Datasheet

eMMC v1.5 Total IP Solution
Including eMMC 5.1 PHY in 40nm, 28nm and 16nm FinFET (16FF+)

eMMC Spec Version 5.1 Compliant

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

No part of this specification may be reproduced in any form or means, without the prior written consent of Arasan Chip Systems, Inc.

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

Contents

Tables

Figures

1 Introduction

1.1 About eMMC

eMMC, short for "embedded Multi-Media Card", is an embedded non-volatile memory system, comprised of both flash memory and a flash memory controller integrated in the same industry-standard BGA package.

eMMC architecture, integrating the flash memory and controller in the same package, simplifies the application interface design and frees the host processor from low-level flash memory management. This benefits product developers by simplifying the non-volatile memory interface design and qualification process – resulting in reducing time-to-market, as well as future proofing against new flash memory technology advances.

The eMMC standard has been developed and published by JEDEC™ Solid State Technology Association (www.jedec.org), the global leader in the development of standards for the microelectronics industry. JEDEC has over 4,000 participants, representing nearly 300 companies, working together in 50 JEDEC committees.

The latest revision of the JEDEC standard is 5.0, released on Sep. 2013, and defines a maximum bandwidth of 400 MB/s over 8 data lanes. The combination of 200 MHz DDR clock rate, and 8 data lanes, requires the use of a hard PHY. Arasan has designed the analog PHY in 40nm, 28nm, and 16nm FinFET+ process technologies, and are all silicon proven. Latest eMMC revision of the JEDEC about to be released is eMMC5.1 (Item # JC-64.1-67.14).

1.2 Arasan's Contribution to JEDEC and SD Association

Arasan Chip Systems has been an executive member with the Multi Media Card Association (MMCA) since 2002. The MMCA was later merged with JEDEC in 2008 as the emphasis of MMCA shifted from removable storage to embedded storage for mobile devices. Arasan is currently a contributing member to JEDEC actively promoting both the eMMC and UFS standards. Arasan is also a contributing member to the SD Association since 2001.

Arasan is the leader of mobile storage, with 300 IP licensees since 2002 for SD, SDIO, NAND, eMMC and UFS. Our eMMC Host and Device IPs were licensed to both Application Processor companies like Intel, LG, Samsung and Huawei, as well as the majority of the Memory companies, and includes SK Hynix, among others.

Arasan's active involvement and contribution to the relevant standards bodies, lead to deep domain expertise, which in turn results in early availability of high quality IP for our customers.

1.3 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

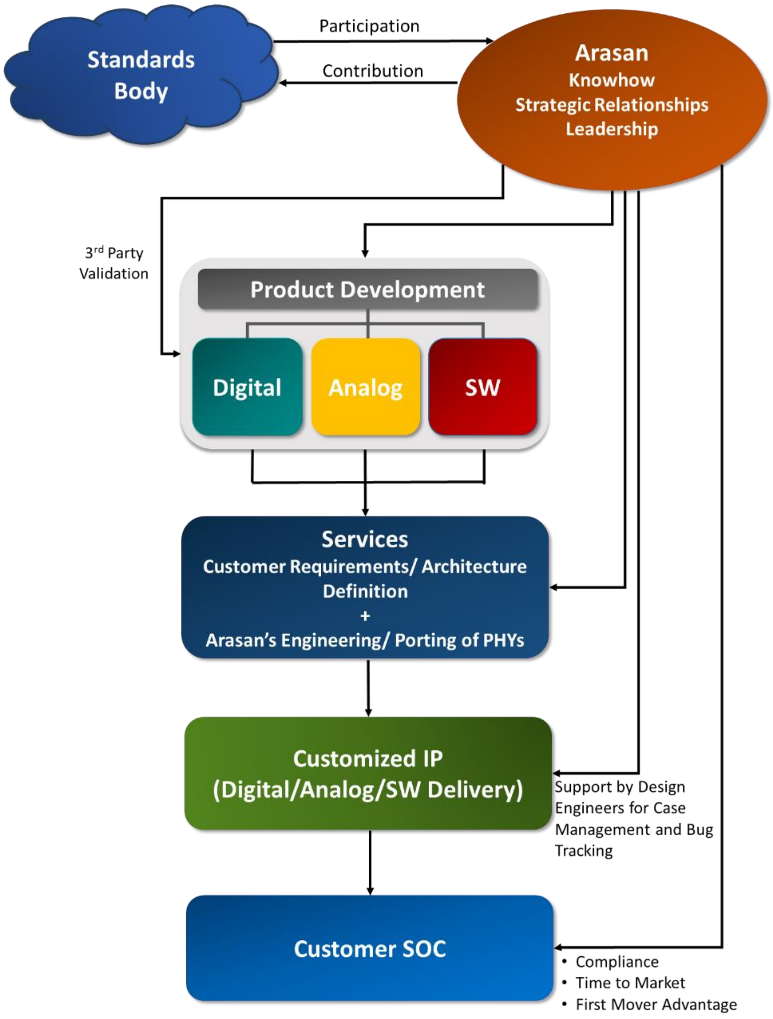

Figure 1: Arasan's Total IP Solution

2 eMMC 5.1 Host Controller Datasheet

2.1 Overview

Arasan Chip Systems' eMMC5.1 Host Controller IP is a highly integrated Intellectual Property (IP) solution that supports three key memory and I/O technologies:

1. JEDEC eMMC (eMMC Multi Media Card) 5.1 (Draft)
2. JEDEC eMMC CQHCI for Command Queueing
3. SDA Secured Digital (SD) 3.01
4. SDA Secured Digital Input Output (SDIO) 3.01

This IP handles all the timing and interface protocol requirements to access these media as well as processing the commands in hardware.

The IP supports connection to a single slot and performs multi-block writes and erases that lower access overhead. In addition, a host can utilize this IP to boot directly from an attached eMMC device, thereby simplifying system initialization during power up. The host interface is based on a standard 32-bit/64-bit Advanced High-Performance Bus (AHB/AXI/OCP) which is used to transfer data and configure the IP.

2.2 Features

- Compliant with the following specifications:
  - JEDEC eMMC Specification Version 5.1
  - JEDEC eMMC CQHCI (part of eMMC5.1)
  - SDA Part A2 SD Host Controller Version 3.00
  - SDA Part 1 Physical Layer Specification Version 3.00
  - SDA Part E1 SDIO Specification Version 3.00
  - AMBA, AHB Specification Version 2.00
  - AMBA, Advanced Extensible Interface (AXI) Specification Version 1.00 (Optional)
  - Open Core Protocol (OCP) Specification Version 2.2 (Optional)

- The core supports:
  - 32-bit and 64-bit system data bus.
  - 32-bit and 64-bit system addressing.
  - Interrupts and wake up functionality
  - Internal Clock divider for various card operational modes
  - One of the AHB, AXI or OCP System/Host bus

- The data is transferred using:
  - Programmed Input/Output (PIO) mode on the Host Bus Slave interface

- Direct Memory Access (DMA) mode using Simple DMA (SDMA) or Advanced DMA (ADMA2) on the Host Bus Master interface*
- Configurable FIFO size to support different block sizes.

Note: The Host Bus is AHB or AXI or OCP.

- eMMC 5.1 features:
  - HS400 high speed interface timing mode of up to 400 MB/s data rate
  - Transfers the data in HS400, HS200, DDR52 modes.
  - 4KB block support
  - Tuning for HS200 mode
  - Command Queuing for High Performance data transfers with Hardware Acceleration.
  - Enhanced strobe function for reliable operation at HS400 mode.
  - MMC Plus and MMC Mobile
  - Host clock rate variable between 0 and 200 MHz
  - Transfers the data in 1-bit, 4-bit and 8-bit modes
  - Supports legacy modes (Default Speed, High Speed).
  - CRC7 for command and CRC16 for data integrity
  - Password protection of cards

- UHS-I features (SD3.0/SDIO3.0):
  - 1.8V voltages switch operation
  - Tuning for SDR104 mode
  - Host clock rate variable between 0 and 208 MHz
  - Up to 832 Mbps data rate using 4 parallel data lines (SDR104 mode)
  - Transfers the data in 1-bit and 4-bit SD modes.
  - Transfers the data in SDR104, DDR50, SDR50, SDR25, SDR12, DS and HS modes
  - Cyclic Redundancy Check (CRC): CRC7 for commands, CRC16 for data integrity
  - Variable-length data transfers
  - Performs Read Wait Control, Suspend/Resume operation with SDIO CARD
  - Designed to work with I/O cards, Read-only cards and Read/Write cards
  - Card Detection (Insertion/Removal)

2.3 Architecture

2.3.1 Functional Description

The Arasan eMMC5.1 Host Controller is a Host Controller with an AHB/AXI/OCP processor interface. This product conforms to upcoming eMMC5.1 Specification from JEDEC. It is also compliant SD Host Controller Standard Specification Version 3.00.

The eMMC5.1 Host Controller handles eMMC (and also SDIO/SD) Protocol at transmission level, packing data, adding CRC, start/end bit, and checking for transaction format correctness. This Host

Controller provides Programmed IO method and DMA data transfer method. In programmed IO method, the Host processor transfers data using the Buffer Data Port Register.

The eMMC 5.1 Host Controller support for DMA can be determined by checking the DMA support in the capabilities register. DMA allows a peripheral to read or write memory without the intervention from the CPU. This Host Controller's Host Controller system address register points to the first data address, and then data is accessed sequentially from that address. It supports connection to a single slot and performs multi-block writes and erases the lower access. The eMMC5.1 Host Controller supports two DMA schemes: Simple DMA (SDMA) and Advanced DMA (ADMA2)

2.3.2 Functional Block Diagram

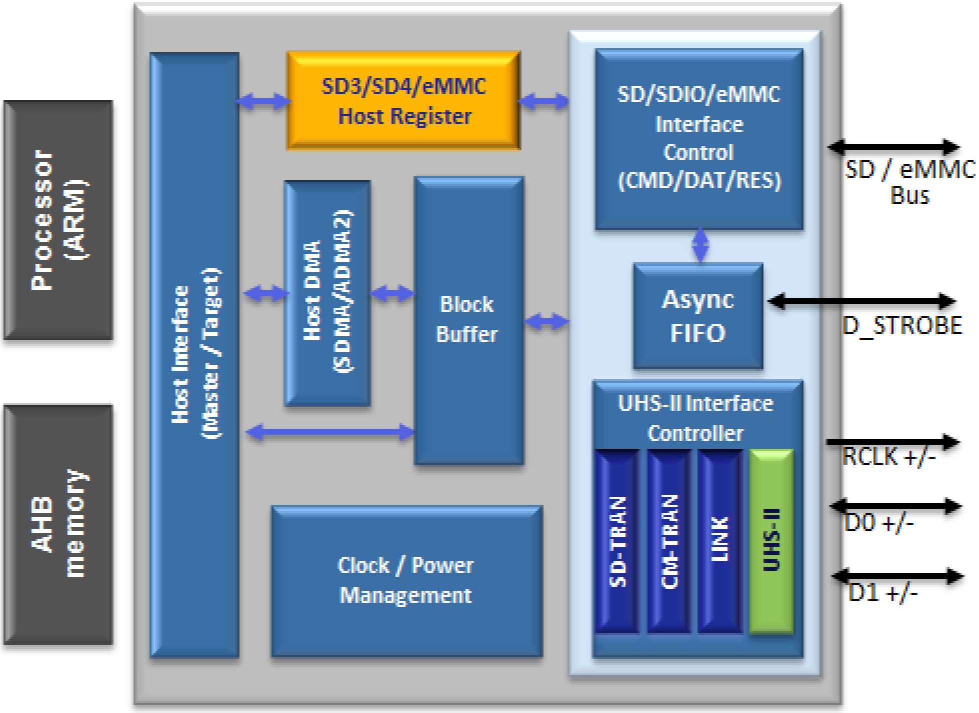

Figure 2: eMMC5.1/SD3.0/SDIO3.0 Host Controller Functional Block Diagram

2.3.3 Functional Block Diagram Description

2.3.3.1 Host Interface (Master/ Target)

The Master Bus Interface is used to access the DMA Controller (when using DMA) or Advanced Direct Memory Access (ADMA2 Modes). The DMA Controller module interfaces with the Host (AHB/AXI/OCP) Master Module to generate Transfers and on the other side it interfaces with the Block Buffer to store/fetch block data. The DMA Controller implements a separate DMA for Simple Direct Memory Access (SDMA) Operation and Separate DMA for the ADMA2 Operation. In addition, it implements Host Transaction Generator that generates controls for the Host Master Interface Module.

The DMA Controller uses the Master DMA interfaces to transfer data between the Host Controller and the System Memory and vice-versa and also to fetch the descriptors while operating in ADMA2 mode.

The Host Controller interfaces with the System bus using the AHB, AXI, or OCP Master and Slave Interface. The Slave Interface is used to access the Registers inside the Host controller. The Slave Interface supports only single transfer access (no Burst Support) and only one outstanding Read/Write transaction in case of AXI or OCP interface.

2.3.3.2 Host DMA

The PIO/DMA Controller module implements the SDMA and ADMA2 engines as defined in the SD Host Controller specification and maintains the block transfer counts for PIO operation. It interacts with the Registers Set and starts the DMA engine when a Command with Data Transfer is involved.

