Open NAND Flash Interface Specification

Revision 1.0
28-December-2006

Hynix Semiconductor

Intel Corporation

Micron Technology, Inc.

Phison Electronics Corp.

Sony Corporation

STMicroelectronics

This 1.0 revision of the Open NAND Flash Interface specification ("Final Specification") is available for download at 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 IMPLEMNETATION 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.

For more information about ONFI, refer to the ONFI Workgroup website at www.onfi.org.

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

Table of Contents

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

1.2. References

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

- Hynix HY27UF084G2M data sheet available at http://www.hynix.com/datasheet/eng/flash/details/flash_11_HY27UF084G2M.jsp

- Micron MT29F4G08AAA data sheet available at http://download.micron.com/pdf/datasheets/flash/nand/4gb_nand_m40a.pdf

- ST NAND04GW3B2B data sheet available at http://www.st.com/stonline/products/literature/ds/12100/nand04gw3b2b.htm

1.3. Definitions, abbreviations, and conventions

1.3.1. 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.

1.3.1.1. address

The address is comprised of a row address and a column address. The row address identifies the page and block to be accessed. The column address identifies the byte or word within a page to access.

1.3.1.2. block

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

1.3.1.3. column

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

1.3.1.4. device

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

1.3.1.5. 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.

1.3.1.6. LUN (logical unit number)

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

1.3.1.7. na

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

1.3.1.8. 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.

1.3.1.9. page

The smallest addressable unit for read and program operations. For targets that support partial page programming, the smallest addressable unit for program operations is a partial page if there are partial programming constraints.

1.3.1.10. 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.

1.3.1.11. read request

A read request is when the RE# signal is toggled by the host.

1.3.1.12. row

Refers to the block and page to be accessed.

1.3.1.13. 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 5.10 for the definition of bit meanings within the status register.

1.3.1.14. target

An independent Flash component with its own CE# signal.

1.3.1.15. 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) byte is byte 0 and the most significant byte (upper) byte is byte 1. See Figure 1 for a description of the relationship between bytes, words and Dwords.

1.3.2. Conventions

The names of abbreviations and acronyms used as signal names are in all uppercase (e.g., CE#). 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.

1.3.2.1. 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.

1.3.2.2. Keywords

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

1.3.2.2.1. mandatory

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

1.3.2.2.2. may

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

1.3.2.2.3. 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.

1.3.2.2.4. 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.

1.3.2.2.5. 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.

1.3.2.2.6. should

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

1.3.2.3. Byte, word and Dword Relationships

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

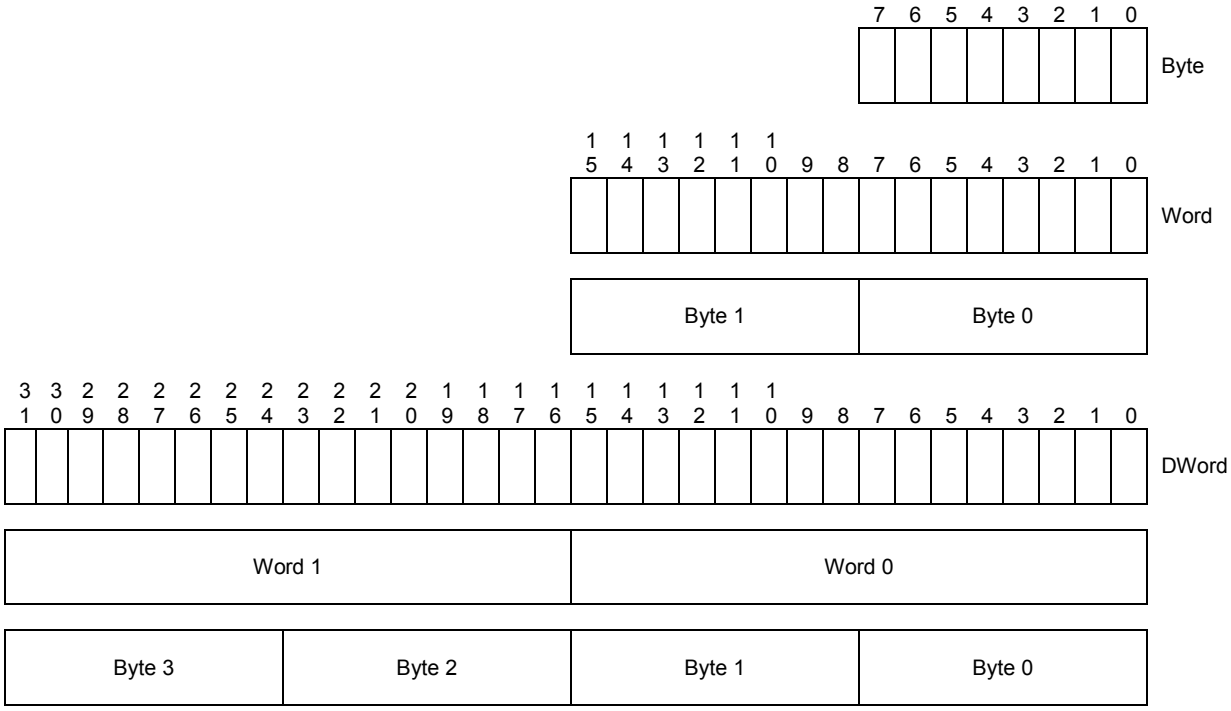

Figure 1 Byte, word and Dword relationships

1.3.2.4. Conventions

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

Table 1 State Table Cell Description

Each state is identified by a state designator and a state name. The state designator is unique among all states in all state diagrams in this document. The state designator consists of a set of letters that are capitalized followed by a unique number. The state name is a brief description of the primary action taken during the state, and the same state name may appear in other state diagrams. If the same primary function occurs in other states in the same state diagram, they are designated with a unique letter at the end of the name. Additional actions may be taken while in a state and these actions are described in the state description text.

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. The pinout for 8-bit data access is defined in the "x8" column and the pinout for 16-bit data access is defined in the "x16" column. 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.

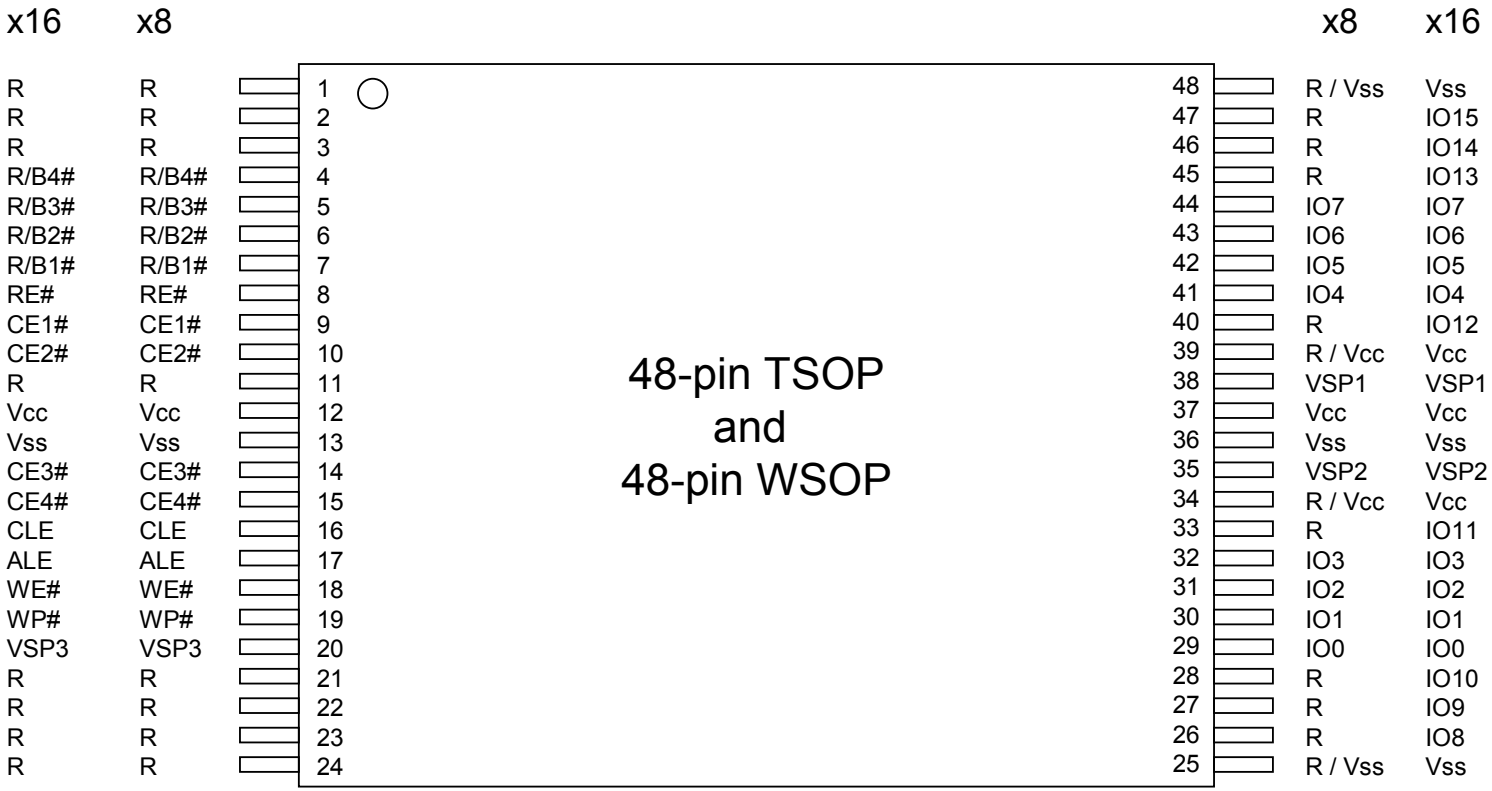

Figure 2 48-pin TSOP/WSOP pinout

2.2. LGA-52 Pad Assignments

Figure 3 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 4 defines the pad assignments for devices using 52-pad LGA packaging with 16-bit data access. The physical dimensions of the package are 12mmx17mm.

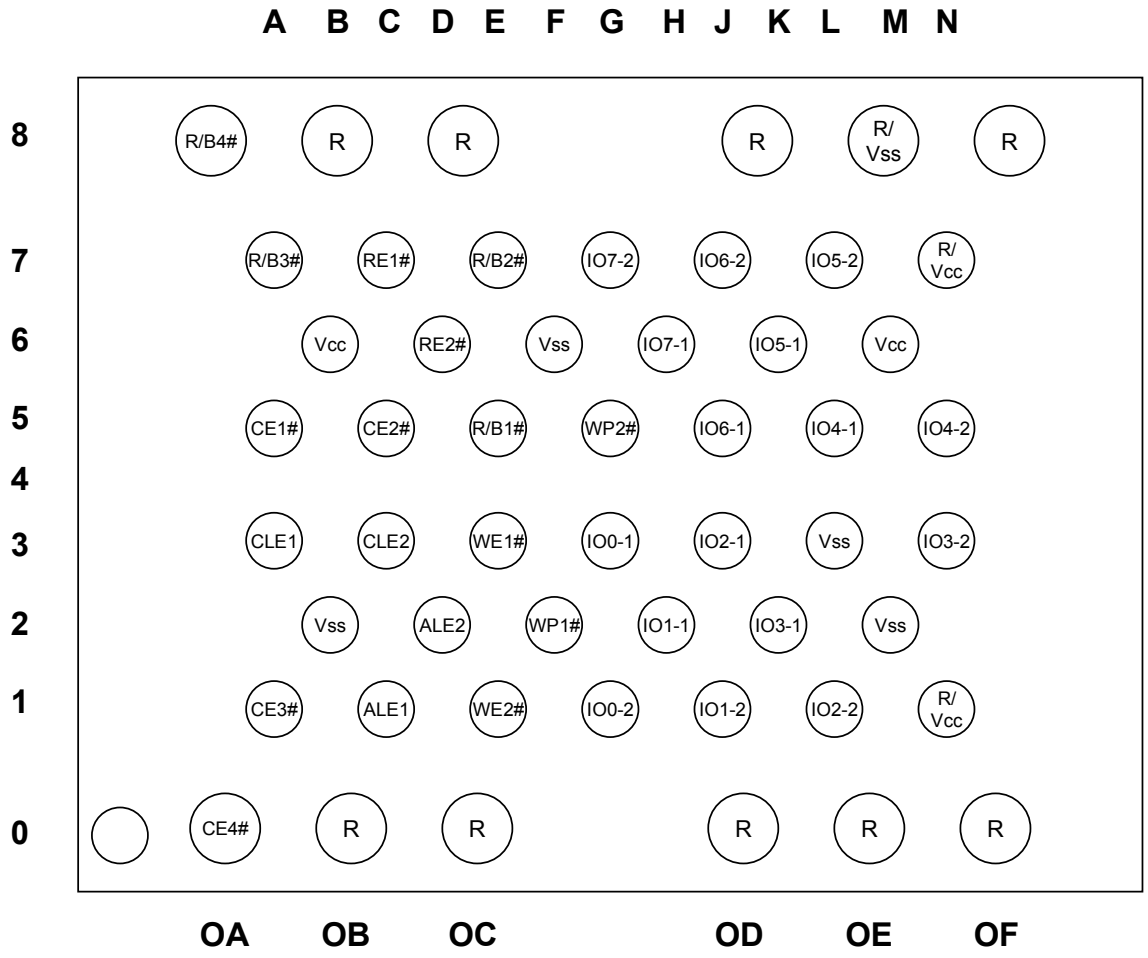

Figure 3 LGA pinout for 8-bit data access

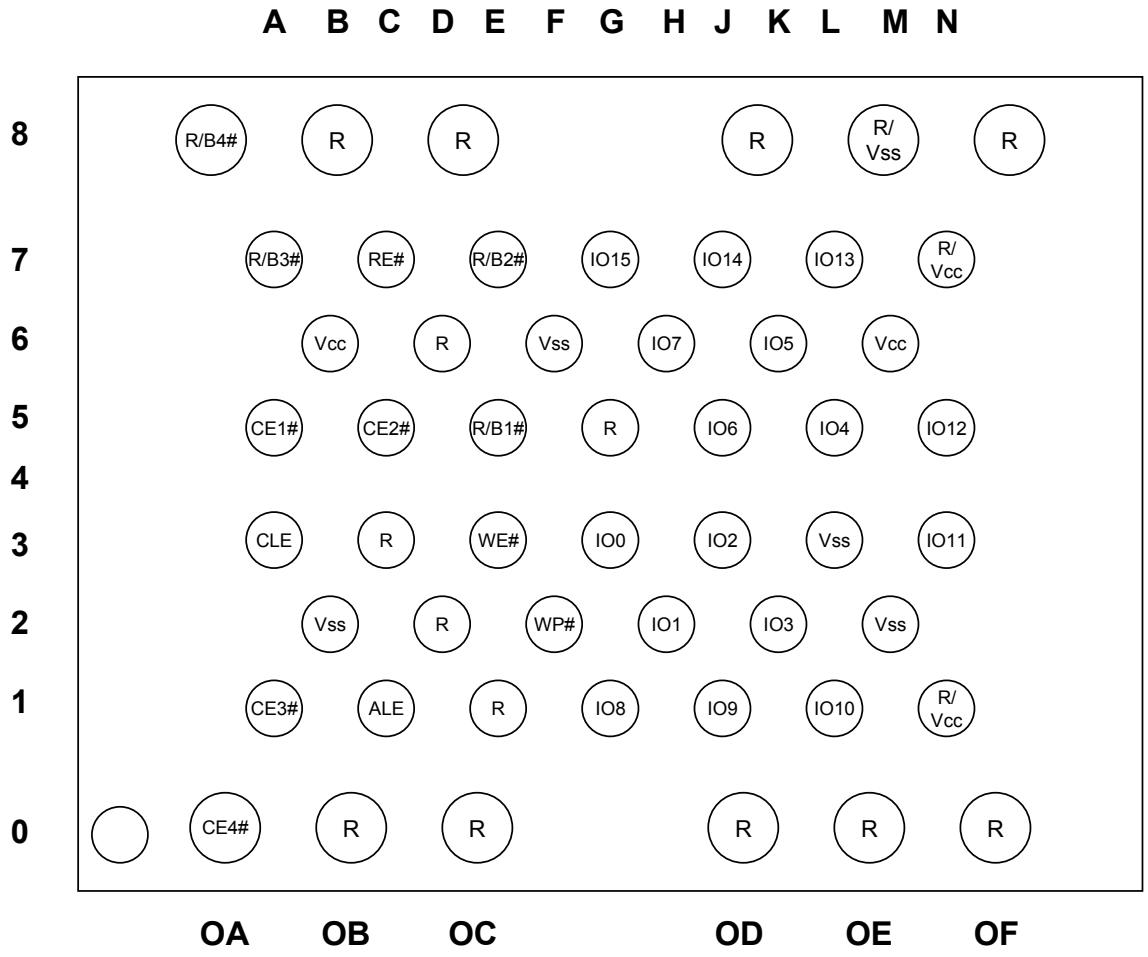

Figure 4 LGA pinout for 16-bit data access

2.3. BGA-63 Ball Assignments

Figure 5 defines the ball assignments for devices using 63-ball BGA packaging with 8-bit data access. Figure 6 defines the ball assignments for devices using 63-ball BGA packaging with 16-bit data access. Figure 7 defines the ball spacing requirements for the 63-ball BGA package.

	1	2	3	4	5	6	7	8	9	10
A	R	R							R	R
B	R								R	R
C			WP#	ALE	VSS	CE#	WE#	R/B#		
D			VCC	RE#	CLE	CE2#	CE3#	R/B2#		
E			R	R	R	R	CE4#	R/B3#		
F			R	R	R	R	VSS	R/B4#		
G			VSP3	VCC	VSP1	R	R	VSP2		
H			R	IO0	R	R	R	VCC		
J			R	IO1	R	VCC	IO5	IO7		
K			VSS	IO2	IO3	IO4	IO6	VSS		
L	R	R							R	R
M	R	R							R	R

Figure 5 BGA ball assignments for 8-bit data access

	1	2	3	4	5	6	7	8	9	10
A	R	R							R	R
B	R								R	R
C			WP#	ALE	VSS	CE#	WE#	R/B#		
D			VCC	RE#	CLE	CE2#	CE3#	R/B2#		
E			R	R	R	R	CE4#	R/B3#		
F			R	R	R	R	VSS	R/B4#		
G			VSP3	VCC	VSP1	IO13	IO15	VSP2		
H			IO8	IO0	IO10	IO12	IO14	VCC		
J			IO9	IO1	IO11	VCC	IO5	IO7		
K			VSS	IO2	IO3	IO4	IO6	VSS		
L	R	R							R	R
M	R	R							R	R

Figure 6 BGA ball assignments for 16-bit data access

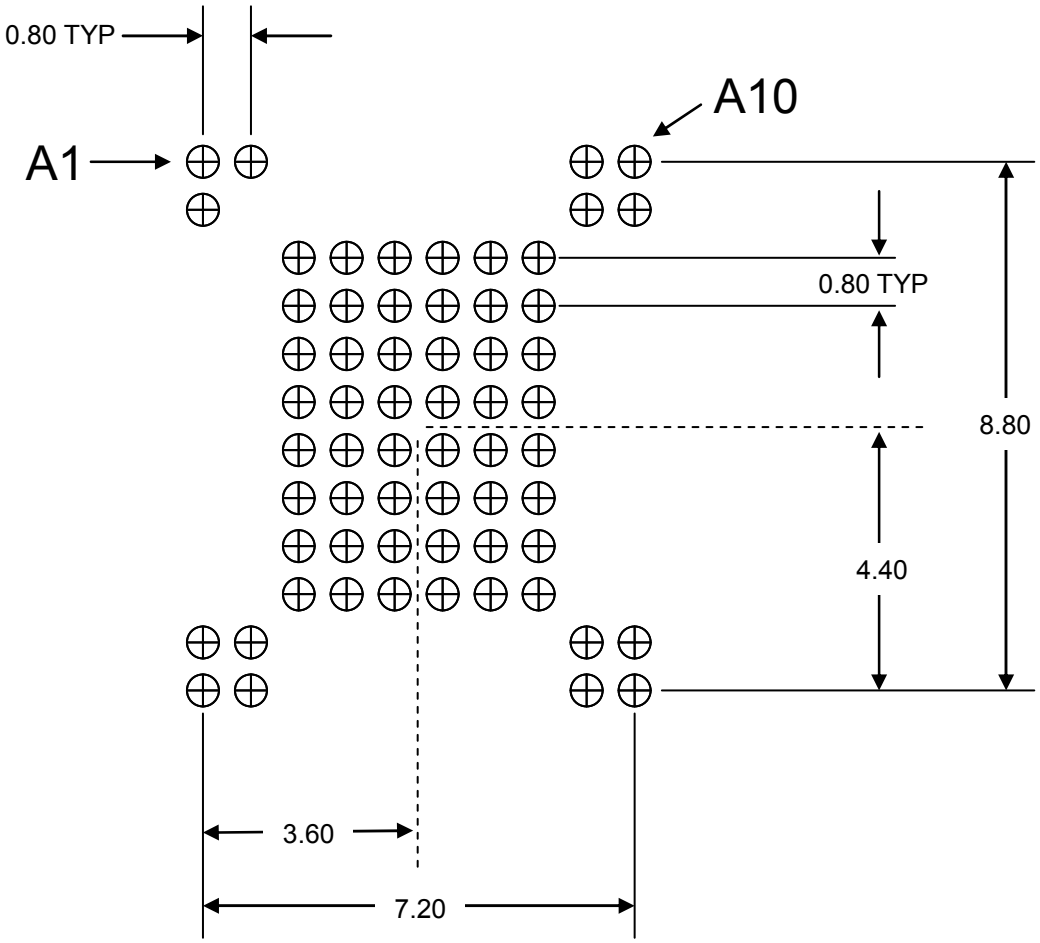

Figure 7 BGA ball spacing requirements (top view, dimensions in millimeters)

2.4. Signal Descriptions

Table 2 provides the signal descriptions.

