NAND Flash Controller Programming Model

Rev: A0-0E

08 August 2016

NAND Flash controller Programming Model

For

Software team VVDN

Revision History:

Date	Rev No.	Description	By
31-Aug-2015	A0-00	First Draft version	VVDN
01-Sept-2015	A0-01	PCIe bar space updated	VVDN
21-Nov-2015	A0-02	Block diagrams updated	VVDN
08-Dec-2015	A0-03	Command structure parameters updated	VVDN
09-Dec-2015	A0-04	Updated the descriptor table structure	VVDN
11-Dec-2015	A0-05	Updated the registers and command structure	VVDN
26-Dec-2015	A0-06	Added ECC enable/disable control signal	VVDN
04-Jan-2016	A0-07	Added Change write column command. Removed Interrupt control register. Arbitration scheme explained.	VVDN
18-Jan-2016	A0-08	Removed Intr bit in CSR register. ECC is changed from 24 to 16 bit-correction for optimizing FPGA resource.	VVDN
29-Jan-2016	A0-09	Descriptor Table offset's and control register offset's changed.	VVDN
15-Feb-2016	A0-0A	Descriptor Control Register offset's changed.	VVDN
08-April-2016	A0-0B	Descriptor structure and Command Structure clubbed together	VVDN
13-May-2016	A0-0C	16KB Page access to NAND Flash	VVDN
01-July-2016	A0-0D	New Descriptor Design Updated	VVDN
08-Aug-2016	A0-0E	New Set Feature Commands Added	VVDN

Table of Contents

List of Figures

List of Tables

1 Introduction

This document describes high level description of NAND flash controller in FPGA logic, to meet the requirements of the customer.

This document is made for the reference of

- Design team of VVDN to convey their design to MIT Team for Software Integration(FTL)
- For understanding the Data Flow
- VVDN for validation/verification of the deliverables.

1.1 Product Overview

The FPGA logic enables the data movement from Host PC via LS2 to the NVDIMM connected to FPGA. The interface between LS2 and FPGA is via PCIe Gen2.0 Interface. The flash controller act as an interface module between PCIe and DIMM module which is having 4 flash IC's (part no: MT29F512G08CKCCBH7-6R:C).

1.2 System Requirement

- System should support 4 DIMM slots
- Each DIMM slot is having a 32 bit interface. Four 8 bit NAND chip tied together to form the 32 data interface
- The DIMM should support NVDDR2 interface, should be compactable with ONFI 3.2
- Host side interface is PCIe Gen 2.0
- 16 bit ECC correction capability for 1 KB.

1.3 High level block diagram

High level system diagram is shown below. The design contains two similar Data path consist of a PCIE end point and two DIMM Interfaces.

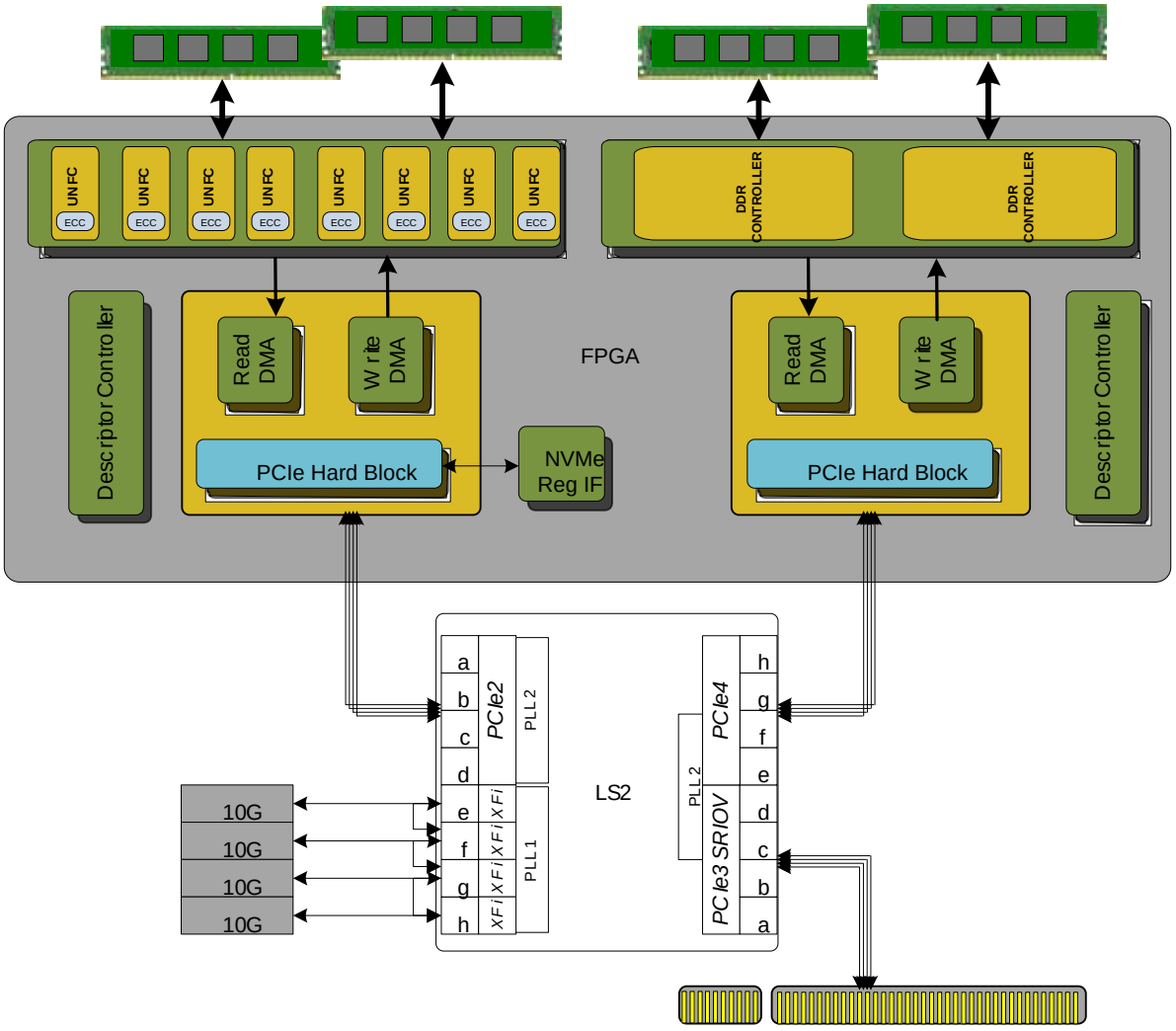


Figure 1: High level system diagram

1.3.1 Data Path Diagram

This is the data path from the PCIe Endpoint interface to the NAND Flash IC in the NVDIMM. One PCIe end point is interfaced with 8 UNFC controllers.

