameter	Value
Positive input transition	VIL (DC) to VIH (AC)
Negative input transition	VIH (DC) to VIL (AC)

**Table 32 Testing Conditions for Input Slew Rate**

The output slew rate requirements that the device shall comply with are defined in Table 33 and Table 34 for a single LUN per 8-bit data bus. The testing conditions that shall be used to verify the output slew rate are listed in Table 35.

Description	Output Slew Rate		Unit	Normative or Recommended
	Min	Max		
18 Ohms	1.5	10.0	V/ns	Normative
25 Ohms	1.5	9.0	V/ns	Normative
35 Ohms	1.2	7.0	V/ns	Normative
50 Ohms	1.0	5.5	V/ns	Recommended

**Table 33 Output Slew Rate Requirements for 3.3V VccQ**

Description	Output Slew Rate		Unit	Normative or Recommended
	Min	Max		
18 Ohms	1.0	5.5	V/ns	Normative
25 Ohms	0.85	5.0	V/ns	Normative
35 Ohms	0.75	4.0	V/ns	Normative
50 Ohms	0.60	4.0	V/ns	Recommended

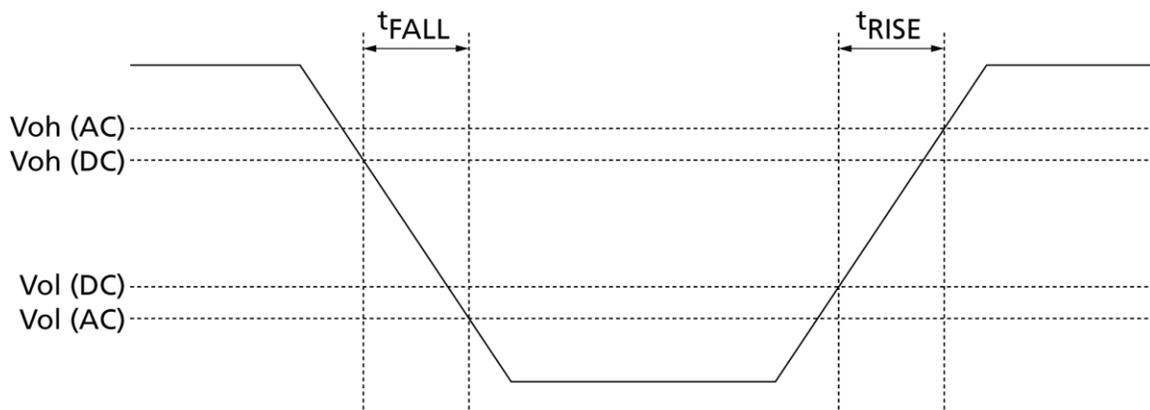
**Table 34 Output Slew Rate Requirements for 1.8V VccQ**

Parameter	Value
VOL(DC)	$0.3 * V_{ccQ}$
VOH(DC)	$0.7 * V_{ccQ}$
VOL(AC)	$0.2 * V_{ccQ}$
VOH(AC)	$0.8 * V_{ccQ}$
Positive output transition	VOL (DC) to VOH (AC)
Negative output transition	VOH (DC) to VOL (AC)
tRISE <sup>1</sup>	Time during rising edge from VOL(DC) to VOH(AC)
tFALL <sup>1</sup>	Time during falling edge from VOH(DC) to VOL(AC)
Output slew rate rising edge	$[VOH(AC) - VOL(DC)] / tRISE$
Output slew rate falling edge	$[VOH(DC) - VOL(AC)] / tFALL$
Output load	Refer to Table 22

NOTE:

1. Refer to Figure 23.
2. Output slew rate is verified by design and characterization; it may not be subject to production test.
3. The minimum slew rate is the minimum of the rising edge and the falling edge slew rate. The maximum slew rate is the maximum of the rising edge and the falling edge slew rate.

**Table 35 Testing Conditions for Output Slew Rate**



**Figure 23 tRISE and tFALL Definition for Output Slew Rate**

The testing conditions that shall be used to verify that a device complies with a particular source synchronous timing mode are listed in Table 36. The test conditions are the same regardless of the number of LUNs per Target.

Parameter	Value
Positive input transition	VIL (DC) to VIH (AC)
Negative input transition	VIH (DC) to VIL (AC)
Minimum input slew rate	tIS = 1.0 V/ns
Input timing levels	VccQ / 2
Output timing levels	VccQ / 2
Driver strength	Nominal
Output capacitive load <sup>1</sup>	CL = 5 pF
NOTE: 1. Assumes small propagation delay from output to CL.	

**Table 36 Testing Conditions for Source Synchronous Timing Modes**

The input capacitance requirements are defined in Table 37. The testing conditions that shall be used to verify the input capacitance requirements are: temperature of 25 degrees Celsius,  $V_{IN} = 0V$ , and a CLK frequency of 100 MHz. The capacitance delta values measure the pin-to-pin capacitance for all LUNs within a package, including across data buses. The capacitance delta values change based on the number of LUNs per x8 data bus. All targets that share an I/O should report an equivalent Typical capacitance in order to meet the capacitance delta requirements. If targets with different Typical capacitance values share an I/O bus, then the values in these tables are not applicable and detailed topology and signal integrity analysis needs to be completed by the implementer to determine the bus speed that is achievable.

EZ NAND has a controller stacked in the NAND package and the LUNs are physically connected to the controller and not to the external data bus. EZ NAND values are defined separately from the values specified in Table 37. For the Typical ranges in Table 38 and Table 39, EZ NAND is specified separately since ASIC capacitance tends to be higher.

Parameter	Number of LUNs per x8 data bus				Unit
	1	2	4	8	
Typical Variance	±1.0	±1.5	±2.0	±4.0	pF
Delta	1.4	1.7	2.0	4.0	pF
NOTE: 1. Typical capacitance values for pin groups CCK, CIN, and CIO are specified in the parameter page. The allowable range for Typical capacitance values is specified in Table 38 for CLK and input pins and Table 39 for I/O pins. 2. Input capacitance for CE_n and WP_n shall not exceed the maximum capacitance value for CLK and input pins. However, CE_n and WP_n do not have to meet delta or variance requirements.					

**Table 37 Input Capacitance, Minimum and Maximums**

The Typical capacitance values shall be constrained to the ranges defined in Table 38 for CLK and input pins and Table 39 for I/O pins for devices in a BGA package. Capacitance is shared for

LUNs that share the same 8-bit data bus in the same package, thus the ranges are specific to the number of LUNs per data bus.

Parameter	Source Synchronous Timing Modes 0-5				Typ Variance	Unit
	Min	Typ Low	Typ High	Max		
1 LUN per x8 data bus	2.0	3.0	5.4	6.4	±0.5	pF
2 LUNs per x8 data bus	3.5	5.0	7.7	9.2	±1.0	pF
4 LUNs per x8 data bus	5.0	7.0	12.3	14.3	±2.0	pF
8 LUNs per x8 data bus	6.0	10.0	23.3	27.3	±4.0	pF
EZ NAND	3.0	5.0	6.5	8.5	±2.0	pF

**Table 38 Input Capacitance for CLK and input pins, Typical Ranges**

Parameter	Source Synchronous Timing Modes 0-5				Typ Variance	Unit
	Min	Typ Low	Typ High	Max		
1 LUN per x8 data bus	3.0	4.0	5.7	6.7	±0.5	pF
2 LUNs per x8 data bus	5.0	6.5	8.5	10.0	±1.0	pF
4 LUNs per x8 data bus	8.9	10.9	14.9	16.9	±2.0	pF
8 LUNs per x8 data bus	16.7	20.7	28.7	32.7	±4.0	pF
EZ NAND	3.0	5.0	6.5	8.5	±2.0	pF

**Table 39 Input Capacitance for I/O pins, Typical Ranges**

**NOTE:** Capacitance ranges are not defined for the TSOP package due to the varying TSOP package construction techniques and bond pad locations. For the TSOP package compared to the BGA package, the input capacitance delta values do not apply, input capacitance values for I/O pins is similar, and input capacitance values for input pins could be significantly higher. For higher speed applications, the BGA-63 or BGA-100 packages are recommended due to their lower and more consistent input capacitance and input capacitance delta values.

Table 40 describes the standard source synchronous timing modes. The host is not required to have a clock period that exactly matches any of the clock periods listed for the standard timing modes. The host shall meet the setup and hold times for the timing mode selected. If the host selects timing mode **n** using Set Features, then its clock period shall be faster than the clock period of timing mode **n-1** and slower than or equal to the clock period of timing mode **n**. For example, if the host selects timing mode 2, then the following equation shall hold:

$$30 \text{ ns} > \text{host clock period} \geq 20 \text{ ns}$$

If timing mode 0 is selected, then the clock period shall be no slower than 100 ns. The only exception to this requirement is when the host is issuing a Reset (FFh) in asynchronous timing mode 0 (see section 4.1.4.3).

Timing parameters that indicate a latency requirement before a data input, data output, address or command cycle shall be satisfied to the rising clock edge after the latency in nanoseconds has elapsed. To calculate the first edge where the associated transition may be made, it is calculated as follows:

$$= \text{RoundUp}\{[t\text{Param} + t\text{CK}] / t\text{CK}\}$$

Parameter	Mode 0		Mode 1		Mode 2		Mode 3		Mode 4		Mode 5		Unit
	50		30		20		15		12		10		
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	MHz
tAC	3	25	3	25	3	25	3	25	3	25	3	25	ns
tADL	100	—	100	—	70	—	70	—	70	—	70	—	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 <sup>4</sup>
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 <sup>4</sup>
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 <sup>2</sup> (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 LUN Reset (FAh) then the values are as shown.

**Table 40 Source Synchronous Timing Modes**

## 4.3. Timing Diagrams

### 4.3.1. Asynchronous

#### 4.3.1.1. Command Latch Timings

The requirements for the R/B\_n signal only apply to commands where R/B\_n is cleared to zero after the command is issued, as specified in the command definitions.

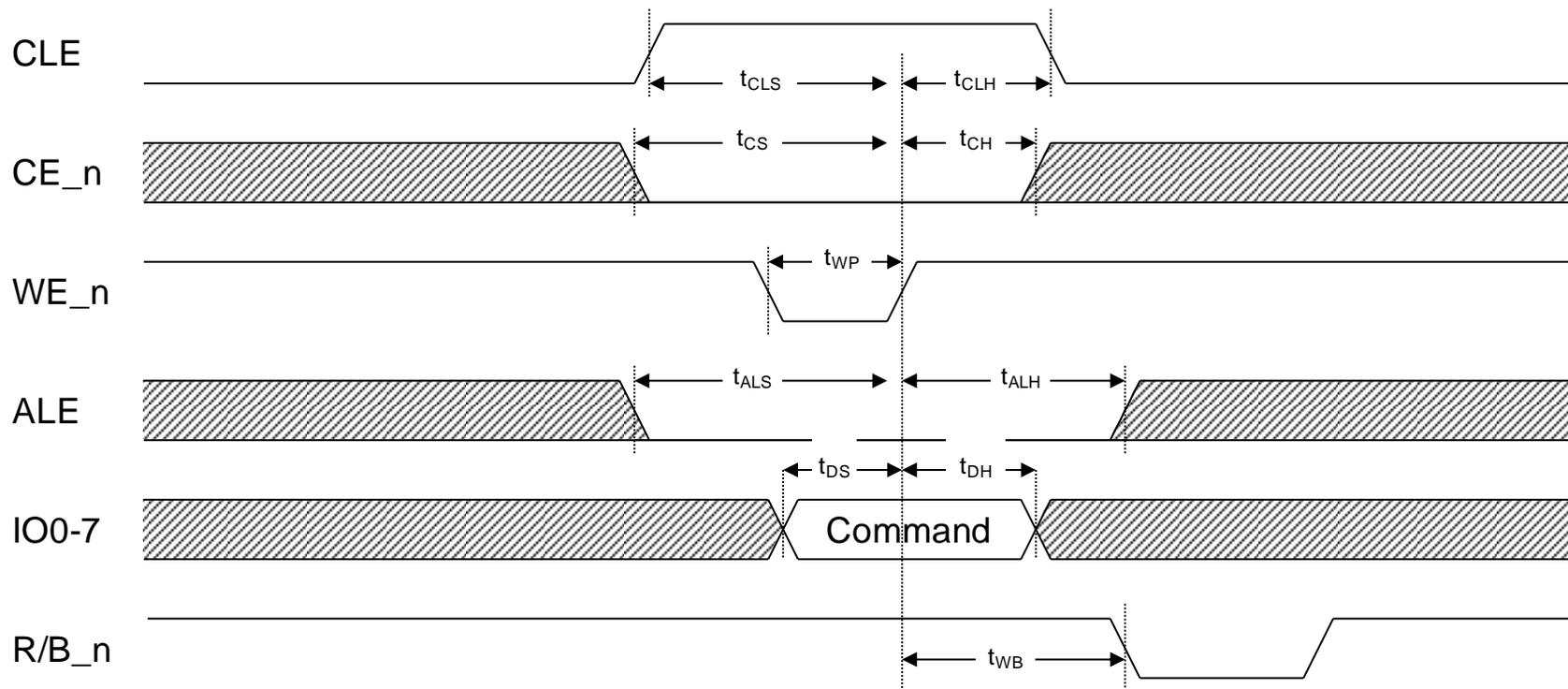


Figure 24 Command latch timings

4.3.1.2. Address Latch Timings

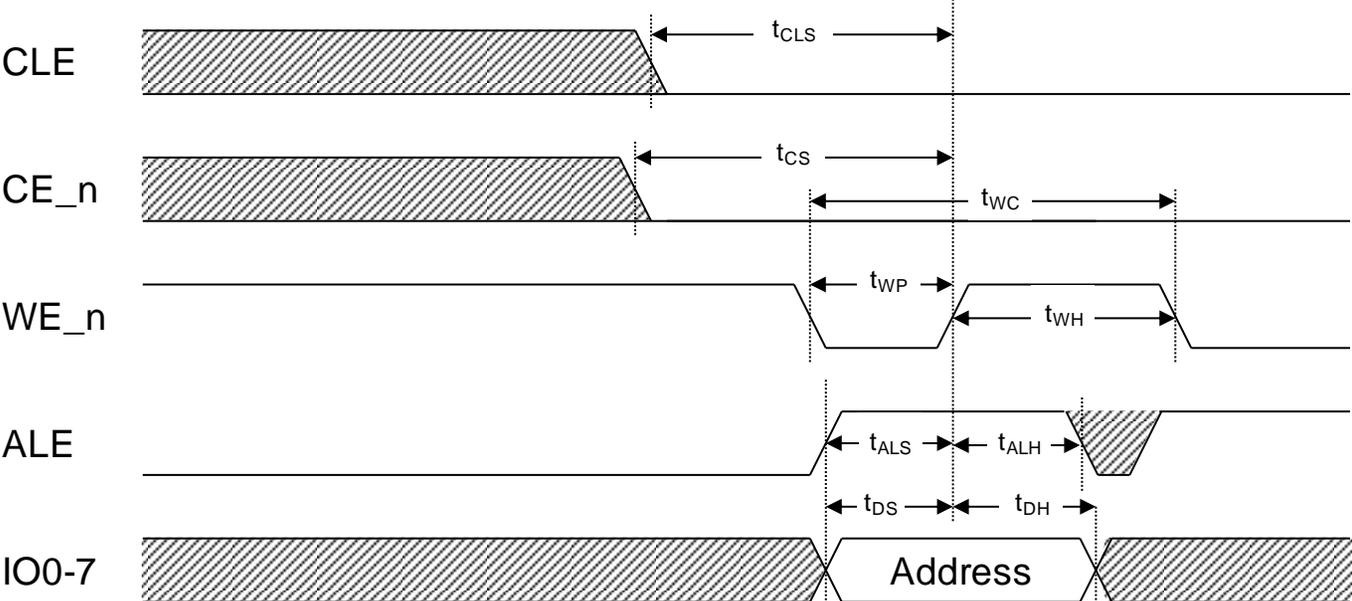


Figure 25 Address latch timings

### 4.3.1.3. Data Input Cycle Timings

Data input may be used with CE\_n don't care. However, if CE\_n don't care is used tCS and tCH timing requirements shall be met by the host, and WE\_n falling edge shall always occur after the CE\_n falling edge (i.e. tCS > tWP).

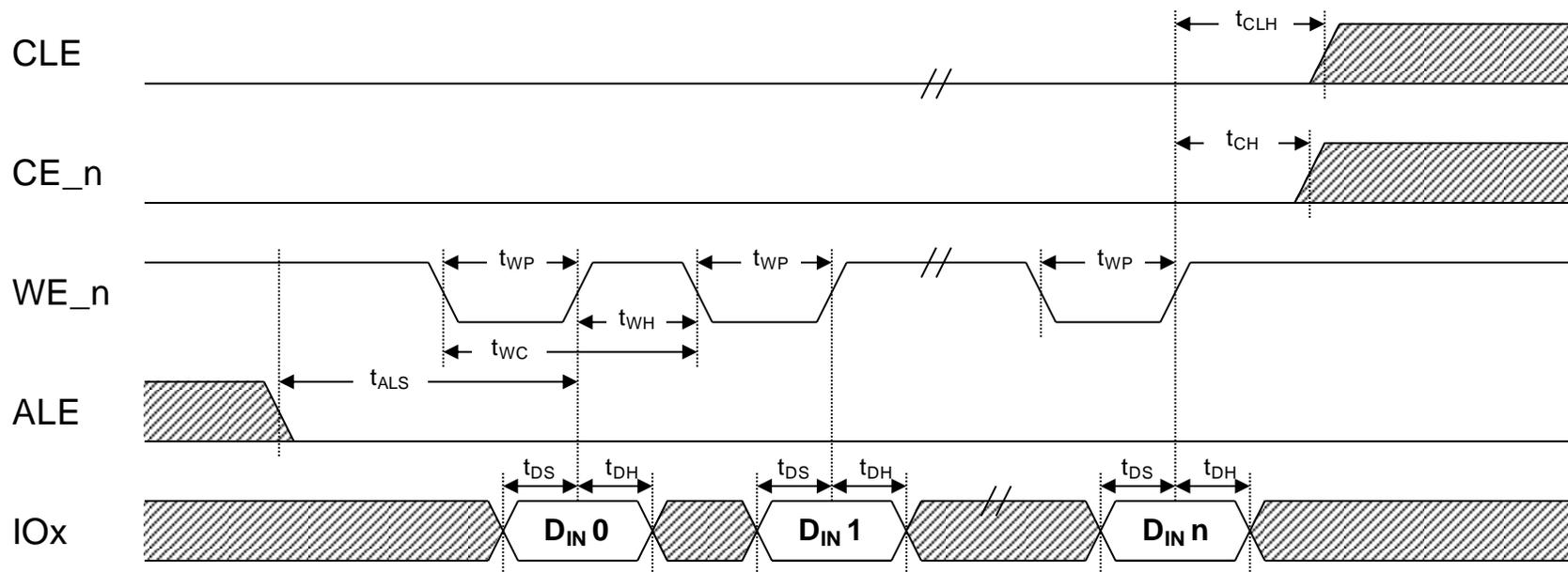


Figure 26 Data input cycle timings

#### 4.3.1.4. Data Output Cycle Timings

Data output may be used with CE\_n don't care. However, if CE\_n don't care is used tCEA and tCOH timing requirements shall be met by the host and RE\_n shall either remain low or RE\_n falling edge shall occur after the CE\_n falling edge.

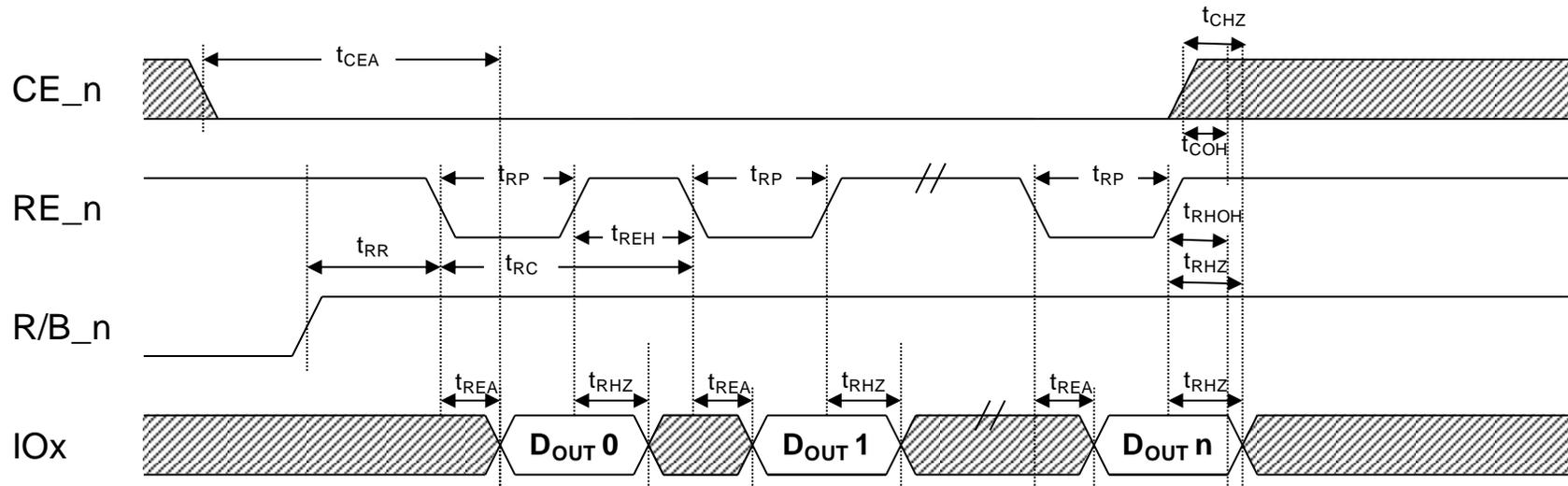


Figure 27 Data output cycle timings

### 4.3.1.5. Data Output Cycle Timings (EDO)

EDO data output cycle timings shall be used if the host drives  $t_{RC}$  less than 30 ns. Data output may be used with CE\_n don't care. However, if CE\_n don't care is used  $t_{CEA}$  and  $t_{COH}$  timing requirements shall be met by the host.

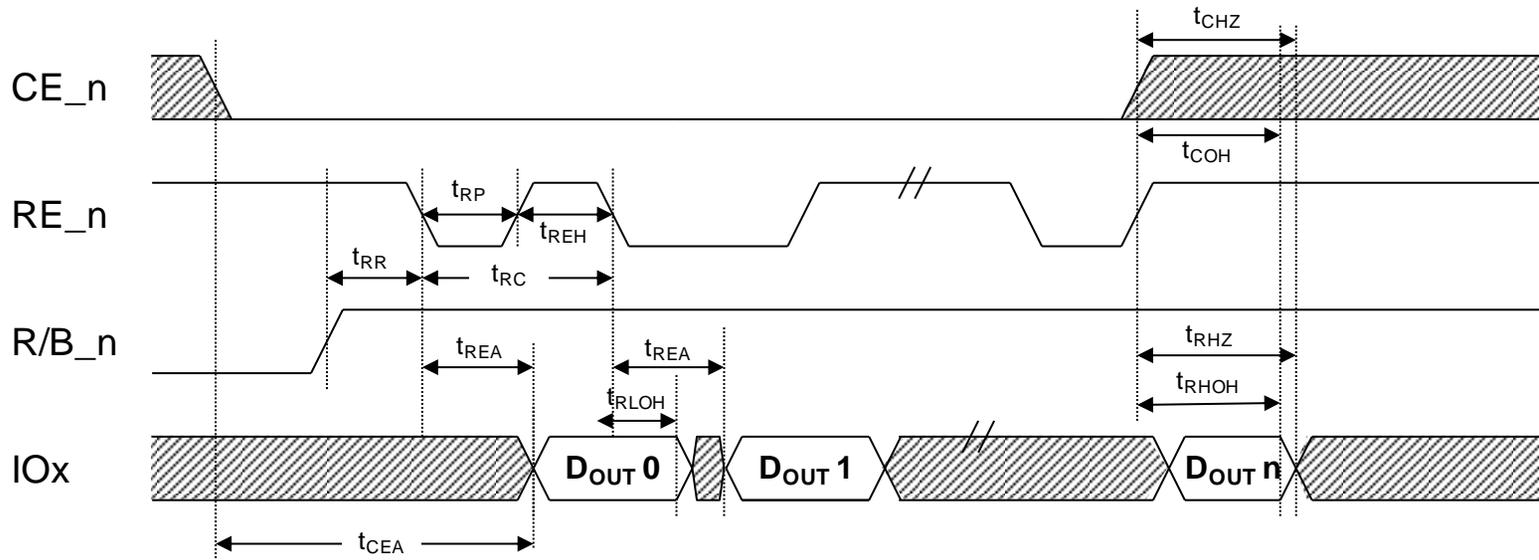


Figure 28 EDO data output cycle timings

4.3.1.6. Read Status Timings

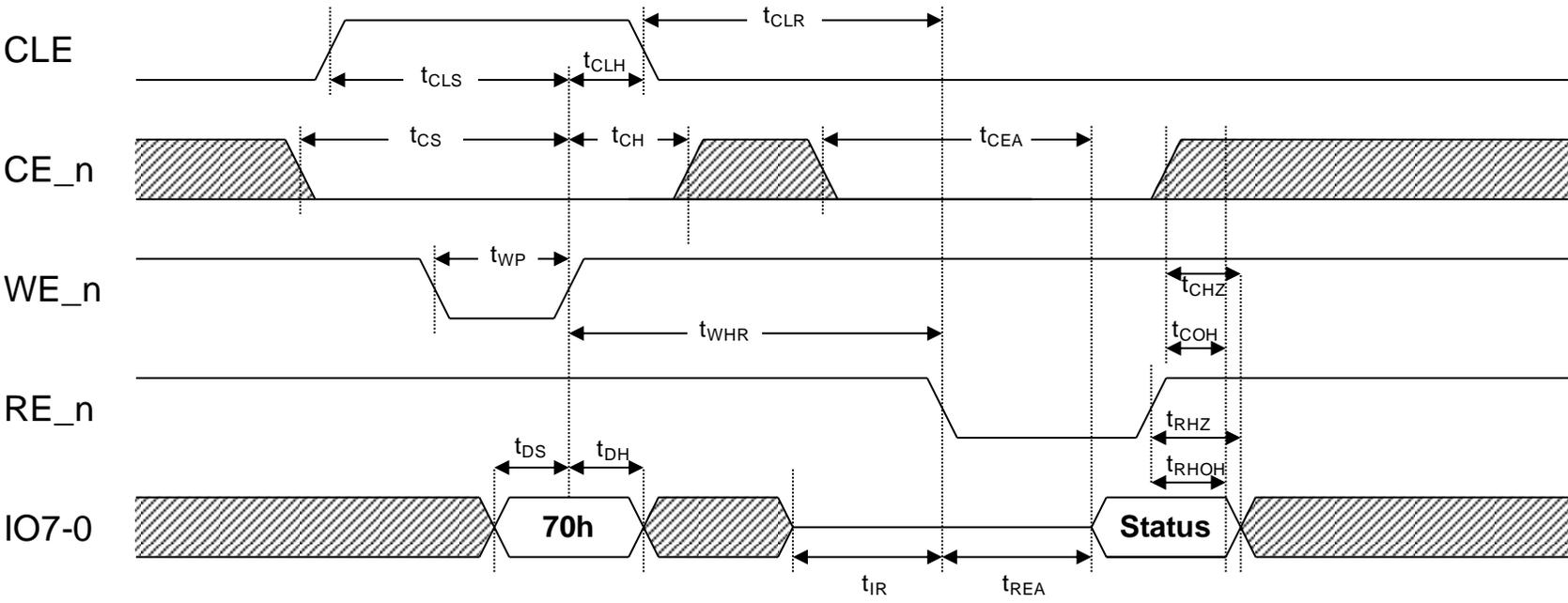


Figure 29 Read Status timings

### 4.3.1.7. Read Status Enhanced Timings

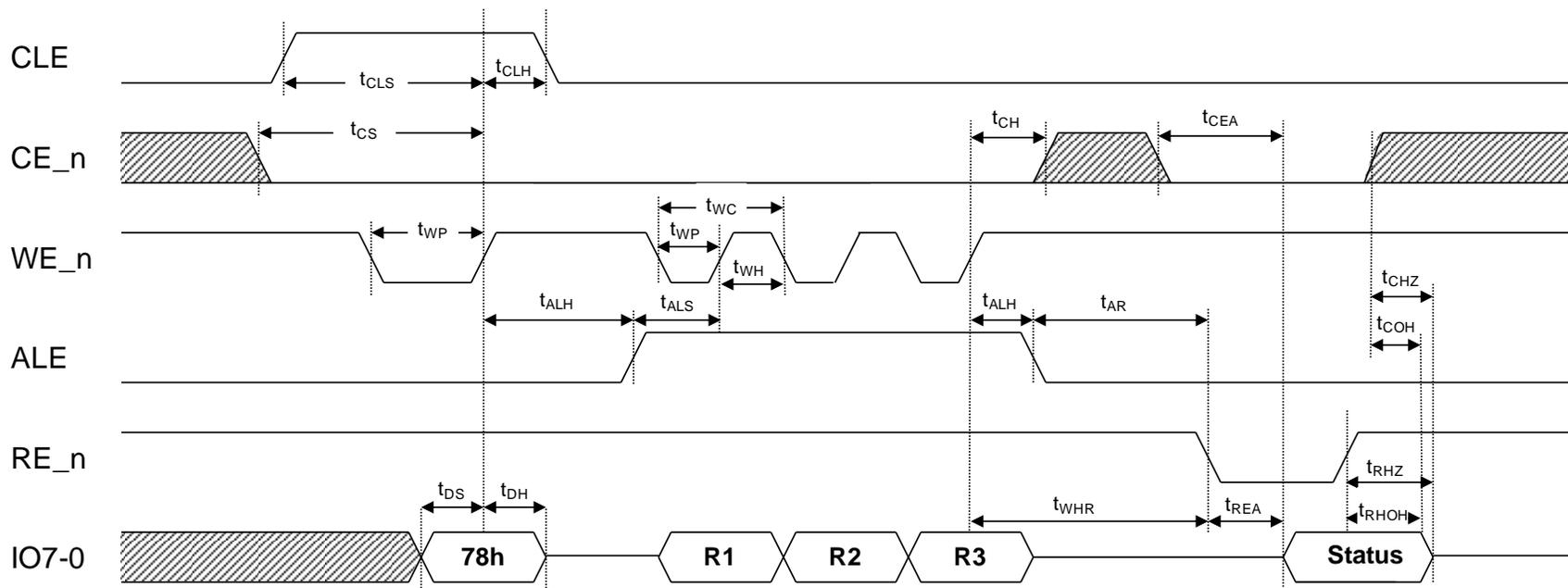


Figure 30 Read Status Enhanced timings

### 4.3.2. Source Synchronous

For the command, address, data input, and data output diagrams, the  $t_{CS}$  timing parameter may consume multiple clock cycles. The host is required to satisfy  $t_{CS}$  by the rising edge of CLK shown in the diagrams, and thus needs to pull  $CE_n$  low far enough in advance to meet this requirement (which could span multiple clock cycles).

#### 4.3.2.1. Command Cycle Timings

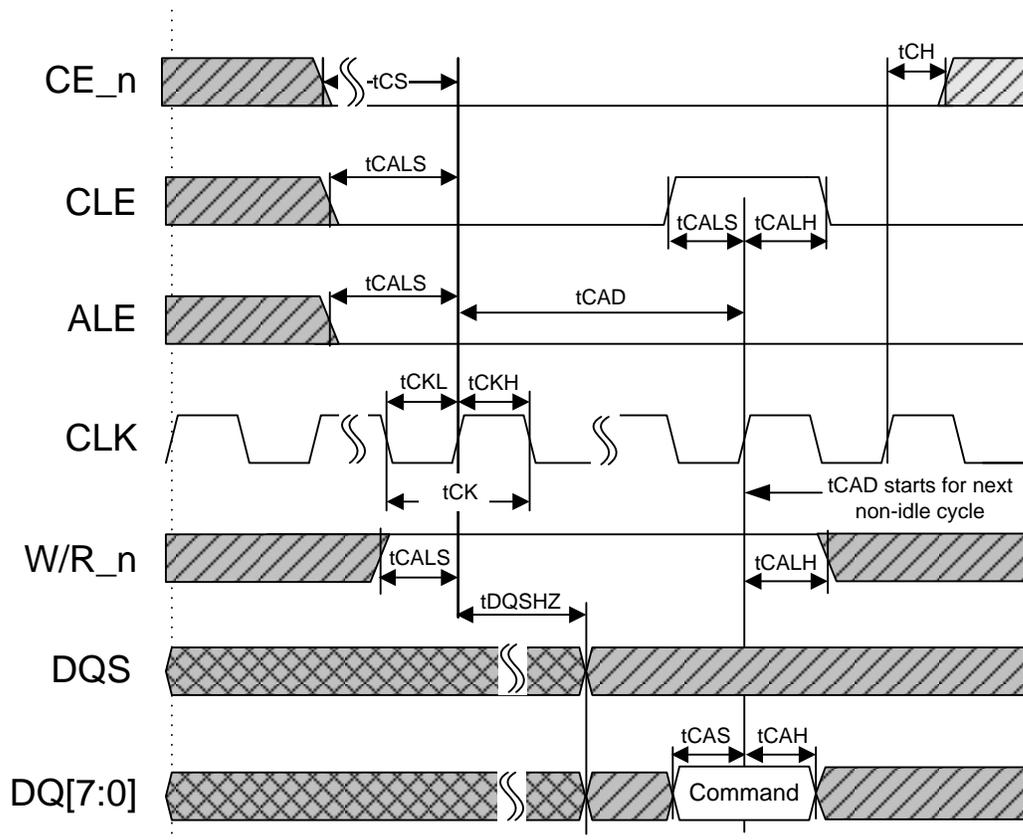


Figure 31 Command cycle timings

**NOTE:**

1. The cycle that  $t_{CAD}$  is measured from may be an idle cycle (as shown), another command cycle, an address cycle, or a data cycle. The idle cycle is shown in this diagram for simplicity.
2.  $ALE$  and  $CLE$  shall be in a valid state when  $CE_n$  transitions from one to zero. In the diagram, it appears that  $t_{CS}$  and  $t_{CALS}$  are equivalent times. However,  $t_{CS}$  and  $t_{CALS}$  values are not the same, the timing parameter values should be consulted in Table 40.

### 4.3.2.2. Address Cycle Timings

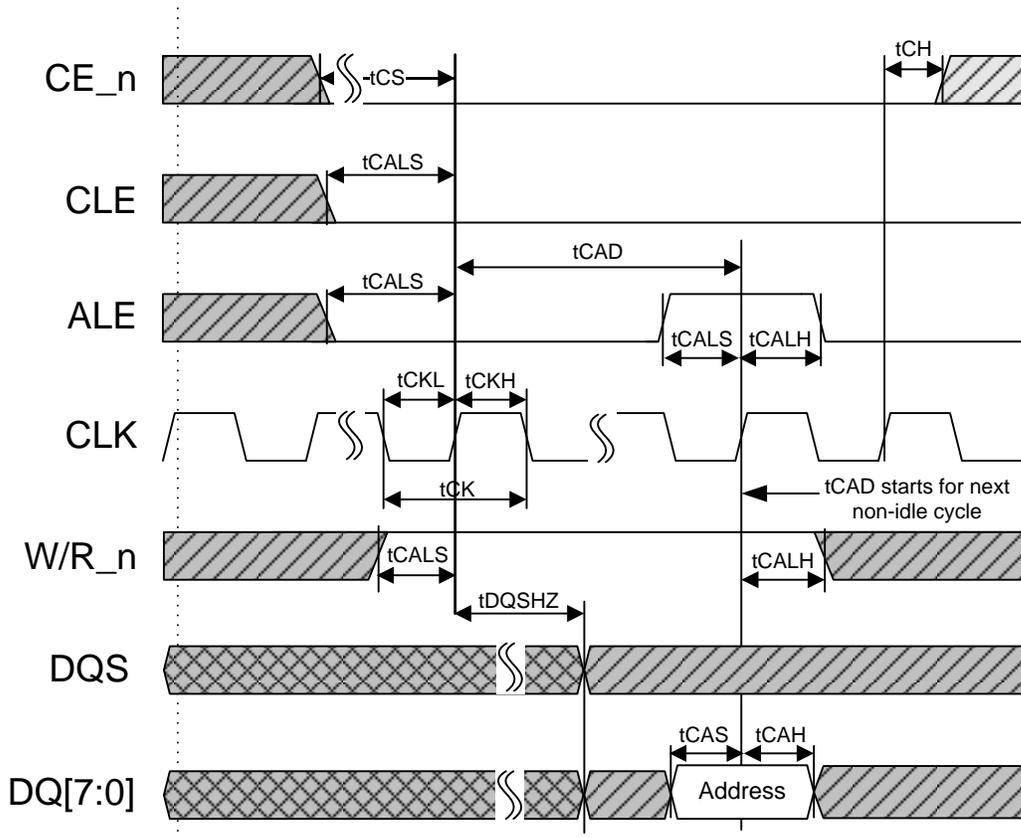


Figure 32 Address cycle timings

**NOTE:**

1. ALE and CLE shall be in a valid state when CE\_n transitions from one to zero. In the diagram, it appears that  $t_{CS}$  and  $t_{CALS}$  are equivalent times. However,  $t_{CS}$  and  $t_{CALS}$  values are not the same, the timing parameter values should be consulted in Table 40.

#### **4.3.2.3. Data Input Cycle Timings**

Data input cycle timing describes timing for data transfers from the host to the device (i.e. data writes).

For the Set Features command, the same data byte is repeated twice. The data pattern in this case is  $D_0 D_0 D_1 D_1 D_2 D_2$  etc. The device shall only latch one copy of each data byte. CLK should not be stopped during data input for the Set Features command. The device is not required to wait for the repeated data byte before beginning internal actions.

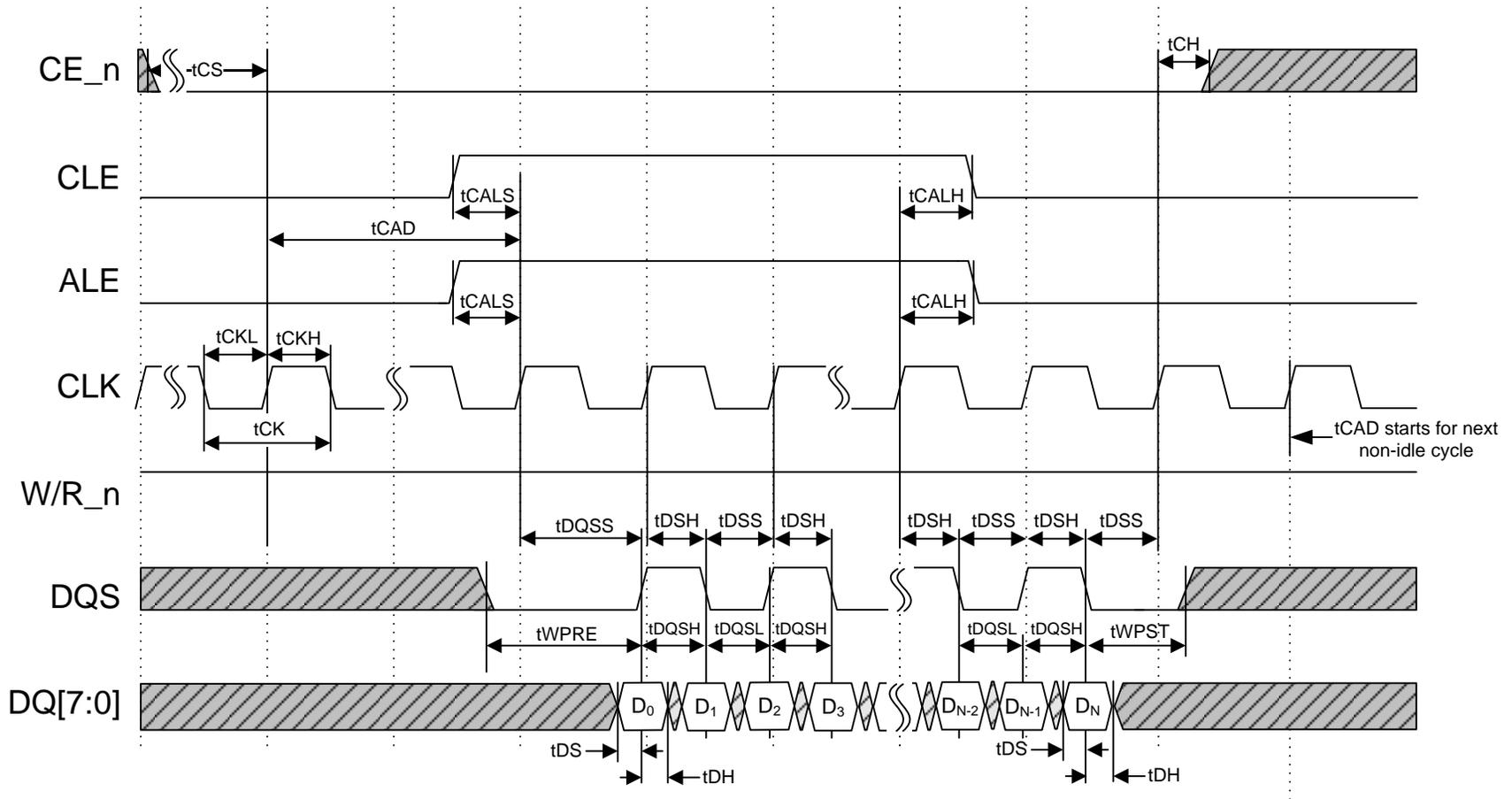


Figure 33 Data input cycle timing

#### **4.3.2.4. Data Input Cycle Timings, CLK stopped**

The host may save power during the data input cycles by holding the CLK signal high (i.e. stopping the CLK). The host may only stop the CLK during data input if the device supports this feature as indicated in the parameter page. Data input cycle timing describes timing for data transfers from the host to the device (i.e. data writes). Figure 34 describes data input cycling with the CLK signal stopped. The values of the ALE, CLE, and W/R\_n signals are latched on the rising edge of CLK and thus while CLK is held high these signals are don't care.

Figure 35 shows data input cycling with the CLK signal stopped where the host has optionally paused data input. The host may pause data input if it observes the tDPZ timing parameter for re-starting data input to the device. When re-starting the CLK, the host shall observe the indicated timing parameters in Figure 34 and Figure 35, which include tDSS and tDSH.