Signal Name	Input / Output	Description
R/Bx#	O	**Ready/Busy** 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 2.13 for requirements.
REx#	I	**Read Enable** The Read Enable signal enables serial data output.
CEx#	I	**Chip Enable** 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 2.5 for additional requirements.

Signal Name	Input / Output	Description
Vcc	I	**Power** The Vcc signal is the power supply to the device.
R / Vcc	I	**Reserved / Power** The host shall always provide Vcc on these pins. The device shall tolerate Vcc on these pins but is not required to draw power from them.
Vss	I	**Ground** The Vss signal is the power supply ground.
R / Vss	I	**Reserved / Ground** The host shall always provide Vss on these pins. The device shall tolerate Vss on these pins but is not required to use them.
CLEx	I	**Command Latch Enable** The Command Latch Enable signal loads a command into the target. When high, the command is latched on the rising edge of WE#.
ALEx	I	**Address Latch Enable** The Address Latch Enable signal loads an address into the target. When high, the address is latched on the rising edge of WE#.
WEx#	I	**Write Enable** The Write Enable signal controls the latching of input data. Data, commands, and addresses are latched on the rising edge of WE#.
WPx#	I	**Write Protect** The Write Protect signal disables Flash array program and erase operations. See section 2.14 for requirements.
IO0 – IO7	I/O	**I/O Port 1, bits 0-7** The I/O port is an 8-bit wide bidirectional port for transferring address, command, and data to and from the device.
IO8 – IO15	I/O	**I/O Port 1, bits 8-15** 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.
IO0-2 – IO7-2	I/O	**I/O Port 2, bits 0-7** 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.
VSPx		**Vendor Specific** 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		**Reserved** 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. Devices may be implemented with other package types and be ONFI compliant if all other ONFI requirements within this specification are satisfied.

Any signal that does not have an associated number is implicitly numbered “1”. For example, WP# is equivalent to WP1#.

Signal Name	TSOP/ WSOP x8		TSOP/ WSOP x16		LGA x8		LGA x16		BGA x8		BGA x16	
R/B1# R/B2# R/B3# R/B4#	7 6 5 4	M O O O	Same as TSOP/ WSOP x8		E5 E7 A7 OA8	M O O O	Same as LGA x8		C8 D8 E8 F8	M O O O	Same as BGA x8	
RE1# RE2#	8 na	M	Same as TSOP/ WSOP x8		C7 D6	M O	C7 na	M	D4 na	M	Same as BGA x8	
CE1# CE2# CE3# CE4#	9 10 14 15	M O O O	Same as TSOP/ WSOP x8		A5 C5 A1 OA0	M O O O	Same as LGA x8		C6 D6 D7 E7	M O O O	Same as BGA x8	
Vcc	12 37	M M	12 34 37 39	M M M M	B6 M6	M M	Same as LGA x8		D3 G4 H8 J6	M M M M	Same as BGA x8	
R / Vcc	34 39	M M			N1 N7	M M	Same as LGA x8					
Vss	13 36	M M	13 25 36 48	M M M M	B2 F6 L3 M2	M M M M	Same as LGA x8		C5 F7 K3 K8	M M M M	Same as BGA x8	
R / Vss	25 48	M M			OE8	M	Same as LGA x8					
CLE1 CLE2	16 na	M	Same as TSOP/ WSOP x8		A3 C3	M O	A3 na	M	D5 na	M	Same as BGA x8	
ALE1 ALE2	17 na	M	Same as TSOP/ WSOP x8		C1 D2	M O	C1 na	M	C4 na	M	Same as BGA x8	
WE1# WE2#	18 na	M	Same as TSOP/ WSOP x8		E3 E1	M O	E3 na	M	C7 na	M	Same as BGA x8	
WP1# WP2#	19 na	M	Same as TSOP/ WSOP x8		F2 G5	M O	F2 na	M	C3 na	M	Same as BGA x8	
IO0 IO1 IO2 IO3 IO4 IO5 IO6 IO7	29 30 31 32 41 42 43 44	M M M M M M M M	Same as TSOP/ WSOP x8		G3 H2 J3 K2 L5 K6 J5 H6	M M M M M M M M	Same as LGA x8		H4 J4 K4 K5 K6 J7 K7 J8	M M M M M M M M	Same as BGA x8	

Signal Name	TSOP/ WSOP x8		TSOP/ WSOP x16		LGA x8		LGA x16		BGA x8		BGA x16	
IO8	na		26	M	na		G1	M	na		H3	M
IO9	na		27	M	na		J1	M	na		J3	M
IO10	na		28	M	na		L1	M	na		H5	M
IO11	na		33	M	na		N3	M	na		J5	M
IO12	na		40	M	na		N5	M	na		H6	M
IO13	na		45	M	na		L7	M	na		G6	M
IO14	na		46	M	na		J7	M	na		H7	M
IO15	na		47	M	na		G7	M	na		G7	M
IO0-2	na		Same as TSOP/ WSOP x8		G1	O	na		na		Same as BGA x8	
IO1-2	na				J1	O	na		na			
IO2-2	na				L1	O	na		na			
IO3-2	na				N3	O	na		na			
IO4-2	na				N5	O	na		na			
IO5-2	na				L7	O	na		na			
IO6-2	na				J7	O	na		na			
IO7-2	na				G7	O	na		na			
VSP1	38	O	Same as TSOP/ WSOP x8						G5	O	Same as BGA x8	
VSP2	35	O							G8	O		
VSP3	20	O							G3	O		

Table 3 Signal mappings

2.5. CE# Signal Requirements

When R/B# is cleared to zero, the CE# signal may be transitioned to a value of one without impacting the ongoing commands within the target. After the CE# signal is transitioned to one, the host may drive a different CE# signal to zero and begin operations on another target.

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

2.6. Absolute Maximum Ratings

Stresses greater than those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only. Operation beyond the conditions specified in the “Recommend Operating Conditions” (Table 5) and “DC and Operating Characteristics” tables (Table 6 and Table 7) is not recommended. Extended or periodic exposure beyond these conditions may affect device reliability.

Parameter	Symbol	Rating	Units
Voltage on any pin relative to Vss for 3.3V devices	V_{CC}	-0.6 to +4.6	V
	V_{IN}	-0.6 to +4.6	
	V_{IO}	-0.6 to +4.6	
Voltage on any pin relative to Vss for 1.8V devices	V_{CC}	-0.2 to +2.4	V
	V_{IN}	-0.2 to +2.4	
	V_{IO}	-0.2 to +2.4	

Table 4 Absolute maximum ratings

2.7. Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Supply voltage for 3.3V devices	V_{CC}	2.7	3.3	3.6	V
Supply voltage for 1.8V devices	V_{CC}	1.7	1.8	1.95	V
Supply voltage	V_{SS}	0	0	0	V

Table 5 Recommended operation conditions

2.7.1. Provisions for I/O power (Vccq) and I/O ground (Vssq)

Vccq and Vcc are identical in this specification. However, Vccq may be lower than Vcc in future revisions of this specification. Future revisions of this specification will specify Vccq voltage ranges as well as signal locations on the packages ONFI specifies.

2.8. 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 (like solid state drives) it may be desirable to draw increased current for ICC1-ICC3. 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 µF capacitor.

The maximum leakage current requirements (ILI and ILO) in Table 6 and Table 7 are tested across the entire allowed Vcc range, specified in Table 5.

Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Operating current Read Page with serial access	ICC1	tRC= t_{RC} (min), CE#=VIL, IOUT=0 mA	-	-	50	mA
Operating current Program	ICC2	-	-	-	50	mA
Operating current Erase	ICC3	-	-	-	50	mA
Standby current	ISB1	CE#=VIH, WP#=0 V	-	-	1	mA
Standby current	ISB2	CE#=Vcc-0.2, WP#=0 V	-	-	50	µA
Input leakage current	ILI	VIN=0 to Vcc	-	-	+-10	µA
Output leakage current	ILO	VOUT=0 to Vcc	-	-	+-10	µA
Input high voltage	VIH	-	Vcc * 0.8	-	Vcc+0.3	V
Input low voltage	VIL	-	-0.3	-	Vcc * 0.2	V
Output high voltage	VOH	IOH=-400 µA	Vcc * 0.67	-	-	V
Output low voltage	VOL	IOL=2.1 mA		-	0.4	V
Output low current (R/B#)	IOL(R/B#)	VOL=0.4 V	8	10	-	mA
Staggered power-up current	IST	tRise = 1 ms cLine = 0.1 µF			10 per LUN	mA

Table 6 DC and Operating Conditions for 3.3V devices

Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Operating current Read Page with serial access	ICC1	tRC= t_{RC} (min), CE#=VIL, IOUT=0 mA	-	-	50	mA
Operating current Program	ICC2	-	-	-	50	mA
Operating current Erase	ICC3	-	-	-	50	mA
Standby current	ISB1	CE#=VIH, WP#=0V	-	-	1	mA
Standby current	ISB2	CE#=Vcc-0.2, WP#=0 V	-	-	50	µA
Input leakage current	ILI	VIN=0 to Vcc	-	-	+-10	µA
Output leakage current	ILO	VOUT=0 to Vcc	-	-	+-10	µA
Input high voltage	VIH	-	Vcc * 0.8	-	Vcc+0.3	V
Input low voltage	VIL	-	-0.3	-	Vcc * 0.2	V
Output high voltage	VOH	IOH=-100 µA	Vcc * 0.67	-	-	V
Output low voltage	VOL	IOL=100 µA	-	-	0.1	V
Output low current (R/B#)	IOL(R/B#)	VOL=0.2 V	3	4	-	mA
Staggered power-up current	IST	tRise = 1 ms cLine = 0.1 µF			10 per LUN	mA

Table 7 DC and Operating Conditions for 1.8V devices

2.9. 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 5.4). The algorithm to use is:

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

This methodology will calculate an accurate pin capacitance, accounting for all targets present.

2.10. 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 command is received by the target after power-on, the target shall not draw more than 10 mA of current per LUN. For example, a target that contains 4 LUNs may draw up to 40 mA of current until a Reset 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 µF. The measurement shall be taken with 1 millisecond averaging intervals and shall begin after Vcc reaches Vcc_min.

2.11. Independent Data Buses

There may be two independent 8-bit data buses when using the LGA pinout. If the device supports two independent data buses, then CE2# and CE4# (if connected) shall use the second data bus. CE1# and CE3# shall always use the first data bus pins. Note that CE1#, CE2#, CE3#, and CE4# may all use the first data bus and the first set of control signals (RE1#, CLE1, ALE1, WE1#, and WP1#) if the device does not support independent data buses.

Table 8 defines the control signal to CE# signal mapping for the LGA x8 package. Note that there is no independent data bus capability for the other ONFI defined pinouts.

Signal Name	CE
R/B1#	CE1#
R/B2#	CE2#
R/B3#	CE3#
R/B4#	CE4#
RE1#	CE1#, CE3#
RE2#	CE2#, CE4#
CLE1	CE1#, CE3#
CLE2	CE2#, CE4#
ALE1	CE1#, CE3#
ALE2	CE2#, CE4#
WE1#	CE1#, CE3#
WE2#	CE2#, CE4#
WP1#	CE1#, CE3#
WP2#	CE2#, CE4#

Table 8 LGA x8 Dual Data Bus Signal to CE# mapping

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

2.12. 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.

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.

2.13. Ready/Busy (R/B#) Requirements

2.13.1. Power-On Requirements

Once V_{CC} reaches the V_{CC} minimum value listed in Table 5 and power is stable, the R/B# signal shall be valid after 10 µs and shall be set to one (Ready) within 1 ms.

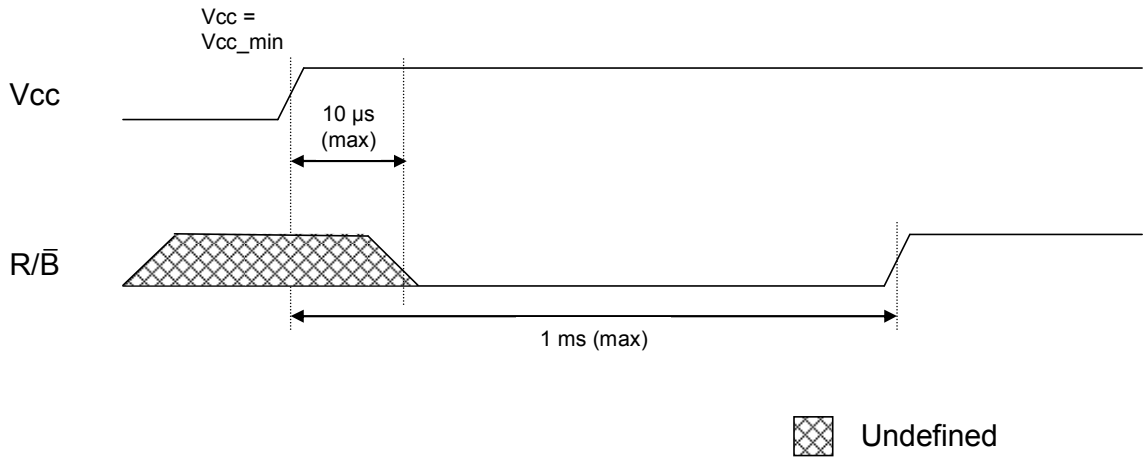

Figure 8 R/B# Power-On Behavior

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# circuit determines the rise time of R/B#.

2.13.2. R/B# and SR[6] relationship

R/B# 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# is the logical AND of the SR[6] values for all LUNs on CE3#. Thus, R/B# reflects whether any LUN is busy on a particular target.

2.14. Write Protect

When cleared to zero, the WP# 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#, the host shall not issue a new command to the device for at least tWW delay time.

3. Memory Organization

Figure 9 describes the basic memory and device organization. In some implementations, additional page registers may be present within each logical unit.

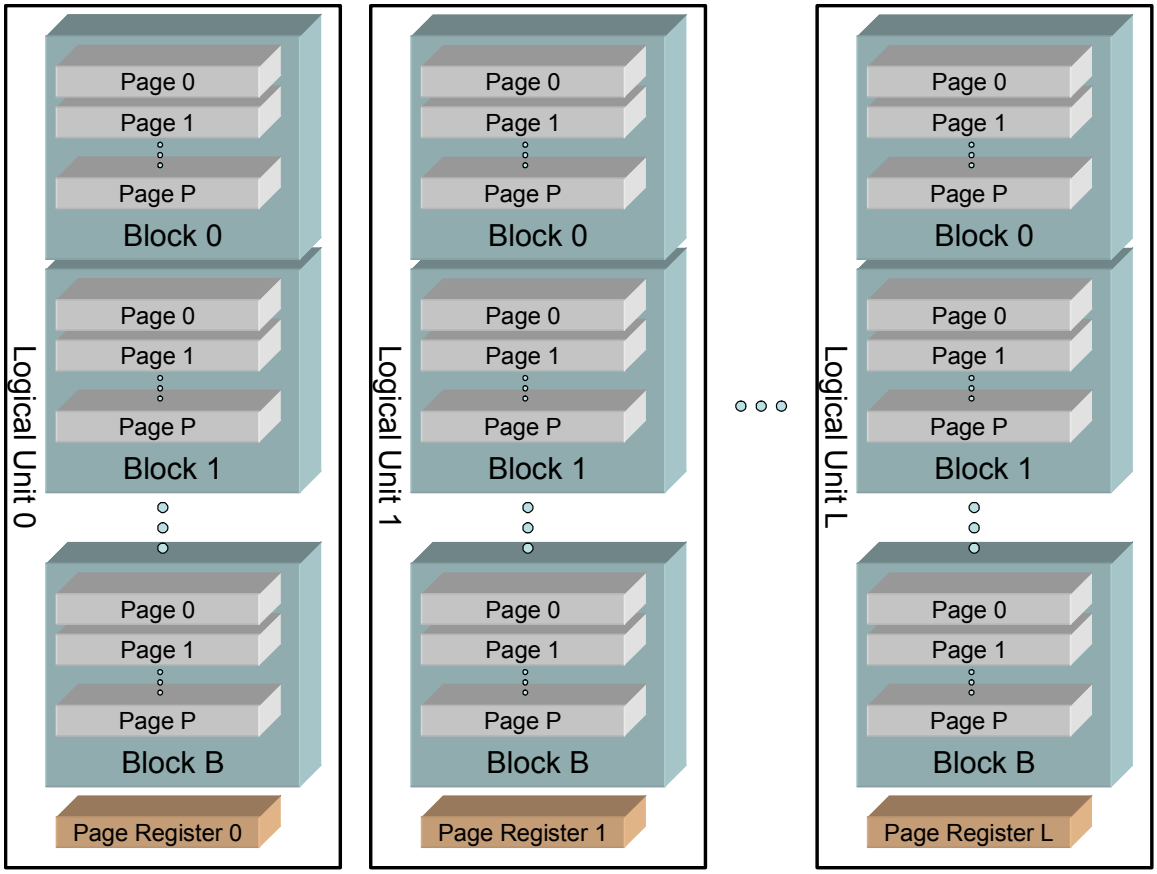

Figure 9 Target memory organization

A device contains one or more targets. A target is controlled by one CE# 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. 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 interleaved 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. For targets that support partial page programming with constraints, the smallest addressable unit for program operations is a partial page. 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.

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

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 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 10 with the least significant row address bit to the right and the most significant row address bit to the left.

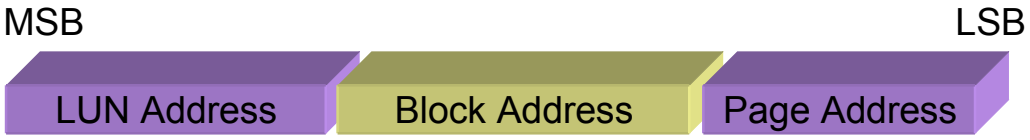

Figure 10 Row Address Layout

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 that range that are 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. Interleaved Addressing

The interleaved address comprises the lowest order bits of the block address as shown in Figure 11.

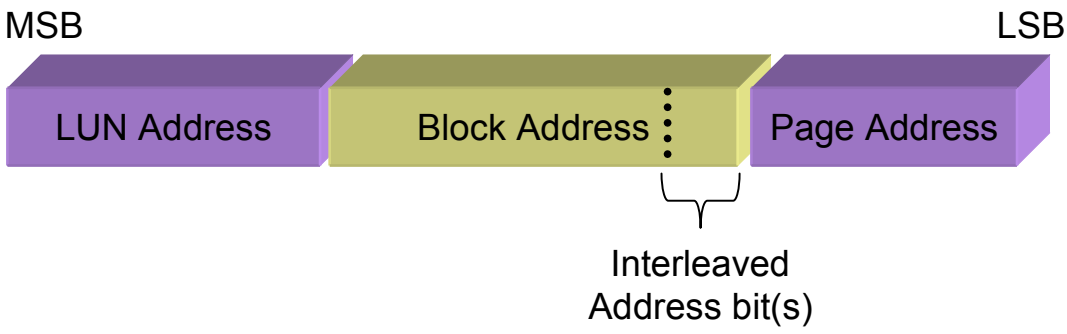

Figure 11 Interleaved Address Location

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 operations. This implies that R/B# 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 will clear their page registers. Thus, the host should not begin a Page Program command on a LUN while a Read operation is ongoing or completed on another LUN, as the contents of the page register for the Read operation will be lost. A Read can be issued to another LUN while a Page Program is ongoing within another LUN without any restriction.

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 toggle of the RE# input signal. 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 subsequent toggle of

the RE# input signal after which data output is selected with the 00h command thus avoiding bus contention.

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 at least one byte in the defect area, as shown in Figure 12, 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 at least one word in the defect area of the first or last page of the defective block to a value of 0000h.

