NAND Controller (ONFi Compliant)

NAND Controller specs

Version18: 2023-08-05 (Verilog version)

Version17: 2015-08-30 (VHDL version)

ONFi compliant NAND controller partially implements ONFi standard of NAND flash communication protocol.

Controller interface....2
Instruction Set....3
Status Register....4
Internal Buffers and Index Register....4
JEDEC ID Buffer....4
ONFI Parameter Page Buffer....4
Data Page Buffer....5
Address Buffer....5
Example Waveforms – READ_ID....5

NAND Flash

"***NAND flash*** memory is a type of non-volatile storage technology that does not require power to retain data. An important goal of ***NAND flash*** development has been to reduce the cost per bit and increase maximum chip capacity so that flash memory can compete with magnetic storage devices like hard disks." WhatIs.com

ONFI

"The Open NAND Flash Interface (ONFI) is an industry Workgroup made up of more than 100 companies that build, design-in, or enable NAND Flash memory. We're dedicated to simplifying NAND Flash integration into consumer electronic products, computing platforms, and any other application that requires solid state mass storage. We define standardized component-level interface specifications as well as connector and module form factor specifications for NAND Flash." Onfi.org

Controller Interface

Avalon MM ports

Port name	Bit width	Direction	Purpose
CLK	1	IN	System clock
$\overline{RESET}$	1	IN	Reset input. Active low
READDATA	32	OUT	Data output port
WRITEDATA	32	IN	Data/command input port
ADDRESS	2	IN	Address index
$\overline{PREAD}$	1	IN	Read strobe
$\overline{PWRITE}$	1	IN	Write strobe

NAND interface ports

Port name	Bit width	Direction	Purpose
NAND_CLE	1	IN	Command latch enable
NAND_ALE	1	IN	Address latch enable
$\overline{NAND_{WE}}$	1	IN	Write enable. Active low
$\overline{NAND_{\wp}}$	1	IN	Write protect. Active low
$\overline{NAND_{CE}}$	1	IN	Chip enable. Active low
$\overline{NAND_{\Re}}$	1	IN	Read enable
$NAND_R\overline{B}$	1	OUT	Ready/busy. Is low when busy
NAND_DATA	16	INOUT	Data/command bus. Upper 8 bits are ignored for x8 NAND flash chips.

Controller interface

Controller's interface to custom logic

Signal	Type/Width	Usage
CLK	IN:1	Clock input.

Enable#	IN:1	Component enable signal. Active LOW.
Reset#	IN:1	Component reset signal. Active LOW.
Busy	OUT:1	Indicates whether the controller is busy (1) or idle (0).
Activate	IN:1	Strobe this input to 1 will instruct the controller to execute command fed to cmd_in port.
Cmd_in	IN:8	Command input.
Data_in	IN:8	Data input.
Data_out	OUT:8	Data output.

CLK – This is a clock input. Be aware of the need to define clock cycle duration in onfi_package.v, as it is used for delay times calculation.

Enable# - Controller is inactive when this input is set to low. Avoid setting it to low when controller's 'Busy' output is high.

Reset# - When set to '0' resets the controller. All internal signals are reset to their defaults.

Busy – Indicates whether the controller is in the middle of something. All commands would be ignored when Busy is set to '1'.

Activate – Setting this input to '1' instructs the controller to read the command on Cmd_in and execute it (if it is a valid command, otherwise no action is taken).

Cmd_in – 8 bit command input register. Make sure the command is fed in prior to strobing 'Activate'.

Data_in – 8 bit data input register. Just as in case of Cmd_in, make sure the data is fed in prior to strobing 'Activate'.

Data_out – 8 bit data output register. Only read from it when 'Busy' is '0', otherwise the value is undefined.

Controller's interface to NAND Flash

This controller implements the standard NAND Flash interface defined in ONFI Specification Rev.4. It supports both x8 and x16 interfaces. Bits 8 to 15 of the NAND data/command bus are ignored when connected to x8 chip. You do not have to tell the controller which chip it is connected to, as it determines this information automatically when reading ONFI Parameter Page. You have to tell the controller to read the parameter page though before issuing page read/write commands.

Instruction Set

Instruction	Data_in	Data_out	Function
0x01 M_RESET			Controller reset. Resets all internal signals to their defaults. This does not contact the NAND Flash.
0x04 M_NAND_RESE			Performs NAND reset sequence.

