Datasheet

ONFI NAND Total IP Solution

Arasan Chip Systems Inc.
2010 North First Street, Suite #510, San Jose, CA 95131
Ph: 408-282-1600
Fax: 408-282-7800
www.arasan.com

Disclaimer

This document is written in good faith with the intent to assist the readers in the use of the product. Circuit diagrams and other information relating to Arasan Chip Systems' products are included as a means of illustrating typical applications. Although the information has been checked and is believed to be accurate, no responsibility is assumed for inaccuracies. Information contained in this document is subject to continuous improvement and development.

Arasan Chip Systems' products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of Arasan Chip Systems Inc. will be fully at the risk of the customer.

Arasan Chip Systems Inc. disclaims and excludes any and all warranties, including, without limitation, any and all implied warranties of merchantability, fitness for a particular purpose, title, and infringement and the like, and any and all warranties arising from any course or dealing or usage of trade.

This document may not be copied, reproduced, or transmitted to others in any manner. Nor may any use of information in this document be made, except for the specific purposes for which it is transmitted to the recipient, without the prior written consent of Arasan Chip Systems Inc. This specification is subject to change at any time without notice. Arasan Chip Systems Inc. is not responsible for any errors contained herein.

In no event shall Arasan Chip Systems Inc. be liable for any direct, indirect, incidental, special, punitive, or consequential damages; or for loss of data, profits, savings or revenues of any kind; regardless of the form of action, whether based on contract; tort; negligence of Arasan Chip Systems Inc or others; strict liability; breach of warranty; or otherwise; whether or not any remedy of buyers is held to have failed of its essential purpose, and whether or not Arasan Chip Systems Inc. has been advised of the possibility of such damages.

Restricted Rights

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Copyright Notice

Questions or comments may be directed to:

Arasan Chip Systems Inc.
2010 North First Street, Suite 510
San Jose, CA 95131
Ph: 408-282-1600
Fax: 408-282-7800
Email: sales@arasan.com

Contents

Tables

Figures

1 Arasan's Total IP Solution

Arasan provides a Total IP Solution, which encompasses all aspects of IP development and integration, including analog and digital IP cores, verification IP, software stacks & drivers, and hardware validation platforms. Benefits of Total IP Solution:

- Seamless integration from PHY to Software
- Assured compliance across all components
- Single point of support
- Easiest acquisition process (one licensing source)
- Lowest overall cost including cost of integration
- Lowest risk for fast time to market

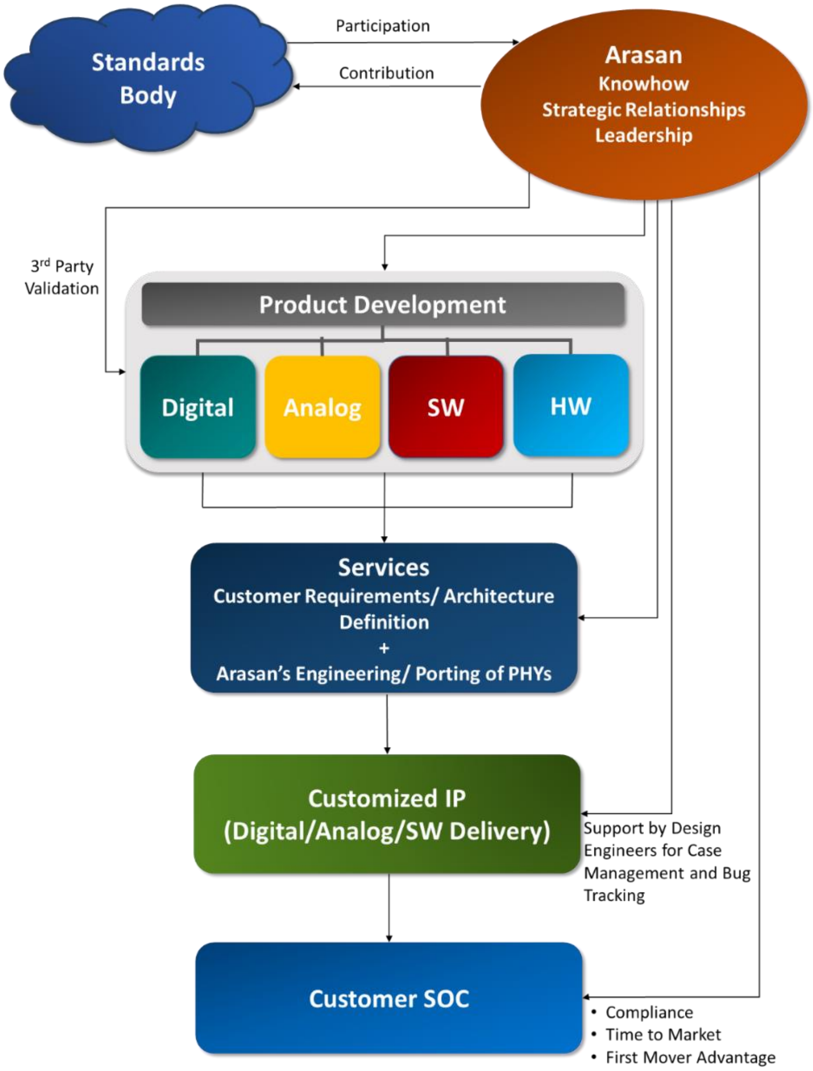

Figure 1: Arasan's Total IP Solution

2 ONFI 4.0 Controller

2.1 Overview

The NAND Flash landscape is changing and the Arasan NAND Flash Controller IP Core is changing in accordance with it. New applications are emerging and innovative IP solutions are needed to keep pace. NAND Flash is being incorporated into all types of products including Portable memory drives, Media players, Digital cameras, Smart phones, eBook Readers, Tablets, Digital TVs, Digital camcorders, PCs, and so on. Arasan is in the perfect position to give you what you need.

The Arasan NAND Flash Controller IP Core is a full featured, easy to use, synthesizable core, easily integrated into any SoC or FPGA development. Designed to support SLC, MLC and TLC flash memories, it is flexible in use and easy in implementation. The controller works with any suitable NAND Flash memory device up to 1024Gb from leading memory providers - Micron, Samsung, Toshiba and Hynix. The IP core includes a host of configuration options from page size to bank selects. The controller offers Hamming Code (1-Bit error correction and 2-Bit error detection) and BCH (option for 4-, 8-, 12-, up to 32-Bit error correction) Error Code Correction (ECC) for optimized performance and reliability. Additional features include the capability to boot from flash.

The IP core supports the Open NAND Flash Interface Working Group (ONFI) 1.0, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 3.2 and 4.0 standards. It can also support a variety of host bus interfaces for easy adoption into any design architecture - AHB, APB, OCP, 8051 or custom buses. The slave AHB IP supports an external DMA interface where the master AHB incorporates an internal DMA controller.

The Arasan NAND Flash Controller IP Cores are delivered in Verilog RTL that can be implemented in an ASIC or FPGA. They are fully tested with vendor models and hardware is tested with FPGA's. The core includes RTL code, test scripts and a test environment for complete simulation and verification.

2.2 Features

2.2.1 General

The following features support for the Arasan ONFI 4.0 Controller.

- Flash devices up to 1024Gb
- NAND Flash memories from Micron, Samsung, Toshiba and Hynix
- Boot mode support
- LUN Set/ LUN Get feature support
- All mandatory commands and selected optional commands
- Full access to spare area
- Speed ranging from 40MB/s to 800MT/s to allow applications to balance performance and power
- Multi LUN/DIE Operations
- Small Data Move

- Change Row Address
- Reset LUN
- EZ - NAND devices
- Chip_en pin reduction mechanism
- ODT Configure
- On-die termination
- Supports Interleaving Operations:
  - Page Program Interleaving
  - Copy back Program Interleaving
  - Block Erase Interleaving
  - Read Interleaving
  - Cache Interleaving

2.2.2 Configuration

- Page Size - 512B, 2KB, 4KB, 8KB, 16KB
- Bank/chip select options
- Programmable timing
- Address cycles - 4, 5
- ECC enable, disable
- RAM size - 1KB, 2KB, and 4KB
- Supports parallel connection of two 8-bit flash devices

2.2.3 ECC

- SLC - Hamming Code
  - 1-Bit error correction for 512bytes
  - 2-Bit error correction for 512bytes
- MLC - BCH
  - Standard support - 4, 8-Bit error correction for 512bytes
  - Additional support - up to 32-Bit error correction either for 512bytes or 1024bytes

2.2.4 Data Interface

- ONFI 1.0, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 3.2 and 4.0 Compliant
- Supports SDR modes [0-5]
- Supports NV- DDR modes [0 -5]
- Supports NV- DDR2 modes [0 -10]
- Supports NV- DDR3 modes [0 -10]
- 8-bit data bus width support for SDR, NV-DDR, NV-DDR2 and NV-DDR3
- 16-bit data bus width support only for SDR mode
- Supports independent data bus

2.2.5 System Interface

- AHB - AMBA 2.0 Compliant
- AXI Interface (Optional)
- PIO Mode
- Slave DMA mode (Optional)
- Master DMA mode

2.3 Architecture

This section describes the ONFI 4.0 Controller architecture and the description of internal blocks in detail. The IP core supports the Open NAND Flash Interface Working Group (ONFI) 4.0 standard and is backwards compatible. It uses differential signaling on the clock and data lines and clocks at any frequency up to 400MHz.