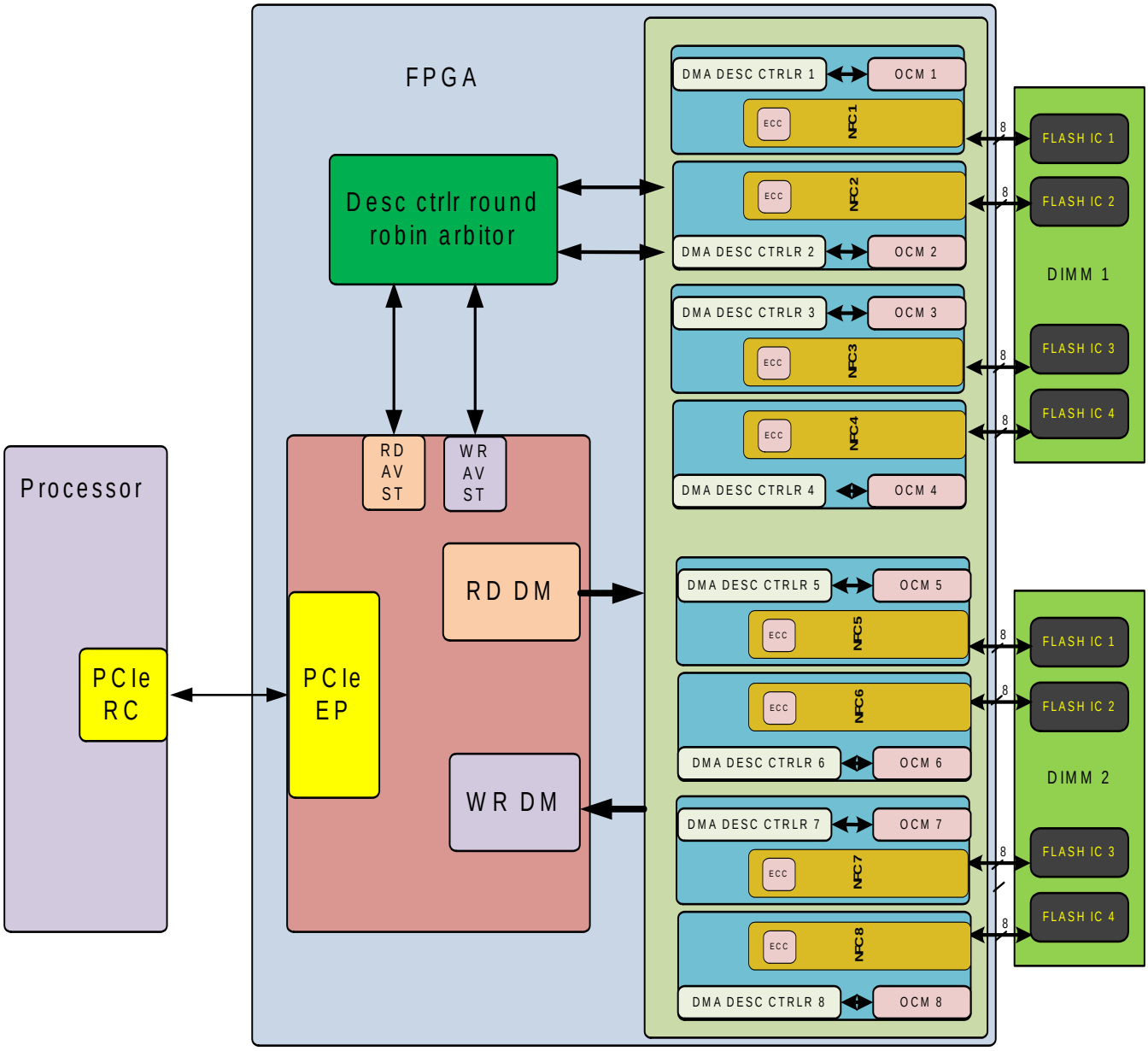


Figure 2: FPGA high level Block Diagram

Each UNFC controller can be independently configured and controlled

- Data from the processor is loading to FPGA via x4 PCIe link.
- Two DIMM's per x4 PCIe links
- One PCIe Gen2.0 IP handles 8 Flash Chips
- DMA Data mover with separate Read and Write channel is implemented inside FPGA
- The DMA data path will be 128 bit wide running at 125 MHz
- DIMM module is 32 bit wide data interface.
- 2 NV-DIMM support
- Data read and write to the flash controller to be controlled by RD DM (read data mover) and WR DM (write data mover).
- Descriptor controller gives command to RD DM and WR DM based on the descriptor table from the OCM (On Chip Memory).
- The descriptor table to contain command and configuration information of the flash controller.

The detailed data and configuration path diagram of one flash IC is shown below.

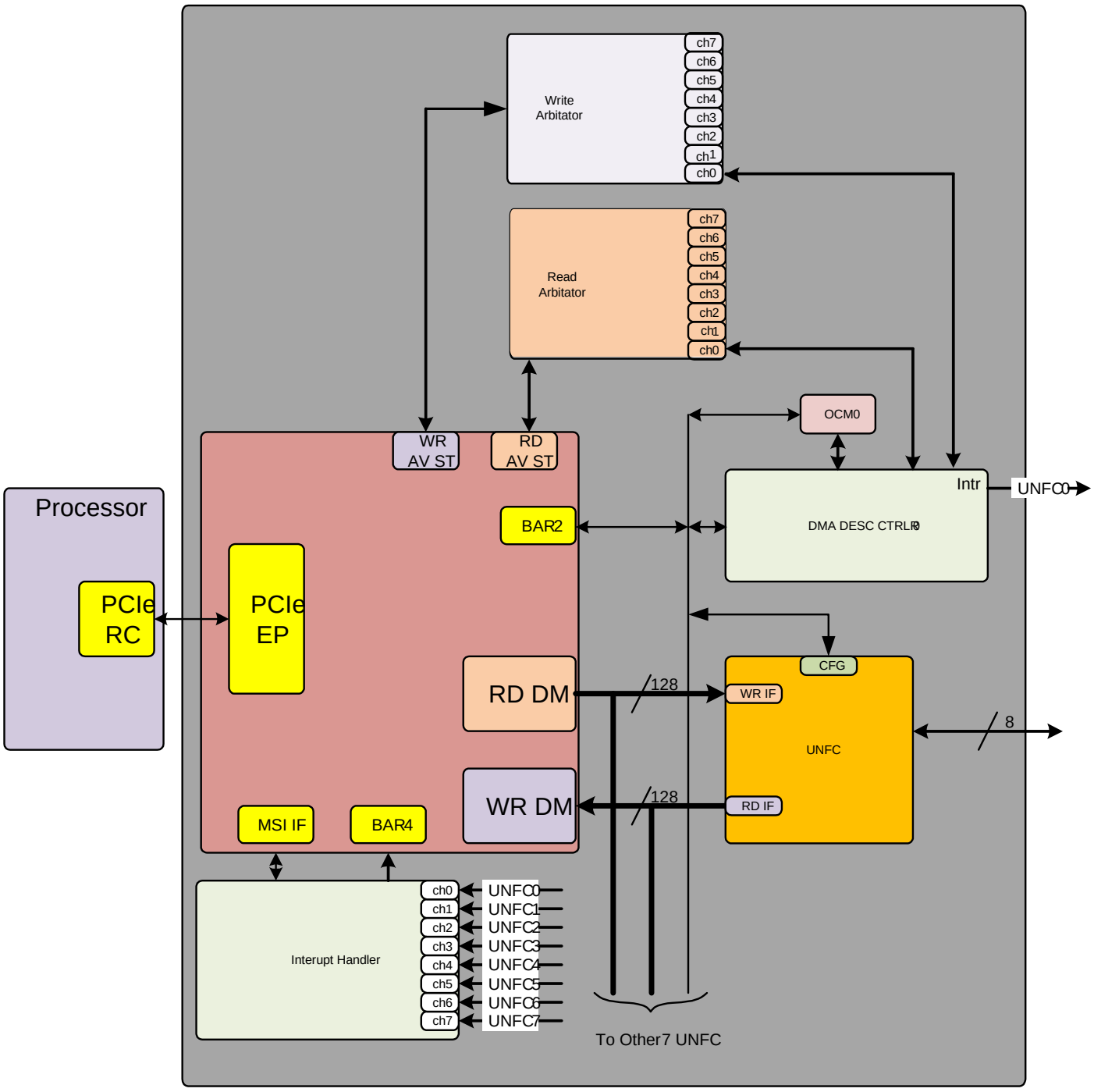
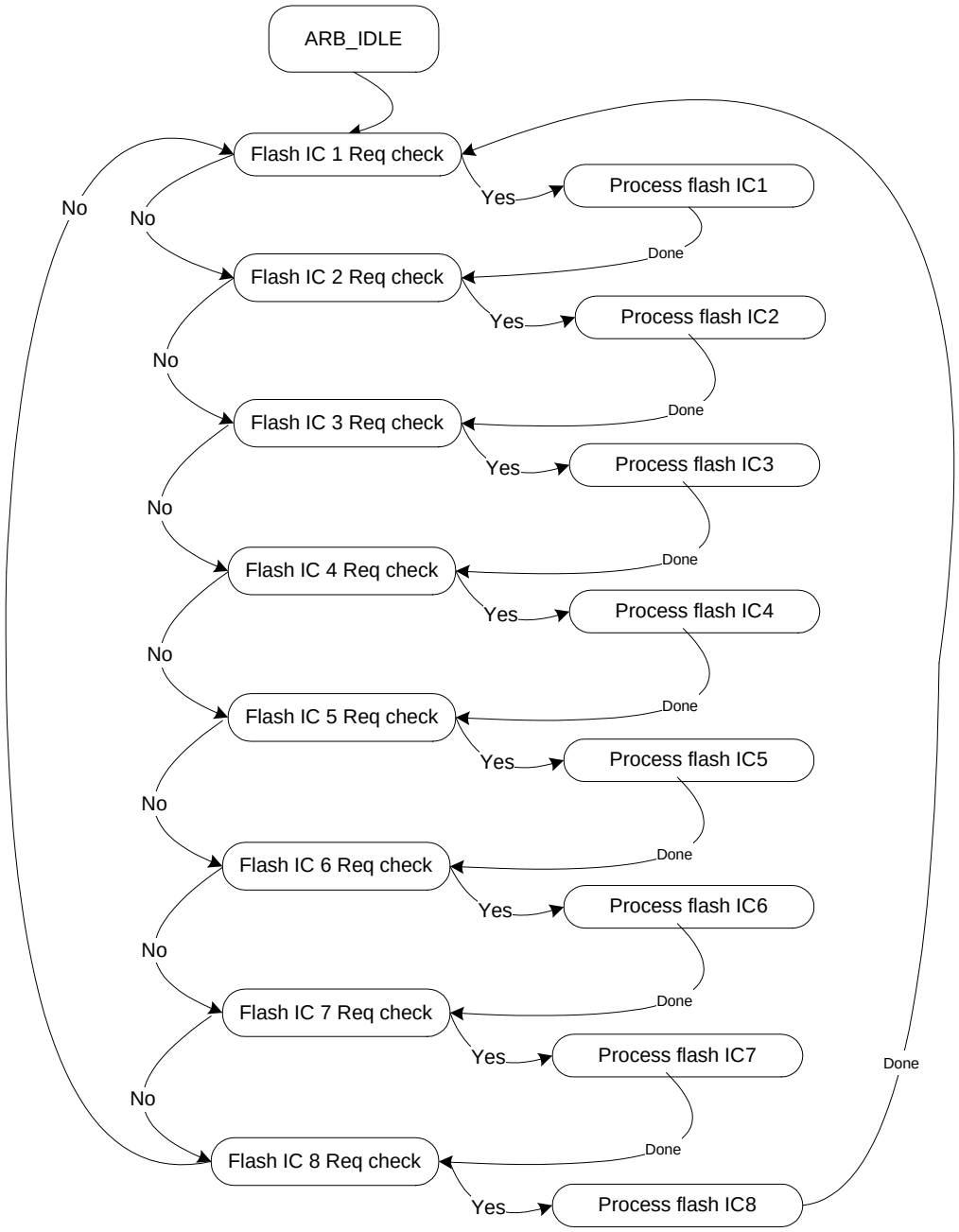


Figure 3: Data and Configuration path Diagram for a single UNFC

1.3.2 Arbitration



Round robin arbitration scheme is used for arbitration between 8 NAND controllers and single PCIe DMA buses. There will be separate round robin arbitrator for Read DMA as well as Write DMA. Arbitrator will start form NAND controller 1 and if there is a request, it will be processed and then move to next NAND flash controller. If there is no request arbitrator will go to next NAND flash controller and check for request.