The DMA Controller interfaces with the Host (AHB/AXI/OCP) Master module to generate Transfers and on the other side it interfaces with the Block Buffer to store/fetch block data. It implements a separate DMA for SDMA operation and separate DMA for the ADMA2 operation. In addition implements Host Transaction Generator that generates controls for the Host Master interface module.

2.3.3.3 Command Queuing Engine

The Command Queueing Engine implements the context for 32 Tasks and is compliant with the CQHCI specification. This will drastically reduce the software overhead for data transfers by queuing up the tasks to the eMMC 5.1 compliant device and polling the device and performing the data transfers associated with these tasks.

The Command Queuing Engine controls the Host DMA and the Host Registers based on the active Task and tasks that are ready.

2.3.3.4 eMMC/ SD Host Register

The Host Controller Register Set implements the Registers defined by the SD Host Controller Specification. The Registers are Byte/DWORD accessible from the Slave interface. The Host Controller Register Set also implements the Data Port Registers for the PIO Mode transfers.

The Register Set provides the control signals and monitors the status signals from the blocks to set Interrupt Status Bits and eventually generate Interrupt signal to the Host Bus.

2.3.3.5 Block Buffer

The SD/SDIO Host Controller uses a Dual Port Block Buffer (Read/Write on both ports) or a Two Port (One Read/One Write) that is used to store the Block Data during SD Transfers. The size of the Block Buffer is Configurable and has to be a minimum of 1 Block Size (Block Size is 512 Bytes in eMMC/SD Memory and up to 2K Bytes in SDIO).

To achieve maximum performance the Block buffer has to be sized to twice the maximum Block Size supported by Host Controller. The Block Buffer uses Circular Buffer Architecture. One side of the Block Buffer is interfaced with the DMA Controller and operates on the Host Clock. The other side of the Block Buffer interfaces with eMMC/SD Interface Control Logic and operates on eMMC/SD Clock. During a write transaction (data transferred from a Host Processor to eMMC5.1/SD3.0/SDIO3.0 card), the data is fetched from the Host System Memory and is stored in the Block Buffer. When a Block of data is available, the SD Control logic will transfer it onto the eMMC Interface.

The DMA Controller continues to fetch additional block of data when the Block Buffer has space. During a Read transaction (data transferred from eMMC5.1/SD3.0/SDIO3.0 card to Host Processor), the data from eMMC5.1 card will be written in to Block Buffer and at the end when the CRC of the Block is valid, the data is committed. When a Block of data is available, then the DMA Controller transfers this data to the Host System Memory. The eMMC/SD Interface Control logic meanwhile receives the next Block of data provided there is space in the Block Buffer. If the Host controller cannot accept any data from eMMC5.10 card, then it will issue Read Wait (if card supports Read Wait mechanism) to stop the data transfer from card or by stopping the clock.

Note: FIFO depth can be varied using parameter passed to the Core using the ‘dot parameter instantiation’. When the Block Buffer size is twice that of the Block Size, the Block Buffer behaves like a ping-pong buffer.

2.3.3.6 eMMC/ SD (UHS – I) Interface Control (CMD/ DAT/ RES)

The eMMC/SD Interface Control block maps the internal signals to the External eMMC/SD Interface and vice versa. Based on the Bus Width (1/4/8) the internal signals are driven out appropriately. In case of DS, the outputs are driven on the negative edge of the sd_clk.

This module performs the Tuning procedure for HS200 (or SDR104) modes to center align the receive clock to the incoming data stream.

2.3.3.7 Clock/ Power Management

The eMMC/SD Interface Clock Generator module generates the eMMC/SD Clock from the Reference Clock (xin_clk), based on the Controls programmed in the Clock Control Register. These include the Clock Divide Value, Clock Enable and so on. The outputs from this module are the EMMC_CLK/SD_CLK and the CARD Clock. The EMMC/SD_CLK is used by the eMMC/SD Interface Control Logic and the CARD Clock connected to the “CLK” Pin on the SD/EMMC Interface. This module also generates system resets to various clock domains.

2.4 Pinouts

2.4.1 I/ O Description

The Arasan eMMC5.1 Host Controller has the following interface groups.

- System (AHB/AXI/OCP) Bus Interface Signals
- eMMC5.1 SD3.0 / SDIO3.0 / eMMC5.0 Interface that forms the main card interface
- Power Control Signals
- Clock, Special Controls and Test Mode Signals
- Block RAM, Static Random Access Memory (SRAM) Interface Signals
- Core Configuration Signals

Note:
1. AW is the Address width on the Master Bus which is 32/ 64-bits based on configuration
2. DW is the Data width on the Master bus which is 32/64-bits based on the configuration

Table 1: AHB Bus Interface Signals

Pin	Direction	Description
ahb_clk	In	AHB System Clock
ahb_reset_n	In	AHB System Reset (Active Low)
ahbmaster_hbusreq	Out	AHB Bus request
ahbmaster_hgrant	In	AHB Bus Grant
ahbmaster_haddr[AW-1:0]	Out	DWord Address
ahbmaster_hwdata[DW-1:0]	Out	AHB master write data
ahbmaster_hrdata[DW-1:0]	In	AHB master read data
ahbmaster_hwrite	Out	Write / Read Direction Indication
ahbmaster_hsize[2:0]	Out	Size (byte, half word or word)
ahbmaster_hburst[2:0]	Out	Burst Size
ahbmaster_hready	In	Ready signal
ahbmaster_htrans[1:0]	Out	Transfer type
ahbmaster_hresp[1:0]	In	Transfer response
ahb_intr	Out	Interrupt to the ARM
ahb_wkup	Out	Wakeup Indication to ARM
ahbtarget_hsel	In	Slave Select
ahbtarget_haddr[15:0]	In	DWord Address (256 bytes)
ahbtarget_hwdata[31:0]	In	Write Data
ahbtarget_hrdata[31:0]	Out	Read Data
ahbtarget_hwrite	In	Write / Read Direction Indication
ahbtarget_hsize[2:0]	In	Size (Byte, Half Word or Word)
ahbtarget_htrans[1:0]	In	Transfer Type
ahbtarget_hready_in	In	Slave Ready Input
ahbtarget_hready	Out	Slave Ready
ahbtarget_hresp[1:0]	Out	Transfer Response

Note: Target interface doesn’t support BURST transaction.

Table 2: OCP Bus Interface Signals

Pin	Direction	Description
clk_ocp	In	OCP System Clock.
OCPMaster_MAddr[AW-1:0]	Out	OCP Master read/write address.
OCPMaster_MCmd[2:0]	Out	Indicates the type of transaction that the OCP Master has initiated
OCPMaster_MData[DW-1:0]	Out	Write data from OCP Master to the slave
OCPMaster_MDataValid	Out	Is the qualifier for OCPMaster_MData
	In	Indicates that the OCP Slave has accepted the command
OCPMaster_SData[DW-1:0]	In	Read data from OCP Slave
OCPMaster_SDataAccept	In	Asserted by OCP slave to indicate that the current Master write data is accepted
OCPMaster_SResp[1:0]	In	Response signal for Master write transfers
OCPMaster_MByteEn[3:0]	Out	Byte enable from the Master for write/read transactions
OCPMaster_MBurstLength[4:0]	Out	Indicates the burst length of the transaction
OCPMaster_MBurstPrecise	Out	Indicates that the given burst length is precise
OCPMaster_MBurstSeq [2:0]	Out	Indicates the type of burst
OCPMaster_MBurstSingleReq	Out	Indicates the number of requests associated with the burst
OCPMaster_MDataLast	Out	Last Data of the burst
OCPMaster_MReqLast	Out	Last request in a burst
OCPMaster_SRespLast	In	Last response in a burst
OCPMaster_MDataByteEn[3:0]	Out	Write Byte enables the OCP slave during Data handshake phase
OCPMaster_max_burst_size_config	In	Configurable burst length
OCPSlave_MCmd[2:0]	In	Type of transaction from the Master
OCPSlave_MAddr[31:0]	In	Transfer address from Master
OCPSlave_MData[31:0]	In	Write data from OCP Master
OCPSlave_SCmdAccept	Out	Acceptance signal to the external OCP Master for the request phase
OCPSlave_SResp[1:0]	Out	Response signal from OCP Slave
OCPSlave_SData[31:0]	Out	Read data from OCP Slave
OCPSlave_MByteEn[3:0]	In	Byte Enable from OCP Master
OCPSlave_MReset_n	In	Reset signaling from OCP Master
OCPSlave_MRespAccept	In	Master accepts response

Table 3: AXI Bus Interface Signals

Pin	Direction	Description
aximst_arid[3:0]	Out	Read address ID. This signal is the identification tag for the read address group of signals.
aximst_araddr[AW-1:0]	Out	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[3:0]	Out	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_arsize[2:0]	Out	Burst size. This signal indicates the size of each transfer in the burst.
aximst_arburst[1:0]	Out	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
aximst_arvalid	Out	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, ARREADY, is high. 1 = address and control information valid 0 = address and control information not valid.
aximst_rid[3:0]	In	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 ARID value of the read transaction to which it is responding.
aximst_rdata[DW-1:0]	In	Read data.
aximst_rresp[1:0]	In	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
aximst_rlast	In	Read last. This signal indicates the last transfer in a read burst.
aximst_rvalid	In	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	Out	Read ready. This signal indicates that the Master can accept the read data and response information: 1= Master ready 0 = Master not ready.
axislv_awid[7:0]	In	Write address ID. This signal is the identification tag for the write address group of signals.

Pin	Direction	Description
axislv_awaddr[AW-1:0]	In	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.
axislv_awlen[3:0]	In	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.
axislv_awsize[2:0]	In	Burst size. This signal indicates the size of each transfer in the burst. Byte lane strobes indicate exactly which byte lanes to update.
axislv_awburst[1:0]	In	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
axislv_awvalid	In	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, AWREADY, goes HIGH.
axislv_awready	Out	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.
axislv_wid[7:0]	In	Write ID tag. This signal is the ID tag of the write data transfer. The WID value must match the AWID value of the write transaction.
axislv_wdata[DW-1:0]	In	Write data.
axislv_wstrb[DW/8-1:0]	In	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, WSTRB[n] corresponds to WDATA[(8 Þ n) + 7:(8 Þ n)].
axislv_wlast	In	Write last. This signal indicates the last transfer in a write burst.
axislv_wvalid	In	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.
axislv_wready	Out	Write ready. This signal indicates that the slave can accept the write data: 1 = slave ready

Pin	Direction	Description
		0 = slave not ready.
axislv_bid[7:0]	Out	Response ID. The identification tag of the write response. The BID value must match the AWID value of the write transaction to which the slave is responding.
axislv_bresp[1:0]	Out	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
axislv_bvalid	Out	Write response valid. This signal indicates that a valid write response is available: 1 = write response available 0 = write response not available.
axislv_bready	In	Response ready. This signal indicates that the Master can accept the response information. 1 = Master ready 0 = Master not ready
axislv_arid[7:0]	In	Read address ID. This signal is the identification tag for the read address group of signals.
axislv_araddr[31:0]	In	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.
axislv_arlen[3:0]	In	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.
axislv_arsize[2:0]	In	Burst size. This signal indicates the size of each transfer in the burst.
axislv_arburst[1:0]	In	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
axislv_arvalid	In	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, ARREADY, is high. 1 = address and control information valid 0 = address and control information not valid.
axislv_arready	Out	Read 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.

Pin	Direction	Description
axislv_rid[7:0]	Out	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 ARID value of the read transaction to which it is responding.
axislv_rdata[31:0]	Out	Read data. The read data bus can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits wide.
axislv_rresp[1:0]	Out	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
axislv_rlast	Out	Read last. This signal indicates the last transfer in a read burst.
axislv_rvalid	Out	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.
axislv_rready	In	Read ready. This signal indicates that the Master can accept the read data and response information: 1= Master ready 0 = Master not ready.
int_to_arm	Out	Interrupt to the ARM
cfg_mstid	In	Programmable ID for Master interface

Table 4: SD3.0/SDIO3.0/eMMC5.0 Interface

Pin	Direction	Description
sdif_cd_n	In	Active Low. Card Detection for single Slot (optional. Used for SD/SDIO interface only)
sdif_wp	In	Active High. SD Card Write Protect (optional. Used for SD/SDIO Interface only)
sdif_clkout	Out	Clock to Card (CLK)
rxclk_in	In	Clock looped back from PAD
sdif_cmdin sdif_cmdout sdif_cmdena	In Out Out	Command Input Command Output Command Output Enable
sdif_dat0in sdif_dat0out sdif_dat0en	In Out Out	Data0 Input Data0 Output Data0 Output Enable
sdif_dat1in sdif_dat1out sdif_dat1en	In Out Out	Data1 Input or Interrupt (for SDIO) Data1 Output Data1 Output Enable
sdif_dat2in sdif_dat2out sdif_dat2en	In Out Out	Data2 Output or Read Wait (optional) Data2 Output Data2 Output Enable

Pin	Direction	Description
sdif_dat3in sdif_dat3out sdif_dat3en	In Out Out	Data3 Input Data3 Output Data3 Output Enable
sdif_dat4in sdif_dat4out sdif_dat4en	In Out Out	Data4 Input Data4 Output Data4 Output Enable
sdif_dat5in sdif_dat5out sdif_dat5en	In Out Out	Data5 Input Data5 Output Data5 Output Enable
sdif_dat6in sdif_dat6out sdif_dat6en	In Out Out	Data6 Input Data6 Output Data6 Output Enable
sdif_dat7in sdif_dat7out sdif_dat7en	IN Out Out	Data7 Input Data7 Output Data7 Output Enable

Note: CMD/DATA output enables are active high signals

Table 5: Power Control Signals (Used for SD/SDIO Mode only)

Pin	Direction	Description
sdhc_ledcontrol	Out	LED ON: To Caution the user not to remove the card while the SD card is being accessed.
sdhc_sdbuspower	Out	Control SD Card Power Supply.
sdhc_sdbusvoltage[2:0]	Out	SD Bus voltage select.
sdhc_1p8vsigenable	Out	1.8V Signaling Enable
sdhc_driverstrength[1:0]	Out	Driver Strength Select 00b Driver Type B is Selected 01b Driver Type A is Selected 10b Driver Type C is Selected 11b Driver Type D is Selected

Table 6: Clock, Special Controls and Test Mode Signals

Pin	Direction	Description
xin_clk	In	This clock input is used to generate eMMC/SD Clock. For maximum efficiency this should be around 200 MHz for eMMC or 208MHz (for SD3.0).
corectrl_itapdlyena	In	Used to enable selective Tap delay line on the Looped back eMMC/SD Clock (rxclk_in). This signal along with the corectrl_itapdlysel[4:0] selects the amount of delay to be inserted on the line.