2.3.1 Functional Block Diagram

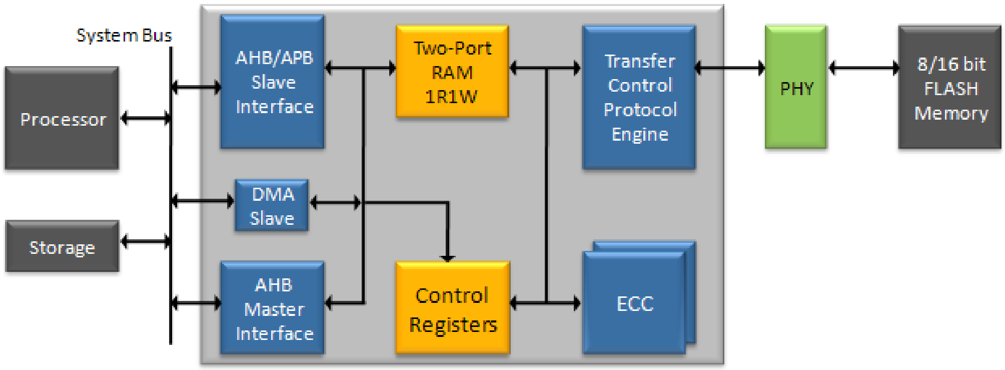

Figure 2: ONFI 4.0 Controller Functional Block Diagram

2.3.2 Functional Block Diagram Description

2.3.2.1 NAND Flash Interface

The NAND Flash Interface handles all the command, address and data sequences and manages all the hardware protocols. It is ONFI 1.0, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1, 3.2 and 4.0 compliant and provides an 8-bit or 16-bit interface to the flash memories. The interface supports a maximum of 1024 Gb of NAND flash memory. All timing modes (0-5) are supported for SDR, NV-DDR and Timing modes (0-10) for NV-DDR2 and NV-DDR3.

2.3.2.2 AHB Slave Interface

The AHB Slave block consists of the operational registers. A processor connecting to the custom interface can control the operation of the NAND Flash controller through the NAND Flash control registers. Read/write operations of the flash memory can be performed through NAND flash interface.

2.3.2.3 AHB Master Interface

NAND Flash controller acts as a master during MDMA mode of transaction. The AHB master interface places control signals in AHB Bus depending upon the FIFO status. During Write transaction, AHB master interface reads data from system memory and stores into the FIFO. During Read transaction, reads data from FIFO and stores into system memory. AHB master interface asserts DMA interrupt when DMA buffer boundary is reached. The AHB master interface can be used to transfer boot code from the NAND flash memory to the system memory during system power-up.

2.3.2.4 Two Port RAM (1R1W)

This block has handshake logic to communicate with the AHB interface and on the other side communicates with the Flash Interface. Typical RAM size is 256x32, to support block size of 512 bytes. The FIFO depth is configurable.

2.3.2.5 ECC

The ECC module provides error detection and correction support for SLC Flash memory as well as the MLC and TLC Flash memory. For SLC Flash, Hamming Code is being used for 1 bit error correction and 2 bit error detection. BCH Code, capable of up to 32-Bit error correction, is used for MLC and TLC Flash devices. An optional pipeline stage in the BCH decoder can be enabled for maximum performance.

2.3.2.6 Control Registers

The host processor controls the configuration and operation of the NAND Flash Controller through the Control Registers. Configuration includes the set up time (tCCS, tDQSQ, tDS), memory configuration (address, page size, packet size, packet count), timing modes (SDR, NV-DDR, NV-DDR2 and NV-DDR3), and so on. The Control Registers also provide operating status such as Busy and Data Ready signals.

2.3.2.7 Slave DMA Interface (Optional)

This interface is used to perform a SDMA mode of Write/ Read data transfer. Slave DMA Interface provides DMA Request, DMA Single and DMA Last and gets acknowledged by DMA Acknowledge and DMA Finish. DMA Request is asserted until the required amount of data is transferred based on the DMA Transfer count. For the generated DMA Request Slave DMA Controller provides write data for write transfer or reads the data from FIFO for read transfer.

2.4 ONFI 4.0 Controller Interface

This section contains description and the direction of the pins from interfaces such as AHB Master, AHB Slave, AXI Master, AXI Slave, Nand_Flash, SDMA, RAM and Environment signals.

The NAND Flash controller has eight main interface groups:

- AHB Master Interface
- AHB Slave Interface
- ONFI4.0 PHY Interface
- Slave DMA Interface (Optional)
- AXI Master Interface (Optional)
- AXI Slave Interface (Optional)
- RAM Interface
- Clock and Reset Signals

Table 1: AHB Master Interface Signals

Pin	Direction	Description
m_hbusreq	Output	AHB Bus Request
m_hgrant	Input	AHB Bus Grant
m_haddr[31:0]	Output	Address Bus
m_hwdata[31:0]	Output	Write Data Bus
m_hrdata[31:0]	Input	Read Data Bus
m_hwrite	Output	Write or Read Direction Indication
m_hsize[2:0]	Output	Size(Byte, Half Word or Word)
m_hburst[2:0]	Output	Burst Size
m_htrans[1:0]	Output	Transfer type
m_hready	Input	Slave Ready
m_hresp[1:0]	Input	Transfer Response
boot_enable	Input	Reads Boot Code from Flash

Table 2: AHB Slave Interface Signals

Pin	Direction	Description
ahb_hsel	Input	Slave Select
ahb_haddr[31:0]	Input	Address Bus (Byte Addresses)
ahb_hwdata[31:0]	Input	Write Data Bus
ahb_hrdata[31:0]	Output	Read Data Bus
ahb_hwrite	Input	Write or Read Direction Indication
ahb_hburst	Input	Burst (Single, Incrementing, wrapping)
ahb_hsize[2:0]	Input	Size (Byte, Half Word or Word)
ahb_htrans[1:0]	Input	Transfer Type
ahb_hready_in	Input	Global Ready

Pin	Direction	Description
ahb_hready	Output	Slave Ready
ahb_hresp[1:0]	Output	Transfer Response
int_to_arm	Output	Interrupt to the ARM

Table 3: ONFI 4.0 PHY Interface

Pin	Direction	Description
CLK_FLASH	Input	Source clock to ONFI 4.0 Digital Controller
R_B_IN	Input	Ready Busy from External Flash Interface
DDR_DATA_VALID	Input	RX Data Valid for DDR mode
DATA_IN[7:0]	Input	RX Data for SDR and DDR mode
DATA_8_EN	Output	Enables SDR DATA, DDR command and Address
DATA_OUT_EN	Output	Enables Data Direction
DQS_OUT_EN	Output	Enables DQS Direction
DATA_OUT[15:0]	Output	TX Data/Address/command for SDR/NV-DDR/NV-DDR2/ NV-DDR3 modes
DQS_VALID	Output	Enables DQS_t & DQS_c
DQS_MASK	Output	Dqs Mask for NV-DDR2 / NV-DDR3 mode
ALE_OUT	Output	Address Latch Enable from Controller
CLE_OUT	Output	Command Latch Enable from Controller
CE_OUT	Output	Chip Enable from Controller
RE_OUT	Output	Read Enable from Controller
WE_out	Output	Write Enable from Controller
DDR_RE_EN	Output	Read Strobe Enable for DDR Mode
DDR_RE_MASK	Output	Read enable mask for NV-DDR2 / NV-DDR3 mode
PLL_EN	Output	Enables the PLL
PLL_FREQ[4:0]	Output	Changes the Frequency of the PLL Output (CLK2X)
MODE[1:0]	Output	SDR/NV-DDR / NV-DDR2/ NV-DDR3
V18_EN	Output	1.8V Enable
V12_EN	Output	1.2V Enable
DQS_BUFF_SEL[3:0]	Output	Configurable value for delaying Dqs strobe
DRIVE_STRENGTH[1:0]	Output	Programmable Drive Strength

Table 4: Slave DMA Interface

Pin	Direction	Description
dma_ack	Input	DMA Acknowledge This is asserted after the data phase of the last transfer in the current transaction (single or burst) to the peripheral has completed. It forms a handshaking loop with dma_req and remains asserted until the

Pin	Direction	Description
		peripheral de-asserts dma_req (deasserted one HCLK cycle later).
dma_finish	Input	DMA Finish DMA Controller block transfer complete signal. The DMAC asserts dma_finish in order to signal block completion. This uses the same timing as dma_ack and forms a handshaking loop with dma_req.
dma_req	Output	DMA Request Transaction request from peripheral. A rising edge on dma_req initiates a transaction request. The type of transaction – single or burst – is qualified by dma_single. Once dma_req is asserted, it must remain asserted until dma_ack is asserted. When the peripheral that is driving dma_req determines that dma_ack is asserted, it must de-assert dma_req.
dma_last	Output	DMA Last Last transaction in block. When the peripheral is the flow controller, it asserts dma_last on the same cycle as dma_req is asserted in order to signal that this transaction request is the last in the block; the block transfer is completed after this transaction is complete. If dma_single is high in the same cycle, then the last transaction is a single transaction. If dma_single is low in the same cycle, then the last transaction is a burst transaction.
dma_single	Output	DMA Single Single or burst transaction request. If dma_single is de-asserted in the same clock cycle as a rising edge on dma_req, a burst transaction is requested by the peripheral. If asserted, the peripheral requests a single transaction.