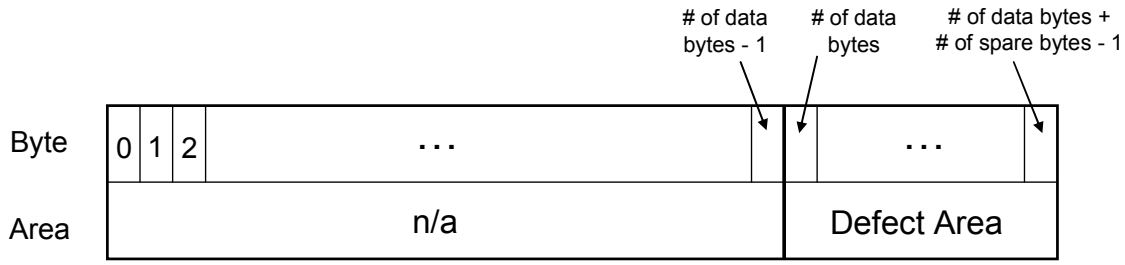

Figure 12 Area marked in factory defect mapping

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 13 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. An 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 any page byte/word location in the defect area of either the first page or last page of the block. The host shall check 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);
        for (column=PageSize; column<PageSize+SpareBytes; column++)
        {
            if (Buff[column] == 00h)        // Value checked for is 0000h for 16-bit access
                Defective=TRUE;
        }

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

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

Figure 13 Factory defect scanning algorithm

3.3. Discovery and Initialization

3.3.1. CE# Discovery

There may be up to four chip enable (CE#) 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# signal. CE# signals shall be used sequentially on the device; CE1# is always connected and CE# signals shall be connected in a numerically increasing order. The host shall attempt to enumerate targets connected to all host CE# 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 package with 8-bit data access is the only defined package within ONFI that has a dual data bus option.

3.3.1.1. Single Data Bus Discovery

The CE# to test is first pulled low by the host to enable the target if connected, while all other CE# signals are pulled high. The host shall then issue the Reset command to the target. Following the reset, the host shall 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# is not connected and no further use of that CE# signal shall be done.

3.3.1.2. Dual Data Bus Discovery

The CE# to test is first pulled low by the host to enable the target if connected, while all other CE# signals are pulled high. The host shall then issue the Reset command to the target. Following

the reset, the host shall 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 command to the target using the second 8-bit data bus. Following the reset, the host shall 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# is not connected and no further use of that CE# signal shall be done.

3.3.2. Target Initialization

To initialize a discovered target, the following steps shall be taken. The initialization process shall be followed for each connected CE# 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 shall 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 5.4.1.36), 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 5.4.1.1), 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 shall next map out all defective blocks in the target. The host may then proceed to utilize the target, including erase and program operations.

3.4. Partial Page Programming

3.4.1. Requirements

Programming multiple partial pages (data and corresponding spare) in a single program operation is allowed.

The “Number of programs per page” parameter (refer to section 5.4.1.22) shall not be exceeded by the host. If this parameter is less than the number of partial pages (defined in section 3.4.2), then multiple partial pages need to be programmed in a single program operation in order to not exceed this attribute.

If the host does not support the partial page layout and boundaries that the target supports, then the host shall not use partial page programming with the target.

3.4.2. Host Discovery

The following flow describes the process by which the host discovers the constraints and attributes of partial page programming that the target may support.

1. The host determines if the target supports partial page programming by checking the “Number of programs per page” field in the parameter page. If set to a value larger than one, then the target supports partial page programming.

2. If the target supports partial page programming, the host determines if there are any constraints for partial page programming. The host checks the “Partial page programming attributes” field in the parameter page. If bit 0 is set to zero, then there are no constraints and the host may issue partial programs starting at any byte/word offset with any size. If bit 0 is set to one, then there are constraints.

3. If there are constraints for partial page programming, then the host checks the Constraints subfield in the “Partial page programming attributes” field in the parameter page. If there are constraints, then partial pages shall be written on partial page boundaries in partial page multiples according to the data / spare layout that the target indicates is supported.

4. The host determines the number of partial pages by the equation below. Note that the target may have an arbitrary amount of additional spare area at the end of the full page.

   Number of partial pages =
   Number of data bytes per page /
   Number of data bytes per partial page

4. Timing Diagrams

All timing parameters are from a host perspective. For example, the “Minimum WE# pulse width” is the minimum allowed WE# 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.

Table 9 defines the descriptions of all timing parameters. Table 12 and Table 13 define the requirements for timing modes 0, 1, 2, 3, 4, and 5. Note that tR, tPROG, tBERS, and tCCS are not part of the timing mode definition and are separately returned in the parameter page. Timing mode 0 shall always be supported and shall be used by the host 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.

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, the value for tR and tCCS contained in the parameter page shall be used.

The host shall use EDO data output cycle timings, as defined in section 4.5, when running with a tRC value less than 30 ns.

There are three maximums listed for tRST in timing modes 1-5. 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.

Parameter	Description
tADL	ALE to data loading time
tALH	ALE hold time
tALS	ALE setup time
tAR	ALE to RE# delay
tBERS[1]	Block erase time
tCEA	CE# access time
tCCS	Change Column setup time
tCH	CE# hold time
tCHZ	CE# high to output hi-Z
tCLH	CLE hold time
tCLR	CLE to RE# delay
tCLS	CLE setup time
tCOH	CE# high to output hold
tCS	CE# setup time
tDH	Data hold time
tDS	Data setup time
tFEAT[1]	Busy time for Set Features and Get Features
tIEBSY[1]	Busy time for interleaved erase operation
tIPBSY[1]	Busy time for interleaved program operation
tIR	Output hi-Z to RE# low
tPCBSY	Program cache busy time
tPROG[1]	Page program time
tR[1]	Page read time
tRC	RE# cycle time
tRCBSY[1]	Read cache busy time
tREA	RE# access time
tREH	RE# high hold time
tRHOH	RE# high to output hold
tRHW	RE# high to WE# low
tRHZ	RE# high to output hi-Z
tRLOH	RE# low to output hold
tRP	RE# pulse width
tRR	Ready to RE# low
tRST	Device reset time, measured from the rising edge of WE# to the rising edge of R/B#.
tWB	WE# high to R/B# low
tWC	WE# cycle time
tWH	WE# high hold time
tWHR	WE# high to RE# low
tWP	WE# pulse width
tWW	WP# transition to WE# low

NOTE:
1. Measured from the falling edge of SR[6] to the rising edge of SR[6].

Table 9 Timing Parameter Descriptions

The testing conditions that shall be used when determining whether a device supports a particular timing mode are listed in Table 10.

Parameter	Value
Input pulse levels	0.0 V to Vcc
Input rise and fall times	5 ns
Input and output timing levels	Vcc / 2
Output load for 3.3V	1 TTL gate and CL = 50 pF
Output load for 1.8V	1 TTL gate and CL = 30 pF

Table 10 Testing Conditions for Timing Modes

There are “short” busy times associated with cache operations (tRCBSY, tPCBSY) and interleaved operations (tIEBSY and tIPBSY). Typical and maximum times for these busy times are listed in Table 11.

Parameter	Typical	Maximum
tIEBSY	500 ns	tBERS
tIPBSY	500 ns	tPROG
tPCBSY	3 µs	tPROG
tRCBSY	3 µs	tR

Table 11 Cache and Interleave Short Busy Times

Parameter	Mode 0		Mode 1		Mode 2		Unit
	100		50		35		ns
	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
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
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		1000		5/10/ 500		5/10/ 500	µs
tWB		200		100		100	ns
tWC	100		45		35		ns
tWH	30		15		15		ns
tWHR	120		80		80		ns
tWP	50		25		17		ns
tWW	100		100		100		ns

NOTE:
1. To easily support EDO capable devices, tCHZ and tRHZ maximums are higher in modes 1, 2, and 3 than typically necessary for a non-EDO capable device.

Table 12 Timing Modes 0, 1, and 2

Parameter	Mode 3		Mode 4 (EDO capable)		Mode 5 (EDO capable)		Unit
	30		25		20		ns
	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
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
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		5/10/ 500		5/10/ 500		5/10/ 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:
1. To easily support EDO capable devices, tCHZ and tRHZ maximums are higher in modes 1, 2, and 3 than typically necessary for a non-EDO capable device.

Table 13 Timing Modes 3, 4, and 5

4.1. Command Latch Timings

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

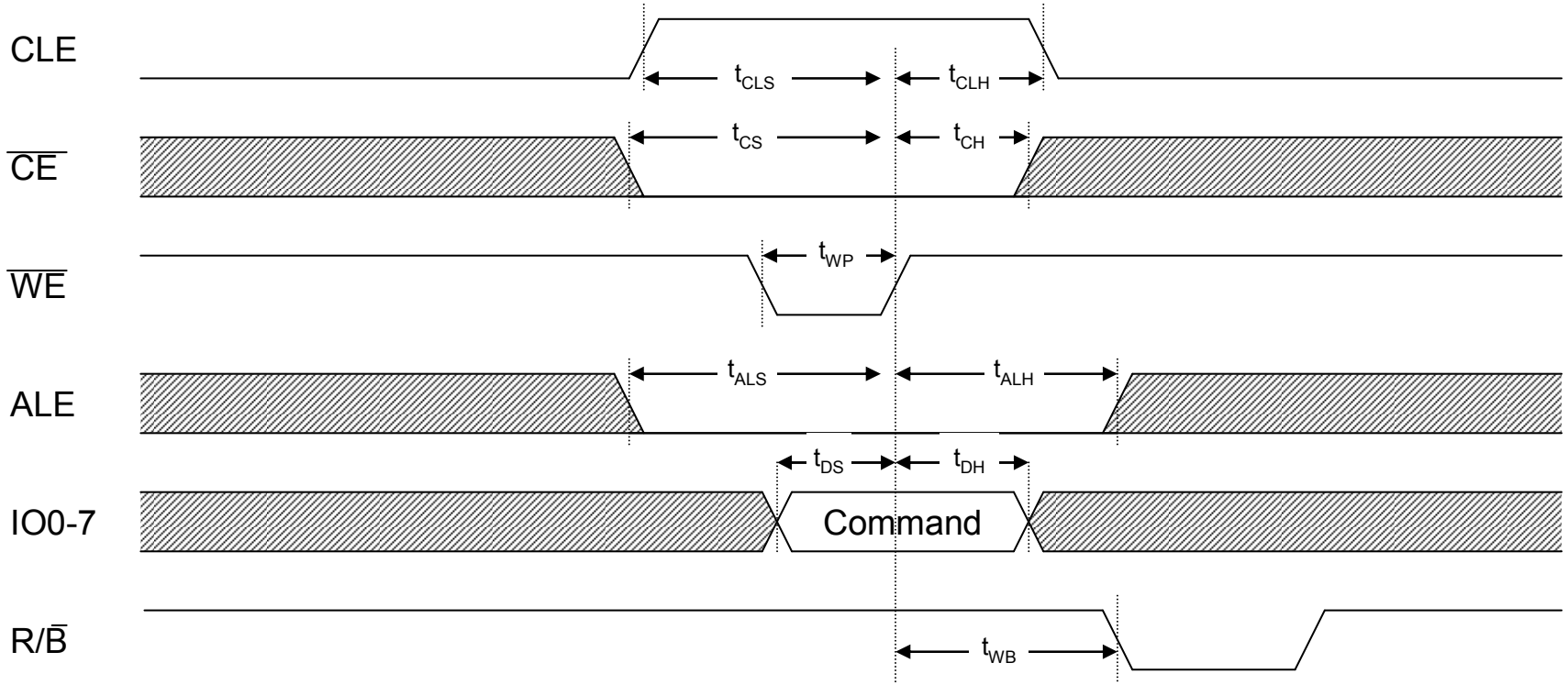

Figure 14 Command latch timings

4.2. Address Latch Timings

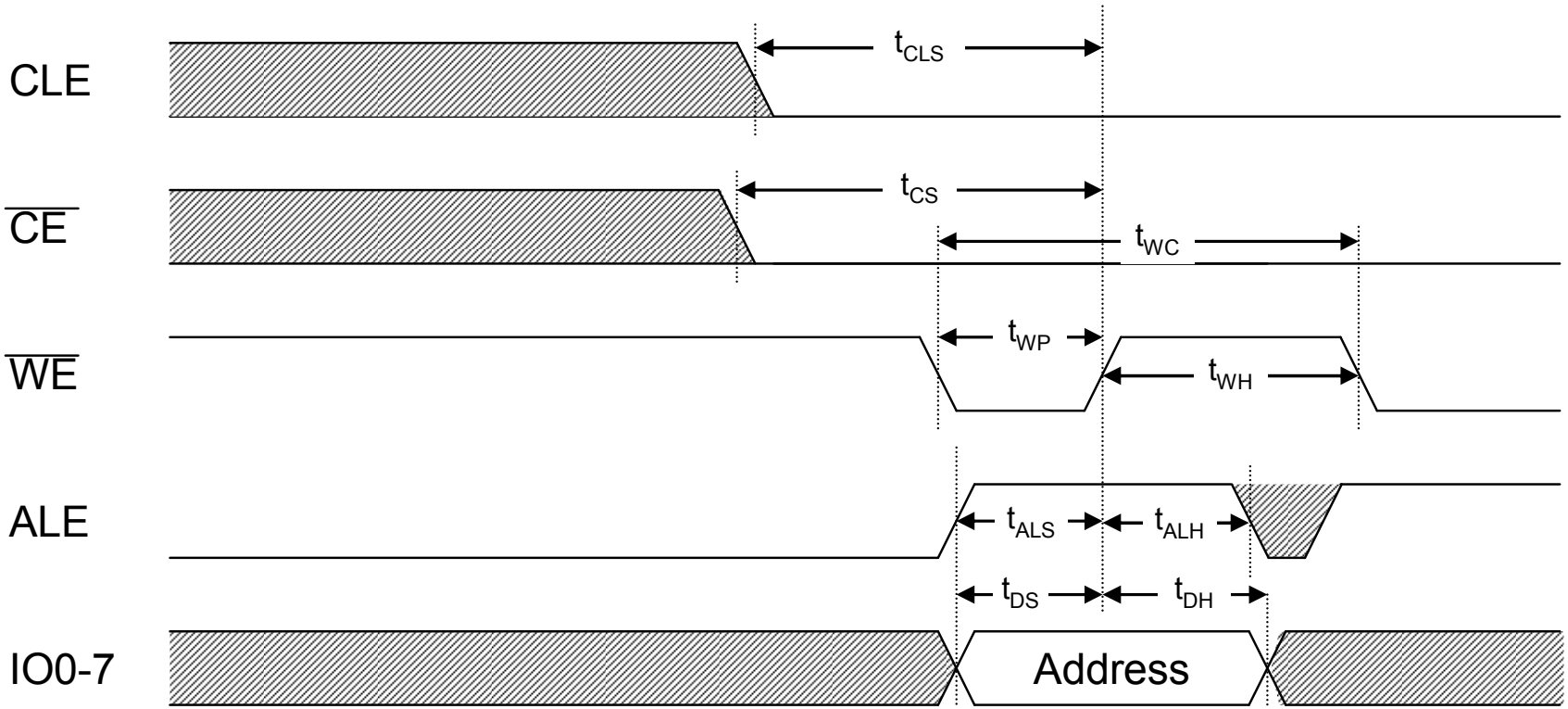

Figure 15 Address latch timings

4.3. Data Input Cycle Timings

Data input may be used with CE# don't care. However, if CE# don't care is used tCS and tCH timing requirements shall be met by the host.

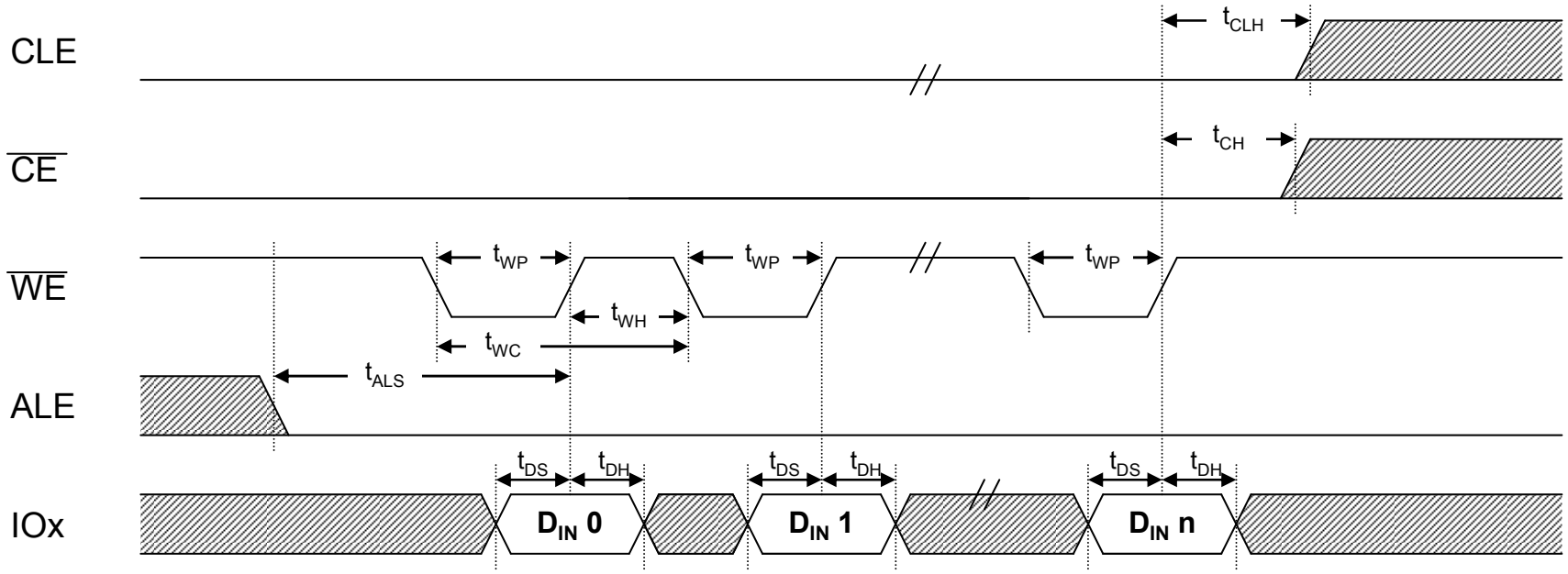

Figure 16 Data input cycle timings

4.4. Data Output Cycle Timings

Data output may be used with CE# don't care. However, if CE# don't care is used tCEA and tCOH timing requirements shall be met by the host.

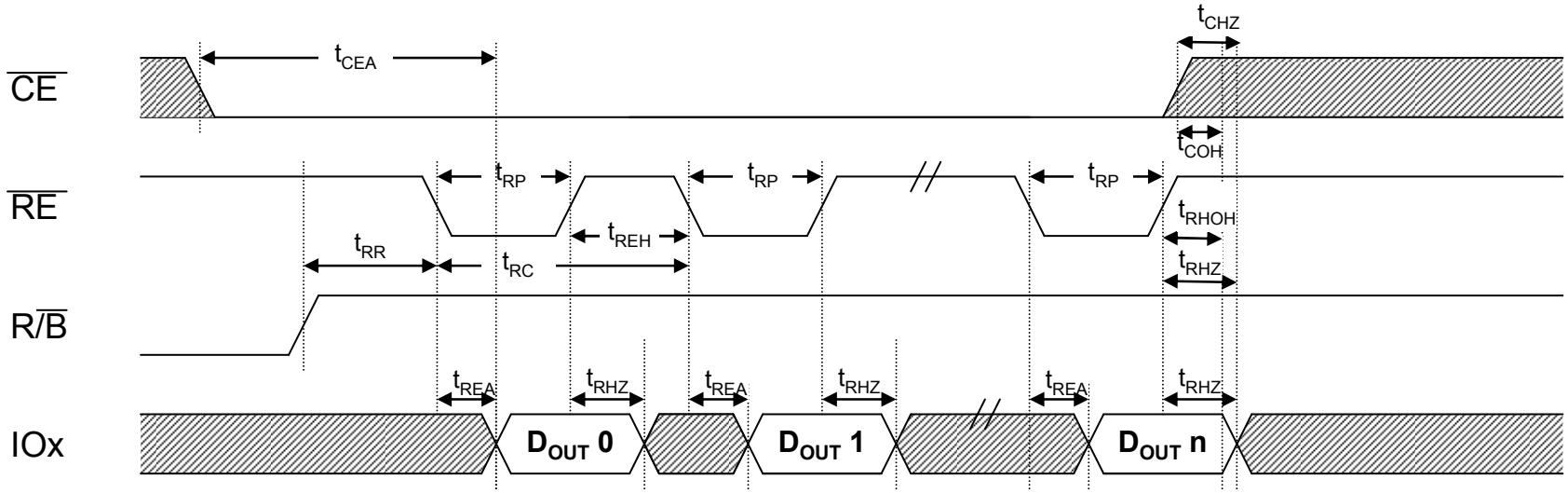

Figure 17 Data output cycle timings

4.5. Data Output Cycle Timings (EDO)

EDO data output cycle timings shall be used if the host drives tRC less than 30 ns. Data output may be used with CE# don't care. However, if CE# don't care is used tCEA and tCOH timing requirements shall be met by the host.