Write Buffer Path: This path is used to transfer the data from Host side to NAND controller buffer via **Read Data Mover** in PCIe DMA Block.

Read Buffer Path: This path is used to transfer the data from NAND controller buffer to host via **Write Data Mover** in PCIe DMA Block.

Command Buffer Path: This path is used to load the command to command buffer in the flash controller via **Write Data Mover** in PCIe DMA Block

Configuration Buffer Path: This path is used to access the configuration registers inside the controller via PCIe BAR register.

On chip memory path: This path is used to load and read back descriptors in the OCM via PCIe BAR register.

Read Avalon Stream Path: This path is used to give address and commands and take status of the **Read Data Mover** in PCIe DMA Block.

Write Avalon Stream Path: This path is used to give address and commands and take status of the **Write Data Mover** in PCIe DMA Block.

1.4 Memory Map

1.4.1 Bar 2 Memory Map

	Descriptor structure OCM
Desc ctrlr 0	0x0000 - 0x03FF
Desc ctrlr 1	0x0800 - 0x0BFF
Desc ctrlr 2	0x1000 - 0x13FF
Desc ctrlr 3	0x1800 - 0x1BFF
Desc ctrlr 4	0x2000 - 0x23FF
Desc ctrlr 5	0x2800 - 0x2BFF
Desc ctrlr 6	0x3000 - 0x33FF
Desc ctrlr 7	0x3800 - 0x3BFF

Figure 4: Bar 2 memory mapping diagram

1.4.2 Bar 4 Memory Map

	Control Registers
Desc ctrlr 0	0x0000 - 0x003F
Desc ctrlr 1	0x0040 - 0x007F
Desc ctrlr 2	0x0080 - 0x00BF
Desc ctrlr 3	0x00C0 - 0x00FF
Desc ctrlr 4	0x0100 - 0x013F
Desc ctrlr 5	0x0140 - 0x017F
Desc ctrlr 6	0x0180 - 0x01BF
Desc ctrlr 7	0x01C0 - 0x01FF

Figure 5: Bar 4 memory mapping diagram

1.5 DMA Read and Write Bus Memory Mapping

The DMA data movers have separate dedicated Bus for read and write movement. So separate address Map is available for both read and write data path. This will be transparent from the Software side.

2 NVDIMM Topology

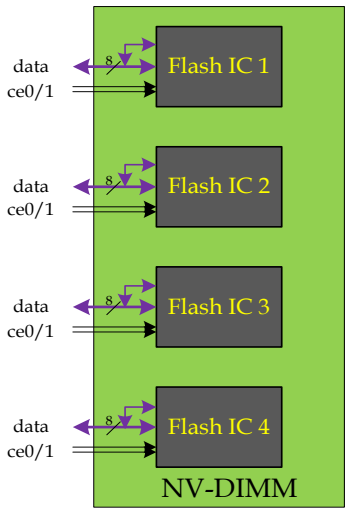


Figure 6: NVDIMM Topology

- NVM DIMM is basically four 8-bit NAND channels, each connected to one ***MT29F512G08CKCCBH7-6R:C*** micron NAND chip (4 total per DIMM).
- Each NAND chip comes with two internal 8-bit busses; they will be shorted on NVM DIMM PCB, providing single 8-bit data bus, but with separate Chip Select lines.

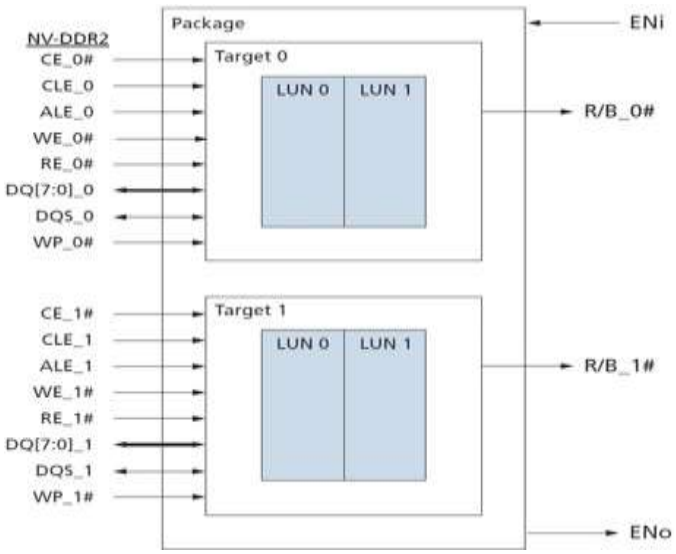


Figure 7: NAND device Configuration

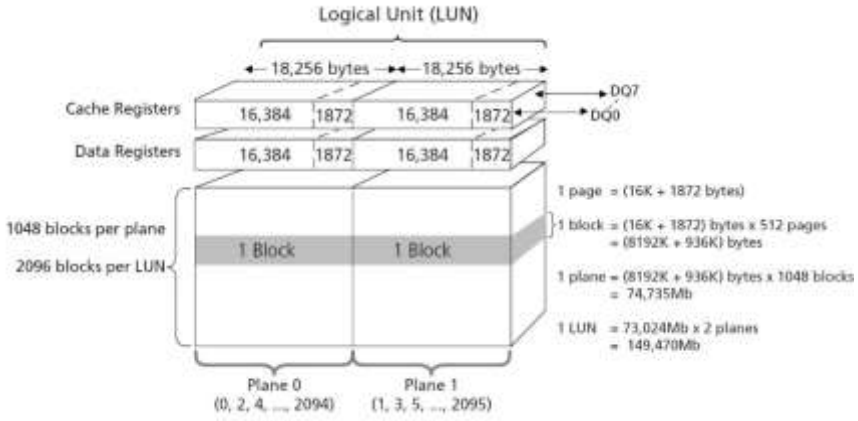


Figure 8: LUN Structure

Page Size	128	Kb
Block Size	512	Page
Plane	1048	Block
LUN	2	Plane
Target	2	LUN
Chip	2	Target
Total	524	Gb

Table 1: NAND Size Calculation

Total DIMM Capacity is 4 X 524 GBit = 2096 GBit.

3 Theory of operation

Each UNFC controller works independently. The below section explained the working of a single UNFC controller.

From the high level block diagram one PCIe DMA needs to serve 8 independent UNFC controllers. This will happen in a round robin fashion which will be transparent to the software.

Based on the requirement from the top layer FTL will create this CMD structure. CMD structure will be clubbed with descriptor structure and it will be stored in the OCM of FPGA. This contains the necessary parameter for the NAND flash controller to execute a read or write operation. Descriptor field contains 5 data pointer including data pointer for Spare area. Data Pointer points to a memory in X86 which contain 17 KB of data for a write operation or a data buffer to hold read data in a read operation. With the help of one Descriptor Main Page Data and OOB data both can be Written or Read Back from NAND Flash. FPGA is having 16KB of internal memory for each NAND Flash Controller to store one page data of NAND Flash During Write and Read operation.

The DMA descriptor controller read the descriptor entry and loads the command buffer parameter from the CMD structure and read/write data from/to the Data buffer pointed by the data pointer in the CMD structure.