Pin	Direction	Description
		When Tuning is enabled (for HS200/SDR104), this signal is ignored and internal controls are used instead. This should not be asserted when operating in DS mode.
corectrl_itapdlysel[4:0]	In	Selects one of the 32 Taps on the rxclk_in line. This is effective only when corectrl_itapdlyena is asserted and Tuning is not enabled.
corectrl_itapchgwin	In	This is used to gate the output of the Tap Delay lines so as to avoid glitches being propagated into the Core. This signal should be asserted few clocks before the corectrl_itapdlysel changes and should be asserted for few clocks after.
corectrl_otapdlyena	In	Used to enable the selective Tap delay on the card_clk so as to generate the delayed card_clk. This is used to latch the CMD/DAT outputs to generate delay on them w.r.t CLK going out. This signal along with corectrl_otapdlysel[3:0] selects the amount of delay to be inserted on the Clock line. This signal should not be asserted when operating in DS mode
corectrl_otapdlysel[3:0]	In	Selects one of the 16 Taps on the sdcard_clk. This is effective only when corectrl_otapdlyena is asserted.
test_mode	In	Test mode signal is used for DFT purpose. Muxes in the AXI_reset_n signal for all internally generated resets. (Active High)
scan_mode	In	Scan Mode signal for selecting Scan Clocks for internally generated clocks
scan_clk1	In	Scan Clock#1 used to mux in for the internally generated sd_clk
scan_clk2	In	Scan Clock#2 used to mux in for the final rxclk_in (after the tap delay etc)

Table 7: Block RAM (SRAM) Interface Signals

Pin	Direction	Description
sram_clka	Out	Clock for PORT A
sram_addra [N-1:0]	Out	Address bus for PORT A. The width of the Address bus is based on the size of the SRAM (SDHC_BUFFER_SIZE)
sram_writea	Out	Write Enable for PORT A
sram_reada	Out	Read Enable for PORT A
sram_wrdata[DW-1:0]	Out	Write Data for PORT A.

sram_rddataa [DW-1:0]	In	Read Data from SRAM on PORT A. N is based on the SDHC_MSTAXI_DW parameter
sram_clkb	Out	Clock for PORT B
sram_addrb [N-1:0]	Out	Address bus for PORT B. The width of the Address bus is based on the size of the SRAM (SDHC_BUFFER_SIZE)
sram_writeb	Out	Write Enable for PORT B
sram_readb	Out	Read Enable for PORT B
sram_wrdatab [DW-1:0]	Out	Write Data for PORT B.
sram_rddatab [DW-1:0]	In	Read Data from SRAM on PORT B.

Table 8: Core Configuration Signals

Pin	Direction	Description
corecfg_tuningcount[5:0]	In	Configures the Number of Taps (Phases) of the rxclk_in that is supported. The Tuning State machine uses this information to select one of the Taps (Phases) of the rxclk_in during the Tuning Procedure.
corecfg_timeoutclkfreq[5:0]	In	Timeout Clock Frequency Suggested Value is 1. (KHz or MHz). Internally the 1msec /1usecTimer is used for Timeout Detection. The 1msec Timer is generated from the xin_clk.
corecfg_timeoutclkunit	In	Timeout Clock Unit Suggested value is 1'b1 to Select MHz Clock.
corecfg_baseclkfreq[7:0]	In	Base Clock Frequency for SD Clock. This is the frequency of the xin_clk.
corecfg_maxblklength[1:0]	In	Max Block Length Maximum Block Length supported by the Core/Device 00: 512 (Bytes) 01: 1024 10: 2048 11: Reserved
corecfg_8bitsupport	In	8-bit Support for Embedded Device Suggested Value is 1'b1 (The Core supports 8-bit Interface). Optionally can be set to 1'b0 if the Application supports only 4-bit SD Interface.
corecfg_adma2support	In	ADMA2 Support Suggested Value is 1'b1 (The ADMA2 is supported by Core). Optionally can be set to 1'b0 if the application doesn't want to support ADMA2 Mode
corecfg_highspeedsupport	In	High Speed Support Suggested Value is 1'b1 (The High Speed mode is supported by Core).

Pin	Direction	Description
corecfg_sdmasupport	In	SDMA Support Suggested Value is 1’b1 (The SDMA is supported by Core). Optionally can be set to 1’b0 if the application doesn’t want to support SDMA Mode
corecfg_suspressupport	In	Suspend/Resume Support Suggested Value is 1’b1 (The Suspend/Resume is supported by Core). Optionally can be set to 1’b0 if the application doesn’t want to support Suspend/Resume Mode
corecfg_3p3voltsupport	In	3.3V Support Suggested Value is 1’b1 as the 3.3 V is the default voltage on the SD Interface.
corecfg_3p0voltsupport	In	3.0V Support Should be set based on whether 3.0V is supported on the SD Interface.
corecfg_1p8voltsupport	In	1.8V Support Suggested Value is 1’b1 (The 1.8 Volt Switching is supported by Core). Optionally can be set to 1’b0 if the application doesn’t want 1.8V switching (SD3.0)
corecfg_64bitsupport	In	64-bit System Bus Support This should be set based on the System Address Bus. When set to 1’b0 the Core supports only 32-bit System Bus. When set to 1’b1 the Core supports 64-bit System Address.
corecfg_asyncintrsupport	In	Asynchronous Interrupt Support Suggested Value is 1’b1 (The Core supports monitoring of Asynchronous Interrupt)
corecfg_slottype[1:0]	In	Slot Type Should be set based on the final product usage 00 - Removable SCard Slot 01 - Embedded Slot for One Device 10 - Shared Bus Slot 11 – Reserved
corecfg_sdr50support	In	SDR50 Support Suggested Value is 1’b1 (The Core supports SDR50 mode of operation) Optionally can be set to 1’b0 if the application doesn’t want to support SDR50
corecfg_sdr104support	In	SDR104 Support Suggested Value is 1’b1 (The Core supports SDR104 mode of operation) Optionally can be set to 1’b0 if the application doesn’t want to support SDR104
corecfg_ddr50support	In	DDR50 Support

Pin	Direction	Description
		Suggested Value is 1’b1 (The Core supports DDR50 mode of operation) Optionally can be set to 1’b0 if the application doesn’t want to support DDR50
corecfg_hs400support	In	HS400 Support Suggested Value is 1’b1 (The Core supports HS400 Mode). This applies only to eMMC5.0 mode. This should be set to 1’b0 for SD3.0 mode Optionally can be set to 1’b0 if the application doesn’t want to support HS400
corecfg_adriversupport	In	Driver Type A Support This bit should be set based on whether Driver Type A for 1.8 Signaling is supported or not.
corecfg_cdriversupport	In	Driver Type C Support This bit should be set based on whether Driver Type C for 1.8 Signaling is supported or not.
corecfg_ddriversupport	In	Driver Type D Support This bit should be set based on whether Driver Type D for 1.8 Signaling is supported or not.
corecfg_retuningtimercnt[3:0]	In	Timer Count for Re-Tuning This is the Timer Count for Re-Tuning Timer for Re-Tuning Mode 1 to 3. Setting to 4’b0 disables Re-Tuning Timer.
corecfg_tuningforsdr50	In	Use Tuning for SDR50 This bit should be set if the Application wants Tuning be used for SDR50 Modes. The Core operates with or without tuning for SDR50 mode as long as the Clock can be manually tuned using tap delay.
corecfg_retuningmodes[1:0]	In	Re-Tuning Modes Should be set to 2’b00 as the Core supports only the Mode0 Retuning.
corecfg_spisupport	In	SPI Mode Support Suggested Value is 1’b1 (The Core supports SPI mode of operation) Optionally can be set to 1’b0 if the application doesn’t want to support SPI Mode
corecfg_spiblkmode	In	SPI Block Mode Reserved and should be set to1’b0
corecfg_type4support	In	Driver Type 4 Support This bit should be set to 1'b1 if the Host Controller supports Type4 Drive Strength (eMMC5.0), otherwise it should be set to 0.
corecfg_initpresetval[12:0]	In	Preset Value for Initialization.
corecfg_dsppresetval[12:0]	In	Preset Value for Default Speed

Pin	Direction	Description
corecfg_hsppresetval[12:0]	In	Preset Value for High Speed
corecfg_sdr12presetval[12:0]	In	Preset Value for SDR12
corecfg_sdr25presetval[12:0]	In	Preset Value for SDR25
corecfg_sdr50presetval[12:0]	In	Preset Value for SDR50
corecfg_sdr104presetval[12:0]	In	Preset Value for SDR104
corecfg_ddr50presetval[12:0]	In	Preset Value for DDR50
corecfg_hs400presetval[12:0]	In	Preset Value for HS400
corecfg_maxcurrent1p8v[7:0]	In	Maximum Current for 1.8V
corecfg_maxcurrent3p0v[7:0]	In	Maximum Current for 3.0V
corecfg_maxcurrent3p3v[7:0]	In	Maximum Current for 3.3V
corecfg_asyncwkupena	In	Determines the Wakeup Signal Generation Mode. **0: Synchronous Wakeup Mode:** The xin_clk has to be running for this mode. The Card Insertion/Removal/Interrupt events are detected synchronously on the xin_clk and the Wakeup Event is generated. The Assertion and deassertion of the wakeup Event signal synchronous to xin_clk. **1: Asyncrhonous Wakeup Mode:** The xin_clk and the host_clk can be stopped in this mode and the Wake up Event is asynchronously generated based on the Card Insertion/Removal/Interrupt Events. The Assertion and de-assertion of the wakeup Event signal is asynchronous.

2.5 IP Deliverables for RTL Version

- Verilog HDL of the IP Core
- Synthesis scripts
- Test environment and test scripts
- User guide

2.6 Verification Environment

This section provides information about the architecture of the SD Host Controller Verification Environment.

The eMMC5.1 Host Controller Design Under Test (DUT) is written in synthesizable Verilog. On the processor side it interfaces with AXI/AHB/OCP Master BFM and AXI/AHB/OCP Slave BFM. On the device side it interfaces with a user selectable device BFM (SD, SDIO, eMMC). The Host Controller Capability is selected by connecting the capability pins to power and ground. The device side interface is 8 bits wide bi-directional data lane supporting DDR (Double Data Rate). The width of the bus is user selectable (1, 4 or 8). The data can be sent or received either by processor input and output or by DMA. The data flow can be aborted by the processor side sending an abort command. The data transfer goes through a ping-pong buffer which achieves back to back frames transfer.

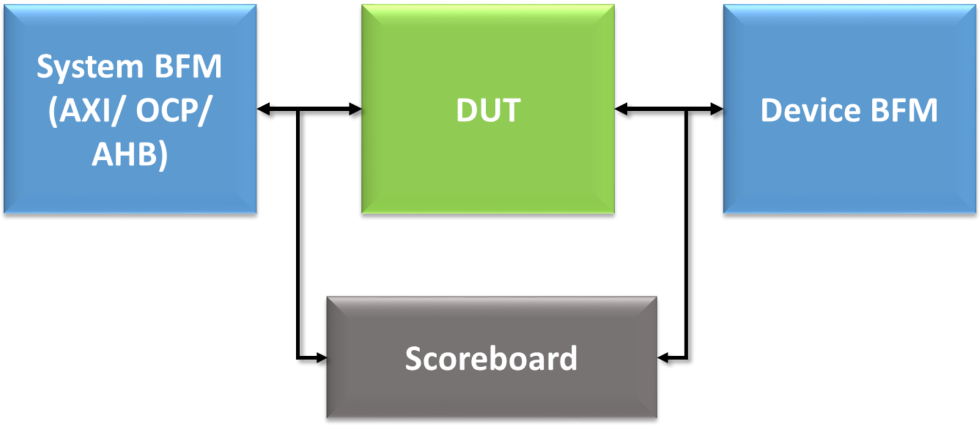

Figure 3: Verification Environment

3 eMMC 5.1 Host PHY Datsheet

3.1 Overview

The ACS's eMMC5.1 PHY is to be fully compliant PHY layer for JEDEC eMMC5.1 when rectified and eMMC5.1 JESD84-B50 specification. It is backward compliant with eMMC4.51 and earlier versions of the specifications. This allows the designers of the SOC to easily support the EMMC interface and optimize the performance and power while maintaining interoperability with eMMC5.0 and eMMC5.1 devices.