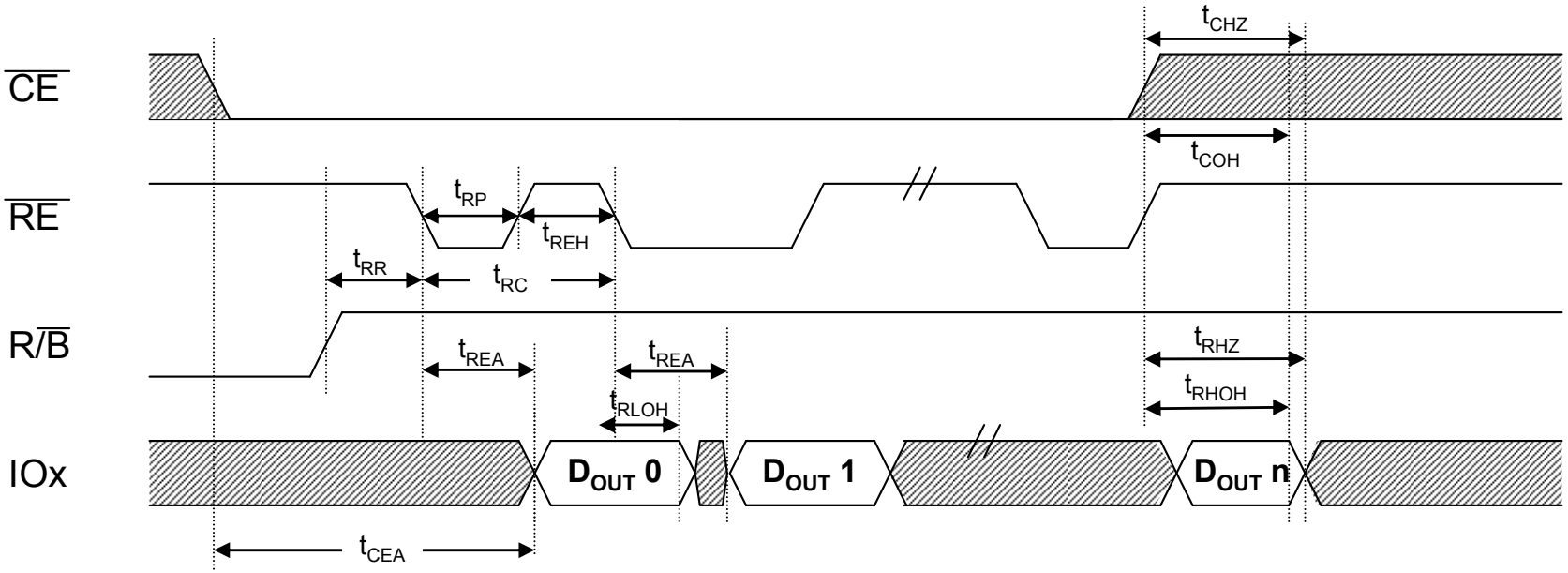

Figure 18 EDO data output cycle timings

4.6. Read Status Timings

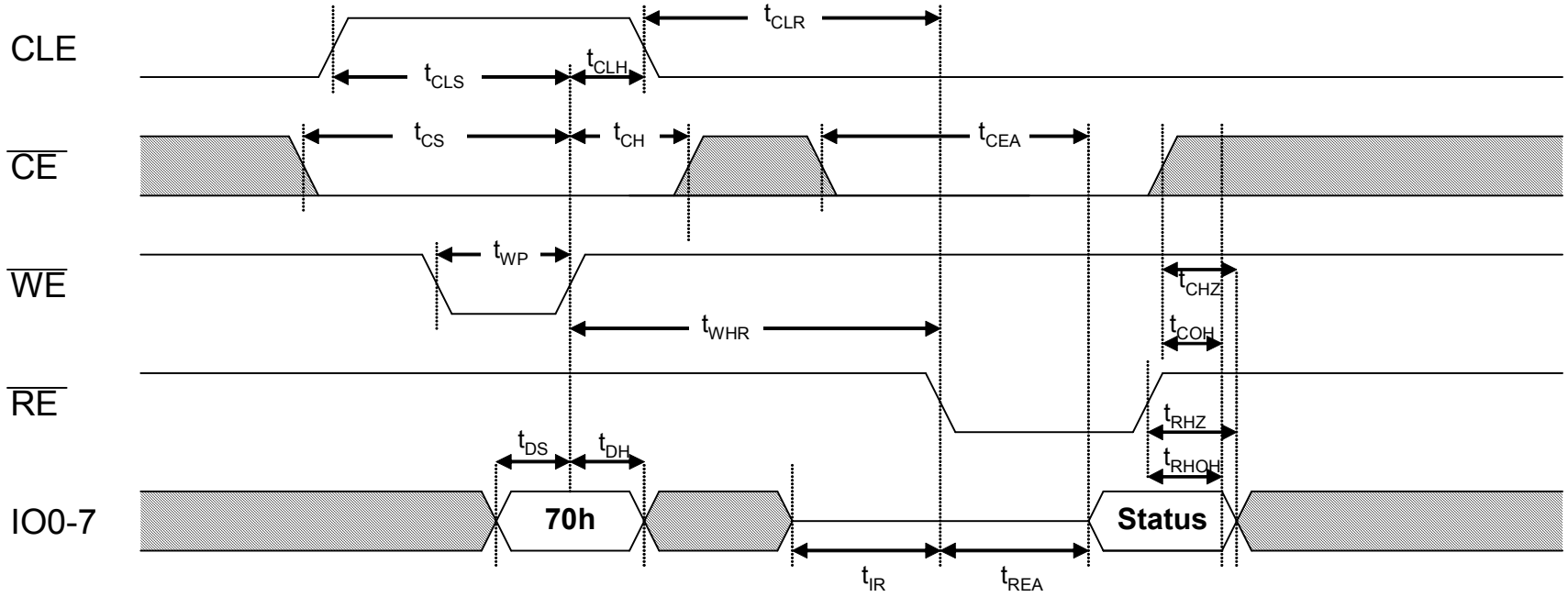

Figure 19 Read Status timings

4.7. Read Status Enhanced Timings

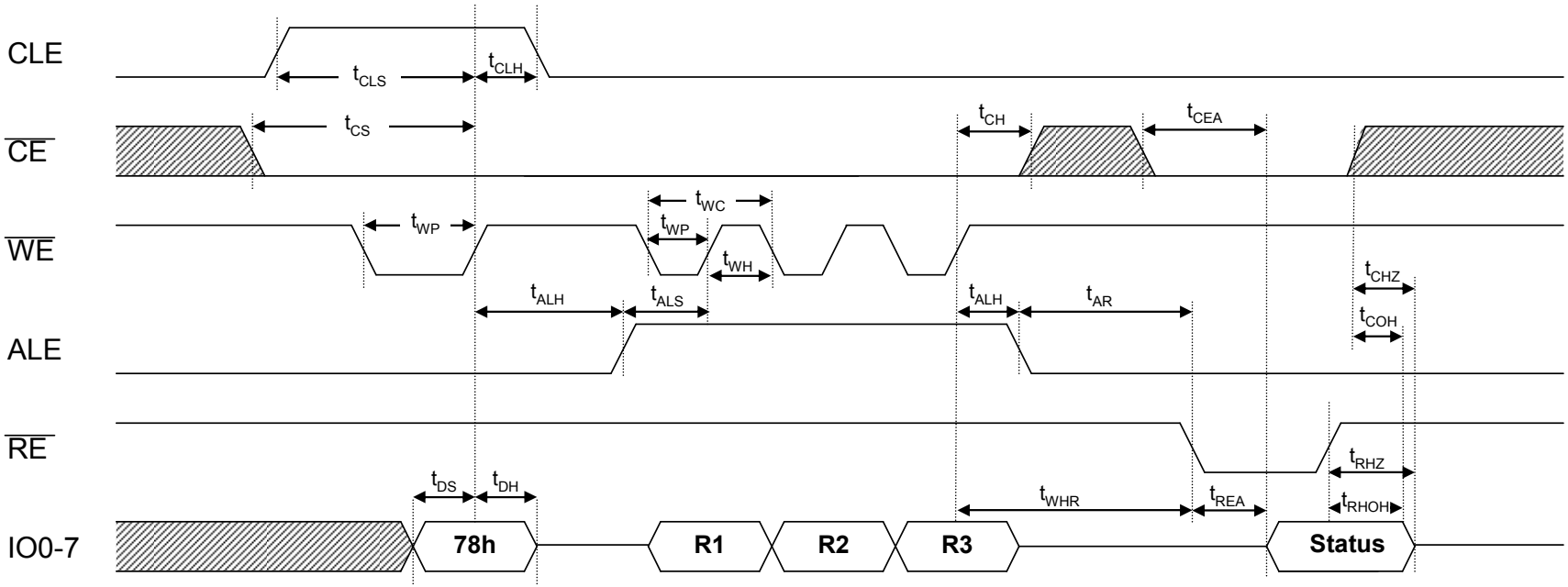

Figure 20 Read Status Enhanced timings

5. Command Definition

5.1. Command Set

Table 14 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 14. Typically, commands that have a second command cycle include an address.

Command	O/M	1st Cycle	2nd Cycle	Acceptable while Accessed LUN is Busy	Acceptable while Other LUNs are Busy	Target level commands
Read	M	00h	30h		Y	
Copyback Read	O	00h	35h		Y	
Change Read Column	M	05h	E0h		Y	
Read Cache Enhanced	O	00h	31h		Y	
Read Cache	O	31h			Y	
Read Cache End	O	3Fh			Y	
Block Erase	M	60h	D0h		Y	
Interleaved	O		D1h		Y	
Read Status	M	70h		Y	Y	
Read Status Enhanced	O	78h		Y	Y	
Page Program	M	80h	10h		Y	
Interleaved	O		11h		Y	
Page Cache Program	O	80h	15h		Y	
Copyback Program	O	85h	10h		Y	
Interleaved	O	85h	11h		Y	
Change Write Column	M	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	M	FFh		Y	Y	Y

Table 14 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.

Type	Opcode
Vendor Specific	02h – 04h, 08h, 16h – 17h, 19h, 1Dh, 20h – 22h, 25h – 29h, 2Bh, 2Dh – 2Fh, 33h, 36h – 3Eh, 40h – 41h, 48h, 4Ch, 53h – 55h, 68h, 72h – 75h, 84h, 87h – 89h, 91h – BFh, CFh, F1-F4h
Future Standardization	06h, 23h – 24h, 2Ah, 2Ch, 32h, 34h, 65h, 71h, 79h – 7Bh, 81h
Reserved	01h, 07h, 09h – 0Bh, 0Dh – 0Fh, 12h-14h, 18h, 1Ah – 1Ch, 1Eh – 1Fh, 42h – 47h, 49h – 4Bh, 4Dh – 52h, 56h – 5Fh, 61h – 64h, 66h – 67h, 69h – 6Fh, 76h – 77h, 7Ch – 7Fh, 82h – 83h, 86h, 8Ah – 8Fh, C0h – CEh, D2h –DFh, E1h – EBh, F0h, F5h – FEh

Table 15 Vendor Specific, Future Standardization, and Reserved Opcodes

5.2. Reset Definition

The Reset function puts the target in its default power-up state. As part of the Reset command, all LUNs are also reset. The command may be executed with the target in any state. Figure 21 defines the Reset behavior and timings.

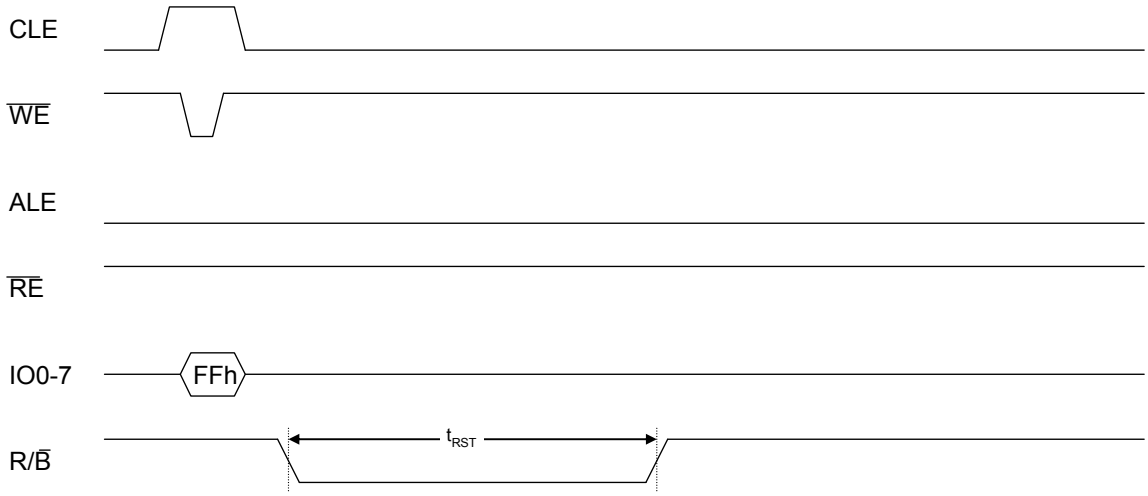

Figure 21 Reset timing diagram

5.3. 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 22 defines the Read ID behavior and timings.

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).

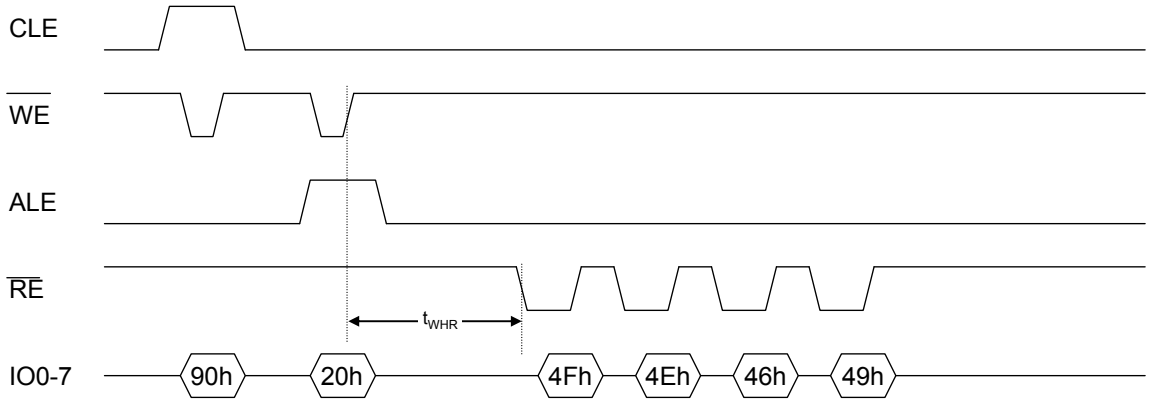

Figure 22 Read ID timing diagram

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 23 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.

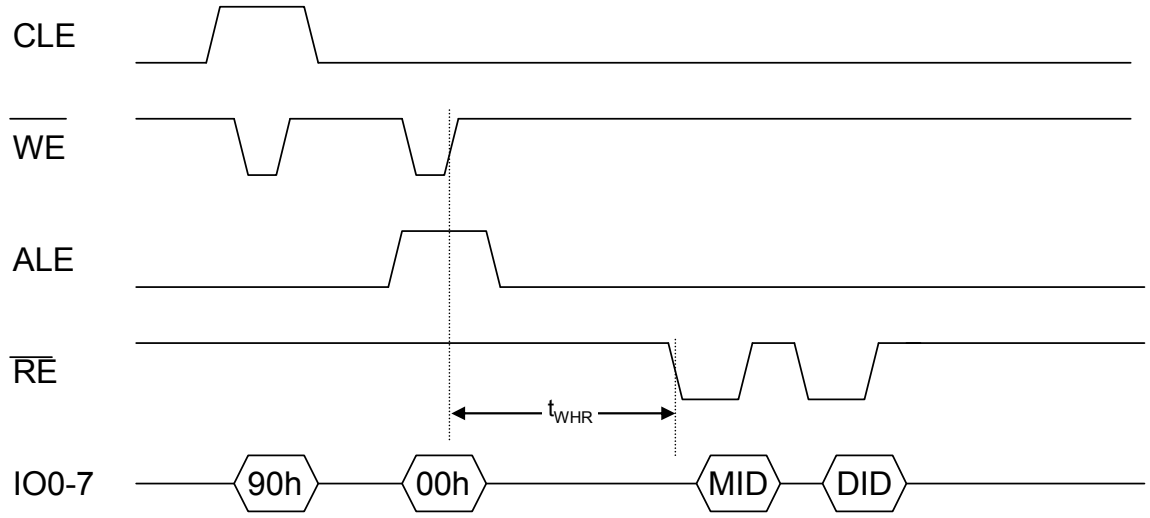

Figure 23 Read ID timing diagram for manufacturer ID

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

5.4. 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. Figure 24 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 can be issued during 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 shall not be used during execution of the Read Parameter Page command.

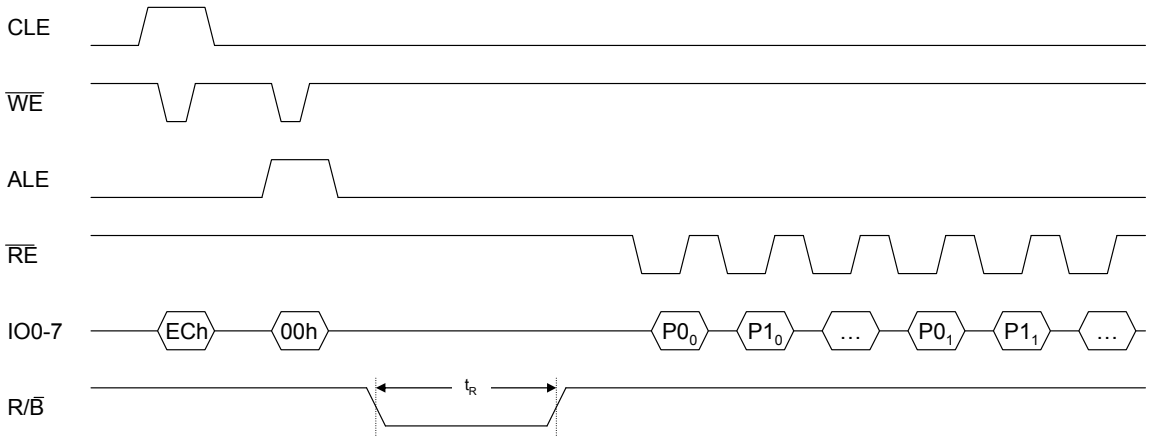

Figure 24 Read Parameter Page command timing

$P0_k$-Pn_k The kth copy of the parameter page data structure. See section 5.4.1. Reading bytes beyond the end of the final parameter page copy returns indeterminate values.

5.4.1. Parameter Page Data Structure Definition

Table 16 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 1.3.2.3 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
	Revision information and features block	
0-3	M	Parameter page signature Byte 0: 4Fh, “O” Byte 1: 4Eh, “N” Byte 2: 46h, “F” Byte 3: 49h, “I”
4-5	M	Revision number 2-15 Reserved (0) 1 1 = supports ONFI version 1.0 0 Reserved (0)
6-7	M	Features supported 5-15 Reserved (0) 4 1 = supports odd to even page Copyback 3 1 = supports interleaved 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 6-15 Reserved (0) 5 1 = supports Read Unique ID 4 1 = supports Copyback 3 1 = supports Read Status Enhanced 2 1 = supports Get Features and Set Features 1 1 = supports Read Cache commands 0 1 = supports Page Cache Program command
10-31		Reserved (0)
	Manufacturer information block	
32-43	M	Device manufacturer (12 ASCII characters)
44-63	M	Device model (20 ASCII characters)
64	M	JEDEC manufacturer ID
65-66	O	Date code
67-79		Reserved (0)
	Memory organization block	
80-83	M	Number of data bytes per page
84-85	M	Number of spare bytes per page
86-89	M	Number of data bytes per partial page
90-91	M	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

Byte	O/M	Description
111	M	Partial programming attributes 5-7 Reserved 4 1 = partial page layout is partial page data followed by partial page spare 1-3 Reserved 0 1 = partial page programming has constraints
112	M	Number of bits ECC correctability
113	M	Number of interleaved address bits 4-7 Reserved (0) 0-3 Number of interleaved address bits
114	O	Interleaved operation attributes 4-7 Reserved (0) 3 Address restrictions for program cache 2 1 = program cache supported 1 1 = no block address restrictions 0 Overlapped / concurrent interleaving support
115-127		Reserved (0)
	Electrical parameters block	
128	M	I/O pin capacitance
129-130	M	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	O	Program cache 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
133-134	M	t_{PROG} Maximum page program time (µs)
135-136	M	t_{BERS} Maximum block erase time (µs)
137-138	M	t_R Maximum page read time (µs)
139-140	M	t_{CCS} Minimum change column setup time (ns)
141-163		Reserved (0)
	Vendor block	
164-165	M	Vendor specific Revision number
166-253		Vendor specific
254-255	M	Integrity CRC

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

Table 16 Parameter page definitions

5.4.1.1. 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.

5.4.1.2. 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.

Bits 2-15 are reserved and shall be cleared to zero.

5.4.1.3. 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.