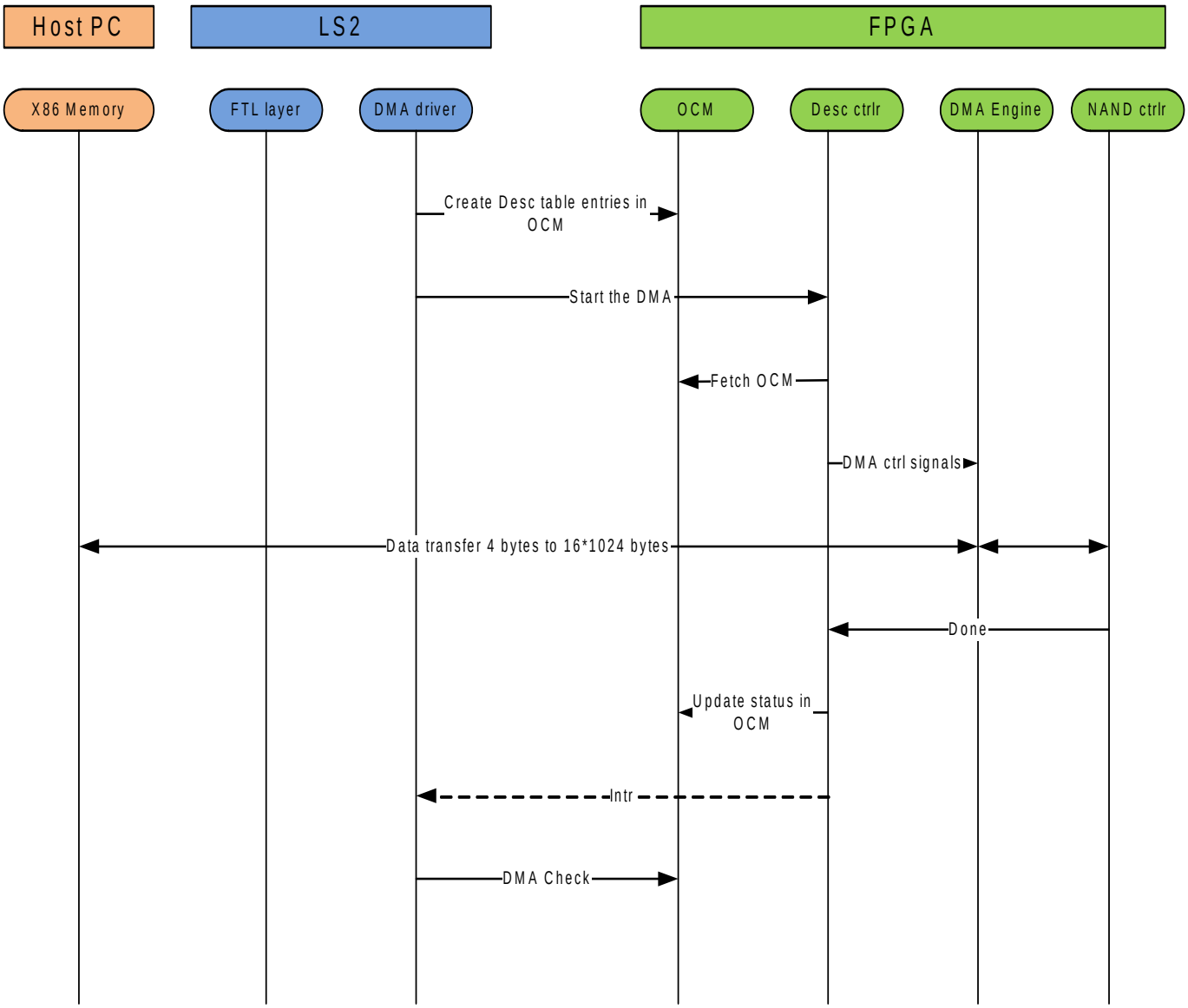


Figure 9 : Sequence Diagram for DMA Driver

3.1 Descriptor Structure

The descriptor table is a collection of descriptors, with each descriptor contain all the necessary information for a data move operation.

Row Address																															
Target	Column Address																														
Data buffer pointer-1 LSB																															
Data buffer pointer-1 MSB																															
Data buffer pointer-2 LSB																															
Data buffer pointer-2 MSB																															
Data buffer pointer-3 LSB																															
Data buffer pointer-3 MSB																															
Data buffer pointer-4 LSB																															
Data buffer pointer-4 MSB																															
Data Length 2																Data Length 1															
Data Length 4																Data Length 3															
Spare data buffer pointer LSB																															
Spare data length																Spare data buffer pointer MSB															
Channel ID and Buffer ID																															
Reserved														Chng_wr_col	Chck rdy_bsy	Ownership	Descriptor_ID									DMA_Cmplt	hold	INTR	Command		ECC	No Data	Dir

Table 2: Descriptor structure

Name	Description
Row Address	This register defines the address row selected in the NAND Flash page: [31: 24]: Reserved [23: 16]: Row 3 [15: 8]: Row 2 [7: 0]: Row 1/ Address The values of Row3, Row2 and Row 1 depend on the NAND Flash component. Address is used for Set feature, Get feature and Volume Select commands.
Column Address and Target	This register defines the address column selected in the NAND Flash page: [28: 16]: Reserved [15: 8]: Column 2 [7: 0]: Column 1
	Target selection [31:29]. Gives the device in a NAND Bank channel targeted by the command 3b000: Device 0 3b001: Device 1 3b010: Device 2 3b011: Device 3 3b100: Device 4 3b101: Device 5 3b110: Device 6 3b111: Device 7
Data Pointer	Host Data Pointer-1 LSB
	Host Data Pointer -1MSB
	Host Data Pointer-2 LSB
	Host Data Pointer -2MSB
	Host Data Pointer-3 LSB
	Host Data Pointer -3MSB
	Host Data Pointer-4 LSB
	Host Data Pointer -4MSB
Length	Length of data Pointer-1 [15:0]

	Length of data Pointer-2 [31:16]
	Length of data Pointer-3 [15:0]
	Length of data Pointer-4 [31:16]
Spare Data Pointer	Spare Data Pointer LSB [31:0]
	Spare Data Pointer MSB [19:0]
	Spare data length [31:20]
Channel ID and Buffer ID	Reserved [31:0]
Direction Control	Direction of the data transfer in Data phase. DIR = 0 means read DMA operation. DIR = 1 means write DMA operation. For command only DMA DIR = 0.
No data	No Data This parameter indicates that “no data” is used with this command if the bit is set to 1.
ECC	ECC disable 1’b1: Deactivate the ECC generation and check 1’b0: Activate ECC generator and checker Minimum 1 KB data to be transferred for ECC generation and correction
Command	This register defines the command selected. 0: Reserved 1: RESET 2: SRESET 3: RESET_LUN 4: RESET_CHANNEL_ALL 5: BLOCKERASE 6-7: Reserved 8: COPYBACKWRITE 9: SET_FEATURE A: PAGEPROG B: PAGEPROG_CACHE C: CHANGE WRITE COLUMN D–F: Reserved 10: GET_FEATURE 11: READ_UNIQUE 12: READ_PARAMETER 13: READID 14: READSTATUS 15: READSTATUS_ENHANCED 16 -17: Reserved 18: CHANGEREADCOL 19: CHANGEREADCOL_ENHANCED

	1A: COPYBACKREAD 1B: READ_CACHE_SEQ 1C: READ_CACHE_RAND 1D: READ_CACHE_END 1E: READ 1F- 20: Reserved 21: LUN_SET_FEATURE 22: LUN_GET_FEATURE 23: BLOCKERASE_MULTIPLANE 24: PAGEPROG_MULTIPLANE 25: CHANGE WRITE COLUMN CACHE 26: CHANGE WRITE COLUMN MULTIPLANE
Interrupt	Host can set this bit to 1 to raise the interrupt after that particular descriptor operation is completed.
Hold	Host can set this bit to one to hold the next descriptor fetch till the start signal from host.
Dma Cmplt	This bit will be set once the particular descriptor is processed successfully.
Desc ID	ID of the descriptor
Ownership	To set the ownership of the descriptor between FPGA and processor. If this bit is set to 1 by host, then the ownership is transferred to FPGA DMA Descriptor Controller, Descriptor controller can fetch and execute that descriptor. After completing a descriptor operation FPGA will update the status of operation in to the status field and set this bit to 0 to transfer the ownership back to processor.
Chk_rdy_bsy	This bit is set to use Multi Lun feature.
Multiplane_bit	This bit is use for Change write column multiplane or Page Program Multiplane Operation. If this bit is set Multiplane operation will be done.

4 NAND Addressing

	Cycle	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
Column 1	First	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
Column 2	Second	LOW	CA14	CA13	CA12	CA11	CA10	CA9	CA8
Row 1	Third	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Row 2	Fourth	BA15	BA14	BA13	BA12	BA11	BA10	BA9	PA8
Row 3	Fifth	LOW	LOW	LA0	BA20	BA19	BA18	BA17	BA16

Table 3 NAND Addressing

Note: Refer L95B_128_256_512Gb_1Tb_2Tb_Async_Sync_NAND.pdf page number 25 for further details.

5 NAND Operations

5.1 RESET (FFh)

The RESET (FFh) command is used to put a target into a known condition and to abort Command sequences in progress

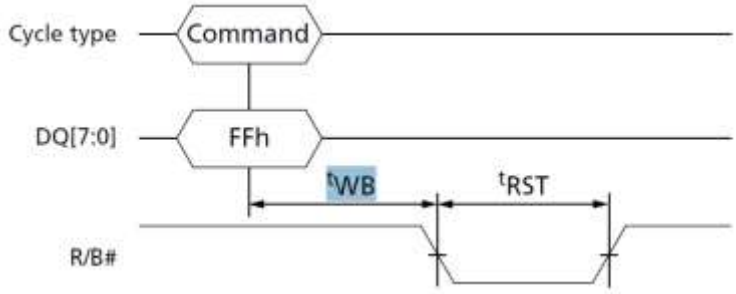


Figure 10: Command Sequence

CMD Structure parameters:

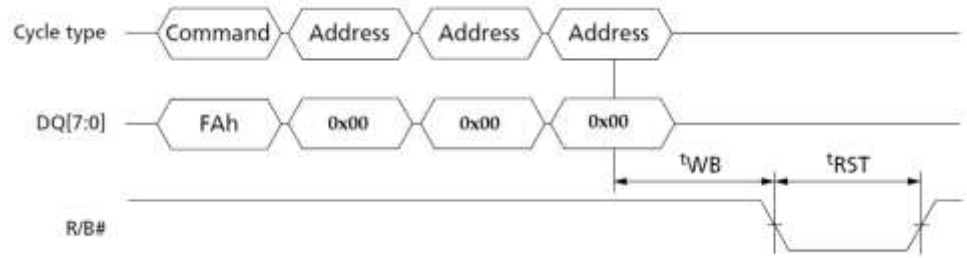
Reset		**Row Address**	NA
		Colom Address	NA
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	0
		ECC disable	1
		No data	1
		No program	0
		Command select	RESET (0x01)
		Data Buffer pointer	NA

Figure 11: CMD Structure parameters

5.2 LUN RESET (FAh)

The RESET LUN (FAh) command is used to put a particular LUN on a target into a Known condition and to abort command sequences in progress

LUN0:



LUN1:

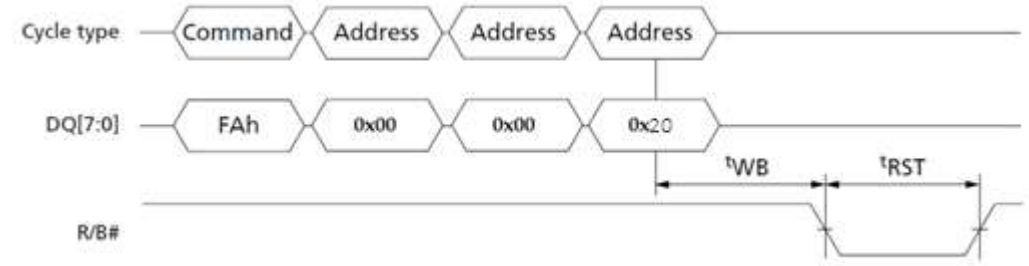


Figure 12: Command Sequencing

CMD Structure parameters:

Reset LUN		**Row Address**	XX<LA>X_XXXX
		Colom Address	NA
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	0
		ECC disable	1
		No data	1
		No program	0
		Command select	RESET_LUN (0x03)
		Data Buffer pointer	NA

Figure 13: CMD Structure parameters

Note: this will reset the LUN1: Bit5 of Row3 will select the LUN Number.