3.2 Features

The following are the high level features of the ACS eMMC5.1 PHY:

- Designed for seamless integration with ACS's eMMC5.1 Host Controller.
- Supports HS400, HS200, DDR50 and legacy operating modes.
- Includes EMMC I/O PADS with ESD protection structures.
- The ACS EMMC I/O PADS are designed to meet eMMC5.1 HS400 specifications.
- Integrate a Master Slave DLL for tuning the Receive clock (in HS200), STRB (in HS400) modes.
- Delay Chain based clock tuning for TX Clock, RX Clock and STRB Clock.
- Built-in diagnostics for monitoring the DLL.
- ACS eMMC5.1 PHY is available in the following TSMC technology nodes:
    - 16nm FF plus v1.0 process (GL and LL)
    - 28nm HPM, HPC and LP
    - 40nm LP
- The design is intended for core supply VCORE +/-10% and I/O supply VCCQ +/-10%.

3.3 Architecture

3.3.1 1HS400 Overview

This Arasan IP consists of hardened PHY IP and RTL block code. The hard-macro consists of analog IPs, such as eMMC 5.1 interface Pads, Impedance Calibration Pad, an analog DLL, and the DLL wrapper. The RTL Block code includes Arasan's Host/Device controller.

To assist with eMMC 5.1 IP integration, Arasan provides all of the back-end views of eMMC 5.1 GPIO Pads and CALIO Pad integrated with TSMC ESD protection structure for I/O VDDQ, VSSQ and Power Clamps.

The ACS eMMC5.1 PHY consists of two major sections DFE and an AFE.

The ACS eMMC5.1 PHY DFE Contains:

- The interface.to ACS's eMMC5.1 Host Controller and Main SOC command processor

- The ACS's Host Controller supports HS400, HS200, DDR50 and legacy data rates.
- Includes the Input / Output flops to support both SDR and DDR operation on the Data Lines. Thus alleviating the timing responsibilities from the eMMC5.1 Host Controller.
- Includes the DLL clocks phase selection and MUXING logic.
- Includes data buffering FIFO and EMMC I/O data synchronizing Flops.

The eMMC5.1 PHY AFE contains:

- EMMC PADS with integrated ESD protection {CMD line, DAT [0:7] lines, CLK line and STRB line}.
- CALIO PAD to automatically calibrate the source and sink impedance of EMMC I/O.
- Analog DLL to provide the following functionality.
- Generates 32 phases equally spaced of STRB for data shifting in HS400 (for reads).
- Generates 32 phases equally spaced of RX clock for Tuning function in the HS200 and HS400 mode of operation.
- Generates 32 phases equally spaced of TX clock for Tuning to support various hold requirements on the EMMC CMD/DAT lines at various mode of operation.
- The DLL clock frequency can be programmed to support any constant clock frequency in the range of 50MHz to 275MHz.

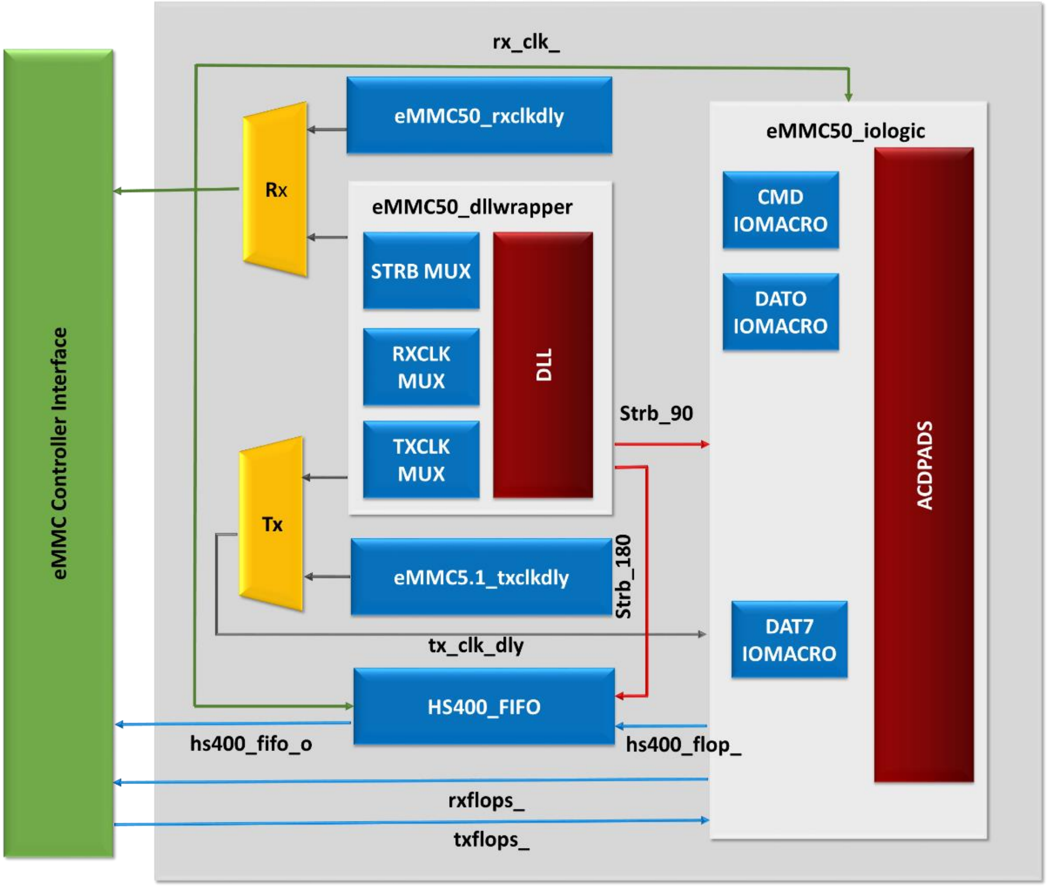

Figure 4: HS400 Block Diagram

The overall pinlist of the eMMC 5.1 macro is shown in Figure 4.

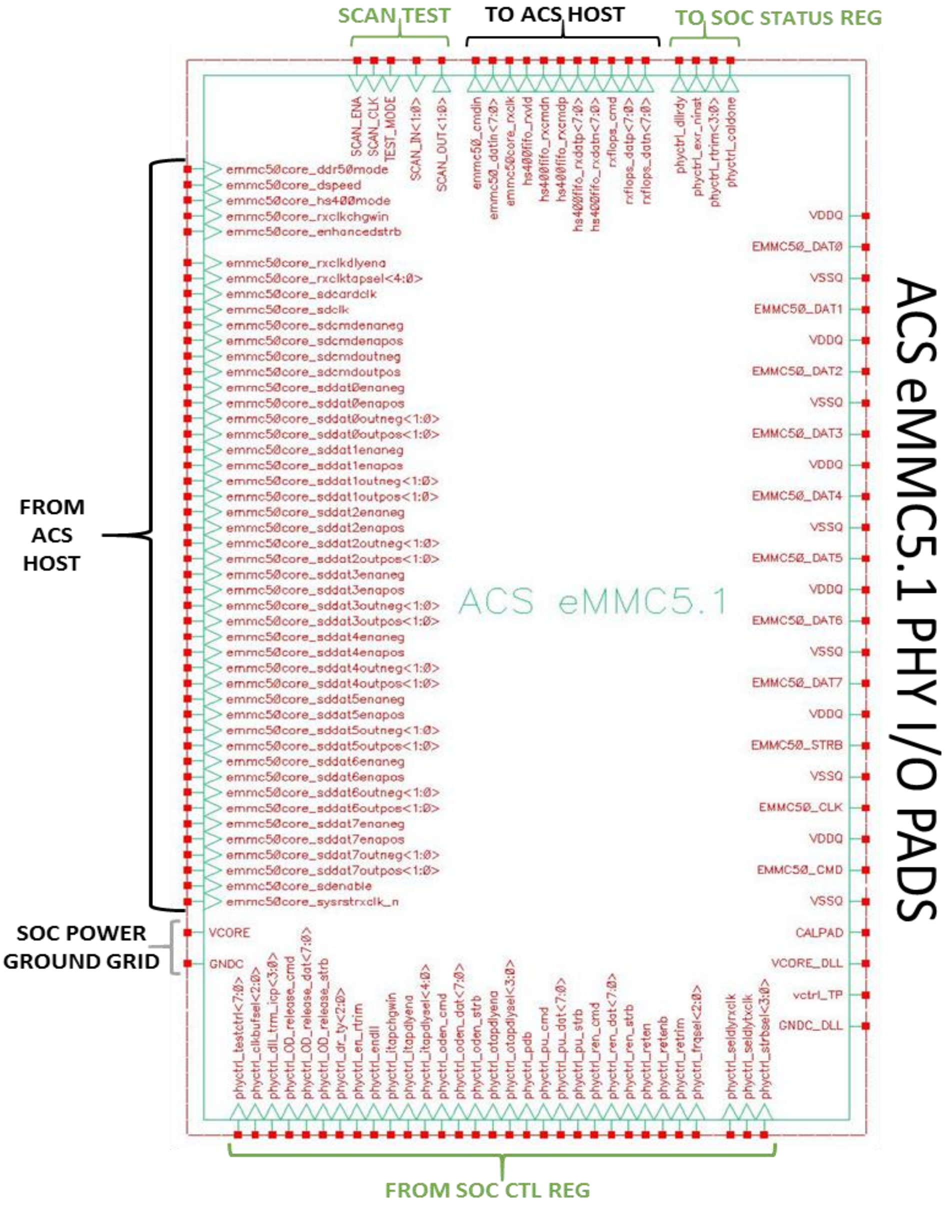

Figure 5: eMMC5.1 PHY I/O Diagram

3.4 Signal Interface

Table 9: eMMC5.1 Pin Description

Signal Name	DIR PWR	Description
ACS eMMC5.1 I/O PADS		
VDDQ	POWER	I/O Power Pad
EMMC50_DAT0	I/O VDDQ	eMMC5.1 data bit 0 port
VSSQ	GROUND	I/O Ground Pad
EMMC50_DAT1	I/O VDDQ	eMMC5.1 data bit 1 port
VDDQ	POWER	I/O Power Pad
EMMC50_DAT2	I/O VDDQ	eMMC5.1 data bit 2 port
VSSQ	GROUND	I/O Ground Pad
EMMC50_DAT3	I/O VDDQ	eMMC5.1 data bit 3 port
VDDQ	POWER	I/O Power Pad
EMMC50_DAT4	I/O VDDQ	eMMC5.1 data bit 4 port
VSSQ	GROUND	I/O Ground Pad
EMMC50_DAT5	I/O VDDQ	eMMC5.1 data bit 5 port
VDDQ	POWER	I/O Power Pad
EMMC50_DAT6	I/O VDDQ	eMMC5.1 data bit 6 port
VSSQ	GROUND	I/O Ground Pad
EMMC50_DAT7	I/O VDDQ	eMMC5.1 data bit 7 port
VDDQ	POWER	I/O Power Pad
EMMC50_STRB	I/O VDDQ	eMMC5.1 strobe port
VSSQ	GROUND	I/O Ground Pad
EMMC50_CLK	OUT VDDQ	eMMC5.1 clock port
VDDQ	POWER	I/O Power Pad
EMMC50_CMD	I/O VDDQ	eMMC5.1 CMD port
VSSQ	GROUND	I/O Ground Pad
CALPAD	I/O VDDQ	ACS CALIO PAD connects to 10k +/- 1% resistor
vctrl_TP	I/O VCORE	ACS analog DLL charge pump Test pad
GNDC_DLL	GROUND	ACS analog DLL dedicated ground
VCORE_DLL	POWER	ACS analog DLL dedicated VCORE power supply
ACS eMMC5.1 HOST CONTROLLER INTERFACE SIGNALS		
emmc50core_sdcardclk	IN VCORE	SDCARD/EMMC Clock. This the input signal to EMMC50_CLK PAD. Power on Default 1b'0
emmc50core_sdclk	IN VCORE	Transmit Clock. TxCLK input to the analog DLL. The analog DLL generates 32 equally spaced phases. An appropriate phase is selected and sent to the data output Flops to maintain Hold requirement on CMD/DAT lines. Power on Default 1b'0
emmc50core_rxclk	OUT VCORE	Receive Clock. RxCLK input to the analog DLL. The analog DLL generates 32 equally spaced phases. One Phase is selected by Controller 2 and is used in the Receive path. Power on Default 1b'0

