Datasheet

UFS 3.0 Total IP Solution

UFS Spec v3.0 Compliant
UniPro Spec v1.8 Compliant
M-PHY Spec v4.1 Compliant

Arasan Chip Systems Inc.
2150 North First Street, Suite #240, 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.
2150 North First Street, Suite 240
San Jose, CA 95131
Ph: 408-282-1600
Fax: 408-282-7800
Email: sales@arasan.com

Contents

Tables

Figures

1 Introduction

1.1 About UFS Standard

UFS (Universal Flash Storage) is a high performance storage interface, designed for use in computing and mobile systems, requiring low power consumption such as smart phones and tablets. Its high speed serial interface and optimized SCSI protocol enable significant improvements in throughput and system performance. UFS v3.0 defines bandwidth of up to 2.4 GB/s over two data lanes.

The UFS 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.

1.2 Arasan's Contribution to MIPI

Arasan Chip Systems has been an executive / contributing member with the UFS since its inception in 2010. Before that, Arasan has been, and still is, a contributor to the eMMC standard, the predecessor for UFS, since 2001.

Arasan is the leader of mobile storage, with 300 IP licensees for SD /SDIO, ONFI Compliant NAND, eMMC and UFS. Our UFS Host and Device IPs were licensed to both Application Processor companies like Qualcomm, LG and Samsung, as well as the majority of the Memory companies, and includes Samsung, Micron, 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 standards compliant 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

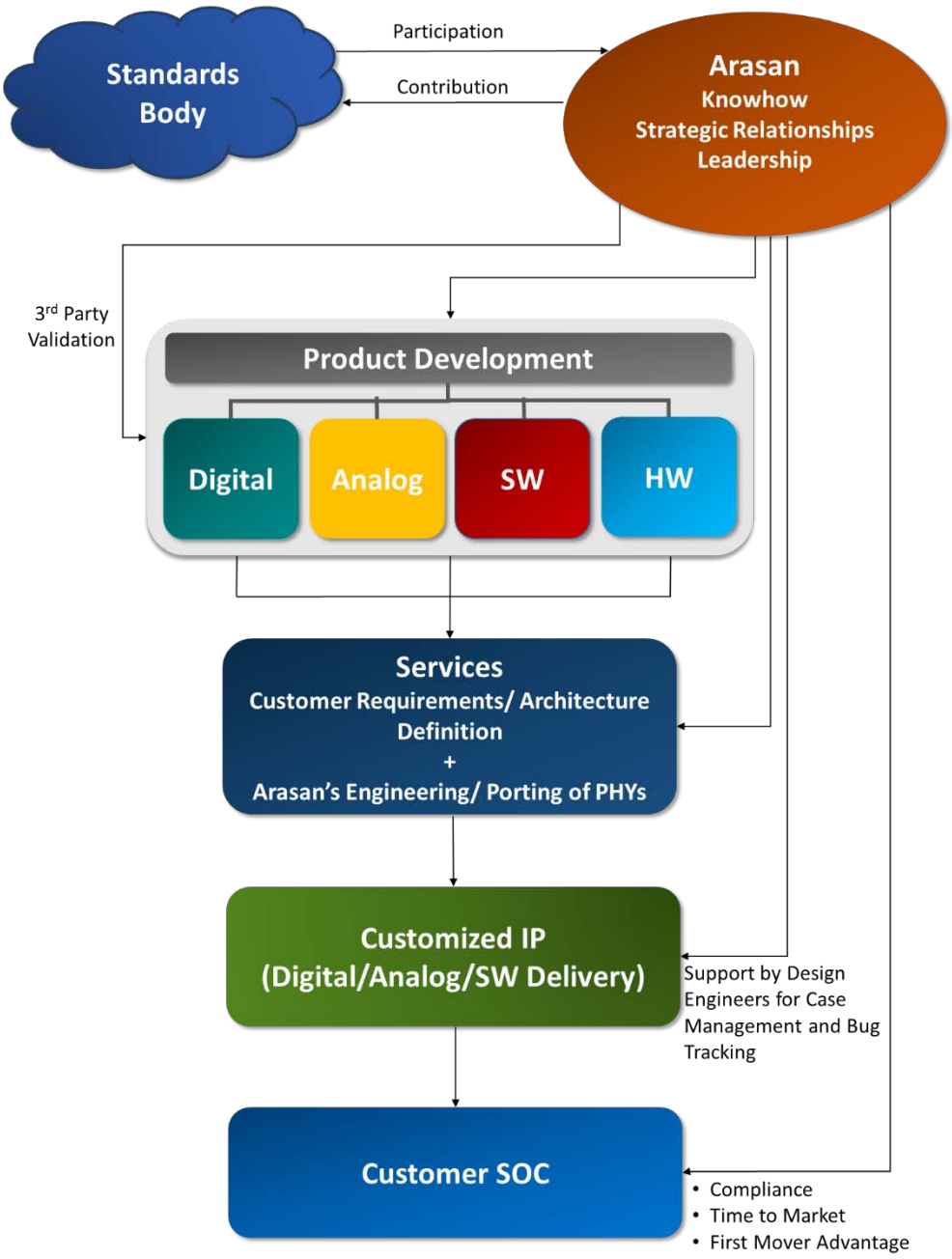

Figure 1: Arasan's Total IP Solution

2 UFS 3.0 Host IP

2.1 Overview

Universal Flash Storage (UFS) is a JEDEC standard for high performance mobile storage devices suitable for next generation data storage. UFS is also adopted by Mobile Industry Processor Interface (MIPI) as a data transfer standard designed for mobile systems. UFS incorporates the MIPI UniPro standard as well as the MIPI Alliance M-PHY standard. Most UFS applications require large storage capacity for data and boot code. Applications include mobile phones, tablets, Digital Still Camera (DSC), Portable Media Player (PMP), MP3, and other applications requiring mass storage, boot storage, XIP or external cards. The UFS standard is a simple but high-performance serial interface that efficiently moves data between a host processor and mass storage devices. USF transfers follow the SCSI model, but with a subset of Small Computer System Interface (SCSI) commands. The Arasan UFS IP family consists of Host controller IP, Device controller IP, and M-PHY.

The UFS compliant IP cores are interface building blocks that simplify interconnect architectures in mobile platforms. This leads to smaller footprint, greater interoperability between mobile IP, chips and devices from diverse sources, and lower power and Electro Magnetic Interface (EMI).

Arasan's UFS Host Controller IP, described in this document, is designed for ease of integration, highest interoperability, and fully compliant to the JEDEC standards. It is implemented based on Arasan's proven MIPI technology, including UniPro and M-PHY.

The UFS 3.0 specification adds HS-GEAR3 and HS-GEAR4 as mandatory. The UniPro 1.8 specification adds new attributes and modified some of the existing attributes for each layer.