Table 5: AXI Master Pin Interface (Optional)

Pin	Direction	Description
axi_clk I	Input	AXI Clock. All signals are sampled on the rising edge of this clock.
axi_reset_n	Input	Active Low Reset
aximst_awid	Output[3:0]	Write address ID. This signal is the identification tag for the write address group of signals.
aximst_awaddr	Output [31:0]	Write address. The write address bus gives the address of the first transfer in a write burst transaction. The associated control signals are used to

Pin	Direction	Description
		determine the addresses of the remaining transfers in the burst.
aximst_awlen	Output [3:0]	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
aximst_awsize	Output [2:0]	Burst size. This signal indicates the size of each transfer in the burst. Byte lane strobes indicate exactly which byte lanes to update.
aximst_awburst	Output[1:0]	Burst type. The burst type, coupled with the size information, details on how the address for each transfer within the burst is calculated.
aximst_awvalid	Output	Write address valid. This signal indicates that valid write address and control information are available: 1 = address and control information available; 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, aximst_awready, goes HIGH.
aximst_awready	Input	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: 1 = slave ready; 0 = slave not ready.
aximst_wid	Output[3:0]	Write ID tag. This signal is the ID tag of the write data transfer. The aximst_wid value must match the aximst_awid value of the write transaction.
aximst_wdata	Output[31:0]	Write data. The write data bus is 32 bits wide
aximst_wstrb	Output[3:0]	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus. Therefore, aximst_wstrb[n] corresponds to aximst_wdata[(8 x n) + 7:(8 x n)].
aximst_wlast	Output	Write Last. This signal indicates the last transfer in a write burst.
aximst_wvalid	Output	Write valid. This signal indicates that valid write data and strobes are available: 1 = write data and strobes available 0 = write data and strobes not available
aximst_wready	Input	Write ready. This signal indicates that the slave can accept the write data: 1 = slave ready; 0 = slave not ready.
aximst_bid]	Input[3:0	Response ID. The identification tag of the write response. The aximst_bid value must match the

Pin	Direction	Description
		aximst_awid value of the write transaction to which the slave is responding.
aximst_bresp	Input[1:0]	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
aximst_bvalid	Input	Write response valid. This signal indicates that a valid write response is available: 1 = write response available; 0 = write response not available.
aximst_bready	Output	Response ready. This signal indicates that the master can accept the response information. 1 = master ready; 0 = master not ready
aximst_arid	Output[3:0]	Read address ID. This signal is the identification tag for the read address group of signals.
aximst_araddr	Output[31:0]	Read address. The read address bus gives the initial address of a read burst transaction. Only the start address of the burst is provided and the control signals that are issued alongside the address detail how the address is calculated for the remaining transfers in the burst
aximst_arlen	Output[3:0]	Burst length. The burst length gives the exact number of transfers in a burst.
aximst_arsize	Output[2:0]	Burst size. This signal indicates the size of each transfer in the burst.
aximst_arburst	Output[1:0]	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
aximst_arvalid	Output	Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and will remain stable until the address acknowledge signal, aximst_arready, is high: 1 - address and control information are valid. 0 - address and control information are not valid.
aximst_arready	Input	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signal. 1 - slave ready; 0 - slave not ready
aximst_rid	Input[3:0]	Read ID tag. This signal is the ID tag of the read data group of signals. The RID value is generated by the slave and must match the aximst_arid value of the read transaction to which it is responding.
aximst_rdata	Input[31:0]	Read data. The read data bus is 32

Pin	Direction	Description
aximst_rresp	Input[1:0]	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR and DECERR.
aximst_rlast	Input	Read Last. This signal indicates the last transfer in a read burst.
aximst_rvalid	Input	Read valid. This signal indicates that the required read data is available and the read transfer can complete: 1 - read data available 0 - read data not available
aximst_rready	Output	Read ready. This signal indicates that the master can accept the read data and response information: 1 - master ready 0 - master not ready

Table 6: AXI Slave Pin Interface (Optional)

Pin	Direction	Description
axi_clk	Input	AXI Clock. All signals are sampled on the rising edge of this clock.
axi_reset_n	Input	Active Low Reset.
axislave_awid	Input[3:0]	Write address ID. This signal is the identification tag for the write address group of signals.
axislave_awaddr	Input[31:0]	Write address. The write address bus gives the address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst.
axislave_awlen	Input[3:0]	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
axislave_awsize	Input[2:0]	Burst size. This signal indicates the size of each transfer in the burst. Byte lane strobes indicate exactly which byte lanes to update.
axislave_awburst	Input[1:0]	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
axislave_awvalid	Input	Write address valid. This signal indicates that valid write address and control information are available: 1 = address and control information available; 0 = address and control information not available.

Pin	Direction	Description
		The address and control information remain stable until the address acknowledge signal, aximst_awready goes HIGH.
axislave_awready	Output	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: 1 = slave ready; 0 = slave not ready.
axislave_wid	Input [3:0]	Write ID tag. This signal is the ID tag of the write data transfer. The aximst_wid value must match the aximst_awid value of the write transaction.
axislave_wdata	Input[31:0]	Write data. The write data is 32 bits wide
axislave_wstrb	Input[3:0]	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus. Therefore, aximst_wstrb[n] corresponds to aximst_wdata[(8 x n) + 7:(8 x n)].
axislave_wlast	Input	Write Last. This signal indicates the last transfer in a write burst.
axislave_wvalid	Input	Write valid. This signal indicates that valid write data and strobes are available: 1 = write data and strobes available 0 = write data and strobes not available
axislave_wready	Output	Write ready. This signal indicates that the slave can accept the write data: 1 = slave ready 0 = slave not ready
axislave_bid	Output[3:0]	Response ID. The identification tag of the write response. The aximst_bid value must match the aximst_awid value of the write transaction to which the slave is responding.
axislave_bresp	Output[1:0]	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR and DECERR.
axislave_bvalid	Output	Write response valid. This signal indicates that a valid write response is available: 1 = write response available 0 = write response not available
axislave_bready	Input	Response ready. This signal indicates that the master can accept the response information. 1 = master ready 0 = master not ready

Pin	Direction	Description
axislave_arid	Input[3:0]	Read address ID. This signal is the identification tag for the read address group of signals.
axislave_araddr	Input[31:0]	Read address. The read address bus gives the initial address of a read burst transaction. Only the start address of the burst is provided and the control signals that are issued alongside the address detail how the address is calculated for the remaining transfers in the burst.
axislave_arlen	Input[3:0]	Burst length. The burst length gives the exact number of transfers in a burst.
axislave_arsize	Input[2:0]	Burst size. This signal indicates the size of each transfer in the burst.
axislave_arburst	Input[1:0]	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
axislave_arvalid	Input	Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and will remain stable until the address acknowledge signal, aximst_arready, is high 1 - address and control information are valid; 0 - address and control information are not valid.
axislave_arready	Output	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signal. 1 - slave ready; 0 - slave not ready
axislave_rdata	Output[31:0]	Read data. The read data bus is 32 bits wide.
axislave_rresp	Output[1:0]	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, DECERR.
axislave_rlast	Output	Read Last. This signal indicates the last transfer in a read burst.
axislave_rvalid	Output	Read valid. This signal indicates that the required read data is available and the read transfer can complete: 1 - read data available; 0 - read data not available
axislave_rready	Input	Read ready. This signal indicates that the master can accept the read data and response information: 1 - master ready; 0 - master not ready

Table 7: RAM Interface

Pin	Direction	Description
data_to_sys [31:0]	Input	AHB domain data output

Pin	Direction	Description
data_to_flash_from_fifo [31:0]	Input	Flash domain data output
ram_addr_a	Output	Address bus for AHB domain
wr_data_a [31:0]	Output	AHB domain data input
cen_a	Output	Active high chip select.(AHB domain)
wren_a	Output	Active high write enable signal.(AHB domain)
ram_addr_b	Output	Address bus for Flash domain
wr_data_b [31:0]	Output	Flash domain data input
cen_b	Output	Active high chip select.(Flash domain)
wren_b	Output	Active high write enable signal.(Flash domain)

Table 8: Clock and Reset Signals

Pin	Direction	Description
clk_sys	Input	System clock
clk_flash	Input	Source clock to ONFI 4.0 Digital Controller
rst_n	Input	Active low Asynchronous power on reset from external environment and it is synchronized inside the IP. The synchronized reset is used to reset all the flops in FLASH, AHB clock domain
Scan_mode	Input	Active Low. 1’b0 - Bypass all reset signals generated internally 1’b1 - POR will be asserted Asynchronously

2.5 Configurable Features

Table 9: ONFI Feature and Compile Time Option

Feature	Compile Time Options
RAM configurability	Configurable

Table 10: ONFI Feature and Run Compile Time Option

Feature	Run Time Options
ECC_ON_OFF	ECC can be enabled or disabled by the software
nfc_bch _mode	BCH mode value is programmed by the software

3 ONFI 3.2 Controller

3.1 Overview

The Arasan NAND Flash Controller IP Core is a full featured, easy to use, synthesizable core, easily integrated into any SoC or FPGA development. Designed to support SLC, MLC and TLC flash memories, it is flexible in use and easy in implementation. The controller works with any suitable NAND Flash memory device up to 1024Gb from leading memory providers - Micron, Samsung, Toshiba and Hynix. The IP core includes a host of configuration options from page size to bank selects. The controller offers Hamming Code (1-Bit error correction and 2-Bit error detection) and BCH (option for 4-, 8-, 12-, up to 32-Bit error correction) Error Code Correction (ECC) for optimized performance and reliability. Additional features include the capability to boot from flash.