emmc50core_ddr50mode	IN VCORE	DDR50 Mode Indication. DDR50 Mode indication signal from Controller. Power on Default 1b’0
emmc50core_rxclkdlyena	IN VCORE	Receive Clock Delay Enable. Enables delay the Receive clock. The actual delay (tap select) is controlled by emmc50core_rxclktapsel bus. Power on Default 1b’0
emmc50core_rxclktapsel<4:0>	IN VCORE	Receive Clock Tap Select. Select bus of the receive clock DLL RxCLK phases. Power on Default 5b’00000
emmc50core_rxclkchgwin	IN VCORE	Receive Clock Change Window. This signal is asserted by the controller, when the Tap is changing. It creates a window which is used by the PHY to gate off the selected Receive clock to avoid unwanted glitches. Power on Default 1b’0
emmc50core_sdenable	IN VCORE	SD/EMMC Interface Enable. This is asserted by the Controller to enable the EMMC Interface. When de-asserted, the EMMC Interface is tri-stated. Power on Default 1b’0
emmc50core_sdcmdoutpos	IN VCORE	CMD_OUT. CMD output is launched on the positive edge of the TX clock by the Controller. It is sent EMMC50_CMD IO. Power on Default 1b’0
emmc50core_sdcmdenapos	IN VCORE	CMD_ENA. CMD output enable that is launched on the positive edge of the TX clock by the Controller. It is used to enable the CMD line of the IO. Power on Default 1b’0.
emmc50core_sddatNoutpos[1:0] (N=0 to 7)	IN VCORE	DATP_OUT. Two bits of DATP (DDR mode uses both bits, SDR mode uses bit0 only) output is launched on the positive edge of the TX clock by the Controller. The selected data is sent to EMMC50_Dn (n=0 to 7) IO. Power on Default 2b’00
emmc50core_sddatNenapos (N=0 to 7)	IN VCORE	DATP_ENA. DATP output enable that is launched on the positive edge of the TX clock by the Controller. Enables the DATN line of the IO. Power on Default 1b’0
emmc50core_sdcmdoutneg	IN VCORE	CMD_OUT. CMD output that is launched on the negative edge of TX clock by the Controller. Drives the CMD line of the IO and is used only when operating in Default Speed. Power on Default 1b’0
emmc50core_enhancedstrb	IN VCORE	Enhanced STRB. Enhanced strobe mode Power on Default 1b’0
emmc50core_sddatNoutneg[1:0] (N=0 to 7)	IN VCORE	DATN_OUT. Two bits of DATN (DDR mode uses both the bits, SDR mode uses bit0 only) is launched on the negative edge of TX clock by the Controller. The selected data is sent to EMMC50_Dn (n=0 to 7) IO in Default Speed mode. Power on Default 2b’00.
emmc50core_sddatNenaneg (N=0 to 7)	IN VCORE	DATN_ENA. DATN output enable that is launched on the negative edge of TX clock by the Controller. Is

		used to enable the DATN line of the IO when operating in Default Speed. Power on Default 1b’0.
emmc50core_sysrstrxclk_n	IN VCORE	Power on Default 1b’0
emmc50_cmdin	OUT VCORE	CMD Line Output. CMD line output from the PHY to the Controller. Used in the Controller for Software Monitoring. Power on Default 1b’0.
emmc50core_dspeed	IN VCORE	Power on Default 1b’0.
emmc50core_hs400mode	IN VCORE	Power on Default 1b’0.
emmc50_datin<7:0>	OUT VCORE	DATN Lines Output. These are the DAT line output from the PHY to the Controller. Used in the Controller for Software Monitoring. Power on Default 8b’00000000.
rxflops_cmd	OUT VCORE	CMD line. CMD line flopped in the PHY on the positive edge of the RX clock. Power on Default 1b’0.
rxflops_datp<7:0>	OUT VCORE	DATP lines. These are the DATP lines flopped in the PHY on the positive edge of the RX clock. Power on Default 8b’00000000
rxflops_datn<7:0>	OUT VCORE.	DATN lines. DATN lines flopped in the PHY on the negative edge of the RX clock. Power on Default 8b’00000000.
hs400fifo_rxcmdp	OUT VCORE	CMD Line. CMD line that is flopped on the positive edge of the STRB (centered) and then synchronized to RX clock. Power on Default 1b’0.
hs400fifo_rxcmdn	OUT VCORE	CMD Line. CMD line that is flopped on the negative edge of the STRB (centered) and then synchronized to RX clock. Power on Default 1b’0
hs400fifo_rxvld	OUT VCORE	Data Valid. Hs400 Data Valid Indication. Power on Default 1b’0
hs400fifo_rxdatp<7:0>	OUT VCORE	DAT Lines. DATN lines that are flopped on the positive edge of the STRB (centered) and then synchronized to RX clock. Power on Default 8b’00000000
hs400fifo_rxdatn<7:0>	OUT VCORE	DAT Lines. DATN lines that are flopped on the negative edge of the STRB (centered) and then synchronized to RX clock. Power on Default 8b’00000000
SOC Interface Signals		
phyctrl_endll	IN VCORE	Enable DLL. Enables the analog DLL circuits. Power on Default 1b’0
phyctrl_exr_ninst	OUT VCORE	External Resistor on CALIO absent. Indicates trim cycle started and external resistor is absent. Power on Default 1b’0.
phyctrl_pdb	IN VCORE	CALIO S/M power down bar. SOC asserts after power up sequence is completed. Power on Default 1b’0.
phyctrl_dr_ty<2:0>	IN VCORE	Drive Source/Sink impedance programming

		‘1b000’ → 50 ohms ‘1b001’ → 33 Ohms ‘1b010’ → 66 Ohms ‘1b011’ → 100 Ohms ‘1b100’ → 40 Ohms Power on Default 3b’000
phyctrl_retrim	IN VCORE	Start CALIO calibration cycle. At positive edge initiates CALIO calibration cycle. Power on Default 1b’0
phyctrl_en_rtrim	IN VCORE	CALIO enable. Enables CALIO, If enabled CALIO will start calibration cycle at phyctrl_pdb positive edge. Power on Default 1b’1
phyctrl_dll_trm_icp<3:0>	IN VCORE	Analog DLL’s Charge Pump Current Trim. Programs the analog DLL loop gain. Power on Default 4b’1000
phyctrl_dllrdy	IN VCORE	DLL ready. Indicates that DLL loop is locked. Power on Default 1b’0
phyctrl_oden_strb	IN VCORE	Open Drain Enable on STRB line. Power on Default 1b’0
phyctrl_oden_cmd	IN VCORE	Open Drain Enable on CMD line. Power on Default 1b’0
phyctrl_oden_dat<7:0>	IN VCORE	Open Drain Enable on DAT lines. Power on Default 8b’00000000
phyctrl_ren_strb	IN VCORE	Enable pull up/down on the STRB line. If phyctrl_pu_strb is high a week pull up is enabled on STRB line, if low week pull down is enabled on STRB line. Power on Default 1b’0
phyctrl_ren_cmd	IN VCORE	Enable pull up/down on CMD line. If phyctrl_pu_cmd is high week pull up is enabled on CMD line, if low week pull down is enabled on CMD line. Power on Default 1b’1
phyctrl_ren_dat<7:0>	IN VCORE	Enable pull up/down on DAT Lines. If phyctrl_pu_dat<7:0> is high week pull up is enabled on DATA lines, if low week pull down is enabled on DATA lines. Power on Default 8b’11111111
phyctrl_pu_strb	IN VCORE	Enable pull up on STRB line. If phyctrl_ren_strb is high week pull up is enabled on STRB line. Power on Default 1b’0
phyctrl_pu_cmd	IN VCORE	Enable pull up on CMD line. If phyctrl_ren_cmd is high week pull up is enabled on CMD line. Power on Default 1b’1
phyctrl_pu_dat<7:0>	IN VCORE	Enable pull up on DAT lines. If phyctrl_ren_dat[7:0] is high week pull up is enabled on DATA lines. Power on Default 8b’11111111
phyctrl_itapdlyena	IN VCORE	Input Tap Delay Enable. This is used for the manual control of the RX clock Tap Delay in non HS200/HS400 modes. Power on default 1b’0

phyctrl_itapdlysel<4:0>	IN VCORE	Input Tap Delay Select. Manual control of the RX clock Tap Delay in the non HS200/HS400 modes. Power on Default 5b’00000
phyctrl_itapchwin	IN VCORE	Input Tap Change Window. It gets asserted by the controller while changing the phyctrl_itapdlysel. Used to gate of the RX clock during switching the clock source while tap is changing to avoid clock glitches. Power on Default 1b’0
phyctrl_otapdlyena	IN VCORE	Output Tap Delay Enable. Enables manual control of the TX clock tap delay, for clocking the final stage flops for maintaining Hold requirements on EMMC Interface. Power on Default 1b’0.
phyctrl_otapdlysel<3:0>	IN VCORE	Output Tap Delay Select. Manual control of the TX clock tap delay for clocking the final stage flops for maintaining Hold requirements on EMMC Interface. Power on Default 4’0000
phyctrl_frqsel<2:0>	IN VCORE	Select the frequency range of DLL operation: 3b’000 => 200MHz to 170 MHz 3b’001 => 170MHz to 140 MHz 3b’010 => 140MHz to 110 MHz 3b’011 => 110MHz to 80MHz 3b’100 => 80MHz to 50 MHz 3b’101 => 275Mhz to 250MHz 3b’110 => 250MHz to 225MHz 3b’111 => 225MHz to 200MHz Power on Default 3b’000.
phyctrl_seldlyrxclk	IN VCORE	Select the Delay chain based rxclk. Enables the RX clock based delay chain rather than analog DLL based delay chain. Power on Default 1b’0.
phyctrl_seldlytxclk	IN VCORE	Select the Delay chain based txclk. Enables the TX clock based delay chain rather than analog DLL based delay chain. Power on Default 1b’0.
phyctrl_reten	IN VCORE*	Retention Mode Enable. Retention mode is enabled before going to sleep mode and turn VCORE off. The EMMCIOs will retain its input programming state in sleep mode. Generated in the always on VCORE power domain. Power on Default 1b’0.
phyctrl_retenb	IN VCORE*	Retention Mode Enable Bar. The inverted phase of phyctrl_reten should be generated in the always on VCORE power domain. Power on Default 1b’1.
phyctrl_OD_release_strb	IN VCORE	Disable an internal 4.7K pull up resistor on STRB line in open drain mode. Power on Default 1b’0.
phyctrl_OD_release_cmd	IN VCORE	Disable an internal 4.7K pull up resistor on CMD line in open drain mode. Power on Default 1b’0

phyctrl_OD_release_dat<7:0>	IN VCORE	Disable an internal 4.7K pull up resistor on data lines in open drain mode. Power on Default 8b’00000000
phyctrl_strbsel<3:0>	IN VCORE	Select the Four Taps for each of STRB_90 and STRB_180 Outputs. phyctrl_strbsel[3:2] selects one of the four for STRB_180 and phyctrl_strbsel[1:0] selects the four taps for STRB_90. Power on default 4b’0000
phyctrl_rtrim<3:0>	OUT VCORE	CALIO Calibration Result. Holds the content of CALIO Impedance Calibration Result. Power on default 4b’1110.
phyctrl_caldone	OUT VCORE	STATUS, indicate that CALIO Calibration is completed successfully. Power on default 1b’0.
TEST_MODE	IN VCORE	Enables DFT Mode for eMMC5 PHY DFE. Power on default 1b’0
SCAN_ENA	IN VCORE	Enables Scan Mode for eMMC5 PHY DFE. Power on default 1b’0
SCAN_CLK	IN VCORE	Scan Clock for DFT Mode of eMMC5 PHY DFE. Power on default 1b’0
SCAN_IN<1:0>	IN VCORE	Two Parallel Scan Chains Scan Inputs of eMMC5 PHY DFE. Power on default 2b’00
SCAN_OUT<1:0>	OUT VCORE	Two Parallel Scan Chains Scan Outputs of eMMC5 PHY DFE. Power on default 2b’00
phyctrl_clkbufsel<2:0>	IN VCORE	Clock Delay Buffer Select. Selects one of the eight taps in the CLK Delay Buffer based on PVT variation. Power on default 3b’000
GNDC	GROUND	DFE and EMMC I/O low Voltage Logic ground return.
VCORE	POWER	DFE and EMMC I/O low Voltage Logic power supply.
phyctrl_testctrll<7:0>	IN VCORE	ACS eMMC50_PHY test control. 8’b00010000 →Test EMMC IOs sink impedance 8’b00010001 →Test EMMC IOs source impedance 8’b00100000 →Test RX clock phases on data lines. ph<0>→EMMC50_DAT<0>, ph<4> →EMMC50_DAT<1> ph<8> →EMMC50_DAT<2>, ph<12> →EMMC50_DAT<3> ph<16> →EMMC50_DAT<4>, ph<20> →EMMC50_DAT<5> ph<24> →EMMC50_DAT<6>, ph<28> →EMMC50_DAT<7> 8’b00110000 →Test TX clock phases on data lines. ph<0> →EMMC50_DAT<0>, ph<4> →EMMC50_DAT<1> ph<8> →EMMC50_DAT<2>, ph<12> →EMMC50_DAT<3> ph<16> →EMMC50_DAT<4>, ph<20> →EMMC50_DAT<5> ph<24> →EMMC50_DAT<6>, ph<28> →EMMC50_DAT<7>

		8’b01000000 →Test STRB clock phases on data lines. ph<0> →EMMC50_DAT<0>, ph<4> →EMMC50_DAT<1> ph<8> →EMMC50_DAT<2>, ph<12> →EMMC50_DAT<3> ph<16> →EMMC50_DAT<4>, ph<20> →EMMC50_DAT<5> ph<24> →EMMC50_DAT<6>, ph<28> →EMMC50_DAT<7> Power on default 8b’00000000

3.5 DC Characteristics

The following tables summarize the DC characteristics and characterization conditions of the eMMC HS400 I/O pads. This is an example table with values for TSMC 28HPM process node (Vcore = 0.8V for 16FF+, 0.9V for 28HPM/HPC, 1.05 for 28LP and 1.1V for 40LP).