2.2 Features

- Compliant with the following specification versions:
  - JESD220D.pdf
  - JESD223D.pdf
  - MIPI UniPro version 1.8
  - MIPI M-PHY version 4.1

- Interfaces Supported:
  - AXI Bus Protocol (AXI)
  - Advanced High Performance Bus (AHB), Open Core Protocol (OCP) (Optional)
  - High-performance M-PHY type 1

- Core Features:
  - Two Lanes
  - Low power with multiple power operating modes
  - Configurable Transmit and Receive First In First Out (FIFO)s

- Error Detection and Reporting:
  - Supports data and task management
  - Supports for multiple commands and tasks

2.3 Architecture

2.3.1 Functional Description

The UFS uses UniPro as the Datalink Layer. UniPro only supports point-to-point links as these require the highest data rates required by the devices. The principal components of the UFS Host UniPro IP are AXI Interface, UFS Host Controller Interface, UFS Transport Protocol Layer (UTP), Transport Layer, Network Layer, Data Link Layer, PHY Adapter Layer and M-PHY.

2.3.2 Functional Block Diagram

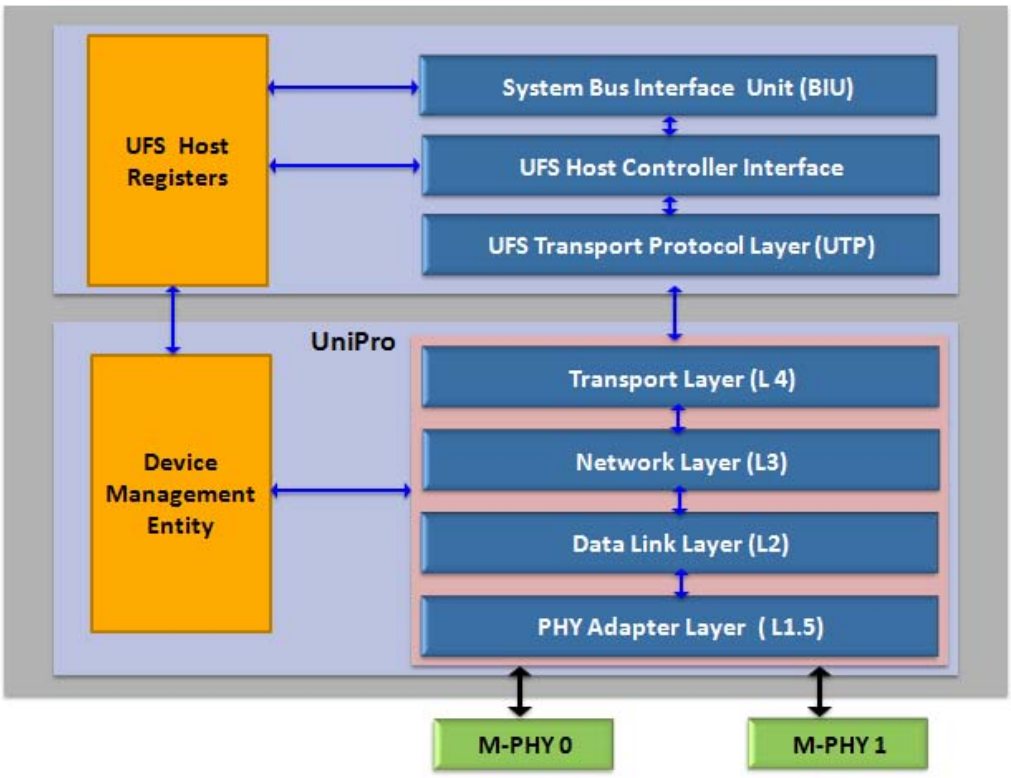

Figure 2: UFS Host Controller Functional Block Diagram

2.3.3 Functional Block Diagram Description

2.3.3.1 PHY Adapter Layer (L1.5)

- Link initialization, Power management, Hibernation, PACP frame handling to Get/Set any feature at local or peer end
- Interface with M-PHY(Type I) to Send /Receive frames across physical lane
- Skip the Symbol insertion to reduce protocol payload bandwidth at the transmitter in order to compensate for symbol interval tolerances between fast transmitter and slow receivers
- Data Scrambling is used to mitigate problems related to Electromagnetic Interference
- HS-G3 and HS-G4 Support:
  - Single Lane and Two Lanes operation support
  - Lane Management for Two Lanes operation

2.3.3.2 Data Link Layer (L2)

The main responsibilities of Data Link Layer are to provide link between a transmitter and a receiver, and to multiplex and arbitrate multiple types of data traffic. The Data Link Layer clusters the PHY Adapter Protocol Data Units' (PA-PDU's) 17-bit symbols into data frames. Every data frame consists of a 1-symbol header, L2 Payload, and a 2-symbol trailer incorporating a checksum (CRC-16). Payloads with uneven number of bytes are extended with an extra padding byte and the existence of the padding byte is flagged in the frame's trailer. Control frames are also available other than data frames. The control frames do not contain application data and this allows both ends of a link to communicate with each other to provide flow control and to handle transmission of errors.

2.3.3.3 Network Layer (L3)

The purpose of the Network Layer is to allow data to be routed to the proper destination in a network environment. A source device is responsible for supplying the address of the destination device it wants to communicate.

2.3.3.4 Transport Layer (L4)

The Transport Layer is the highest protocol layer covered by UniPro. The Transport Layer supports multiple bidirectional connections between the endpoint devices. UniPro guarantees that data sent over a single connection arrive in the same order in which it was sent. All data sent over a single connection has the same Traffic Class number (TC0). The Protocol Data Units (PDU's) of the Transport Layer are called Segments

2.3.3.5 Device Management Entity (DME)

Device Management Entity uses UFS Interconnect Layer (UIC) command registers to configure and control UniPro. For example, Link Initialization, Power Mode change, Hibernate Enter/Exit, End Point Reset, Set/Get any attributes in UniPro local or peer end are done by the SW using the UIC command registers.

2.3.3.6 UFS Host Registers

The UFS Host Registers are used to control the operation of the Host controller and to read the status and interrupt information from the Host controller. This is implemented as Memory Mapped IO space (MMIO).

2.3.3.7 UFS Host Controller Interface

The communication between Host SW and UFS device is managed by UFS Host Controller Interface using Direct Memory Access (DMA).

2.3.3.8 UFS Transport Protocol Layer

The UFS Host sends command, task, and data in UFS Protocol Information Unit (UPIU) format. The UTP block is responsible for framing the command, task and data in UPIU format.

2.3.3.9 System Bus Interface Unit

UFS Host IP uses AXI as the system bus interface. AXI slave operates in 32 bit data width which is used for register read/write transactions. The AXI master interface uses 32-bit/64-bit/128-bit data transfer width.

2.3.3.10 M-PHY

The M-PHY supports differential signaling technique for communication. It supports transfer in both HS-MODE and PWM-MODE with different Speed range (PWM) or fixed rates (HS) of communication in LS or HS mode (GEAR) settings. It controls line termination and drive strength. It uses 8b10b symbol encoding.