5.3 Set Feature Operation

This command is used to modify the target's default power-on behavior. This command use a one-byte feature address to determine which sub feature parameters will be read or modified. This command is handled in a specific way in the current design compared to other commands.

Writing EFh to the command register puts the target in the set features mode, which is followed by a valid feature address. After all four sub feature parameters are input, the target goes busy for tFEAT.

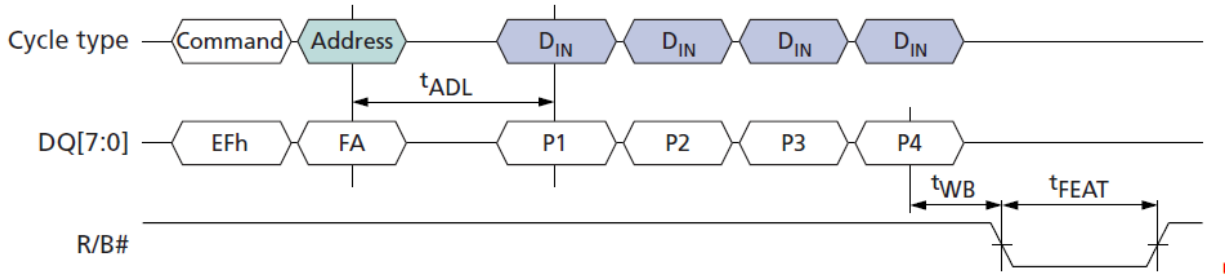


Figure 14: Command sequencing

CMD Structure parameters:

Set feature		**Row Address**	XXXX_XX<FEATURE_ADDRESS>
		Colom Address	NA
		Buffer ID	0
		Channel ID	0
		Length	4
		Target Select	<Target>
		Direction control	0
		ECC disable	1
		No data	0
		No program	0
		Command select	SET FEATURE (0x09)
		Data Buffer pointer	Write Buffer pointer

Figure 15: CMD Structure parameters

If the SETFEATURE is used to change the interface configuration like interface timing / Transfer mode (Async, NVDDR, NVDDR2).

5.3.1 Set Feature for ODT

On-die termination may be required at higher speeds depending on system topology. On-die termination applies to the DQ[7:0], DQS_t, DQS_c, RE_t, and RE_c signals. On die termination is an optional capability that may be employed to meet higher speeds in particular topologies. If power needs to be optimized in a particular condition, then on-die termination may be disabled and the topology may be run at a slower timing mode. On-die termination shall only be enabled for use in the NV-DDR2 data interface.

For enabling ODT with 100 ohm termination, give the command as:
If the current mode is Asynchronous,

Feature Address : 02h
Length : 04h
Value : 20h
Eg: sudo ./a.out 9 2 4 20 00 00 00

If the current mode is Synchronous (NVDDR2),

Feature Address : 02h
Length : 08h
Value : 20h
Eg: sudo ./a.out 9 2 8 20 20 00 00 00 00 00 00

5.3.2 Set Feature for Programmable Output Drive Strength

Because NAND Flash is designed for use in systems that are typically point-to-point connections, an option to control the drive strength of the output buffers is provided. Drive strength should be selected based on the expected loading of the memory bus. There are four supported settings for the output drivers: 18 ohms, 25 ohms, 35 ohms, and 50 ohms. The 35 ohm output drive strength setting is the power-on default value. The host can select a different drive strength setting using the SET FEATURES (EFh) command.

For enabling Output Drive Strength with 18 ohm , give the command as:
If the current mode is Asynchronous,

Feature Address : 10h & 80h
Length : 04h
Value : 00h
Eg: sudo ./a.out 9 10 4 00 00 00 00
sudo ./a.out 9 80 4 00 00 00 00

If the current mode is Synchronous (NVDDR2),

Feature Address : 10h & 80h
Length : 08h
Value : 00h
Eg: sudo ./a.out 9 10 8 00 00 00 00 00 00 00 00
sudo ./a.out 9 80 8 00 00 00 00 00 00 00 00

5.3.3 Set Feature for Selecting Data Interface and Timing Mode

For selecting desired data interface and timing mode, give the command as:

Feature Address : 01h
Length : 04h
Value : 22h (NVDDR2 Mode2)

Eg: sudo ./a.out 9 1 4 22 00 00 00

Note: Refer L95B_128_256_512Gb_1Tb_2Tb_Async_Sync_NAND.pdf page number 103 for further details.

5.4 Get Feature Operation

The GET FEATURES (EEh) command reads the sub feature parameters (P1-P4) from the specified feature address. When the EEh command is followed by a valid feature address the target goes busy for tFEAT. After tFEAT completes, the host enables data output mode to read the sub feature parameters.

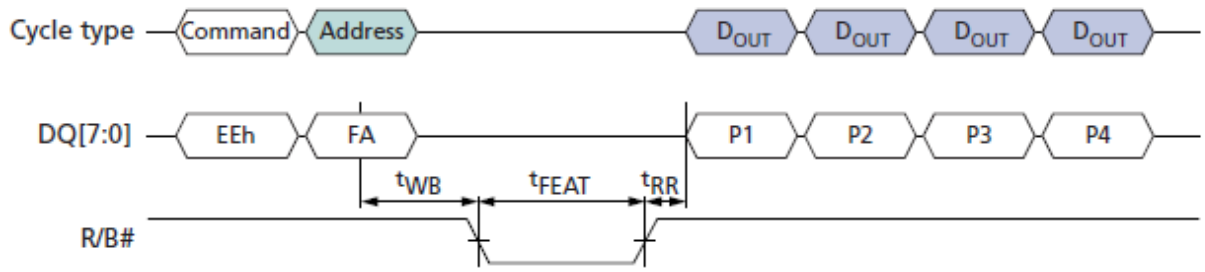


Figure 16: Command sequencing

CMD Structure parameters:

Get feature		**Row Address**	XXXX_XX<FEATURE_ADDRESS>
		Colom Address	NA
		Buffer ID	0
		Channel ID	0
		Length	4
		Target Select	<Target>
		Direction control	1
		ECC disable	1
		No data	0
		No program	0
		Command select	GET FEATURE (0x10)
		Data Buffer pointer	Read Buffer pointer

Figure 17: CMD Structure parameters

5.5 Read Parameter Page

The Read Page Parameter command is used to read the ONFI or JEDEC parameter page programmed into the target.
A minimum of three copies of the parameter page are stored in the device. Each parameter page is 256 bytes.

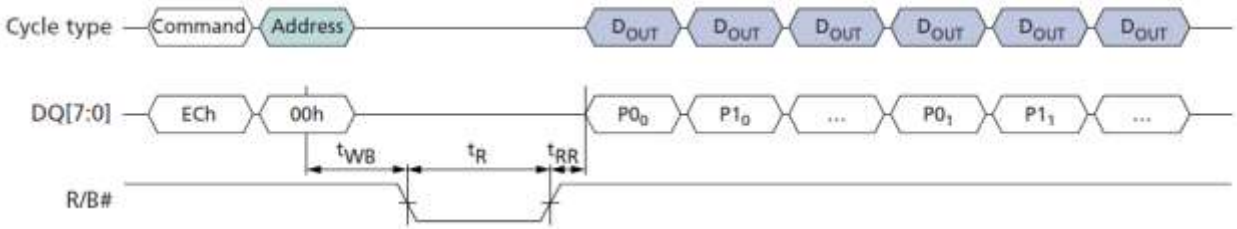


Figure 18: Command Sequencing for ONFI

CMD Structure parameters:

Read Parameter		**Row Address**	NA
		Colom Address	NA
		Buffer ID	0
		Channel ID	0
		Length	3*256
		Target Select	<Target>
		Direction control	1
		ECC disable	1
		No data	0
		No program	0
		Command select	READ PARAMETER (0x12)
		Data Buffer pointer	Read Buffer pointer

Figure 19: CMD Structure parameters

5.6 Read status

The READ STATUS (70h) command returns the status of the last-selected die (LUN) on a target.

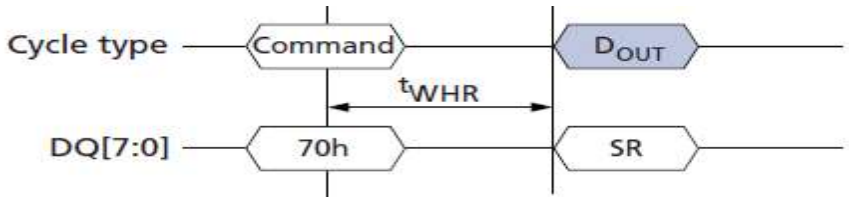


Figure 20: Command Sequencing

CMD Structure parameters:

Read Status		**Row Address**	NA
		Colom Address	NA
		Buffer ID	0
		Channel ID	0
		Length	1
		Target Select	<Target>
		Direction control	1
		ECC disable	1
		No data	0
		No program	0
		Command select	READ STATUS (0x14)
		Data Buffer pointer	Read Buffer pointer

Figure 21: CMD Structure parameters

5.7 Read Status Enhanced

The READ STATUS ENHANCED (78h) command returns the status of the addressed die (LUN) on a target even when it is busy (RDY = 0). This command is accepted by all die (LUNs), even when they are BUSY (RDY = 0).