Table 10: Recommended Operating Conditions

Parameter		Min	Nom	Max	Units
VCORE	Pre driver voltage	0.81	0.9	0.99	V
VCCQ (1.8/3.3V)	Post driver voltage	1.72/2.7	1.8/3.3	1.98/3.6	V
VCCQ (1.8V only)		1.62	1.8	1.98	V
TJ	Junction temperature	-20	25	100	°C
VIMAX (1.8/3.3V)	Maximum input voltage			3.7	V
VIMAX (1.8V only)				2.1	V

Table 11: DC Characteristics

Parameter		Min	Max	Units
VIL	Input Low Voltage	VSSQ-0.3	0.25* VCCQ	V
VIH	Input High Voltage	0.625*VCCQ	VCCQ +0.3	V
II	Input Leakage Current		2	µA
IOZ	Tri-State Output Leakage Current		2	µA
VOL	Output Low Voltage		0.125* VCCQ	V
VOH	Output High Voltage	0.75* VCCQ		V

3.5.1 Driver Strength Support

The following drive strengths are supported and are programmable.

Table 12: Drive Strength

Driver Type	Nominal Impedance	Relative Driving Capability	Remark
0	50 Ohms	x1	Default driver type, supports up to 200MHz operation
1	33 Ohms	x1.5	Supports up to 200MHz operation
2	66 Ohms	x0.75	The weakest driver that supports up to 200MHz operation.
3	100 Ohms	x0.5	For Low noise and low EMI systems, Minimum operating frequency is system dependent.
4	40 ohms	x1.2	Supports up to DDR 200MHz operation.

3.6 Deliverables

- GDSII database
- LVS Netlist
- Physical Abstract Model (LEF)
- Timing Models
- Behavioral Models
- Design Integration Guide
- Technical Documents

4 eMMC 5.1 Device Controller Datasheet

4.1 Overview

eMMC 5.1 (JEDEC Item # JC-64.1-67.14) is the latest specification defined by JEDEC and is designed to meet the requirements for next level of high performance data transfer for mobile electronic products. With its low-pin count, higher bandwidth, multiple boot mechanisms, and content security features, eMMC 5.1 provides an easy migration path and greatly simplifies system design for new products.

Arasan's eMMC 5.1 Device Controller IP is compliant to the latest eMMC specification. The controller provides a bandwidth of up to 3.2 Gbps (400 MB/s) in HS400 DDR mode running with 200 MHz clock. A NAND Flash controller can be connected to the eMMC controller. In such an implementation, the controller's AHB interface provides a channel for data transfers between the eMMC device controller and a NAND flash controller which is also available from Arasan.

The eMMC device IP supports all new features. The eMMC device IP is backward compatible to previous versions of eMMC, including HS400, HS200, MMC 1-bit, 4-bit, and 8-bit modes.

eMMC device IP simplifies system design by supporting power-on booting without the upper level of software driver. The eMMC controller shields the host system from the functional differences among various NAND flash architectures such as MLC. The explicit sleep mode allows the host system to instruct the eMMC device controller to directly enter a low power sleep mode. The controller supports block lengths or sector sizes of 512, 1024, 2048 and 4096 bytes.

4.2 Features

- eMMC 5.1 Features
  - Supports Command Queuing
  - Supports Enhanced Strobe in HS400 Mode
  - Supports Cache Barrier
  - Supports Added Cache Flushing report
  - Supports RPMB Throughput Improve.
  - Supports write data size 8KB (thirty two 512B frames)
  - Supports Background Operation Control
  - Supports Secure Write Protection

- eMMC 5.0 Features:
  - Supports HS400 high speed interface timing mode up to 400MB/s data rate
  - Supports field firmware update
  - Supports eMMC device health report
  - Supports eMMC production state awareness
  - Supports secure removal types

  - Supports sleep notification

- eMMC 4.51 Features:
  - Complies with JESD84-B451 Embedded Multimedia Card, Electrical Standard (4.51 Device)
  - Supports high speed interface timing mode of up to 200 MB/s single data rate bus (HS200)
  - Supports tuning concept for read operation
  - Supports DDR interface with data transfer rate of 104 MB/s
  - Supports new H/W Reset pin
  - Supports Power-On Write Protection groups, Temporary and Permanent Write Protection features
  - Supports Alternative Boot operation mode
  - Supports Write Protection features on H/W Reset
  - Supports Replay Protected Memory Block (RPMB) and access control
  - Supports partition management by defining general purpose partition with enhancement attribute features
  - Supports Write Protection on Boot area
  - Supports Boot operation in High Speed and DDR mode
  - Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
  - Full backward compatibility with previous multimedia card systems (1-bit data bus, multi card systems)
  - Supports general purpose R/W command
  - Size of eMMC device capacity can be more than 2GB
  - Supports block lengths of 512 bytes and sector size of 4K bytes for high capacity
  - Error injection capabilities
  - Supports Boot operation mode in simple boot sequence method
  - Supports Sleep mode for power saving
  - Supports Reliable Write operation that performs safe-updates on sudden power failure
  - Supports Packed command flow sequence
  - Supports Power-Off notification flow
  - Supports Discard and Sanitize command
  - Supports context IDs and data tag mechanism
  - Supports security protocol write and read command
  - Supports secure erase and secure trim

- AMBA Compliance:
  - Complies with AMBA specification version 2.0
  - Supports Incremental Burst Transfers on DMA mode
  - Supports Register Transfer on non DMA mode
  - Supports Retry and Split

- Speed Classes Supported:
  - Arasan’s IP supports MMC speed classes: 0, 2, 4, 5, 6, 7, 8, 9, 10.

- Card Enumeration:
  - The Arasan eMMC Device IP is enumerated by a host. Card insertion or removal in the slot is signaled to the host. The type of the card and supported memory size are informed to the host controller via CID, CSD, Ext_CSD registers housed inside the Arasan eMMC Device IP.

4.3 Architecture

4.3.1 Functional Description

The Arasan eMMC 5.1 device IP provides a solution to integrate the controller with memory. This IP core is fully tested to meet the requirements of the JEDEC eMMC5.1 specs (Item # JC-64.1-67.14) and eMMC5.0 specs (JESD84-B451) Multi-Media Card Electrical Standard.

The controller's interfacing pins support MMC 1-bit, 4-bit and 8-bit modes. The controller portion of the core operates at eMMC clock provided by an eMMC Host that can vary from 0 to 200 MHz.

The eMMC device is accessed using a 10-pin or 14-pin interface (clock, command, 4 or 8 data pins, data strobe, 3 power pins) with an option for an additional Reset pin. It is designed to operate at a maximum operating frequency of 200 MHz and supports high speed interface timing mode of up to 400 MB/s. eMMC supports power-on booting without the upper level of software driver. The explicit sleep mode allows the host to instruct the controller to directly enter the sleep mode. The functional block diagram of Arasan eMMC 5.0 device IP is shown in the following figure.

4.3.2 Functional Block Diagram

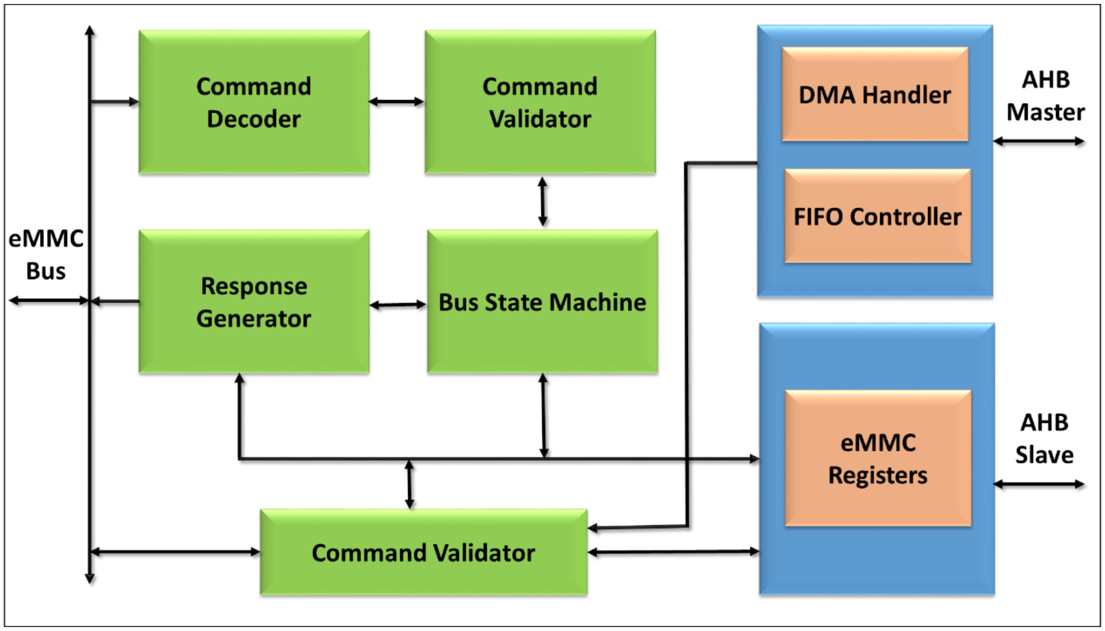

Figure 6: Device Controller Functional Block Diagram

4.3.3 Functional Block Diagram Description

4.3.3.1 Clocks

The IP operates on MMC_clk provided by the MMC host that ranges from 0-200 MHz. AHB clock is used by the IP for data transfer and programming operations.

4.3.3.2 Reset

Power-on reset is generated externally upon sensing the VDD signals. This Power-on reset is ORed with the Soft_Reset and results in the Reset signal to all the blocks inside the controller. The Soft Reset is generated whenever a GO_IDLE_STATE [CMD0] without argument is sent by the host.

4.3.3.3 Protocol Layer

This layer comprises of command decoder and validator, response generator, receiver, and transmitter sections.

4.3.3.4 Command Decoder [cmd_dec]

The Command Decoder registers the 48-bit command for all modes (MMC1_bit, MMC4_bit and MMC8_bit). This block performs cyclic redundancy checks (CRC) or the received commands.

4.3.3.5 Response Generator [resp_gen]

The Response Generator sends appropriate responses for the commands for all SD/ MMC modes.

4.3.3.6 Transmitter/ Receiver [tx_rx]

The Transmitter/Receiver handles the data transactions for all SD/MMC modes. This block controls the data lines of the MMC/SD bus.

4.3.3.7 Command Validator [cmd_valdr]

All commands received by eMMC devices are validated based on the eMMC's state. This Command Validator checks for parameter errors, address errors, and all the errors in the argument field of the command. This block generates the control for the resp_gen block.

4.3.3.8 Bus State Machine

This block handles SD and eMMC bus states as mentioned in the eMMC specification and SD physical specification.

4.3.3.9 eMMC 5.1 Register Sets/ Operational Registers

This block holds the OCR, CID, CSD, SCR, EXT_CSD and RCA for the SD or eMMC devices. These register contents are written using AHB Slave Writes.

4.3.3.10 RAM

Two 128 x32-bit Dual-port RAMs (DPRAM) are used as sector buffers to support 512-byte block size. A 128 x 32-bits Dual-Port RAM is used for storing some of the contents of the Operational Registers. A 32x32 fifo is used for storing cmd queuing pending commands.

4.3.3.11 DMA Handler

The DMA Handler contains a FIFO and initiates DMA Read or Write operations based on the interrupt serviced by the ARM processor.

4.4 Signal Interfaces

The Arasan eMMC 5.0 Device IP has five main interface groups:

- eMMC Bus Interface Pins
- AHB Target Interface Signals
- AHB Master Interface Signals
- Two 128x32 Dual-Port RAM Interface Signals
- 128x32 RAM Interface Signals

Note: Active low signals have a suffix of '_n'

Table 13: eMMC Bus Interface Signals

Pin	Direction	Description
mmc_clk	Input	mmc clock (0-52 MHz)
pow_on_rst	Input	VDD line
cmd_in	Input	Command line input
data_in[7:0]	Input	Data bus input
rst_n	Input	Hardware Reset pin
cmd_en	Output	Command line enable
data_out[7:0]	Output	Data bus output
cmd_out	Output	Command line output
data_out_en[7:0]	Output	Data bus enable
pull_up_en_dat[7:0]	Output	This output signal is used to connect or disconnect the pull-up for the data lines
exit_low_power	Output	Trigger signal to exit low power mode
data_strobe	Output	Data Strobe in HS400 mode

Table 14: AHB Target Interface Signals:

Pin	Direction	Description
hclk	Input	This clock times all bus transfers. All signal timings are related to the rising edge of hclk
hrst_n	Input	The bus reset signal is active low and is used to reset the system and the bus. This reset signal is asserted asynchronously and de-asserted synchronously after the rising edge of hclk
hrdy_in	Input	External Master ready. This signal is routed to the AHB target core
tar_haddr[31:0]	Input	The 32-bit system addresses bus. This signal is routed to the AHB target core.
tar_htrans[1:0]	Input	Indicates the type of the current transfer which can be non-sequential, sequential, idle or busy
tar_hsize[2:0]	Input	Indicates the size of the transfer which is typically byte, half word or word
tar_hburst[2:0]	Input	Indicates if the transfer forms part of a burst
tar_hwdata[31:0]	Input	The write data bus is used to receive data from the master during write operations
tar_hwrite	Input	When high, this signal indicates a write transfer and when low a read transfer
tar_hsel	Input	When high, this signal selects the target and when low de-selects target
tar_hr_data[31:0]	Output	The read data bus is used to transfer data to the bus master during read operations
tar_hrdy	Output	When high, this signal indicates that a transfer has finished on the bus. This signal is driven low to extend a transfer
tar_hrsp[1:0]	Output	The transfer response provides additional information on the status of a transfer whether it is okay, error, retry or split
tar_intr	Output	Processor interrupt

Table 15: AHB Master Interface Signals

Pin	Direction	Description
mas_hb_req	Output	AHB bus request
mas_hgrant	Input	AHB bus grant
mas_haddr [31:0]	Output	Bus address (byte address)
mas_hwdata[31:0]	Output	The write data bus is used to transfer data to the bus slaves during write operations
mas_hrdata [31:0]	Input	The read data bus is used to receive data from bus slaves during read operations

Pin	Direction	Description
mas_hwrite	Output	When high, this signal indicates a write transfer and when low a read transfer
mas_hsize [2:0]	Output	Indicates the size of the transfer which is typically byte, half word or word
mas_htrans [1:0]	Output	Indicates the type of the current transfer which can be non-sequential, sequential, idle or busy
mas_hburst [2:0]	Output	Indicates if the transfer forms part of a burst. Supports four beat incremental burst.
mas_hrdy	Input	When high, this signal indicates that a transfer has finished on the bus. This signal is used to extend a transfer when it is low.
mas_hrsp [1:0]	Input	The transfer response indicates additional information on the status of a transfer whether it is okay, error, retry or split

Notes:

1. Arasan SD/eMMC device IP supports two types of Data Packet format.
   a) Usual data (8-bit width): The usual data (8-bit width) are sent in LSB (Least Significant Byte) first, MSB (Most Significant Byte) last sequence. But in the individual byte, it is MSB (Most Significant Bit) first, LSB (Least Significant Bit) last.
   b) Wide width data (SD Memory Register): The wide width data is shifted from the MSB bit
2. HPROT signal is not used.