2.4 PIN Diagram

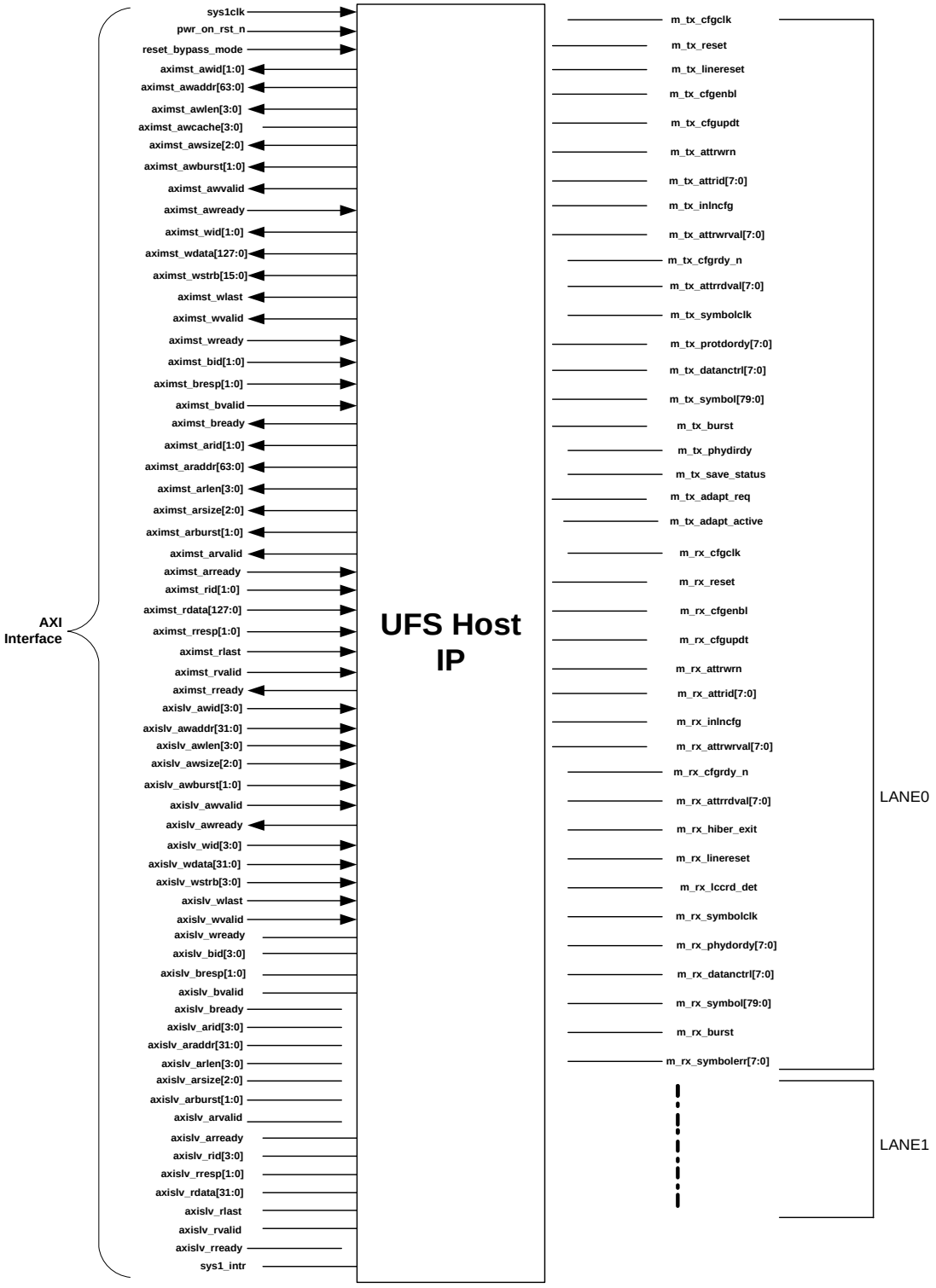

Figure 3 : UFS Host IP Pinout Diagram

2.5 IP Deliverables

- Synthesizable RMM compliant Verilog RTL code.
- Easy-to-use comprehensive OVM/UVM based randomized test environment (Ref. Sec 8, UFS VIP).
- Synthesis scripts
- Technical documents
- User guide

3 UFS 3.0 Device IP

3.1 Features

- Compliant with the following specification versions:
  - JESD220D.pdf
  - MIPI UniPro version 1.8
  - MIPI M-PHY version 4.1

- Interface Supported:
  - AXI Bus Protocol (AXI)
  - Advanced High Performance Bus (AHB), Open Core Protocol (OCP) (Optional)
  - High-performance M-PHY type 1

- Core Features:
  - Two Lanes
  - Low power with multiple power operating modes
  - Configurable Transmit and Receive First In/First Out (FIFO)s

- Error Detection and Reporting:
  - Supports data and task management
  - Supports for multiple commands and tasks

3.2 Architecture

3.2.1 Functional Description

The Arasan UFS 3.0 Device controller IP includes PHY Adapter Layer, Data Link Layer, Network Layer, Transport Layer, UFS Transport Protocol Layer (UTP), UFS Device Controller Interface, UFS Device Registers, Device Management Entity and System Bus Interface Unit.

3.2.2 Functional Block Diagram

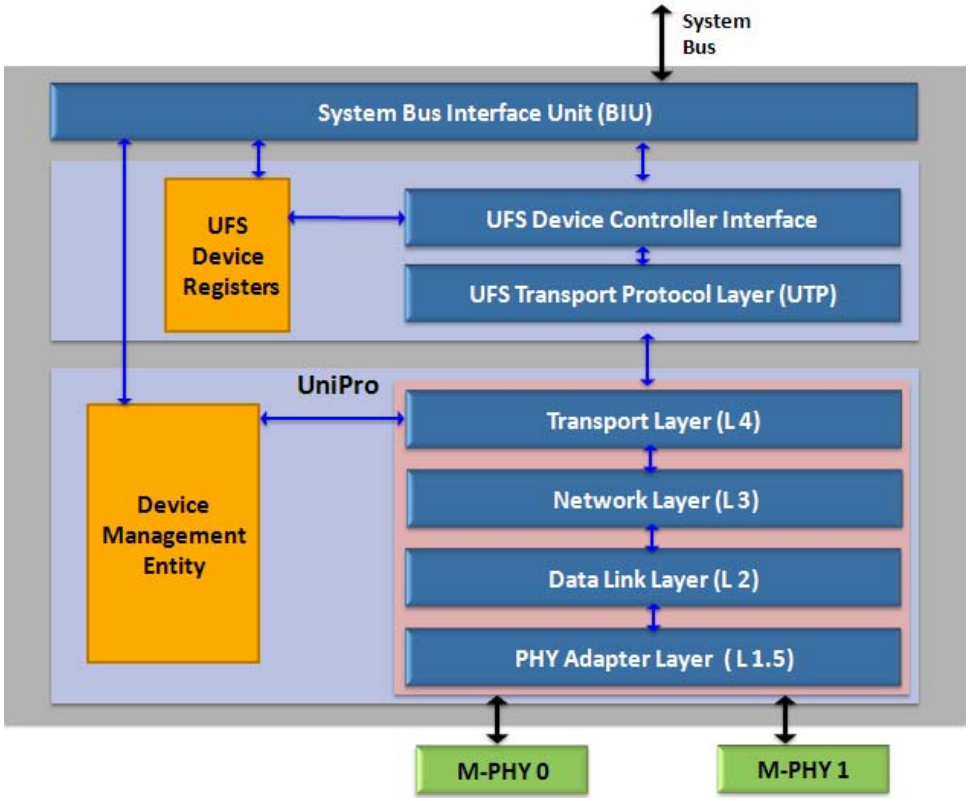

Figure 4 UFS Device Controller Functional Block Diagram

3.2.3 Functional Block Diagram Description

3.2.3.1PHY Adapter Layer (L1.5)