The IP core supports the Open NAND Flash Interface Working Group (ONFI) 1.0, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1 and 3.2 standards. It can also support a variety of host bus interfaces for easy adoption into any design architecture - AHB, APB, OCP, 8051 or custom buses. The slave AHB IP supports an external DMA interface where the master AHB incorporates an internal DMA controller. The Arasan NAND Flash Controller IP Cores are delivered in Verilog RTL that can be implemented in an ASIC or FPGA. They are fully tested with vendor models and hardware is tested with FPGA's. The core includes RTL code, test scripts and a test environment for complete simulation and verification.

3.2 Features

3.2.1 General

The following features are supported by the Arasan ONFI 3.2 Controller:

- Micron, Samsung, Toshiba and Hynix NAND Flash devices, up to 1024Gb
- Boot mode support
- LUN Set / LUN Get feature support
- All mandatory commands, and select optional commands
- Full access to spare area
- Speeds of 40MB/s to 533MT/s - allows to balance power / performance
- Multi LUN/DIE Operations
- Small Data Move
- Change Row Address
- Reset LUN
- EZ - NAND devices
- Chip_en pin reduction mechanism
- ODT Configure
- On-die termination
- Supports Interleaving Operations:
  - Page Program Interleaving
  - Copy back Program Interleaving

  - Block Erase Interleaving
  - Read Interleaving
  - Cache Interleaving

3.2.2 Configuration

- Page Size - 512B, 2KB, 4KB, 8KB, 16KB
- Bank/chip select options
- Programmable timing
- Address cycles - 4, 5
- ECC - enable, disable
- RAM size - 1KB, 2KB, and 4KB
- Supports parallel connection of two 8-bit flash devices

3.2.3 ECC

- SLC - Hamming Code
  - 1-Bit error correction for 512bytes
  - 2-Bit error correction for 512bytes
- MLC - BCH
  - Standard support - 4, 8-Bit error correction for 512bytes
  - Additional support - up to 32-Bit error correction either for 512bytes or 1024bytes

3.2.4 Data Interface

- ONFI 1.0, 2.0, 2.1, 2.2, 2.3, 3.0, 3.1 and 3.2 Compliant
- Supports SDR modes [0-5]
- Supports NV- DDR modes [0 -5]
- Supports NV- DDR2 modes [0 -8]
- 8-bit data bus width support for SDR, NV-DDR, and NV-DDR2
- 16-bit data bus width support only for SDR mode
- Supports independent data bus

3.2.5 System Interface

- 32-bit and 64-bit Address Bus support
- AHB - AMBA 2.0 Compliant
- AXI Interface (Optional)
- Master DMA mode

3.3 Architecture

This section describes the ONFI 3.2 Controller architecture and the description of internal blocks in detail.

The IP core supports the Open NAND Flash Interface Working Group (ONFI) 3.2 standard and is backwards compatible. It uses differential signaling on the clock and data lines and clocks at any frequency up to 267MHz.

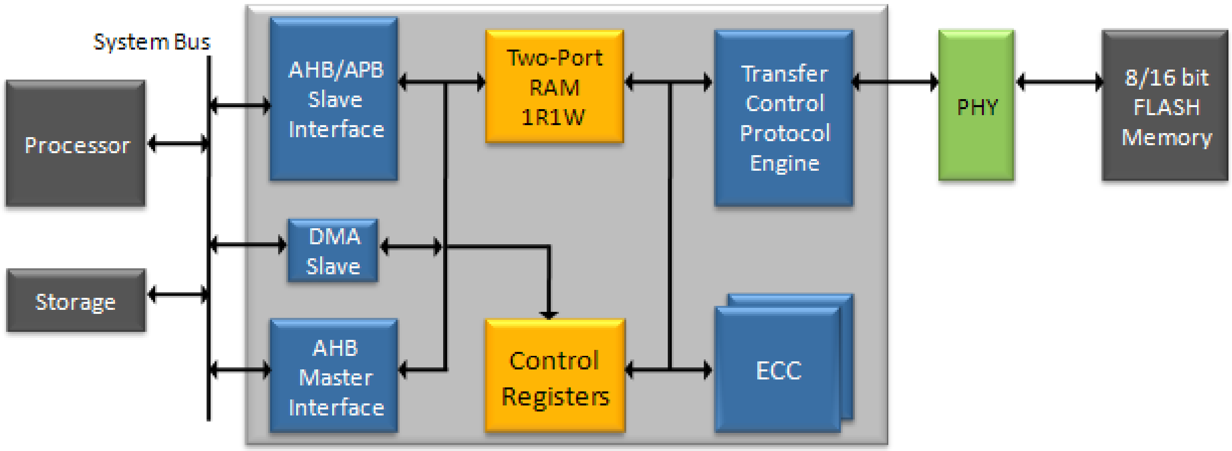

Figure 3: ONFI 3.2 Controller Functional Block Diagram

3.3.1 Functional Block Diagram Description

3.3.1.1 TX Block

The TX Block consists of logic for driving out SDR and DDR data. It also does the DQS_t and DQS_c differential signaling. It operates on 2x clock for phase-shifting the DQS 90 degrees with respect to DQ. It also does generation of differential read strobes for NV-DDR2 mode. It multiplexes the SDR and DDR data paths.

3.3.1.2 RX Block

The RX Block consists of logic to sample the incoming data. It has a dynamic phase shifter which compensates for PVT variations and aligns the DQS correctly in the middle of DQ. It also does sampling of SDR mode data. The received data is then written into an asynchronous FIFO.

3.3.1.3 ASYNC FIFO

The async FIFO is eight deep and is used for synchronizing the RX data from the NAND flash to the internal clock. The DDR data is written into the async FIFO whereas the SDR mode data bypasses the FIFO. The FIFO has sufficient depth to prevent overflow/underflow.

3.3.1.4 PLL

The PLL uses an internal 20MHz reference clock. The PLL can be configured to support various frequencies from 20MHz to 533MHz. The internal clock to the digital controller is derived by dividing the reference clock by 2.

3.3.1.5 I/O PAD

The I/O PADs are ONFI 3.2 compliant. The pad set consists of ONFI pad supporting 1.8V/3.3V I/O operation*, ONFI differential clock pad supporting 1.8V/3.3V clock*, I/O power and ground pads, core power and ground pads, and support pads like corner pad, spacer pads and so on.

* Note: I/O and clock voltages supported, depend on the process technology node.

3.4 Signal Interfaces

The ONFI 3.2 IP has the following interfaces:

- ONFI 3.2 PHY External Interface
- ONFI 3.2 PHY Controller Interface

Table 11: ONFI 3.2 PHY External Flash Interface

Pin	Direction	Description
ALE	Output	Address Latch Enable
CE_n	Output	Chip Enable
CLE	Output	Command Latch Enable
RE_t	Output	Read Enable
RE_c	Output	Read Enable Complement
WE_n	Output	Write Enable
WP_n	Output	Write Protection
R_B_n	Input	Ready/ Busy_n
DQ[7:0]	I/O	Data Inputs/Outputs
DQS_t	I/O	Data Strobe for DDR mode
DQS_c	I/O	Data Strobe Complement

Table 12: ONFI 3.2 PHY Controller Interface

Pin	Direction	Description
CLK1X	Output	Source clock to ONFI 3.2 Digital Controller, Divide by 2 of CLK2X
R_B_IN	Output	Ready Busy from External Flash Interface
DDR_DATA_VALID	Output	RX Data Valid for DDR mode
DATA_IN[15:0]	Output	RX Data for SDR and DDR mode
PLL_STABLE	Output	PLL Stable indication
RST_n	Input	Asynchronous, active low, Power on reset