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# 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 interleaved operations. Refer to section 5.4.1.26.

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.

Bits 5-15 are reserved and shall be cleared to zero.

5.4.1.4. 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.

Bit 1 when set to one indicates that the target supports the Read Cache, Read Cache Enhanced, and Read Cache End commands. If bit 1 is cleared to zero, the host shall not issue the Read Cache, Read Cache Enhanced, or Read Cache End commands to the target.

Bit 2 when set to one indicates that the target supports 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 interleaved 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 interleaved operations are supported and this bit is set to one, then interleaved 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.

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

5.4.1.5. 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 5.4.1.7).

5.4.1.6. 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.

5.4.1.7. Byte 64: JEDEC manufacturer ID

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

5.4.1.8. 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.

5.4.1.9. 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.

5.4.1.10. 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.

5.4.1.11. Byte 86-89: Number of data bytes per partial page

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

5.4.1.12. Byte 90-91: Number of spare bytes per partial page

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

5.4.1.13. 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.

5.4.1.14. 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.

5.4.1.15. 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.

5.4.1.16. 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.

5.4.1.17. 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.

5.4.1.18. 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.

5.4.1.19. 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: value x $10^{multiplier}$. 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 x 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.

5.4.1.20. 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 5.4.1.21) when the host follows the specified number of bits to correct.

5.4.1.21. Byte 108-109: Block endurance for guaranteed valid blocks

This field indicates the maximum number of program/erase cycles per addressable page/block in the guaranteed valid block area (see section 5.4.1.20). This value requires that the host is using at least the minimum ECC correctability reported in the parameter page. This value is not encoded.

5.4.1.22. 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.

If partial page programming is supported, the host shall follow any constraints listed in section 5.4.1.23. Programming the same portion of a page without an erase operation results in indeterminate page contents. Refer to section 3.4.

5.4.1.23. Byte 111: Partial programming attributes

This field indicates the attributes for partial page programming that the target supports. Refer to section 3.4.

General Attributes:

Bit 0 when set to one indicates that there are constraints for partial page programming. When cleared to zero, there are no constraints for partial page programming. If there are constraints, the constraints are specified in the constraints section of this entry. If there are no constraints specified, a partial page may be a single byte (for x8 targets) or a single word (for x16 targets).

Bits 3-1 are reserved.

Constraints:

If bit 0 is cleared to zero, then all bits in this field shall be cleared to zero (since there are no constraints). If bit 0 is set to one, then bit 4 shall be set to one to indicate the type of constraint to the host.

Bit 4 when set to one indicates that the target supports partial page programming where the spare area for each partial page unit is directly following each partial page. Partial pages shall be written on partial page boundaries. The partial page layout for a target that supports four partial pages is shown below.

Data for Partial Page 0	Spare for Partial Page 0	Data for Partial Page 1	Spare for Partial Page 1	Data for Partial Page 2	Spare for Partial Page 2	Data for Partial Page 3	Spare for Partial Page 3

Bits 7-5 are reserved.

5.4.1.24. 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 will not be exceeded. All used bytes in the page shall be protected by ECC including the spare bytes if the minimum ECC requirement has a value greater than zero.

When this value is cleared to zero, the target shall return only valid data and if it cannot it shall fail the command.

5.4.1.25. Byte 113: Interleaved addressing

This field describes parameters for interleaved addressing.

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

Bits 4-7 are reserved.

5.4.1.26. Byte 114: Interleaved operation attributes

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

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

Bit 1 indicates that there are no block address restrictions for the interleaved operation. If set to one all block address bits may be different between interleaved operations. If cleared to zero all block address bits (other than the interleaved address bits) shall be the same. This restriction applies to all interleaved operations (Program, Erase, and Copyback Program).

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

Bit 3 indicates whether interleaved addresses may change during a program cache sequence between 15h commands. If set to one, then the host may change the number and value of interleaved addresses in the cache program sequence. If cleared to zero, then for each program cache operation the interleaved addresses and number of interleaved addresses issued to the LUN shall be the same.

Bits 4-7 are reserved.

5.4.1.27. Byte 128: I/O pin capacitance

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 2.9.

5.4.1.28. Byte 129-130: Timing mode support

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

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

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

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

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

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

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

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

5.4.1.29. Byte 131-132: Program cache timing mode support

This field indicates the timing modes supported for Page Cache Program operations. This value is mandatory if Page Cache Program is implemented. If Page Cache Program is not implemented, this field shall be cleared to zero.

Bit 0 when set to one indicates that the target supports timing mode 0 for program cache operations.

Bit 1 when set to one indicates that the target supports timing mode 1 for program cache operations.

Bit 2 when set to one indicates that the target supports timing mode 2 for program cache operations.

Bit 3 when set to one indicates that the target supports timing mode 3 for program cache operations.

Bit 4 when set to one indicates that the target supports timing mode 4 for program cache operations.

Bit 5 when set to one indicates that the target supports timing mode 5 for program cache operations.

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

5.4.1.30. Byte 133-134: Maximum page program time

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

5.4.1.31. Byte 135-136: Maximum block erase time

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

5.4.1.32. Byte 137-138: Maximum page read time

This field indicates the maximum page read time (tR) in microseconds.

5.4.1.33. 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.

5.4.1.34. 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.

5.4.1.35. Byte 166-253: Vendor specific

This field is reserved for vendor specific use.

5.4.1.36. Byte 254-255: Integrity CRC

The Integrity CRC (Cyclic Redundancy Check) field is used to verify that the contents of the parameters 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 word (16-bit) quantities starting with bytes 1:0 in the parameter page. The bits in the 16-bit quantity are processed from the most significant bit (bit 15) to the least significant bit (bit 0). Even bytes of the parameter page (i.e. 0, 2, 4, etc) shall be used in the lower byte of the CRC calculation (bits 7:0). Odd bytes of the parameter page (i.e. 1, 3, 5, etc) shall be used in the upper byte of the CRC calculation (bits 15:8).

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.

5.4.1.37. 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.

5.4.1.38. 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.

5.4.1.39. 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.

5.5. 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 17. If the XOR of the UID and its bit-wise complement is all ones, then the UID is valid.

Bytes	Value
0-15	UID
16-31	UID complement (bit-wise)

Table 17 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 25 defines the Read Unique ID behavior. The host may use any timing mode supported by the target in order to retrieve the UID data.

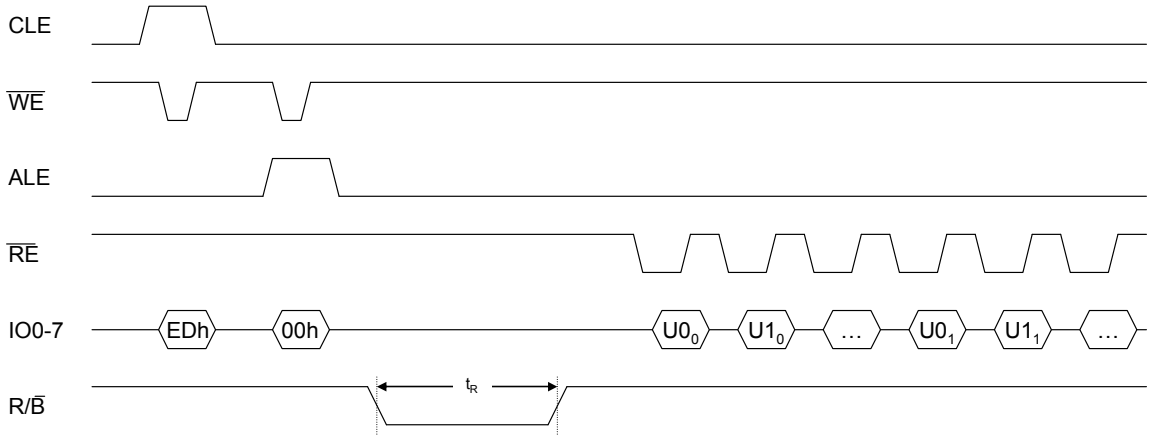

Figure 25 Read Unique ID command timing

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

5.6. 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 26 defines the Block Erase behavior and timings.

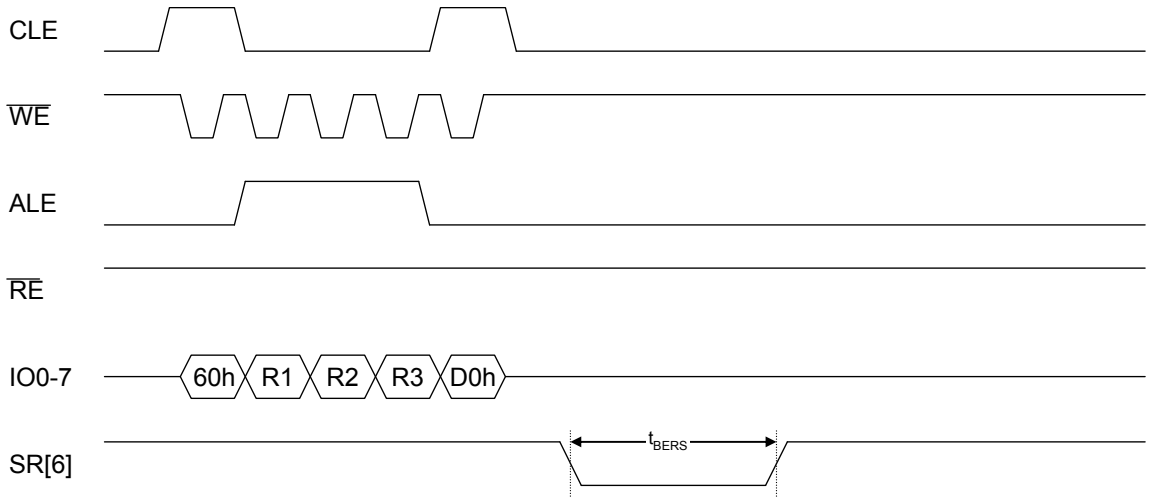

Figure 26 Block Erase timing

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

5.7. Read Status Definition

In the case of non-interleaved operations, the Read Status function retrieves a status value for the last operation issued. If multiple interleaved operations are in progress on a single LUN, then Read Status returns the composite status value for status register bits that are independent per interleaved address. Specifically, Read Status shall return the combined status value of the independent status register bits according to Table 18. See section 5.10 for status register bit definitions.

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

Table 18 Composite Status Value

Figure 27 defines the Read Status behavior and timings.

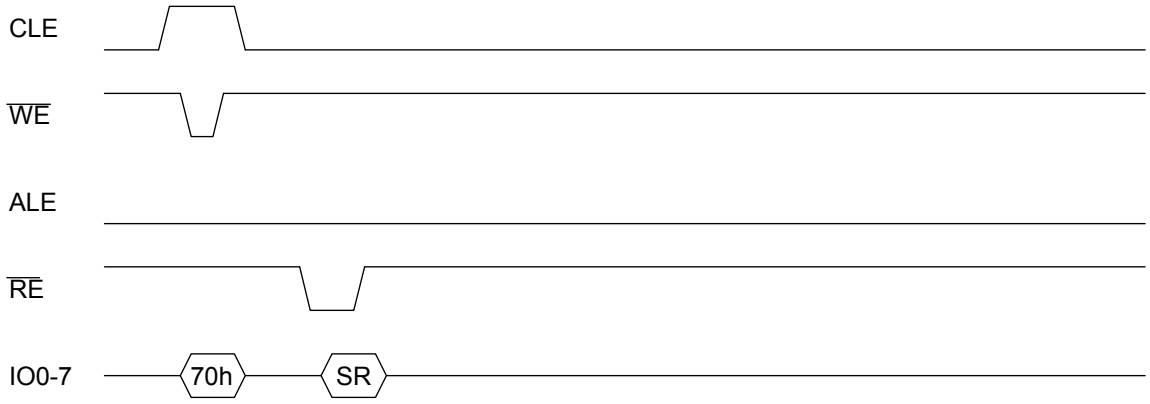

Figure 27 Read Status timing

SR Status value as defined in section 5.10.

5.8. Read Status Enhanced Definition

The Read Status Enhanced function retrieves the status value for a previous operation on the particular LUN and interleaved address specified. Figure 28 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.

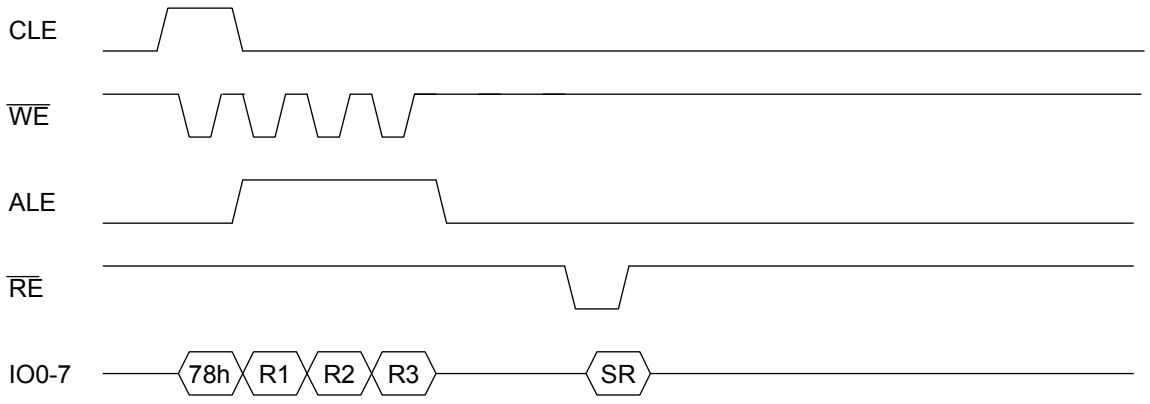

Figure 28 Read Status Enhanced timing

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 interleaved address to retrieve status for.

SR Status value as defined in section 5.10.

5.9. 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# 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# 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# input signal after data output is selected with the 00h command.

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.

5.10. 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#) are invalid and shall be ignored by the host.

Value	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Status Register	WP#	RDY	ARDY	R	R	R	FAILC	FAIL

FAIL If set to one, then the last command failed. If cleared to zero, then the last command was successful. This bit is only valid for 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.

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 interleaved operations or cache commands are not supported, this bit is not used.

RDY If set to one, then the LUN or interleaved 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#, refer to section 2.13.2. 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# 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)

5.11. 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 29 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.

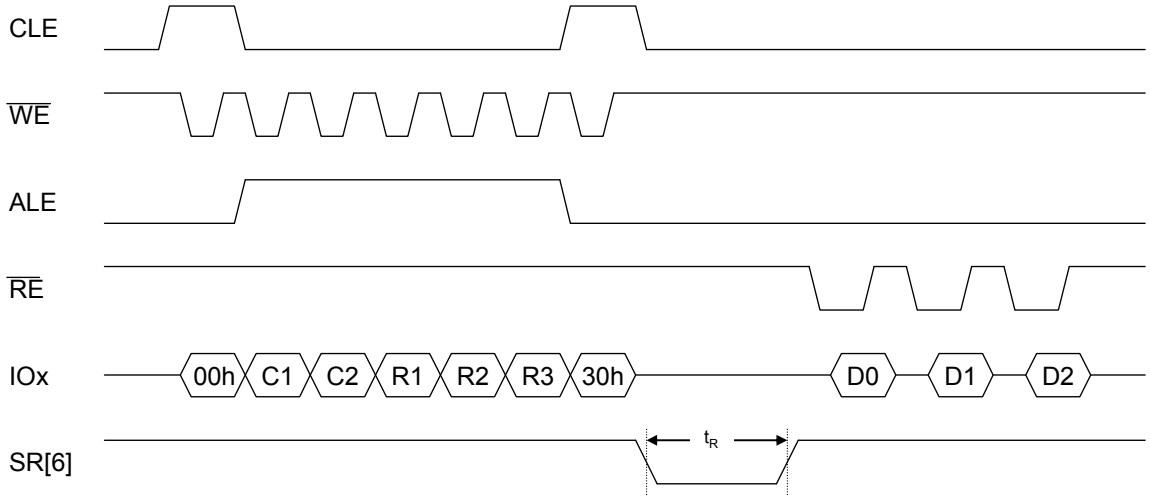

Figure 29 Read timing

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.

5.12. Read Cache Definition

The Read Cache function permits 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 5.11, shall be issued prior to the initial sequential or random Read Cache command in a read cache sequence. A Read Cache or Read Cache Enhanced command shall be issued prior to a Read Cache End (3Fh) command being issued.

The Read Cache 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 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 function. Issuing an additional Read Cache 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 sequential Read Cache (31h) command after the last page of the LUN 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 sequential Read Cache (31h) or Read Cache End (3Fh) command for that LUN.

Figure 30 defines the Read Cache behavior and timings for the beginning of the cache operations subsequent to a Read command being issued to the target. SR[6] conveys whether the next selected page can be read from the page register.

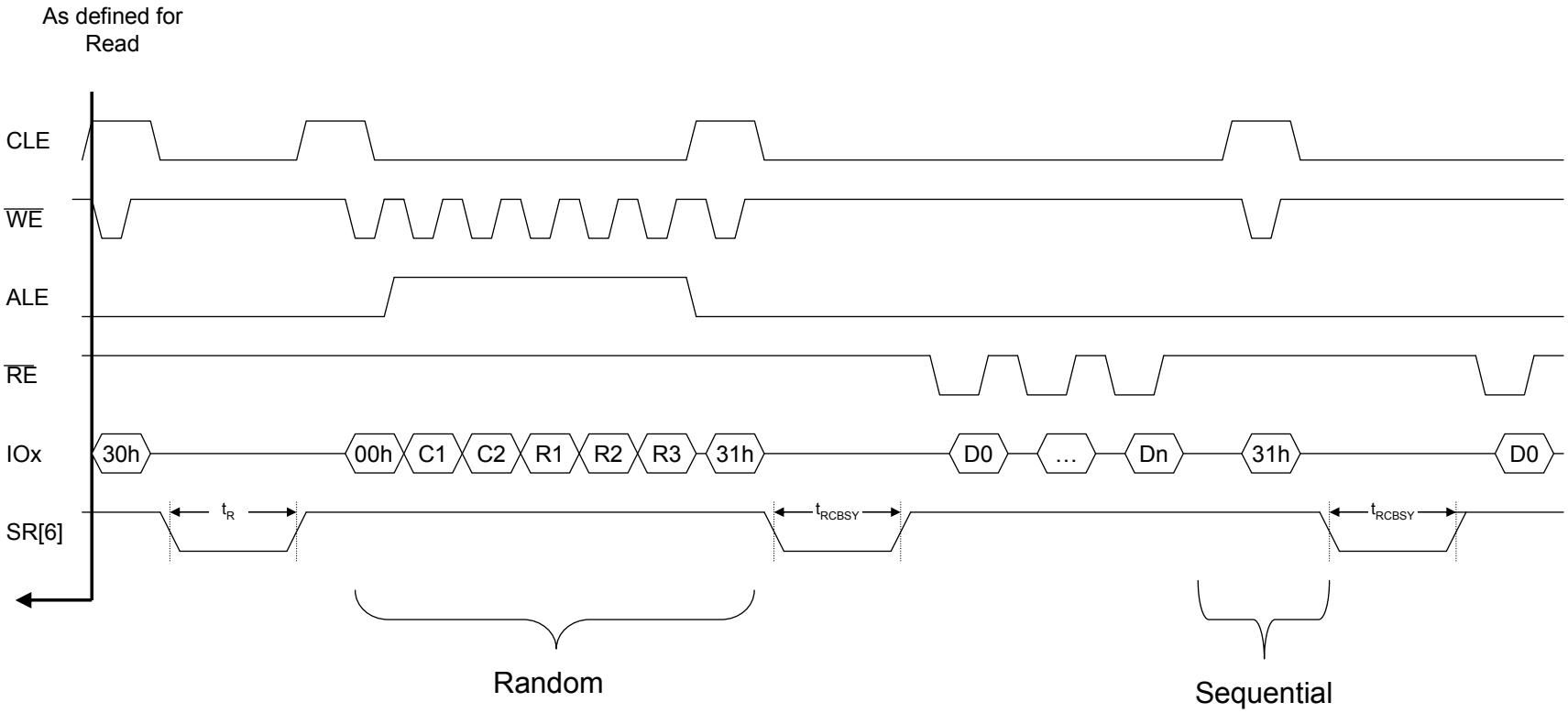

Figure 30 Read Cache timing, start of cache operations

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 31 defines the Read Cache behavior and timings for the end of cache operations. 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.