- Link initialization, Power management, Hibernation, PACP frame handling to Get/Set any feature at local or peer end
- Interface with M-PHY(Type I) to Send /Receive frames across physical lane
- Skip the Symbol insertion to reduce protocol payload bandwidth at the transmitter in order to compensate for symbol interval tolerances between fast transmitter and slow receivers
- Data Scrambling is used to mitigate problems related to Electromagnetic Interference
- HS-G3 and HS-G4 Support

3.2.3.2 Data Link Layer (L2)

The main responsibilities of Data Link Layer are to provide link between a transmitter and a receiver, and to multiplex and arbitrate multiple types of data traffic. The Data Link Layer clusters the PHY Adapter Protocol Data Units (PA-PDUs) 17-bit symbols into data frames. Every data frame consists of a 1-symbol header, L2 Payload, and a 2-symbol trailer incorporating a checksum (CRC-16). Payloads with uneven number of bytes are extended with an extra padding byte and the existence of the padding byte is flagged in the frame's trailer. Control frames are also available

other than data frames. The control frames do not contain application data and this allows both ends of a link to communicate with each other to provide flow control and to handle transmission of errors.

3.2.3.3 Network Layer (L3)

The purpose of the Network Layer is to allow data to be routed to the proper destination in a network environment. A source device is responsible for supplying the address of the destination device with which it intends to communicate.

3.2.3.4 Transport Layer (L4)

The Transport Layer is the highest protocol layer covered by UniPro. The Transport Layer supports multiple bidirectional connections between the endpoint devices. UniPro guarantees that data sent over a single connection arrive in the same order in which it is sent. All data sent over a single connection has the same Traffic Class number (TC0). The Protocol Data Units (PDUs) of the Transport Layer are called Segments.

3.2.3.5 Device Management Entity (DME)

Device Management Entity uses UFS Interconnect Layer (UIC) command registers to configure and control UniPro. For example, Link Initialization, Power Mode change, Hibernate Enter/Exit, End Point Reset, Set/Get any attributes in UniPro local or peer end are done by the SW using the UIC command registers.

3.2.3.6 UFS Device Registers

The UFS Device Registers are used to control the operation of the Device controller and to read the status and interrupt information from the Device controller. This is implemented as Memory Mapped IO space (MMIO).

3.2.3.7 UFS Device Controller Interface

The communication between Host SW and UFS device is managed by UFS Device Controller Interface using Direct Memory Access (DMA).

3.2.3.8 UFS Transport Protocol Layer

The UFS Device receives command, task, and data in UFS protocol information unit (UPIU) format. The UTP block is responsible for framing the DATA Payload from Device System Memory in UPIU format and send to UFS Host. This module is also responsible for parsing the DATAOUT UPIU received from UFS Host and send only the data payload to Device System Memory.

3.2.3.9 System Bus Interface Unit

UFS Host IP uses AXI as the system bus interface. AXI slave operates in 32 bit data width which is used for register read/write transactions. The AXI master interface uses 32-bit/64-bit/128-bit data transfer width.

3.3 PIN Diagram

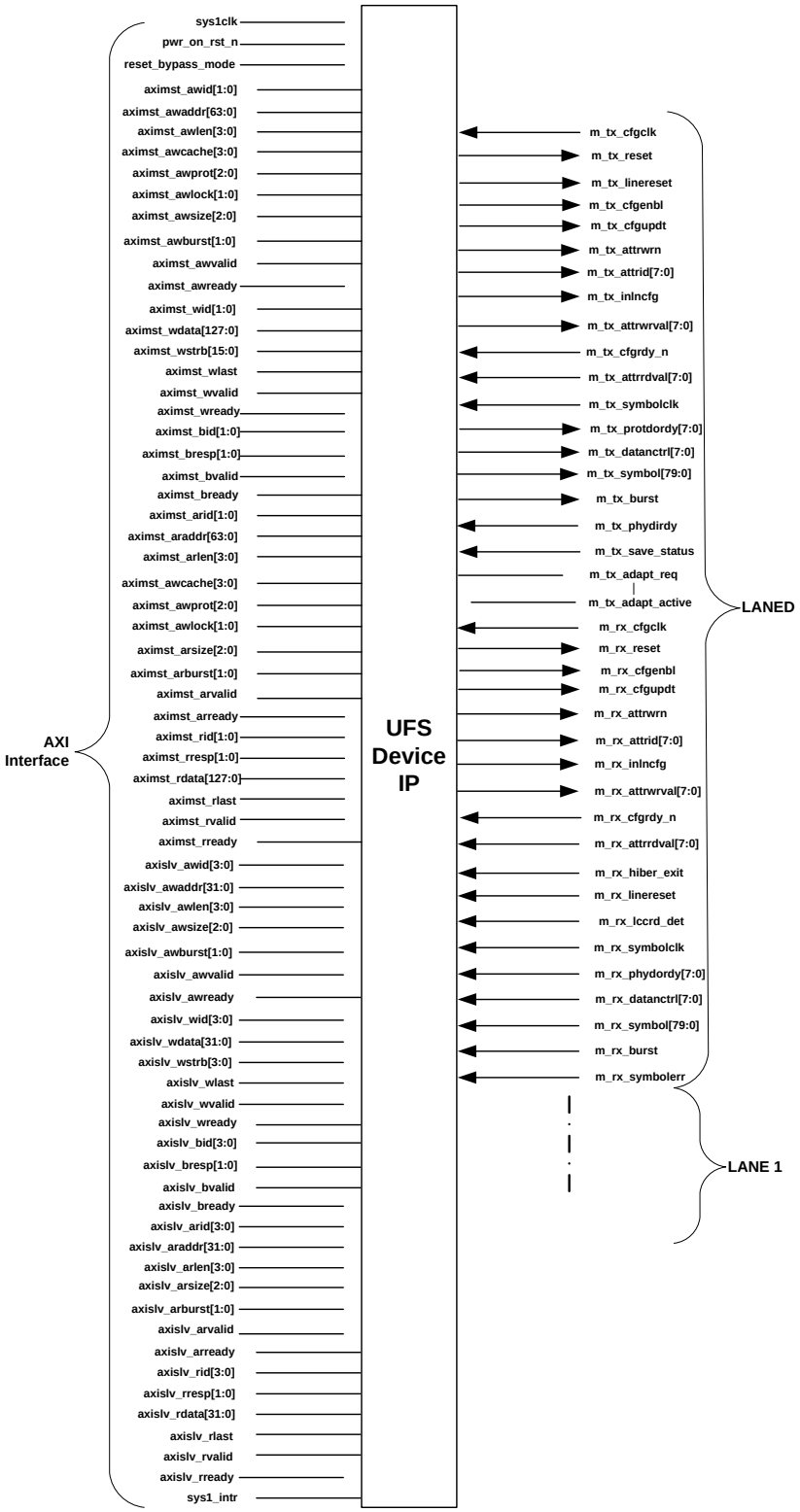

Figure 5: UFS Device IP Pinout Diagram

3.4 Soc Level Integration

3.4.1 IP Deliverables

- Synthesizeable RMM Compliant Verilog RTL of the IP Core
- Technical documents
- Gate count estimates available upon request
- Easy to use test environment
- Synthesis scripts

3.4.2 Verification Environment

The Arasan UFS Device Verification IP, bundled and delivered with the UFS Device silicon IP, is a comprehensive test environment for verification, validation, debugging, and testing of UFS Device application for functionality, compliance, and interoperability. The UFS Device verification allows the complete and extensive testing of a UFS Device application; it also ensures full functionality of the System on Chip (SoC) or Application Specific Integrated Circuit (ASIC) before the design is finalized.