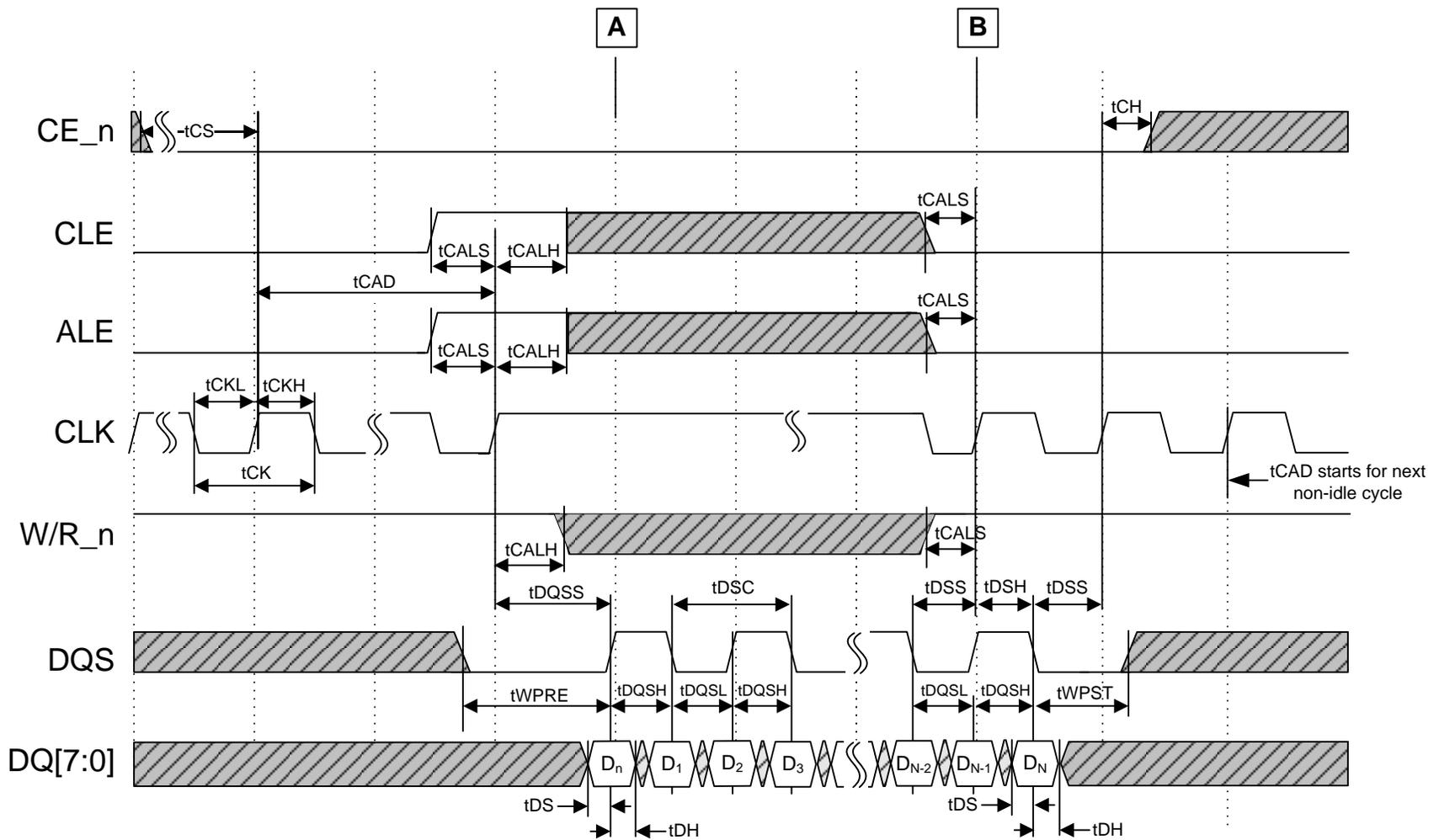
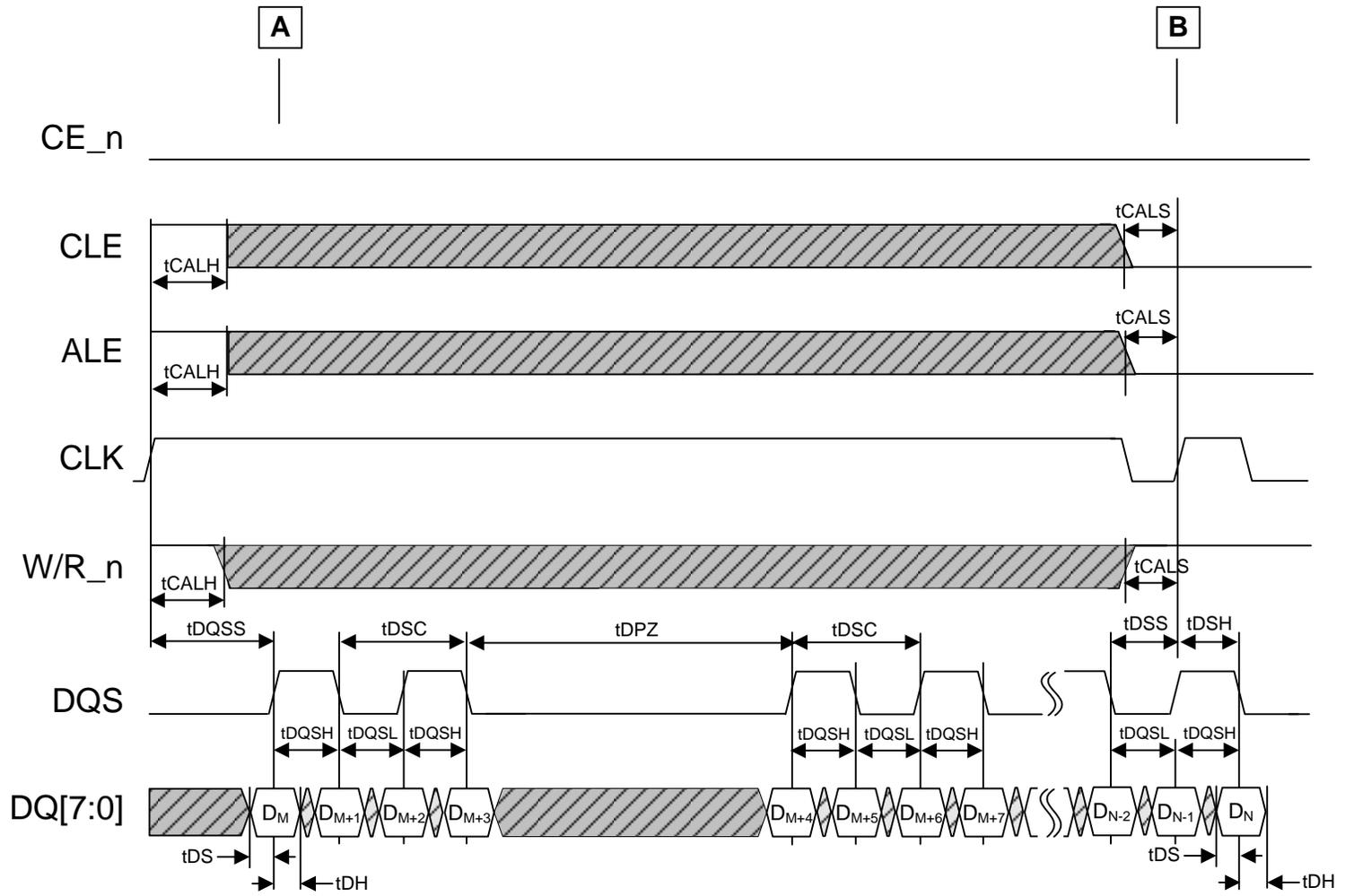


Figure 34 Data input cycle timing, CLK stopped



**Figure 35 Data input cycle timing, CLK stopped with data pause**

#### 4.3.2.5. Data Output Cycle Timings

Data output cycle timing describes timing for data transfers from the device to the host (i.e. data reads). The host shall not start data output (i.e. transition ALE/CLE to 11b) until the tDQSD time has elapsed.

For the Read ID, Get Features, Read Status, and Read Status Enhanced commands, the same data byte is repeated twice. The data pattern in this case is D<sub>0</sub> D<sub>0</sub> D<sub>1</sub> D<sub>1</sub> D<sub>2</sub> D<sub>2</sub> etc. The host shall only latch one copy of each data byte.

A calculated parameter, tCKWR, indicates when W/R\_n may be transitioned from a zero to one. This parameter is calculated as:

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

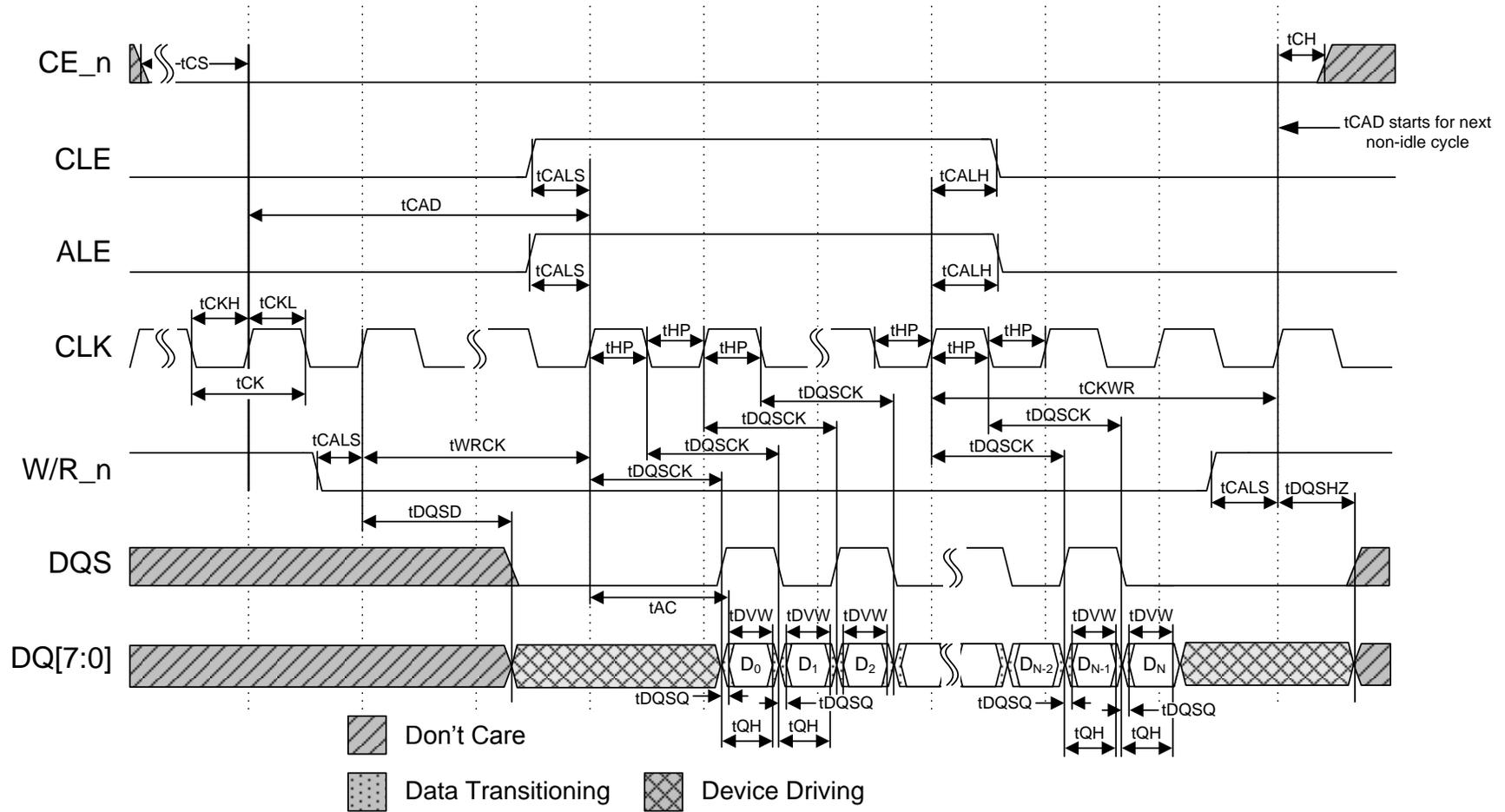


Figure 36 Data output cycle timing

#### 4.3.2.6. W/R\_n Behavior Timings

Figure 37 describes the ownership transition of the DQ bus and DQS signal. The host owns the DQ bus and DQS signal when W/R\_n is one. The device owns the DQ bus and DQS signal when W/R\_n is zero. The host shall tri-state the DQ bus and DQS signal whenever W/R\_n is zero.

When W/R\_n transitions from one to zero, the bus ownership is assumed by the device. The device shall start driving the DQS signal low within  $t_{DQSD}$  after the transition of W/R\_n to zero. When W/R\_n transitions from zero to one, the bus ownership is assumed by the host. The device shall tri-state the DQ bus and DQS signal within  $t_{DQSHZ}$  after the transition of W/R\_n to one.

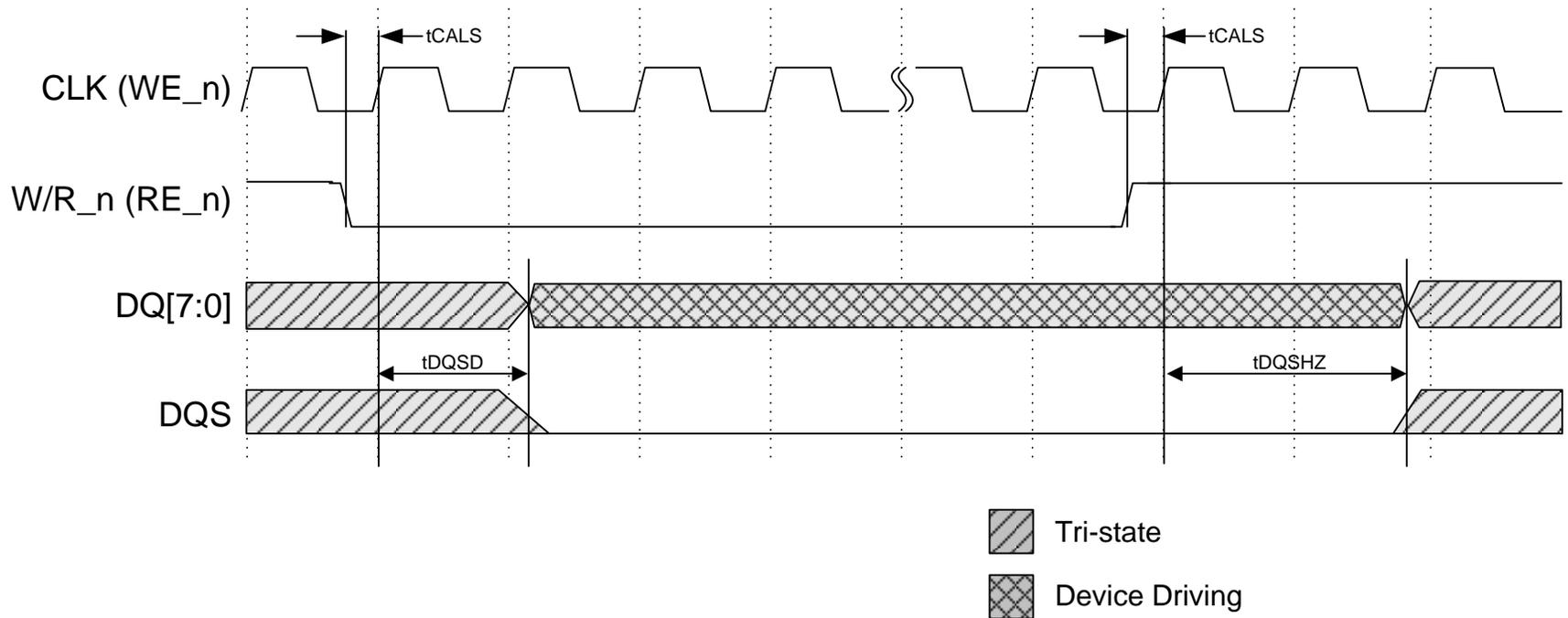
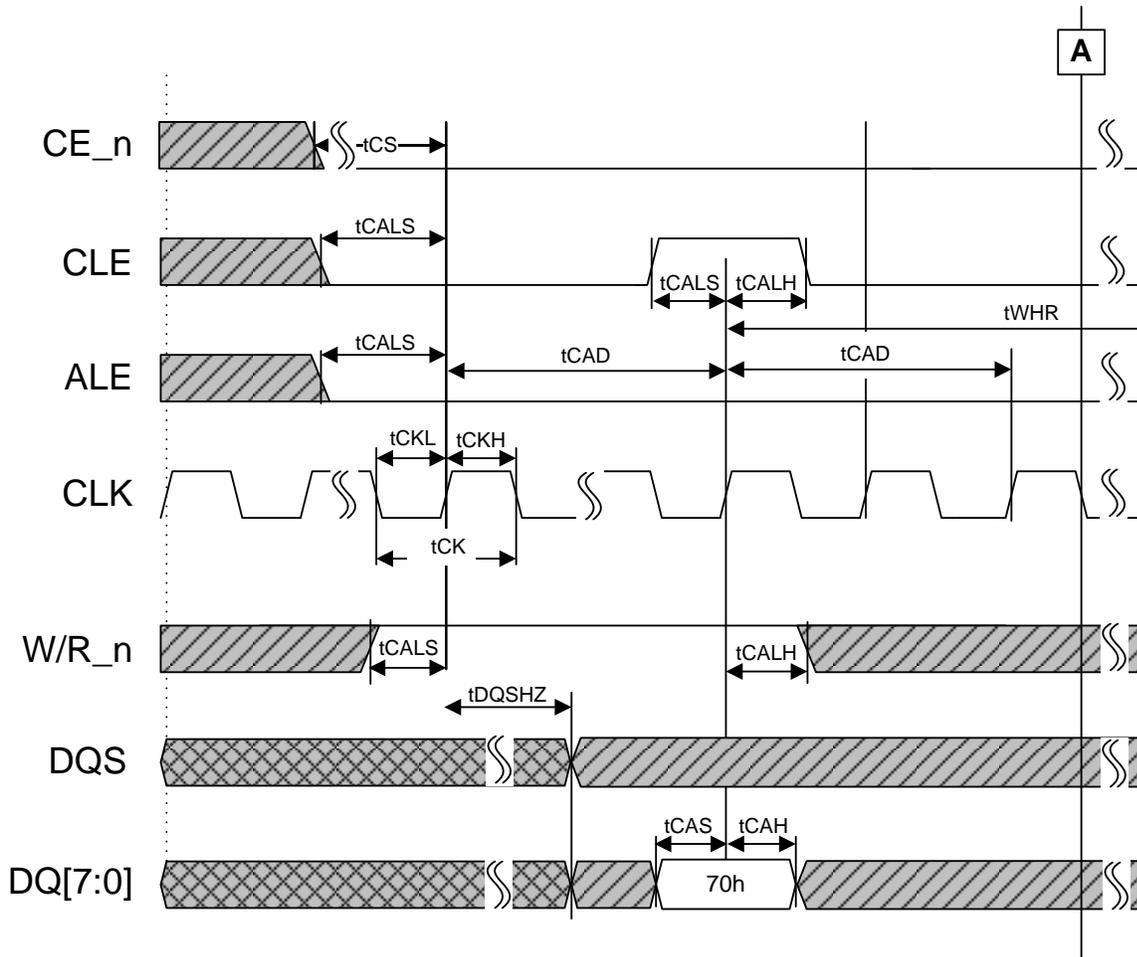


Figure 37 W/R\_n timing

### 4.3.2.7. Satisfying Timing Requirements

In some cases there are multiple timing parameters that shall be satisfied prior to the next phase of a command operation. For example, both  $t_{DQSD}$  and  $t_{CCS}$  shall be satisfied prior to data output commencing for the Change Write Column command. The host and device shall ensure all timing parameters are satisfied. In cases where  $t_{ADL}$ ,  $t_{CCS}$ ,  $t_{RHW}$ , or  $t_{WHR}$  are required, then these are the governing parameters (i.e. these parameters are the longest times).

Figure 38 and Figure 39 show an example of a Read Status command that includes all the timing parameters for both the command and data phase of the operation. It may be observed that  $t_{WHR}$  is the governing parameter prior to the data transfer starting. Also note that the same data byte is transmitted twice ( $D_0$ ,  $D_0$ ) for the Read Status command.



**Figure 38** Read Status including  $t_{WHR}$  and  $t_{CAD}$  timing requirements

**NOTE:**

1.  $ALE$  and  $CLE$  shall be in a valid state when  $CE_n$  transitions from one to zero. In the diagram, it appears that  $t_{CS}$  and  $t_{CALS}$  are equivalent times. However,  $t_{CS}$  and  $t_{CALS}$  values are not the same, the timing parameter values should be consulted in Table 40.

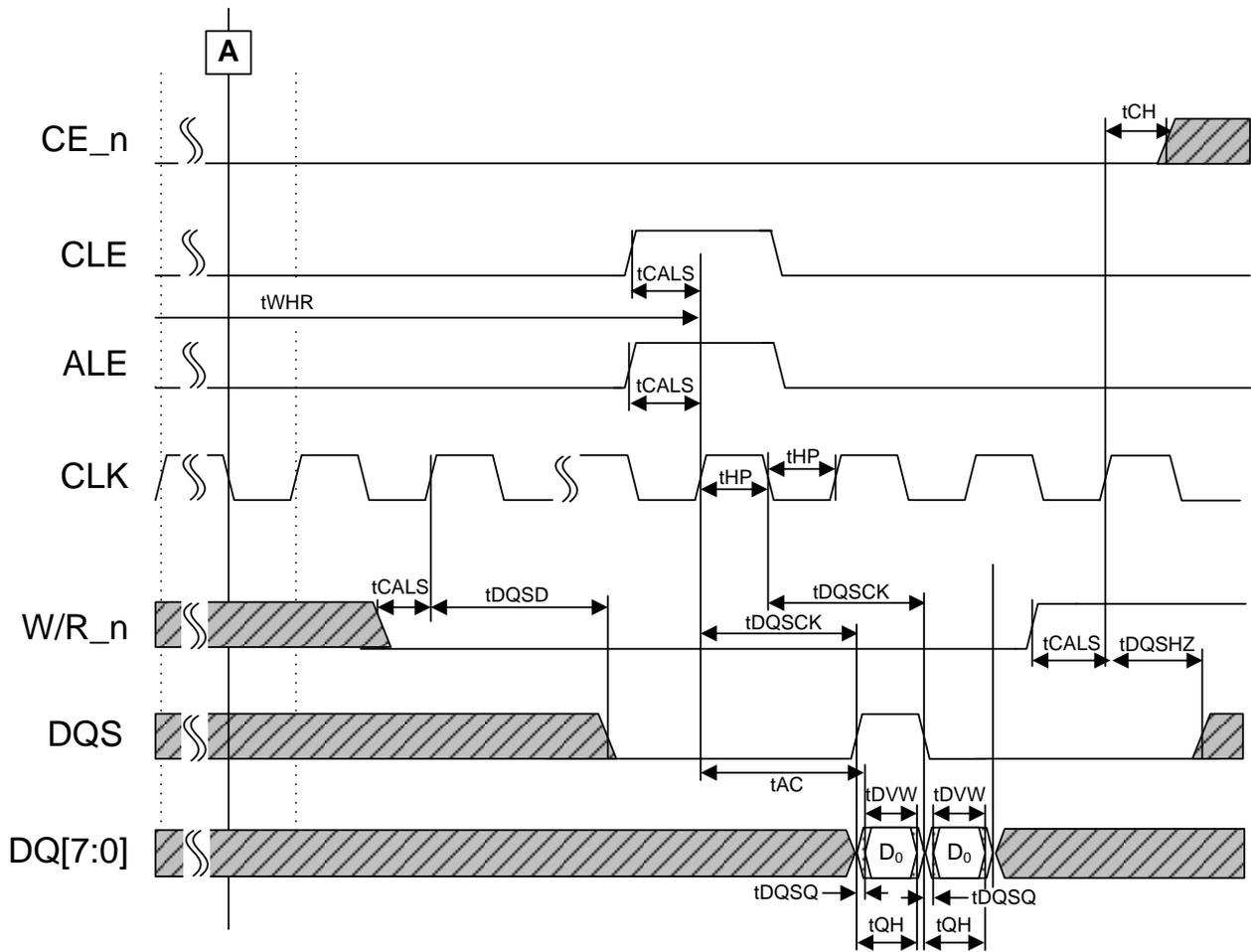
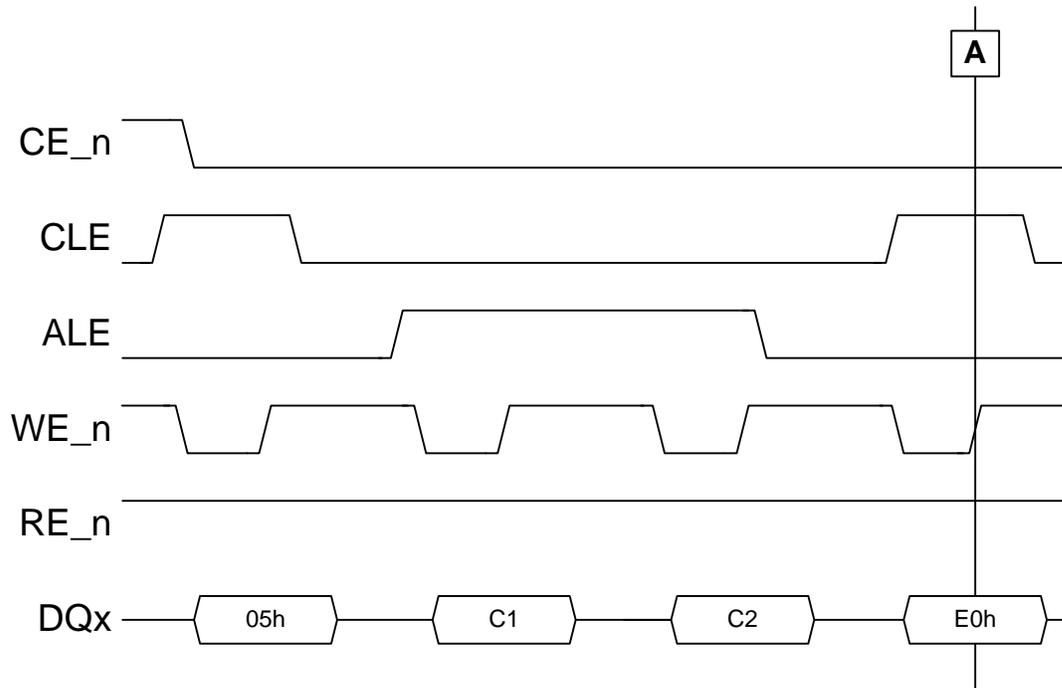


Figure 39 Read Status including  $t_{WHR}$  and  $t_{CAD}$  timing requirements, continued

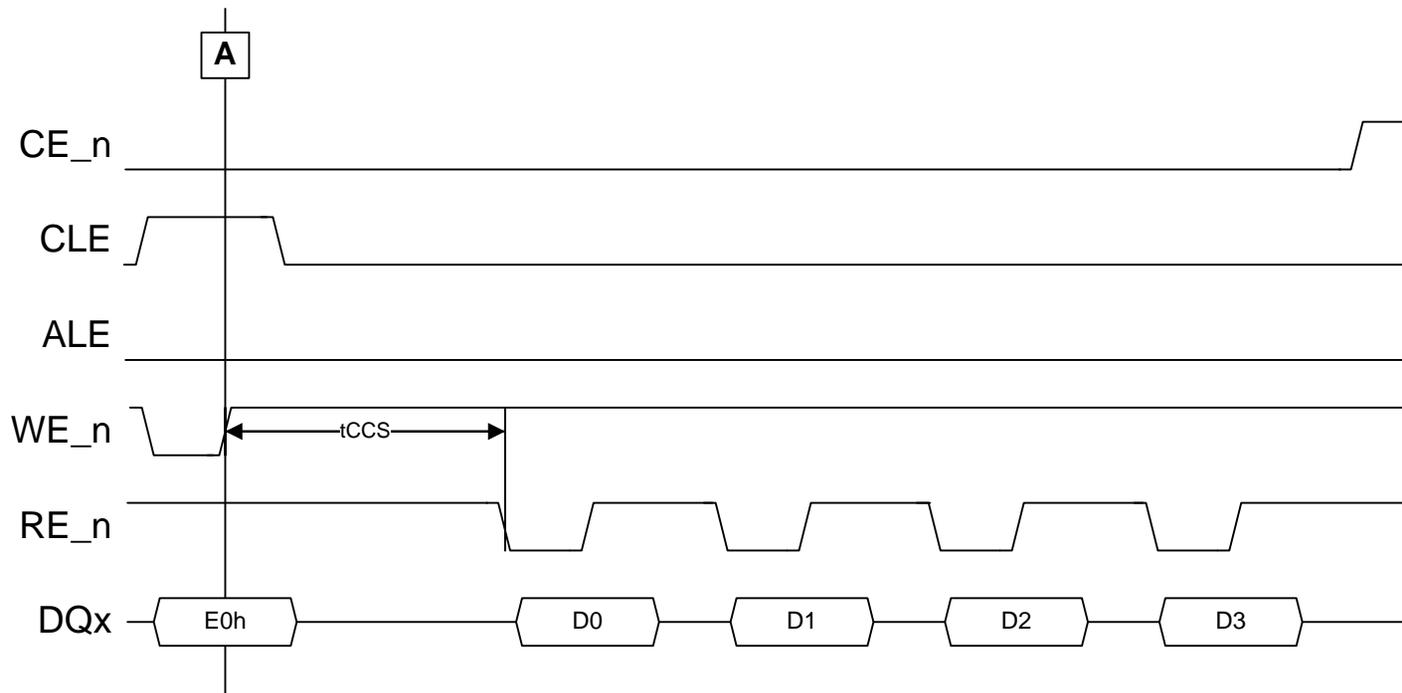
## 4.4. Command Examples

### 4.4.1. Asynchronous

This section shows examples of commands using the asynchronous data interface. Figure 40 and Figure 41 show an example of Change Read Column.

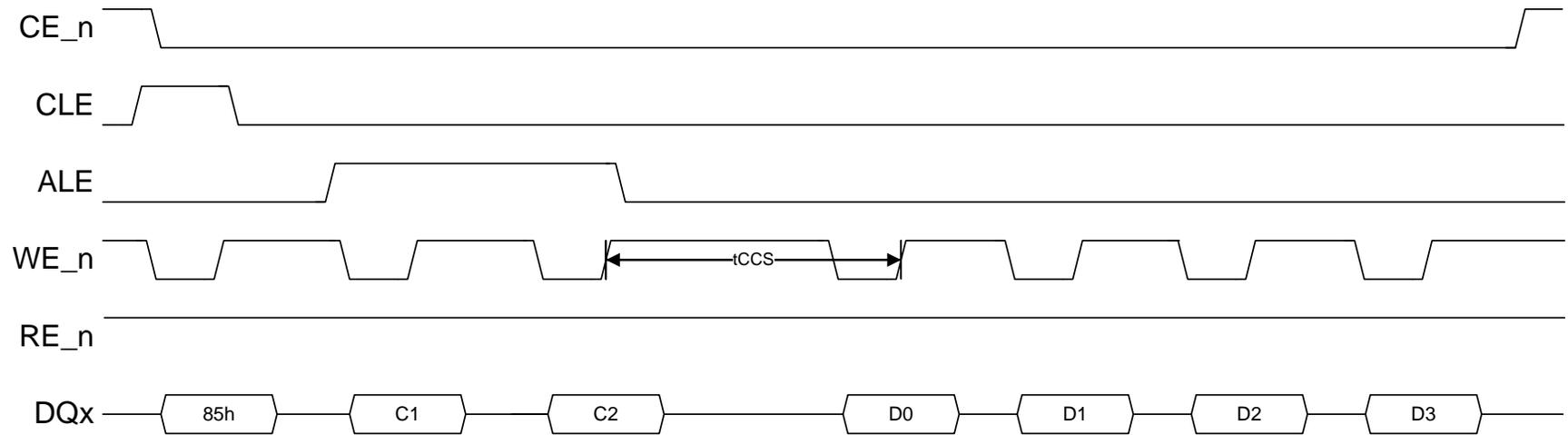


**Figure 40** Asynchronous Data Interface: Change Read Column example



**Figure 41 Asynchronous Data Interface: Change Read Column example, continued**

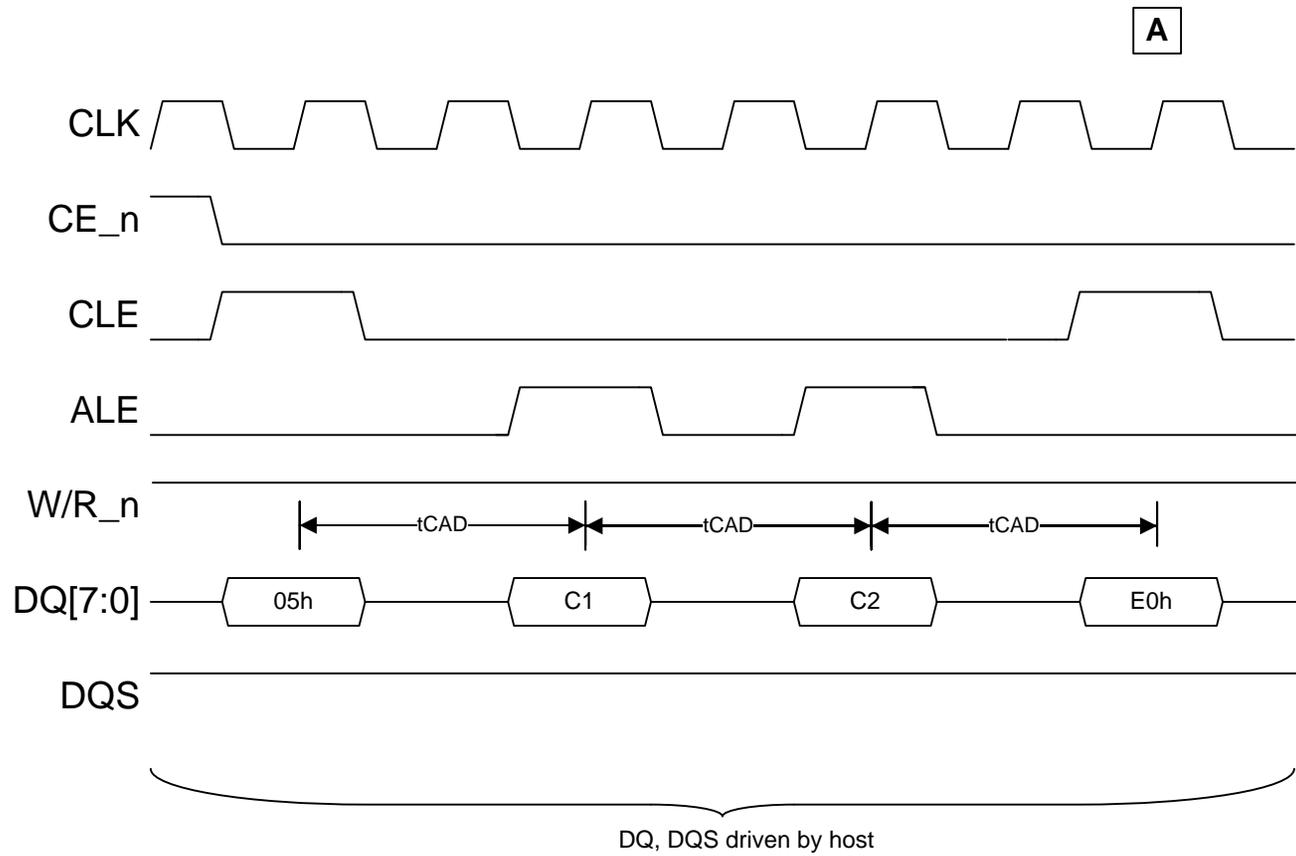
Figure 42 shows an example of Change Write Column.



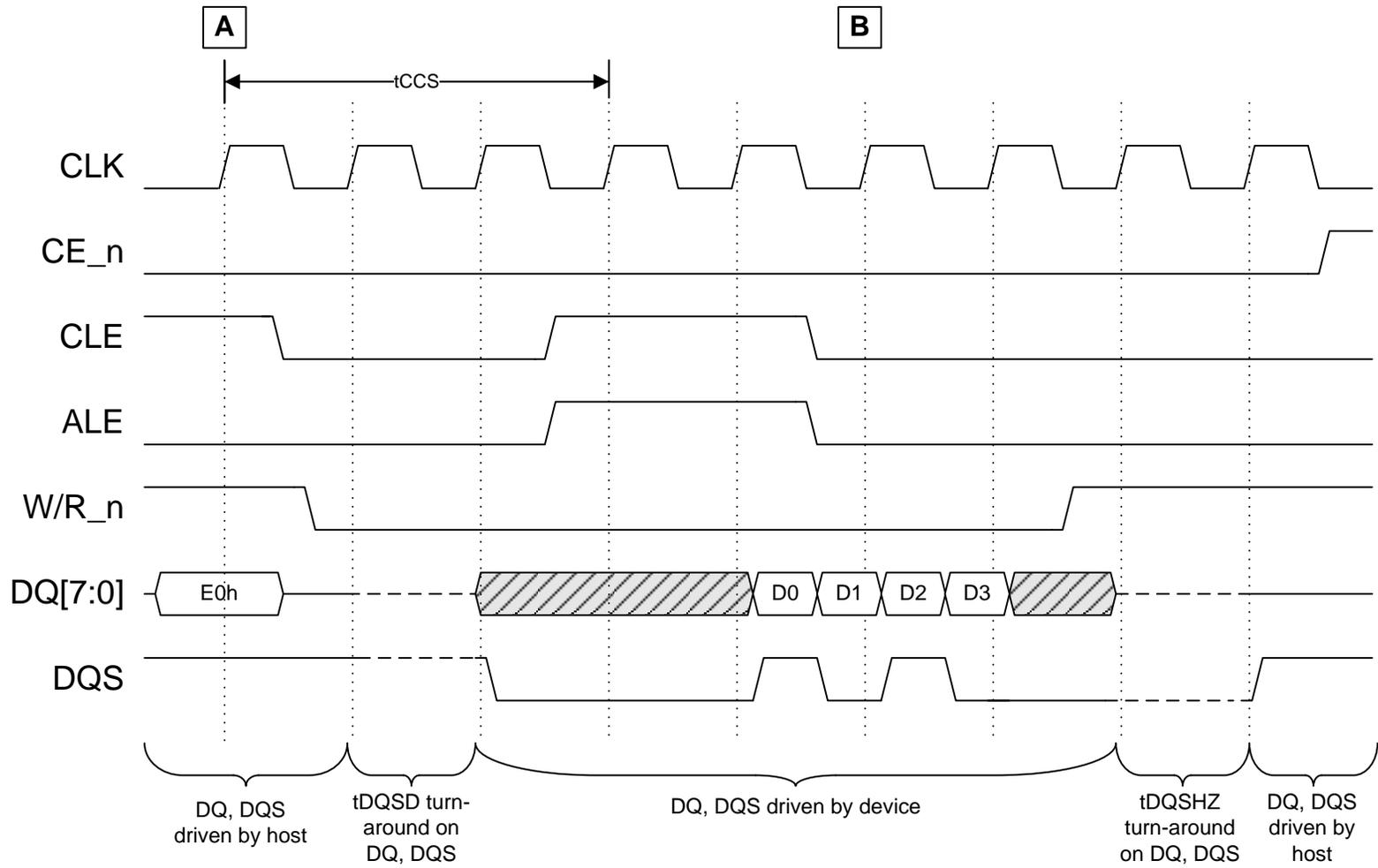
**Figure 42 Asynchronous Data Interface: Change Write Column example**

#### **4.4.2. Source Synchronous**

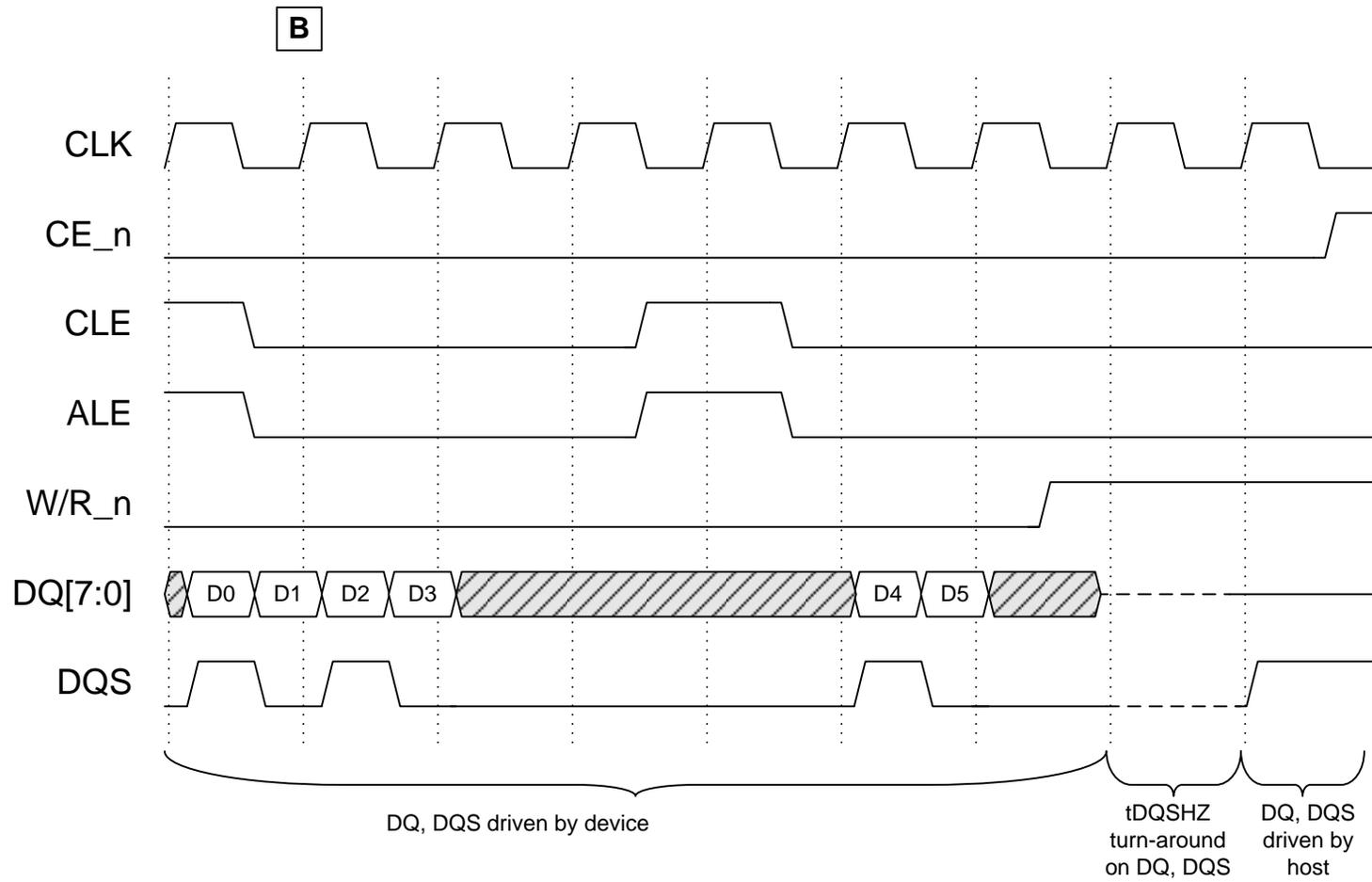
This section shows examples of commands using the source synchronous data interface. Figure 43 through Figure 45 show an example of Change Read Column. Figure 45 shows a continuation of data transfer, after the host stops the transfer for a period of time.



**Figure 43** Source Synchronous Data Interface: Change Read Column example, command issue

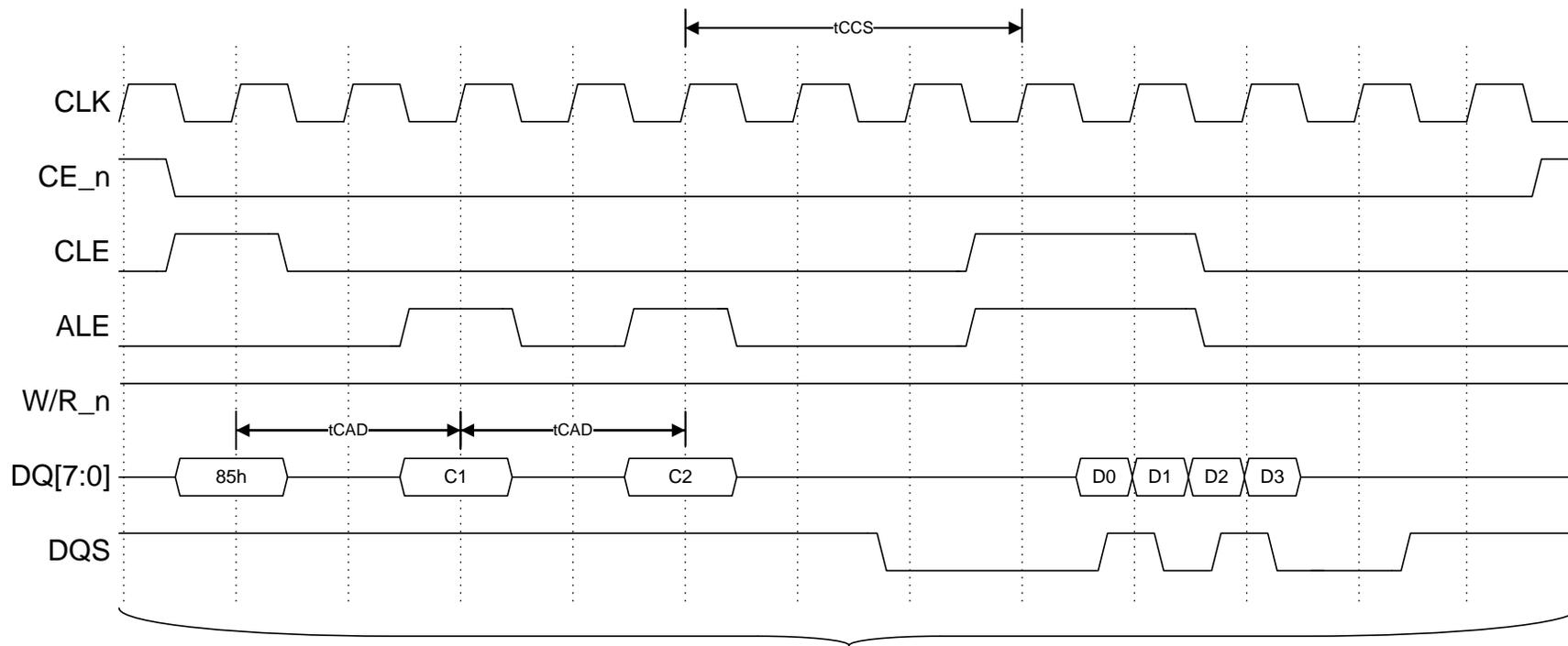


**Figure 44 Source Synchronous Data Interface: Change Read Column example, data transfer**



**Figure 45 Source Synchronous Data Interface: Change Read Column example, data transfer continue**

Figure 46 shows an example of Change Write Column.