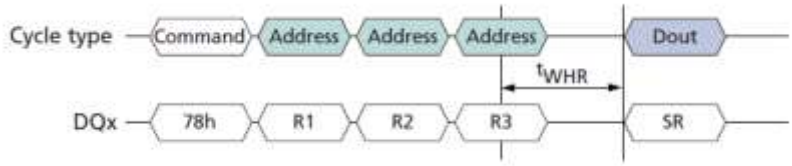


Figure 22: Command Sequencing

Read Status Enhanced		**Row Address**	NA
		Colom Address	NA
		Buffer ID	0
		Channel ID	0
		Length	1
		Target Select	<Target>
		Direction control	1
		ECC disable	1
		No data	0
		No program	0
		Command select	READ STATUS ENHANCED (0x15)
		Data Buffer pointer	Read Buffer pointer

Figure 23: CMD Structure parameters

5.8 Read Operation

5.8.1 Single page Read

A page is read from NAND to Host Memory.

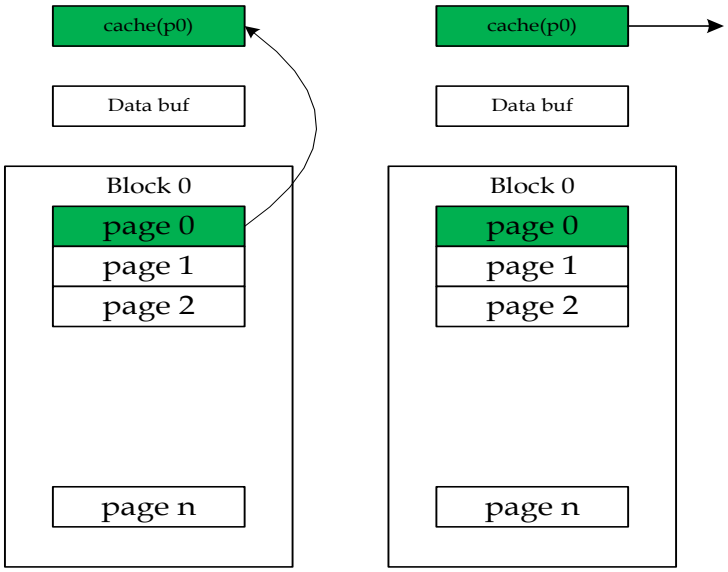


Figure 24: Data Flow Diagram

Figure 25: Command Sequence

CMD Structure parameters:

Read Page		**Row Address**	<LUN><Plane><Block><Page>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	0
		No program	0
		Command select	READ PAGE (0x1E)
		Data Buffer pointer	Read Buffer pointer

Figure 26: CMD Structure parameters

DATA: A 16KB location need to be allocated for the read data by the DMA Driver.

5.8.2 Partial Page Read

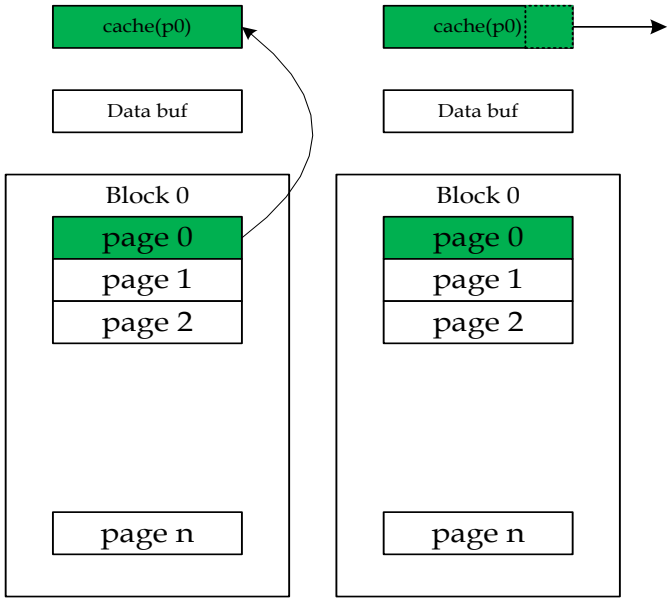


Figure 27: Data Flow Command

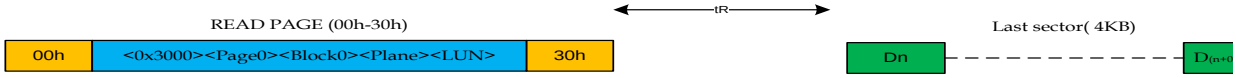


Figure 28: Command Sequence

CMD Structure parameters:

Partial Page Read		**Row Address**	<LUN><Plane><Block><Page>
		Colom Address	<Column Address>
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	0
		No program	0
		Command select	READ PAGE (0x1E)
		Data Buffer pointer	Read Buffer pointer

Figure 29: CMD Structure parameters

5.8.3 Sequential Page Read

This is used to transfer a bulk of data from sequential pages

Use Case: Bulk Read

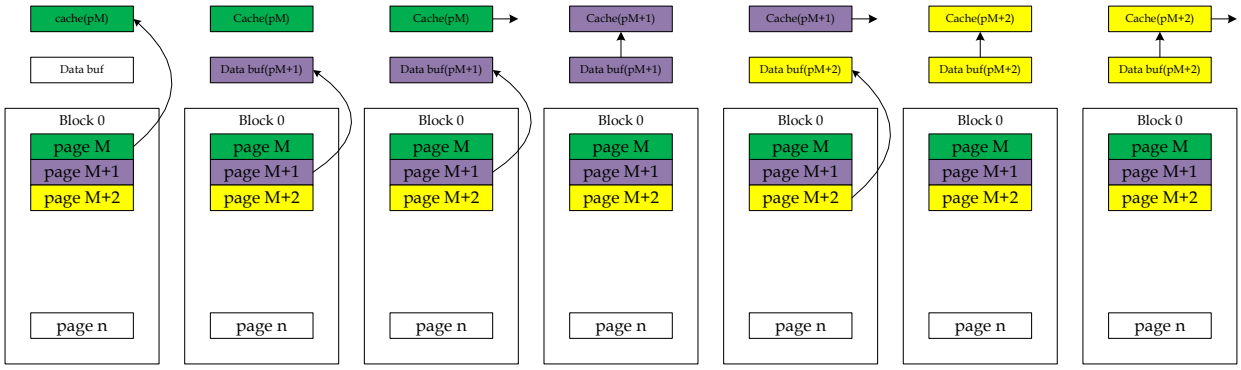


Figure 30: Data Flow Diagram

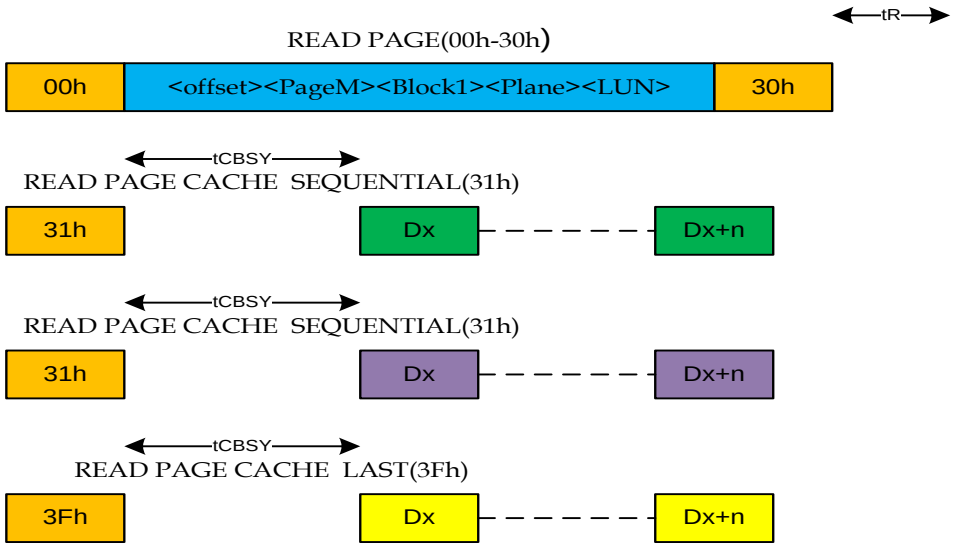


Figure 31: Command Sequence

CMD Structure parameters:

Read Page	0	**Row Address**	<LUN><Plane><Block><Page M>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	1
		No program	0
		Command select	READ PAGE (0x1E)
		Data Buffer pointer	NA
Read Cache Seq	1	**Row Address**	NA
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	0
		No program	0
		Command select	READ CACHE SEQ (0x1B)
		Data Buffer pointer	Read Buffer pointer 0
Read Cache Seq	2	**Row Address**	NA
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	0
		No program	0
		Command select	READ CACHE SEQ (0x1B)
		Data Buffer pointer	Read Buffer pointer 1
Read Cache End	2	**Row Address**	NA
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	0
		No program	0
		Command select	READ CACHE END (0x1D)
		Data Buffer pointer	Read Buffer pointer 2

Figure 30: CMD Structure parameters

5.8.4 Random Page Read Cache

This is used to read a bulk of data that are not in continuous NAND pages.