The UFS Device Verification IP is delivered with UVM test bench which includes UniPro functional model, M-PHY function model, UFS Device UVM model, UFS Host UVM model, AXI master model, and AXI slave model. The Arasan UFS Device UVM Verification test environment consists of UVM agents for UFS Host and UFS Device, UVM Scoreboard, UVM Monitor for checking UFS Host and UFS Device Functionality, UVM Functional Coverage Model for UFS Device, System Verilog Assertion Checkers and Design Under Test (DUT) components. Deliverables:

- Comprehensive suite with simulation tests for ease of SoC integration
- Verification components and test files provided
- Verification environment well documented
- Verification test plan well documented
- Functional and Code Coverage reports

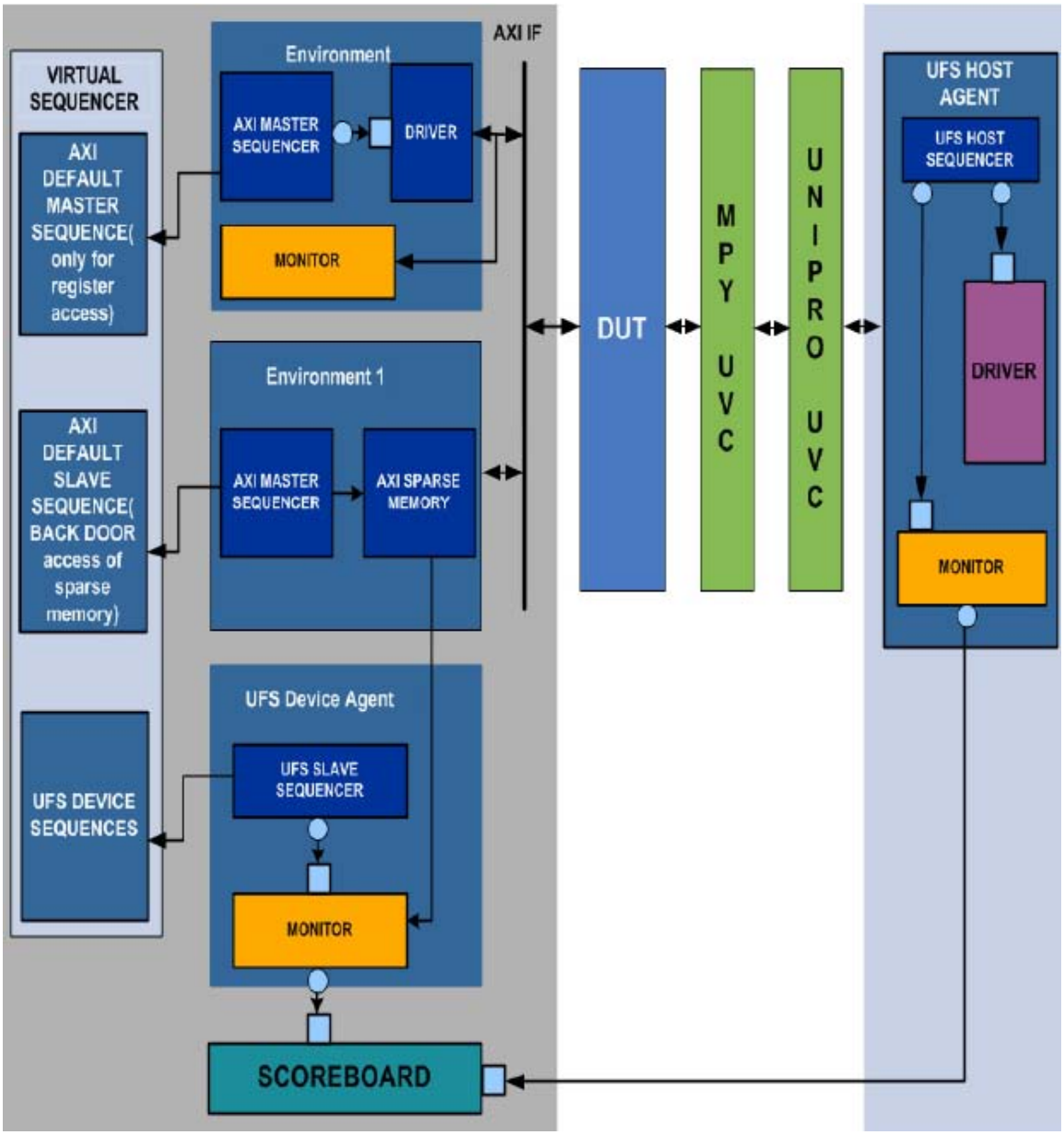

Figure 6: UFS Device IP – Verification Environment

4 UniPro 1.8

4.1 Overview

Unified Protocol (UniPro) specified by the MIPI® Alliance defines a layered protocol for interconnecting devices within a mobile system such as cellular telephone, handheld computer, digital camera or other mobile terminal. UniPro allows devices to exchange data at high data rates, with low pin counts and at low energy per transferred bit. It is applicable to a wide range of device types such as application processors, co-processors, modems etc. and to different types of data traffic.

4.2 Features

The following is a top-level feature support for the Arasan MIPI® UniPro 1.8 digital IP core.

- In-system configurable power modes
- Multi-Lane Support [Optional – 1,2,3 or 4 Lanes)
- Two Traffic Classes [TC0, TC1] on priority-based transmission
- Pre-emption during data frame transmission
- Support end-to end flow control

4.3 UniPro System Bus

The Arasan MIPI® UniPro 1.8 digital IP core supports an integrated AXI system bus for direct connection to a host system bus which supports the following features:

- Separate read/write channels to enable low cost DMA
- Ability to issue multiple outstanding addresses
- Separate address/control and data phases.