DQ, DQS driven by host

**Figure 46 Source Synchronous Data Interface: Change Write Column example**

## **5. Command Definition**

### **5.1. Command Set**

Table 41 outlines the ONFI command set.

The value specified in the first command cycle identifies the command to be performed. Some commands have a second command cycle as specified in Table 41. Typically, commands that have a second command cycle include an address.

Command	O/M	1 <sup>st</sup> Cycle	2 <sup>nd</sup> 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 <sup>2</sup>	O	85h	11h		Y	
Change Write Column <sup>1</sup>	M	85h			Y	
Change Row Address <sup>1</sup>	O	85h			Y	
Read ID	M	90h				Y
Read Parameter Page	M	ECh				Y
Read Unique ID	O	EDh				Y
Get Features	O	EEh				Y
Set Features	O	EFh				Y
Reset LUN	O	FAh		Y	Y	
Synchronous Reset	O	FCh		Y	Y	Y
Reset	M	FFh		Y	Y	Y

NOTE:

1. Change Write Column specifies the column address only. Change Row Address specifies the row address and the column address. Refer to the specific command definitions.
2. Small Data Move's first opcode may be 80h if the operation is a program only with no data output. For the last second cycle of a Small Data Move, it is a 10h command to confirm the Program or Copyback operation.

**Table 41 Command set**

Reserved opcodes shall not be used by the device, as the ONFI specification may define the use of these opcodes in a future revision. Vendor specific opcodes may be used at the discretion of the vendor and shall never be defined for standard use by ONFI. Future Standardization opcodes are those opcodes already being used commonly in the industry and may be defined for standard use by ONFI for those same purposes. Future Standardization opcodes may be used by compliant ONFI implementations with the common industry usage. Block abstracted NAND opcodes are opcodes used in a BA NAND implementation.

Type	Opcode
Vendor Specific	02h – 04h, 08h, 16h – 17h, 19h, 1Dh, 20h – 22h, 25h – 29h, 2Bh, 2Dh – 2Fh, 33h, 36h – 39h, 3B – 3Eh, 40h – 41h, 48h, 4Ch, 53h – 55h, 68h, 72h – 75h, 84h, 87h – 89h, 91h – BFh, CFh, F1-F4h
Future Standardization	23h – 24h, 2Ah, 2Ch, 34h, 3Ah, 65h, 71h, 79h – 7Bh, 81h, 8Ch
Block Abstracted NAND	C0h - CEh
Reserved	01h, 07h, 09h – 0Fh, 12h-14h, 18h, 1Ah – 1Ch, 1Eh – 1Fh, 42h – 47h, 49h – 4Bh, 4Dh – 52h, 56h – 5Fh, 62h – 64h, 66h – 67h, 69h – 6Fh, 76h – 77h, 7Ch – 7Fh, 82h – 83h, 86h, 8Ah – 8Bh, 8Dh – 8Fh, D2h – DFh, E1h – EBh, F0h, F5h – F9h, FBh, FDh – FEh

**Table 42      Opcode Reservations**

## 5.2. Command Descriptions

The command descriptions in section 5 are shown in a format that is agnostic to the data interface being used (when the command may be used in either data interface). An example of the agnostic command description for Change Write Column is shown in Figure 47. The agnostic command examples shall be translated to a command description for the particular data interface selected. The command description for Change Write Column in the asynchronous data interface is shown in Figure 48. The command description for Change Write Column in the source synchronous data interface is shown in Figure 49. Note that the timing parameters defined in section 4 shall be observed for each command (e.g. the tCAD timing parameter for the source synchronous data interface).

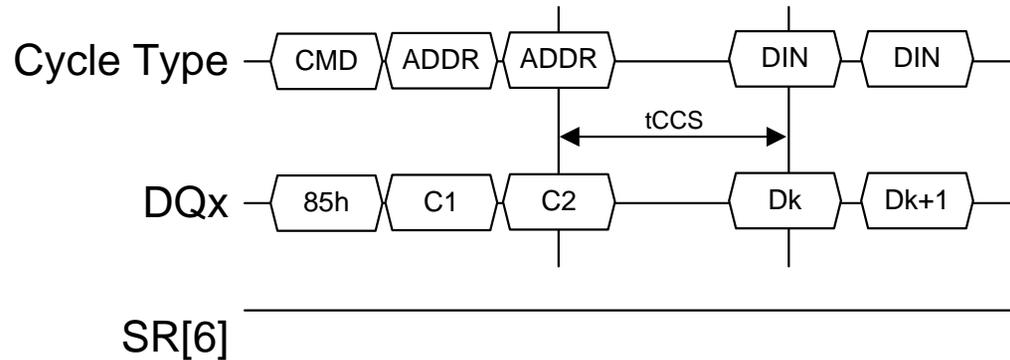
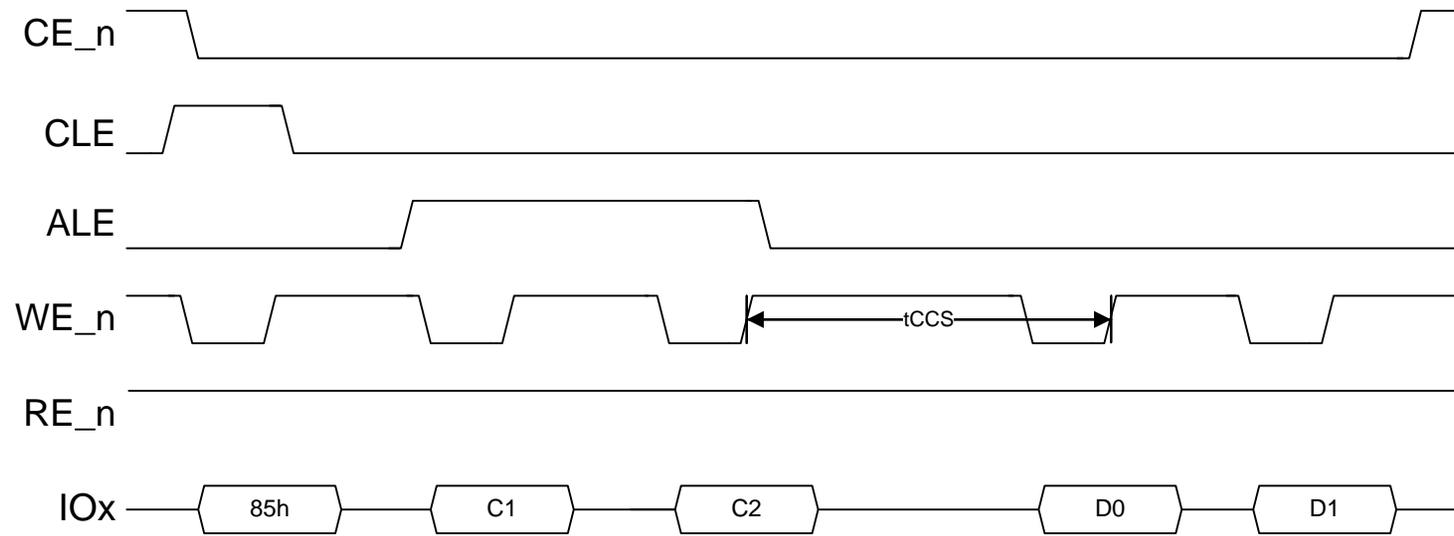
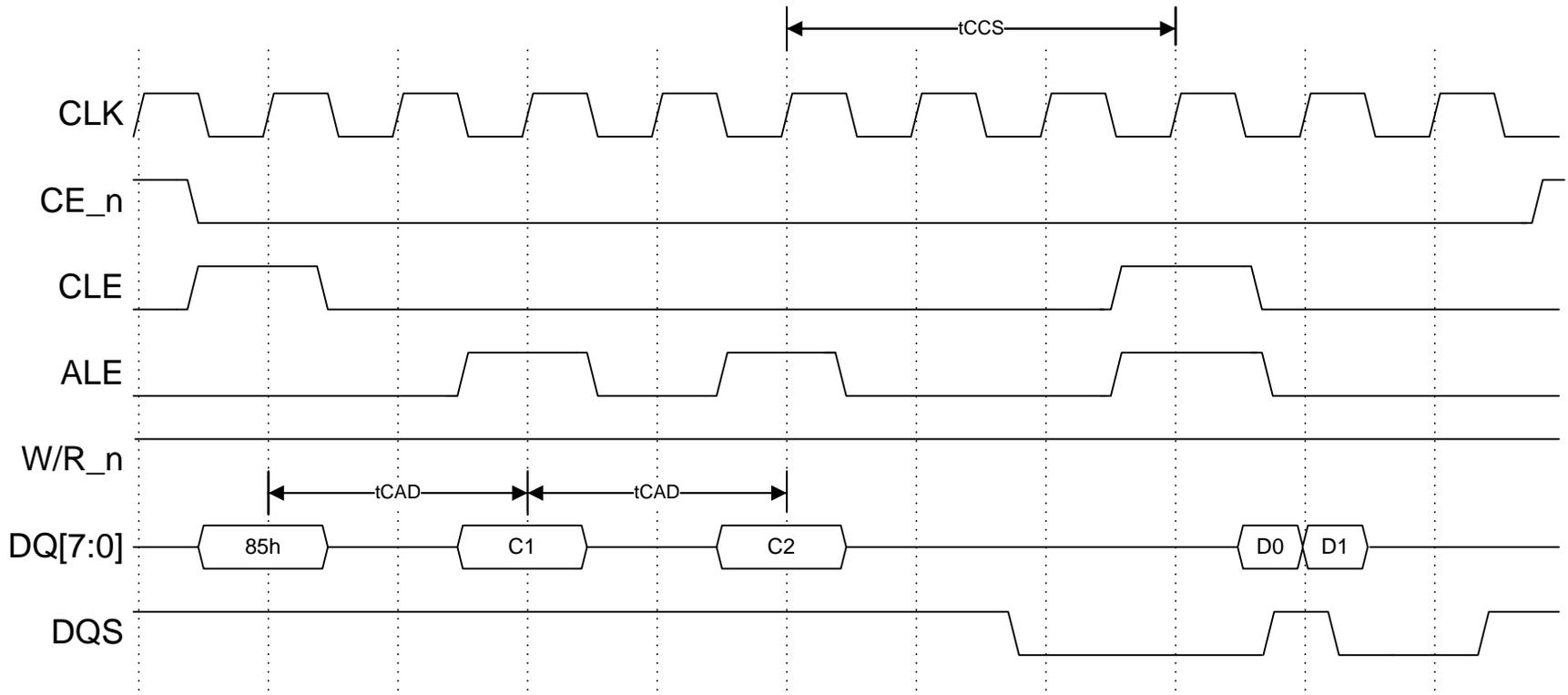


Figure 47 Agnostic command description



**Figure 48 Asynchronous data interface command description**



**Figure 49 Source synchronous data interface command description**

### 5.3. Reset Definition

The Reset function puts the target in its default power-up state and places the target in the asynchronous data interface. This command shall only be issued when the host is configured to the asynchronous data interface. The device shall also recognize and execute the Reset command when it is configured to the source synchronous data interface. The R/B\_n value is unknown when Reset is issued; R/B\_n is guaranteed to be low tWB after the Reset is issued.

Note that some feature settings are retained across Reset commands (as specified in section 5.26). As part of the Reset command, all LUNs are also reset. The command may be executed with the target in any state, except during power-on when Reset shall not be issued until R/B\_n is set to one. Figure 50 defines the Reset behavior and timings.

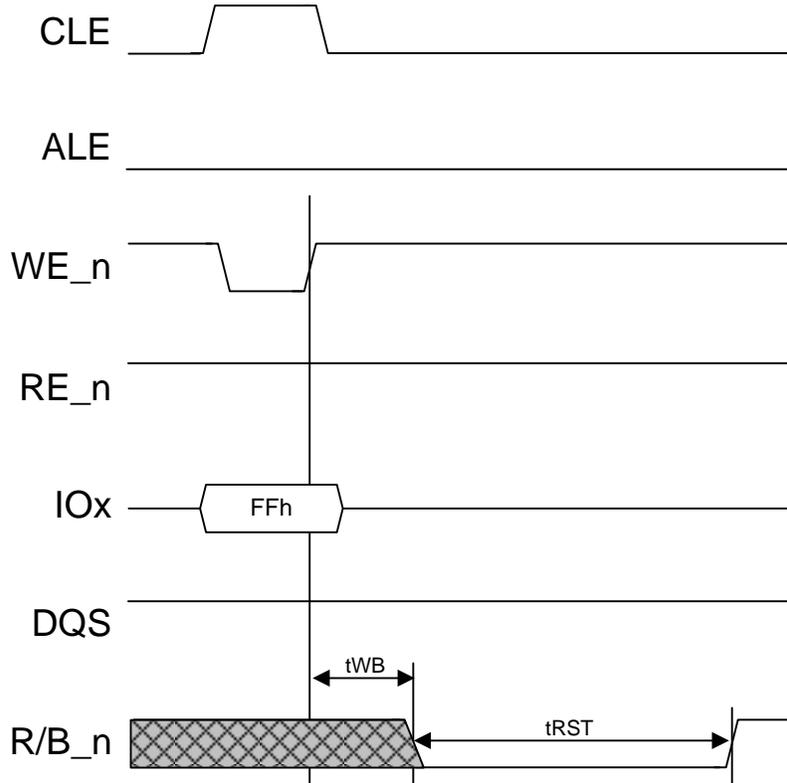


Figure 50 Reset timing diagram

### 5.4. Synchronous Reset Definition

The Synchronous Reset command resets the target and all LUNs. The command may be executed with the target in any state. Figure 51 defines the Synchronous Reset behavior and

timings. The R/B\_n value is unknown when Synchronous Reset is issued; R/B\_n is guaranteed to be low  $t_{WB}$  after the Synchronous Reset is issued.

This command shall be supported by devices that support the source synchronous data interface. This command is only accepted in source synchronous operation. The host should not issue this command when the device is configured to the asynchronous data interface. The target shall remain in the source synchronous data interface following this command.

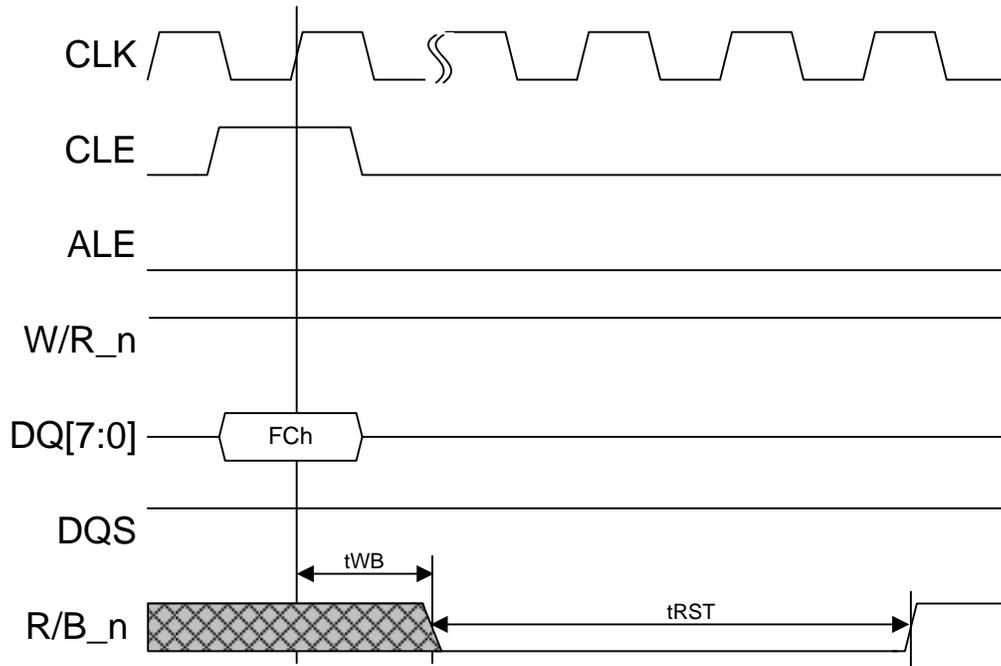
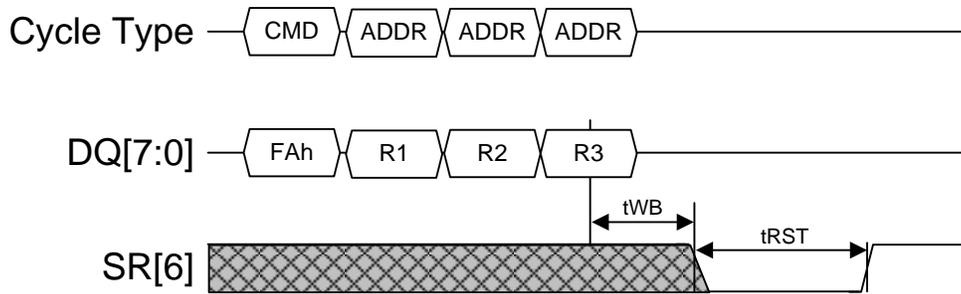


Figure 51 Synchronous Reset timing diagram

## 5.5. Reset LUN Definition

The Reset LUN command is used to reset a particular LUN. This command is accepted by only the LUN addressed as part of the command. The command may be executed with the LUN in any state. Figure 52 defines the Reset LUN behavior and timings. The SR[6] value is unknown when Reset LUN is issued; SR[6] is guaranteed to be low  $t_{WB}$  after the Reset LUN command is issued. This command does not affect the data interface configuration for the target.

Reset LUN should be used to cancel ongoing command operations, if desired. When there are issues with the target, e.g. a hang condition, the Reset (FFh) or Synchronous Reset (FCh) commands should be used.



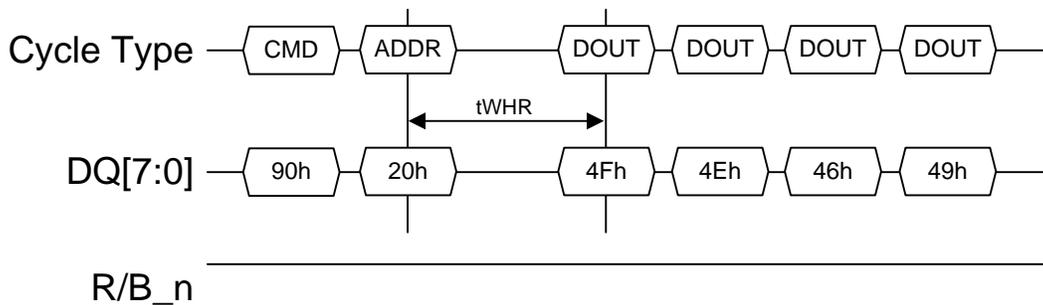
**Figure 52** Reset LUN timing diagram

### 5.6. Read ID Definition

The Read ID function identifies that the target supports the ONFI specification. If the target supports the ONFI specification, then the ONFI signature shall be returned. The ONFI signature is the ASCII encoding of 'ONFI' where 'O' = 4Fh, 'N' = 4Eh, 'F' = 46h, and 'I' = 49h. Reading beyond four bytes yields indeterminate values. Figure 53 defines the Read ID behavior and timings.

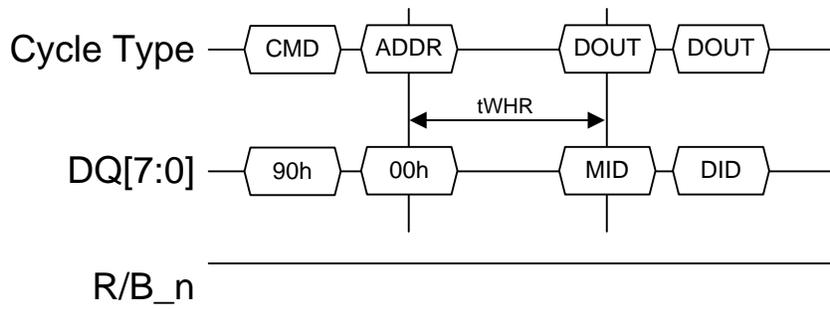
When issuing Read ID in the source synchronous data interface, each data byte is received twice. The host shall only latch one copy of each data byte. See section 4.3.2.5.

For the Read ID command, only addresses of 00h and 20h are valid. To retrieve the ONFI signature an address of 20h shall be entered (i.e. it is not valid to enter an address of 00h and read 36 bytes to get the ONFI signature).



**Figure 53** Read ID timing diagram for ONFI signature

The Read ID function can also be used to determine the JEDEC manufacturer ID and the device ID for the particular NAND part by specifying an address of 00h. Figure 54 defines the Read ID behavior and timings for retrieving the JEDEC manufacturer ID and device ID. Reading beyond the first two bytes yields values as specified by the manufacturer.

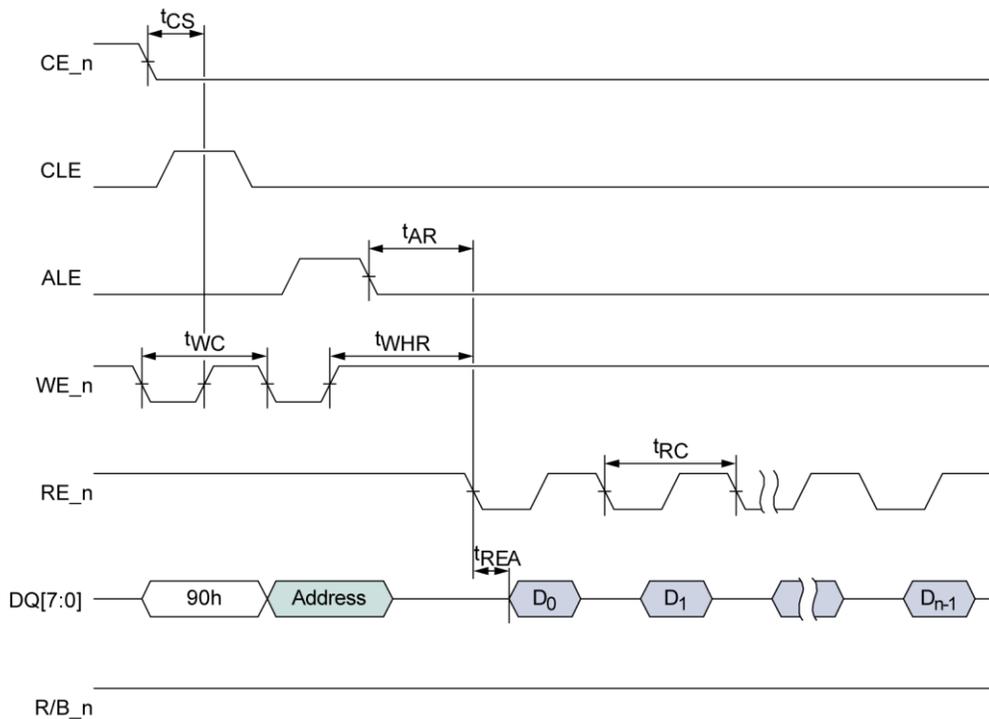


**Figure 54 Read ID timing diagram for manufacturer ID**

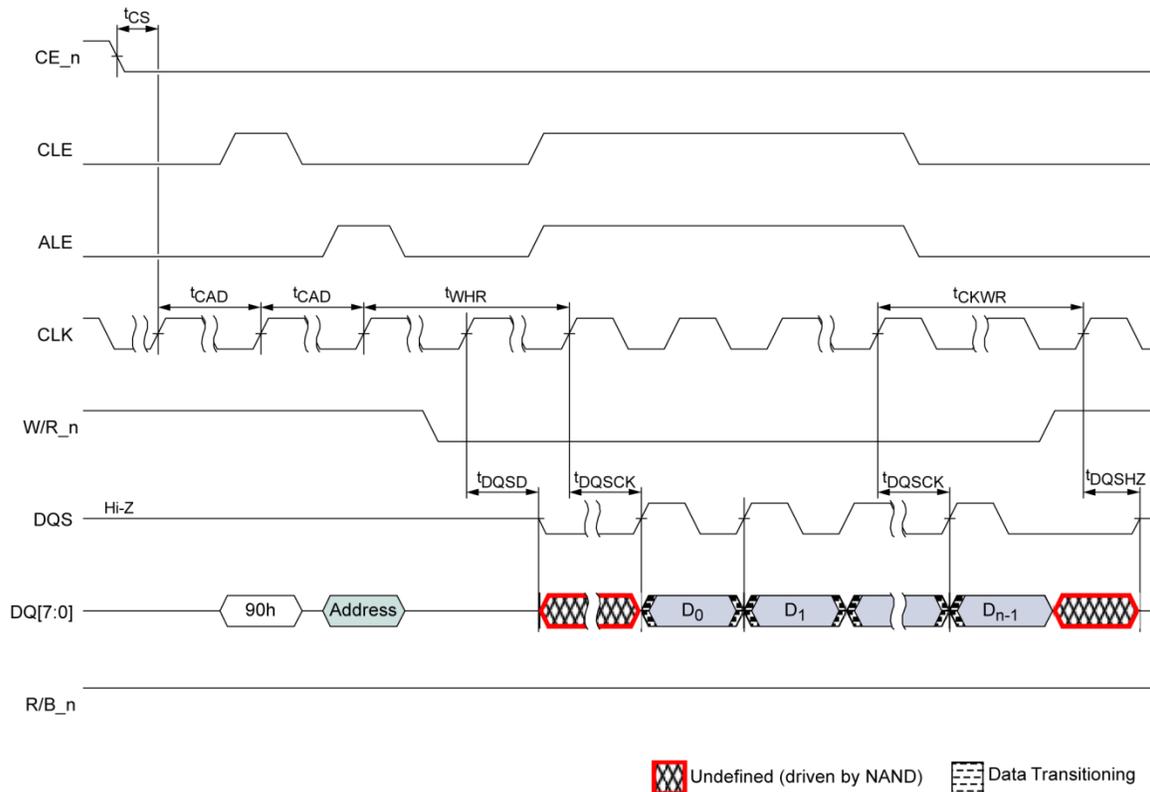
MID Manufacturer ID for manufacturer of the part, assigned by JEDEC.

DID Device ID for the part, assigned by the manufacturer.

The Read ID command may be issued using either the asynchronous or source synchronous data interfaces. The timing parameters for each data interface are shown in Figure 55 and Figure 56.



**Figure 55 Read ID command using asynchronous data interface**



**Figure 56 Read ID command using source synchronous data interface**

**NOTE:** The data bytes in Figure 56 are repeated twice (on the rising and falling edge of CLK).

## 5.7. Read Parameter Page Definition

The Read Parameter Page function retrieves the data structure that describes the target's organization, features, timings and other behavioral parameters. There may also be additional information provided in an extended parameter page. Figure 57 defines the Read Parameter Page behavior.

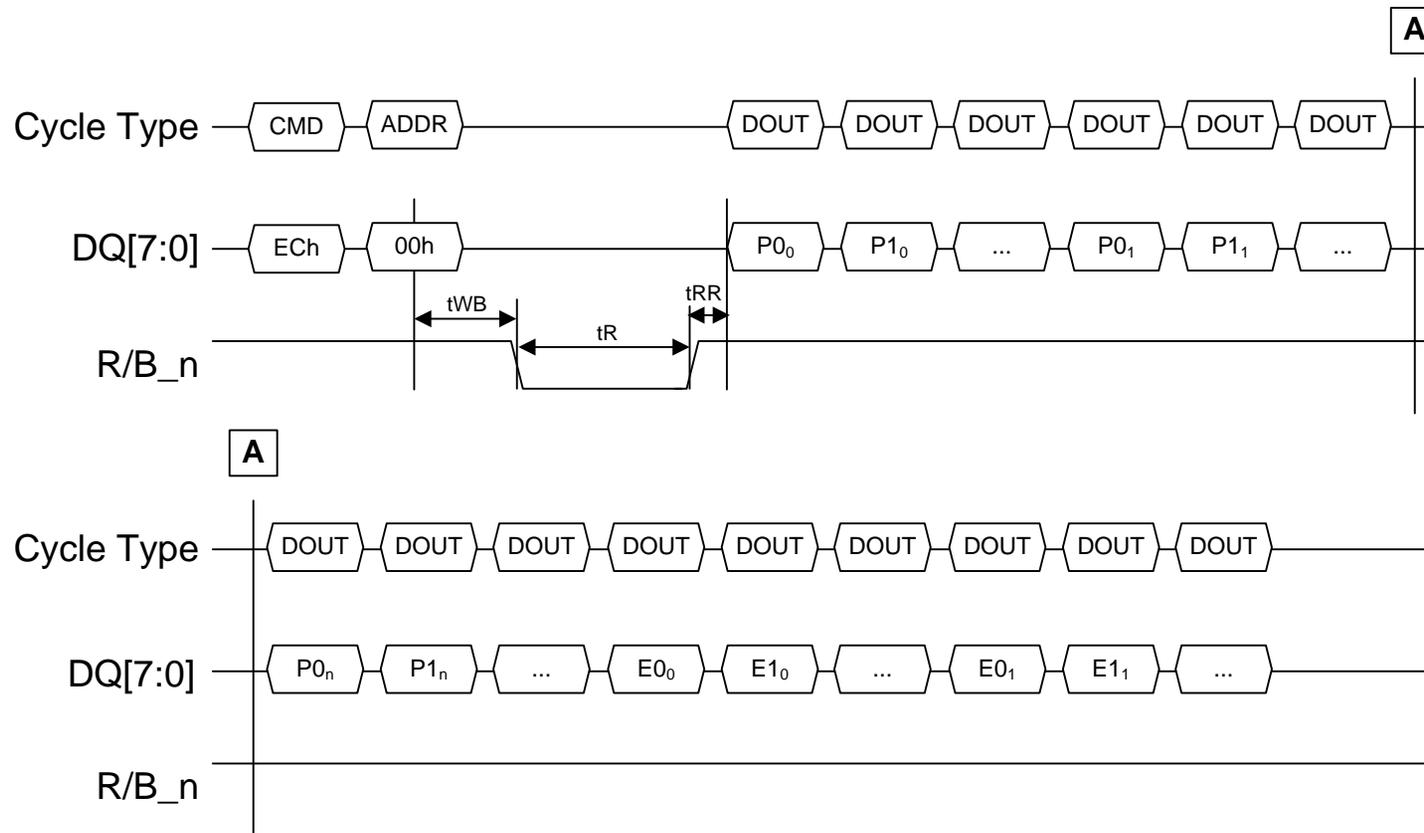
Values in the parameter page are static and shall not change. The host is not required to read the parameter page after power management events.

The first time the host executes the Read Parameter Page command after power-on, timing mode 0 shall be used. If the host determines that the target supports more advanced timing modes, those supported timing modes may be used for subsequent execution of the Read Parameter Page command.

The Change Read Column command may be issued following execution of the Read Parameter Page to read specific portions of the parameter page.

Read Status may be used to check the status of Read Parameter Page during execution. After completion of the Read Status command, 00h shall be issued by the host on the command line to continue with the data output flow for the Read Parameter Page command.

Read Status Enhanced and Change Read Column Enhanced shall not be used during execution of the Read Parameter Page command.



**Figure 57 Read Parameter Page command timing**

$P0_k$ - $Pn_k$  The kth copy of the parameter page data structure. See section 5.7.1. Reading bytes beyond the end of the final parameter page copy (or beyond the final extended parameter page copy if supported) returns indeterminate values.

$E0_k$ - $En_k$  The kth copy of the extended parameter page data structure. See section 5.7.2. Reading bytes beyond the end of the final extended parameter page copy returns indeterminate values. This field is only present when the extended parameter page is supported, as indicated in the Features supported field of the parameter page.

### 5.7.1. Parameter Page Data Structure Definition

Table 43 defines the parameter page data structure. For parameters that span multiple bytes, the least significant byte of the parameter corresponds to the first byte. See section 1.4.2.3 for more information on the representation of word and Dword values.

Values are reported in the parameter page in units of bytes when referring to items related to the size of data access (as in an 8-bit data access device). For example, the target will return how many data *bytes* are in a page. For a device that supports 16-bit data access, the host is required to convert byte values to word values for its use.

Unused fields should be cleared to 0h by the target.

Byte	O/M	Description
<b>Revision information and features block</b>		
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 6-15 Reserved (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 10-15 Reserved (0) 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 source synchronous 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 10-15 Reserved (0) 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

Byte	O/M	Description
		0 1 = supports Page Cache Program command
10-11		Reserved (0)
12-13	O	Extended parameter page length
14	O	Number of parameter pages
15-31		Reserved (0)
<b>Manufacturer information block</b>		
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)
<b>Memory organization block</b>		
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
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)
<b>Electrical parameters block</b>		
128	M	I/O pin capacitance, maximum

Byte	O/M	Description
129-130	M	Asynchronous 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 – Asynchronous program cache timing mode support
133-134	M	$t_{\text{PROG}}$ Maximum page program time ( $\mu\text{s}$ )
135-136	M	$t_{\text{BERS}}$ Maximum block erase time ( $\mu\text{s}$ )
137-138	M	$t_{\text{R}}$ Maximum page read time ( $\mu\text{s}$ )
139-140	M	$t_{\text{CCS}}$ Minimum change column setup time (ns)
141-142	O	Source synchronous 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
143	O	Source synchronous features 3-7 Reserved (0) 2 1 = device supports CLK stopped for data input 1 1 = typical capacitance values present 0 $t_{\text{CAD}}$ 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	$t_{\text{R}}$ Maximum multi-plane page read time ( $\mu\text{s}$ )
154-155	O	$t_{\text{ADL}}$ Program page register clear enhancement $t_{\text{ADL}}$ value (ns)
156-157	O	$t_{\text{R}}$ Typical page read time for EZ NAND ( $\mu\text{s}$ )
158-163		Reserved (0)
<b>Vendor block</b>		
164-165	M	Vendor specific Revision number
166-253		Vendor specific
254-255	M	Integrity CRC
<b>Redundant Parameter Pages</b>		
256-511	M	Value of bytes 0-255
512-767	M	Value of bytes 0-255
768+	O	Additional redundant parameter pages

**Table 43      Parameter page definitions**

**5.7.1.1. Byte 0-3: Parameter page signature**

This field contains the parameter page signature. When two or more bytes of the signature are valid, then it denotes that a valid copy of the parameter page is present.

Byte 0 shall be set to 4Fh.

Byte 1 shall be set to 4Eh.

Byte 2 shall be set to 46h.

Byte 3 shall be set to 49h.

**5.7.1.2. Byte 4-5: Revision number**

This field indicates the revisions of the ONFI specification that the target complies to. The target may support multiple revisions of the ONFI specification. This is a bit field where each defined bit corresponds to a particular specification revision that the target may support.

Bit 0 shall be cleared to zero.

Bit 1 when set to one indicates that the target supports the ONFI revision 1.0 specification.

Bit 2 when set to one indicates that the target supports the ONFI revision 2.0 specification.

Bit 3 when set to one indicates that the target supports the ONFI revision 2.1 specification.

Bit 4 when set to one indicates that the target supports the ONFI revision 2.2 specification.

Bit 5 when set to one indicates that the target supports the ONFI revision 2.3 specification.

Bits 6-15 are reserved and shall be cleared to zero.

**5.7.1.3. Byte 6-7: Features supported**

This field indicates the optional features that the target supports.

Bit 0 when set to one indicates that the target's data bus width is 16-bits. Bit 0 when cleared to zero indicates that the target's data bus width is 8-bits. The host shall use the indicated data bus width for all ONFI commands that are defined to be transferred at the bus width (x8 or x16). Note that some commands, like Read ID, always transfer data as 8-bit only. If the source synchronous data interface is supported, then the data bus width shall be 8-bits.

Bit 1 when set to one indicates that the target supports multiple LUN operations (see section 3.1.3). If bit 1 is cleared to zero, then the host shall not issue commands to a LUN unless all other LUNs on the Target are idle (i.e. R/B\_n is set to one).

Bit 2 when set to one indicates that the target supports non-sequential page programming operations, such that the host may program pages within a block in arbitrary order. Bit 2 when cleared to zero indicates that the target does not support non-sequential page programming operations. If bit 2 is cleared to zero, the host shall program all pages within a block in order starting with page 0.

Bit 3 when set to one indicates that the target supports multi-plane program and erase operations. Refer to section 5.7.1.28.

Bit 4 when set to one indicates that there are no even / odd page restrictions for Copyback operations. Specifically, a read operation may access an odd page and then program the contents to an even page using Copyback. Alternatively, a read operation may access an even page and then program the contents to an odd page using Copyback. Bit 4 when cleared to zero indicates that the host shall ensure that Copyback reads and programs from odd page to odd page or alternatively from even page to even page.

Bit 5 when set to one indicates that the source synchronous data interface is supported by the target. If bit 5 is set to one, then the target shall indicate the source synchronous timing modes supported in the source synchronous timing mode support field. Bit 5 when cleared to zero indicates that the source synchronous data interface is not supported by the target.

Bit 6 when set to one indicates that the target supports multi-plane read operations. Refer to section 5.7.1.28.

Bit 7 when set to one indicates the target includes an extended parameter page that is stored in the data bytes following the last copy of the parameter page. If bit 7 is cleared to zero, then an extended parameter page is not supported. Refer to section 5.7.2. This bit shall be cleared to zero for devices that support EZ NAND. **Note:** This bit was inadvertently specified in the BA NAND specification to show support for BA NAND. If the device supports BA NAND, then the number of bits of ECC correctability should be cleared to 0h in byte 112 of the parameter page.

Bit 8 when set to one indicates that the target supports clearing only the page register for the LUN addressed with the Program (80h) command. If bit 8 is cleared to zero, then a Program (80h) command clears the page register for each LUN that is part of the target. At power-on, the device clears the page register for each LUN that is part of the target. Refer to section 5.26.1 for how to enable this feature.

Bit 9 when set to one indicates that the target supports EZ NAND. If bit 9 is cleared to zero, then the target does not support EZ NAND and is configured as raw NAND.

Bits 10-15 are reserved and shall be cleared to zero.

#### **5.7.1.4. Byte 8-9: Optional commands supported**

This field indicates the optional commands that the target supports.

Bit 0 when set to one indicates that the target supports the Page Cache Program command. If bit 0 is cleared to zero, the host shall not issue the Page Cache Program command to the target. For targets that support EZ NAND, this field shall be cleared to zero indicating that the target does not support the Page Cache Program command.

Bit 1 when set to one indicates that the target supports the Read Cache Random, Read Cache Sequential, and Read Cache End commands. If bit 1 is cleared to zero, the host shall not issue the Read Cache Sequential, Read Cache Random, or Read Cache End commands to the target. For targets that support EZ NAND, this field shall be cleared to zero indicating that the target does not support Read Cache Random, Read Cache Sequential, or Read Cache End commands.

Bit 2 when set to one indicates that the target supports the Get Features and Set Features commands. If bit 2 is cleared to zero, the host shall not issue the Get Features or Set Features commands to the target.

Bit 3 when set to one indicates that the target supports the Read Status Enhanced command. If bit 3 is cleared to zero, the host shall not issue the Read Status Enhanced command to the target. Read Status Enhanced shall be supported if the target has multiple LUNs or supports multi-plane operations.

Bit 4 when set to one indicates that the target supports the Copyback Program and Copyback Read commands. If bit 4 is cleared to zero, the host shall not issue the Copyback Program or Copyback Read commands to the target. If multi-plane operations are supported and this bit is set to one, then multi-plane copyback operations shall be supported.

Bit 5 when set to one indicates that the target supports the Read Unique ID command. If bit 5 is cleared to zero, the host shall not issue the Read Unique ID command to the target.

Bit 6 when set to one indicates that the target supports the Change Read Column Enhanced command. If bit 6 is cleared to zero, the host shall not issue the Change Read Column Enhanced command to the target.

Bit 7 when set to one indicates that the target supports the Change Row Address command. If bit 7 is cleared to zero, the host shall not issue the Change Row Address command to the target.

Bit 8 when set to one indicates that the target supports the Small Data Move command for both Program and Copyback operations. If bit 8 is cleared to zero, the target does not support the Small Data Move command for Program or Copyback operations. The Small Data Move command is mutually exclusive with overlapped multi-plane support. Refer to section 5.19. When bit 8 is set to one, the device shall support the 11h command to flush any internal data pipeline regardless of whether multi-plane operations are supported.

Bit 9 when set to one indicates that the target supports the Reset LUN command. If bit 9 is cleared to zero, the host shall not issue the Reset LUN command.

Bits 10-15 are reserved and shall be cleared to zero.

#### **5.7.1.5. Byte 12-13: Extended parameter page length**

If the target supports an extended parameter page as indicated in the Features supported field, then this field specifies the length of the extended parameter page in multiples of 16 bytes. Thus, a value of 2 corresponds to 32 bytes and a value of 3 corresponds to 48 bytes. The minimum size is 3, corresponding to 48 bytes.

#### **5.7.1.6. Byte 14: Number of parameter pages**

If the target supports an extended parameter page as indicated in the Features supported field, then this field specifies the number of parameter pages present, including the original and the subsequent redundant versions. As an example, a value of 3 means that there are three parameter pages present and thus the extended parameter page starts at byte 768. The number of extended parameter pages should match the number of parameter pages.

#### **5.7.1.7. Byte 32-43: Device manufacturer**

This field contains the manufacturer of the device. The content of this field is an ASCII character string of twelve bytes. The device shall pad the character string with spaces (20h), if necessary, to ensure that the string is the proper length.

There is no standard for how the manufacturer represents their name in the ASCII string. If the host requires use of a standard manufacturer ID, it should use the JEDEC manufacturer ID (refer to section 5.7.1.9).

#### **5.7.1.8. Byte 44-63: Device model**

This field contains the model number of the device. The content of this field is an ASCII character string of twenty bytes. The device shall pad the character string with spaces (20h), if necessary, to ensure that the string is the proper length.

#### **5.7.1.9. Byte 64: JEDEC manufacturer ID**

This field contains the JEDEC manufacturer ID for the manufacturer of the device.

#### **5.7.1.10. Byte 65-66: Date code**

This field contains a date code for the time of manufacture of the device. Byte 65 shall contain the two least significant digits of the year (e.g. a value of 05h to represent the year 2005). Byte 66 shall contain the workweek, where a value of 00h indicates the first week of January.

If the date code functionality is not implemented, the value in this field shall be 0000h.

#### **5.7.1.11. Byte 80-83: Number of data bytes per page**

This field contains the number of data bytes per page. The value reported in this field shall be a power of two. The minimum value that shall be reported is 512 bytes.

#### **5.7.1.12. Byte 84-85: Number of spare bytes per page**

This field contains the number of spare bytes per page. There are no restrictions on the value.

Appendix B lists recommendations for the number of bytes per page based on the page size and the number of bits of ECC correctability for the device.

#### **5.7.1.13. Byte 86-89: Obsolete – Number of data bytes per partial page**

This field is obsolete. It previously contained the number of data bytes per partial page.