Pin	Direction	Description
DATA_8_EN	Input	Enables SDR DATA, DDR command and Address
DATA_OUT_EN	Input	Enables Data Direction
DQS_OUT_EN	Input	Enables DQS Direction
DATA_OUT[15:0]	Input	TX Data/Address/Command for SDR/NVDDR/NV-DDR2 mode
DQS_VALID	Input	Enables DQS_t and DQS_c
DQS_MASK	Input	DQS Mask for NV-DDR2 mode
ALE_OUT	Input	Address Latch Enable from Controller
CLE_OUT	Input	Command Latch Enable from Controller
CE_OUT	Input	Chip Enable from Controller
RE_OUT	Input	Read Enable from Controller
WE_out	Input	Write Enable from Controller
WP_OUT	Input	Write Protect Enable from Controller
DDR_RE_EN	Input	Read Strobe Enable for DDR Mode
DDR_RE_MASK	Input	Read enable mask for NV-DDR2 mode
REF_CLK_IN	Input	PLL Reference Clock, 200MHz
PLL_EN	Input	Enables the PLL
PLL_FREQ[4:0]	Input	Changes the Frequency of the PLL Output (CLK2X)
V18_EN	Input	1.8V Enable
VccQ	Input	I/O supply voltage (3.3V/1.8V)
VccS	Input	I/O Ground voltage (0V)
Vcc	Input	Core supply voltage (1.2V)
Vss	Input	Core Ground voltage (0V)
VREF	Input	Reference Voltage for Differential Pad
MODE[1:0]	Input	SDR mode/NV-DDR mode/NV_DDR2 mode
DQS_BUFF_SEL[3:0]	Input	Configurable value for delaying Dqs strobe
DRIVE_STRENGTH[1:0]	Input	Programmable Drive Strength

3.5 Deliverables

The full IP package complete with:

- Graphic Design System II (GDSII) database
- Layout Versus Schematic (LVS) Netlist
- Physical Abstract Model (LEF)
- Timing Models
- Behavioral Models
- Design Integration Guide
- Technical Documentation

4 ONFI 3.2 PHY

4.1 Overview

The Open NAND Flash Interface (ONFI) is an Open standard for NAND Flash Memory chips. ONFI seeks to standardize the low level interface. ONFI 3.2 is the standard for High-Speed NAND Flash interface. It has multiple modes of operation like SDR, NV-DDR and NV-DDR2 modes. Micron's ClearNAND operation such as Queue page read and Program page pause, Program page resume, Program page delay is being supported.

ONFI 3.2 provides a high speed interface supporting transfer rates up to 533MT/s. It also supports the EZ-NAND interface. Performance enhancing features, such as Interleaving and Multi-plane operations, are supported.

ONFI 3.2 PHY incorporates the full TX/RX logic for NV-DDR2/NV-DDR modes of operation, and is backwards compatible to SDR mode of operation.

4.2 Features

- Compliant to ONFI specification version 3.2
- Supports NV-DDR2 mode of operation supporting up to 267MHz
- Supports NV-DDR mode of operation supporting up to 100MHz
- Supports legacy Asynchronous devices operating from 10MHz to 50MHz
- Can be used with any other ONFI digital controller
- Supports 1.8V/3.3V operation I/O pads (process node dependant)
- Contains a PLL to support all frequencies from 10MHz to 267MHz
- Uses 2X clock for better timing
- Dynamically center aligns the DQS for better noise margin and immune to PVT variations
- Supports differential signaling of DQS and RE signals
- Supports four levels of drive strength as mentioned in the ONFI 3.2 standard

4.3 Architecture

4.3.1 Functional Description

The PHY is intended to work with Arasan's ONFI3.2 NAND Flash Controller IP. The PHY supports the NV-DDR mode of operation which has maximum of 200MT/s, and NV-DDR2 mode of operation which has maximum of 533MT/s. The PHY also supports the legacy SDR mode of operation. Arasan's ONFI 3.2 PHY can be used with EZ-NAND and Micron's ClearNAND.

Arasan's ONFI 3.2 PHY supports the optional differential signaling for DQS and RE. It has the dynamic flexibility for phase shifting the DQS for center aligning for better noise margin and PVT variations.

4.3.2 Functional Block Diagram

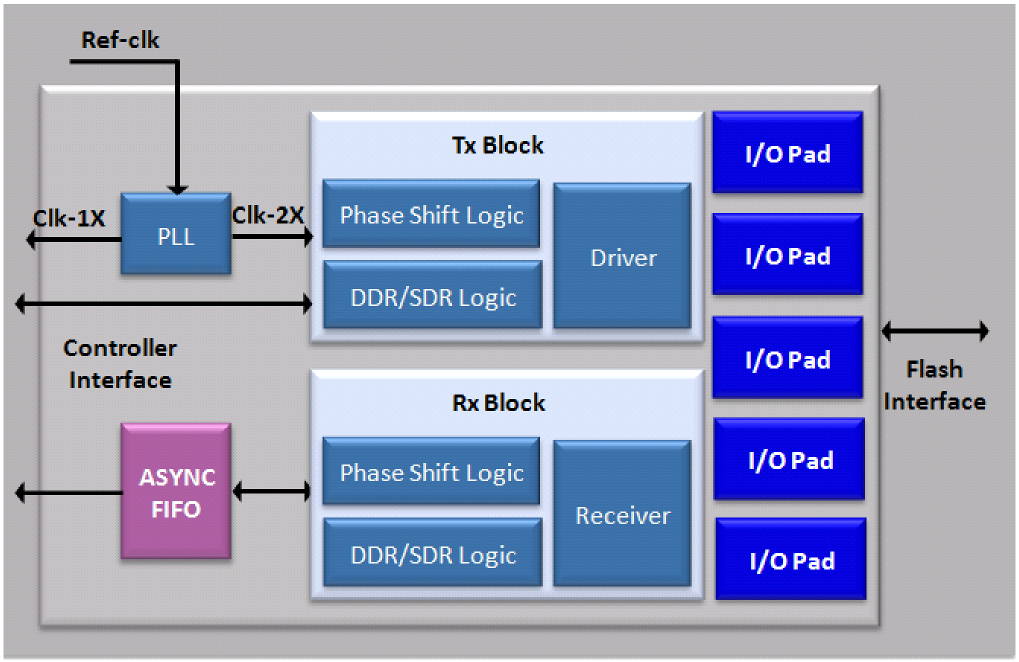

Figure 4: ONFI 3.2 Functional Block Diagram

4.3.3 Functional Block Diagram Description

4.3.3.1 TX Block

The TX Block consists of logic for driving out SDR and DDR data. It also does the DQS_t and DQS_c differential signaling. It operates on 2x clock for phase-shifting the DQS 90 degrees with respect to DQ. It also does generation of differential read strobes for NV-DDR2 mode. It multiplexes the SDR and DDR data paths.

4.3.3.2 RX Block

The RX Block consists of logic to sample the incoming data. It has a dynamic phase shifter which compensates for PVT variations and aligns the DQS correctly in the middle of DQ. It also does sampling of SDR mode data. The received data is then written into an asynchronous FIFO.

4.3.3.3 ASYNC FIFO

The async FIFO is eight deep and is used for synchronizing the RX data from the NAND flash to the internal clock. The DDR data is written into the async FIFO whereas the SDR mode data bypasses the FIFO. The FIFO has sufficient depth to prevent overflow/underflow.

4.3.3.4 PLL

The PLL uses an internal 20MHz reference clock. The PLL can be configured to support various frequencies from 20MHz to 533MHz. The internal clock to the digital controller is derived by dividing the reference clock by 2.

4.3.3.5 I/O PAD

The I/O PADs are ONFI 3.2 compliant. The pad set consists of ONFI pad supporting 1.8V/3.3V I/O operation*, ONFI differential clock pad supporting 1.8V/3.3V clock*, I/O power and ground pads, core power and ground pads, and support pads like corner pad, spacer pads and so on.

* Note: I/O and clock voltages supported, depend on the process technology node.

4.4 Signal Interfaces

The ONFI 3.2 IP has the following interfaces:

- ONFI 3.2 PHY External Interface
- ONFI 3.2 PHY Controller Interface

Table 13: ONFI 3.2 PHY External Flash Interface

Pin	Direction	Description
ALE	Output	Address Latch Enable
CE_n	Output	Chip Enable
CLE	Output	Command Latch Enable
RE_t	Output	Read Enable
RE_c	Output	Read Enable Complement
WE_n	Output	Write Enable
WP_n	Output	Write Protection
R_B_n	Input	Ready/ Busy_n
DQ[7:0]	I/O	Data Inputs/Outputs
DQS_t	I/O	Data Strobe for DDR mode
DQS_c	I/O	Data Strobe Complement

Table 14: ONFI 3.2 PHY Controller Interface

Pin	Direction	Description
CLK1X	Output	Source clock to ONFI 3.2 Digital Controller, Divide by 2 of CLK2X
R_B_IN	Output	Ready Busy from External Flash Interface
DDR_DATA_VALID	Output	RX Data Valid for DDR mode
DATA_IN[15:0]	Output	RX Data for SDR and DDR mode
PLL_STABLE	Output	PLL Stable indication