4.4 Architecture

4.4.1 Functional Block Diagram

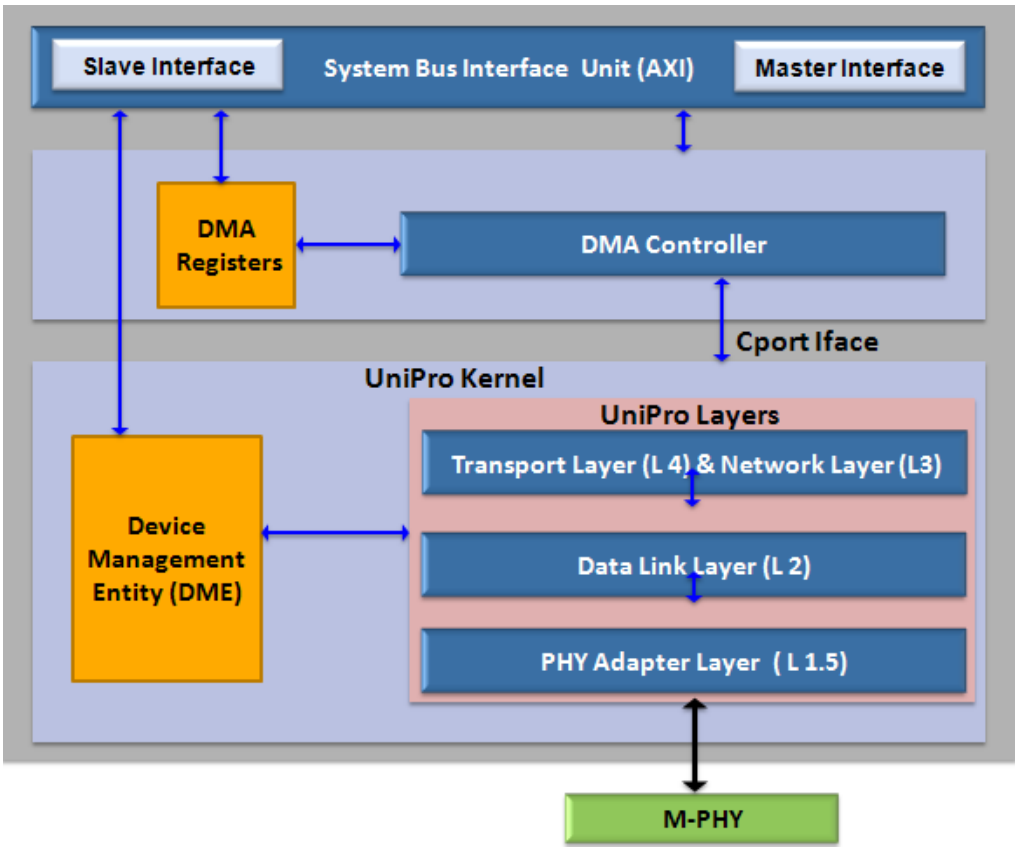

Figure 7: UniPro Controller Architecture Block Diagram

4.4.2 Functional Block Diagram Description

4.4.2.1 System Bus Interface Unit

UniPro IP uses AXI as the system bus interface. The AXI slave interface uses 32-bit data transfer width for register Read/Write operations. The AXI master interface uses 32-bit/64-bit/128-bit data transfer width for DMA operations.

4.4.2.2 DMA Registers

The DMA Registers are used to control the operation of the DMA Controller and also to read the status and interrupt information. This is implemented as Memory Mapped IO space (MMIO).

4.4.2.3 DMA Controller

The DMA Controller is a powerful scatter-gather based direct memory access (DMA) with three descriptor threads, with programmable Burst size, providing low CPU utilization. The DMA can operate in 32-bit or 64-bit or 128-bit data transfer mode. The transfer size on the AXI bus is sized

accordingly. The DMA transfers data to/from the FIFO interface and the data buffers in the host memory. Descriptors that reside in the host memory act as pointers to these buffers. The DMA supports Transmit and Receive operations independently and is controlled by a set of registers called the DMA Registers.

4.4.2.4 Device Management Entity (DME)

Device Management Entity controls UniPro operation such as Link Initialization, Power Mode Change, Hibernate Enter/Exit, End Point Reset, Set/Get any attributes in UniPro local or peer end. This is controlled by software through the register set.

4.4.2.5 Test Feature

Traffic Generator (TstSrc) - Programmable traffic generator that creates sequences of Messages (T_SDUs) containing well-defined byte sequences of well-defined lengths.

Traffic Analyzer (TstDt) - TstDst acts as a consumer and analyzer of the incoming T_SDUs. The analysis capability of TstDst is configured and can be enabled and disabled as desired.

4.4.2.6 Transport and Network Layers

- Segmentation
- L3/L4 header addition to the UPIU packet to form the UniPro frame. Removal of L3/L4 header information from UniPro frames and sends to the upper layer

4.4.2.7 Data Link Layer

- Flow control mechanism
- AFC frame sending / receiving
- CRC generation / checking
- Addition of L2 header and trailer information to the segment from Transport
- Layer/Network Layer and send across PHY Adapter Layer
- Removal of L2 header and trailer information from frames received from PHY Adapter Layer and send across Transport Layer/Network Layer

4.4.2.8 PHY Adaptor Layer

- Link initialization, Power management, Hibernation, PACP frame handling to Get/Set any feature at local or peer end
- Interface with M-PHY(Type I) to Send /Receive frames across physical lane
- Skip the Symbol insertion to reduce protocol payload bandwidth at the transmitter in order to compensate for symbol interval tolerances between fast transmitter and slow receivers
- Data Scrambling is used to mitigate problems related to Electromagnetic Interference
- HS-G3 and HS-G4 Support