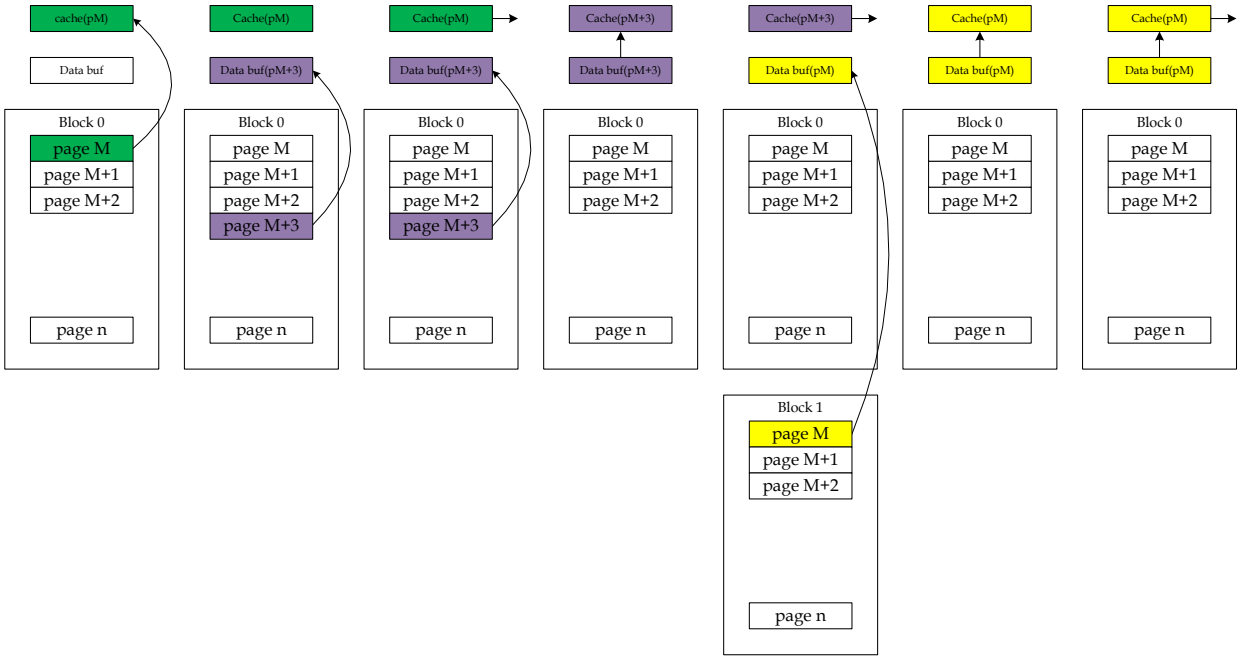


Figure 31: Data Flow Diagram

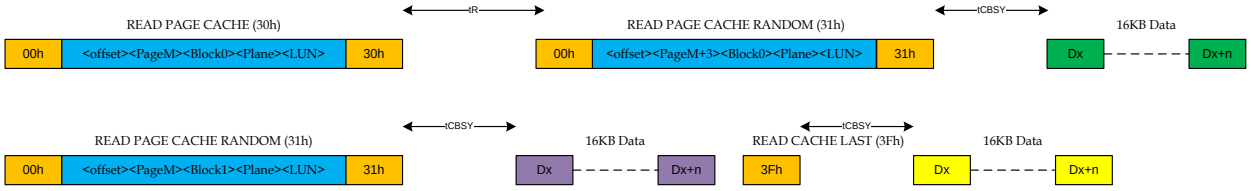


Figure 32: Command Sequence

CMD Structure parameters:

Read Page	0	**Row Address**	<LUN><Plane><Block><Page M>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	1
		No program	0
		Command select	READ PAGE (0x1E)
		Data Buffer pointer	NA
Read Cache Rand	1	**Row Address**	<LUN><Plane><Block0><PageM+3>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	0
		No program	0
		Command select	READ CACHE RAND (0x1C)
		Data Buffer pointer	Read Buffer pointer 0
Read Cache Rand	2	**Row Address**	<LUN><Plane><Block1><PageM>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	1
		ECC disable	0
		No data	0
		No program	0
		Command select	READ CACHE RAND (0x1C)
		Data Buffer pointer	Read Buffer pointer 1
Read Cache End	3	**Row Address**	NA
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		ECC disable	0
		Direction control	1
		No data	0
		No program	0
		Command select	READ CACHE END (0x1D)
		Data Buffer pointer	Read Buffer pointer 2

Figure 33: CMD Structure parameters

5.9 Page Program

5.9.1 Single Page Program

This is used to program a single page.

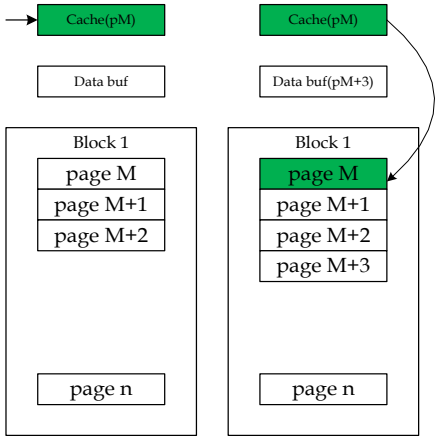


Figure 36: Data Flow Diagram

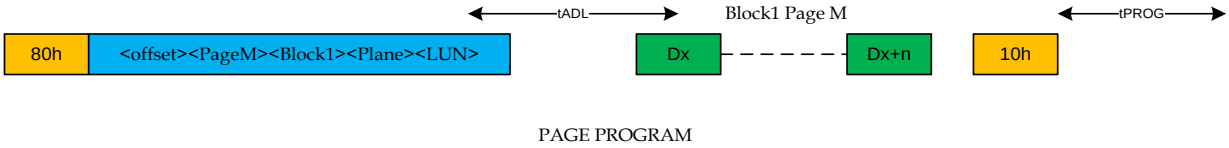


Figure 34: Command Sequence

CMD Structure parameters:

Page Program		**Row Address**	<LUN><Plane><Block><Page>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	0
		No program	0
		Command select	PAGE PROGRAM (0x10)
		Data Buffer pointer	Write Buffer pointer

Figure 35: CMD Structure parameters

5.9.2 Program Page Cache

This can used to write a large bulk of data to NAND Flash. Using the Page Cache increase the page programming throughput.

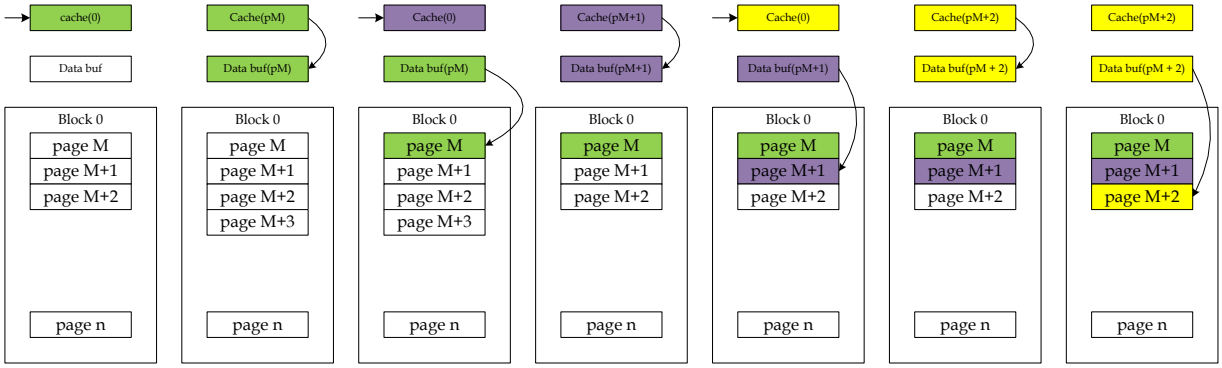


Figure 36: Data Flow Diagram

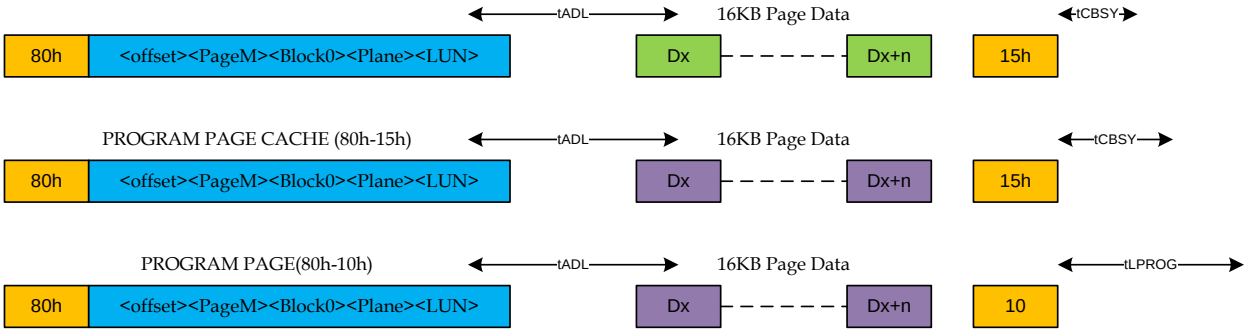


Figure 40: Command Sequence

CMD Structure parameters:

Page Program Cache	0	**Row Address**	<LUN><Plane><Block><PageM>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	0
		No program	0
		Command select	PAGE PROGRAM CACHE (0x11)
		Data Buffer pointer	Write Buffer pointer
Page Program Cache	1	**Row Address**	<LUN><Plane><Block><Page M + 1>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	0
		No program	0
		Command select	PAGE PROGRAM CACHE (0x11)
		Data Buffer pointer	Write Buffer pointer
Page Program	2	**Row Address**	<LUN><Plane><Block><Page M + 2>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	0
		No program	0
		Command select	PAGE PROGRAM (0x10)
		Data Buffer pointer	Write Buffer pointer

Figure 41: CMD Structure parameters

5.10 ERASE Operation

This is a block operation. Only the erased block can be used for page programming. This command will be issued as part of garbage collection operation.