Pin	Direction	Description
RST_n	Input	Asynchronous, active low, Power on reset
DATA_8_EN	Input	Enables SDR DATA, DDR command and Address
DATA_OUT_EN	Input	Enables Data Direction
DQS_OUT_EN	Input	Enables DQS Direction
DATA_OUT[15:0]	Input	TX Data/Address/Command for SDR/NVDDR/NV-DDR2 mode
DQS_VALID	Input	Enables DQS_t and DQS_c
DQS_MASK	Input	DQS Mask for NV-DDR2 mode
ALE_OUT	Input	Address Latch Enable from Controller
CLE_OUT	Input	Command Latch Enable from Controller
CE_OUT	Input	Chip Enable from Controller
RE_OUT	Input	Read Enable from Controller
WE_out	Input	Write Enable from Controller
WP_OUT	Input	Write Protect Enable from Controller
DDR_RE_EN	Input	Read Strobe Enable for DDR Mode
DDR_RE_MASK	Input	Read enable mask for NV-DDR2 mode
REF_CLK_IN	Input	PLL Reference Clock, 200MHz
PLL_EN	Input	Enables the PLL
PLL_FREQ[4:0]	Input	Changes the Frequency of the PLL Output (CLK2X)
V18_EN	Input	1.8V Enable
VccQ	Input	I/O supply voltage (3.3V/1.8V)
VccS	Input	I/O Ground voltage (0V)
Vcc	Input	Core supply voltage (1.2V)
Vss	Input	Core Ground voltage (0V)
VREF	Input	Reference Voltage for Differential Pad
MODE[1:0]	Input	SDR mode/NV-DDR mode/NV_DDR2 mode
DQS_BUFF_SEL[3:0]	Input	Configurable value for delaying Dqs strobe
DRIVE_STRENGTH[1:0]	Input	Programmable Drive Strength

4.5 Deliverables

The full IP package complete with:

- Graphic Design System II (GDSII) database
- Layout Versus Schematic (LVS) Netlist
- Physical Abstract Model (LEF)
- Timing Models
- Behavioral Models
- Design Integration Guide
- Technical Documentation

5 ONFI 2.3 NAND Flash Controller

5.1 Overview

The Arasan ONFI 2.3 NAND Flash Controller IP Core is a full featured, easy to use, synthesizable core, easily integrated into any SoC or FPGA development. Designed to support both SLC and MLC flash memories, it is flexible in use and easy in implementation. The controller works with any suitable NAND Flash memory device up to 128 Gb from leading memory providers. The IP core includes a host of configuration options from page size to band selects. The controller offers Hamming Code (1-bit error correction and 2-bit error detection) and BCH (option for 4-, 8-, 12-, to more than 32-bit error correction) error code correction (ECC) for optimized performance and reliability. Additional features include the capability to boot from flash.

The IP core supports the Open NAND Flash Interface Working Group (ONFI) 2.3 standard and Micron ClearNAND. It can also support a variety of host bus interfaces for easy adoption into any design architecture - AHB, APB, OCP, 8051 or custom buses. The slave AHB IP supports an external DMA interface where the master AHB incorporates an internal DMA controller.

An optional NAND Flash File system is available to support advanced features. The file system converts complicated tasks of NAND flash memory interfacing to simple memory access. Flash memory read, write, garbage collection, bad block management and other functions are handled in the background by the file system.

The Arasan ONFI 2.3 NAND Flash Controller IP Cores are delivered in Verilog RTL that can be implemented in an ASIC or FPGA. They are fully tested with vendor models and hardware is tested with FPGA's. The core includes RTL code, test scripts and a test environment for complete simulation and verification.

5.2 Features

5.2.1 General

- Supports ONFI 2.3 source synchronous mode DDR [0-5], and legacy asychronous mode [0-5]
- Supports ONFI EZ NAND devices
- Supports Micron Enhanced ClearNAND and Standard ClearNAND
- Performance ranging from 40MB/s to 200MB/s to allow applications to balance performance and power
- Supports NAND Flash devices up to 128Gb
- Boot mode support
- Supports all mandatory commands and selected optional commands
- Full access to spare area
- Supports Interleaving Operations
  - Page Program Interleaving
  - Copyback Program Interleaving

  - Block Erase Interleaving
  - Read Interleaving
  - Cache Interleaving
- Supports Multi LUN/DIE Operations
- Supports Small Data Move
- Supports Change Row Address
- Supports Copyback operations with no plane and LUN restrictions
- Supports Multi plane Erase operation
- Supports Page program Multi plane operation

5.2.2 Features support to Micron Enhanced ClearNAND

- Supports CE# pin reduction mechanism
- Supports volume addressing and volume select command
- Supports a Interrupt(INTR#) signal which provides hardware method of indicating when a command is complete
- Supports commands like Synchronous reset (FCh), Reset LUN (FAh), Volume select (E1h), LUN status (71h), Device status (72h), Get next operation status (77h), Operation status(7Dh)
- Supports Queue page read and Program page pause, Program page resume, program page delay

5.2.3 Configuration

- Page Size - 512B, 2KB, 4KB, 8KB, 16KB
- Bank/chip select options
- Programmable timing
- Address cycles - 4, 5
- ECC - enable, disable
- Flash data bus width
  - Standard support - 8-bit for both Asynchronous and Synchronous mode
  - Additional support - 16-bit only for Asynchronous mode
- RAM size - 1KB, 2KB, 4KB
- Supports parallel connection of two 8-bit flashes

5.2.4 ECC

- SLC - Hamming Code
  - 1-bit error correction
  - 2-bit error detection
- MLC - BCH
  - Standard support - 4, 8-bit error correction
  - Additional support - 32-bit or more error correction

5.2.5 Data Interface

- 8-bit data bus width support for both Asynchronous mode and Synchronous mode
- 16-bit data bus width support only for Asynchronous mode.
- Supports independent data bus

5.2.6 System Interface

- AHB - AMBA 2.0 Compliant
- Optional Avalon Bus - Available up on request
- PIO Mode
- Slave DMA mode (Optional)
- Master DMA mode (Optional)

5.3 Architecture

5.3.1 Functional Block Diagram

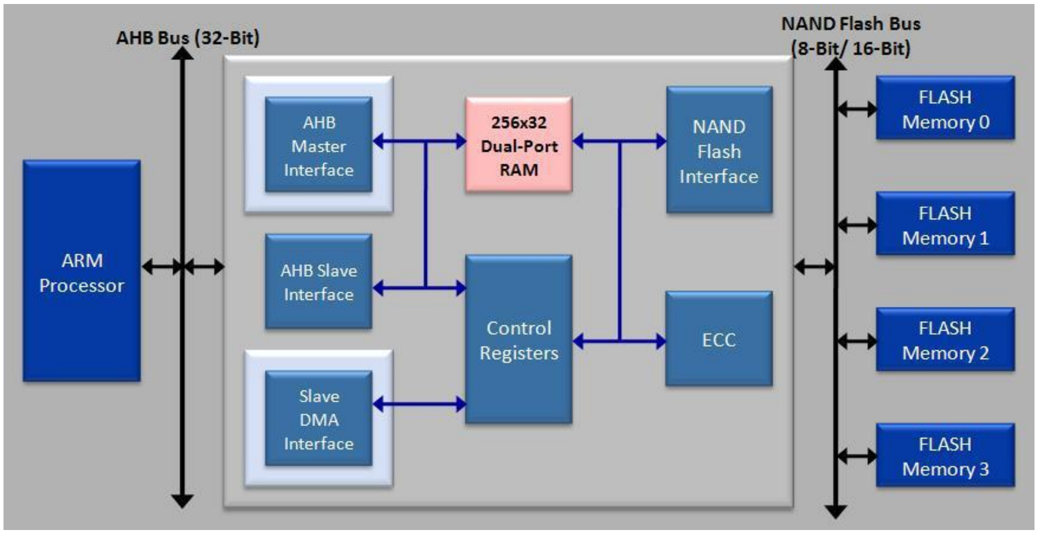

Figure 5: ONFI 2.3 AHB IP Controller Functional Block Diagram

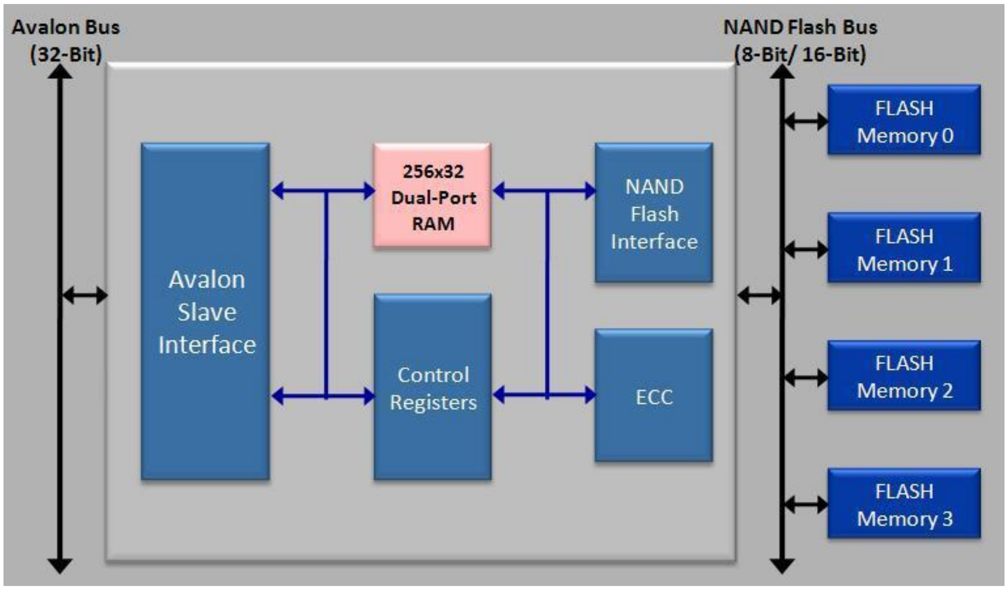

Figure 6: ONFI 2.3 Avalon IP Controller Functional Block Diagram

5.3.2 Functional Block Diagram Description

5.3.2.1 NAND Flash Interface

The NAND Flash Interface handles all the command, address, and data sequences and manages all the hardware protocols. It is ONFI 2.3 compliant and Micron ClearNAND support and provides an 8-bit or 16-bit interface to the flash memories. The interface supports a maximum of 128 Gb of NAND flash memory. Both Asynchronous and Synchronous data interfaces are supported. All timing modes (0-5) are supported for both Asynchronous mode and Synchronous mode.