4.5 PIN Diagram

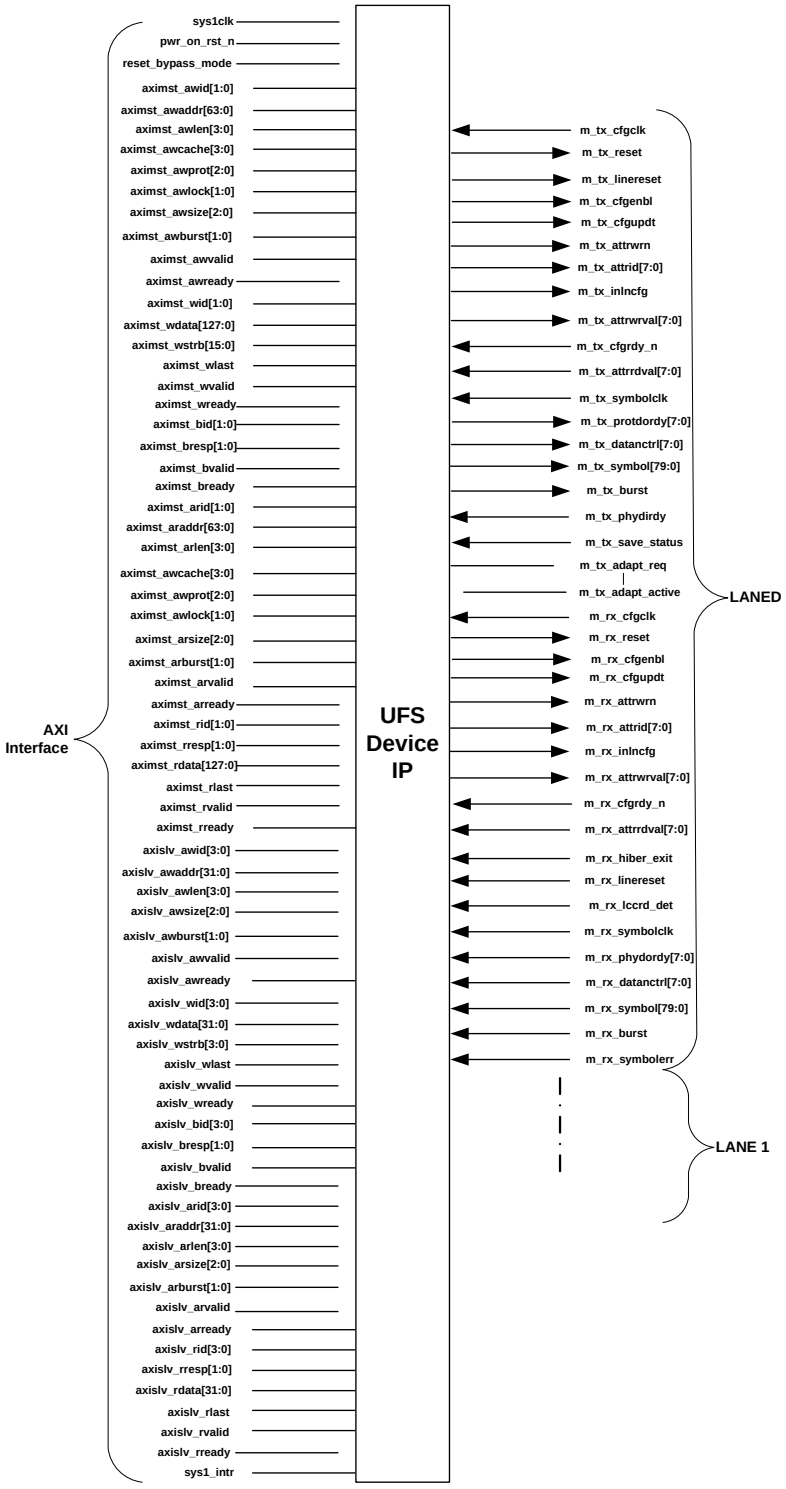

Figure 8: UniPro IP Pinout Diagram

4.6 Configurable Features

Table 1: UniPro Features and Compile Time Options

Feature	Compile Time Options
AXI data bus width	32-bit, 64-bit and 128-bit data transfer is supported
RAM configurability	Configurable

Table 2: UniPro Features and Run Time Options

Feature	Run Time Options
Cports	1-3 Cports are supported. Each Cport could be TCO or TCI which is programmable by the software
E2E	E2E feature could be enabled or disabled by the software
Pre-emption	Pre-emption could be enabled or disabled by the software

4.7 Deliverables

- Arasan MIPI® UniPro 1.8 digital IP core in RMM-compliant synthesizable RTL design in Verilog
- Test Plan Document
- User Guide
- Verilog based Test Environment
- Verification Report
- Spyglass Report
- Synthesis Script and Report
- Code Coverage Metrics(Exceptions Documented)

5 M-PHY 4.1

5.1 Overview

An M-PHY configuration (LINK) consists of minimum two unidirectional Lanes. Each of these PHY Lane modules communicates via two Lines to a complementary part on the other side of the Lane Interconnect. The M-PHY is intended to be a low pin count, power efficient serial interface with high bandwidth capabilities.

5.1.1 Lanes

Each Lane Module is unidirectional and consists of M-Tx or an M-Rx. The serial interconnect consists of two differential lines each. These I/O functions are controlled by a Lane Control and Interface Logic block. An overview of a typical M-PHY Link is shown below:

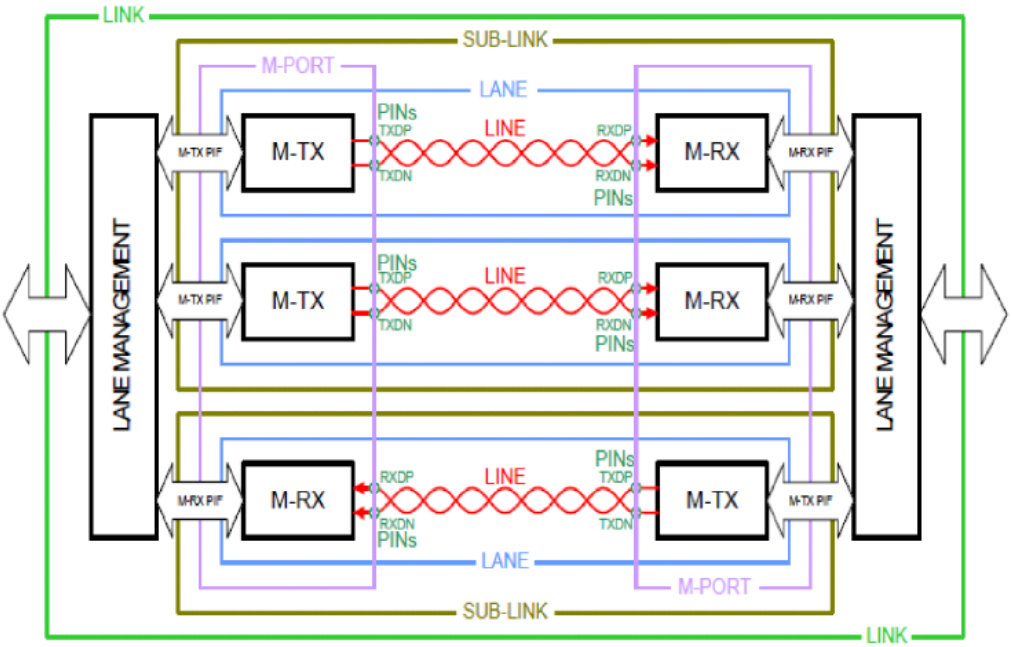

Figure 9: Illustration of MIPI M-PHY Link

5.1.2 Signaling

Both High-Speed (HS) and Low-Speed (LS) signals have a low voltage swing of 100 mV (Small Amplitude) or 200mV (Large amplitude). High-Speed functions are used for High-Speed Data traffic. The Low-Speed functions are mainly used for control and can have data transfer support.

5.1.3 Line Coding

All data transmitted (high-speed and low-speed) are 8b10b encoded. The target BER for any M-PHY based PHY link is $10^{-12.}$

5.1.4 M-PHY Type

M-PHY can be of two types Type-I or Type-II. Both systems have similar functionality and features for the High-Speed mode with support for HS-G1, HS-G2 ,HS-G3 and HS-G4. They however do differ in the Low-Speed functionality. Figure below summarizes the available speeds between Type-I and Type-II systems.