Table 16: 128x32 Dual-Port RAM1 Interface Signals

Pin	Direction	Description
sd_dat_out1[31:0]	Input	SD data output from the RAM during SD read
ahb_dat_out1[31:0]	Input	AHB data output from the RAM during AHB read
sd_cs1	Output	Active low chip select for SD
sd_wren1	Output	Active low write enable signal for SD write
sd_addr1 [6:0]	Output	SD read/write address
sd_dat_in1 [31:0]	Output	SD data input to the RAM during SD write
sd_out_en1	Output	Active low output enable signal for SD read
ahb_cs1	Output	Active low chip select for AHB
ahb_wren1	Output	Active low write enable signal for AHB write
ahb_addr1[6:0]	Output	AHB read/write address
ahb_dat_in1 [31:0]	Output	AHB data input to the RAM during AHB write
ahb_out_en1	Output	Active low output enable signal for AHB read

Table 17: 128x32 Dual-Port RAM2 Interface Signals

Pin	Direction	Description
sd_dat_out2[31:0]	Input	SD data output from the RAM during SD read
ahb_dat_out2[31:0]	Input	AHB data output from the RAM during AHB read
sd_cs2	Output	Active low chip select for SD
sd_wren2	Output	Active low write enable signal for SD write
sd_addr2 [6:0]	Output	SD read/write address
sd_dat_in2 [31:0]	Output	SD data input to the RAM during SD write
sd_out_en2	Output	Active low output enable signal for SD read
ahb_cs2	Output	Active low chip select for AHB
ahb_wren2	Output	Active low write enable signal for AHB write
ahb_addr2[6:0]	Output	AHB read/write address
ahb_dat_in2 [31:0]	Output	AHB data input to the RAM during AHB write
ahb_out_en2	Output	Active low output enable signal for AHB read

Note: This RAM module is used for the data transfer from and to the AHB during the DMA Read and Write operations.

Table 18: 128x32 RAM Interface Signals

Pin	Direction	Description
sd_ram_out[31:0]	Input	SD data output from the RAM during SD read
ahb_ram_op[31:0]	Input	AHB data output from the RAM during AHB read
ram_cs_en	Output	Active low chip select for SD
sd_ram_wr	Output	Active low write enable signal for SD write
sd_ram_addr [6:0]	Output	SD read/write address
sd_ram_in [31:0]	Output	SD data input to the RAM during SD write
ram_wr_data [31:0]	Output	AHB data input to the RAM during AHB write
ram_cs	Output	Active low chip select for AHB
ahb_ram_we	Output	Active low write enable signal for AHB write
ahb_ram_addr[6:0]	Output	AHB read/write address
out_en_ram_256	Output	Active low output enable signal for SD read
int_ram_out_en	Output	Active low output enable signal for AHB read

Note: This RAM is used to store the values of the SD/MMC card registers.

Table 19: 32x32 fifo Interface signals

Pin	Direction	Description
fifo_rd_data[31:0]	Input	Read data from FIFO for cmd queuing contants. This is generated when fifo address is read from AHB side.
fifo_rd	Output	Read fifo signal. This is generated when fifo address is read from AHB side.

Pin	Direction	Description
fifo_wr	Output	Write fifo signal. Generated when valid cmd45 is received.
fifo_wr_data[31:0]	Output	Fifo write data. Used to store cmd queuing contants.

4.5 SOC Level Integration

4.5.1 Verification Environment

The eMMC 5.1 Device IP core has been verified in the simulation environment using the behavioral models of the surrounding environment (RTL verification).

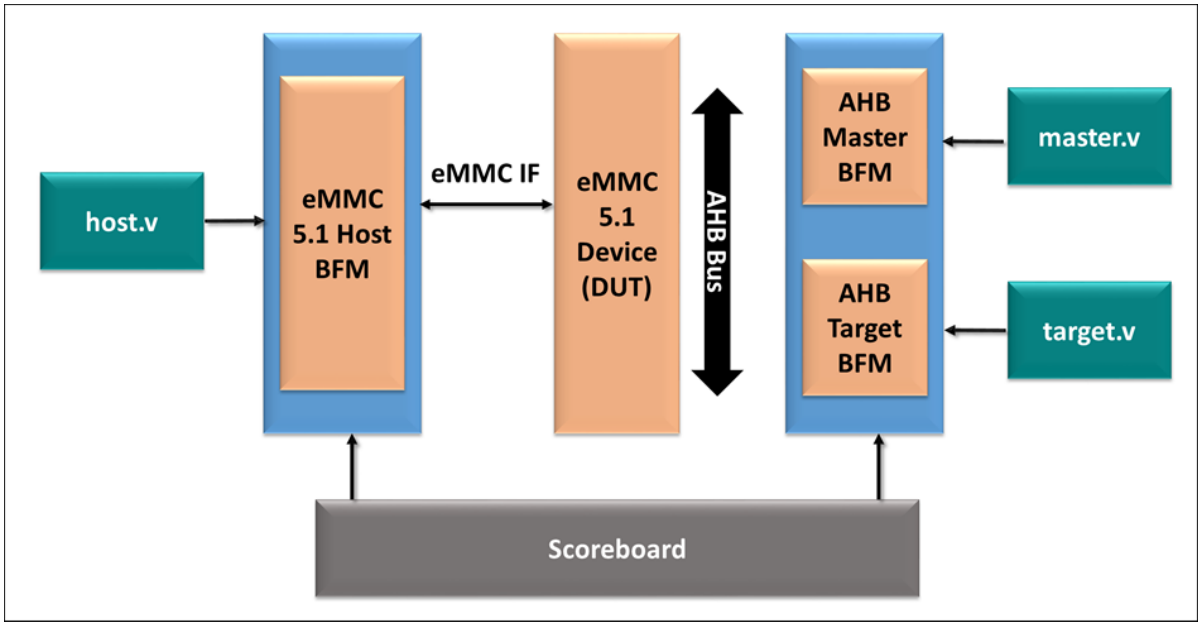

Figure 7: Verification Environment of eMMC 5.1 Device

4.5.2 Verification Deliverables

- Comprehensive suite of simulation tests for ease of SoC integration
- Verification components and test files provided
- Verification environment and test suite well documented

4.5.3 IP Deliverables

- Verilog HDL
- Synthesis scripts
- Test Environment and test scripts
- User guide

5 eMMC Device I/ O Datasheet

5.1 Overview

The ACS's eMMC5.1 I/O is verified to be fully compliant I/O interface for JEDEC eMMC5.1 when rectified and eMMC5.0 JESD84-B50 specification. It is backward compliant with eMMC4.51 and earlier versions of the specifications. This allows the designers of the SOC to easily support the EMMC interface and optimize the performance and power while maintaining interoperability with eMMC5.0 and eMMC5.1 hosts.

5.2 Features

The following are the high level features of the ACS eMMC5.1 PHY:

- Designed for seamless integration with ACS's eMMC5.1 Device Controller.
- Supports HS400, HS200, DDR50 and legacy operating modes.
- Includes EMMC I/O PADS with ESD protection structures.
- The ACS EMMC I/O PADS are designed to meet eMMC5.1 HS400 specifications.
- ACS eMMC5.1 PHY is available in the following TSMC technology nodes:
  - 16nm FF plus v1.0 process (GL and LL)
  - 28nm HPM, HPC and LP
  - 40nm LP
- The design is intended for core supply VCORE +/-10% and I/O supply VCCQ +/-10%.

5.3 Architecture

5.3.1 eMMC 5.1 I/ O Overview

The eMMC5.1 I/O PAD is a multipurpose PAD which can be programmed to operate in different modes:

- Output with predetermined source/ sink impedance
- Open drain
- Input
- Tristate
- Weak pull up or pull down

The PAD mode of operation is determined by DR_EN, OD_EN, REN and PU control signal. When push pull mode is selected, the source/sink impedance can be programmed to 50, 33, 66, 100 or 40 Ohms. DR_TY control bits are used to select the desired source/sink impedance.

The source/sink impedance variation with PVT exceeds 25 % of its nominal value, trimming bits are provided to greatly reduce the variation. RTRIM<3:0> bits are used for this purpose.

ARASAN CAL I/O can be used to automatically reduce the variation to less than 8%.

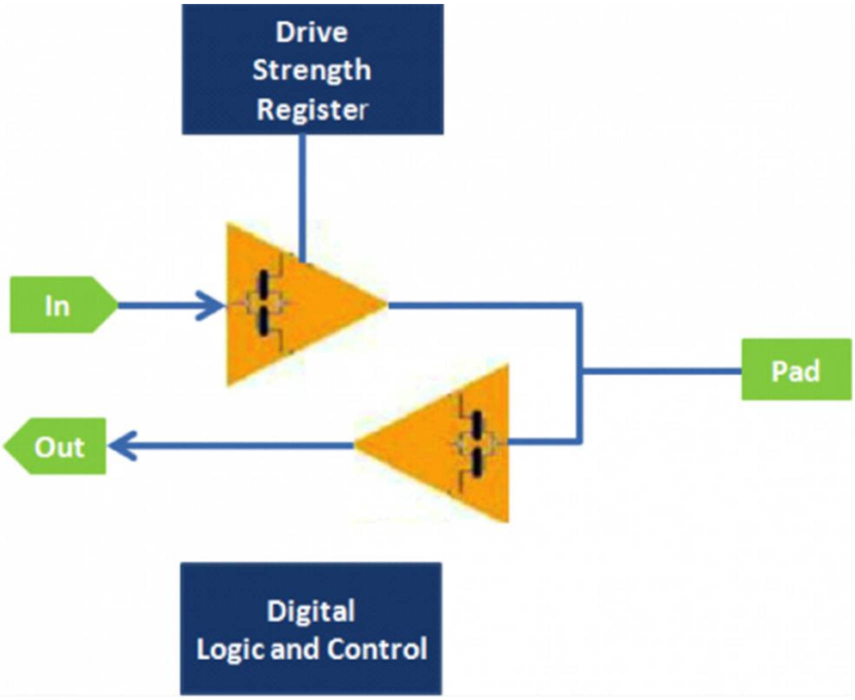

Figure 8: eMMC 5.1 I/O Block Diagram

5.4 Signal Interfaces

Table 20: Pin Details for eMMC5.1 I/O PAD

Pin	Direction/ Power Domain	Description
VDDQ	Supply	I/O Supply
VCORE	Supply	Core Supply
DR_EN	Input / VCORE	Enable PAD Driver
DR_TY[2:0]	Input / VCORE	PAD Driver Impedance Programming
DIN	Input / VCORE	Data to be transmitted`
DOUT	Output / VDDQ	Data Received
RTRIM[3:0]	Input / VCORE	Pad Driver Source/Sink Impedance Trimming
REN	Input / VCORE	Pull-up or pull down resistor enable
OD_EN	Input / VCORE	Open Drain enable
PU	Input / VCORE	1 for pull up and 0 for pull down
PAD	I/O / VDDQ	I/O Pad Connection
GNDC	Supply	Core Ground
VSSQ	Supply	I/O Ground

5.5 I/ O Configuration Settings

Table 21: Pad Mode of Operation Programming

DRIVER MODE	DR_EN	OD_EN	DIN	REN	PU	DOUT	PAD
Push Pull out	1	0	0	0	x	0	0
Push Pull out	1	0	1	0	x	1	1
Open Drain out	1	1	0	0	x	0	0
Open Drain out	1	1	1	0	x	1	Note 1
Input	0	0	X	0	x	0	0
Input	0	0	X	0	x	1	1
Idle	0	0	X	1	1	1	Pull Up
Idle	0	0	x	1	0	0	Pull Down

5.6 DC Characteristics

The following tables summarize the DC characteristics and characterization conditions of the eMMC HS400 I/O pads. This is an example table with values for TSMC 28HPM process node (Vcore = 0.8V for 16FF+, 0.9V for 28HPM/HPC, 1.05 for 28LP and 1.1V for 40LP).