#### **5.7.1.14. Byte 90-91: Obsolete – Number of spare bytes per partial page**

This field is obsolete. It previously contained the number of spare bytes per partial page.

#### **5.7.1.15. Byte 92-95: Number of pages per block**

This field contains the number of pages per block. This value shall be a multiple of 32. Refer to section 3.1 for addressing requirements.

#### **5.7.1.16. Byte 96-99: Number of blocks per logical unit**

This field contains the number of blocks per logical unit. There are no restrictions on this value. Refer to section 3.1 for addressing requirements.

#### **5.7.1.17. Byte 100: Number of logical units (LUNs)**

This field indicates the number of logical units the target supports. Logical unit numbers are sequential, beginning with a LUN address of 0. This field shall be greater than zero.

#### **5.7.1.18. Byte 101: Number of Address Cycles**

This field indicates the number of address cycles used for row and column addresses. The reported number of address cycles shall be used by the host in operations that require row and/or column addresses (e.g. Page Program).

Bits 0-3 indicate the number of address cycles used for the row address. This field shall be greater than zero.

Bits 4-7 indicate the number of address cycles used for the column address. This field shall be greater than zero.

**NOTE:** Throughout this specification examples are shown with 2-byte column addresses and 3-byte row addresses. However, the host is responsible for providing the number of column and row address cycles in each of these sequences based on the values in this field.

#### **5.7.1.19. Byte 102: Number of bits per cell**

This field indicates the number of bits per cell in the Flash array. This field shall be greater than zero.

For some devices, including implementations supporting EZ NAND, the device may be constructed of Flash arrays of different types. A value of FFh indicates that the number of bits per cell is not specified.

#### **5.7.1.20. Byte 103-104: Bad blocks maximum per LUN**

This field contains the maximum number of blocks that may be defective at manufacture and over the life of the device per LUN. The maximum rating assumes that the host is following the block endurance requirements and the ECC requirements reported in the parameter page.

#### **5.7.1.21. Byte 105-106: Block endurance**

This field indicates the maximum number of program/erase cycles per addressable page/block. This value assumes that the host is using at least the minimum ECC correctability reported in the parameter page.

A page may be programmed in partial operations subject to the value reported in the Number of programs per page field. However, programming different locations within the same page does not count against this value more than once per full page.

The block endurance is reported in terms of a value and a multiplier according to the following equation:  $\text{value} \times 10^{\text{multiplier}}$ . Byte 105 comprises the value. Byte 106 comprises the multiplier. For example, a target with an endurance of 75,000 cycles would report this as a value of 75 and a multiplier of 3 ( $75 \times 10^3$ ). For a write once device, the target shall report a value of 1 and a multiplier of 0. For a read-only device, the target shall report a value of 0 and a multiplier of 0. The value field shall be the smallest possible; for example 100,000 shall be reported as a value of 1 and a multiplier of 5 ( $1 \times 10^5$ ).

#### **5.7.1.22. Byte 107: Guaranteed valid blocks at beginning of target**

This field indicates the number of guaranteed valid blocks starting at block address 0 of the target. The minimum value for this field is 1h. The blocks are guaranteed to be valid for the endurance specified for this area (see section 5.7.1.23) when the host follows the specified number of bits to correct.

#### **5.7.1.23. Byte 108-109: Block endurance for guaranteed valid blocks**

This field indicates the minimum number of program/erase cycles per addressable page/block in the guaranteed valid block area (see section 5.7.1.22). This value requires that the host is using at least the minimum ECC correctability reported in the parameter page. This value is not encoded. If the value is 0000h, then no minimum number of cycles is specified, though the block(s) are guaranteed valid from the factory.

#### **5.7.1.24. Byte 110: Number of programs per page**

This field indicates the maximum number of times a portion of a page may be programmed without an erase operation. After the number of programming operations specified have been performed, the host shall issue an erase operation to that block before further program operations to the affected page. This field shall be greater than zero. Programming the same portion of a page without an erase operation results in indeterminate page contents.

#### **5.7.1.25. Byte 111: Obsolete – Partial programming attributes**

This field is obsolete. It previously indicated the attributes for partial page programming that the target supports.

#### **5.7.1.26. Byte 112: Number of bits ECC correctability**

This field indicates the number of bits that the host should be able to correct per 512 bytes of data. With this specified amount of error correction by the host, the target shall achieve the block endurance specified in the parameter page. When the specified amount of error correction is applied by the host and the block endurance is followed, then the maximum number of bad blocks shall not be exceeded by the device. All used bytes in the page shall be protected by host controller ECC including the spare bytes if the minimum ECC requirement has a value greater than zero.

If the recommended ECC codeword size is not 512 bytes, then this field shall be set to FFh. The host should then read the Extended ECC Information that is part of the extended parameter page to retrieve the ECC requirements for this device.

When this value is cleared to zero, the target shall return valid data. For targets that support EZ NAND, this field shall be cleared to zero indicating that the target returns corrected data to the host.

#### **5.7.1.27. Byte 113: Multi-plane addressing**

This field describes parameters for multi-plane addressing.

Bits 0-3 indicate the number of bits that are used for multi-plane addressing. This value shall be greater than 0h when multi-plane operations are supported. For information on the plane address location, refer to section 3.1.1.

Bits 4-7 are reserved.

#### **5.7.1.28. Byte 114: Multi-plane operation attributes**

This field describes attributes for multi-plane operations. This byte is mandatory when multi-plane operations are supported as indicated in the Features supported field.

Bit 0 indicates whether overlapped multi-plane operations are supported. If bit 0 is set to one, then overlapped multi-plane operations are supported. If bit 0 is cleared to zero, then concurrent multi-plane operations are supported.

Bit 1 indicates that there are no block address restrictions for the multi-plane operation. If set to one all block address bits may be different between multi-plane operations. If cleared to zero, there are block address restrictions. Refer to bit 5 for the specific block address restrictions required.

Bit 2 indicates whether program cache is supported with multi-plane programs. If set to one then program cache is supported for multi-plane program operations. If cleared to zero then program cache is not supported for multi-plane program operations. Note that program cache shall not be used with multi-plane copyback program operations. See bit 3 for restrictions on the multi-plane addresses that may be used.

Bit 3 indicates whether the block address bits other than the multi-plane address bits of multi-plane addresses may change during either: a) a program cache sequence between 15h commands, or b) a read cache sequence between 31h commands. If set to one and bit 2 is set to one, then the host may change the number of multi-plane addresses and the value of the block address bits (other than the multi-plane address bits) in the program cache sequence. If set to one and bit 4 is set to one, then the host may change the number of multi-plane addresses and the value of the block address bits (other than the multi-plane address bits) in the read cache sequence. If cleared to zero and bit 2 is set to one, then for each program cache operation the block address bits (other than the plane address bits) and number of multi-plane addresses issued to the LUN shall be the same. If cleared to zero and bit 4 is set to one, then for each read cache operation the block address bits (other than the multi-plane address bits) and number of multi-plane addresses issued to the LUN shall be the same.

Bit 4 indicates whether read cache is supported with multi-plane reads. If set to one then read cache is supported for multi-plane read operations. If cleared to zero then read cache is not supported for multi-plane read operations. Note that read cache shall not be used with multi-plane copyback read operations.

Bit 5 indicates the type of block address restrictions required for the multi-plane operation. If set to one then all block address bits (other than the multi-plane address bits) shall be the same if the XNOR of the lower multi-plane address bits between two multi-plane addresses is one. If cleared to zero, all block address bits (other than the multi-plane address bits) shall be the same regardless of the multi-plane address bits between two plane addresses. See section 3.1.1.1 for a detailed definition of interleaved block address restrictions. These restrictions apply to all multi-plane operations (Read, Program, Erase, and Copyback Program).

Bits 6-7 are reserved.

#### **5.7.1.1. Byte 115: EZ NAND support**

This field describes the EZ NAND attributes that the target supports. This field is only used if EZ NAND is supported by the target.

Bit 0 indicates whether the target supports automatic retries being enabled and disabled explicitly by the host using Set Features. If this bit is set to one, then the host may enable or disable automatic retries using Set Features. If this bit is cleared to zero, then the EZ NAND controller determines whether to perform a retry without host intervention. If the EZ NAND controller executes an automatic retry, the typical page read time (tR) may be exceeded.

Bit 1 indicates whether Copyback is supported with destination planes or LUNs that are different from the source. If this bit is set to one, then the plane or LUN of the destination for the Copyback may be different than the source. If this bit is cleared to zero, then the plane or LUN of the destination for a Copyback shall be the same as the source.

Bit 2 indicates whether the target requires Copyback Read and Copyback Program command adjacency. If this bit is set to one, then each Copyback Read command issued shall be followed by a Copyback Program command prior to any additional Copyback Read command being issued to the target. I.e., A Copyback Read command shall be explicitly paired with a Copyback Program command). If this bit is cleared to zero, then there may be multiple Copyback Read commands prior to issuing a Copyback Program command. For ONFI 2.3, Copyback Adjacency is required and this bit shall be set to one.

Bits 3-7 are reserved.

#### **5.7.1.2. Byte 128: I/O pin capacitance, maximum**

This field indicates the maximum I/O pin capacitance for the target in pF. This may be used by the host to calculate the load for the data bus. Refer to section 2.11.

#### **5.7.1.3. Byte 129-130: Asynchronous timing mode support**

This field indicates the asynchronous timing modes supported. The target shall always support asynchronous timing mode 0.

Bit 0 shall be set to one. It indicates that the target supports asynchronous timing mode 0.

Bit 1 when set to one indicates that the target supports asynchronous timing mode 1.

Bit 2 when set to one indicates that the target supports asynchronous timing mode 2.

Bit 3 when set to one indicates that the target supports asynchronous timing mode 3.

Bit 4 when set to one indicates that the target supports asynchronous timing mode 4.

Bit 5 when set to one indicates that the target supports asynchronous timing mode 5.

Bits 6-15 are reserved and shall be cleared to zero.

#### **5.7.1.4. Byte 131-132: Obsolete – Asynchronous program cache timing mode support**

This field is obsolete. It previously indicated the asynchronous timing modes supported for Page Cache Program operations.

#### **5.7.1.5. Byte 133-134: Maximum page program time**

This field indicates the maximum page program time (tPROG) in microseconds.

#### **5.7.1.6. Byte 135-136: Maximum block erase time**

This field indicates the maximum block erase time (tBERS) in microseconds.

#### **5.7.1.7. Byte 137-138: Maximum page read time**

This field indicates the maximum page read time (tR) in microseconds. For devices that support EZ NAND, this value is the tR maximum at the end of life and is related to the uncorrectable bit error rate (UBER) specified for the device.

#### **5.7.1.8. Byte 139-140: Minimum change column setup time**

This field indicates the minimum change column setup time (tCCS) in nanoseconds. After issuing a Change Read Column command, the host shall not read data until a minimum of tCCS time has elapsed. After issuing a Change Write Column command including all column address cycles, the host shall not write data until a minimum of tCCS time has elapsed. The value of tCCS shall always be longer than or equal to tWHR and tADL when the source synchronous data interface is supported.

#### **5.7.1.9. Byte 141-142: Source synchronous timing mode support**

This field indicates the source synchronous timing modes supported. If the source synchronous data interface is supported by the target, at least one source synchronous timing mode shall be supported. The target shall support an inclusive range of source synchronous timing modes (i.e. if timing mode n-1 and n+1 are supported, then the target shall also support timing mode n).

Bit 0 when set to one indicates that the target supports source synchronous timing mode 0.

Bit 1 when set to one indicates that the target supports source synchronous timing mode 1.

Bit 2 when set to one indicates that the target supports source synchronous timing mode 2.

Bit 3 when set to one indicates that the target supports source synchronous timing mode 3.

Bit 4 when set to one indicates that the target supports source synchronous timing mode 4.

Bit 5 when set to one indicates that the target supports source synchronous timing mode 5.

Bits 6-15 are reserved and shall be cleared to zero.

#### **5.7.1.10. Byte 143: Source synchronous features**

This field describes features and attributes for source synchronous operation. This byte is mandatory when the source synchronous data interface is supported.

Bit 0 indicates the tCAD value that shall be used by the host. If bit 0 is set to one, then the host shall use the tCADs (slow) value in source synchronous command, address and data transfers. If bit 0 is cleared to zero, then the host shall use the tCADf (fast) value in source synchronous command, address and data transfers.

Bit 1 indicates if the typical CLK, I/O and input pin capacitance values are reported in the parameter page. If bit 1 is set to one, then the typical CLK, I/O and input pin capacitance values are reported in the parameter page. If bit 1 is cleared to zero, then the typical capacitance fields are not used.

Bit 2 indicates that the device supports the CLK being stopped during data input, as described in Figure 34. If bit 2 is set to one, then the host may optionally stop the CLK during data input for power savings. If bit 2 is set to one, the host may pause data while the CLK is stopped. If bit 2 is cleared to zero, then the host shall leave CLK running during data input.

Bits 3-7 are reserved.

#### **5.7.1.11. Byte 144-145: CLK input pin capacitance, typical**

This field indicates the typical CLK input pin capacitance for the target. This value applies to the CLK\_t and CLK\_c signals. This field is specified in 0.1 pF units. For example, a value of 31 corresponds to 3.1 pF. The variance from this value is less than +/- 0.5 pF per LUN per x8 data

bus. As an example, if two LUNs are present per x8 data bus than the total variance is less than +/- 1.0 pF. This value is only valid if the typical capacitance values are supported as indicated in the source synchronous features field. Additional constraints on the CLK input pin capacitance are specified in section 4.2.3.

#### **5.7.1.12. Byte 146-147: I/O pin capacitance, typical**

This field indicates the typical I/O pin capacitance for the target. This field is specified in 0.1 pF units. For example, a value of 31 corresponds to 3.1 pF. The variance from this value is less than +/- 0.5 pF per LUN per x8 data bus. As an example, if two LUNs are present than the total variance is less than +/- 1 pF. This value is only valid if the typical capacitance values are supported as indicated in the source synchronous features field. Additional constraints on the I/O pin capacitance are specified in section 4.2.3.

#### **5.7.1.13. Byte 148-149: Input pin capacitance, typical**

This field indicates the typical input pin capacitance for the target. This value applies to all inputs except the following: CLK\_t, CLK\_c, CE\_n and WP\_n signals. This field is specified in 0.1 pF units. For example, a value of 31 corresponds to 3.1 pF. The variance from this value is less than +/- 0.5 pF per LUN per x8 data bus. As an example, if two LUNs are present than the total variance is less than +/- 1 pF. This value is only valid if the typical capacitance values are supported as indicated in the source synchronous features field. Additional constraints on the input pin capacitance are specified in section 4.2.3.

#### **5.7.1.14. Byte 150: Input pin capacitance, maximum**

This field indicates the maximum input pin capacitance for the target in pF. This value applies to all inputs, including CLK\_t, CLK\_c, CE\_n, and WP\_n. This may be used by the host to calculate the load for the data bus. Refer to section 2.11.

#### **5.7.1.15. Byte 151: Driver strength support**

This field describes if the target supports configurable driver strengths and its associated features.

Bit 0 when set to one indicates that the target supports configurable driver strength settings as defined in Table 26. If this bit is set to one, then the device shall support both the 35 Ohm and 50 Ohm settings. If this bit is set to one, then the device shall power-on with a driver strength at the 35 Ohm value defined in Table 26. If this bit is cleared to zero, then the driver strength at power-on is undefined. This bit shall be set to one for devices that support the source synchronous data interface.

Bit 1 when set to one indicates that the target supports the 25 Ohm setting in Table 26 for use in the I/O Drive Strength setting. This bit shall be set to one for devices that support the source synchronous data interface.

Bit 2 when set to one indicates that the target supports the 18 Ohm setting in Table 26 for use in the I/O Drive Strength setting. This bit shall be set to one for devices that support the source synchronous data interface.

Bits 3-7 are reserved.

#### **5.7.1.16. Byte 152-153: Maximum multi-plane page read time**

This field indicates the maximum page read time (tR) for multi-plane page reads in microseconds. Multi-plane page read times may be longer than non-multi-plane pages read times. This field

shall be supported if the target supports multi-plane reads as indicated in the Features supported field.

#### **5.7.1.17. Byte 154-155: Program page register clear enhancement tADL value**

This field indicates the ALE to data loading time (tADL) in nanoseconds when the program page register clear enhancement is enabled. If the program page register clear enhancement is disabled, then the tADL value is as defined for the selected timing mode. This increased tADL value only applies to Program (80h) command sequences; it does not apply for Set Features, Copyback, or other commands.

#### **5.7.1.1. Byte 156-157: Typical page read time for EZ NAND**

This field indicates the typical page read time (tR) in microseconds for devices that support EZ NAND. This field is not used for devices that do not support EZ NAND. For devices that include multiple bits per cell in the NAND used, this value is an average of the tR typical values for the pages (e.g. lower and upper pages).

#### **5.7.1.2. Byte 164-165: Vendor specific Revision number**

This field indicates a vendor specific revision number. This field should be used by vendors to indicate the supported layout for the vendor specific parameter page area and the vendor specific feature addresses. The format of this field is vendor specific.

#### **5.7.1.3. Byte 166-253: Vendor specific**

This field is reserved for vendor specific use.

#### **5.7.1.4. Byte 254-255: Integrity CRC**

The Integrity CRC (Cyclic Redundancy Check) field is used to verify that the contents of the parameter page were transferred correctly to the host. The CRC of the parameter page is a word (16-bit) field. The CRC calculation covers all of data between byte 0 and byte 253 of the parameter page inclusive.

The CRC shall be calculated on byte (8-bit) quantities starting with byte 0 in the parameter page. The bits in the 8-bit quantity are processed from the most significant bit (bit 7) to the least significant bit (bit 0).

The CRC shall be calculated using the following 16-bit generator polynomial:

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

This polynomial in hex may be represented as 8005h.

The CRC value shall be initialized with a value of 4F4Eh before the calculation begins. There is no XOR applied to the final CRC value after it is calculated. There is no reversal of the data bytes or the CRC calculated value.

#### **5.7.1.5. Byte 256-511: Redundant Parameter Page 1**

This field shall contain the values of bytes 0-255 of the parameter page. Byte 256 is the value of byte 0.

The redundant parameter page is used when the integrity CRC indicates that there was an error in bytes 0-255. The redundant parameter page shall be stored in non-volatile media; the target shall not create these bytes by retransmitting the first 256 bytes.

### 5.7.1.6. Byte 512-767: Redundant Parameter Page 2

This field shall contain the values of bytes 0-255 of the parameter page. Byte 512 is the value of byte 0.

The redundant parameter page is used when the integrity CRC indicates that there was an error in bytes 0-255 and in the first redundant parameter page. The redundant parameter page shall be stored in non-volatile media; the target shall not create these bytes by retransmitting the first 256 bytes.

### 5.7.1.7. Byte 768+: Additional Redundant Parameter Pages

Bytes at offset 768 and above may contain additional redundant copies of the parameter page. There is no limit to the number of redundant parameter pages that the target may provide. The target may provide additional copies to guard against the case where all three mandatory copies have invalid CRC checks.

The host should determine whether an additional parameter page is present by checking the first Dword. If at least two out of four bytes match the parameter page signature, then an additional parameter page is present.

## 5.7.2. Extended Parameter Page Data Structure Definition

The extended parameter page, if present, provides additional information to the host that there was insufficient space to include in the parameter page. The extended parameter page is organized in sections. Each section is a multiple of 16 bytes in length. The section types are specified in Table 44.

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 44 Section Type Definitions**

Section types shall be specified in the extended parameter page in order (other than section type value 0). For example, if section type 12 and section type 15 were both present in the extended parameter page then section type 12 shall precede section type 15. There shall only be one instantiation of each section type. All unused sections shall be marked with a section type value of 0. When software encounters a section type value of 0, this marks the end of the valid sections.

Table 45 defines the layout of section type 1. Section type 1 specifies additional sections when more than eight sections are present in the extended parameter page. The length of section type 1 shall be a multiple of 16 bytes.

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 45 Section Type 1: Additional Section Type and Length Specifiers**

Table 46 defines the layout of section type 2. Section type 2 specifies extended ECC information. Each extended ECC information block is eight bytes in length. If an extended ECC information block is not specified, then all values in that block shall be cleared to 0h. The length of section type 2 shall be a multiple of 16 bytes.

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 46 Section Type 2: Extended ECC Information**

The definition of the extended ECC information block is specified in section 3.3.

Table 47 defines the extended parameter page data structure. For parameters that span multiple bytes, the least significant byte of the parameter corresponds to the first byte. See section 1.4.2.3 for more information on the representation of word and Dword values.

Values are reported in the extended parameter page in units of bytes when referring to items related to the size of data access (as in an 8-bit data access device). For example, the target will return how many data *bytes* are in a page. For a device that supports 16-bit data access, the host is required to convert byte values to word values for its use.

Unused fields should be cleared to 0h by the target.

Byte	O/M	Description
<b>Revision information and features block</b>		
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 47      Extended Parameter Page definition**

#### **5.7.2.1. Byte 0-1: Integrity CRC**

The Integrity CRC (Cyclic Redundancy Check) field is used to verify that the contents of the extended parameter page were transferred correctly to the host. The CRC of the extended parameter page is a word (16-bit) field. The CRC calculation covers all of data between byte 2 and the end of the extended parameter page inclusive.

The CRC shall be calculated on byte (8-bit) quantities starting with byte 2 in the extended parameter page to the end of the extended parameter page. The bits in the 8-bit quantity are processed from the most significant bit (bit 7) to the least significant bit (bit 0).

The CRC shall be calculated using the following 16-bit generator polynomial:

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

This polynomial in hex may be represented as 8005h.

The CRC value shall be initialized with a value of 4F4Eh before the calculation begins. There is no XOR applied to the final CRC value after it is calculated. There is no reversal of the data bytes or the CRC calculated value.

#### **5.7.2.2. Byte 2-5: Extended parameter page signature**

This field contains the extended parameter page signature. When two or more bytes of the signature are valid, then it denotes that a valid copy of the extended parameter page is present.

Byte 2 shall be set to 45h.

Byte 3 shall be set to 50h.

Byte 4 shall be set to 50h.

Byte 5 shall be set to 53h.

### **5.7.2.3. Byte 16: Section 0 type**

Section 0 is the first section in the extended parameter page and begins at byte offset 32. This field specifies the type of section 0. Section types are defined in Table 44.

### **5.7.2.4. Byte 17: Section 0 length**

Section 0 is the first section in the extended parameter page and begins at byte offset 32. This field specifies the length of section 0. The length is specified in multiples of 16 bytes. Thus, a value of 1 corresponds to 16 bytes and a value of 2 corresponds to 32 bytes.

### **5.7.2.5. Byte 18: Section 1 type**

Section 1 is the second section in the extended parameter page and starts immediately following section 0. This field specifies the type of section 1. Section types are defined in Table 44. If section 1 is not present, then the type field shall be cleared to 0.

### **5.7.2.6. Byte 19: Section 1 length**

Section 1 is the second section in the extended parameter page and starts immediately following section 0. This field specifies the length of section 1. The length is specified in multiples of 16 bytes. Thus, a value of 1 corresponds to 16 bytes and a value of 2 corresponds to 32 bytes. If section 1 is not present, then the length field shall be cleared to 0.

### **5.7.2.7. Byte 20: Section 2 type**

Section 2 is the third section in the extended parameter page and starts immediately following section 1. This field specifies the type of section 2. Section types are defined in Table 44. If section 2 is not present, then the type field shall be cleared to 0.

### **5.7.2.8. Byte 21: Section 2 length**

Section 2 is the third section in the extended parameter page and starts immediately following section 1. This field specifies the length of section 2. The length is specified in multiples of 16 bytes. Thus, a value of 1 corresponds to 16 bytes and a value of 2 corresponds to 32 bytes. If section 2 is not present, then the length field shall be cleared to 0.

### **5.7.2.9. Byte 22-31: Section 3 – 7 types and lengths**

Bytes 22-31 define the type and lengths for sections 3 – 7 in order, following the same definition and layout as section 0 and 1 type and length definitions. If a section is not present, then the type and length fields for that section shall be cleared to 0.

### **5.7.2.10. Byte 32 – (end): Section information**

Section 0 begins at byte offset 32 and is a multiple of 16 bytes. If there are additional sections (section 1, 2, 3, etc), each section starts immediately following the previous section and is a multiple of 16 bytes.

## **5.8. Read Unique ID Definition**

The Read Unique ID function is used to retrieve the 16 byte unique ID (UID) for the device. The unique ID when combined with the device manufacturer shall be unique.

The UID data may be stored within the Flash array. To allow the host to determine if the UID is without bit errors, the UID is returned with its complement, as shown in Table 48. If the XOR of the UID and its bit-wise complement is all ones, then the UID is valid.

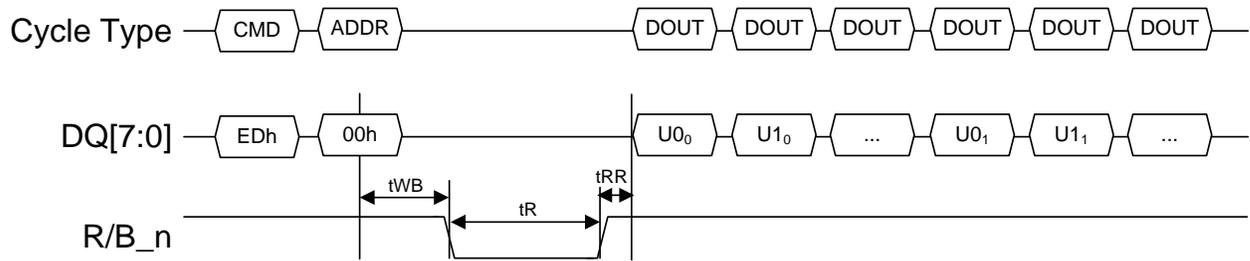
<b>Bytes</b>	<b>Value</b>
0-15	UID
16-31	UID complement (bit-wise)

**Table 48                      UID and Complement**

To accommodate robust retrieval of the UID in the case of bit errors, sixteen copies of the UID and the corresponding complement shall be stored by the target. For example, reading bytes 32-63 returns to the host another copy of the UID and its complement.

Read Status Enhanced shall not be used during execution of the Read Unique ID command.

Figure 58 defines the Read Unique ID behavior. The host may use any timing mode supported by the target in order to retrieve the UID data.

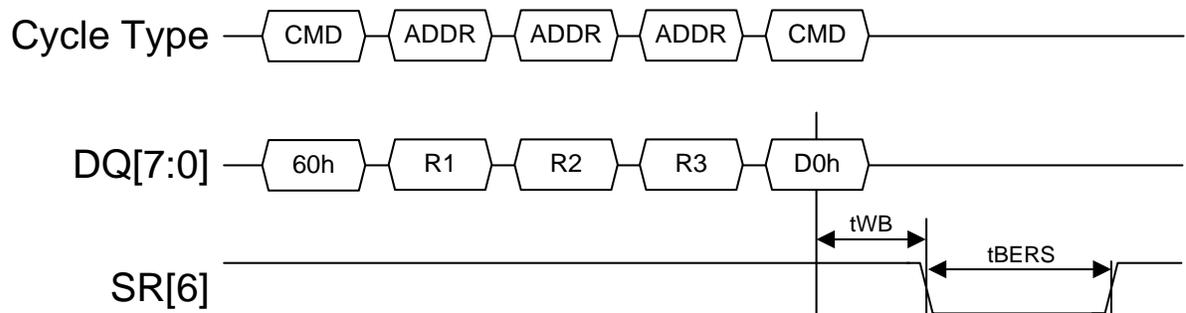


**Figure 58 Read Unique ID command timing**

$U0_k-U1_k$  The  $k$ th copy of the UID and its complement. Sixteen copies are stored. Reading beyond 512 bytes returns indeterminate values.

### 5.9. Block Erase Definition

The Block Erase function erases the block of data identified by the block address parameter on the LUN specified. After a successful Block Erase, all bits shall be set to one in the block. SR[0] is valid for this command after SR[6] transitions from zero to one until the next transition of SR[6] to zero. Figure 59 defines the Block Erase behavior and timings.



**Figure 59 Block Erase timing**

R1-R3 The row address of the block to be erased. R1 is the least significant byte in the row address.

### 5.10. Read Status Definition

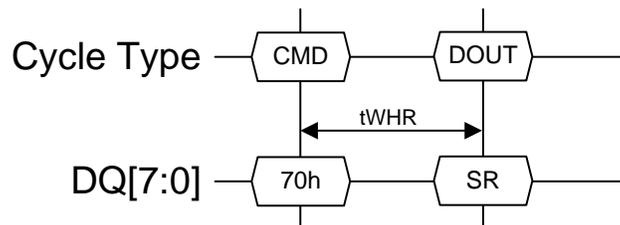
In the case of non-multi-plane operations, the Read Status function retrieves a status value for the last operation issued. If multiple multi-plane operations are in progress on a single LUN, then Read Status returns the composite status value for status register bits that are independent per plane address. Specifically, Read Status shall return the combined status value of the independent status register bits according to Table 49. See section 5.13 for status register bit definitions.

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

**Table 49 Composite Status Value**

When issuing Read Status in the source synchronous data interface, each data byte is received twice. The host shall only latch one copy of each data byte. See section 4.3.2.5.

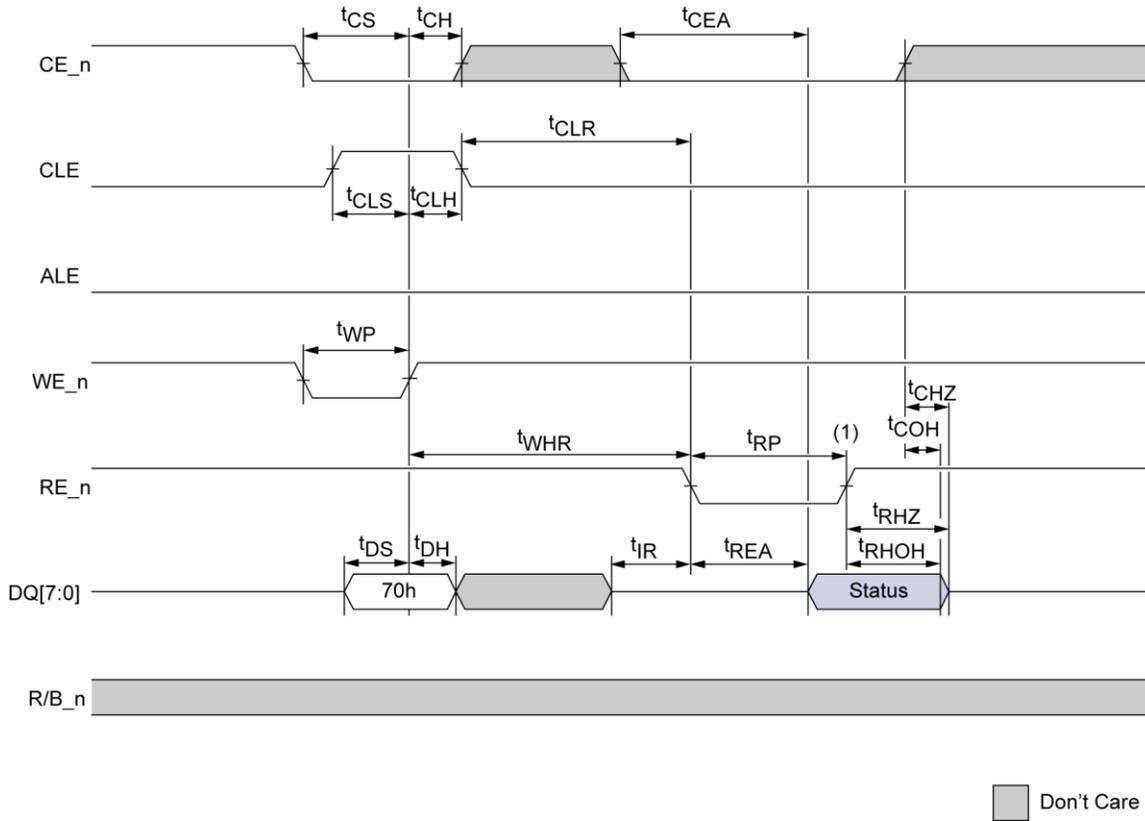
Figure 60 defines the Read Status behavior and timings.



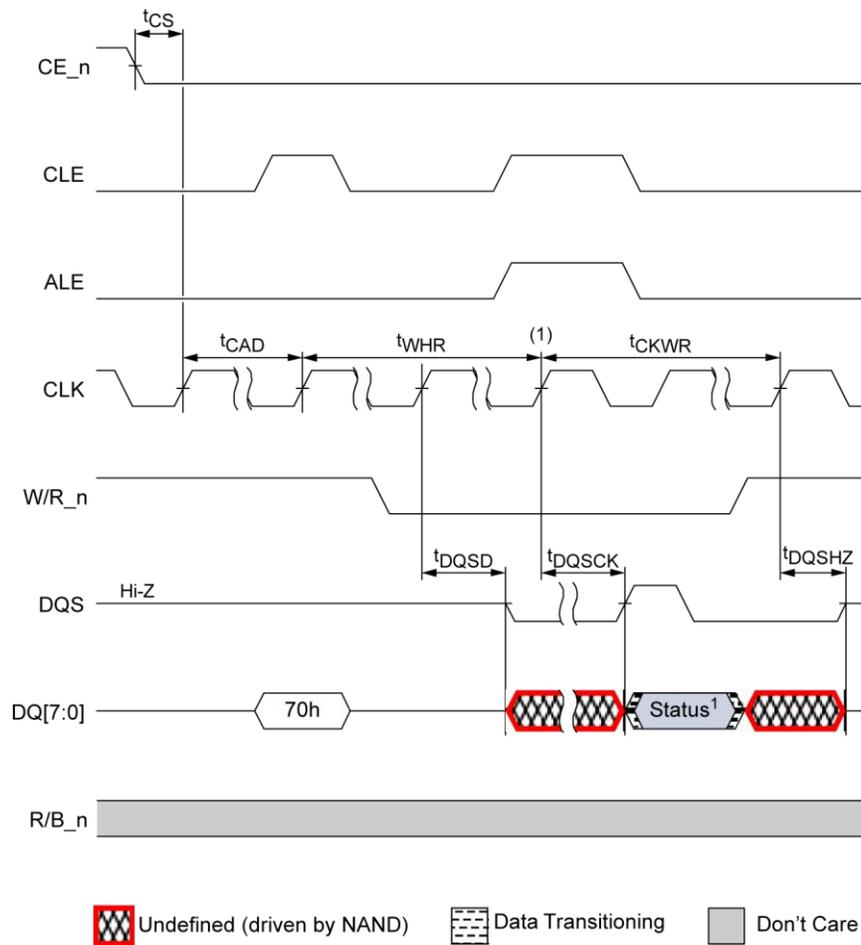
**Figure 60 Read Status timing**

SR Status value as defined in section 5.13.

The Read Status command may be issued using either the asynchronous or source synchronous data interfaces. The timing parameters for each data interface are shown in Figure 61 and Figure 62.



**Figure 61** Read Status command using asynchronous data interface



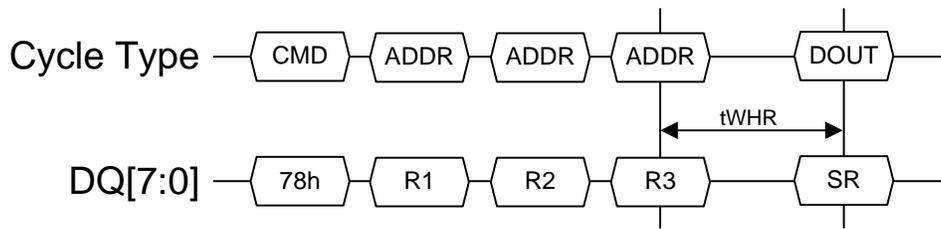
**Figure 62 Read Status command using source synchronous data interface**

Note (1): For the asynchronous data interface, status may be continually read by pulsing RE\_n or leaving RE\_n low. For the source synchronous interface, status may continually be read by leaving ALE/CLE at a value of 11b.

### 5.11. Read Status Enhanced Definition

The Read Status Enhanced function retrieves the status value for a previous operation on the particular LUN and plane address specified. Figure 63 defines the Read Status Enhanced behavior and timings. If the row address entered is invalid, the Status value returned has an indeterminate value. The host uses Read Status Enhanced for LUN selection (refer to section 3.1.2). Note that Read Status Enhanced has no effect on which page register is selected for data output within the LUN.

When issuing Read Status Enhanced in the source synchronous data interface, each data byte is received twice. The host shall only latch one copy of each data byte. See section 4.3.2.5.



**Figure 63 Read Status Enhanced timing**

R1-R3 Row address of the previous operation to retrieve status for. R1 is the least significant byte. The row address contains both the LUN and plane address to retrieve status for.

SR Status value as defined in section 5.13.

### 5.12. Read Status and Read Status Enhanced required usage

In certain sequences only one status command shall be used by the host. This section outlines situations in which a particular status command is required to be used.

If a command is issued to a LUN while R/B\_n is cleared to zero, then the next status command shall be Read Status Enhanced. Read Status Enhanced causes LUNs that are not selected to turn off their output buffers. This ensures that only the LUN selected by the Read Status Enhanced commands responds to a subsequent toggle of the RE\_n input signal.

When the host has issued Read Page commands to multiple LUNs at the same time, the host shall issue Read Status Enhanced before reading data from either LUN. Read Status Enhanced causes LUNs that are not selected to turn off their output buffers. This ensures that only the LUN selected by the Read Status Enhanced commands responds to a subsequent toggle of the RE\_n input signal after data output is selected with the 00h command. Refer to section 3.1.3 for additional requirements if a Change Read Column (Enhanced) command is used as part of a multiple LUN read sequence.

During and after Target level commands, the host shall not issue the Read Status Enhanced command. In these sequences, the host uses Read Status to check for the status value. The only exception to this requirement is if commands were outstanding to multiple LUNs when a Reset was issued. In this case, the Read Status Enhanced command shall be used to determine when each active LUN has completed Reset.

### 5.13. Status Field Definition

The returned status register byte value (SR) for Read Status and Read Status Enhanced has the format described below. If the RDY bit is cleared to zero, all other bits in the status byte (except WP\_n) are invalid and shall be ignored by the host.

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

FAIL If set to one, then the last command failed. If cleared to zero, then the last command was successful. For raw NAND operation, this bit is only valid for

program and erase operations. For EZ NAND operation, this bit is valid for read, program, and erase operations. During program cache operations, this bit is only valid when ARDY is set to one.

**FAILC** If set to one, then the command issued prior to the last command failed. If cleared to zero, then the command issued prior to the last command was successful. This bit is only valid for program cache operations. This bit is not valid until after the second 15h command or the 10h command has been transferred in a Page Cache Program sequence. When program cache is not supported, this bit is not used and shall be cleared to zero. For EZ NAND operation, this bit is not used (cache commands are not supported with EZ NAND).

**CSP** Command Specific: This bit has command specific meaning.

For EZ NAND read operations, if CSP (Threshold) bit is set to one then the last read operation exceeded the ECC threshold and the host should take appropriate action (e.g. rewrite the data to a new location). When FAIL is set to one, the CSP (Threshold) bit is don't care.

For all other operations, this bit is reserved.

**ARDY** If set to one, then there is no array operation in progress. If cleared to zero, then there is a command being processed (RDY is cleared to zero) or an array operation in progress. When overlapped multi-plane operations or cache commands are not supported, this bit is not used.

**RDY** If set to one, then the LUN or plane address is ready for another command and all other bits in the status value are valid. If cleared to zero, then the last command issued is not yet complete and SR bits 5:0 are invalid and shall be ignored by the host. This bit impacts the value of R/B\_n, refer to section 2.16.2. When caching operations are in use, then this bit indicates whether another command can be accepted, and ARDY indicates whether the last operation is complete.

**WP\_n** If set to one, then the device is not write protected. If cleared to zero, then the device is write protected. This bit shall always be valid regardless of the state of the RDY bit.

**R** Reserved (0)

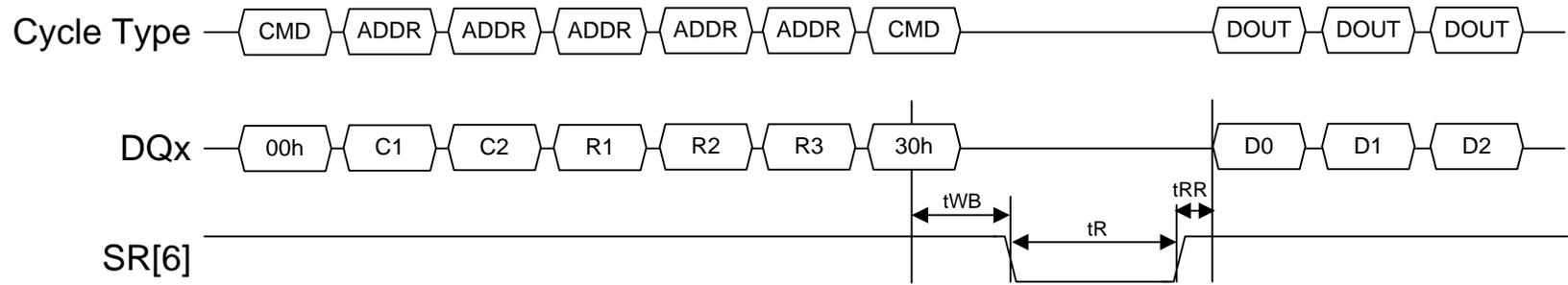
**VSP** Vendor Specific

## 5.14. Read Definition

The Read function reads a page of data identified by a row address for the LUN specified. The page of data is made available to be read from the page register starting at the column address specified. Figure 64 defines the Read behavior and timings. Reading beyond the end of a page results in indeterminate values being returned to the host.

While monitoring the read status to determine when the tR (transfer from Flash array to page register) is complete, the host shall re-issue a command value of 00h to start reading data.

Issuing a command value of 00h will cause data to be returned starting at the selected column address.



**Figure 64 Read timing**

C1-C2 Column address of the page to retrieve. C1 is the least significant byte.

R1-R3 Row address of the page to retrieve. R1 is the least significant byte.

Dn Data bytes read from the addressed page.

## 5.15. Read Cache Definition

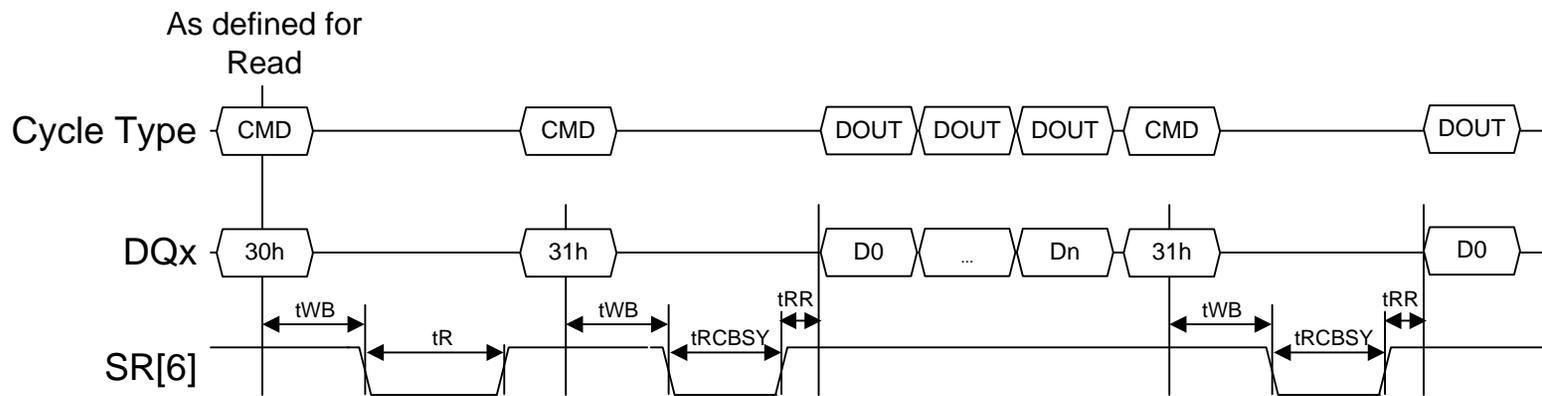
This command is not supported for EZ NAND.