5.3.2.2 AHB Slave Interface

The AHB Slave block consists of the operational registers. A processor connecting to this interface can control the operation of the ONFI NAND Flash controller through the NAND Flash control registers.

5.3.2.3 AHB Master Interface

ONFI NAND Flash controller acts as a master during MDMA mode of transaction. The AHB master interface places control signals in AHB Bus depends upon the FIFO status. During write transaction, AHB master interface read data from system memory and stores into the FIFO. During read transaction, read data from FIFO and write data into system memory. AHB master interface asserts DMA interrupt when DMA buffer boundary is reached. The AHB master interface can be used to transfer boot code from the NAND flash memory to the system memory during system power-up.

5.3.2.4 Dual-Port RAM

This block has handshake logic to communicate with the AHB interface and on the other side communicates with the Flash Interface. Typical RAM size is 256x32, to support block size of 512 bytes. The FIFO depth is configurable.

5.3.2.5 ECC

The ECC module provides error detection and correction support for SLC and MLC Flash memory. The Hamming Code supports 1-bit error correction and 2-bit error detection. The BCH Code supports more than 32-bit error correction. For SLC Flash devices, Hamming Code logic is enabled and for MLC Flash devices, BCH is enabled. An optional pipeline stage in the BCH decoder can be enabled for maximum performance.

5.3.2.6 Control Registers

The host processor controls the configuration and operation of the NAND Flash Controller through the Control Registers. Configuration includes the set up time (tCCS, tDQSQ, tDS), memory configuration (address, page size, packet size, packet count), timing modes (async mode, sync mode) and so on. The Control Registers also provide operating status such as Busy and Data Ready signals.

5.3.2.7 Slave DMA Interface

This interface is used to perform a SDMA mode of Write/ Read data transfer. Slave DMA Interface provides DMA Request, DMA Single and DMA Last and get acknowledged by DMA Acknowledge and DMA Finish. DMA Request is asserted until the required amount of data is transferred based on the DMA Transfer count. For the generated DMA Request Slave DMA Controller provides write data for write transfer or reads the data from FIFO for read transfer.

5.3.2.8 Avalon Slave Interface (Optional)

The ONFI NAND Flash Controller provides Programmed I/O method in which the Avalon Master transfers data using the Buffer Data Port Register. The Avalon master programs the Control registers using Avalon Slave Interface. Data transaction is performed through Avalon Slave Interface.

5.4 ONFI 2.3 Controller Signal Interfaces

This chapter contains description and the direction of the pins from interfaces such as AHB-Master, AHB- Slave, Avalon- Slave, NAND_Flash, SDMA, RAM and Clock and Reset signals.

The NAND Flash controller has nine main interface groups:

- AHB Master Interface
- AHB Slave Interface
- NAND Flash Interface

- Slave DMA Interface
- Avalon Slave Interface (Optional)
- RAM Interface
- Clock and Reset Signal

Table 15: AHB Master Interface

Pin	Direction	Description
m_hbusreq	Output	AHB Bus Request
m_hgrant	Input	AHB Bus Grant
m_haddr[31:0]	Output	Address Bus
m_hwdata[31:0]	Output	Write Data Bus
m_hrdata[31:0]	Input	Read Data Bus
m_hwrite	Output	Write or Read Direction Indication
m_hsize[2:0]	Output	Size(Byte, Half Word or Word)
m_hburst[2:0]	Output	Burst Size
m_htrans[1:0]	Output	Transfer type
m_hready	Input	Slave Ready
m_hresp[1:0]	Input	Transfer Response
boot_enable	Input	Reads Boot Code from Flash

Table 16: AHB Slave Interface

Pin	Direction	Description
ahb_hsel	Input	Slave Select
ahb_haddr[31:0]	Input	Address Bus (Byte Addresses)
ahb_hwdata[31:0]	Input	Write Data Bus
ahb_hrdata[31:0]	Output	Read Data Bus
ahb_hwrite	Input	Write or Read Direction Indication
ahb_hburst	Input	Burst (Single, Incrementing, wrapping)
ahb_hsize[2:0]	Input	Size (Byte, Half Word or Word)
ahb_htrans[1:0]	Input	Transfer Type
ahb_hready_in	Input	Global Ready
ahb_hready	Output	Slave Ready
ahb_hresp[1:0]	Output	Transfer Response
int_to_arm	Output	Interrupt to the ARM

Table 17: NAND Flash Interface

Pin	Direction	Description
cle	Output	Command Latch Enable
ale	Output	Address Latch Enable
ce_n0	Output	Chip Enable for Flash device 0
ce_n1	Output	Chip Enable for Flash device 1

Pin	Direction	Description
ce_n2	Output	Chip Enable for Flash device 2
ce_n3	Output	Chip Enable for Flash device 3
re_n	Output	Read Enable for Asynchronous interface. write / read direction for source synchronous interface
we_n	Output	Write Enable for Asynchronous interface. clock strobe for source synchronous interface
rb_n0	Input	Ready/Busy Output from device 0 (Not used for Micron Enhanced ClearNAND)
rb_n1	Input	Ready/Busy Output from device 1 (Not used for Micron Enhanced ClearNAND)
rb_n2	Input	Ready/Busy Output from device 2 (Not used for Micron Enhanced ClearNAND)
rb_n3	Input	Ready/Busy Output from device 3 (Not used for Micron Enhanced ClearNAND)
io_in[31:0]	Input	DATA Input
io_out[31:0]	Output	DATA Output
io_out_en	Output	Enable signal for DATA Output
dqs_in	Input	Signal for Read in Source synchronous timing mode
dqs_out	Output	Signal for Write in Source synchronous timing mode
dqs_out_en	Output	Enable signal for Write in Source synchronous timing mode.
INTR	Input	Interrupt signal from the device which provides a hardware method of indicating when a command is complete. (Used only for Micron Enhanced ClearNAND support)

Table 18: Slave DMA Interface

Pin	Direction	Description
dma_ack	Input	DMA Acknowledge This is asserted after the data phase of the last transfer in the current transaction (single or burst) to the peripheral has completed. It forms a handshaking loop with dma_req and remains asserted until the peripheral de-asserts dma_req (deasserted one hclk cycle later).
dma_finish	Input	DMA Finish DMA Controller block transfer complete signal. The DMAC asserts dma_finish in order to signal block completion. This uses the same timing as dma_ack and forms a handshaking loop with dma_req.
dma_req	Output	DMA Request

Pin	Direction	Description
		Transaction request from peripheral. A rising edge on dma_req initiates a transaction request. The type of transaction – single or burst – is qualified by dma_single. Once dma_req is asserted, it must remain asserted until dma_ack is asserted. When the peripheral that is driving dma_req determines that dma_ack is asserted, it must de-assert dma_req.
dma_last	Output	DMA Last Last transaction in block. When the peripheral is the flow controller, it asserts dma_last on the same cycle as dma_req is asserted in order to signal that this transaction request is the last in the block; the block transfer is completed after this transaction is complete. If dma_single is high in the same cycle, then the last transaction is a single transaction. If dma_single is low in the same cycle, then the last transaction is a burst transaction.