T			
0x05 M_NAND_READ_PARAM_PAGE			Reads ONFI Parameter Page into the internal buffer.
0x06 M_NAND_READ_ID			Reads JEDEC ID into the internal buffer.
0x07 M_NAND_BLOCK_ERASE			Performs NAND Block Erase sequence. The address of the block MUST be set prior to issuing this command.
0x08 M_NAND_READ_STATUS		**Status**	Performs NAND Read Status operation and puts the result on Data_out.
0x09 M_NAND_READ			Reads a page into the internal buffer. Page's address should be set prior to issuing this command. Note – you should specify 5 bytes of the address regardless of the ADDRESS_CYCLES value in the parameter page, the controller will issue appropriate amount of address cycles.
0x0C (12d) M_NAND_PAGE_PROGRAM			Programs one page with the content of the internal page buffer. The address should be set appropriately prior to issuing this command.
0x0D (13d) MI_GET_STATUS		**Controller status**	Returns the content of the controller's 8-bit status register.
0x0E (14d) MI_CHIP_ENABLE	**CE#**		Sets CE# pin of the chip to LOW toEnable the chip. If there are several CE# lines, you can specify which CE# should be set to LOW by selecting the CE line in the Data_in register.
0x0F (15d) MI_CHIP_DISABLE	**CE#**		Sets CE# pin of the chip to HIGH to Disable the chip. If there are several CE# lines, you can specify which CE# should be set to LOW by selecting the CE line in the Data_in register.
0x10 (16d) MI_WRITE_PROTECT	**WP#**		Sets the WP# pin of the chip to LOW (enables write protection). This pins is set to LOW upon controller's reset.
0x11 (17d) MI_WRITE_ENABLE	**WP#**		Sets the WP# pin of the chip to HIGH (disables write protection).
0x12 (18d) MI_RESET_INDEX			Resets internal buffer index to 0. Although, the controller has three internal buffers, it uses single index register for all of them.
0x13 (19d) MI_GET_ID_BYTE		**JEDEC ID byte**	Returns the byte pointed by index register from the JEDEC ID buffer and increments the register. Check status register for OUT_OF_BOUNDS error (if index > 4) if the returned byte is 0. Basically, you should reset the index register before reading from JEDEC ID buffer. [1]
0x14 (20d) MI_GET_PARAM_PAGE_BYTE		**Parameter Page byte**	Returns the byte pointed by index register from the ONFI Parameter Page internal buffer and increments the register. Check status register for OUT_OF_BOUNDS error (if index > 255) if the returned byte is 0. Basically, you should reset the index register before reading from Parameter Page

			buffer. [1]
0x15 (21d) MI_GET_PARAM_PAGE_BYTE		**Data Page byte**	Returns the byte pointed by index register from the Data Page internal buffer and increments the register. Check status register for OUT_OF_BOUNDS error (if index > data_bytes_per_page + oob_bytes_per_page) if the returned byte is 0. Basically, you should reset the index register before reading from Data Page buffer. [1]
0x16 (22d) MI_SET_DATA_PAGE_BYTE	**Byte**		Writes single byte into the Data Page internal buffer at position pointed by the index register. Make sure you reset the index register prior to starting filling the page buffer. [1]
0x17 (23d) MI_GET_CURRENT_ADDRESS_BYTE		**Address byte**	Returns the byte pointed by the index register from the Address internal buffer. Don't forget to reset the index register before reading address bytes. [1]
0x18 (24) MI_SET_CURRENT_ADDRESS_BYTE	**Byte**		Writes single byte into the Address internal buffer at position pointed by the index register.[1]
0x19 (25) MI_BYPASS_ADDRESS	**Address**		Send address byte directly to NAND chip (**This is useful when sending vendor specific commands (e.g. accessing OTP area on different chips) or commands that are not yet supported by this controller.)**
0x1A (26) MI_BYPASS_COMMAND	**Cmd**		Send command byte directly to NAND chip (**This is useful when sending vendor specific commands (e.g. accessing OTP area on different chips) or commands that are not yet supported by this controller.**)
0x1B (27) MI_BYPASS_DATA_WR	**Data**		Send data byte directly to NAND chip (for vendor specific commands). Data is written/read to/from the NAND chip without modifying the page_data buffer.
0x1C (28) MI_BYPASS_DATA_RD		**Data**	Read data byte directly from NAND chip (for vendor specific commands). Data is written/read to/from the NAND chip without modifying the page_data buffer.
0x1D (29) M_SET_PAGESIZE	**Size in Bytes**		Set the PageSize, this defines the actually to be read page size in bytes
0x1E (30) M_GET_PAGESIZE		**Size in Bytes**	Get the PageSize, this should deliver the maximum possible pagesize before the parameter page has been read and the actual pagesize after the parameter page has been read or after the pagesize has been set with the

- **[1] Operation increments the index register or resets it to 0 if the register points out of the bounds of the related register/buffer.**

Status Register

Bit position	Meaning
0	1-the NAND flash chip is ONFI compliant. '0' after reading the ONFI Parameter Page means that the chip is not ONFI compliant and the controller does not know how to handle it.