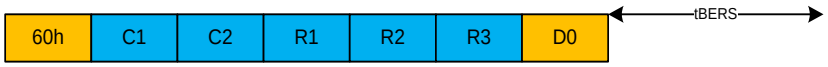


Figure 37: Command Sequence

CMD Structure parameters:

Block Erase		**Row Address**	XXXX_<BLOCK_ADDRESS>XX
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	0
		ECC disable	1
		No data	1
		No program	0
		Command select	BLOCK ERASE (0x05)
		Data Buffer pointer	NA

Figure 38: CMD Structure parameters

5.11 Copy Back Operations

5.11.1 Page Copy-Program

A page will be moved to another page in same block or to another Block. Data is not transfer to Host. If required, data can be read back to host, for ECC verification.

Use Case: This will be used for garbage collection operation. If many pages in a block are invalid, then the valid pages will be coped to another block and that block will be erased.

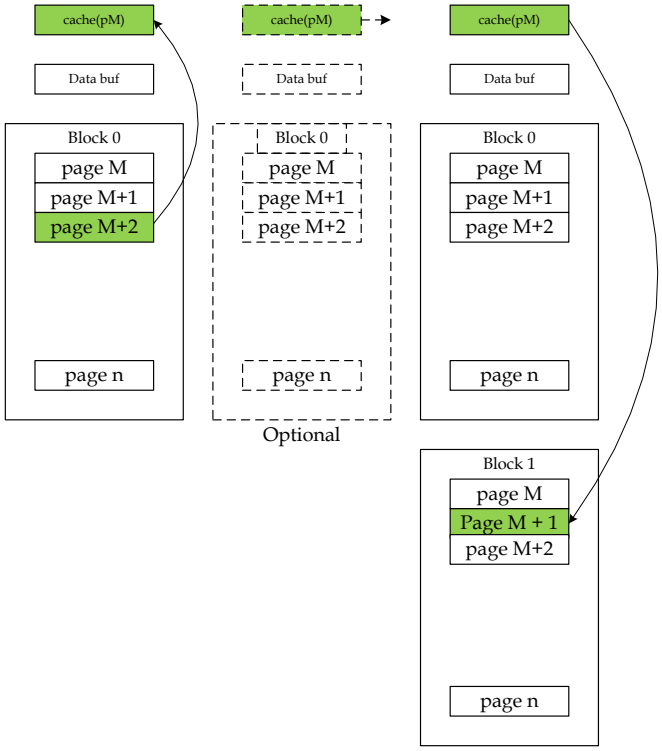


Figure 39: Data Flow

Figure 40: Command Sequence

CMD Structure parameters:

Copy Back Read		**Row Address**	<LUN><Plane><Block0><PageM+2>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	1
		No program	0
		Command select	COPY BACK READ (0x1A)
		Data Buffer pointer	NA
Copy Back Write		**Row Address**	<LUN><Plane><Block1><PageM+3>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	1
		No program	0
		Command select	COPY BACK WRITE (0x08)
		Data Buffer pointer	NA

Figure 41: CMD Structure parameters

5.11.2 Page Update Copy

A 16 KB page is subdivided into four 4KB Sectors. Each Sector can be independently updated.

Use Case: if we have to update only a 4KB sector in a page. The page will be coped to Cache, 4KB sector will be updated and write back to a new erased page (page will all bytes set to 0xff)

Figure 42: Data Flow Diagram

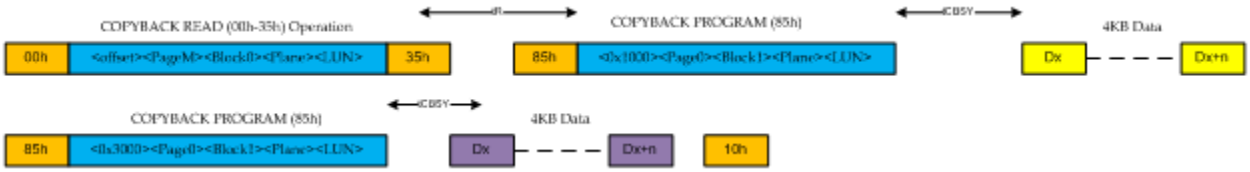


Figure 43: Command Sequence

CMD Structure parameters:

Copy Back Read		**Row Address**	<LUN><Plane><Block0><PageM+2>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	1
		No program	0
		Command select	COPY BACK READ (0x1A)
		Data Buffer pointer	NA
Copy Back Write		**Row Address**	<LUN><Plane><Block1><PageM+3>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	0
		No program	1
		Command select	COPY BACK WRITE (0x08)
		Data Buffer pointer	Write Buffer pointer
Copy Back Write		**Row Address**	<LUN><Plane><Block1><PageM+3>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	0
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	1
		No program	0
		Command select	COPY BACK WRITE (0x08)
		Data Buffer pointer	NA

Figure 44: CMD Structure parameters

5.12 Change write column Operation

The CHANGE WRITE COLUMN (85h) command changes the column address of the selected cache register and enables data input on the last-selected die (LUN).

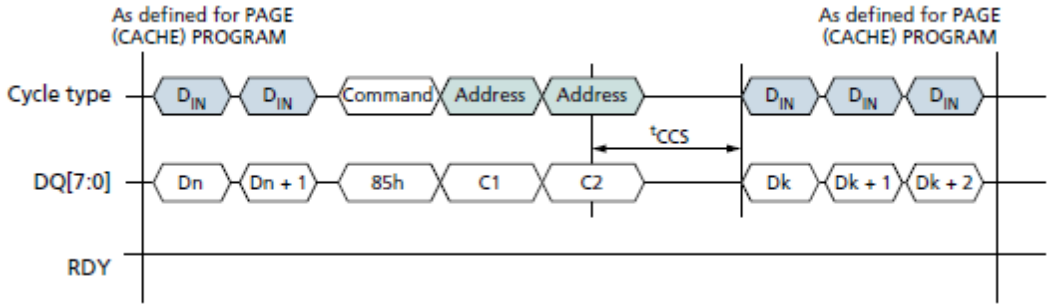


Figure 50: Command Sequence

CMD Structure parameters:

Page Program		**Row Address**	<LUN><Plane><Block><Page>
		Colom Address	0
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	0
		No program	1
		Command select	PAGE PROGRAM (0x10)
		Data Buffer pointer	Write Buffer pointer
Change Write Column		**Row Address**	NA
		Colom Address	<column Address>
		Buffer ID	0
		Channel ID	0
		Length	<Length in bytes>
		Target Select	<Target>
		Direction control	0
		ECC disable	0
		No data	0
		No program	0
		Command select	CHANGE WRITE COLUMN(0xC)
		Data Buffer pointer	Write Buffer pointer

Figure 51: CMD Structure parameters

6 Descriptor Controller register table

Base Address	Field
0X0000000000000000	CSR Descriptor controller
0X0000000000000004	Descriptor table size Register
0X0000000000000008	Reserved

Table 4 Control and Status register Table

6.1 Control and status register (CSR) for descriptor controller

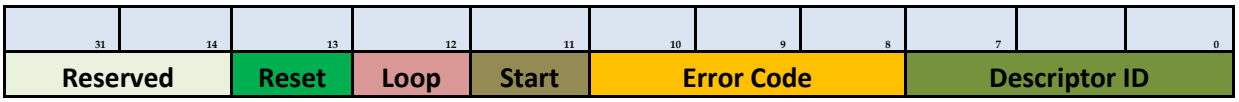


Figure 45: Control and status register

- **Descriptor ID:** This 8 bit field specifies which descriptor in the Descriptor table is the cause for Interrupt.
- **Error Code:** This three bit field specifies the type of error.

Error Code			Description
0	0	0	No IRQ
0	0	1	Write Time Out
0	1	0	Read Time Out
0	1	1	No Error, User Requested Interrupt
1	0	0	No Error, User Requested Interrupt with hold

Table 5 DMA Error Code

- **Start:** Start bit to start DMA transaction for the specified Descriptor controller. Software can write a '1' to start DMA.
- **Loop:** If this bit is set to '0' by the software, Descriptor controller stops after reading and processing the 256 descriptors. Else the Descriptor controller starts again to process from descriptor '0'.
- **Reset:** Logic reset, which will reset all the state machines to IDLE and all pointers to initial value.
- CSR register is updated from FPGA logic on each Descriptor processing completion.

6.2 Descriptor Table Size Register

By default descriptor table size will be 256, which is the maximum value. Software can update this size using the Descriptor table size register, there by the maximum number of descriptors is configurable

31		8	7		0
Reserved			Descriptor Table Size		

Table 6 Descriptor Table Size Register

7 GPIO and MSI Interrupts

7.1 MSI Interrupt Registers

BAR2 MSI Register Interrupt Offset : 0x0700 (64bit addressing)

Writing the following values into the register generates corresponding MSI interrupts:

MSI0 => 0x01

MSI1 => 0x02

MSI2 => 0x04

MSI3 => 0x08

MSI4 => 0x10

MSI5 => 0x20

MSI6 => 0x40

MSI7 => 0x80

A zero to one transition will only generates the interrupts.

7.2 GPIO Interrupt Registers

NAND Descriptor Interrupt : AF19

IO Doorbell Interrupt : AG19

FIFO Interrupt : AG18

Interrupt flags should be cleared from the software with offset 0x10004 (byte addressing).

8 References

a) UNFC hardware reference manual
b) L95B_128_256_512Gb_1Tb_2Tb_Async_Sync_NAND datasheet
c) ONFI 3.2 specifications document