Table 19: Avalon Slave Interface (Optional)

Pin	Direction	Description
s_address[7:0]	Input	Address lines from Avalon Bus Module
s_byteenable[3:0]	Input	Byte enable signal to enable specific byte during slave transfer
s_write	Input	1’b1 - Write request signal to slave port
s_read	Input	1’b1 - Read request signal to slave port
s_writedata[31:0]	Input	Data lines from Avalon Bus Module for write transfers.
s_begintransfer	Input	Asserted during the first bus cycle of each new Avalon Bus transfer
s_beginbursttransfer	Input	Asserted for the first cycle of a burst to indicate when a burst transfer is starting
s_burstcount[7:0]	Input	Used for burst transfer to indicate the number of transfers in a burst
s_waitrequest	Output	Asserted when slave port is not able to respond immediately
s_readdatavalid	Output	Used only by slaves with variable latency. Marks the rising clock edge when the slave asserts valid read data
s_readdata[31:0]	Output	Data lines to the Avalon bus module for read transfers

Table 20: RAM Interface

Pin	Direction	Description
data_to_sys [31:0]	Input	AHB domain data output
data_to_flash_from_fifo [31:0]	Input	Flash domain data output
ram_addr_a	Output	Address bus for AHB domain
wr_data_a [31:0]	Output	AHB domain data input
cen_a	Output	Active high chip select.(AHB domain)
wren_a	Output	Active high write enable signal.(AHB domain)
ram_addr_b	Output	Address bus for Flash domain
wr_data_b [31:0]	Output	Flash domain data input
cen_b	Output	Active high chip select.(Flash domain)
wren_b	Output	Active high write enable signal.(Flash domain)

Table 21: Clock and Reset Signals

Pin	Direction	Description
clk_sys	Input	System clock
clk_flash	Input	Source clock to ONFI 2.3 Digital Controller
rst_n	Input	Active low Asynchronous power on reset from external environment and it is synchronized inside the IP. The synchronized reset is used to reset all the flops in FLASH, AHB clock domain

Note: ONFI clk_flash Frequency

Asynchronous Mode

1. Firmware should program clock for 100MHz
2. Firmware should wait for clock stable in PLL
3. Firmware should enable the internal clock and program the mode of operation in program register.

Synchronous Mode

1. Firmware should program the appropriate frequency.
2. Firmware should wait for clock stable in PLL
3. Firmware should enable the internal clock and program the mode of operation in program register.

Note: rst_flash_n

Active low Asynchronous reset to rest the FLASH domain Flops. This synchronized reset signal is generated internally based on the rst_n signal in FLASH clock domain.

Note: rst_sys_n

Active low Asynchronous reset to reset the AHB domain Flops. This synchronized reset signal is generated internally based on the rst_n signal in AHB clock domain.

5.5 SoC Level Integration

5.5.1 Verification Environment

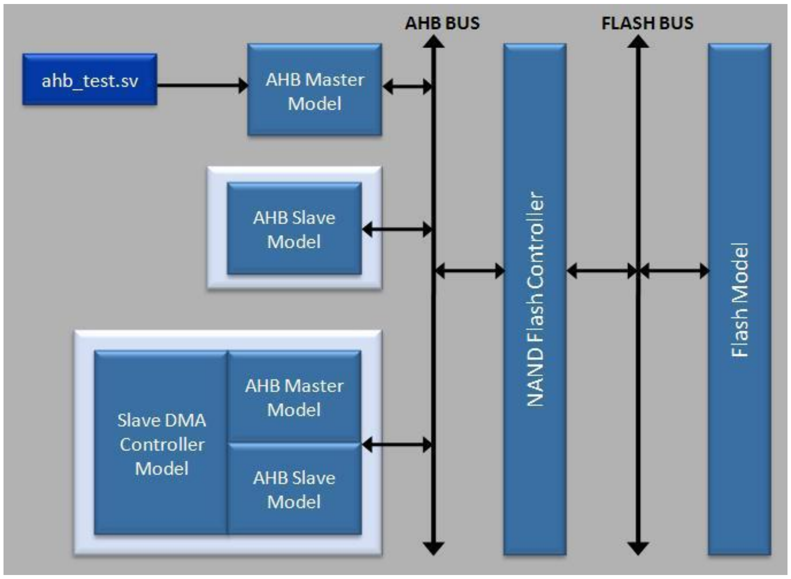

Figure 7: ONFI 2.3 Verification Environment

5.5.2 Deliverables

- Verilog HDL of the IP core
- User Guide
- Test environment and test scripts
- Synthesis scripts
- Gate count estimates available upon request
- Sample ARM firmware and software drivers

6 ONFI NAND Controller Software Driver

6.1 Overview

The ONFI NAND Controller Software Driver is developed as the bridge for the Linux MTD subsystem to communicate to the NAND devices. The modular NAND Controller Driver is architected to be easily ported to different operating systems with minimum efforts.

The Driver consists of three layers - the Interface layer, Hardware layer and OS Abstraction Layer. The Interface layer acts as an intermediate between the hardware layer and the linux MTD driver. The MTD driver uses the APIs of this layer to access, control and configure hardware interface driver and the underlying hardware. This layer can be replaced with user-specific interface to bridge with any OS dependent Flash File System (FFS).

The Hardware layer takes the inputs from Interface layer, translates into hardware specific command and protocol, communicates directly to the underlying NAND flash devices, receives the data and status, and sends the data and status back to the MTD subsystem through the Interface layer. The low level Hardware layer is purely OS independent and users can use this layer alone for NAND flash device validation with no driver complexity.

The OS Abstraction layer provides all OS specific calls. When porting the driver to some other platform this layer may be changed with corresponding calls of OS to be ported.

6.2 Features

- Compliant with ONFI 3.2 and backward compatible to ONFI 3.0, 2.3 and 1.0
- Supports SDR, NV-DDR and NV-DDR2, Toggle DDR/DDR2 modes
- Easy-to-use interface for applications
- Support up to NAND with 16K page size
- Fully documented generic device operation API
- Supports Read, Program and Erase Operations
- Supports Cache Program, Cache Sequential and Random Read Operations
- Supports LUN Resets, Multi LUN/DIE Operations
- Supports On Die Termination
- Supports Interleaving Operations
- Easily portable to any OS, processors or hardware

6.3 Description

The Arasan ONFI NAND Software Driver consists of the following layers:

1. Interface Layer
2. Hardware Layer

3. OS Abstraction Layer

The layered architecture allows for easy porting to various operating systems and various platforms. The low level details of the protocol are abstracted for the end-user and handled in the software Driver. The Driver includes functions to initialize, program, erase and read the NAND flash. It also provides API for NAND management including interleave, copy back, program cache, etc.

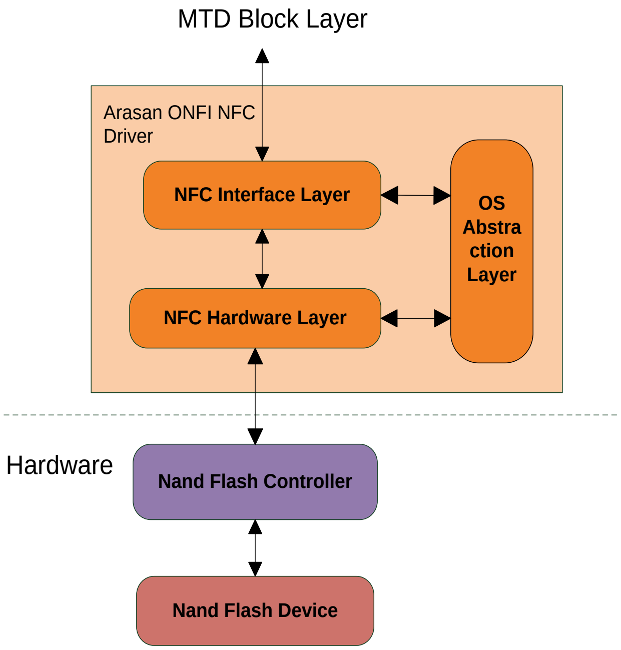

Figure 8: ONFI NAND Controller Driver Architecture

6.4 Benefits

- System manufacturers can port the ONFI NAND Controller Driver to respective system hardware and operating systems
- Silicon developers can use the driver and board environment to test the device silicon during development
- Silicon vendors can use the driver to create a reference system design for their customers

6.5 Deliverables

- Source code (in c language) and binaries for ONFI NAND Controller Driver
- User Manual

7 Services & Support

7.1 Global Support

Arasan Chip Systems provide global support to its IP customers. The technical support is not geographically bound to any specific site or location, and therefore our customers can easily get support for design teams that are distributed in several locations at no extra cost.

7.2 Arasan Support Team

Our technical support is provided by the engineers who have designed the IP. That is a huge benefit for our customers, who can communicate directly with the engineers who have the deepest knowledge and domain expertise of the IP, and the standard to which it complies.

7.3 Professional Services & Customization

At Arasan Chip Systems we understand that no two Application Processors are the same. We realize that often the standard itself needs some tweaks and optimizations to fit your design better. Sometimes, the interface between the IP blocks and your design need some customization. Therefore, we provide professional services and customization to our IP customers. We do not sell our IP blocks as “black box” that cannot be touched. Please contact us for more details on our customization services.

7.4 The Arasan Porting Engine

Analog IP blocks, such as eMMC 5.1 HS400 PHY, are designed for a specific Fab and process technology. Arasan’s analog design team, utilizing its deep domain expertise and vast experience, is capable of porting the PHYs into any specific process technology required by the customer. That is “The Arasan Porting Engine”.

7.5 Pricing & Licensing

Arasan charges a one-time licensing fee, with no additional royalties. The licensing fee gives the right to use our IP for 1 project. Licensing fee for additional projects, using the same IP, is discounted. We also offer unlimited-use license. For any additional information regarding pricing and licensing – please contact our sales at: sales@arasan.com.