1	Indicates whether the chip is x8 (‘0’) or x16 (‘1’).
2	Indicates whether the chip is enabled (‘1’). Note – the controller does not pay attention to this bit!
3	Indicates whether write protection is enabled (whether WP# is set to LOW).
4	‘1’ means that the index register pointed beyond buffer bounds during the last operation on internal buffers.
5	Reserved
6	Reserved
7	Reserved

Internal Buffers and Index Register

The ONFI Compliant NAND Controller utilizes three internal buffers and a common index register. NOTE: Do not forget to reset the index register before starting to read from any buffer, so that the first byte read would be the first byte in buffer. When the index register reaches a value outside the buffer being accessed it is reset to 0, also 0 is returned as operation’s result as well as bit 4 of the status register gets set to ‘1’.

JEDEC ID Buffer

The JEDEC ID Buffer is a 5 bytes buffer that contains chip’s ID. It is only provided for convenience as a way to speed up READ ID procedures. You only have to read the ID from flash once and then access this buffer if needed, which may be up to 12 times faster (depends on your clock settings).

ONFI Parameter Page Buffer

256 bytes buffer that holds the ONFI Parameter Page (once it has been read).

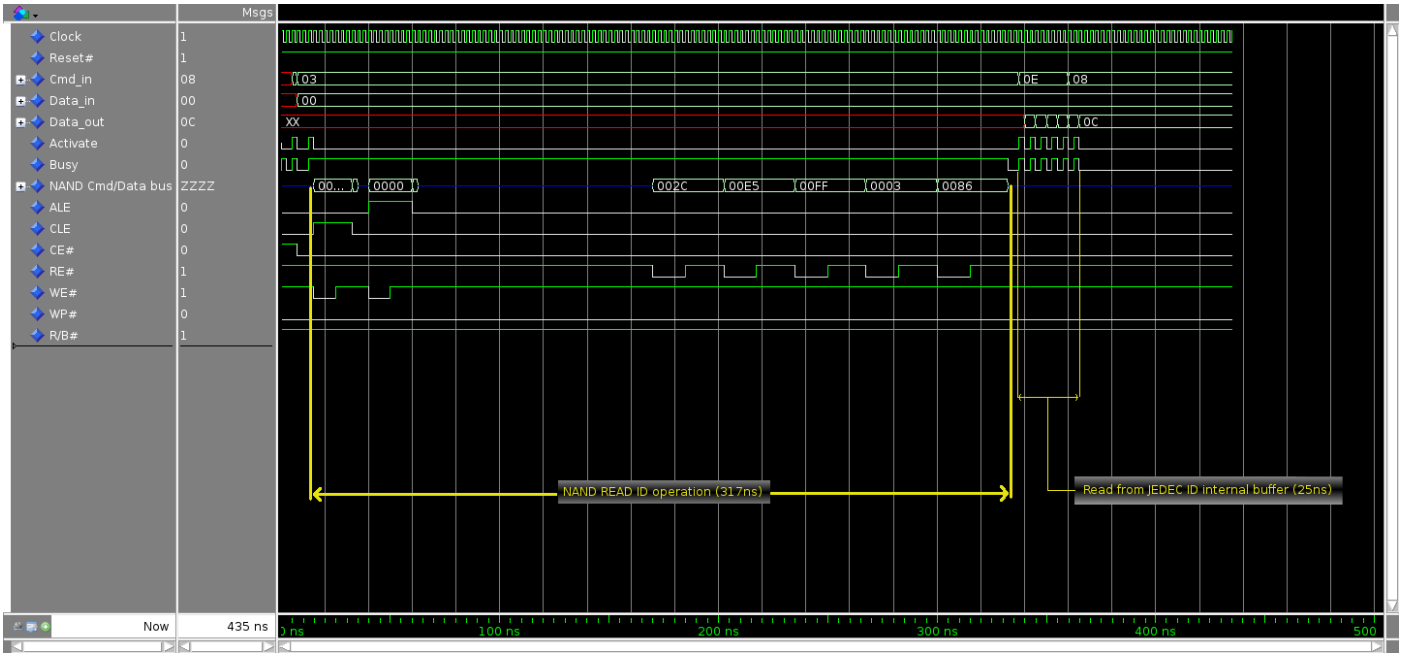
Data Page Buffer

This is the largest buffer – 8628 bytes, enough to have the largest page fit in.

Address Buffer

This buffer is 5 bytes long and contains the address of the page currently in use.

Example Waveforms – READ_ID



- Instruction 0x03 – READ_ID instructs the controller to read JEDEC ID into internal buffer.
- Instruction 0x0e – READ_ID_BYTE is executed 5 times to get all bytes of the ID stored in the internal buffer.
- Instruction 0x08 – GET_STATUS reads the content of the controller's status register, which in this case is 0x0c, meaning the chip is enabled and write protection is on. As you see, bit 0 is not set in this particular case as we have not read ONFI Parameter Page during this simulation.