The Read Cache Sequential and Read Cache Random functions permit a page to be read from the page register while another page is simultaneously read from the Flash array for the selected LUN. A Read Page command, as defined in section 5.14, shall be issued prior to the initial Read Cache Sequential or Read Cache Random command in a read cache sequence. A Read Cache Sequential or Read Cache Random command shall be issued prior to a Read Cache End (3Fh) command being issued.

The Read Cache (Sequential or Random) function may be issued after the Read function is complete (SR[6] is set to one). The host may enter the address of the next page to be read from the Flash array. Data output always begins at column address 00h. If the host does not enter an address to retrieve, the next sequential page is read. When the Read Cache (Sequential or Random) function is issued, SR[6] is cleared to zero (busy). After the operation is begun SR[6] is set to one (ready) and the host may begin to read the data from the previous Read or Read Cache (Sequential or Random) function. Issuing an additional Read Cache (Sequential or Random) function copies the data most recently read from the array into the page register. When no more pages are to be read, the final page is copied into the page register by issuing the 3Fh command. The host may begin to read data from the page register when SR[6] is set to one (ready). When the 31h and 3Fh commands are issued, SR[6] shall be cleared to zero (busy) until the page has finished being copied from the Flash array.

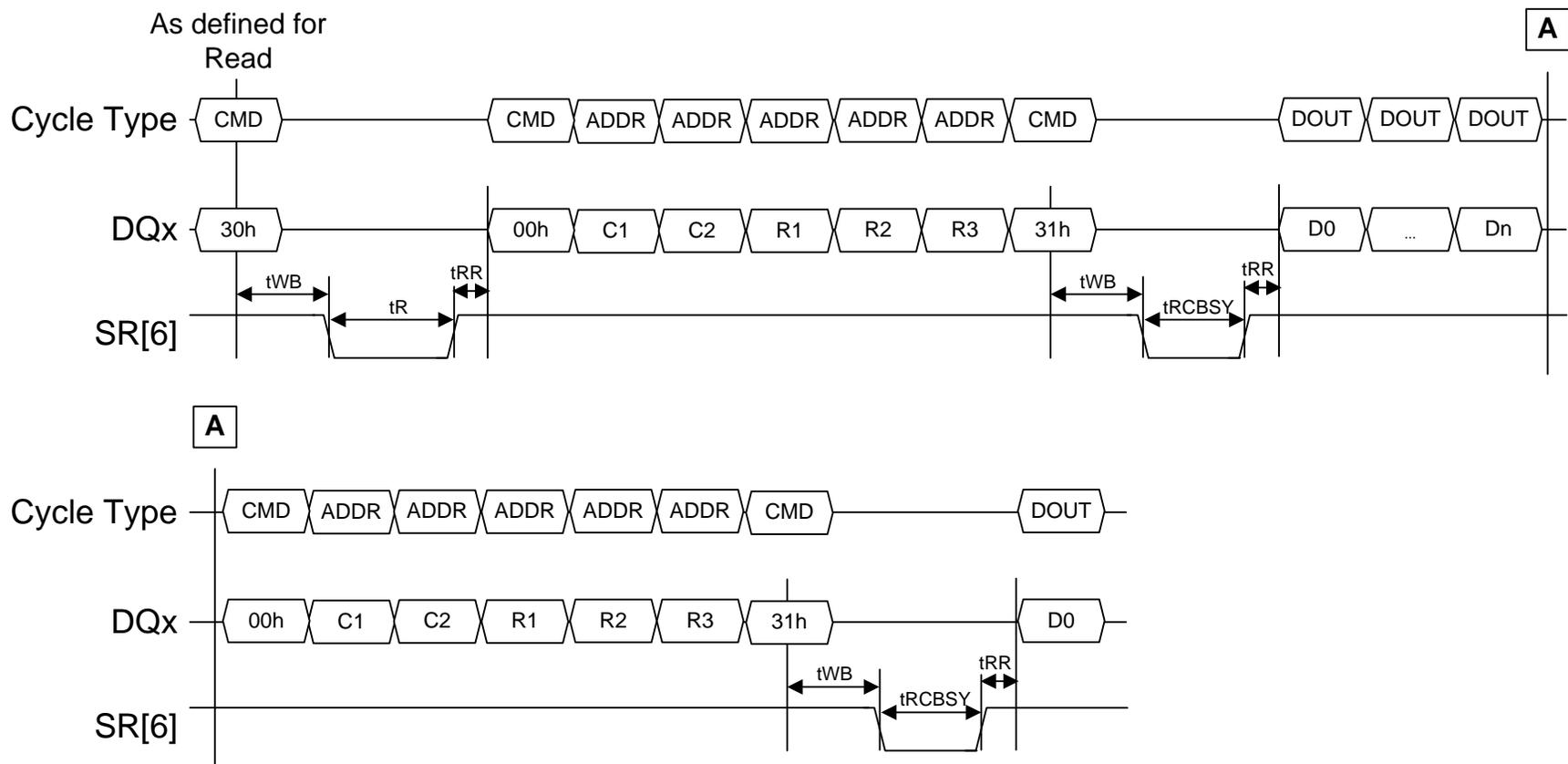
The host shall not issue a Read Cache Sequential (31h) command after the last page of a block is read. If commands are issued to multiple LUNs at the same time, the host shall execute a Read Status Enhanced (78h) command to select the LUN prior to issuing a Read Cache Sequential (31h) or Read Cache End (3Fh) command for that LUN.

Figure 65 defines the Read Cache Sequential behavior and timings for the beginning of the cache operations subsequent to a Read command being issued to the target. Figure 66 defines the Read Cache Random behavior and timings for the beginning of the cache operations subsequent to a Read command being issued to the target. In each case, SR[6] conveys whether the next selected page can be read from the page register.



**Figure 65 Read Cache Sequential timing, start of cache operations**

D0-Dn Data bytes/words read from page requested by the original Read or the previous cache operation.



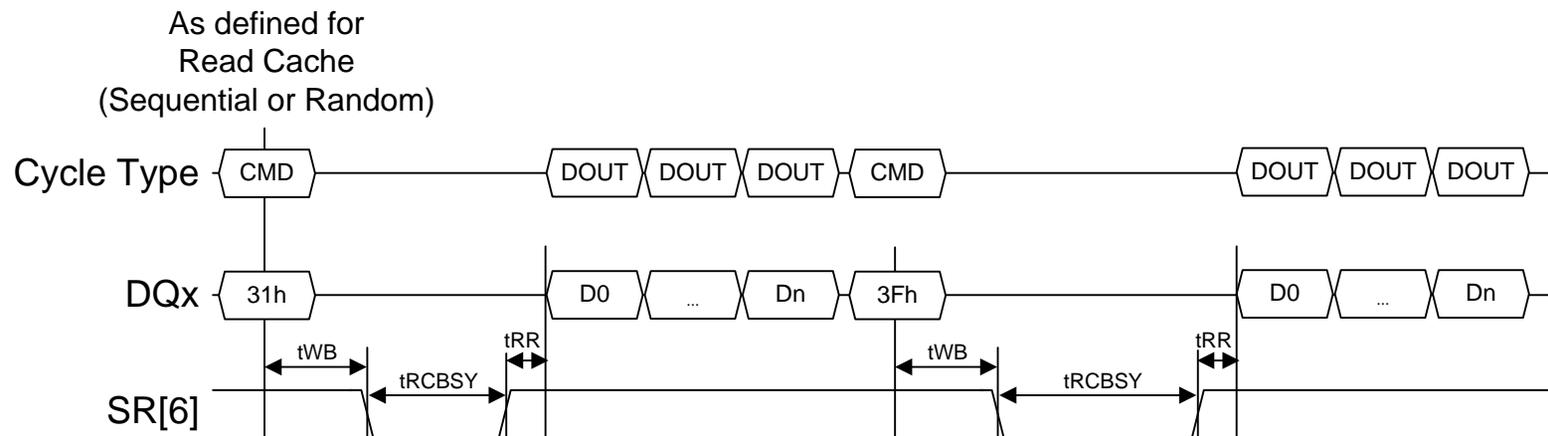
**Figure 66 Read Cache Random timing, start of cache operations**

C1-C2 Column address of the page to retrieve. C1 is the least significant byte. The column address is ignored.

R1-R3 Row address of the page to retrieve. R1 is the least significant byte.

D0-Dn Data bytes/words read from page requested by the original Read or the previous cache operation

Figure 67 defines the Read Cache (Sequential or Random) behavior and timings for the end of cache operations. This applies for both Read Cache Sequential and Read Cache Random. A command code of 3Fh indicates to the target to transfer the final selected page into the page register, without beginning another background read operation.

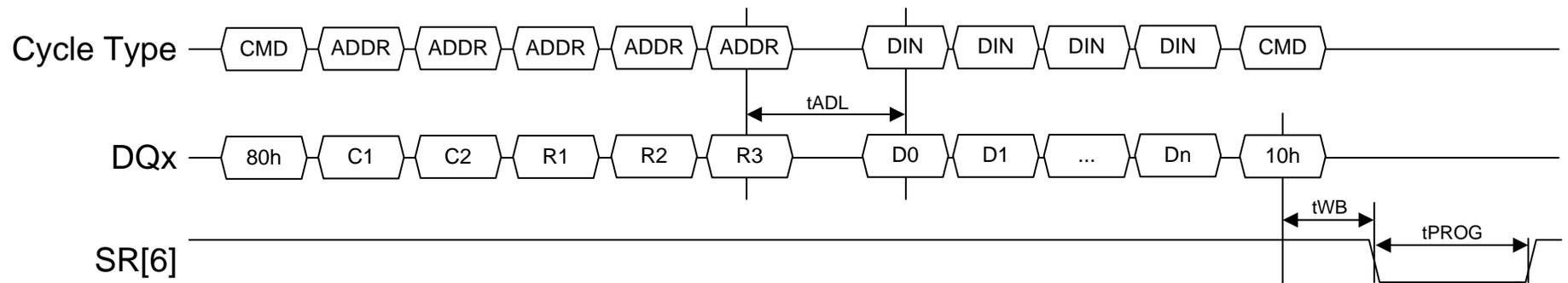


**Figure 67** Read Cache timing, end of cache operations

D0-Dn Data bytes/words read from page requested by the previous cache operation.

## **5.16. Page Program Definition**

The Page Program command transfers a page or portion of a page of data identified by a column address to the page register. The contents of the page register are then programmed into the Flash array at the row address indicated. SR[0] is valid for this command after SR[6] transitions from zero to one until the next transition of SR[6] to zero. Figure 68 defines the Page Program behavior and timings. Writing beyond the end of the page register is undefined.



**Figure 68 Page Program timing**

C1-C2 Column address of the starting buffer location to write data to. C1 is the least significant byte.

R1-R3 Row address of the page being programmed. R1 is the least significant byte.

D0-Dn Data bytes/words to be written to the addressed page.

## 5.17. Page Cache Program Definition

This command is not supported for EZ NAND.

The Page Cache Program function permits a page or portion of a page of data to be written to the Flash array for the specified LUN in the background while the next page to program is transferred by the host to the page register. After the 10h command is issued, all data is written to the Flash array prior to SR[6] being set to one (ready). SR[0] is valid for this command after SR[5] transitions from zero to one until the next transition. SR[1] is valid for this command after SR[6] transitions from zero to one, and this is not the first operation.

Figure 69 and Figure 70 define the Page Cache Program behavior and timings. Note that tPROG at the end of the caching operation may be longer than typical as this time also accounts for completing the programming operation for the previous page. Writing beyond the end of the page register is undefined.

If the program page register clear enhancement is supported, then the host may choose to only clear the page register for the selected LUN and plane address when a Program (80h) command is received. In this case, the tADL time may be longer than defined for the selected timing mode, refer to section 5.7.1.17. Refer to section 5.26.1 for details on how to enable this feature.

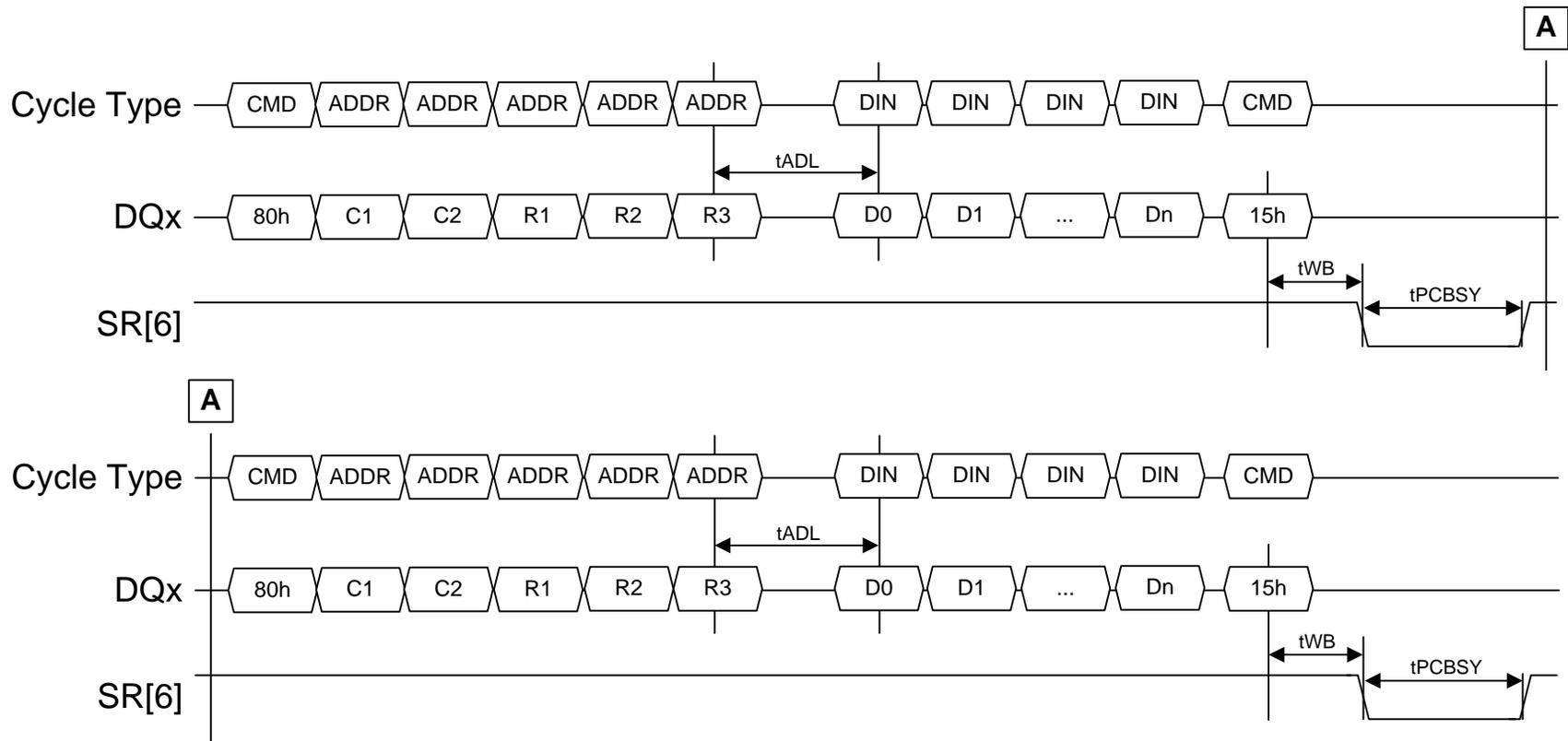
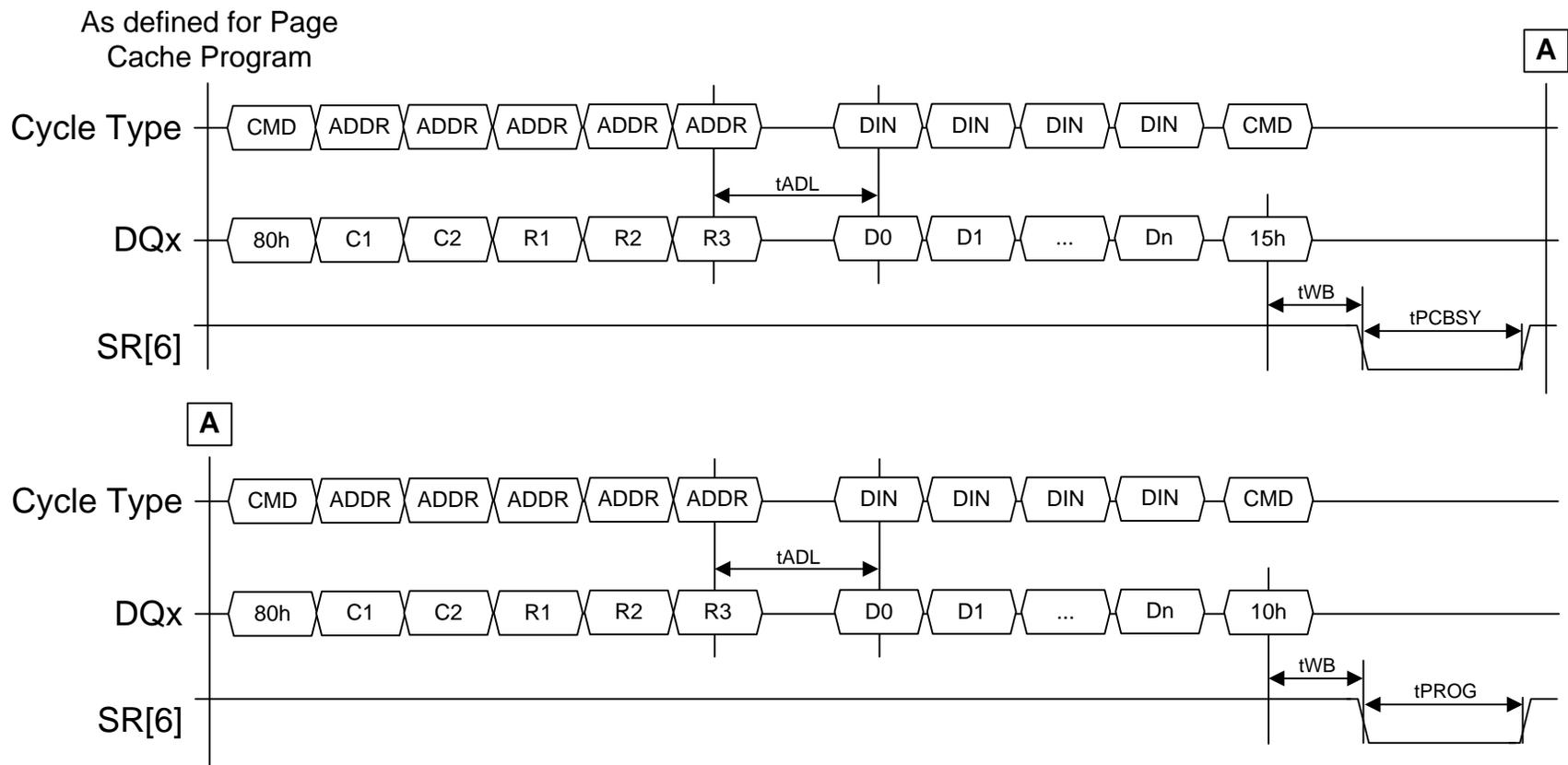


Figure 69 Page Cache Program timing, start of operations



**Figure 70 Page Cache Program timing, end of operations**

C1-C2 Column address of the starting buffer location to write data to. C1 is the least significant byte.

R1-R3 Row address of the page being programmed. R1 is the least significant byte.

D0-Dn Data bytes/words to be written to the addressed page.

## 5.18. Copyback Definition

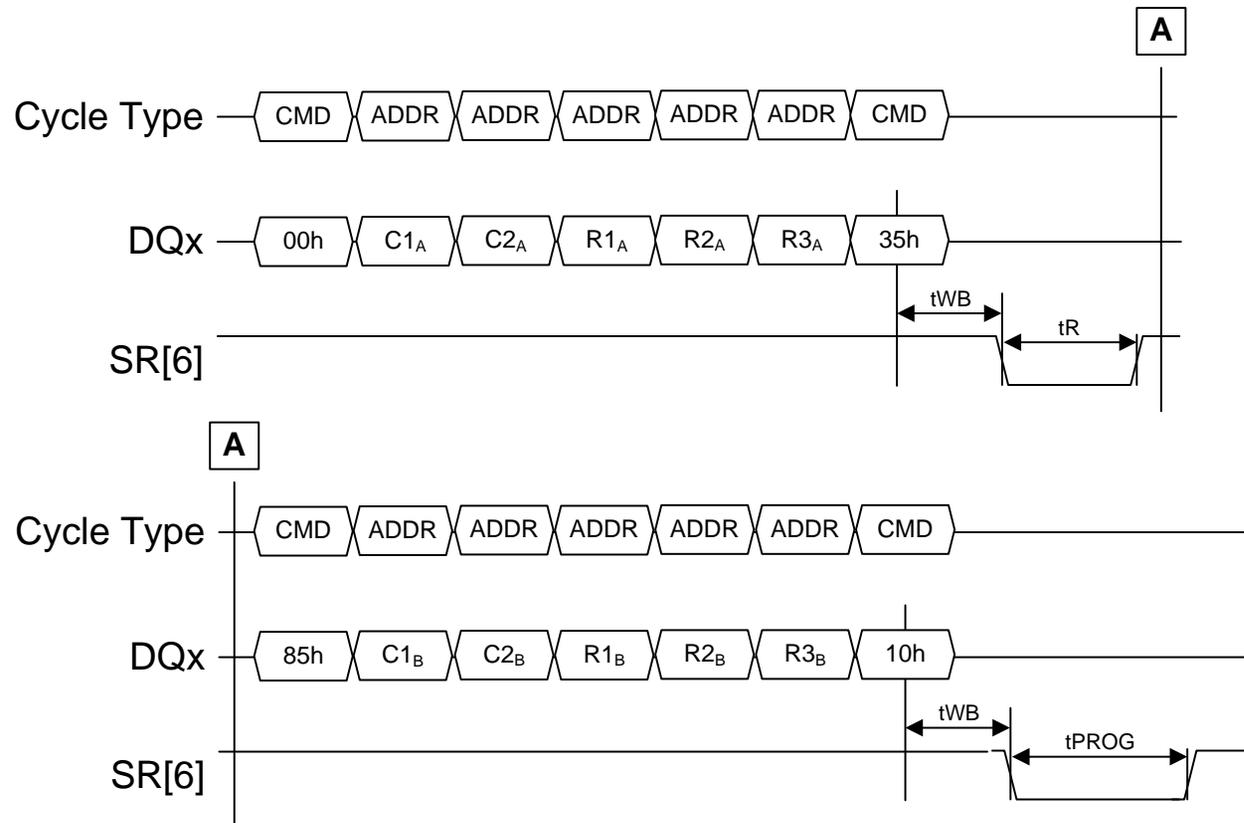
The Copyback function reads a page of data from one location and then moves that data to a second location on the same LUN. If the target supports EZ NAND then the data may be moved to a second location on a different plane or LUN. Refer to the parameter page to determine the destinations support for Copyback. The data read from the first location may be read by the host, including use of Change Read Column. After completing any data read out and issuing Copyback Program, the host may perform data modification using Change Write Column as needed. Figure 71 defines the Copyback behavior and timings.

Copyback uses a single page register for the read and program operation. If the target supports EZ NAND, the buffer in the EZ NAND controller is used for read and program operations and the page register within each LUN is not explicitly accessible.

When multi-plane addressing is supported, the multi-plane address for Copyback Read and Copyback Program for a non-multi-plane Copyback operation shall be the same. If EZ NAND is supported, this restriction may not apply; refer to the parameter page.

Copyback may also have odd/even page restrictions. Specifically, when reading from an odd page, the contents may need to be written to an odd page. Alternatively, when reading from an even page, the contents may need to be written to an even page. Refer to section 5.7.1.3.

This revision of the ONFI specification requires Copyback Adjacency for all implementations. In future ONFI revisions, for EZ NAND implementations it is possible that Copyback Read and Copyback Program are not required to be adjacent. I.e., there may be multiple Copyback Read commands prior to a Copyback Program being issued. This requirement also applies to multi-plane Copyback operations.



**Figure 71 Copyback timing**

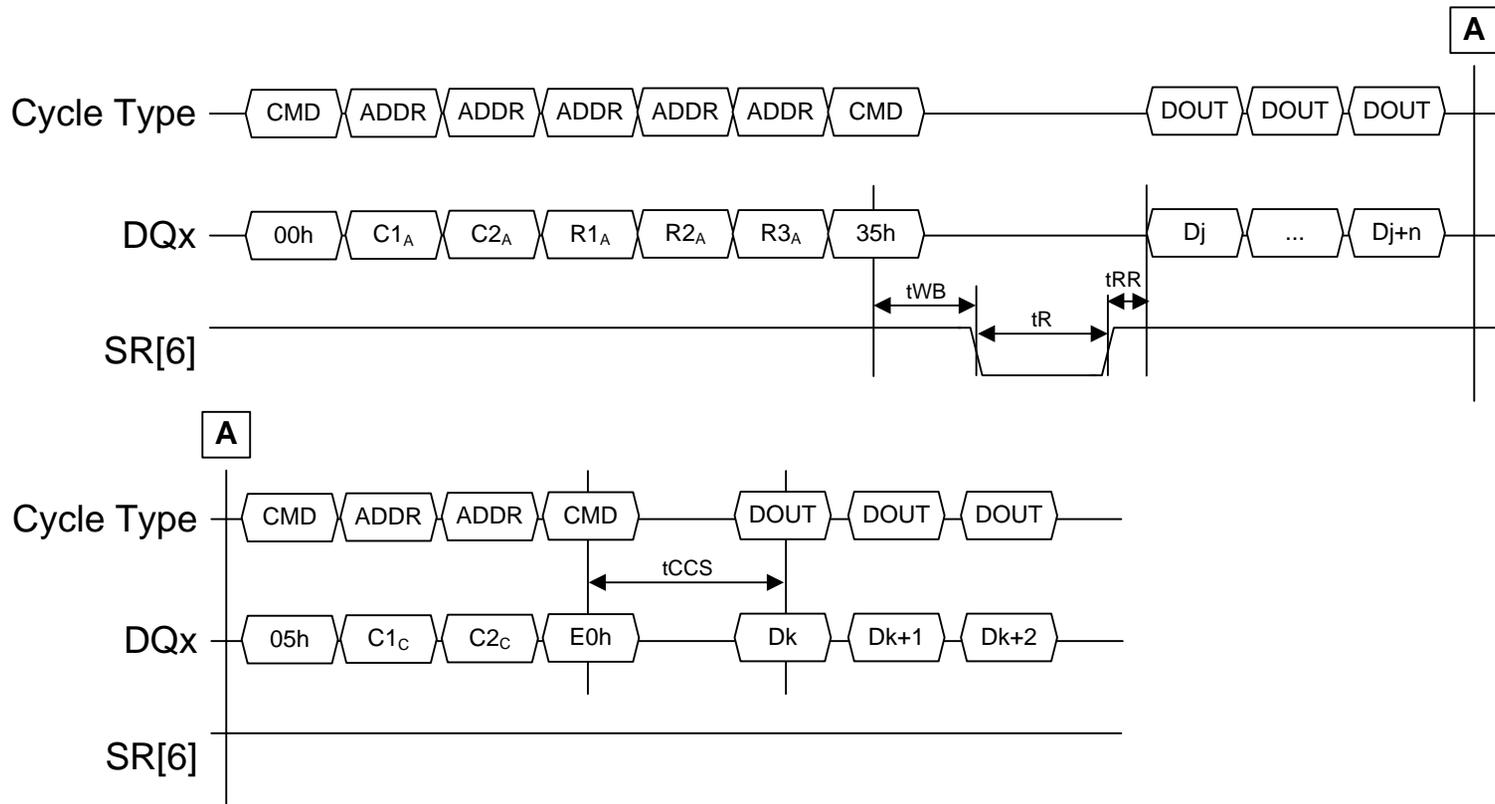
C1-C2<sub>A</sub> Column address of the page to retrieve. C1<sub>A</sub> is the least significant byte.

R1-R3<sub>A</sub> Row address of the page to retrieve. R1<sub>A</sub> is the least significant byte.

C1-C2<sub>B</sub> Column address of the page to program. C1<sub>B</sub> is the least significant byte.

R1-R3<sub>B</sub> Row address of the page to program. R1<sub>B</sub> is the least significant byte.

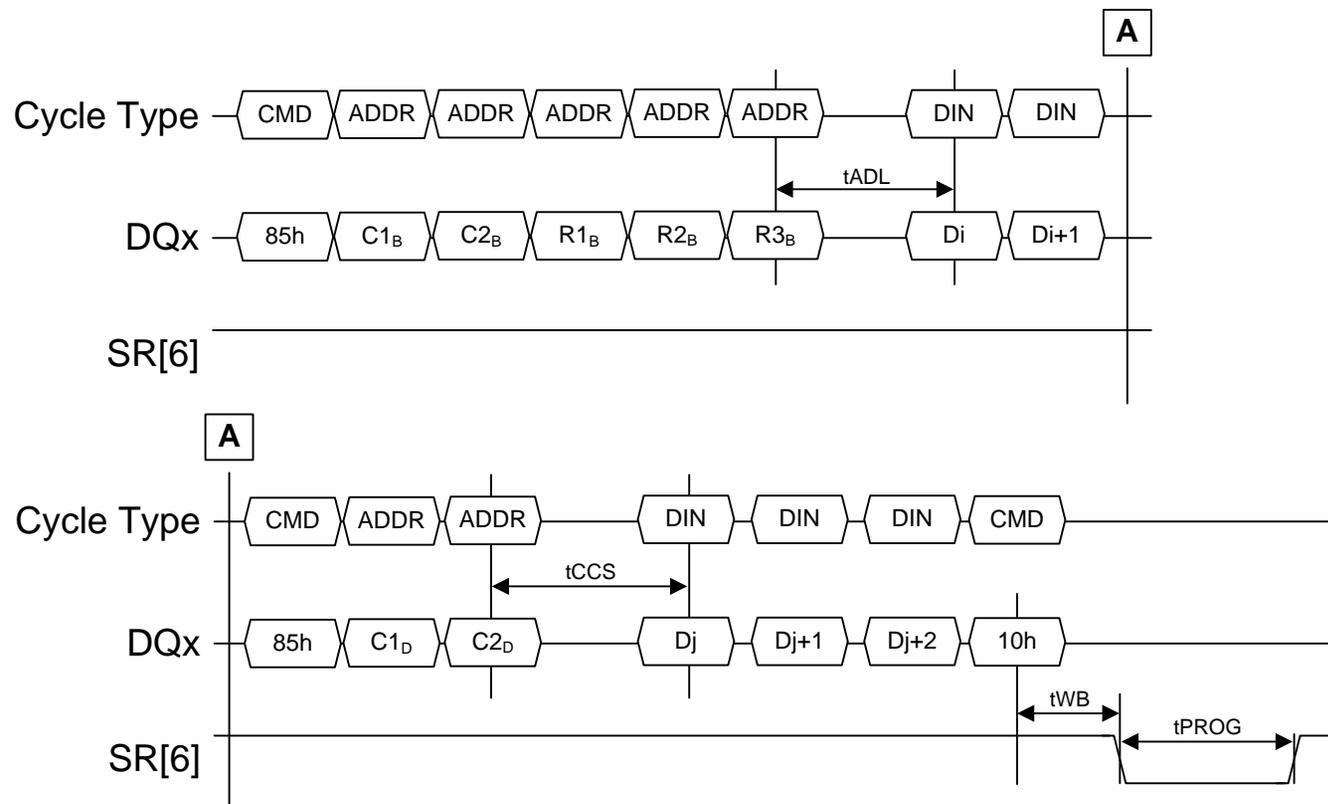
Figure 72 and Figure 73 define Copyback support for data output and data modification.



**Figure 72 Copyback Read with data output**

C1-C2<sub>A</sub> Column address of the page to retrieve. C1<sub>A</sub> is the least significant byte.

R1-R3 <sub>A</sub>	Row address of the page to retrieve. R1 <sub>A</sub> is the least significant byte.
Dj-(Dj+n)	Data bytes read starting at column address specified in C1-C2 <sub>A</sub> .
C1-C2 <sub>C</sub>	Column address of new location (k) to read out from the page register. C1 <sub>C</sub> is the least significant byte.
Dk-Dk+n	Data bytes read starting at column address specified in C1-C2 <sub>C</sub> .



**Figure 73 Copyback Program with data modification**

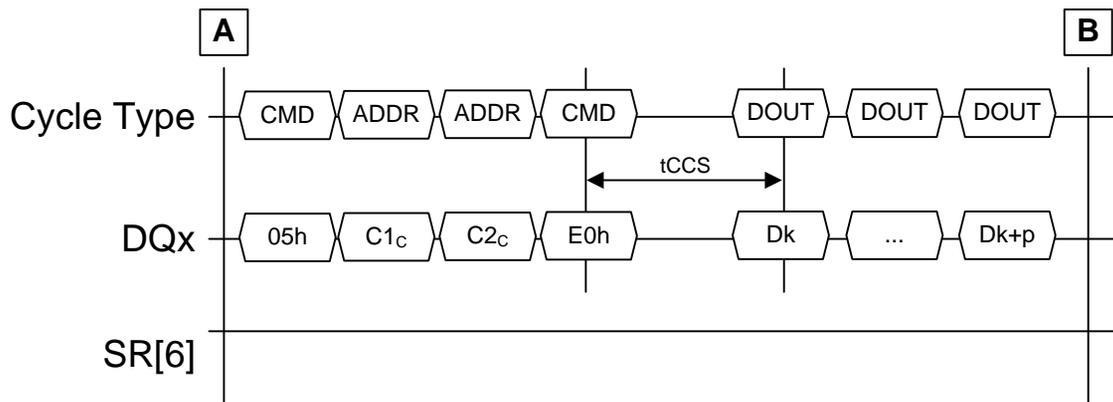
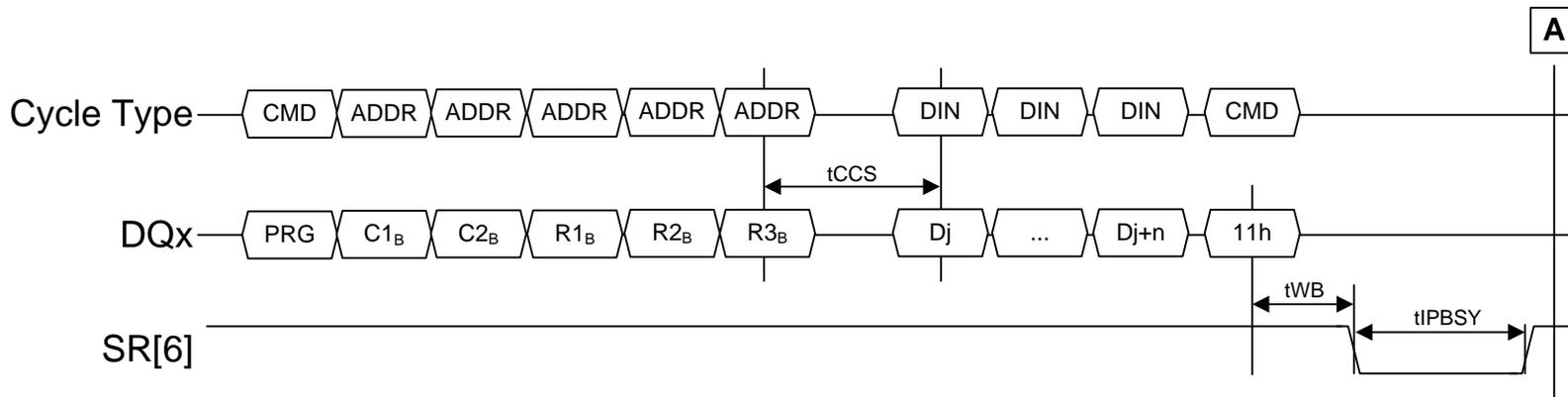
- C1-C2<sub>B</sub> Column address of the page to program. C1<sub>B</sub> is the least significant byte.
- R1-R3<sub>B</sub> Row address of the page to program. R1<sub>B</sub> is the least significant byte.
- Di-Di+n Data bytes overwritten in page register starting at column address specified in C1-C2<sub>B</sub>.
- C1-C2<sub>D</sub> Column address of new location (j) to overwrite data at in the page register. C1<sub>D</sub> is the least significant byte.

Dj-Dj+n      Data bytes overwritten starting at column address specified in C1-C2<sub>D</sub>.

### 5.19.      **Small Data Move**

If the Small Data Move command is supported, as indicated in the parameter page, then the host may transfer data to the page register in increments that are less than the page size of the device for both Program and Copyback operations (including multi-plane Program and Copyback operations). The host may also read data out as part of the operation. If the Small Data Move is a program operation with no data output, then the 80h opcode may be used for the first cycles. For Copyback and program operations that include data output, the 85h opcode shall be used for the first cycles.

Figure 74 defines the data modification portion of a Program or Copyback Program with small data moves; this sequence may be repeated as necessary to complete the data transfer. Figure 75 defines the final program operation that is used to complete the Program or Copyback Program with small data move operation. The row address (R1<sub>B</sub> – R3<sub>B</sub>) shall be the same for all program portions of the sequence destined for the same plane address. The function of the 11h command in a small data move operation is to flush any internal data pipeline in the device prior to resuming data output.

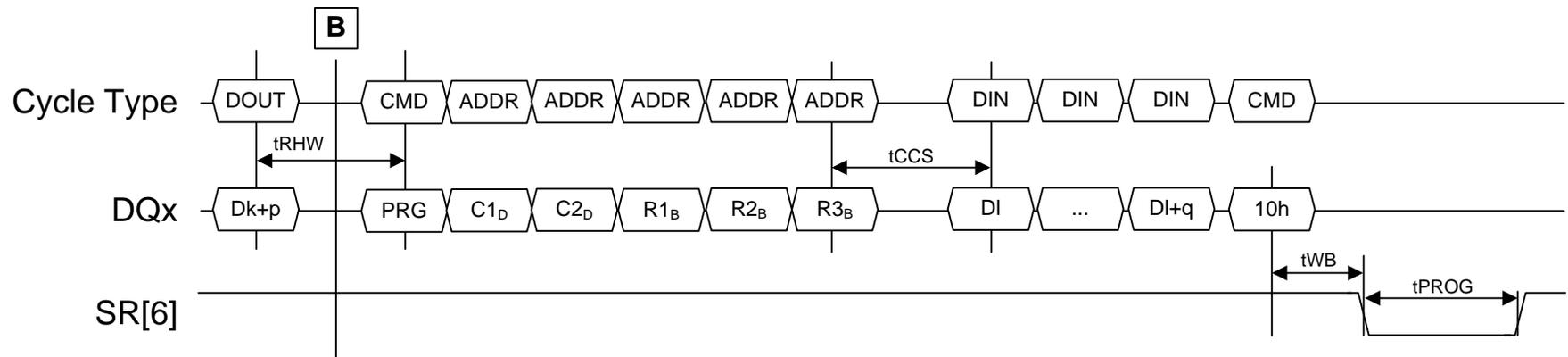


**Figure 74 Small data moves, data modification**

- PRG Program command, either 80h or 85h. Following any data output, the command shall be 85h.
- C1-C<sub>2B</sub> Column address to write to in the page register. C1<sub>B</sub> is the least significant byte.
- R1-R<sub>3B</sub> Row address of the page to program. R1<sub>B</sub> is the least significant byte.

- D<sub>j</sub>-(D<sub>j</sub>+n) Data bytes to update in the page register starting at column address specified in C1-C2<sub>B</sub>.
- C1-C2<sub>C</sub> Column address of the byte/word in the page register to retrieve. C1<sub>C</sub> is the least significant byte.
- D<sub>k</sub>-(D<sub>k</sub>+p) Data bytes read starting at column address specified in C1-C2<sub>C</sub>.

**NOTE:** If Change Read Column Enhanced is supported, this command may be substituted for Change Read Column in Figure 74. Use of the Change Read Column (Enhanced) command and data output in this flow is optional; this flow may be used to incrementally transfer data for a Program or Copyback Program.



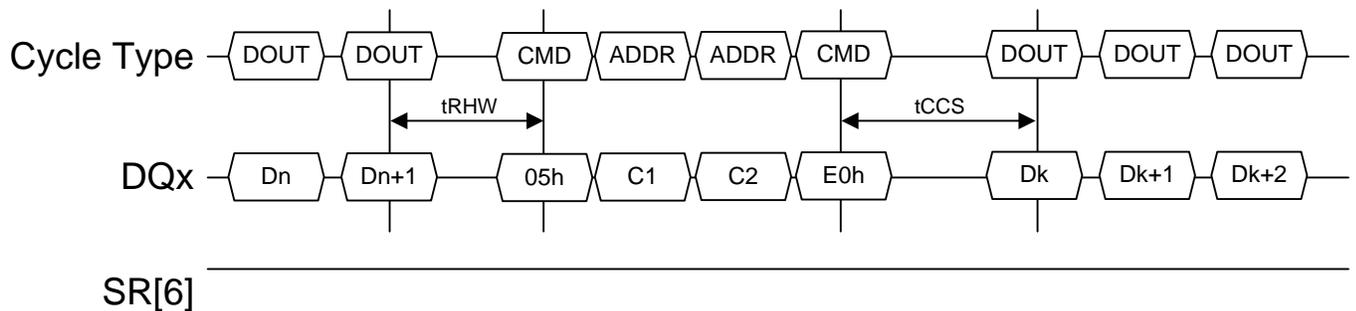
**Figure 75 Small data moves, end**

- PRG Program command, either 80h or 85h. 85h shall be used if there is any data output as part of the command.
- C1-C2<sub>D</sub> Column address to write to in the page register. C1<sub>D</sub> is the least significant byte.
- R1-R3<sub>B</sub> Row address of the page to program. R1<sub>B</sub> is the least significant byte.
- DI-(DI+q) Data bytes to update in the page register starting at column address specified in C1-C2<sub>D</sub>.

## 5.20. Change Read Column Definition

The Change Read Column function changes the column address from which data is being read in the page register for the selected LUN. Change Read Column shall only be issued when the LUN is in a read idle condition. Figure 76 defines the Change Read Column behavior and timings.

The host shall not read data from the LUN until  $t_{CCS}$  ns after the E0h command is written to the LUN. Refer to Figure 76.



**Figure 76 Change Read Column timing**

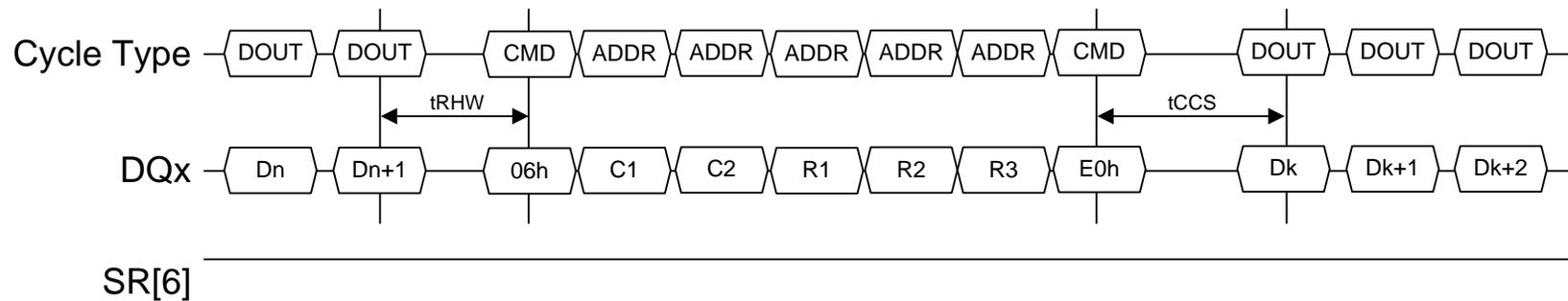
- Dn Data bytes read prior to the column address change.
- C1-C2 New column address to be set for subsequent data transfers. C1 is the least significant byte.
- Dk Data bytes being read starting with the new addressed column.