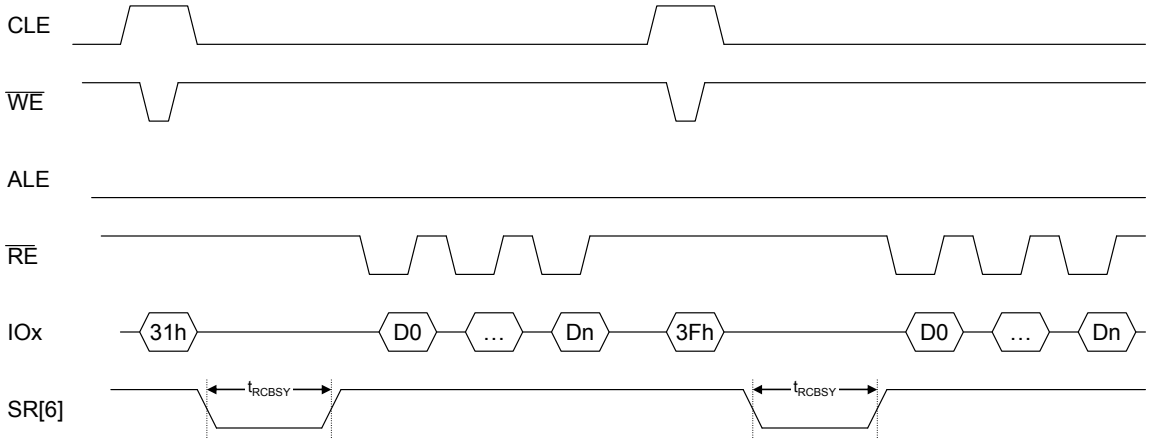

Figure 31 Read Cache timing, end of cache operations

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

5.13. 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 32 defines the Page Program behavior and timings. Writing beyond the end of the page register is undefined.

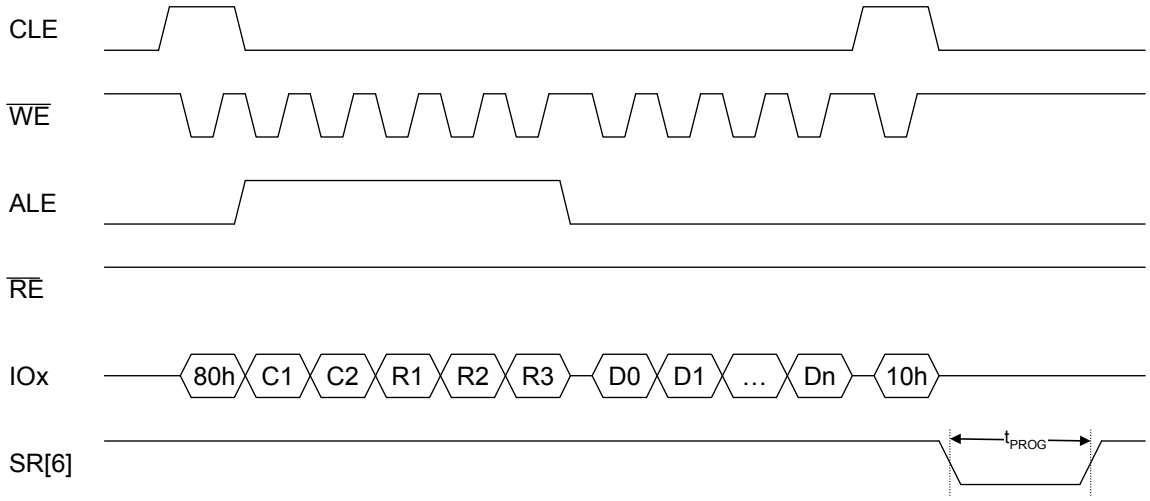

Figure 32 Page Program timing

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.

5.14. Page Cache Program Definition

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 33 and Figure 34 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.

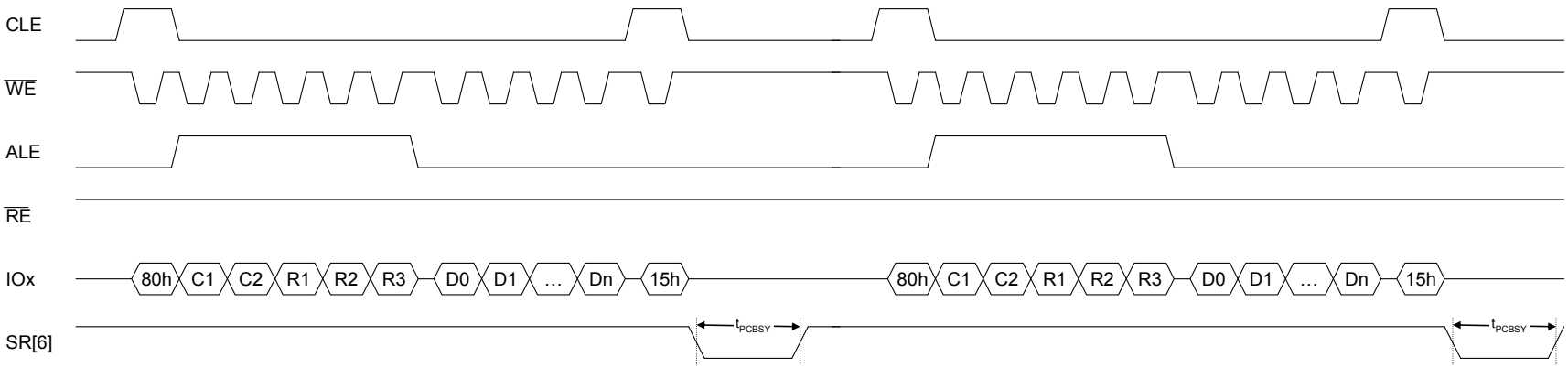

Figure 33 Page Cache Program timing, start of operations

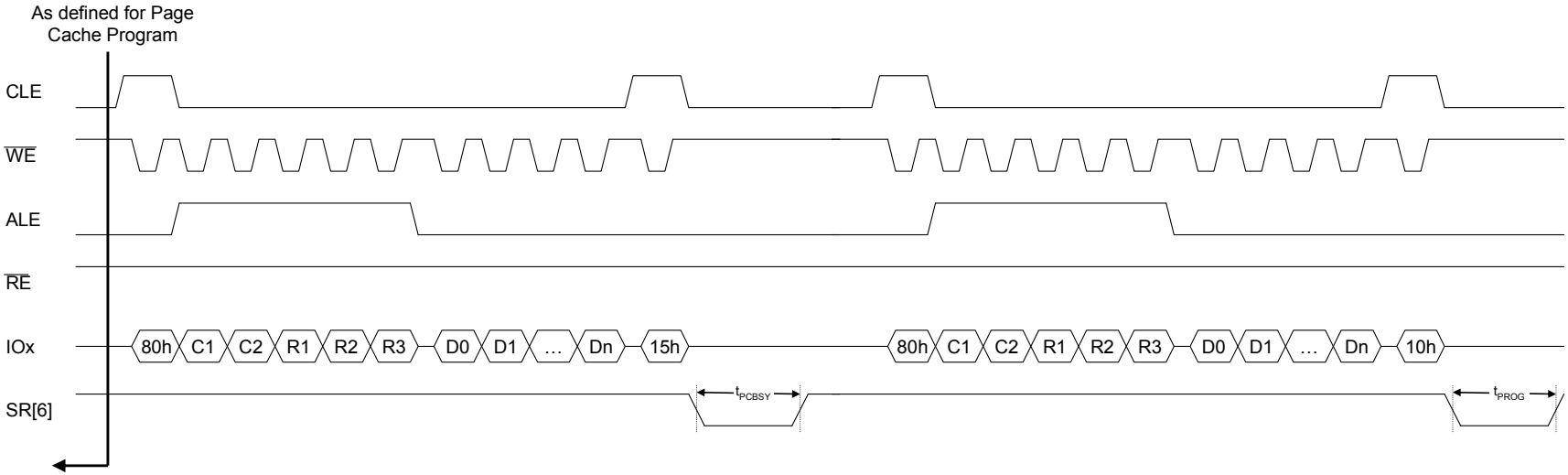

Figure 34 Page Cache Program timing, end of operations

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.

5.15. 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. 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 35 defines the Copyback behavior and timings.

Copyback uses a single page register for the read and program operation. When interleaved addressing is supported, the interleaved address for Copyback Read and Copyback Program for a non-interleaved Copyback operation shall be the same.

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 5.4.1.3.

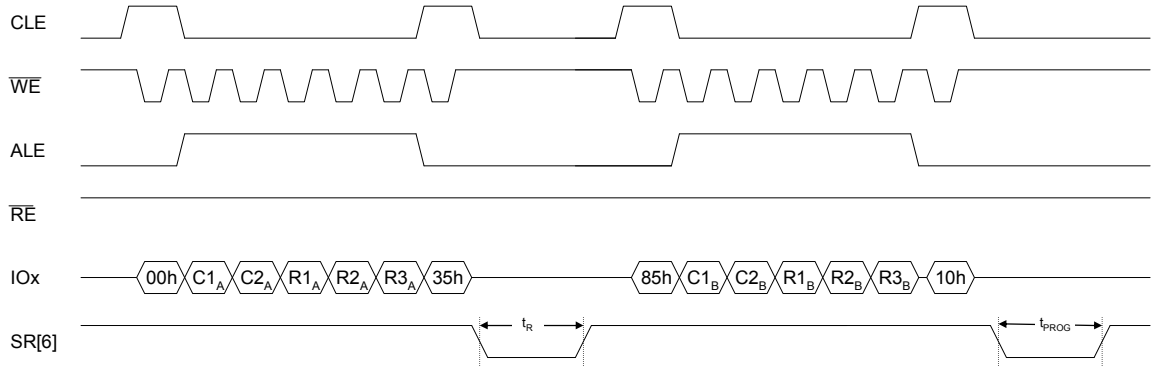

Figure 35 Copyback timing

$C1\text{-}C2_A$ Column address of the page to retrieve. $C1_A$ is the least significant byte.

$R1\text{-}R3_A$ Row address of the page to retrieve. $R1_A$ is the least significant byte.

$C1\text{-}C2_B$ Column address of the page to program. $C1_B$ is the least significant byte.

$R1\text{-}R3_B$ Row address of the page to program. $R1_B$ is the least significant byte.

Figure 36 and Figure 37 define Copyback support for data output and data modification.

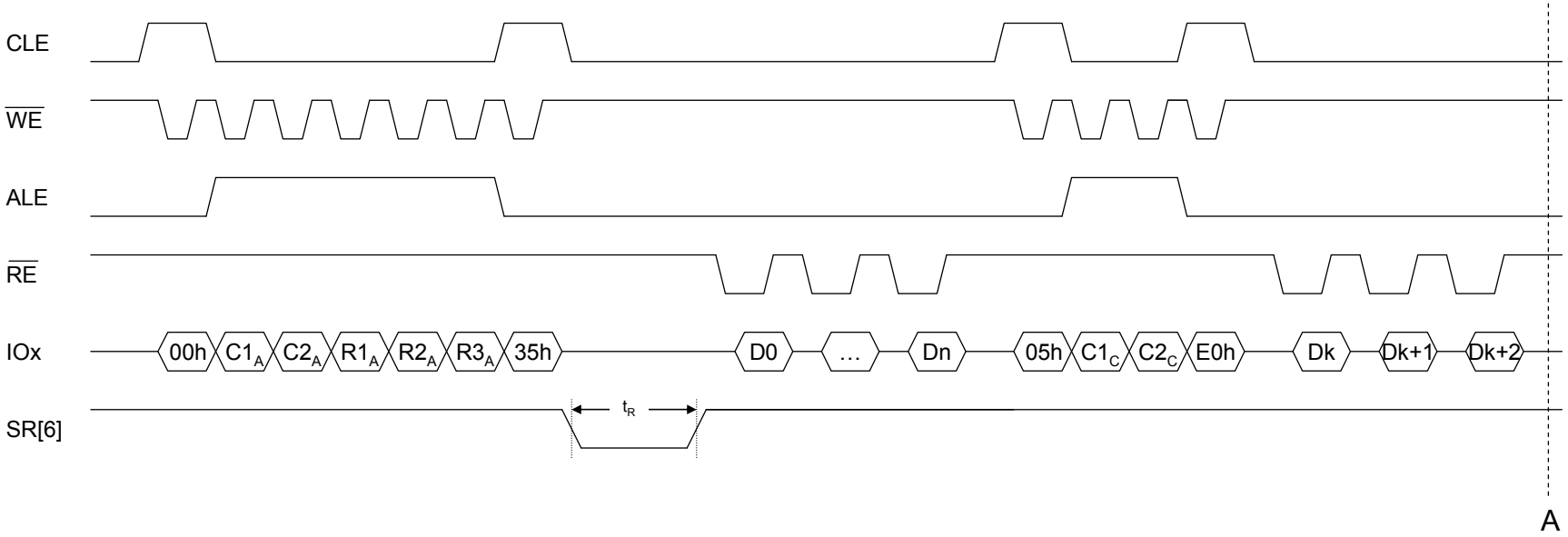

Figure 36 Copyback with data output

$C1\text{-}C2_A$	Column address of the page to retrieve. $C1_A$ is the least significant byte.
$R1\text{-}R3_A$	Row address of the page to retrieve. $R1_A$ is the least significant byte.
D0-Dn	Data bytes read starting at column address specified in $C1_A$.
$C1\text{-}C2_C$	Column address of new location (k) to read out from the page register. $C1_C$ is the least significant byte.
Dk-Dk+n	Data bytes read starting at column address specified in $C1_C$.

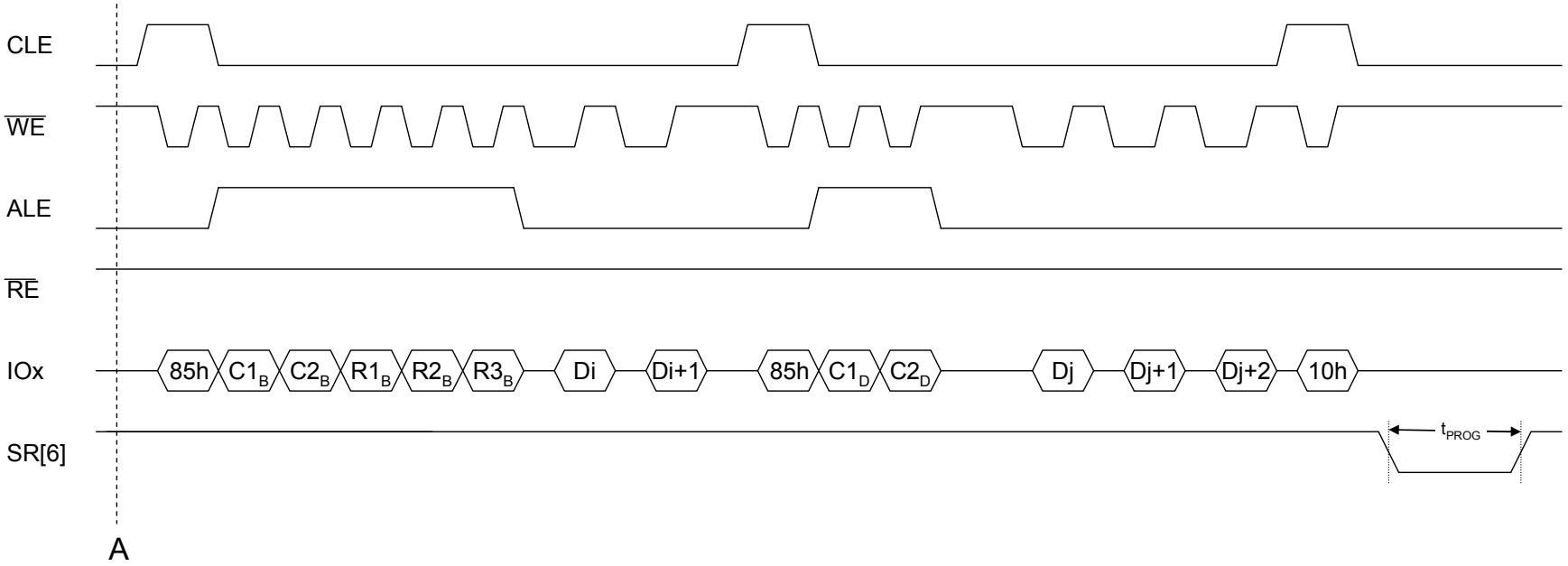

Figure 37 Copyback with data modification

$C1\text{-}C2_B$	Column address of the page to program. $C1_B$ is the least significant byte.
$R1\text{-}R3_B$	Row address of the page to program. $R1_B$ is the least significant byte.
Di-Di+n	Data bytes overwritten in page register starting at column address specified in $C1_B$.
$C1\text{-}C2_D$	Column address of new location (j) to overwrite data at in the page register. $C1_D$ is the least significant byte.
Dj-Dj+n	Data bytes overwritten starting at column address specified in $C1_D$.

5.16. 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 38 defines the Change Read Column behavior and timings.

The host shall not read data from the LUN until tCCS ns after the E0h command is written to the LUN. Refer to Figure 38.

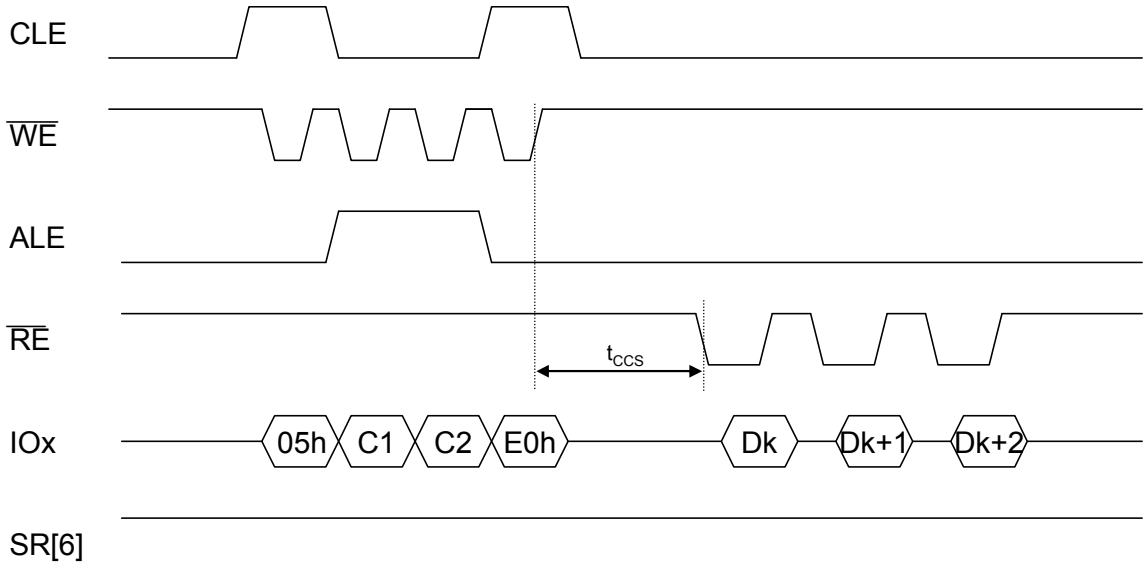

Figure 38 Change Read Column timing

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

5.17. 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 39 defines the Change Write Column behavior and timings.

The host shall not write data to the LUN until tCCS ns after the last column address is written to the LUN. Refer to Figure 38.

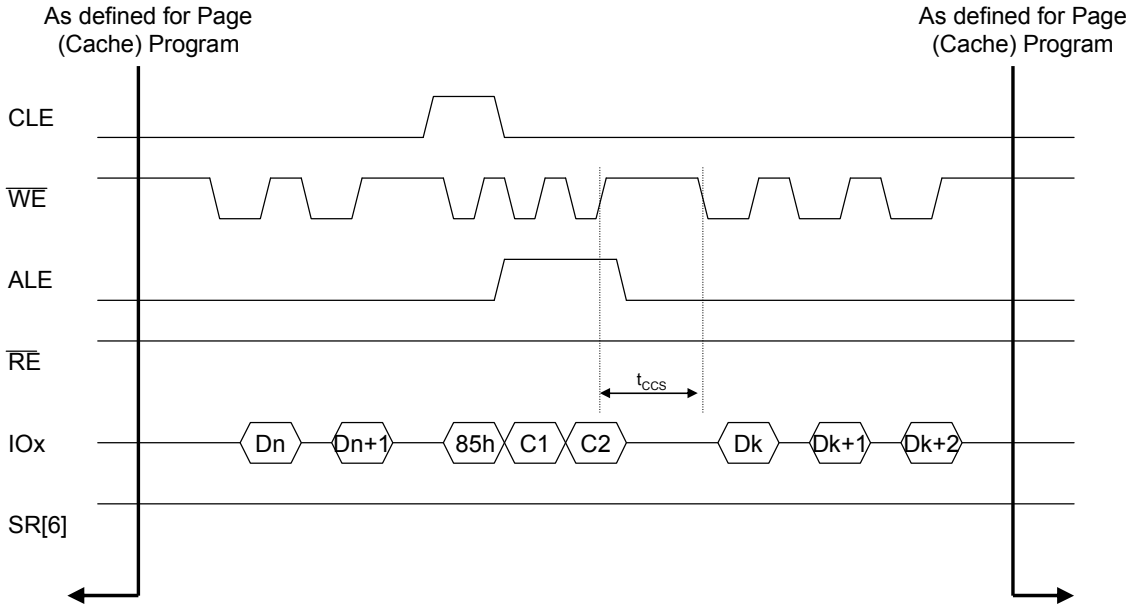

Figure 39 Change Write Column timing

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

5.18. 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 40 defines the Set Features behavior and timings.