Table 22: Recommended Operating Conditions

Parameter		Min	Nom	Max	Units
VCORE	Pre driver voltage	0.81	0.9	0.99	V
VCCQ (1.8/3.3V)	Post driver voltage	1.72/2.7	1.8/3.3	1.98/3.6	V
VCCQ (1.8V only)		1.62	1.8	1.98	V
TJ	Junction temperature	-20	25	100	oC
VIMAX (1.8/3.3V)	Maximum input voltage			3.7	V
VIMAX (1.8V only)				2.1	V

Table 23: DC Characteristics

Parameter		Min	Max	Units
VIL	Input Low Voltage	VSSQ-0.3	0.25* VCCQ	V
VIH	Input High Voltage	0.625*VCCQ	VCCQ +0.3	V
II	Input Leakage Current		2	µA
IOZ	Tri-State Output Leakage Current		2	µA
VOL	Output Low Voltage		0.125* VCCQ	V
VOH	Output High Voltage	0.75* VCCQ		V

5.6.1 Driver Strength Support

The following drive strengths are supported and are programmable.

Table 24: Drive Strength

Driver Type	Nominal Impedance	Relative Driving Capability	Remark
0	50 Ohms	x1	Default driver type, supports up to 200MHz operation
1	33 Ohms	x1.5	Supports up to 200MHz operation
2	66 Ohms	x0.75	The weakest driver that supports up to 200MHz operation.
3	100 Ohms	x0.5	For Low noise and low EMI systems, Minimum operating frequency is system dependent.
4	40 ohms	x1.2	Supports up to DDR 200MHz operation.

5.7 Deliverables

- GDSII database
- LVS Netlist
- Physical Abstract Model (LEF)
- Timing Models
- Behavioral Models
- Design Integration Guide
- Technical Documents

6 Hardware Validation Platform

6.1 Overview

Designed to be cost effective and Linux based, the eMMC 5.1 hardware validation platform (HVP) consists of Arasan's eMMC 5.1 IP mapped into an FPGA, offering full speed physical connectivity to a complementary SoC host or memory card device. Arasan's active participation in the JEDEC/eMMC standards bodies and early customer engagement has made Arasan's eMMC controller IP a "gold master" reference in the industry.

An equally important part of the eMMC 5.1 HVP is the software stack, which abstracts all the low level software drivers and hardware to a loadable shared object with easy to use API's. Hence, an applications software, validation or systems engineer does not need to delve into the protocol and signaling details, and considers the Arasan platform as a programmable complementary device. The eMMC 5.1 HVP can be used by system developers, system integrator, software developers and system quality analysts to debug as well as validate their products during the product life-cycle.

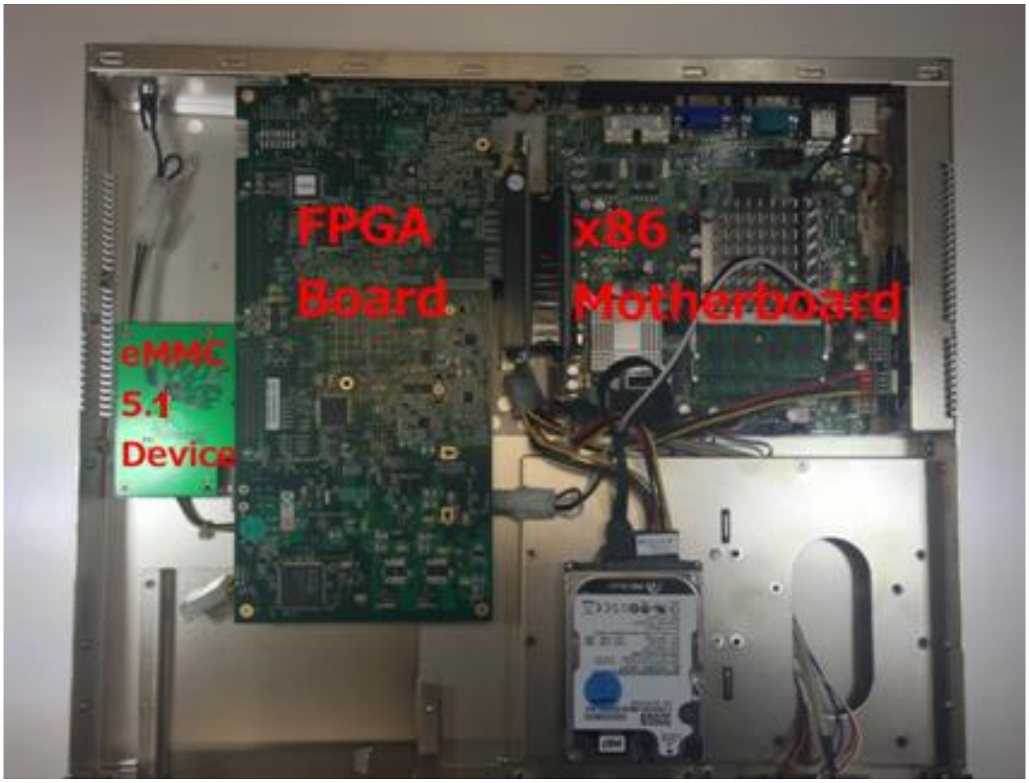

Figure 9: Photo of Arasan's Hardware Validation Platform

6.2 Features

- Specification Compliance:
  - Meets eMMC specification Version 4.2/4.3/4.4/4.51/5.0/5.1
- eMMC Device Validation:
  - Command Queueing
  - Enhanced strobe for HS400 mode with 200 MHz DDR operation
  - Background Operation Control
  - Cache Flushing
  - Cache Barrier
  - Secure Write Protection
  - RPMB Throughput Improve
  - Field firmware update
  - eMMC device health report
  - eMMC production state awareness
  - Secure removal types
  - CRC7 and CRC16 Cyclic Redundancy Checks
  - eBoot read on power-on
  - eBoot read/write in boot partitions after card initialization
  - eBoot & Normal Read/Write access in 1/4/8 bit wide modes
  - Alternate Boot Operation Mode

6.3 HVP Architecture

The eMMC5.1 Hardware Validation Platform (HVP) is a complete Linux-based system for validation of eMMC compliant devices. It may be used for SoC validation, early software development or limited production testing. The HVP contains everything you need to launch your products in the shortest possible time including a binary FPGA-implementation of Arasan's market leading eMMC Host IP and software drivers running on a Linux OS which enables user written programs to fully utilize the controller functions.

The eMMC Host Controller is fully compliant with the eMMC specifications version 5.1. Arasan's HVP supports both 3.3V and 1.8V signaling and handles the complete eMMC Protocol including transmission, data packing, CRC generation, start/end bit generation, and transaction format correctness checks.

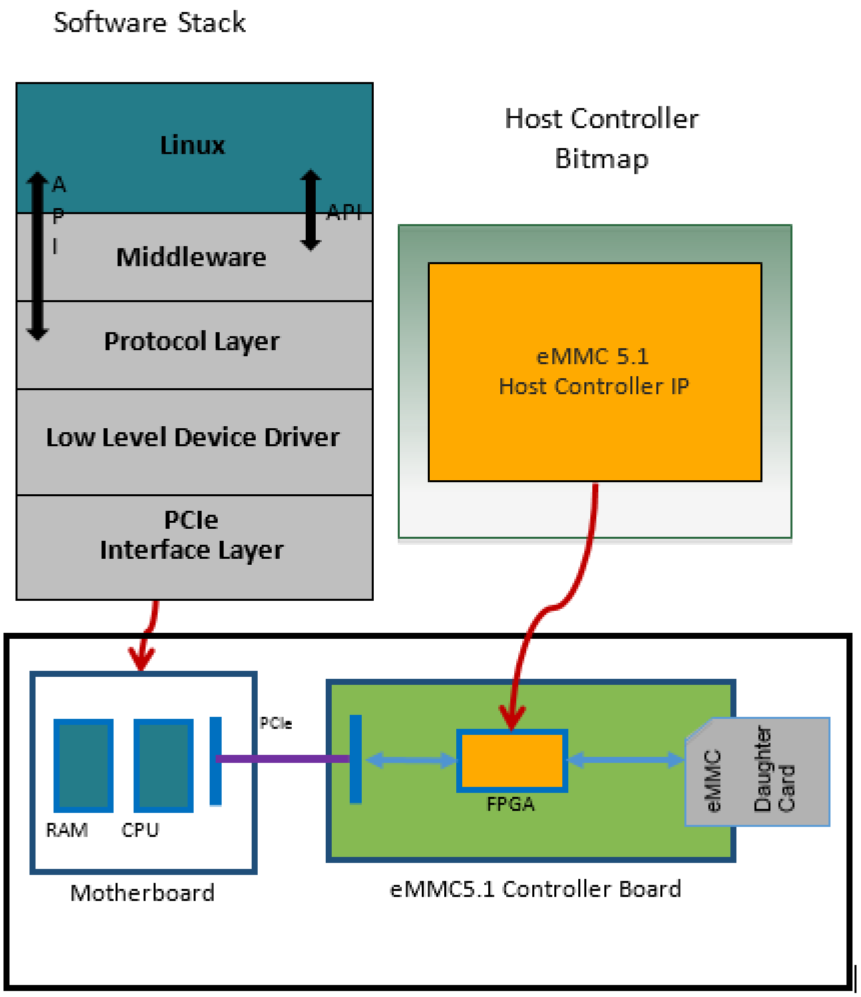

Figure 10: HVP Architecture

6.4 Deliverables

- eMMC5.1 Validation Platform including:
  - eMMC 5.1 Controller IP implemented in FPGA
  - Software Stack
- Linux Installer Package with Documentation
- Documentation

7 eMMC 5.1 NEXBus Driver

7.1 Overview

This is a production ready software stack for eMMC 5.1 Host Controller IP that is used to connect eMMC devices.

The eMMC 5.1 stack can also be used for validating a device during its development and integration life cycles thereby helping designers to reduce the time to market for their product. The modular neX+ stack is architected to be OS and platform independent which eases porting effort. It has a thin OS and hardware abstraction layers making it highly portable. The neX+ stack provides a generic API set to access, control and configure the bus driver, host controller driver and the underlying hardware.

The eMMC 5.1 software stack include functions for initialization, sending of commands, data transfer, power management, bus configuration, client driver matching, host controller hardware configuration and shutdown. The neX+ stack can support a single host controller with multiple slots or multiple host controllers with multiple slots. The software stack complies with the latest eMMC standards. It supports eMMC cards with the option for a device to boot directly from these cards using the boot mode feature.

7.2 Features

- JEDEC eMMC 5.1 complaint
- Backward Compatible to eMMC 5.0/4.x
- Command Queueing
- Enhanced strobe for HS400 mode with 200 MHz DDR operation
- Background Operation Control
- Cache Flushing
- Cache Barrier
- Secure Write Protection
- RPMB Throughput Improve
- Field FW Update
- Device Health Report
- Production State Awareness
- Sleep Notification
- Hardware& Device Initialization
- Card Register Read/Write
- Bus Width Switching
- Bus Speed Switching
- Packed Read / Write Commands
- Boot Partition and User Data Partition

- Multiple Block Operations

7.3 Architecture

The NEX driver consists of the following high level modules:

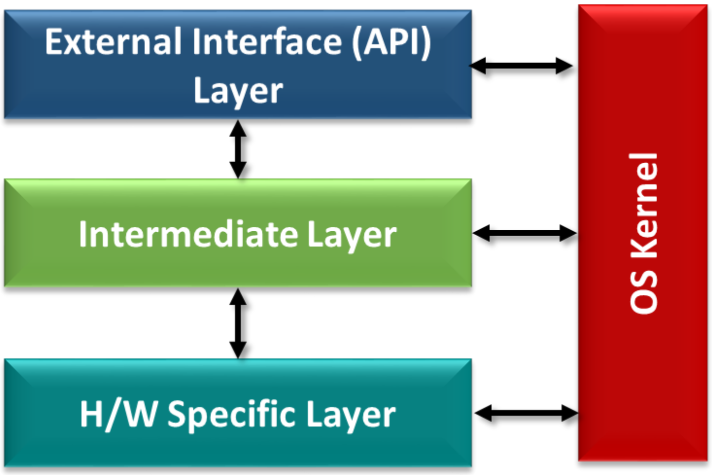

Figure 11: eMMC 5.1 Bus Driver Architecture

The NEX Bus Driver stack will be organized and follow the layered architecture for exposing APIs to control and manage protocol specific implementation and hardware interfacing etc. The NEX Bus Driver consists of the following layers:

- **External API interface Layer:** consists of exposed APIs by the driver.
- **Intermediate Layer:** consists of protocol specific implementations for the eMMC specifications.
- **HW Specific Layer**: consists of hardware specific details and implementations.
- **OS Layer:** consists of wrapper functions to the OS kernel API used in the driver.

7.4 Deliverables

- Source code in C language
- API Guide
- User Manual

8 Services & Support

8.1 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.

8.2 Arasan Support Team

Our technical support is being 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.

8.3 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.

8.4 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”.

8.5 Pricing & Licensing

Arasan is charging 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.