## 5.21. Change Read Column Enhanced Definition

The Change Read Column Enhanced function changes the LUN address, plane address and column address from which data is being read in a page previously retrieved with the Read command. This command is used when independent LUN operations or multi-plane operations are being performed such that the entire address for the new column needs to be given. Figure 77 defines the Change Read Column Enhanced behavior and timings.

The Change Read Column Enhanced command shall not be issued by the host unless it is supported as indicated in the parameter page. Change Read Column Enhanced shall not be issued while Target level data output commands (Read ID, Read Parameter Page, Read Unique ID, Get Features) are executing or immediately following Target level commands.

Change Read Column Enhanced causes idle LUNs (SR[6] is one) that are not selected to turn off their output buffers. This ensures that only the LUN selected by the Change Read Column Enhanced command responds to subsequent data output. If unselected LUNs are active (SR[6] is zero) when Change Read Column Enhanced is issued, then the host shall issue a Read Status Enhanced (78h) command prior to subsequent data output to ensure all LUNs that are not selected turn off their output buffers.



**Figure 77 Change Read Column Enhanced timing**

Dn Data bytes read prior to the row and column address change.

C1-C2 New column address to be set for subsequent data transfers. C1 is the least significant byte.

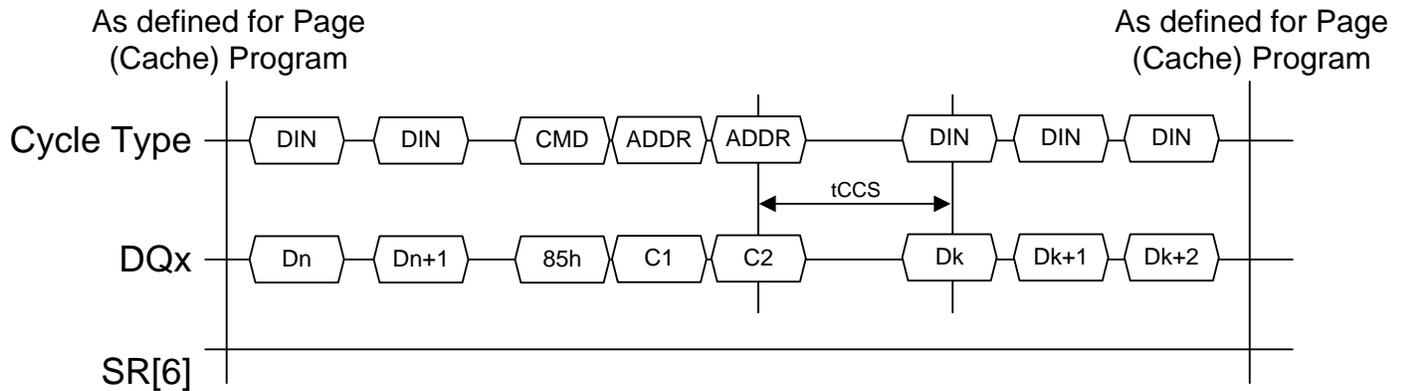
R1-R3 New row address to be set for subsequent data transfers. R1 is the least significant byte.

Dk Data bytes being read starting with the new addressed row and column.

## 5.22. Change Write Column Definition

The Change Write Column function changes the column address being written to in the page register for the selected LUN. Figure 78 defines the Change Write Column behavior and timings.

The host shall not write data to the LUN until  $t_{CCS}$  ns after the last column address is written to the LUN. Refer to Figure 76.



**Figure 78 Change Write Column timing**

C1-C2 New column address to be set for subsequent data transfers. C1 is the least significant byte.

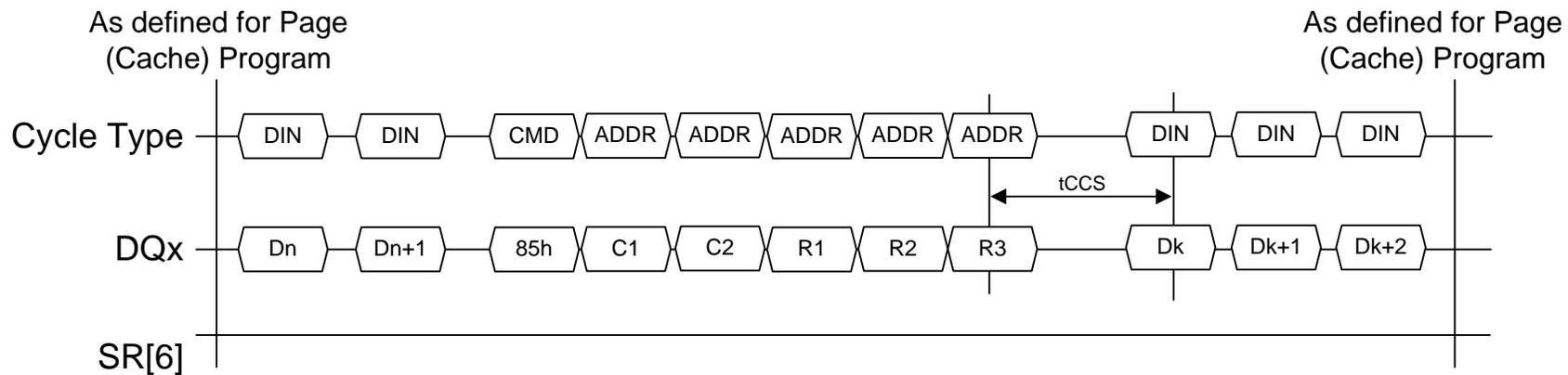
Dn Data bytes being written to previous addressed column

Dk Data bytes being written starting with the new addressed column

## 5.23. Change Row Address Definition

The Change Row Address function changes the row and column address being written to for the selected LUN. This mechanism may be used to adjust the block address, page address, and column address for a Program that is in execution. The LUN and plane address shall be the same as the Program that is in execution. Figure 79 defines the Change Row Address behavior and timings.

The host shall not write data to the LUN until  $t_{CCS}$  ns after the last row address is written to the LUN. Refer to Figure 79.



**Figure 79 Change Row Address timing**

C1-C2 New column address to be set for subsequent data transfers. C1 is the least significant byte.

R1-R3 Row address of the page being programmed. The LUN address and plane address shall be the same as the Program in execution. R1 is the least significant byte.

Dn Data bytes being written prior to row address change; will be written to new row address

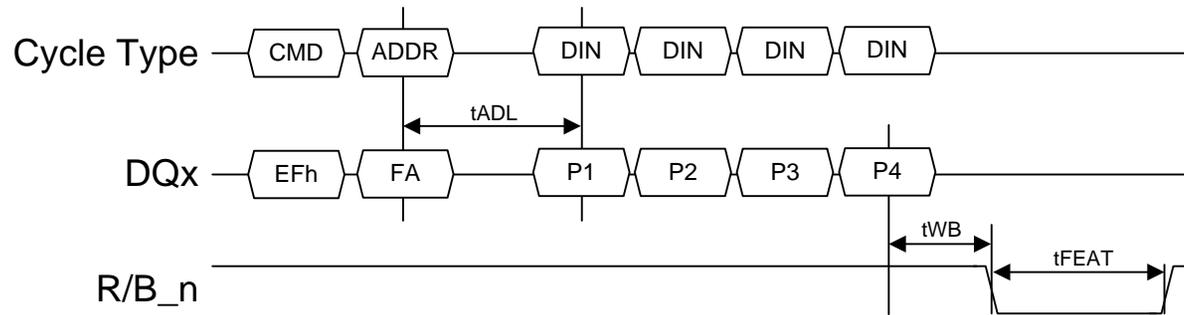
Dk Data bytes being written to the new block and page, starting with the newly addressed column

## **5.24. Set Features Definition**

The Set Features function modifies the settings of a particular feature. For example, this function can be used to enable a feature that is disabled at power-on. Parameters are always transferred on the lower 8-bits of the data bus. Figure 80 defines the Set Features behavior and timings.

When issuing Set Features in the source synchronous data interface, each data byte is transmitted twice. The device shall only latch one copy of each data byte. See section 4.3.2.3.

Set Features is used to change the timing mode and data interface type. When changing the timing mode, the device is busy for tITC, not tFEAT. During the tITC time the host shall not poll for status.



**Figure 80 Set Features timing**

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

FA Feature address identifying feature to modify settings for.

P1-P4 Parameters identifying new settings for the feature specified.

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

Refer to section 5.26 for the definition of features and sub feature parameters.

## 5.25. Get Features Definition

The Get Features function is the mechanism the host uses to determine the current settings for a particular feature. This function shall return the current settings for the feature (including modifications that may have been previously made with the Set Features function). Parameters are always transferred on the lower 8-bits of the data bus. After reading the first byte of data, the host shall complete reading all desired data before issuing another command (including Read Status or Read Status Enhanced). Figure 81 defines the Get Features behavior and timings.

When issuing Get Features in the source synchronous data interface, each data byte is received twice. The host shall only latch one copy of each data byte. See section 4.3.2.5.

If Read Status is used to monitor when the tFEAT time is complete, the host shall issue a command value of 00h to begin transfer of the feature data starting with parameter P1.

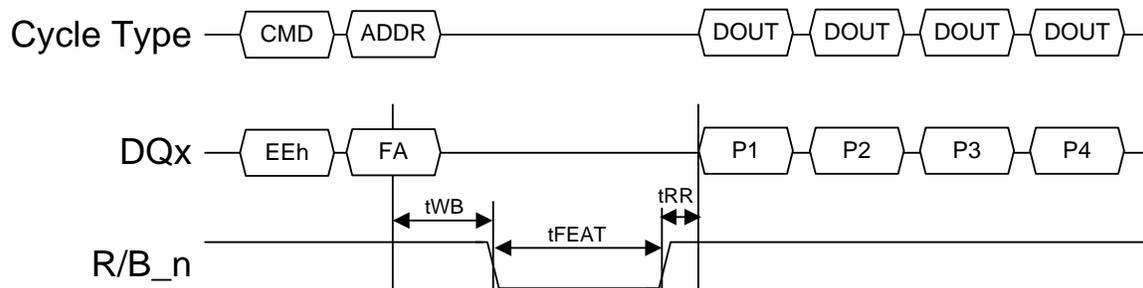


Figure 81 Get Features timing

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 5.26 for the definition of features and sub feature parameters.

## 5.26. Feature Parameter Definitions

If the Set Features and Get Features commands are not supported by the Target, then no feature parameters are supported. Additionally, the Target only supports feature parameters defined in ONFI specification revisions that the Target complies with.

Feature settings are volatile across power cycles. For each feature setting, whether the value across resets is retained is explicitly stated.

Feature Address	Description
00h	Reserved
01h	Timing Mode
02h-0Fh	Reserved

10h	I/O Drive Strength
11h-1Fh	Reserved for programmable I/O settings
20h-4Fh	Reserved
50h	EZ NAND control
51h-5Fh	Reserved
60h-7Fh	Reserved for Block Abstracted NAND
80h-FFh	Vendor specific

### 5.26.1. Timing Mode

This setting shall be supported if the Target complies with ONFI specification revision 1.0.

The Data Interface setting is not retained across Reset (FFh); after a Reset (FFh) the Data Interface shall be asynchronous. All other settings for the timing mode are retained across Reset (FFh) and Synchronous Reset (FCh) commands. If the Reset (FFh) command is issued when the Data Interface is configured as source synchronous, then the host shall use the asynchronous data interface with Timing Mode 0 until a new data interface and/or timing mode is selected with Set Features. Hosts shall only set a timing mode that is explicitly shown as supported in the Read Parameter Page.

The results of the host using Set Features to transition from the source synchronous data interface to the asynchronous data interface is indeterminate. To transition to the asynchronous data interface, the host should use the Reset (FFh) command.

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** Set to the numerical value of the maximum timing mode in use by the host. Default power-on value is 0h.
- Data Interface** 00b = asynchronous (default power-on value)  
01b = source synchronous  
10-11b = Reserved
- PC** The Program Clear bit controls the program page register clear enhancement which defines the behavior of clearing the page register when a Program (80h) command is received. If cleared to zero, then the page register(s) for each LUN that is part of the target is cleared when the Program (80h) command is received. If set to one, then only the page register for the LUN and interleave address selected with the Program (80h) command is cleared and the tADL time for Program commands is as reported in the parameter page.
- Reserved / R** Reserved values shall be cleared to zero by the host. Targets shall not be sensitive to the value of reserved fields.

### 5.26.2. I/O Drive Strength

This setting shall be supported if the Target supports the source synchronous data interface. The I/O drive strength setting shall be retained across Reset (FFh) and Synchronous Reset (FCh) commands. The power-on default drive strength value is the Nominal (10b) setting.

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  
                                   01b = 25 Ohm  
                                   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.

### 5.26.3. EZ NAND control

This setting shall be supported if the device supports EZ NAND. This feature is used to control settings for the EZ NAND device.

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)        If set to one, then the EZ NAND device shall not automatically perform retries. If cleared to zero, then the EZ NAND device may automatically perform retries during error conditions at its discretion. If automatic retries are disabled, the device may exceed the UBER specified. Automatic retries shall only be disabled if the device supports this capability as indicated in the parameter page. If an EZ NAND controller executes an automatic retry, the typical page read time (tR) may be exceeded.

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

## 6. Multi-plane Operations

A LUN may support multi-plane read, program and erase operations. Multi-plane operations are when multiple commands of the same type are issued to different blocks on the same LUN. Refer to section 5.7.1.28 for addressing restrictions with multi-plane operations. There are two methods for multi-plane operations: concurrent and overlapped.

When performing multi-plane operations, the operations/functions shall be the same type. The functions that may be used in multi-plane operations are:

- Page Program
- Copyback Read and Program
- Block Erase
- Read

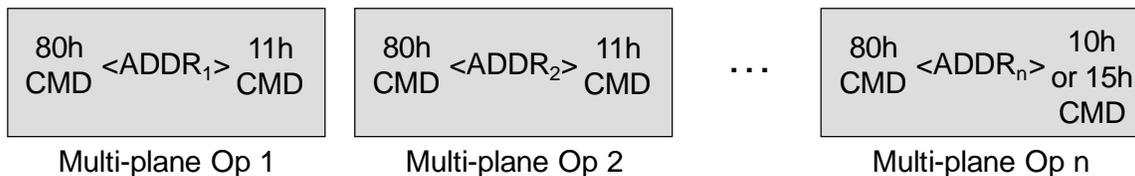
### 6.1. Requirements

When supported, the plane address comprises the lowest order bits of the block address as shown in Figure 19. The LUN and page addresses are required to be the same. The block address (other than the plane address bits) may be required to be the same, refer to section 5.7.1.28.

For copyback program operations, the restrictions are the same as for a multi-plane program operation. However, copyback reads shall be previously issued to the same plane addresses as those in the multi-plane copyback program operations. The reads for copyback may be issued non-multi-plane or multi-plane. If the reads are non-multi-plane then the reads may have different page addresses. If the reads are multi-plane then the reads shall have the same page addresses.

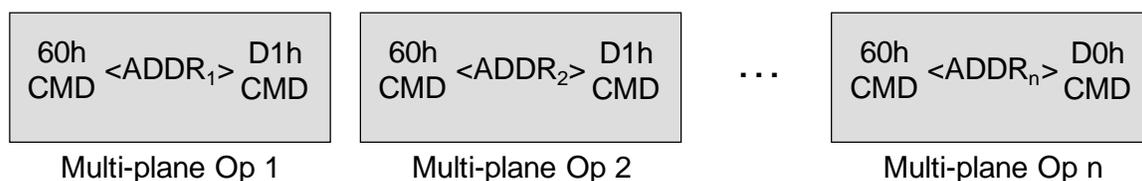
Multi-plane operations enable operations of the same type to be issued to other blocks on the same LUN. There are two methods for multi-plane operations: concurrent and overlapped. The concurrent multi-plane operation waits until all command, address, and data are entered for all plane addresses before accessing the Flash array. The overlapped multi-plane operation begins its operation immediately after the command, address and data are entered and performs it in the background while the next multi-plane command, address, and data are entered.

The plane address component of each address shall be distinct. A single multi-plane (cached) program operation is shown in Figure 82. Between “Multi-plane Op 1” and “Multi-plane Op n”, all plane addresses shall be different from each other. After the 10h or 15h (cached) command cycle is issued, previously issued plane addresses can be used in future multi-plane operations.



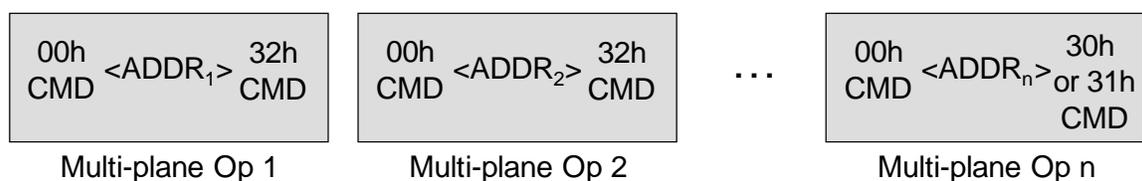
**Figure 82 Multi-plane Program (Cache)**

For multi-plane erase operations, the plane address component of each address shall be distinct. A single multi-plane erase operation is shown in Figure 83. Between “Multi-plane Op 1” and “Multi-plane Op n”, all plane addresses shall be different from each other. After the D0h command cycle is issued, previously issued plane addresses can be used in future multi-plane operations.



**Figure 83 Multi-plane Erase**

The plane address component of each address shall be distinct. A single multi-plane read (cache) operation is shown in Figure 84. Between “Multi-plane Op 1” and “Multi-plane Op n”, all plane addresses shall be different from each other. After the 30h or 31h (cached) command cycle is issued, previously issued plane addresses can be used in future multi-plane operations.



**Figure 84 Multi-plane Read (Cache)**

## 6.2. Status Register Behavior

Some status register bits are independent per plane address. Other status register bits are shared across the entire LUN. This section defines when status register bits are independent per plane address. This is the same for concurrent and overlapped operations.

For multi-plane program and erase operations, the FAIL/FAILC bits are independent per plane address. Table 50 lists whether a bit is independent per plane address or shared across the entire LUN for multi-plane operations.

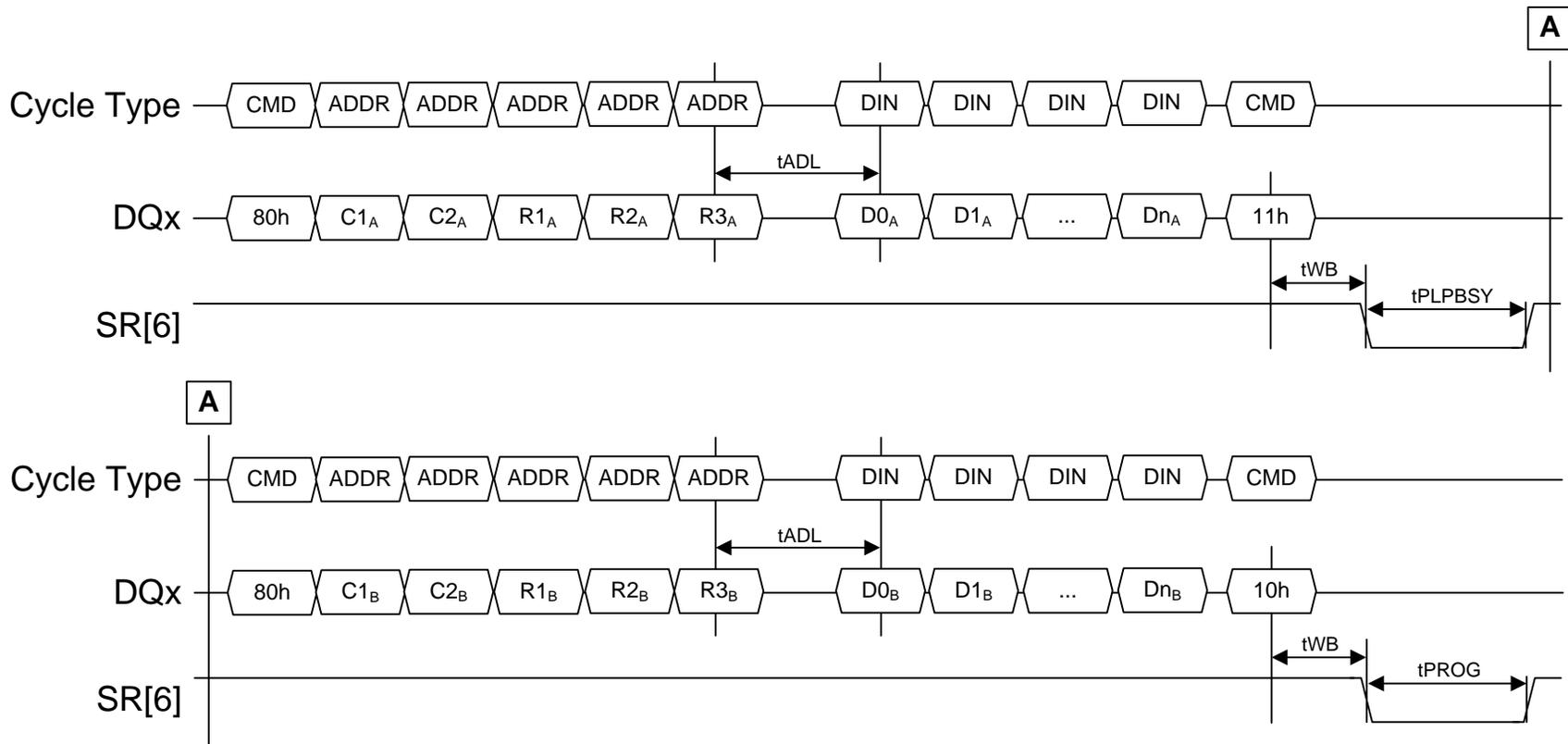
Value	7	6	5	4	3	2	1	0
Status Register	WP <sub>n</sub>	RDY	ARDY	VSP	R	R	FAILC	FAIL
Independent	N	N	N	N	N	N	Y	Y

**Table 50 Independent Status Register bits**

## 6.3. Multi-plane Page Program

The Page Program command transfers a page or portion of a page of data identified by a column address to the page register. The contents of the page register are then programmed into the Flash array at the row address indicated. With a multi-plane operation, multiple programs can be issued back to back to the LUN, with a shorter busy time between issuance of the next program operation. Figure 85 defines the behavior and timings for two multi-plane page program commands.

Cache operations may be used when doing multi-plane page program operations, as shown, if supported by the target as indicated in the parameter page. Refer to section 5.7.1.27.



**Figure 85 Multi-plane Page Program timing**

- C1<sub>A</sub>-C2<sub>A</sub> Column address for page A. C1<sub>A</sub> is the least significant byte.
- R1<sub>A</sub>-R3<sub>A</sub> Row address for page A. R1<sub>A</sub> is the least significant byte.
- D0<sub>A</sub>-Dn<sub>A</sub> Data to program for page A.
- C1<sub>B</sub>-C2<sub>B</sub> Column address for page B. C1<sub>B</sub> is the least significant byte.

R1<sub>B</sub>-R3<sub>B</sub> Row address for page B. R1<sub>B</sub> is the least significant byte.

D0<sub>B</sub>-Dn<sub>B</sub> Data to program for page B.

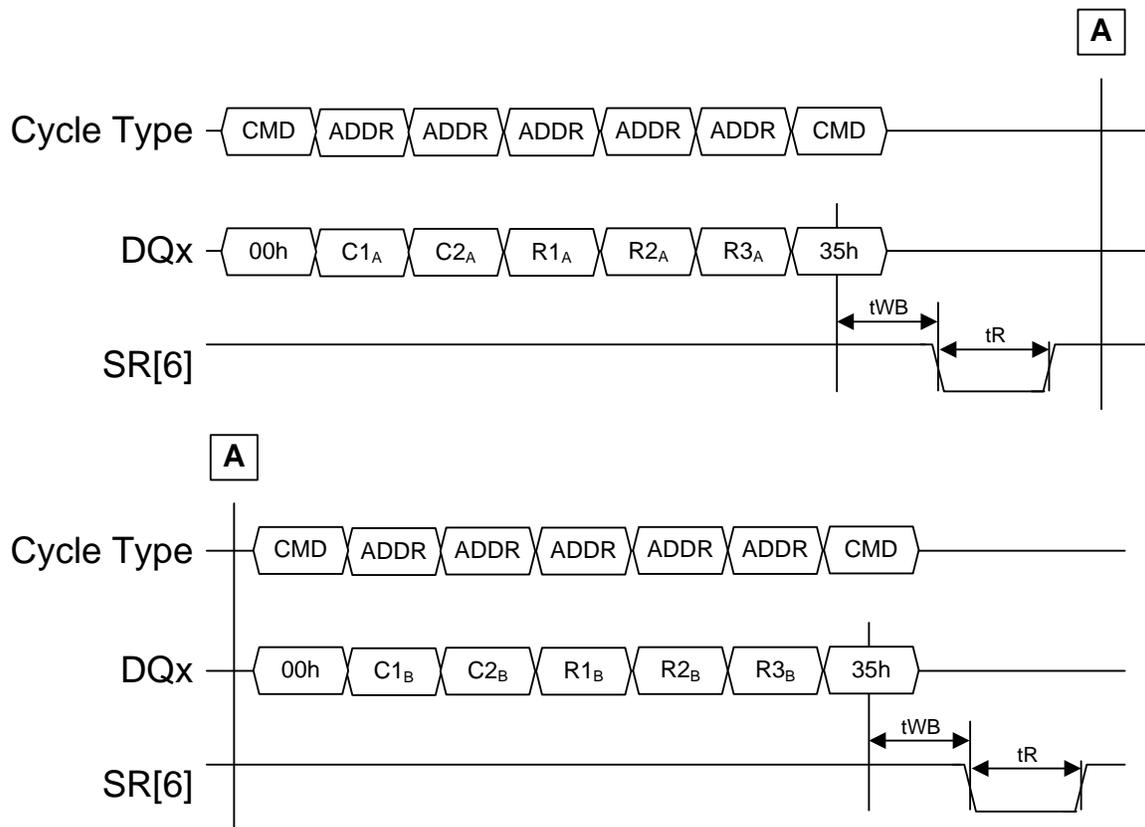
The row addresses for page A and B shall differ in the plane address bits.

Finishing a multi-plane program with a command cycle of 15h rather than 10h indicates that this is a cache operation. The host shall only issue a command cycle of 15h to complete an multi-plane program operation if program cache is supported with multi-plane program operations, as described in section 5.7.1.27.

## 6.4. Multi-plane Copyback Read and Program

The Copyback function reads a page of data from one location and then moves that data to a second location. With a multi-plane operation, the Copyback Program function can be issued back to back to the target, with a shorter busy time between issuance of the next Copyback Program. Figure 86, Figure 87, and Figure 88 define the behavior and timings for two Copyback Program operations. The reads for the Copyback Program may or may not be multi-plane. Figure 86 defines the non-multi-plane read sequence and Figure 87 defines the multi-plane read sequence.

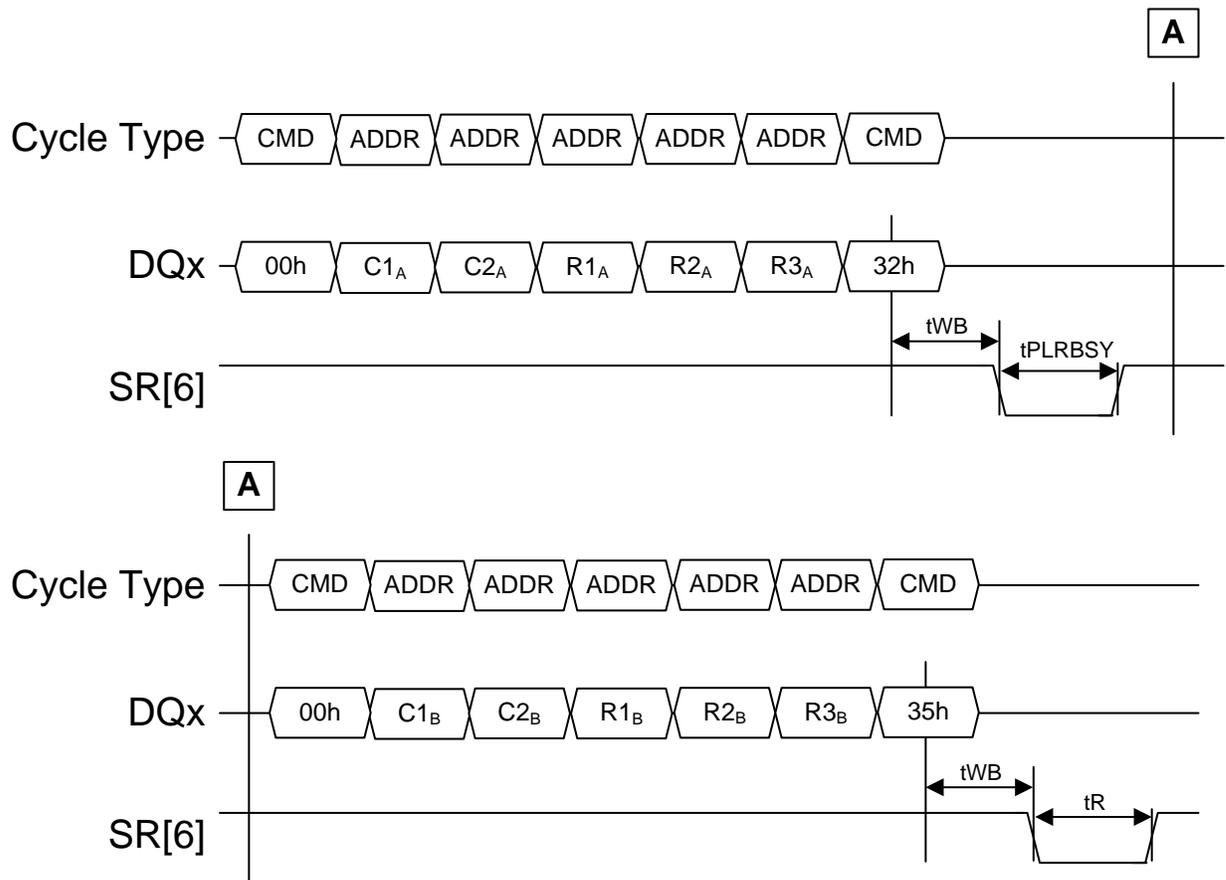
The plane addresses used for the Copyback Read operations (regardless of multi-plane) shall be the same as the plane addresses used in the subsequent multi-plane Copyback Program operations. If EZ NAND is supported, this restriction may not apply; refer to the parameter page.



**Figure 86 Non-multi-plane Copyback Read timing for multi-plane Copyback Program**

- C1<sub>A</sub>-C2<sub>A</sub> Column address for source page A. C1<sub>A</sub> is the least significant byte.
- R1<sub>A</sub>-R3<sub>A</sub> Row address for source page A. R1<sub>A</sub> is the least significant byte.
- C1<sub>B</sub>-C2<sub>B</sub> Column address for source page B. C1<sub>B</sub> is the least significant byte.
- R1<sub>B</sub>-R3<sub>B</sub> Row address for source page B. R1<sub>B</sub> is the least significant byte.

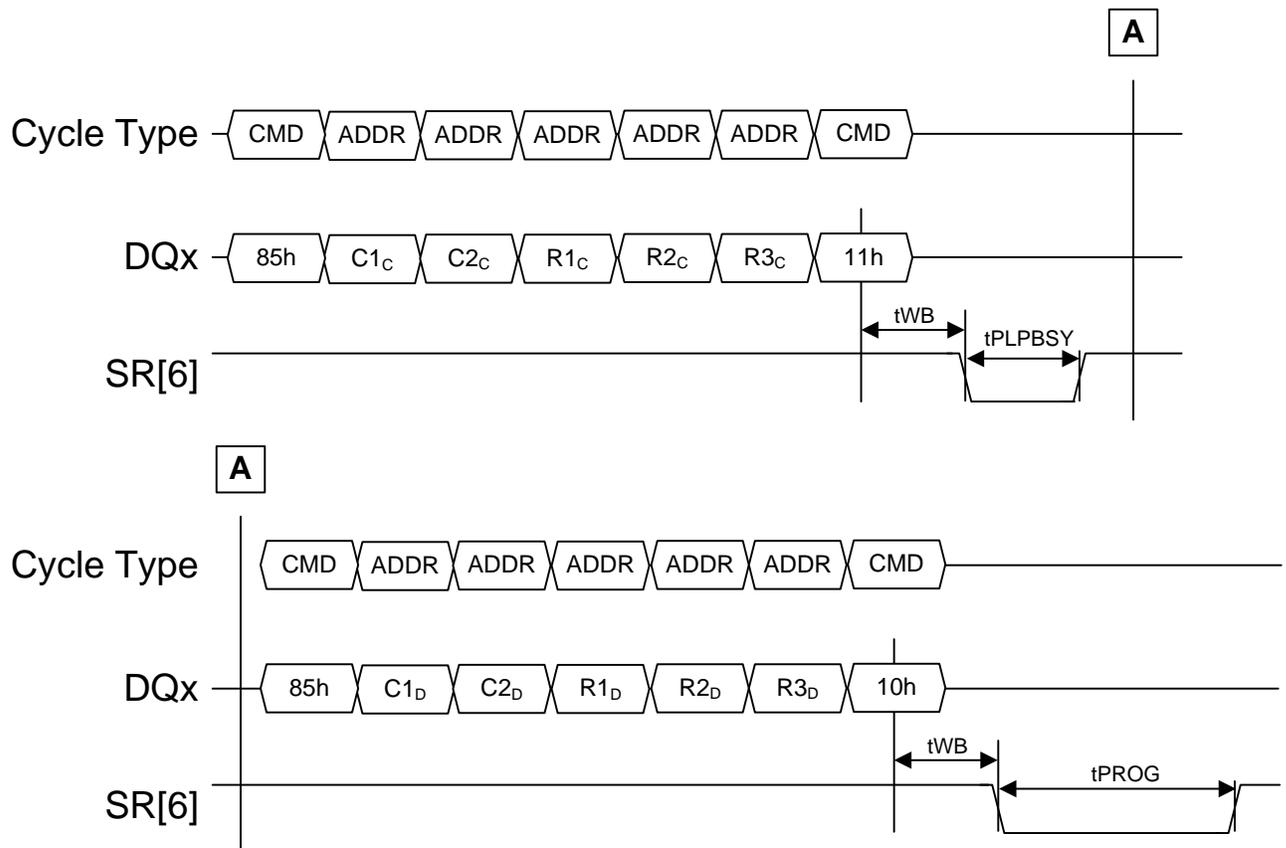
The row addresses for all source pages shall differ in their plane address bits.



**Figure 87 Multi-plane Copyback Read timing for Multi-plane Copyback Program**

- C1<sub>A</sub>-C2<sub>A</sub> Column address for source page A. C1<sub>A</sub> is the least significant byte.
- R1<sub>A</sub>-R3<sub>A</sub> Row address for source page A. R1<sub>A</sub> is the least significant byte.
- C1<sub>B</sub>-C2<sub>B</sub> Column address for source page B. C1<sub>B</sub> is the least significant byte.
- R1<sub>B</sub>-R3<sub>B</sub> Row address for source page B. R1<sub>B</sub> is the least significant byte.

The row addresses for all source pages shall differ in their plane address bits. The source page addresses shall be the same for multi-plane reads.



**Figure 88 Multi-plane Copyback Program**

C1<sub>c</sub>-C2<sub>c</sub> Column address for destination page C. C1<sub>c</sub> is the least significant byte.

R1<sub>c</sub>-R3<sub>c</sub> Row address for destination page C. R1<sub>c</sub> is the least significant byte.

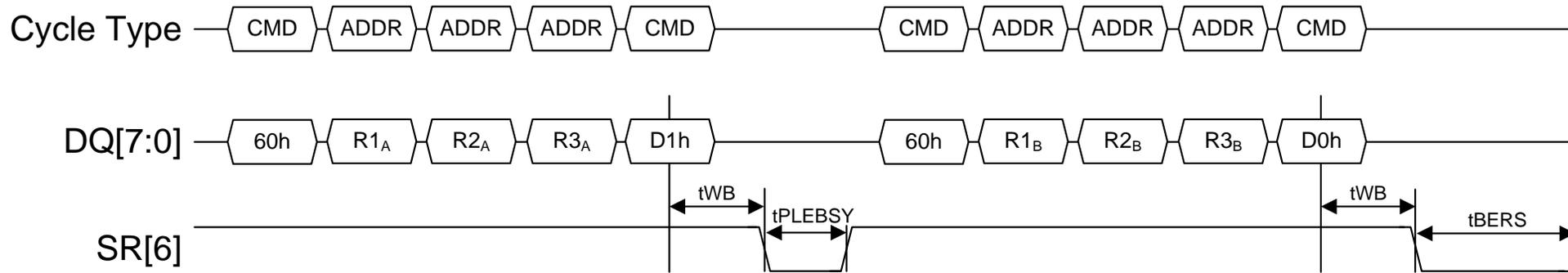
C1<sub>d</sub>-C2<sub>d</sub> Column address for destination page D. C1<sub>d</sub> is the least significant byte.

R1<sub>d</sub>-R3<sub>d</sub> Row address for destination page D. R1<sub>d</sub> 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. Multi-plane Block Erase

Figure 89 defines the behavior and timings for a multi-plane block erase operation. Only two operations are shown, however additional erase operations may be issued with a 60h/D1h sequence prior to the final 60h/D0h sequence depending on how many multi-plane operations the LUN supports.



**Figure 89 Multi-plane Block Erase timing**

R1<sub>A</sub>-R3<sub>A</sub> Row address for erase block A. R1<sub>A</sub> is the least significant byte.

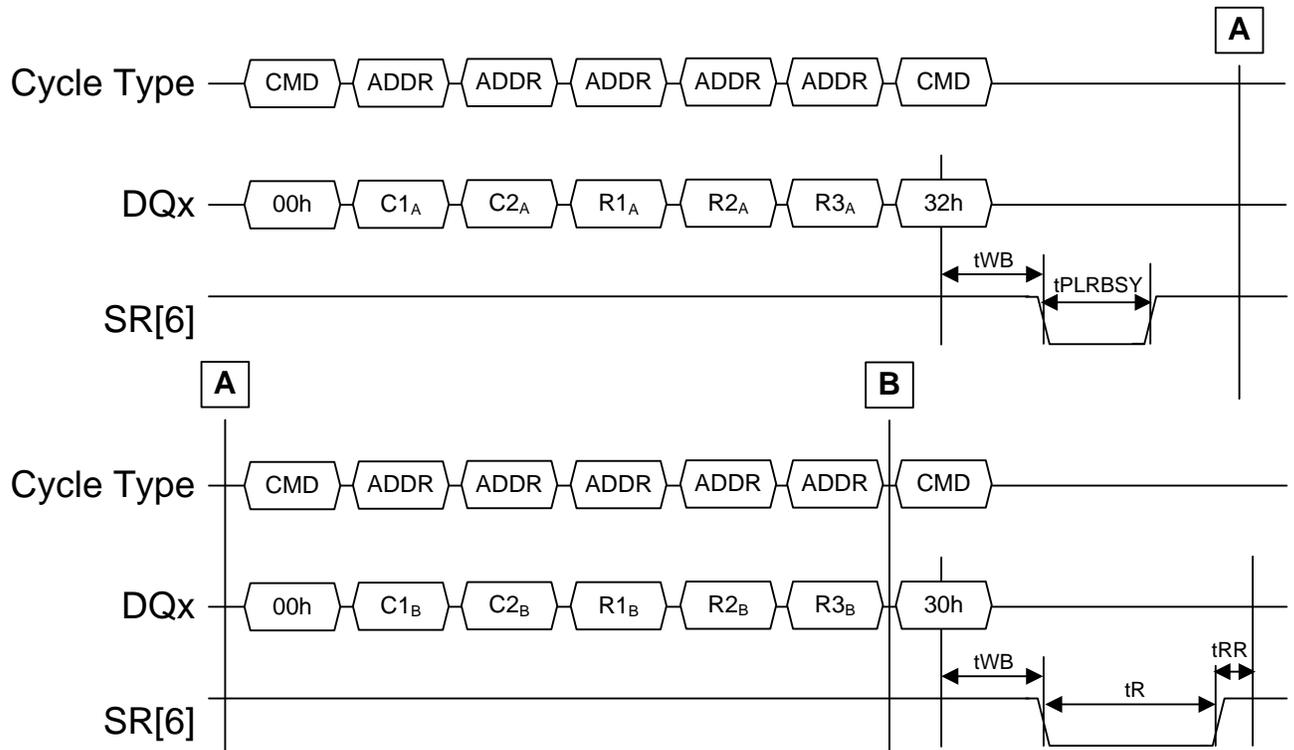
R1<sub>B</sub>-R3<sub>B</sub> Row address for erase block B. R1<sub>B</sub> is the least significant byte.

## 6.6. Multi-plane Read

The Read command reads a page of data identified by a row address for the LUN specified. The page of data is made available to be read from the page register starting at the column address specified. With a multi-plane operation, multiple reads can be issued back to back to the LUN, with a shorter busy time between issuance of the next read operation. Figure 90 defines the behavior and timings for issuing two multi-plane read commands. Figure 91 defines the behavior and timings for reading data after the multi-plane read commands are ready to return data.

Cache operations may be used when doing multi-plane read operations, as shown, if supported by the target as indicated in the parameter page. Refer to section 5.7.1.27.

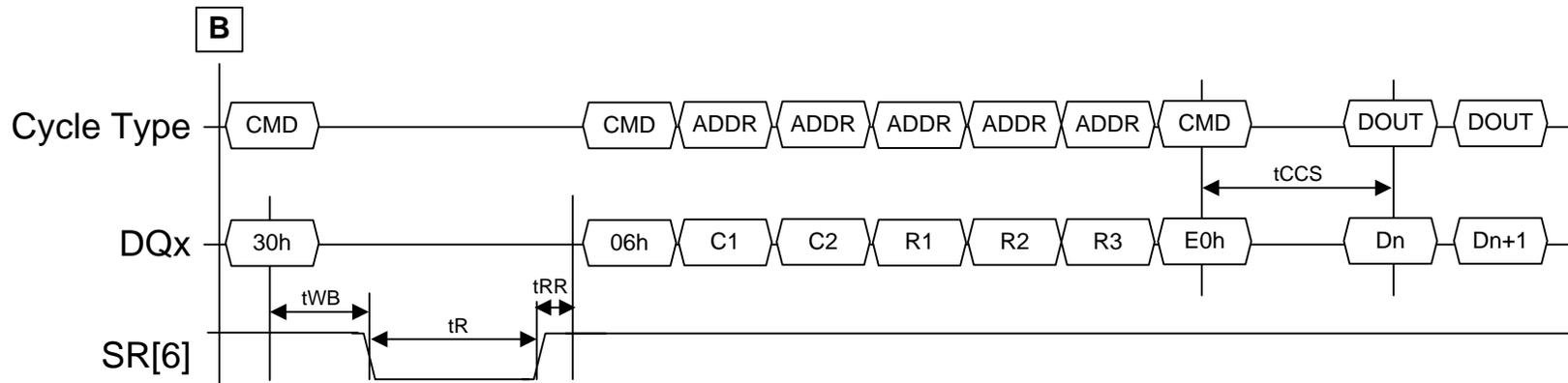
Change Read Column Enhanced shall be issued prior to reading data from a LUN. If data is read without issuing a Change Read Column Enhanced, the output received is undefined.



**Figure 90 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 91 Multi-plane Read data output timing, continued from command issue**

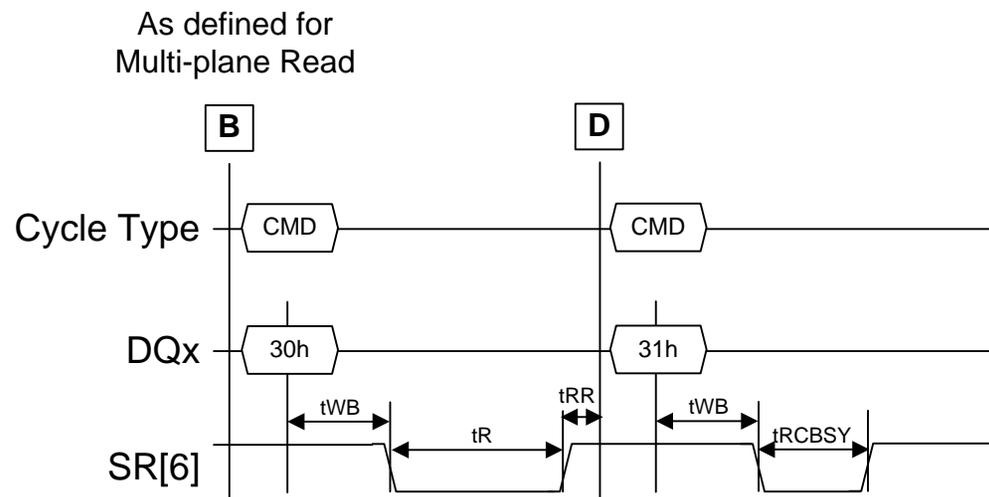
C1-C2 Column address to read from. C1 is the least significant byte.