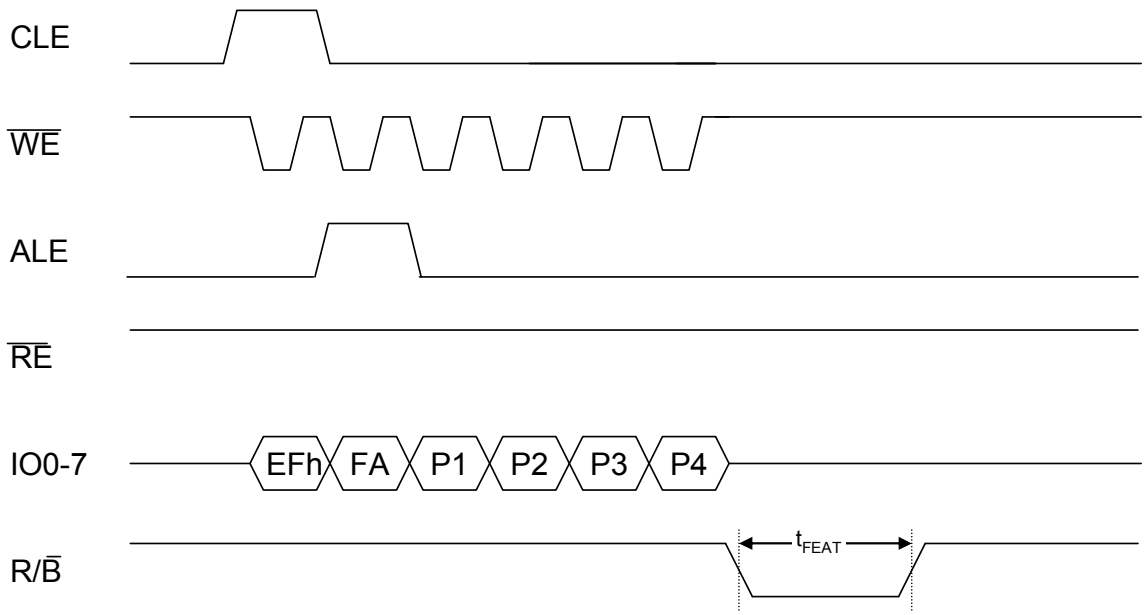

Figure 40 Set Features timing

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 5.20 for the definition of features and sub feature parameters.

5.19. 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 41 defines the Get Features behavior and timings.

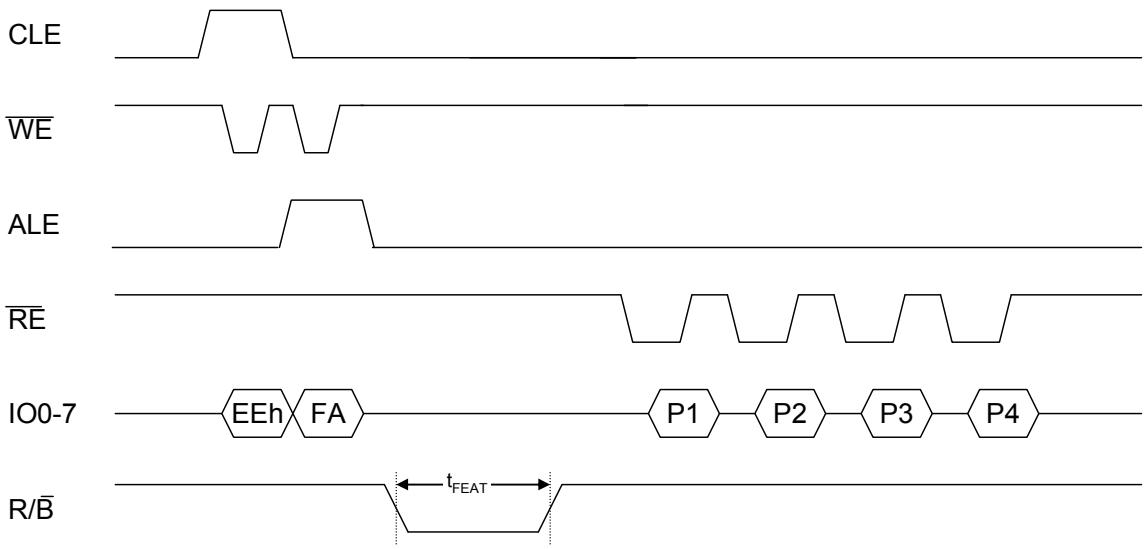

Figure 41 Get Features timing

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 5.20 for the definition of features and sub feature parameters.

5.20. 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-7Fh	Reserved
80h-FFh	Vendor specific

5.20.1. Timing Mode

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

The settings for the timing mode are retained across Reset commands. Hosts shall only set a timing mode that is explicitly shown as supported in the Read Parameter Page.

Sub Feature Parameter	7	6	5	4	3	2	1	0
P1	Reserved (0)				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.

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

6. Interleaved Operations

A LUN may support interleaved program and erase operations. Interleaved operations are when multiple commands of the same type are issued to different blocks on the same LUN. Refer to section 3.1.1 for addressing restrictions with interleaved operations. There are two methods for interleaved operations: concurrent and overlapped.

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

- Page Program
- Copyback Program
- Block Erase

There are no interleaved read operations. The Read Cache command should be used to enhance read performance.

6.1. Requirements

When supported, the interleaved address comprises the lowest order bits of the block address as shown in Figure 11. The LUN and page addresses are required to be the same. The block address (other than the interleaved address bits) may be required to be the same, refer to section 5.4.1.26.

For copyback program operations, the restrictions are the same as for an interleaved program operation. However, copyback reads shall be previously issues to the same interleaved addresses as those in the interleaved copyback program operations. Note that for copyback reads, the reads may have different page addresses since the copyback reads are not interleaved.

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

The interleaved address component of each address shall be distinct. A single interleaved (cached) program operation is shown in Figure 42. Between “Interleave Op 1” and “Interleave Op n”, all interleaved addresses shall be different from each other. After the 10h or 15h (cached) command cycle is issued, previously issued interleaved addresses can be used in future interleaved operations.

Figure 42 Interleaved Program (Cache)

For interleaved erase operations, the interleaved address component of each address shall be distinct. A single interleaved erase operation is shown in Figure 43. Between “Interleave Op 1” and “Interleave Op n”, all interleaved addresses shall be different from each other. After the D0h

command cycle is issued, previously issued interleaved addresses can be used in future interleaved operations.

Figure 43 Interleaved Erase

6.2. Status Register Behavior

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

For interleaved operations, the FAIL bits are independent per interleaved address. Table 19 lists whether a bit is independent per interleaved address or shared across the entire LUN for interleaved operations.

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

Table 19 Independent Status Register bits

6.3. Interleaved 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 an interleaved 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 44 defines the behavior and timings for two interleaved page program commands.

Cache operations may be used when doing interleaved page program operations, as shown, if supported by the target as indicated in the parameter page. Refer to section 5.4.1.25.

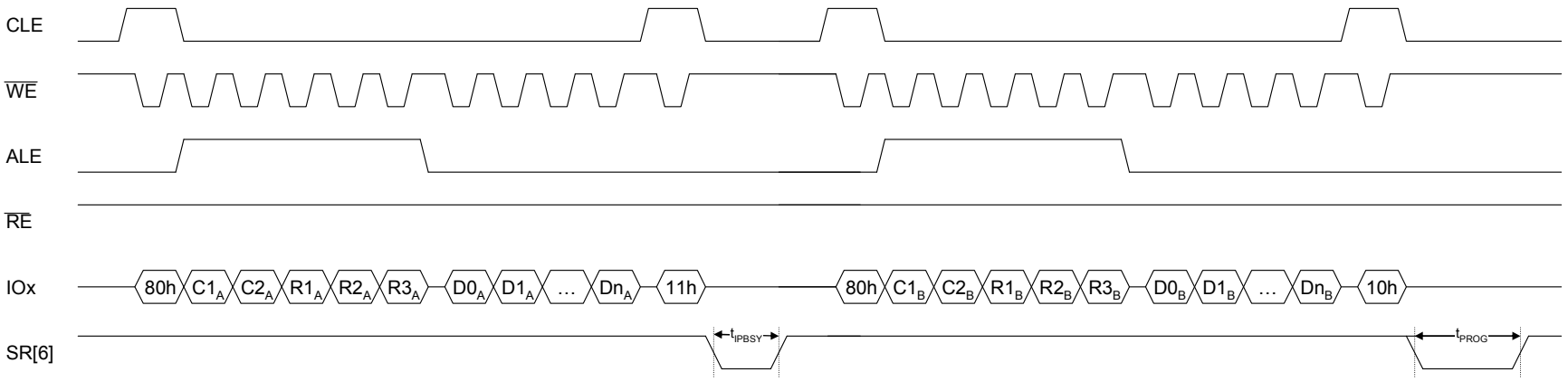

Figure 44 Interleaved Page Program timing

$C1_A$-$C2_A$	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.
$D0_A$-Dn_A	Data to program for page A.
$C1_B$-$C2_B$	Column address for page B. $C2_B$ is the least significant byte.
$R1_B$-$R3_B$	Row address for page B. $R1_B$ is the least significant byte.
$D0_B$-Dn_B	Data to program for page B.

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

Finishing an interleaved 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 interleaved program operation if program cache is supported with interleaved program operations, as described in section 5.4.1.25.

6.4. Interleaved Copyback Program

The Copyback function reads a page of data from one location and then moves that data to a second location. With an interleaved 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 45 and Figure 46 define the behavior and timings for two Copyback Program operations, including the preceding Copyback Read operations.

The interleaved addresses used for the Copyback Read operations shall be the same as the interleaved addresses used in the subsequent interleaved Copyback Program operations.

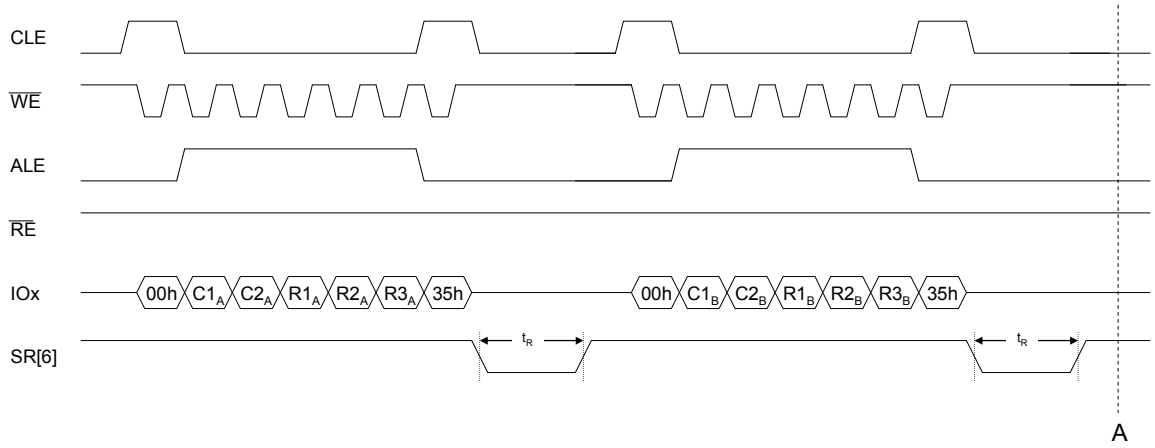

Figure 45 Copyback Read timing for Interleaved Copyback Program

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

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

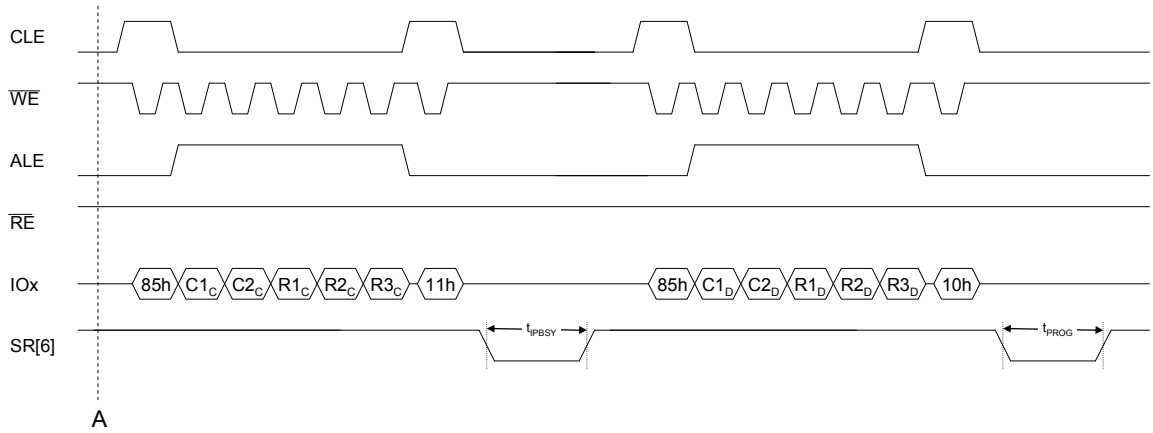

Figure 46 Interleaved Copyback Program

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

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

6.5. Interleaved Block Erase

Figure 47 defines the behavior and timings for an interleaved 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 interleaved operations the LUN supports.

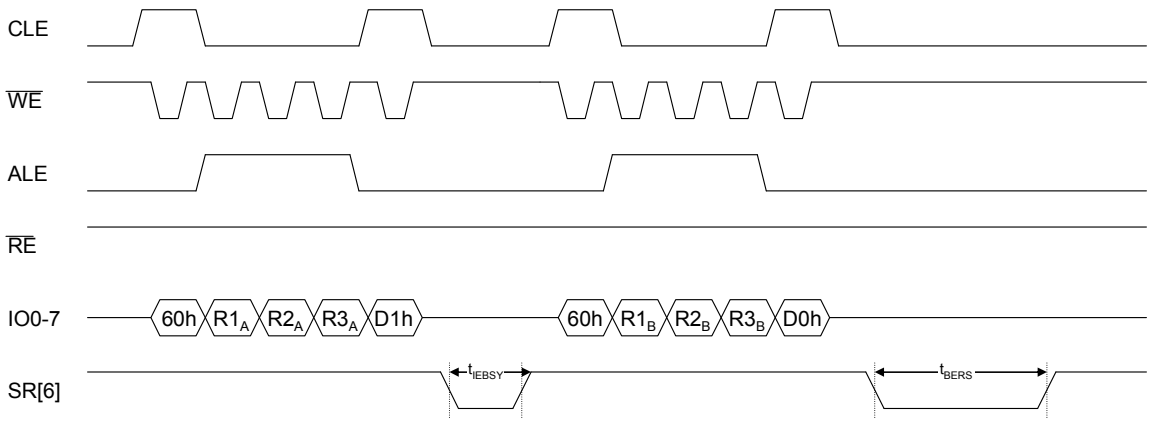

Figure 47 Interleaved Block Erase timing

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

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

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.

tbStatusOut	This variable is set to TRUE when toggling of RE# should return the status value. The power-on value for this variable is FALSE.
tbChgCol	This variable is set to TRUE when changing the column is allowed. The power-on value for this variable is FALSE.
tCopyback	This variable is set to TRUE if the Target is issuing a copyback command. The power-on value for this variable is FALSE.
tLunSelected	This variable contains the LUN that is currently selected by the host. The power-on value for this variable is 0.
tLastCmd	This variable contains the first cycle of the last command (other than 70h/78h) received by the Target.
tReturnState	This variable contains the state to return to after status operations.
tbStatus78hReq	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[1]	The target performs the following actions: 1. R/B# is cleared to zero. 2. Each LUN shall draw less than 10 mA of power per staggered power-up requirement.	
1. Target is ready to accept Reset command[2]	→	T_PowerOnReady
NOTE: 1. This state is entered asynchronously as a result of a power-on event when Vcc reaches Vcc_min. 2. This arc shall be taken within 1 millisecond of Vcc reaching Vcc_min.		

T_PowerOnReady	The target performs the following actions: 1. R/B# is set to one. 2. Each LUN shall draw less than 10mA of power per staggered power-up requirement.	
1. Command cycle FFh (Reset) received	→	T_RST_PowerOn

T_Idle	tCopyback set to FALSE. tReturnState set to T_Idle.		
1.	WP# signal transitioned	→	T_Idle_WP_Transition
2.	LUN indicates its SR[6] value transitioned	→	T_Idle_RB_Transition
3.	Command cycle received	→	T_Cmd_Decode

T_Cmd_Decode[1]	Decode command received. If R/B# 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# is low	→	T_Idle
2.	Command FFh (Reset) decoded	→	T_RST_Execute
3.	Command 90h (Read ID) decoded	→	T_RID_Execute
4.	Command ECh (Read Parameter Page) decoded	→	T_RPP_Execute
5.	Command EDh (Read Unique ID) decoded	→	T_RU_Execute
6.	Command 80h (Page Program) decoded and WP# is high	→	T_PP_Execute
7.	Command 60h (Block Erase) decoded and WP# is high	→	T_BE_Execute
8.	Command 00h (Read) decoded	→	T_RD_Execute
9.	Command EFh (Set Features) decoded	→	T_SF_Execute
10.	Command EEh (Get Features) decoded	→	T_GF_Execute
11.	Command 70h (Read Status) decoded	→	T_RS_Execute
12.	Command 78h (Read Status Enhanced) decoded	→	T_RSE_Execute
NOTE: 1. The host shall ensure R/B# 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# value to all LUN state machines.		
1.	State entered from T_Idle_Rd	→	T_Idle_Rd
2.	Else	→	T_Idle

T_Idle_RB_Transition	R/B# is set to the AND of all LUN status register SR[6] values.[1]		
1.	State entered from T_Idle_Rd	→	T_Idle_Rd
2.	Else	→	T_Idle
NOTE: 1. R/B# 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# signal transitioned	→	T_Idle_WP_Transition
2. LUN indicates its SR[6] value transitioned	→	T_Idle_RB_Transition
3. Read request received and tbStatusOut set to TRUE	→	T_Idle_Rd_Status
4. Read request received and (tLastCmd set to 90h or EEh)	→	T_Idle_Rd_XferByte
5. Read request received and (tLastCmd set to ECh or EDh)	→	T_Idle_Rd_LunByte
6. Read request received and tbStatus78hReq set to FALSE[1]	→	T_Idle_Rd_LunData
7. Command cycle 05h (Change Read Column) received and tbChgCol set to TRUE	→	T_CR_Execute
8. Command cycle of 31h received and tbStatus78hReq set to FALSE	→	T_Idle_Rd_CacheCmd
9. Command cycle of 3Fh received and tLastCmd set to 31h and tbStatus78hReq set to FALSE	→	T_Idle_Rd_CacheCmd
10. Command cycle received	→	T_Cmd_Decode
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.		

T_Idle_Rd_CacheCmd	Set tLastCmd to the command received. Pass command received to LUN tLunSelected	
1. Unconditional	→	T_Idle_Rd

T_Idle_Rd_XferByte	Return next byte of data.	
1. Unconditional	→	T_Idle_Rd

T_Idle_Rd_LunByte	Request byte of data from page register of LUN tLunSelected.	
1. Byte received from LUN tLunSelected	→	T_Idle_Rd_XferHost

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	→	T_Idle_Rd_XferHost

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_RD_Status_Off	→	T_Idle_Rd
	2. Else	→	tReturnState

7.1.4. Reset command states

T_RST_PowerOn	The target performs the following actions: 1. tLastCmd set to FFh. 2. tbStatusOut is set to FALSE. 3. The target sends a Reset request to each LUN.		
	1. Unconditional	→	T_RST_PowerOn_Exec

T_RST_PowerOn_Exec	The target performs the following actions: 1. Target level reset actions are performed. 2. R/B# is set to zero.		
1. Target and LUN reset actions are complete		→	T_RST_End

T_RST_Execute[1]	The target performs the following actions: 1. tLastCmd set to FFh. 2. tbStatusOut is set to FALSE. 3. The target sends a Reset request to each LUN. 4. Set tbChgCol to FALSE. 5. Request all LUNs invalidate page register(s).		
1. Unconditional		→	T_RST_Perform
NOTE: 1. This state is entered asynchronously as a result of receiving a Reset (FFh) command, except if this is the first Reset after power-on.			