R1-R3 Row address to read from (specifies LUN and plane address). R1 is the least significant byte.

Dn Data bytes read starting with addressed row and column.

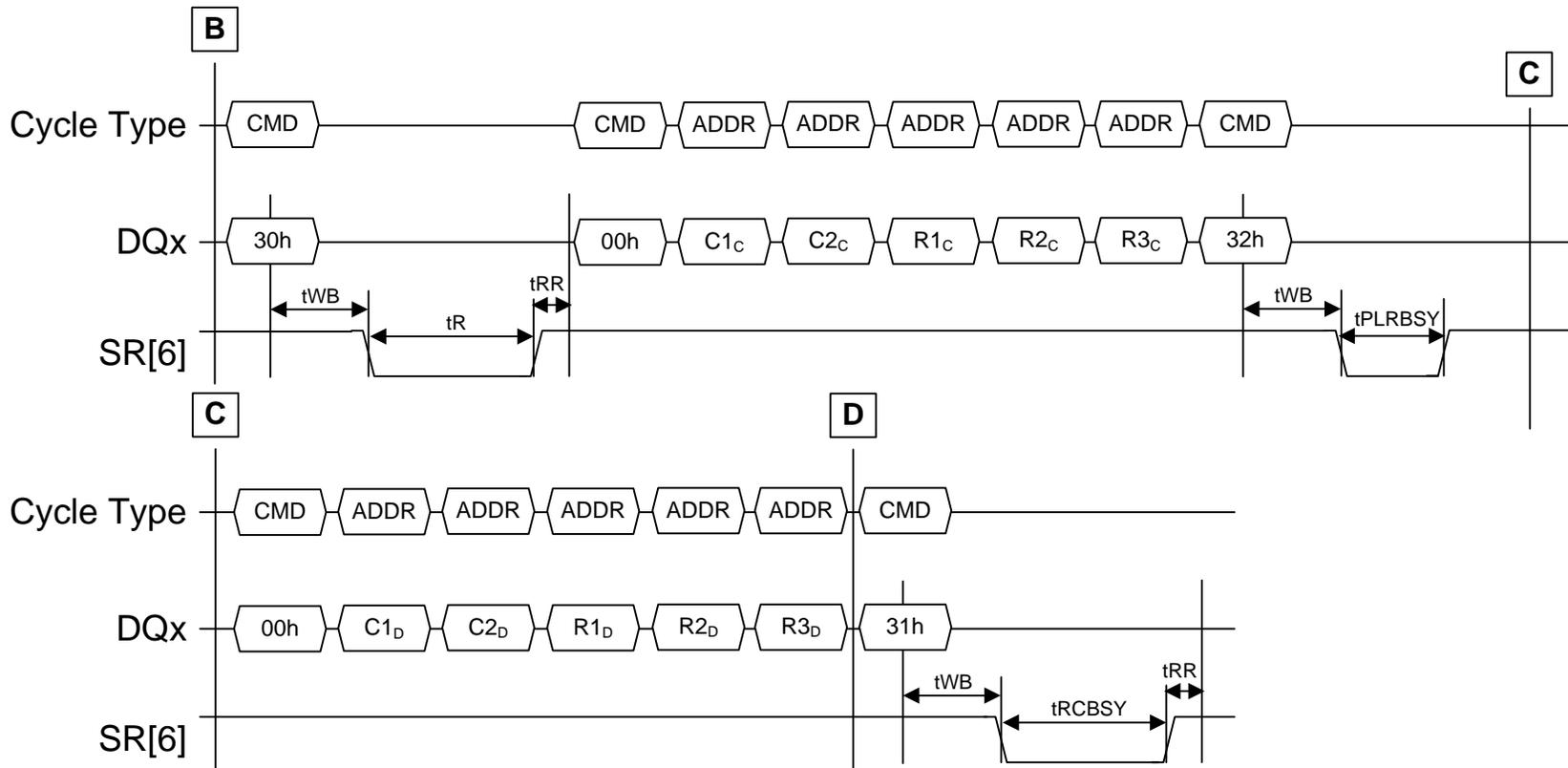
The row address provided shall specify a LUN and plane address that has valid read data.

For Multi-plane Read Cache Sequential operations, the initial Multi-plane Read command issue is followed by a Read Cache confirmation opcode 31h, as shown in Figure 92.



**Figure 92 Multi-plane Read Cache Sequential command issue timing**

For Multi-plane Read Cache Random operations, the initial multi-plane Read command issue is followed by another Read Multi-plane command sequence where the last confirmation opcode is 31h, as shown in Figure 93.



**Figure 93 Multi-plane Read Cache Random command issue timing**

- C1<sub>c</sub>-C2<sub>c</sub> Column address for page C. C1<sub>c</sub> is the least significant byte.
- R1<sub>c</sub>-R3<sub>c</sub> Row address for page C. R1<sub>c</sub> is the least significant byte.
- C1<sub>d</sub>-C2<sub>d</sub> Column address for page D. C1<sub>d</sub> is the least significant byte.
- R1<sub>d</sub>-R3<sub>d</sub> Row address for page D. R1<sub>d</sub> is the least significant byte.

The row addresses for page C and D shall differ in the plane address bits.

For Multi-plane Read Cache operations, two data output operations follow each Multi-plane Read Cache operation. The individual data output sequences are described in Figure 91. Prior to the last set (i.e. two) data output operations, a Read Cache End command (3Fh) should be issued by the host.

## 7. Behavioral Flows

### 7.1. Target behavioral flows

The Target state machine describes the allowed sequences when operating with the target. If none of the arcs are true, then the target remains in the current state.

#### 7.1.1. Variables

This section describes variables used within the Target state machine.

<b>tbStatusOut</b>	This variable is set to TRUE when a data read cycle should return the status value. The power-on value for this variable is FALSE.
<b>tbChgCol</b>	This variable is set to TRUE when changing the column using Change Read Column is allowed. The power-on value for this variable is FALSE.
<b>tbChgColEnh</b>	This variable is set to TRUE when changing the column using Change Read Column Enhanced is allowed. The power-on value for this variable is FALSE.
<b>tCopyback</b>	This variable is set to TRUE if the Target is issuing a copyback command. The power-on value for this variable is FALSE.
<b>tLunSelected</b>	This variable contains the LUN that is currently selected by the host. The power-on value for this variable is 0.
<b>tLastCmd</b>	This variable contains the first cycle of the last command (other than 70h/78h) received by the Target.
<b>tReturnState</b>	This variable contains the state to return to after status operations.
<b>tbStatus78hReq</b>	This variable is set to TRUE when the next status operation shall be a 78h command (and not a 70h command). The power-on value for this variable is FALSE.

#### 7.1.2. Idle states

T_PowerOn <sup>1</sup>	The target performs the following actions: <ol style="list-style-type: none"> <li>1. R/B_n is cleared to zero.</li> <li>2. Each LUN shall draw less than 10 mA of power per staggered power-up requirement.</li> </ol>	
<ol style="list-style-type: none"> <li>1. Target is ready to accept FFh (Reset) command<sup>2</sup></li> </ol>	→	<u>T_PowerOnReady</u>
NOTE: <ol style="list-style-type: none"> <li>1. This state is entered as a result of a power-on event when Vcc reaches Vcc_min.</li> <li>2. This arc shall be taken within 1 millisecond of Vcc reaching Vcc_min.</li> </ol>		
T_PowerOnReady	The target performs the following actions: <ol style="list-style-type: none"> <li>1. R/B_n is set to one.</li> <li>2. Each LUN shall draw less than 10mA of power per staggered power-up requirement.</li> </ol>	
<ol style="list-style-type: none"> <li>1. Command cycle FFh (Reset) received</li> </ol>	→	<u>T_RST_PowerOn</u>

T_Idle	tCopyback set to FALSE. tReturnState set to T_Idle.	
1. WP_n signal transitioned	→	<u>T_Idle_WP_Transition</u>
2. LUN indicates its SR[6] value transitioned	→	<u>T_Idle_RB_Transition</u>
3. Command cycle received	→	<u>T_Cmd_Decode</u>

T_Cmd_Decode <sup>1</sup>	Decode command received. tbStatusOut is set to FALSE. If R/B_n is set to one and command received is not 70h (Read Status), then tbStatus78hReq is set to FALSE.	
1. (Command 80h (Page Program) or command 60h (Block Erase) decoded) and WP_n is low	→	<u>T_Idle</u>
2. Command FFh (Reset) decoded	→	<u>T_RST_Execute</u>
3. Command FCh (Synchronous Reset) decoded	→	<u>T_RST_Execute_Sync</u>
4. Command FAh (Reset LUN) decoded	→	<u>T_RST_Execute_LUN</u>
5. Command 90h (Read ID) decoded	→	<u>T RID_Execute</u>
6. Command ECh (Read Parameter Page) decoded	→	<u>T_RPP_Execute</u>
7. Command EDh (Read Unique ID) decoded	→	<u>T RU_Execute</u>
8. Command 80h (Page Program) decoded and WP_n is high	→	<u>T_PP_Execute</u>
9. Command 60h (Block Erase) decoded and WP_n is high	→	<u>T_BE_Execute</u>
10. Command 00h (Read) decoded	→	<u>T_RD_Execute</u>
11. Command EFh (Set Features) decoded	→	<u>T_SF_Execute</u>
12. Command EEh (Get Features) decoded	→	<u>T_GF_Execute</u>
13. Command 70h (Read Status) decoded	→	<u>T_RS_Execute</u>
14. Command 78h (Read Status Enhanced) decoded	→	<u>T_RSE_Execute</u>
NOTE:		
1. The host shall ensure R/B_n is set to one before issuing Target level commands (Reset, Read ID, Read Parameter Page, Read Unique ID, Set Features, Get Features).		

T_Idle_WP_Transition	Indicate WP_n value to all LUN state machines.	
1. State entered from T_Idle_Rd	→	<u>T_Idle_Rd</u>
2. Else	→	<u>T_Idle</u>

T_Idle_RB_Transition	R/B_n is set to the AND of all LUN status register SR[6] values. <sup>1</sup>	
1. Unconditional	→	tReturnState
NOTE:		
1. R/B_n may transition to a new value prior to the Target re-entering an idle condition when LUN level commands are in the process of being issued.		

### 7.1.3. Idle Read states

T_Idle_Rd	Wait for read request (data or status) or other action. tReturnState set to T_Idle_Rd.	
1. WP_n signal transitioned	→	<u>T_Idle_WP_Transition</u>
2. LUN indicates its SR[6] value transitioned	→	<u>T_Idle_RB_Transition</u>
3. Read request received and tbStatusOut set to TRUE	→	<u>T_Idle_Rd_Status</u>
4. Read request received and (tLastCmd set to 90h or EEh)	→	<u>T_Idle_Rd_XferByte</u>
5. Read request received and (tLastCmd set to ECh or EDh)	→	<u>T_Idle_Rd_LunByte</u>
6. Read request received and tbStatus78hReq set to FALSE <sup>1</sup>	→	<u>T_Idle_Rd_LunData</u>
7. Command cycle 05h (Change Read Column) received and tbChgCol set to TRUE	→	<u>T_CR_Execute</u> <sup>2</sup>
8. Command cycle 06h (Change Read Column Enhanced) received and tbChgColEnh set to TRUE	→	<u>T_CRE_Execute</u> <sup>2</sup>
9. Command cycle of 31h received and tbStatus78hReq set to FALSE	→	<u>T_Idle_Rd_CacheCmd</u>
10. Command cycle of 3Fh received and tLastCmd set to 31h and tbStatus78hReq set to FALSE	→	<u>T_Idle_Rd_CacheCmd</u>
11. Command cycle received	→	<u>T_Cmd_Decode</u>
<p>NOTE:</p> <ol style="list-style-type: none"> <li>When tbStatus78hReq is set to TRUE, a Read Status Enhanced (78h) command followed by a 00h command shall be issued by the host prior to reading data from a particular LUN.</li> <li>If there are reads outstanding on other LUNs for this target, a Change Read Column (Enhanced) shall be issued before transferring data. Refer to section 3.1.3 that describes multiple LUN operation restrictions.</li> </ol>		

T_Idle_Rd_CacheCmd	Set tLastCmd to the command received. Pass command received to LUN tLunSelected	
1. Unconditional	→	<u>T_Idle_Rd</u>

T_Idle_Rd_XferByte	Return next byte of data.	
1. Unconditional	→	<u>T_Idle_Rd</u>

T_Idle_Rd_LunByte	Request byte of data from page register of LUN tLunSelected.	
1. Byte received from LUN tLunSelected	→	<u>T_Idle_Rd_XferHost</u>

T_Idle_Rd_LunData	Request byte (x8) or word (x16) of data from page register of LUN tLunSelected.	
1. Byte or word received from LUN tLunSelected	→	<u>T_Idle_Rd_XferHost</u>

T_Idle_Rd_XferHost	Transfer data byte or word received from LUN tLunSelected to host.
1. tReturnState set to T_RD_StatusOff and tCopyback set to TRUE	→ T_RD_Copyback
2. tReturnState set to T_RD_StatusOff	→ T_Idle_Rd
3. Else	→ tReturnState

T_Idle_Rd_Status	Request status from LUN tLunSelected.
1. Status from LUN tLunSelected received	→ T_Idle_Rd_StatusEnd

T_Idle_Rd_StatusEnd	Transfer status byte received from LUN tLunSelected to host.
1. Unconditional	→ tReturnState

T_CR_Execute	Wait for a column address cycle.
1. Column address cycle received	→ T_CR_Addr

T_CR_Addr	Store the column address cycle received.
1. More column address cycles required	→ T_CR_Execute
2. All column address cycles received	→ T_CR_WaitForCmd

T_CR_WaitForCmd	Wait for a command cycle.
1. Command cycle E0h received	→ T_CR_ReturnToData

T_CR_ReturnToData	Request that LUN tLunSelected select the column in the page register based on the column address received.
1. tReturnState set to T_PP_MplWait	→ T_PP_WaitForDataOut
2. tReturnState set to T_RD_Status_Off	→ T_Idle_Rd
3. Else	→ tReturnState

T_CRE_Execute	Wait for a column address cycle.
1. Column address cycle received	→ T_CRE_ColAddr

T_CRE_ColAddr	Store the column address cycle received.
1. More column address cycles required	→ T_CRE_Execute
2. All column address cycles received	→ T_CRE_RowAddrWait

T_CRE_RowAddrWait	Wait for a row address cycle.
1. Row address cycle received	→ T_CRE_RowAddr

T_CRE_RowAddr	Store the row address cycle received.
1. More row address cycles required	→ T_CRE_RowAddrWait
2. All row address cycles received	→ T_CRE_WaitForCmd

T_CRE_WaitForCmd	Wait for a command cycle.
1. Command cycle E0h received	→ T_CRE_ReturnToData

T_CRE_ReturnToData	<p>The target performs the following actions:</p> <ol style="list-style-type: none"> <li>1. Set tLunSelected to LUN selected by row address received.</li> <li>2. Request that LUN tLunSelected select the column in the page register based on the column address received.</li> <li>3. Indicate plane address received to tLunSelected for use in data output.</li> <li>4. Request all idle LUNs not selected turn off their output buffers.<sup>1</sup></li> </ol>
1. tReturnState set to T_PP_MplWait	→ T_PP_WaitForDataOut
2. tReturnState set to T_RD_Status_Off	→ T_Idle_Rd
3. Else	→ tReturnState
<p>NOTE:</p> <ol style="list-style-type: none"> <li>1. LUNs not selected only turn off their output buffers if they are in an idle condition (SR[6] is one) when Change Read Column Enhanced is received. If LUNs are active (SR[6] is zero) when Change Read Column Enhanced is issued, then the host shall issue a Read Status Enhanced (78h) command prior to subsequent data output to ensure all LUNs that are not selected turn off their output buffers.</li> </ol>	

#### 7.1.4. Reset command states

T_RST_PowerOn	<p>The target performs the following actions:</p> <ol style="list-style-type: none"> <li>1. tLastCmd set to FFh.</li> <li>2. tbStatusOut is set to FALSE.</li> <li>3. The target sends a Reset request to each LUN.</li> </ol>
1. Unconditional	→ T_RST_PowerOn_Exec

T_RST_PowerOn_Exec	<p>The target performs the following actions:</p> <ol style="list-style-type: none"> <li>1. Target level reset actions are performed.</li> <li>2. R/B_n is set to zero.</li> </ol>
1. Unconditional	→ T_RST_Perform

T_RST_Execute <sup>1</sup>	The target performs the following actions: <ol style="list-style-type: none"> <li>1. tLastCmd set to FFh.</li> <li>2. The target selects the asynchronous data interface.</li> <li>3. The target sends a Reset request to each LUN.</li> <li>4. Set tbChgCol to FALSE.</li> <li>5. Set tbChgColEnh to FALSE.</li> <li>6. Request all LUNs invalidate page register(s).</li> </ol>
1. Unconditional	→ T_RST_Perform
NOTE: 1. This state is entered as a result of receiving a Reset (FFh) command in any other state, except if this is the first Reset after power-on.	

T_RST_Execute_Sync <sup>1</sup>	The target performs the following actions: <ol style="list-style-type: none"> <li>1. tLastCmd set to FCh.</li> <li>2. tbStatusOut is set to FALSE.</li> <li>3. The target sends a Reset request to each LUN.</li> <li>4. Set tbChgCol to FALSE.</li> <li>5. Set tbChgColEnh to FALSE.</li> <li>6. Request all LUNs invalidate page register(s).</li> </ol>
1. Unconditional	→ T_RST_Perform
NOTE: 1. This state is entered as a result of receiving a Synchronous Reset (FCh) command in any other state.	

T_RST_Execute_LUN <sup>1</sup>	The target performs the following actions: <ol style="list-style-type: none"> <li>1. tLastCmd set to FAh.</li> <li>2. tbStatusOut is set to FALSE.</li> <li>3. Set tbChgCol to FALSE.</li> <li>4. Set tbChgColEnh to FALSE.</li> <li>5. Wait for an address cycle.</li> </ol>
1. Unconditional	→ T_RST_LUN_Addr

T_RST_LUN_AddrWait	Wait for an address cycle.
1. Address cycle received	→ T_RST_LUN_Addr

T_RST_LUN_Addr	Store the address cycle received.
1. More address cycles required	→ T_RST_LUN_AddrWait
2. All address cycles received	→ T_RST_LUN_Perform

T_RST_LUN_Perform	The target performs the following actions: 1. The target sends a Reset request to the addressed LUN. 2. R/B_n is cleared to zero. 3. Request the addressed LUN invalidate its page register.
1. Addressed LUN reset actions are complete and tbStatusOut is set to FALSE	→ T_Idle
2. Addressed LUN reset actions are complete and tbStatusOut is set to TRUE	→ T_Idle_Rd

T_RST_Perform	The target performs the following actions: 1. Target level reset actions are performed. 2. R/B_n is set to zero. 3. tReturnState set to T_RST_Perform.
1. Target and LUN reset actions are complete	→ T_RST_End
2. Command cycle 70h (Read Status) received	→ T_RS_Execute
3. Read request received and tbStatusOut is set to TRUE	→ T_Idle_Rd_Status

T_RST_End	The target performs the following actions: 1. R/B_n is set to one.
1. tbStatusOut is set to FALSE	→ T_Idle
2. tbStatusOut is set to TRUE	→ T_Idle_Rd

### 7.1.5. Read ID command states

T_RID_Execute	The target performs the following actions: 1. tLastCmd set to 90h. 2. Wait for an address cycle. 3. Set tbChgCol to FALSE. 4. Set tbChgColEnh to FALSE. 5. Request all LUNs invalidate page register(s).
1. Address cycle of 00h received	→ T_RID_Addr_00h
2. Address cycle of 20h received	→ T_RID_Addr_20h

T_RID_Addr_00h	Wait for the read request.		
1. Read byte request received		→	<u>T_RID_ManufacturerID</u>
2. Command cycle received		→	<u>T_Cmd_Decode</u>

T_RID_ManufacturerID	Return the JEDEC manufacturer ID.		
1. Read byte request received		→	<u>T_RID_DeviceID</u>
2. Command cycle received		→	<u>T_Cmd_Decode</u>

T_RID_DeviceID	Return the device ID. <sup>1</sup>		
1. Unconditional		→	<u>T_Idle_Rd</u>
NOTE:			
1. Reading bytes beyond the device ID returns vendor specific values.			

T_RID_Addr_20h	Wait for the read request.		
1. Read byte request received		→	<u>T_RID_Signature</u>
2. Command cycle received		→	<u>T_Cmd_Decode</u>

T_RID_Signature	Return next ONFI signature byte. <sup>1</sup>		
1. Last ONFI signature byte returned		→	<u>T_Idle_Rd</u>
2. Else		→	<u>T_RID_Addr_20h</u>
NOTE:			
1. Reading beyond the fourth byte returns indeterminate values.			

### 7.1.6. Read Parameter Page command states

T_RPP_Execute	The target performs the following actions:		
	<ol style="list-style-type: none"> <li>1. tLastCmd set to ECh.</li> <li>2. Set tbChgCol to TRUE.</li> <li>3. Set tbChgColEnh to FALSE.</li> <li>4. Wait for an address cycle.</li> <li>5. Request all LUNs invalidate page register(s).</li> <li>6. Target selects LUN to execute parameter page read, sets tLunSelected to the address of this LUN.</li> </ol>		
1. Address cycle of 00h received		→	<u>T_RPP_ReadParams</u>

T_RPP_ReadParams	The target performs the following actions: 1. Request LUN tLunSelected clear SR[6] to zero. 2. R/B_n is cleared to zero. 3. Request LUN tLunSelected make parameter page data available in page register. 4. tReturnState set to T_RPP_ReadParams_Cont.		
1. Read of page complete	→	<u>T_RPP_Complete</u>	
2. Command cycle 70h (Read Status) received	→	<u>T_RS_Execute</u>	
3. Read request received and tbStatusOut set to TRUE	→	<u>T_Idle_Rd_Status</u>	

T_RPP_ReadParams_Cont			
1. Read of page complete	→	<u>T_RPP_Complete</u>	
2. Command cycle 70h (Read Status) received	→	<u>T_RS_Execute</u>	
3. Read request received and tbStatusOut set to TRUE	→	<u>T_Idle_Rd_Status</u>	

T_RPP_Complete	Request LUN tLunSelected set SR[6] to one. R/B_n is set to one.		
1. Unconditional	→	<u>T_Idle_Rd</u>	

### 7.1.7. Read Unique ID command states

T_RU_Execute	The target performs the following actions: 1. tLastCmd set to EDh. 2. Set tbChgCol to TRUE. 3. Set tbChgColEnh to FALSE. 4. Request all LUNs invalidate page register(s). 5. Wait for an address cycle. 6. Target selects LUN to execute unique ID read, sets tLunSelected to the address of this LUN.		
1. Address cycle of 00h received	→	<u>T_RU_ReadUid</u>	

T_RU_ReadUid	The target performs the following actions: 1. Request LUN tLunSelected clear SR[6] to zero. 2. R/B_n is cleared to zero. 3. Request LUN tLunSelected make Unique ID data available in page register. 4. tReturnState set to T_RU_ReadUid.		
1. LUN tLunSelected indicates data available in page register	→	<u>T_RU_Complete</u>	
2. Command cycle 70h (Read Status) received	→	<u>T_RS_Execute</u>	
3. Read request received and tbStatusOut set to TRUE	→	<u>T_Idle_Rd_Status</u>	

T_RU_Complete	Request LUN tLunSelected set SR[6] to one. R/B_n is set to one.		
1. Unconditional	→	<u>T_Idle_Rd</u>	

### 7.1.8. Page Program and Page Cache Program command states

T_PP_Execute	The target performs the following actions: <ol style="list-style-type: none"> <li>1. tLastCmd set to 80h.</li> <li>2. If R/B_n is cleared to zero, then tbStatus78hReq is set to TRUE.</li> <li>3. If the program page register clear enhancement is not supported or disabled, request all LUNs clear their page register(s).<sup>1</sup></li> </ol>
1. Unconditional	→ T_PP_AddrWait
NOTE: <ol style="list-style-type: none"> <li>1. Idle LUNs may choose to not clear their page register if the Program is not addressed to that LUN.</li> </ol>	
T_PP_Copyback	If R/B_n is cleared to zero, then tbStatus78hReq is set to TRUE.
1. Unconditional	→ T_PP_AddrWait
T_PP_AddrWait	Wait for an address cycle.
1. Address cycle received	→ T_PP_Addr
T_PP_Addr	Store the address cycle received.
1. More address cycles required	→ T_PP_AddrWait
2. All address cycles received	→ T_PP_LUN_Execute

T_PP_LUN_Execute	The target performs the following actions: <ol style="list-style-type: none"> <li>1. tLunSelected is set to the LUN indicated by the row address received.</li> <li>2. If the program page register clear enhancement is enabled, request LUN tLunSelected clear the page register for the plane address specified.</li> <li>3. Target issues the Program with associated address to the LUN tLunSelected.</li> </ol>
1. Unconditional	→ T_PP_LUN_DataWait

T_PP_LUN_DataWait	Wait for data byte/word or command cycle to be received from the host.
1. Data byte/word received from the host	→ T_PP_LUN_DataPass
2. Command cycle of 15h received and tCopyback set to FALSE	→ T_PP_Cmd_Pass
3. Command cycle of 10h or 11h received	→ T_PP_Cmd_Pass
4. Command cycle of 85h received	→ T_PP_ColChg

T_PP_LUN_DataPass	Pass data byte/word received from host to LUN tLunSelected
1. Unconditional	→ T_PP_LUN_DataWait

T_PP_Cmd_Pass	Pass command received to LUN tLunSelected
1. Command passed was 11h	→ T_PP_MplWait
2. Command passed was 10h or 15h	→ T_Idle

T_PP_MplWait	Wait for next Program to be issued. tReturnState set to T_PP_MplWait.	
1. Command cycle of 85h received <sup>1</sup>	→	<u>T_PP_AddrWait</u>
2. Command cycle of 80h received <sup>2</sup> and tCopyback set to FALSE	→	<u>T_PP_AddrWait</u>
3. Command cycle of 05h received	→	<u>T_CR_Execute</u>
4. Command cycle of 06h received	→	<u>T_CRE_Execute</u>
5. Command cycle of 70h received	→	<u>T_RS_Execute</u>
6. Command cycle of 78h received	→	<u>T_RSE_Execute</u>
7. Read request received and tbStatusOut set to TRUE	→	<u>T_Idle_Rd_Status</u>
<p>NOTE:</p> <p>1. If the 85h is part of a Copyback, Change Row Address, or Small Data Move operation, then the LUN address and plane address shall be the same as the preceding Program operation. If the 85h is part of a Small Data Move operation, then the page address shall also be the same as the preceding Program operation.</p> <p>2. Address cycles for the Program operation being issued shall have the same LUN address and page address as the preceding Program operation. The plane address shall be different than the one issued in the preceding Program operation.</p>		

T_PP_ColChg	Wait for column address cycle.	
1. Address cycle received	→	<u>T_PP_ColChg_Addr</u>

T_PP_ColChg_Addr	Store the address cycle received.	
1. More column address cycles required	→	<u>T_PP_ColChg</u>
2. All address cycles received	→	<u>T_PP_ColChg_LUN</u>

T_PP_ColChg_LUN	Request that LUN tLunSelected change column address to column address received.	
1. Unconditional	→	<u>T_PP_ColChg_Wait</u>

T_PP_ColChg_Wait	Wait for an address cycle, data byte/word, or command cycle to be received from the host	
1. Address cycle received	→	<u>T_PP_RowChg_Addr</u>
2. Data byte/word received from the host	→	<u>T_PP_LUN_DataPass</u>
3. Command cycle of 15h received and tCopyback set to FALSE	→	<u>T_PP_Cmd_Pass</u>
4. Command cycle of 10h or 11h received	→	<u>T_PP_Cmd_Pass</u>
5. Command cycle of 85h received	→	<u>T_PP_ColChg</u>

T_PP_RowChg	Wait for row address cycle.	
1. Address cycle received	→	<u>T_PP_RowChg_Addr</u>

T_PP_RowChg_Addr	Store the address cycle received.		
1. More row address cycles required		→	<u>T_PP_RowChg</u>
2. All address cycles received		→	<u>T_PP_RowChg_LUN</u>

T_PP_RowChg_LUN	Request that LUN tLunSelected change row address to row address received. <sup>1</sup>		
1. Unconditional		→	<u>T_PP_LUN_DataWait</u>
NOTE: 1. The LUN address and plane address shall be the same as previously specified for the Program operation executing.			

T_PP_WaitForDataOut	Wait for read request (data or status) or other action. tReturnState set to T_PP_WaitForDataOut.		
1. Read request received and tbStatusOut set to TRUE		→	<u>T_Idle_Rd_Status</u>
2. Read request received and tbStatus78hReq set to FALSE <sup>1</sup>		→	<u>T_Idle_Rd_LunData</u>
3. Command cycle of 70h received		→	<u>T_RS_Execute</u>
4. Command cycle of 78h received		→	<u>T_RSE_Execute</u>
5. Command cycle of 00h received		→	<u>T_RD_Execute</u>
6. Command cycle received		→	<u>T_PP_MpIWait</u>
NOTE: 1. When tbStatus78hReq is set to TRUE, a Read Status Enhanced (78h) command followed by a 00h command shall be issued by the host prior to reading data from a particular LUN.			

### 7.1.9. Block Erase command states

T_BE_Execute	The target performs the following actions: 1. tLastCmd set to 60h. 2. If R/B_n is cleared to zero, then tbStatus78hReq is set to TRUE. 3. Wait for a row address cycle.		
1. Address cycle received		→	<u>T_BE_Addr</u>

T_BE_Addr	Store the row address cycle received.		
1. More address cycles required		→	<u>T_BE_Execute</u>
2. All address cycles received		→	<u>T_BE_LUN_Execute</u>

T_BE_LUN_Execute	tLunSelected is set to the LUN indicated by the row address received. Target issues the Erase with associated row address to the LUN tLunSelected.		
1. Unconditional		→	<u>T_BE_LUN_Confirm</u>

T_BE_LUN_Confirm	Wait for D0h or D1h command cycle.
1. Command cycle of D0h or D1h received	→ T_BE_Cmd_Pass

T_BE_Cmd_Pass	Pass command received to LUN tLunSelected
1. Command passed was D1h	→ T_BE_MplWait
2. Command passed was D0h	→ T_Idle

T_BE_MplWait	Wait for next Erase to be issued. tReturnState set to T_BE_MplWait.
1. Command cycle of 60h received	→ T_BE_Execute
2. Command cycle of 70h received	→ T_RS_Execute
3. Command cycle of 78h received	→ T_RSE_Execute
4. Read request received and tbStatusOut set to TRUE	→ T_Idle_Rd_Status

### 7.1.10. Read command states

T_RD_Execute		
1. tbStatusOut set to TRUE	→	<u>T_RD_StatusOff</u>
2. Else	→	<u>T_RD_AddrWait</u>

T_RD_StatusOff	tbStatusOut set to FALSE. tReturnState set to T_RD_StatusOff.	
1. Address cycle received	→	<u>T_RD_Addr</u>
2. Read request received and tLastCmd set to 80h	→	<u>T_PP_WaitForDataOut</u>
3. Read request received and tLastCmd set to EEh	→	<u>T_Idle_Rd_XferHost</u>
4. Read request received	→	<u>T_Idle_Rd_LunData</u>
5. Command cycle of 05h received	→	<u>T_CR_Execute</u>
6. Command cycle of 06h received	→	<u>T_CRE_Execute</u>

T_RD_AddrWait	tLastCmd set to 00h. Set tbChgCol to TRUE. Set tbChgColEnh to TRUE. If R/B_n is cleared to zero, then tbStatus78hReq is set to TRUE. Wait for an address cycle.	
1. Address cycle received	→	<u>T_RD_Addr</u>

T_RD_Addr	Store the address cycle received.	
3. More address cycles required	→	<u>T_RD_AddrWait</u>
4. All address cycles received	→	<u>T_RD_LUN_Execute</u>

T_RD_LUN_Execute	The target performs the following actions: <ul style="list-style-type: none"> <li>1. tLunSelected is set to the LUN indicated by the row address received.</li> <li>2. Issues the Read Page with address to LUN tLunSelected.</li> <li>3. Requests all idle LUNs not selected to turn off their output buffers.<sup>1</sup></li> </ul>	
1. Unconditional	→	<u>T_RD_LUN_Confirm</u>
NOTE:		
1. LUNs not selected will only turn off their output buffers if they are in an Idle state. If other LUNs are active, the host shall issue a Read Status Enhanced (78h) command to ensure all LUNs that are not selected turn off their output buffers prior to issuing the Read (00h) command.		

T_RD_LUN_Confirm	Wait for 30h, 31h, 32h, or 35h to be received.	
1. Command cycle of 30h, 31h, 32h, or 35h received	→	<u>T_RD_Cmd_Pass</u>

T_RD_Cmd_Pass	Pass command received to LUN tLunSelected	
1. Command passed was 35h	→	<u>T_RD_Copyback</u>
2. Command passed was 30h, 31h, or 32h	→	<u>T_Idle_Rd</u>

T_RD_Copyback	tCopyback set to TRUE. tReturnState set to T_RD_Copyback.	
1. Command cycle of 00h received	→	<u>T_RD_Execute</u>
2. Command cycle of 05h received	→	<u>T_CR_Execute</u>
3. Command cycle of 06h received	→	<u>T_CRE_Execute</u>
4. Command cycle of 85h received	→	<u>T_PP_Copyback</u>
5. Command cycle of 70h received	→	<u>T_RS_Execute</u>
6. Command cycle of 78h received	→	<u>T_RSE_Execute</u>
7. LUN indicates its SR[6] value transitions	→	<u>T_Idle_RB_Transition</u>
8. Read request received and tbStatusOut set to TRUE	→	<u>T_Idle_Rd_Status</u>
9. Read request received	→	<u>T_Idle_Rd_LunData</u>

### 7.1.11. Set Features command states

T_SF_Execute	The target performs the following actions: 1. tLastCmd set to EFh. 2. Request all LUNs invalidate page register(s). 3. Wait for an address cycle.
1. Address cycle received	→ T_SF_Addr

T_SF_Addr	Store the feature address received.
1. Unconditional	→ T_SF_WaitForParams

T_SF_WaitForParams	Wait for data byte to be received.
1. Data byte written to target	→ T_SF_StoreParam

T_SF_StoreParam	Store parameter received.
1. More parameters required	→ T_SF_WaitForParams
2. All parameters received	→ T_SF_Complete

T_SF_Complete	The target performs the following actions: 1. Request LUN tLunSelected clear SR[6] to zero. 2. R/B_n is cleared to zero. 3. Finish Set Features command. 4. tReturnState set to T_SF_Complete.
1. Set Features command complete	→ T_SF_UpdateStatus
2. Command cycle 70h (Read Status) received	→ T_RS_Execute
3. Read request received and tbStatusOut set to TRUE	→ T_Idle_Rd_Status

T_SF_UpdateStatus	The target performs the following actions: 1. Request LUN tLunSelected set SR[6] to one. 2. R/B_n is set to one.
1. tbStatusOut is set to FALSE	→ T_Idle
2. tbStatusOut is set to TRUE	→ T_Idle_Rd

### 7.1.12. Get Features command states

T_GF_Execute	The target performs the following actions: 1. tLastCmd set to EEh. 2. Request all LUNs invalidate page register(s). 3. Set tbChgCol to FALSE. 4. Set tbChgColEnh to FALSE. 5. Wait for an address cycle.
1. Address cycle received	→ T_GF_Addr

T_GF_Addr	Store the feature address received.
1. Unconditional	→ T_GF_RetrieveParams

T_GF_RetrieveParams	The target performs the following actions: 1. Request LUN tLunSelected clear SR[6] to zero. 2. R/B_n is cleared to zero. 3. Retrieve parameters. 4. tReturnState set to T_GF_RetrieveParams.
1. Parameters are ready to be transferred to the host	→ T_GF_Ready
2. Command cycle 70h (Read Status) received	→ T_RS_Execute
3. Read request received and tbStatusOut set to TRUE	→ T_Idle_Rd_Status

T_GF_Ready	Request LUN tLunSelected set SR[6] to one. R/B_n is set to one.
1. Unconditional	→ T_Idle_Rd

### 7.1.13. Read Status command states

T_RS_Execute	
1. tbStatus78hReq is set to FALSE <sup>1</sup>	→ T_RS_Perform
NOTE: 1. When tbStatus78hReq is set to TRUE, issuing a Read Status (70h) command is illegal.	

T_RS_Perform	The target performs the following actions: 1. tbStatusOut is set to TRUE. 2. Indicate 70h command received to LUN tLunSelected.
1. tReturnState set to T_Idle	→ T_Idle_Rd
2. Else	→ tReturnState

### 7.1.14. Read Status Enhanced command states

T_RSE_Execute <sup>1</sup>	tbStatus78hReq is set to FALSE. tbStatusOut is set to TRUE. Wait for a row address cycle.	
1. Row address cycle received	→	<u>T_RSE_Addr</u>
<p>NOTE:</p> <p>1. The host should not issue Read Status Enhanced following a Target level command (Reset, Read ID, Read Parameter Page, Read Unique ID, Set Features, Get Features). The status value read from the LUN selected with Read Status Enhanced may not correspond with the LUN selected during the Target level command.</p>		
T_RSE_Addr	Store the row address cycle received.	
1. More row address cycles required	→	<u>T_RSE_Execute</u>
2. All row address cycles received	→	<u>T_RSE_Select</u>
T_RSE_Select	The target performs the following actions:	
	<ol style="list-style-type: none"> <li>1. Set tLunSelected to LUN selected by row address received.</li> <li>2. Indicate 78h command and row address received to all LUNs.</li> </ol>	
1. tReturnState set to T_Idle	→	<u>T_Idle_Rd</u>
2. Else	→	tReturnState

## 7.2. LUN behavioral flows

The LUN state machine describes the allowed sequences when operating with the LUN. If none of the arcs are true, then the LUN remains in the current state.

### 7.2.1. Variables

This section describes variables used within the LUN state machine.

<b>lunStatus</b>	This variable contains the current LUN status register value contents. The power on value for this variable is 00h.
<b>lunFail[]</b>	This array contains the FAIL and FAILC bits for each interleave address. For example, lunFail[3][1] contains the FAILC bit for plane address 3. The power on value for each variable in this array is 00b.
<b>lunLastConfirm</b>	This variable contains the last confirm command cycle (30h, 31h, 32h, 35h, 10h, 15h, 11h, D0h, D1h). The power on value for this variable is FFh.
<b>lunOutputMpl</b>	This variable contains the plane address requested for data output. The power on value for this variable is 0h.
<b>lunReturnState</b>	This variable contains the state to return to after status operations. The power on value for this variable is L_Idle.
<b>lunStatusCmd</b>	This variable contains the last status command received. The power on value for this variable is 70h.
<b>lunStatusMpl</b>	This variable contains the plane address indicated in a previous 78h command. The power on value for this variable is 0h.
<b>lunbInterleave</b>	This variable is set to one when the LUN is performing a multi-plane operation. The power on value for this variable is FALSE.
<b>lunbMplNextCmd</b>	This variable is set to TRUE when the LUN is ready to receive the next multi-plane command.
<b>lunEraseAddr[]</b>	This variable contains the block addresses of erases that have been suspended.

### 7.2.2. Idle command states

L_Idle <sup>1</sup>	lunReturnState is set to L_Idle.
1. Target request received	→ L_Idle TargetRequest
NOTE: 1. This state is entered as a result of a power-on event when Vcc reaches Vcc_min.	

L_Idle_TargetRequest	If Target indicates an address, the address is stored by the LUN.	
1. Target requests LUN perform a Reset	→	<u>L_RST_Execute</u>
2. Target indicates WP_n value	→	<u>L_WP_Update</u>
3. Target requests SR register update	→	<u>L_SR_Update</u>
4. Target requests status or status command received	→	<u>L_Status_Execute</u>
5. Target indicates plane address for use in data output	→	<u>L_Idle_Mpl_DataOutAddr</u>
6. Target indicates output buffer should be turned off	→	<u>L_Idle</u>
7. Target requests page register clear	→	<u>L_Idle_ClearPageReg</u>
8. Target requests page register invalidate	→	<u>L_Idle_InvalidPageReg</u>
9. Target indicates Program request for this LUN	→	<u>L_PP_Execute</u>
10. Target indicates Erase request for this LUN	→	<u>L_BE_Execute</u>
11. Target indicates Erase Resume request for this LUN	→	<u>L_ER_Execute</u>
12. Target indicates Read Page request for this LUN	→	<u>L_RD_Addr</u>
13. Target indicates Read Parameter Page request	→	<u>L_Idle_RdPp</u>
14. Target indicates Read Unique ID request	→	<u>L_Idle_RdUid</u>

L_WP_Update	Set lunStatus[7] to the WP_n value indicated by the target.	
1. Unconditional	→	lunReturnState

L_SR_Update	Update lunStatus as indicated by the target.	
1. Unconditional	→	lunReturnState

L_Idle_Mpl_DataOutAddr	Set lunOutputMpl to plane address indicated by the target.	
1. Unconditional	→	lunReturnState

L_Idle_ClearPageReg	Set page register to all ones value.	
1. Unconditional	→	lunReturnState

L_Idle_InvalidPageReg	Invalidate page register.	
1. Unconditional	→	lunReturnState

L_Idle_RdPp	The LUN performs the following actions: 1. LUN reads parameter page data into the page register. 2. lunReturnState set to L_Idle_RdPp_Cont.	
1. Parameter page data transferred to page register	→	<u>L_Idle_RdPp_End</u>
2. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_Idle_RdPp_Cont			
	1. Parameter page data transferred to page register	→	<u>L_Idle_RdPp_End</u>
	2. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_Idle_RdPp_End	LUN indicates to Target that parameter page data is in page register.		
	1. Unconditional	→	<u>L_Idle_Rd</u>

L_Idle_RdUid	The LUN performs the following actions: 1. LUN reads Unique ID data into the page register. 2. lunReturnState set to L_Idle_RdUid.		
	1. Unique ID data transferred to page register	→	<u>L_Idle_RdUid_End</u>
	2. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_Idle_RdUid_End	LUN indicates to Target that Unique ID data is in page register.		
	1. Unconditional	→	<u>L_Idle_Rd</u>

### 7.2.3. Idle Read states

L_Idle_Rd	lunReturnState is set to L_Idle_Rd.		
	1. Background read operation complete	→	<u>L_Idle_Rd_Finish</u>
	2. Target requests column address be selected	→	<u>L_Idle_Rd_ColSelect</u>
	3. Read request received from Target	→	<u>L_Idle_Rd_Xfer</u>
	4. Command cycle 31h (Read Cache Sequential) received	→	<u>L_RD_Cache_Next</u>
	5. Command cycle 3Fh (Read Cache End) received and lunLastConfirm is 31h	→	<u>L_RD_Cache_Xfer_End</u>
	6. Target request received	→	<u>L_Idle_TargetRequest</u>

L_Idle_Rd_Finish	Set lunStatus[5] to one.		
	1. Unconditional	→	<u>L_Idle_Rd</u>

L_Idle_Rd_Xfer	Return to the target the next byte (x8) or word (x16) of data from page register based on Target requested. Increments column address.		
	1. lunReturnState set to L_PP_Mpl_Wait	→	<u>L_PP_Mpl_Wait</u>
	2. Unconditional	→	<u>L_Idle_Rd</u>

L_Idle_Rd_ColSelect	Select the column in the page register based on the column address received from the target.		
	1. lunReturnState set to L_PP_Mpl_Wait	→	<u>L_PP_Mpl_Wait</u>
	2. Unconditional	→	<u>L_Idle_Rd</u>

## 7.2.4. Status states

L_Status_Execute			
1. Target requests status value	→		<u>L_Status_Value</u>
2. Target indicates 78h was received	→		<u>L_Status_Enhanced</u>
3. Target indicates 70h was received	→		<u>L_Status_Legacy</u>

L_Status_Value			
1. lunbInterleave set to TRUE and lunStatusCmd set to 70h	→		<u>L_Status_Mpl_Comp</u>
2. lunbInterleave set to TRUE and lunStatusCmd set to 78h	→		<u>L_Status_Mpl_Addr</u>
3. lunbInterleave set to FALSE	→		<u>L_Status_Lun</u>

L_Status_Enhanced			
1. LUN in row address indicated matches this LUN	→		<u>L_Status_Record_78h</u>
2. Else	→		<u>L_Status_Output_Off</u>

L_Status_Record_78h	lunStatusCmd is set to 78h and lunStatusMpl is set to plane address indicated by Target. The LUN turns on its output buffer.		
1. lunReturnState set to L_Idle and (lunLastConfirm set to 30h, 31h, 32h, or 35h)	→		<u>L_Idle_Rd</u>
2. Else	→		lunReturnState

L_Status_Output_Off	LUN turns off its output buffer.		
1. lunReturnState set to L_Idle_Rd	→		<u>L_Idle</u>
2. Else	→		lunReturnState

L_Status_Legacy	lunStatusCmd is set to 70h.		
1. Unconditional	→		lunReturnState

L_Status_Mpl_Comp	<p>The LUN composes the status value to return as shown:</p> <ul style="list-style-type: none"> <li>• status[7:2] = lunStatus[7:2]</li> <li>• status[1] = for all x, OR of lunFail[x][1]</li> <li>• status[0] = for all x, OR of lunFail[x][0]</li> </ul> <p>Return status to the Target.</p>		
1. Unconditional	→		lunReturnState

L_Status_Mpl_Addr	<p>The LUN composes the status value to return as shown:</p> <ul style="list-style-type: none"> <li>• status[7:2] = lunStatus[7:2]</li> <li>• status[1:0] = lunFail[lunStatusMpl][1:0]</li> </ul>		
-------------------	---	--	--

	Return status to the Target.		
1. Unconditional		→	lunReturnState

L_Status_Lun	Return lunStatus to the Target.		
1. Unconditional		→	lunReturnState

### 7.2.5. Reset states

L_RST_Execute <sup>1</sup>	The LUN performs the following actions: 1. lunStatus[6] is cleared to zero. 2. lunStatus[6] value is indicated to the Target. 3. Perform reset of the LUN. 4. lunbInterleave is set to FALSE. 5. lunReturnState is set to L_RST_Execute.		
1. Reset of the LUN is complete		→	<u>L_RST_Complete</u>
2. Target requests status or status command received		→	<u>L_Status_Execute</u>
NOTE: 1. This state is entered as a result of receiving an indication from the Target state machine to perform a Reset in any other state.			

L_RST_Complete	The LUN performs the following actions: 1. lunStatus[1:0] are cleared to 00b. 2. For all plane addresses x, clear lunFail[x][1:0] to 00b. 3. lunStatus[6] is set to one. 4. lunStatus[6] value is indicated to the Target. 5. Indicate to the Target state machine that Reset for this LUN is complete.		
1. Unconditional		→	<u>L_Idle</u>

### 7.2.6. Block Erase command states

L_BE_Execute	lunbInterleave set to FALSE.		
1. Unconditional		→	<u>L_BE_WaitForCmd</u>

L_BE_WaitForCmd	Wait for a command cycle.		
1. Command cycle D0h received		→	<u>L_BE_Erase</u>
2. Command cycle D1h received		→	<u>L_BE_Mpl</u>

L_BE_Erase	The LUN performs the following actions: 1. lunStatus[6] is cleared to zero. 2. If lunbInterleave is TRUE, lunStatus[5] is cleared to zero. 3. lunStatus[6] value is indicated to the Target. 4. lunLastConfirm set to D0h. 5. Erase the requested block and any previously requested blocks if lunbInterleave is set to TRUE and concurrent		
------------	--	--	--

	interleaving is supported.	
1. Unconditional	→	<u>L_BE_Erase_Wait</u>

<u>L_BE_Erase_Wait</u>	lunReturnState set to <u>L_BE_Erase_Wait</u> .
1. Erase of requested block(s) complete and lunbInterleave set to TRUE	→ <u>L_BE_Mpl_Sts</u>
2. Erase of requested block complete	→ <u>L_BE_Sts</u>
3. Target requests page register clear	→ <u>L_Idle_ClearPageReg</u>
4. Target requests status or status command received	→ <u>L_Status_Execute</u>

<u>L_BE_Mpl</u>	The LUN performs the following actions in the order specified: 1. lunbInterleave set to TRUE. 2. lunLastConfirm set to D1h. 3. lunStatus[6:5] is cleared to 00b. lunStatus[6] value is indicated to the Target. 4. LUN begins erasing block specified if overlapped is supported. 5. lunbMplNextCmd is set to FALSE. 6. LUN prepares to receive the next block to erase.
1. Unconditional	→ <u>L_BE_Mpl_Wait</u>

<u>L_BE_Mpl_Wait</u>	lunReturnState set to <u>L_BE_Mpl_Wait</u> .
1. An overlapped multi-plane Erase completed	→ <u>L_BE_Mpl_Overlap</u>
2. Ready to receive the next Erase command and lunbMplNextCmd is set to FALSE	→ <u>L_BE_Mpl_NextCmd</u>
3. Target indicates Erase request for this LUN and lunbMplNextCmd is set to TRUE	→ <u>L_BE_WaitForCmd</u>
4. Target requests status or status command received	→ <u>L_Status_Execute</u>

<u>L_BE_Mpl_NextCmd</u>	The LUN performs the following actions in the order specified: 1. lunbMplNextCmd is set to TRUE. 2. If no array operations are in progress, lunStatus[5] is set to one. 3. lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.
1. Unconditional	→ <u>L_BE_Mpl_Wait</u>

<u>L_BE_Mpl_Overlap</u>	The LUN performs the following actions in the order specified for the overlapped multi-plane operation that completed: 1. mplComplete set to plane address of completed operation 2. lunFail[mplComplete][0] is set to program status of operation. If all array operations are complete, lunStatus[5] is set to one.
1. Unconditional	→ lunReturnState

L_BE_Sts	The LUN performs the following actions in the order specified: <ol style="list-style-type: none"> <li>1. lunStatus[0] is set to erase status.</li> <li>2. lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.</li> </ol>
1. Unconditional	→ <u>L_Idle</u>

L_BE_Mpl_Sts	The LUN performs the following actions in the order specified for each multi-plane operation that completed: <ol style="list-style-type: none"> <li>1. mplComplete set to interleave address of completed operation.</li> <li>2. lunFail[mplComplete][0] is set to erase status value. lunStatus[6:5] is set to 11b and lunStatus[6] value is indicated to the Target.</li> </ol>
1. Unconditional	→ <u>L_Idle</u>

### 7.2.7. Read command states

If caching is not supported, then all actions for status bit 5 are ignored.

L_RD_Addr	The LUN performs the following actions in the order specified: <ol style="list-style-type: none"> <li>1. Records address received from the target.</li> <li>2. If multi-plane addressing is supported, selects the correct page register based on the plane address.</li> <li>3. Selects the column in the page register based on the column address received.</li> </ol>
1. Unconditional	→ <u>L_RD_WaitForCmd</u>

L_RD_WaitForCmd	lunInterleave set to FALSE. Wait for a command cycle.
1. Command cycle 30h or 35h received	→ <u>L_RD_ArrayRead</u>
2. Command cycle 31h received and lunLastConfirm equal to 30h or 31h	→ <u>L_RD_Cache_Xfer</u>
3. Command cycle 32h received	→ <u>L_RD_Mpl_Xfer</u>

L_RD_ArrayRead	The LUN performs the following actions: <ol style="list-style-type: none"> <li>1. lunStatus[6:5] is cleared to 00b.</li> <li>2. lunStatus[6] value is indicated to the target.</li> <li>3. lunLastConfirm set to last command cycle (30h or 35h).</li> <li>4. Read the requested page from the array. If concurrent multi-plane operation, read all pages requested from the array.</li> <li>5. lunReturnState set to L_RD_ArrayRead_Cont.</li> </ol>
1. Read of requested page(s) complete	→ <u>L_RD_Complete</u>
2. Target requests status or status command received	→ <u>L_Status_Execute</u>

L_RD_ArrayRead_Cont			
	1. Read of requested page(s) complete	→	<u>L_RD_Complete</u>
	2. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_RD_Complete	lunStatus[6:5] is set to 11b. lunStatus[6] value is indicated to the target.		
	1. Unconditional	→	<u>L_Idle_Rd</u>

L_RD_Cache_Next	Select the next row address as the sequential increasing row address to the last page read.		
	1. Unconditional	→	<u>L_RD_Cache_Xfer</u>

L_RD_Cache_Xfer	The LUN performs the following actions: 1. lunStatus[6:5] is cleared to 00b. lunStatus[6] value is indicated to the Target. 2. lunLastConfirm set to 31h. 3. Begin background read operation for selected address. 4. lunReturnState set to L_RD_Cache_Xfer.		
	1. Data available in page register for previous read operation	→	<u>L_RD_Cache_Sts</u>
	2. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_RD_Cache_Xfer_End	The LUN performs the following actions: 1. lunStatus[6] is cleared to zero. 2. lunStatus[6] value is indicated to the target. 3. lunLastConfirm set to 3Fh. 4. lunReturnState set to L_RD_Cache_Xfer_End.		
	1. Data available in page register for previous read operation	→	<u>L_RD_Cache_Sts_End</u>
	2. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_RD_Cache_Sts	lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.		
	1. Unconditional	→	<u>L_Idle_Rd</u>

L_RD_Cache_Sts_End	lunStatus[6:5] is set to 11b. lunStatus[6] value is indicated to the Target.		
	1. Unconditional	→	<u>L_Idle_Rd</u>

L_RD_Mpl_Xfer	The LUN performs the following actions: 1. lunStatus[6:5] is cleared to 00b. 2. lunStatus[6] value is indicated to the target. 3. lunLastConfirm set to 32h. 4. lunbMplNextCmd is set to FALSE. 5. LUN begins reading page specified if overlapped interleaving is supported. 6. Prepare to receive the next page to read. 7. lunReturnState set to L_RD_Mpl_Xfer.
1. Target ready to receive next page to read	→ L_RD_Mpl_Wait
2. Target requests status or status command received	→ L_Status_Execute

L_RD_Mpl_Wait	lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target. lunReturnState set to L_RD_Mpl_Wait.
1. An overlapped multi-plane Read completed	→ L_RD_Mpl_Overlap
2. Target indicates Read Page request for this LUN	→ L_RD_Addr
3. Target requests status or status command received	→ L_Status_Execute

L_RD_Mpl_Overlap	The LUN performs the following actions in the order specified for the overlapped multi-plane operation that completed: 1. mplComplete set to plane address of completed operation. If all array operations are complete, lunStatus[5] is set to one.
1. Unconditional	→ lunReturnState

### 7.2.8. Page Program and Page Cache Program command states

If caching or overlapped interleaving is not supported, then all actions for status bit 5 are ignored.  
If caching is not supported, then all actions for status bit 1 are ignored.

L_PP_Execute	lunbInterleave set to FALSE.
1. Unconditional	→ L_PP_Addr

L_PP_Addr	The LUN performs the following actions in the order specified: 1. Records address received from the Target. 2. If multi-plane addressing is supported, selects the correct page register based on the plane address. 3. Selects the column in the page register based on the column address received.
1. Unconditional	→ L_PP_WaitForData

L_PP_WaitForData	Wait for data to be received. lunReturnState is set to L_PP_WaitForData.
1. Target passes data byte or word to LUN	→ L_PP_AcceptData
2. Command cycle 10h (program execute) received	→ L_PP_Prog

3. Command cycle 15h (cache program) received	→	<u>L_PP_Cache</u>
4. Command cycle 11h (interleave) received	→	<u>L_PP_Mpl</u>
5. Target requests column address be selected	→	<u>L_PP_ColSelect</u>
6. Target requests row address be selected	→	<u>L_PP_RowSelect</u>

L_PP_AcceptData	Write the byte (x8) or word (x16) of data into the selected column address in the page register. Increments column address.	
1. Unconditional	→	<u>L_PP_WaitForData</u>

L_PP_Prog	The LUN performs the following actions in the order specified: <ol style="list-style-type: none"> <li>1. lunStatus[6:5] is cleared to 00h. lunStatus[6] value is indicated to the Target.</li> <li>2. lunLastConfirm set to 10h.</li> <li>3. If only one page is specified to be programmed, clear lunbInterleave to FALSE.</li> <li>4. LUN begins programming page specified and any previous pages specified if lunbInterleave is TRUE and concurrent interleaving is supported.</li> </ol>	
1. Unconditional	→	<u>L_PP_Prog_Wait</u>

L_PP_Prog_Wait	lunReturnState set to L_PP_Prog_Wait.	
1. Write of all requested pages are complete and lunbInterleave is set to TRUE	→	<u>L_PP_Mpl_Sts</u>
2. Write of requested page is complete and lunbInterleave is cleared to FALSE	→	<u>L_PP_Sts</u>
3. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_PP_Cache	The LUN performs the following actions in the order specified: <ol style="list-style-type: none"> <li>1. lunStatus[6:5] is cleared to 00b. lunStatus[6] value is indicated to the Target.</li> <li>2. lunLastConfirm set to 15h.</li> <li>3. Wait for the page register to become available for data input.</li> <li>4. Start background program operation.</li> </ol>	
1. Unconditional	→	<u>L_PP_Cache_Wait</u>

L_PP_Cache_Wait	lunReturnState is set to L_PP_Cache_Wait.	
1. Page register available for data input	→	<u>L_PP_CacheRdy</u>
2. Target requests status or status command received	→	<u>L_Status_Execute</u>

L_PP_CacheRdy	The LUN performs the following actions: <ol style="list-style-type: none"> <li>1. If lunbInterleave is set to FALSE, then lunStatus[1] is set to the value of lunStatus[0].</li> <li>2. If lunbInterleave is set to TRUE, then for all multi-plane addresses, x, lunFail[x][1] is set to the value of lunFail[x][0].</li> <li>3. lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.</li> </ol>	
---------------	--	--

1. Unconditional	→	<u>L_PP_CacheRdy_Wait</u>
------------------	---	---------------------------

<u>L_PP_CacheRdy_Wait</u>	lunReturnState set to <u>L_PP_CacheRdy_Wait</u> .	
1. Previous cache operation complete and lunbInterleave set to TRUE	→	<u>L_PP_Mpl_Cache_Sts</u>
2. Previous cache operation complete	→	<u>L_PP_Cache_Sts</u>
3. Target indicates Program request for this LUN	→	<u>L_PP_Addr</u>
4. Target requests page register clear	→	<u>L_Idle_ClearPageReg</u>
5. Target requests status or status command received	→	<u>L_Status_Execute</u>

<u>L_PP_Mpl</u>	The LUN performs the following actions in the order specified: <ul style="list-style-type: none"> <li>1. lunbInterleave set to TRUE.</li> <li>2. lunStatus[6:5] is cleared to 00b. lunStatus[6] value is indicated to the Target.</li> <li>3. lunLastConfirm set to 11h.</li> <li>4. lunbMplNextCmd is set to FALSE.</li> <li>5. LUN begins programming page specified if overlapped interleaving is supported.</li> </ul>	
1. Unconditional	→	<u>L_PP_Mpl_Wait</u>

<u>L_PP_Mpl_Wait</u>	lunReturnState set to <u>L_PP_Mpl_Wait</u> .	
1. An overlapped multi-plane Program completed	→	<u>L_PP_Mpl_Overlap</u>
2. A previous cache Program completed	→	<u>L_PP_Mpl_Cache_Sts</u>
3. LUN is ready to receive the next Program command and lunbMplNextCmd is set to FALSE	→	<u>L_PP_Mpl_NextCmd</u>
4. Target indicates Program request for this LUN and lunbMplNextCmd is set to TRUE	→	<u>L_PP_Addr</u>
5. Target requests column address be selected	→	<u>L_Idle_Rd_ColSelect</u>
6. Target indicates plane address for use in data output	→	<u>L_Idle_Mpl_DataOutAddr</u>
7. Target requests status or status command received	→	<u>L_Status_Execute</u>
8. Read request received from Target	→	<u>L_Idle_Rd_Xfer</u>

<u>L_PP_Mpl_NextCmd</u>	The LUN performs the following actions in the order specified: <ul style="list-style-type: none"> <li>1. lunbMplNextCmd is set to TRUE.</li> <li>2. If no array operations are in progress, lunStatus[5] is set to one.</li> <li>3. lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.</li> </ul>	
1. Unconditional	→	<u>L_PP_Mpl_Wait</u>

<u>L_PP_Sts</u>	The LUN performs the following actions in the order specified: <ul style="list-style-type: none"> <li>1. lunStatus[1] is set to program status of previous operation</li> <li>2. lunStatus[0] is set to program status of final operation</li> <li>3. lunStatus[6:5] is set to 11b.</li> <li>4. lunStatus[6] value is indicated to the Target.</li> </ul>	
-----------------	---	--

1. Unconditional	→	<u>L_Idle</u>
------------------	---	---------------

L_PP_Cache_Sts	The LUN performs the following actions in the order specified: 1. lunStatus[0] is set to program status. 2. lunStatus[5] is set to one.
1. Unconditional	→ lunReturnState

L_PP_Mpl_Cache_Sts	The LUN performs the following actions in the order specified for all completed cache operations: 1. mplAddr set to interleave address of cache operation. 2. lunFail[mplAddr][0] is set to program status. If all array operations are complete, lunStatus[5] is set to one.
1. Unconditional	→ lunReturnState

L_PP_Mpl_Overlap	The LUN performs the following actions in the order specified for the overlapped multi-plane operation that completed: 1. mplComplete set to interleave address of completed operation 2. lunFail[mplComplete][0] is set to program status of operation. If all array operations are complete, lunStatus[5] is set to one.
1. Unconditional	→ lunReturnState

L_PP_Mpl_Sts	The LUN performs the following actions in the order specified for each multi-plane operation that completed: 1. mplComplete set to plane address of completed operation 2. lunFail[mplComplete][1] is set to program status of previous operation. 3. lunFail[mplComplete][0] is set to program status of final operation. lunStatus[6:5] is set to 11b and lunStatus[6] value is indicated to the Target.
1. Unconditional	→ <u>L_Idle</u>

L_PP_ColSelect	Select the column in the page register based on the column address received that the target requested.
1. Unconditional	→ <u>L_PP_WaitForData</u>

L_PP_RowSelect	Select the block and page to program based on the row address received from the target.
1. Unconditional	→ <u>L_PP_WaitForData</u>

## A. SAMPLE CODE FOR CRC-16 (INFORMATIVE)

This section provides an informative implementation of the CRC-16 polynomial. The example is intended as an aid in verifying an implementation of the algorithm.

```
int main(int argc, char* argv[])
{
    // Bit by bit algorithm without augmented zero bytes
    const unsigned long crcinit = 0x4F4E; // Initial CRC value in the shift register
    const int order = 16; // Order of the CRC-16
    const unsigned long polynom = 0x8005; // Polynomial
    unsigned long i, j, c, bit;
    unsigned long crc = crcinit; // Initialize the shift register with 0x4F4E
    unsigned long data_in;
    int dataByteCount = 0;
    unsigned long crcmask, crchighbit;
    crcmask = (((unsigned long)1<<(order-1))-1)<<1|1;
    crchighbit = (unsigned long)1<<(order-1);

    // Input byte stream, one byte at a time, bits processed from MSB to LSB
    printf("Input byte value in hex(eg. 0x30):");
    printf("\n");
```

```
while(scanf("%x", &data_in) == 1)
{
    c = (unsigned long)data_in;
    dataByteCount++;
    for (j=0x80; j>=>1) {
        bit = crc & crchighbit;
        crc<<= 1;
        if (c & j) bit^= crchighbit;
        if (bit) crc^= polynom;
    }
    crc&= crcmask;
    printf("CRC-16 value: 0x%x\n", crc);
}
printf("Final CRC-16 value: 0x%x, total data bytes: %d\n", crc, dataByteCount);

return 0;
}
```

## B. SPARE SIZE RECOMMENDATIONS (INFORMATIVE)

This appendix describes recommendations for the spare bytes per page based on the ECC requirements reported in the parameter page. These recommendations are for raw NAND implementations and do not apply to devices that support EZ NAND. Table 51 lists recommendations for 2KB, 4KB, and 8KB page size devices.

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 <sup>2</sup>	> 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 51 Spare Area Size Recommendations for raw NAND**

The host transfers bytes from the page register in discrete units that include data, metadata, and the ECC check bytes. This discrete unit is recommended to be an even number of bytes for devices that support the source synchronous data interface.

As an example, assume the page size is 8192 bytes and the ECC codeword size used is 1KB. Then 1024 bytes of data will be transferred in each discrete unit, resulting in eight discrete units of data being transferred for this page. The spare bytes for this page should be allocated to allow enough storage for the metadata and check bytes, and should also be an even number when divided by eight (i.e. the number of discrete units contained in that page).

## C. DEVICE SELF-INITIALIZATION WITH PSL (INFORMATIVE)

Some devices store configuration information for the Flash array within the Flash array itself. The device loads this information either at power-on or during the first Reset after power-on.

Vendors may choose to support PSL as one of the vendor specific pins. If PSL is supported, then it shall have the following behavior:

- PSL = 0 V: Configuration information is loaded at power-on. The IST current may be up to 15 mA and the time for R/B\_n to become one is up to 5 ms.
- PSL = Vcc or not connected: Configuration information if supported is loaded during the first Reset after power-on. There is no change to the IST current requirement. This corresponds to the normally expected ONFI device operation.

If PSL is not supported by the device, then the IST requirement shall be met.

Refer to the device vendor's datasheet to determine if self-initialization at power-on is supported.

## D. ICC MEASUREMENT METHODOLOGY

This section defines the technique used to measure the ICC parameters defined in section 2.10.

The common testing conditions that shall be used to measure the DC and Operating Conditions are defined in Table 52. The testing conditions that shall be used to measure the DC and Operating Conditions that are data interface specific are defined in Table 53.

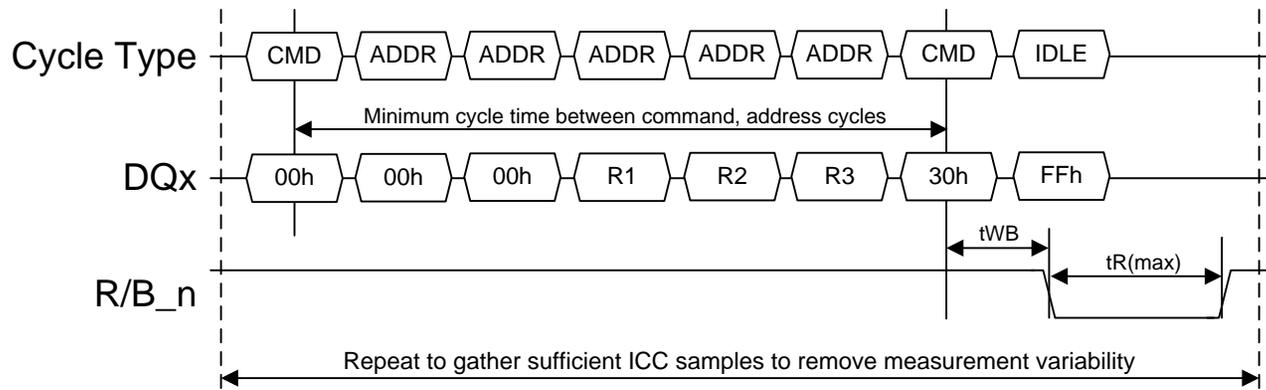
Parameter	Testing Condition
General conditions	<ol style="list-style-type: none"> <li>1. <math>V_{cc} = V_{cc}(\min)</math> to <math>V_{cc}(\max)</math></li> <li>2. <math>V_{ccQ} = V_{ccQ}(\min)</math> to <math>V_{ccQ}(\max)</math></li> <li>3. <math>CE_n = 0\text{ V}</math></li> <li>4. <math>WP_n = V_{ccQ}</math></li> <li>5. <math>IO_{UT} = 0\text{ mA}</math></li> <li>6. Measured across operating temperature range</li> <li>7. N data input or data output cycles, where N is the number of bytes or words in the page</li> <li>8. No multi-plane operations</li> <li>9. Sample sufficient number of times to remove measurement variability</li> <li>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).</li> <li>11. Choose the first good even/odd block pair beginning at blocks 2-3</li> </ol>
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: $t_R = t_R(\max)$ ICC2: $t_{PROG} = t_{PROG}(\max)$ ICC3: $t_{BERS} = t_{BERS}(\max)$

**Table 52 Common Testing Conditions for ICC**

Parameter	Asynchronous	Source Synchronous
AC Timing Parameters	$t_{WC} = t_{WC}(\min)$ $t_{RC} = t_{RC}(\min)$ $t_{ADL} = \sim t_{ADL}(\min)$ $t_{CCS} = \sim t_{CCS}(\min)$ $t_{RHW} = \sim t_{RHW}(\min)$	$t_{CK} = t_{CK}(\text{avg})$ $t_{ADL} = \sim t_{ADL}(\min)$ $t_{CCS} = \sim t_{CCS}(\min)$ $t_{RHW} = \sim t_{RHW}(\min)$
Bus idle data pattern	$IO[7:0] = FFh$ $IO[15:0] = FFFFh$	$DQ[7:0] = FFh$
Repeated data pattern (Used for ICC2 and ICC4 <sub>W</sub> )	$IO[7:0] = A5h, AAh, 5Ah, 55h$ $IO[15:0] = A5A5h, AAAAh, 5A5Ah, 5555h$	$DQ[7:0] = A5h, AAh, 5Ah, 55h$
Array preconditioning for ICC4 <sub>R</sub>	The array is preconditioned to match the following repeating data pattern: $IO[7:0] = A5h$ $IO[15:8] = A5A5h$	The array is preconditioned to match the following repeating data pattern: $DQ[7:0] = A5h$
NOTE: 1. The value of $t_{CK}(\text{avg})$ used should be the minimum $t_{CK}(\text{avg})$ of the timing modes supported for the device. The source synchronous timing modes supported by the device are indicated in the parameter page.		

**Table 53 Data Interface Specific Testing Conditions for ICC**

The following figures detail the testing procedure for ICC1, ICC2, ICC3, ICC4R, ICC4W, and ICC5.

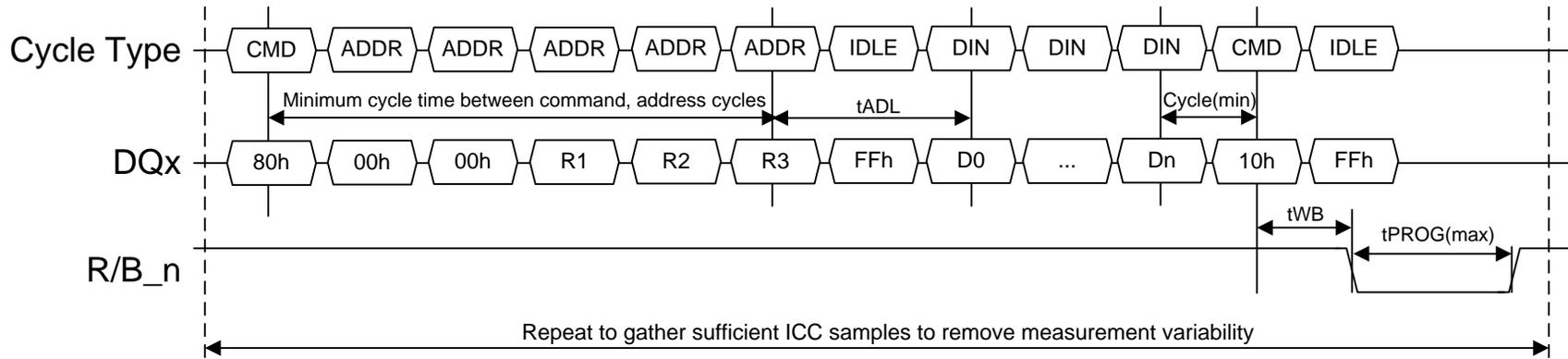


**Figure 94 ICC1 measurement procedure**

To calculate the active current for ICC1, the following equations may be used.

$$I_{cc1}(measured) = \frac{tR(typ)}{tR(max)} I_{cc1}(active) + \frac{tR(max) - tR(typ)}{tR(max)} I_{cc5}$$

$$I_{cc1}(active) = \frac{I_{cc1}(measured) \times tR(max)}{tR(typ)} - \frac{I_{cc5} \times tR(max)}{tR(typ)} + I_{cc5}$$



**Figure 95 ICC2 measurement procedure**

To calculate the active current for ICC2, the following equations may be used.

$$I_{cc2}(measured) = \frac{t_{IO}}{t_{IO} + t_{PROG}(max)} I_{cc4w} + \frac{t_{PROG}(typ)}{t_{IO} + t_{PROG}(max)} I_{cc2}(active) + \frac{t_{PROG}(max) - t_{PROG}(typ)}{t_{IO} + t_{PROG}(max)} I_{cc5}$$

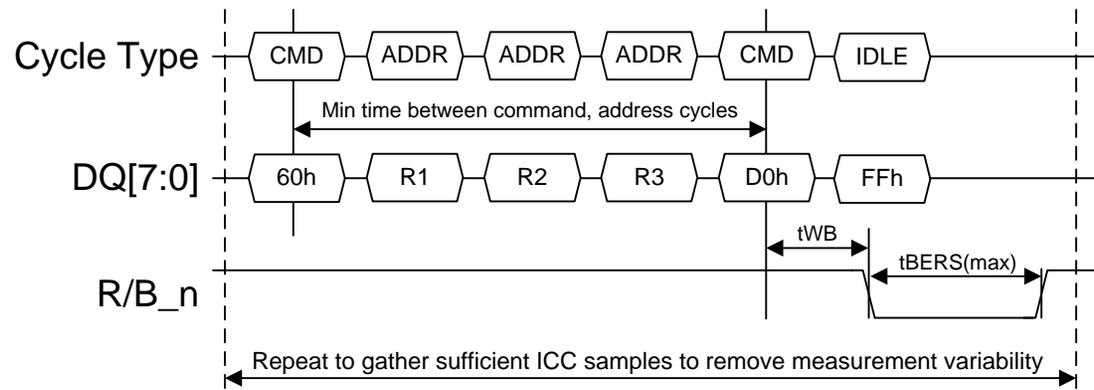
$$I_{cc2}(active) = \frac{I_{cc2}(measured) \times (t_{IO} + t_{PROG}(max))}{t_{PROG}(typ)} - \frac{t_{IO} \times I_{cc4w}}{t_{PROG}(typ)} - \frac{I_{cc5} \times t_{PROG}(max)}{t_{PROG}(typ)} + I_{cc5}$$

For the asynchronous interface, the  $t_{IO}$  value is calculated as:

$$t_{IO} = NAND\ Page\ Size(bytes\ (x8)\ or\ words\ (x16)) \times t_{WC}(min)$$

For the source synchronous data interface, the  $t_{IO}$  value is calculated as:

$$t_{IO} = NAND\ Page\ Size(bytes) \times \frac{1}{2} t_{CK}(avg)$$

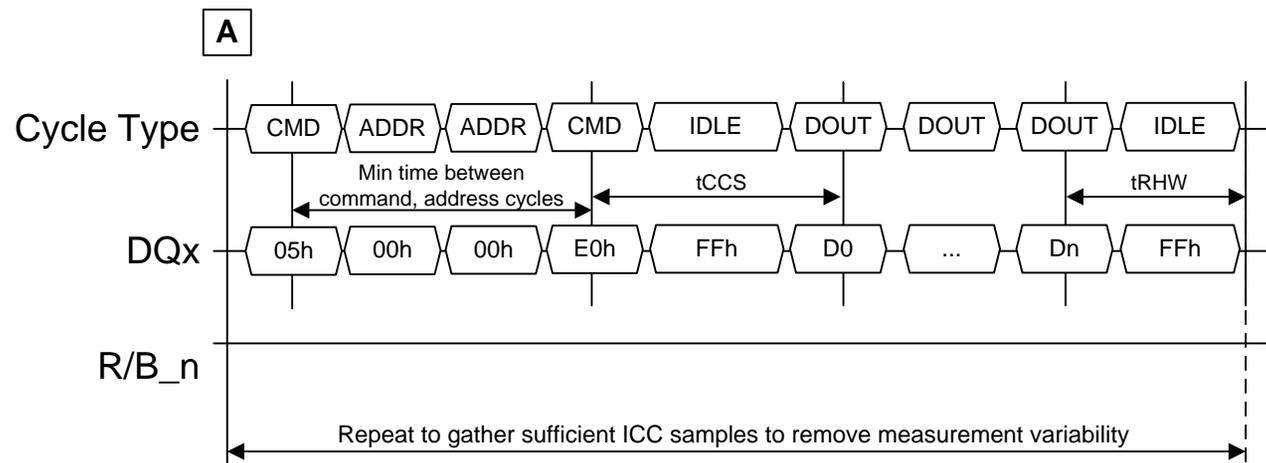
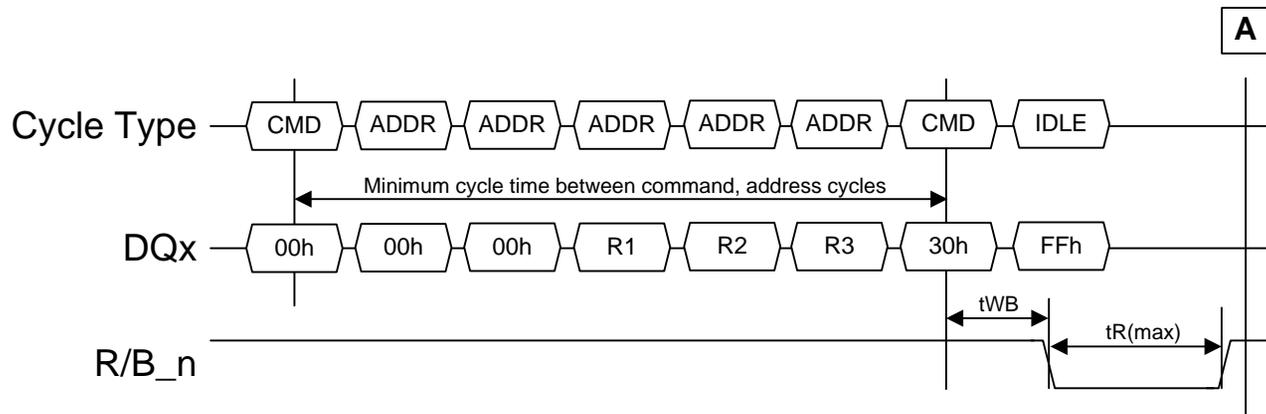


**Figure 96 ICC3 measurement procedure**

To calculate the active current for ICC3, the following equations may be used.

$$I_{cc3(measured)} = \frac{tBERS(typ)}{tBERS(max)} I_{cc3(active)} + \frac{tBERS(max) - tBERS(typ)}{tBERS(max)} I_{cc5}$$

$$I_{cc3(active)} = \frac{I_{cc3(measured)} \times tBERS(max)}{tBERS(typ)} - \frac{I_{cc5} \times tBERS(max)}{tBERS(typ)} + I_{cc5}$$



**Figure 97 ICC4R measurement procedure**

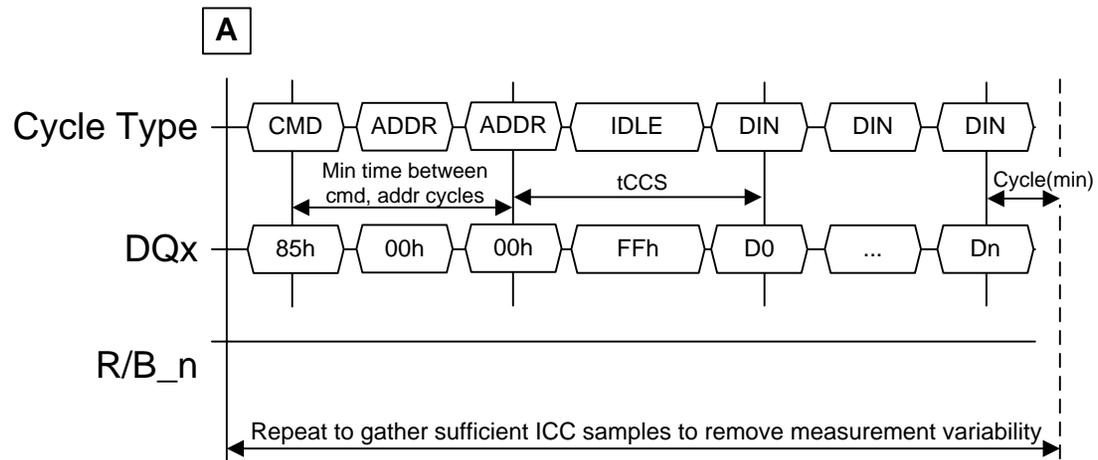
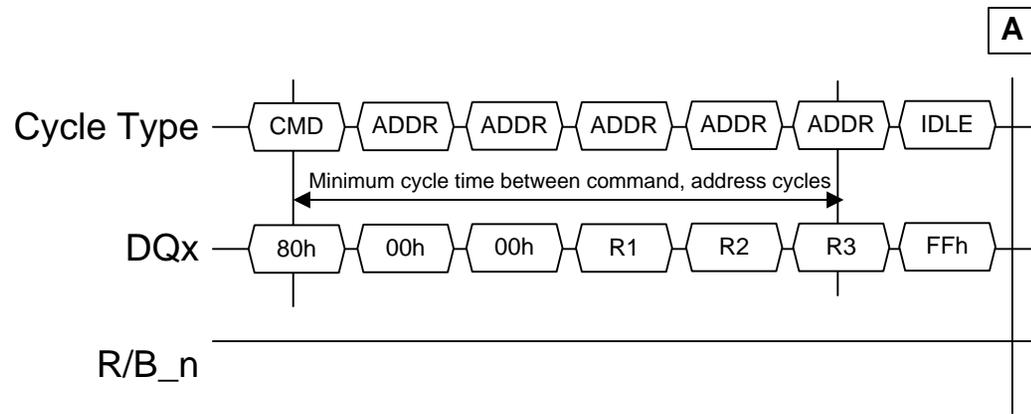
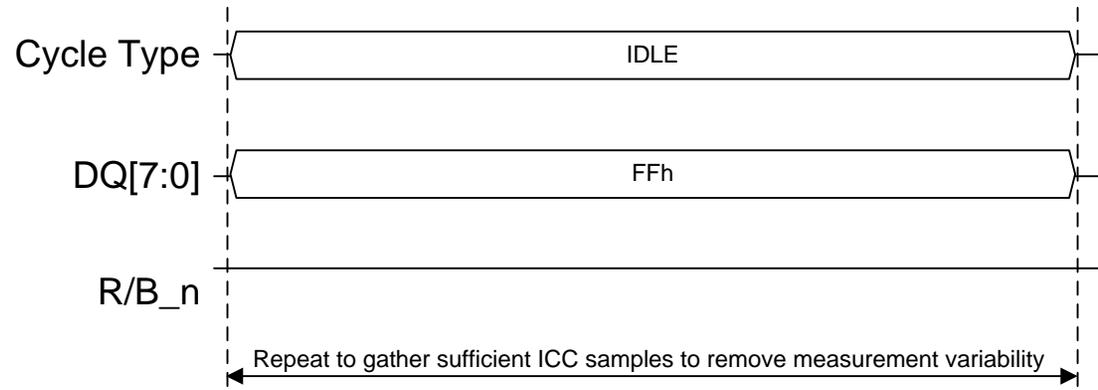


Figure 98 ICC4W measurement procedure



**Figure 99** ICC5 measurement procedure

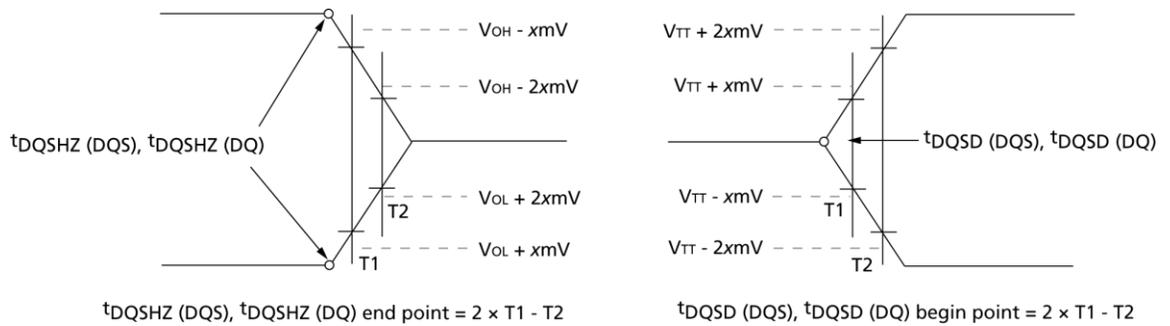
## E. MEASURING TIMING PARAMETERS TO/FROM TRI-STATE

There are several timing parameters that are measured to or from when:

- The device is no longer driving the NAND bus or a tri-state (hi-Z) condition
- The device begins driving from a tri-state (hi-Z) condition

These timing parameters include:  $t_{DQSD}$ ,  $t_{DQSHZ}$ ,  $t_{CHZ}$ ,  $t_{RHZ}$ , and  $t_{IR}$ . See section 4.2.

This appendix defines a two point method for measuring timing parameters that involve a tri-state condition. Figure 100 defines a method to calculate the point when the device is no longer driving the NAND bus or begins driving by measuring the signal at two different voltages. The voltage measurement points are acceptable across a wide range ( $x = 20$  mV up to  $x < 1/4$  of  $V_{CCQ}$ ). The figure uses  $t_{DQSHZ}$  and  $t_{DQSD}$  as examples. However, the method should be used for any timing parameter (asynchronous or source synchronous) that specifies that the device output is no longer driving the NAND bus or specifies that the device begins driving the NAND bus from a tri-state condition.



**Figure 100** Two point method for measuring timing parameters with tri-state condition

## F. EZ NAND: END TO END DATA PATH PROTECTION (INFORMATIVE)

An EZ NAND implementation may choose to support end to end data protection. In certain topologies with longer trace lengths or discontinuities, it may be helpful to include this capability. However, it is not expected that most EZ NAND implementations need this type of protection at speeds up to 200 MT/s.

One potential implementation is to include a two to four byte CRC for each data unit in the page. The data unit is called the CRC sub-page, which is the individual unit that is protected. The capability of end to end data protection should be reported in a vendor specific parameter page bit. The feature may then be configured utilizing a vendor specific Set Features. If there is an error detected as part of end to end data path protection, the error should be reported in the FAIL bit as part of the associated Read or Program operation.

Included below is a recommended method for configuring the CRC capabilities as part Set Features. The CRC polynomial to use is not specified.

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	Reserved (0)				CRC Mode		CRC Poly Length	
P2	# of CRC sub-pages							
P3	Offset of CRC bytes (bits 31:00)							
P4	Offset of CRC bytes (bits 63:32)							

CRC Poly Length	CRC polynomial length in bytes. Valid values are 2, 3 or 4.
CRC Mode	00b = no CRC check or generation 01b = CRC check on Program/store to Flash; CRC check on read from Flash 10b = CRC check and discard on Program/store to Flash; re-generate CRC on read from Flash 11b = Reserved
# of CRC sub-pages	Specifies the number of CRC sub-pages within a page. The CRC sub-page size is the page size divided by the number of CRC sub-pages.
Offset of CRC bytes	Specifies the offset of the CRC bytes within each CRC sub-page. E.g., a value of 0h indicates that the CRC is stored at the beginning of the CRC sub-page.