T_RST_Perform	The target performs the following actions: 1. Target level reset actions are performed. 2. R/B# 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# 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. 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		→	T_RID_ManufacturerID
2. Command cycle received		→	T_Cmd_Decode

T_RID_ManufacturerID	Return the JEDEC manufacturer ID.	
1. Read byte request received	→	T_RID_DeviceID
2. Command cycle received	→	T_Cmd_Decode

T_RID_DeviceID	Return the device ID.[1]	
1. Unconditional	→	T_Idle_Rd
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	→	T_RID_Signature
2. Command cycle received	→	T_Cmd_Decode

T_RID_Signature	Return next ONFI signature byte.[1]	
1. Last ONFI signature byte returned	→	T_Idle_Rd
2. Else	→	T_RID_Addr_20h
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: 1. tLastCmd set to ECh. 2. Set tbStatusOut to FALSE. 3. Set tbChgCol to TRUE. 4. Wait for an address cycle. 5. Request all LUNs invalidate page register(s). 6. Target selects LUN to execute parameter page read, sets tLunSelected to the address of this LUN.	
1. Address cycle of 00h received	→	T_RPP_ReadParams

T_RPP_ReadParams	The target performs the following actions: 1. Request LUN tLunSelected clear SR[6] to zero. 2. R/B# is cleared to zero. 3. Request LUN tLunSelected make parameter page data available in page register. 4. tReturnState set to T_RPP_ReadParams.	
1. Read of page complete	→	T_RPP_Complete
2. Command cycle 70h (Read Status) received	→	T_RS_Execute
3. Read request received and tbStatusOut set to TRUE	→	T_Idle_Rd_Status

T_RPP_Complete	Request LUN tLunSelected set SR[6] to one. R/B# is set to one.		
	1. Unconditional	→	T_Idle_Rd

7.1.7. Read Unique ID command states

T_RU_Execute	The target performs the following actions: 1. tLastCmd set to EDh. 2. Set tbStatusOut to FALSE. 3. Set tbChgCol to TRUE. 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	→	T_RU_ReadUid

T_RU_ReadUid	The target performs the following actions: 1. Request LUN tLunSelected clear SR[6] to zero. 2. R/B# 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	→	T_RU_Complete
	2. Command cycle 70h (Read Status) received	→	T_RS_Execute
	3. Read request received and tbStatusOut set to TRUE	→	T_Idle_Rd_Status

T_RU_Complete	Request LUN tLunSelected set SR[6] to one. R/B# is set to one.		
	1. Unconditional	→	T_Idle_Rd

7.1.8. Page Program and Page Cache Program command states

T_PP_Execute	The target performs the following actions: 1. tLastCmd set to 80h. 2. If R/B# is cleared to zero, then tbStatus78hReq is set to TRUE. 3. Request all LUNs clear their page register(s).		
	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	tLunSelected is set to the LUN indicated by the row address received. Target issues the Program with associated address to the LUN tLunSelected.		
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_IlvWait
2. Command passed was 10h or 15h		→	T_Idle

T_PP_IlvWait	Wait for next Program to be issued. tReturnState set to T_PP_IlvWait.		
1. Command cycle of 85h received[1] and tCopyback set to TRUE		→	T_PP_AddrWait
2. Command cycle of 80h received[1] and tCopyback set to FALSE		→	T_PP_AddrWait
3. Command cycle of 70h received		→	T_RS_Execute
4. Command cycle of 78h received		→	T_RSE_Execute
5. Read request received and tbStatusOut set to TRUE		→	T_Idle_Rd_Status
NOTE: 1. Address cycles for the Program operation being issued shall have the same LUN address and page address as the preceding Program operation. The interleaved address shall be different than the one issued in the preceding Program operation.			

T_PP_ColChg	Wait for column address cycle.		
1. Address cycle received		→	T_PP_ColChg_Addr

T_PP_ColChg_Addr	Store the address cycle received.		
	1. More column address cycles required	→	T_PP_ColChg
	2. All address cycles received	→	T_PP_ColChg_LUN

T_PP_ColChg_LUN	Request that LUN tLunSelected change column address to column address received.		
	1. Unconditional	→	T_PP_LUN_DataWait

7.1.9. Block Erase command states

T_BE_Execute	The target performs the following actions: 1. tLastCmd set to 60h. 2. tbStatusOut set to FALSE. 3. If R/B# is cleared to zero, then tbStatus78hReq is set to TRUE. 4. Wait for a row address cycle.		
	1. Address cycle received	→	T_BE_Addr

T_BE_Addr	Store the row address cycle received.		
	1. More address cycles required	→	T_BE_Execute
	2. All address cycles received	→	T_BE_LUN_Execute

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	→	T_BE_LUN_Confirm

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_IlvWait
	2. Command passed was D0h	→	T_Idle

T_BE_IlvWait	Wait for next Erase to be issued. tReturnState set to T_BE_IlvWait.		
	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	→	T_RD_StatusOff
2. Else	→	T_RD_AddrWait

T_RD_StatusOff	tbStatusOut set to FALSE. tReturnState set to T_RD_StatusOff.	
1. Address cycle received	→	T_RD_Addr
2. Read request received and tLastCmd set to EEh	→	T_Idle_Rd_XferHost
3. Read request received	→	T_Idle_Rd_LunData
4. Command cycle of 05h received	→	T_CR_Execute

T_RD_AddrWait	tLastCmd set to 00h. If R/B# is cleared to zero, then tbStatus78hReq is set to TRUE. Wait for an address cycle.	
1. Address cycle received	→	T_RD_Addr

T_RD_Addr	Store the address cycle received.	
1. More address cycles required	→	T_RD_AddrWait
2. All address cycles received	→	T_RD_LUN_Execute

T_RD_LUN_Execute	The target performs the following actions: 1. tLunSelected is set to the LUN indicated by the row address received. 2. Issues the Read Page with address to LUN tLunSelected. 3. Requests all idle LUNs not selected to turn off their output buffers.[1]	
1. Unconditional	→	T_RD_LUN_Confirm
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, or 35h to be received.	
1. Command cycle of 30h, 31h, or 35h received	→	T_RD_Cmd_Pass

T_RD_Cmd_Pass	Pass command received to LUN tLunSelected	
1. Command passed was 35h	→	T_RD_Copyback
2. Command passed was 30h or 31h	→	T_Idle_Rd

T_RD_Copyback	tCopyback set to TRUE. tReturnState set to T_RD_Copyback.		
	1. Command cycle of 00h received	→	T_RD_Execute
	2. Command cycle of 05h received	→	T_CR_Execute
	3. Command cycle of 85h received	→	T_PP_AddrWait
	4. Command cycle of 70h received	→	T_RS_Execute
	5. Command cycle of 78h received	→	T_RSE_Execute
	6. LUN indicates its SR[6] value transitions	→	T_Idle_RB_Transition
	7. Read request received and tbStatusOut set to TRUE	→	T_Idle_Rd_Status
	8. Read request received	→	T_Idle_Rd_LunData

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# 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# 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 tbStatusOut 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# 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# is set to one.		
1. Unconditional		→	T_GF_DataWait

T_GF_DataWait	Wait for the read request.		
1. Read byte request received		→	T_GF_DataXfer
2. Command cycle received		→	T_Cmd_Decode

T_GF_DataXfer	Return next data byte.[1]		
1. Last data byte returned		→	T_Idle
2. Else		→	T_GF_DataWait
NOTE: 1. Reading beyond the fourth byte returns indeterminate values.			

7.1.13. Read Status command states

T_RS_Execute		
1. tbStatus78hReq is set to FALSE[1]	→	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[1]	tbStatus78hReq is set to FALSE. tbStatusOut is set to TRUE. Wait for a row address cycle.	
1. Row address cycle received	→	T_RSE_Addr
NOTE: 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.		

T_RSE_Addr	Store the row address cycle received.	
1. More row address cycles required	→	T_RSE_Execute
2. All row address cycles received	→	T_RSE_Select

T_RSE_Select	The target performs the following actions: 1. Set tLunSelected to LUN selected by row address received. 2. Indicate 78h command and row address received to all LUNs.	
1. tReturnState set to T_Idle	→	T_Idle_Rd
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.

lunStatus	This variable contains the current LUN status register value contents. The power on value for this variable is 00h.
lunFail[]	This array contains the FAIL and FAILC bits for each interleave address. For example, lunFail[3][1] contains the FAILC bit for interleaved address 3. The power on value for each variable in this array is 00b.
lunLastConfirm	This variable contains the last confirm command cycle (30h, 31h, 35h, 10h, 15h, 11h, D0h, D1h). The power on value for this variable is FFh.
lunReturnState	This variable contains the state to return to after status operations. The power on value for this variable is L_Idle.
lunStatusCmd	This variable contains the last status command received. The power on value for this variable is 70h.
lunStatusIlv	This variable contains the interleaved address indicated in a previous 78h command. The power on value for this variable is 0h.
lunbInterleave	This variable is set to one when the LUN is performing an interleaved operation. The power on value for this variable is FALSE.
lunbIlvNextCmd	This variable is set to TRUE when the LUN is ready to receive the next interleaved command.

7.2.2. Idle command states

L_Idle[1]	lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target. lunReturnState is set to L_Idle.		
1. Target request received		→	L_Idle_TargetRequest
NOTE: 1. This state is entered asynchronously 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		→	L_RST_Execute
2. Target indicates WP# value		→	L_WP_Update
3. Target requests SR register update		→	L_SR_Update
4. Target requests status or status command received		→	L_Status_Execute
5. Target indicates output buffer should be turned off		→	L_Idle
6. Target requests page register clear		→	L_Idle_ClearPageReg
7. Target requests page register invalidate		→	L_Idle_InvalidPageReg
8. Target indicates Program request for this LUN		→	L_PP_Execute
9. Target indicates Erase request for this LUN		→	L_BE_Execute
10. Target indicates Read Page request for this LUN		→	L_RD_WaitForCmd
11. Target indicates Read Parameter Page request		→	L_Idle_RdPp
12. Target indicates Read Unique ID request		→	L_Idle_RdUid

L_WP_Update	Set lunStatus[7] to the WP# value indicated by the target.		
1. Unconditional		→	lunReturnState

L_SR_Update	Update lunStatus as 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.		
1. Parameter page data transferred to page register		→	L_Idle_RdPp_End
2. Target requests status or status command received		→	L_Status_Execute

L_Idle_RdPp_End	LUN indicates to Target that parameter page data is in page register.		
1. Unconditional		→	L_Idle_Rd

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		→	L_Idle_RdUid_End
2. Target requests status or status command received		→	L_Status_Execute

L_Idle_RdUid_End	LUN indicates to Target that Unique ID data is in page register.		
1. Unconditional		→	L_Idle_Rd

7.2.3. Idle Read states

L_Idle_Rd	lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target. lunReturnState is set to L_Idle_Rd.		
1. Background read operation complete		→	L_Idle_Rd_Finish
2. Target requests column address be selected		→	L_Idle_Rd_ColSelect
3. Read request received from Target		→	L_Idle_Rd_Xfer
4. Target request received		→	L_Idle_TargetRequest
5. Command cycle 31h (Read Cache) received		→	L_RD_Cache_Next
6. Command cycle 3Fh (Read Cache End) received and lunLastConfirm is 31h		→	L_RD_Cache_Xfer_End

L_Idle_Rd_Finish	Set lunStatus[5] to one.		
1. Unconditional		→	L_Idle_Rd

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. Unconditional		→	L_Idle_Rd

L_Idle_Rd_ColSelect	Select the column in the page register based on the column address received from the target.		
1. Unconditional		→	L_Idle_Rd

7.2.4. Status states

L_Status_Execute			
1. Target requests status value		→	L_Status_Value
2. Target indicates 78h was received		→	L_Status_Enhanced
3. Target indicates 70h was received		→	L_Status_Legacy

L_Status_Value			
1. lunbInterleave set to TRUE and lunStatusCmd set to 70h	→	L_Status_Ilv_Comp
2. lunbInterleave set to TRUE and lunStatusCmd set to 78h	→	L_Status_Ilv_Addr
3. lunbInterleave set to FALSE	→	L_Status_Lun

L_Status_Enhanced		
1. LUN in row address indicated matches this LUN	→	L_Status_Record_78h
2. Else	→	L_Status_Output_Off

L_Status_Record_78h	lunStatusCmd is set to 78h and lunStatusIlv is set to interleaved address indicated by Target. The LUN turns on its output buffer.	
1. Unconditional	→	lunReturnState

L_Status_Output_Off	LUN turns off its output buffer.	
1. lunReturnState set to L_Idle_Rd	→	L_Idle
2. Else	→	lunReturnState

L_Status_Legacy	lunStatusCmd is set to 70h.	
1. Unconditional	→	lunReturnState

L_Status_Ilv_Comp	The LUN composes the status value to return as shown: • status[7:2] = lunStatus[7:2] • status[1] = for all x, OR of lunFail[x][1] • status[0] = for all x, OR of lunFail[x][0] Return status to the Target.	
1. Unconditional	→	lunReturnState

L_Status_Ilv_Addr	The LUN composes the status value to return as shown: • status[7:2] = lunStatus[7:2] • status[1:0] = lunFail[lunStatusIlv][1:0] 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[1]	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	→	L_RST_Complete
2. Target requests status or status command received	→	L_Status_Execute

NOTE:
1. This state is entered asynchronously as a result of receiving an indication from the Target state machine to perform a Reset.

L_RST_Complete	The LUN performs the following actions: 1. lunStatus[1:0] are cleared to 00b. 2. For all interleaved 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	→	L_Idle

7.2.6. Block Erase command states

L_BE_Execute	lunbInterleave set to FALSE.	
1. Unconditional	→	L_BE_WaitForCmd

L_BE_WaitForCmd	Wait for a command cycle.	
1. Command cycle D0h received	→	L_BE_Erase
2. Command cycle D1h received	→	L_BE_Ilv

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	→	L_BE_Erase_Wait

L_BE_Erase_Wait	lunReturnState set to L_BE_Erase_Wait.		
	1. Erase of requested block(s) complete and lunbInterleave set to TRUE	→	L_BE_Ilv_Sts
	2. Erase of requested block complete	→	L_BE_Sts
	3. Target requests page register clear	→	L_Idle_ClearPageReg
	4. Target requests status or status command received	→	L_Status_Execute

L_BE_Ilv	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. lunbIlvNextCmd is set to FALSE. 6. LUN prepares to receive the next block to erase.		
	1. Unconditional	→	L_BE_Ilv_Wait

L_BE_Ilv_Wait	lunReturnState set to L_BE_Ilv_Wait.		
	1. An overlapped interleaved Erase completed	→	L_BE_Ilv_Overlap
	2. Ready to receive the next Erase command and lunbIlvNextCmd is set to FALSE	→	L_BE_Ilv_NextCmd
	3. Target indicates Erase request for this LUN and lunbIlvNextCmd is set to TRUE	→	L_BE_WaitForCmd
	4. Target requests status or status command received	→	L_Status_Execute

L_BE_Ilv_NextCmd	The LUN performs the following actions in the order specified: 1. lunbIlvNextCmd 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	→	L_BE_Ilv_Wait

L_BE_Ilv_Overlap	The LUN performs the following actions in the order specified for the overlapped interleaved operation that completed: 1. ilvComplete set to interleave address of completed operation 2. lunFail[ilvComplete][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: 1. lunStatus[0] is set to erase status. 2. lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.		
1. Unconditional		→	L_Idle

L_BE_Ilv_Sts	The LUN performs the following actions in the order specified for each interleaved operation that completed: 1. ilvComplete set to interleave address of completed operation. 2. lunFail[ilvComplete][0] is set to erase status value. lunStatus[6:5] is set to 11b and lunStatus[6] value is indicated to the Target.		
1. Unconditional		→	L_Idle

7.2.7. Read command states

If caching is not supported, then all actions for status bit 5 are ignored.

L_RD_WaitForCmd	lunbInterleave set to FALSE. Wait for a command cycle.		
1. Command cycle 30h or 35h received		→	L_RD_ArrayRead
2. Command cycle 31h received and lunLastConfirm equal to 30h or 31h		→	L_RD_Cache_Xfer

L_RD_ArrayRead	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 last command cycle (30h or 35h). 4. Read the requested page from the array. 5. lunReturnState set to L_RD_ArrayRead.		
1. Read of requested page complete		→	L_RD_Complete
2. Target requests status or status command received		→	L_Status_Execute

L_RD_Complete	lunStatus[6:5] is set to 11b. lunStatus[6] value is indicated to the Target.		
1. Unconditional		→	L_Idle_Rd

L_RD_Cache_Next	Select the next row address as the sequential increasing row address to the last page read.		
1. Unconditional		→	L_RD_Cache_Xfer

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		→	L_RD_Cache_Sts
2. Target requests status or status command received		→	L_Status_Execute

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		→	L_RD_Cache_Sts_End
2. Target requests status or status command received		→	L_Status_Execute

L_RD_Cache_Sts	lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.		
1. Unconditional		→	L_Idle_Rd

L_RD_Cache_Sts_End	lunStatus[6:5] is set to 11b. lunStatus[6] value is indicated to the Target.		
1. Unconditional		→	L_Idle_Rd

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 interleaved addressing is supported, selects the correct page register based on the interleaved 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	→	L_PP_Cache
4. Command cycle 11h (interleave) received	→	L_PP_Ilv
5. Target requests column address be selected	→	L_PP_ColSelect

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	→	L_PP_WaitForData

L_PP_Prog	The LUN performs the following actions in the order specified: 4. lunStatus[6:5] is cleared to 00h. lunStatus[6] value is indicated to the Target. 5. lunLastConfirm set to 10h. 6. LUN begins programming page specified and any previous pages specified if lunbInterleave is TRUE and concurrent interleaving is supported.		
	1. Unconditional	→	L_PP_Prog_Wait

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	→	L_PP_Ilv_Sts
	2. Write of requested page is complete and lunbInterleave is cleared to FALSE	→	L_PP_Sts
	3. Target requests status or status command received	→	L_Status_Execute

L_PP_Cache	The LUN performs the following actions in the order specified: 1. lunStatus[6:5] is cleared to 00b. lunStatus[6] value is indicated to the Target. 2. lunLastConfirm set to 15h. 3. Wait for the page register to become available for data input. 4. Start background program operation.		
	1. Unconditional	→	L_PP_Cache_Wait

L_PP_Cache_Wait	lunReturnState is set to L_PP_Cache_Wait.		
	1. Page register available for data input	→	L_PP_CacheRdy
	2. Target requests status or status command received	→	L_Status_Execute

L_PP_CacheRdy	The LUN performs the following actions: 1. If lunbInterleave is set to FALSE, then lunStatus[1] is set to the value of lunStatus[0]. 2. If lunbInterleave is set to TRUE, then for all interleaved addresses, x, lunFail[x][1] is set to the value of lunFail[x][0]. 3. lunStatus[6] is set to one. lunStatus[6] value is indicated to the Target.

1. Unconditional	→	L_PP_CacheRdy_Wait

L_PP_CacheRdy_Wait	lunReturnState set to L_PP_CacheRdy_Wait.	
1. Previous cache operation complete and lunbInterleave set to TRUE	→	L_PP_Ilv_Cache_Sts
2. Previous cache operation complete	→	L_PP_Cache_Sts
3. Target indicates Program request for this LUN	→	L_PP_Addr
4. Target requests page register clear	→	L_Idle_ClearPageReg
5. Target requests status or status command received	→	L_Status_Execute

L_PP_Ilv	The LUN performs the following actions in the order specified: 1. lunbInterleave set to TRUE. 2. lunStatus[6:5] is cleared to 00b. lunStatus[6] value is indicated to the Target. 3. lunLastConfirm set to 11h. 4. lunbIlvNextCmd is set to FALSE. 5. LUN begins programming page specified if overlapped interleaving is supported. 6. Prepare to receive next page to program.	
1. Unconditional	→	L_PP_Ilv_Wait

L_PP_Ilv_Wait	lunReturnState set to L_PP_Ilv_Wait.	
1. An overlapped interleaved Program completed	→	L_PP_Ilv_Overlap
2. A previous cache Program completed	→	L_PP_Ilv_Cache_Sts
3. Ready to receive the next Program command and lunbIlvNextCmd is set to FALSE	→	L_PP_Ilv_NextCmd
4. Target indicates Program request for this LUN and lunbIlvNextCmd is set to TRUE	→	L_PP_Addr
5. Target requests status or status command received	→	L_Status_Execute

L_PP_Ilv_NextCmd	The LUN performs the following actions in the order specified: 1. lunbIlvNextCmd 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	→	L_PP_Ilv_Wait

L_PP_Sts	The LUN performs the following actions in the order specified: 1. lunStatus[1] is set to program status of previous operation 2. lunStatus[0] is set to program status of final operation 3. lunStatus[6:5] is set to 11b. 4. lunStatus[6] value is indicated to the Target.	
1. Unconditional	→	L_Idle

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_Ilv_Cache_Sts	The LUN performs the following actions in the order specified for all completed cache operations: 1. ilvAddr set to interleave address of cache operation. 2. lunFail[ilvAddr][0] is set to program status. If all array operations are complete, lunStatus[5] is set to one.		
1. Unconditional		→	lunReturnState

L_PP_Ilv_Overlap	The LUN performs the following actions in the order specified for the overlapped interleaved operation that completed: 1. ilvComplete set to interleave address of completed operation 2. lunFail[ilvComplete][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_Ilv_Sts	The LUN performs the following actions in the order specified for each interleaved operation that completed: 1. ilvComplete set to interleave address of completed operation 2. lunFail[ilvComplete][1] is set to program status of previous operation. 3. lunFail[ilvComplete][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		→	L_Idle

L_PP_ColSelect	Select the column in the page register based on the column address received that the target requested.		
1. Unconditional		→	L_PP_WaitForData

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;
      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; 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;
}
```