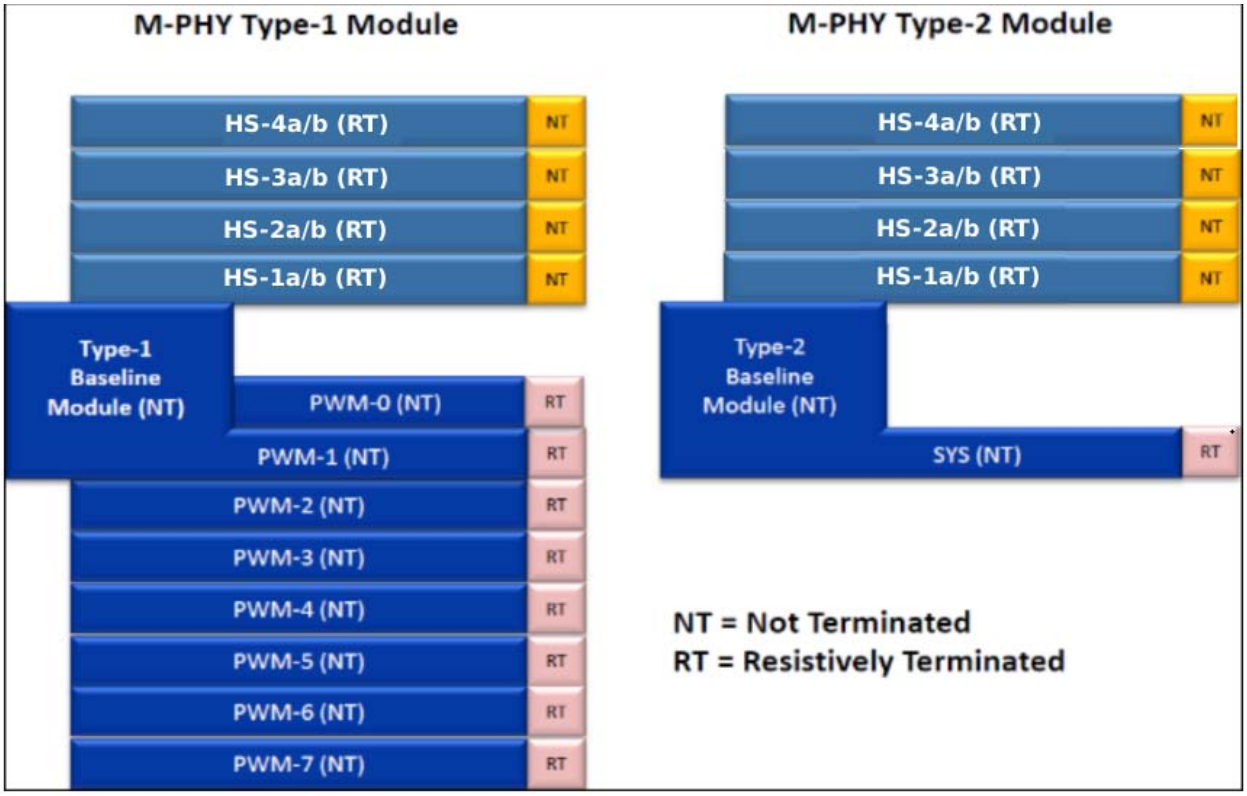

Figure 10: M-PHY Type1 and Type 2 Module

5.2 Overview of Arasan M-PHY for UFS

Arasan's M-PHY utilizes the UFS requirements for M-PHY and consists of a power-efficient solution for the architecture. The M-PHY consists of the analog transceivers, high speed PLL, data recovery units as well as the state-machine control all in a single GDSII. The interface is compliant to M-PHY pin configurations for seamless integration to the controller.

5.3 Arasan's M-PHY for UFS Features

- Complaint to MIPI Alliance Standard for M-PHY specification Version 4.1
- Supports high speed data transfer G1A/B, G2A/B , G3A/B and G4A/B with data rates of up to 11660.8 Mb/s
- Supports M-PHY Type-I system
- Support for reference clock frequencies of 19.2MHz/26MHz/38.4MHz/52MHz
- Reference clock shared between Host and device, as per UFS specification
- Supports low speed transfer G0-G7 with a bit rate of up to 576 Mb/s
- PWM signaling for Low speed [LS] data
- Supports error detection mechanism for sequence errors and contentions
- Data lanes support transfer of data in high speed mode.
- Supports LS burst, HS burst, STALL, SLEEP, HIBERN8 states.
- Supports squelch detection
- Has clock divider unit to generate clock for parallel data reception and transmission from and to the PIF.
- Activates and disconnects high speed terminators for reception and transmission.
- Supports standard PHY transceiver compliant to MIPI Specification
- Supports standard PIF interface compliant to MIPI Specification.
- On-chip clock generation configurable for either transmitter or a receiver
- Testability for Tx, Rx and PLL

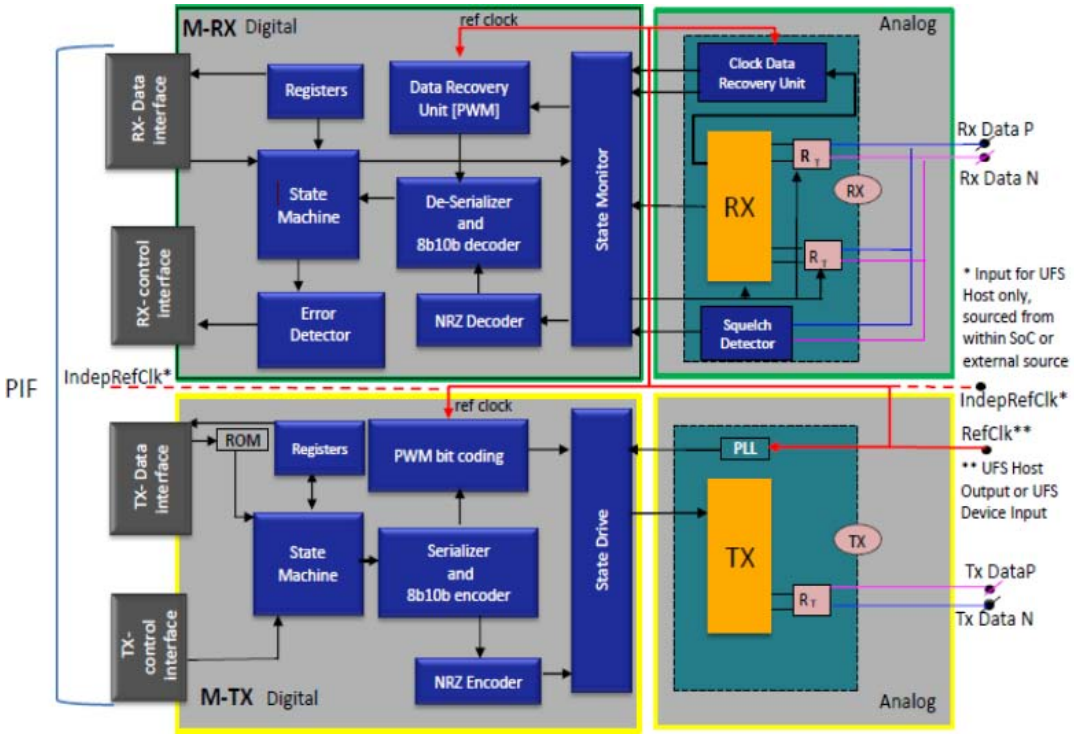

Figure 11: ACS M-PHY for UFS Block Diagram

5.4 M-PHY Pad Table

5.4.1 Functional Description of M-PHY Pads for Tx Lane

Table 3: Functional description of M-PHY Pads for Tx lane

Pin	Direction	Description
TXDP	Output	Positive polarity of low voltage differential data signal of transmitter
TXDN	Output	Negative polarity of low voltage differential data signal of transmitter

5.4.2 Functional Description of M-PHY Pads for Rx Lane

Table 4: Functional Description of M-PHY Pads for Rx Lane

Pin	Direction	Description
RXDP	Input	Positive polarity of low voltage differential data signal for receiver
RXDN	Input	Negative polarity of low voltage differential data signal for receiver

5.4.3 UFS Specific Pin

Table 5: UFS Specific Pins

Pin	Direction	Description
REFCLK_IN	Input	Reference clock (local for the Host) for the synthesizer

5.4.4 Impedance Precision Specific Pin

Table 6: Impedance Precision Specific Pin

Pin	Direction	Description
REXT	Input	External Reference resistor 10Kohm +/-1%. Consumer specific

5.4.5 Protocol Interface (PIF) Signals

Table 7: M-TX-CTRL Signals

Pin	Direction	Description
Tx_CfgClk	Input	Transmit Control Interface Clock.
Tx_Reset	Input	Active-high asynchronous reset to reset M-TX.

		Should be asserted for at least 100 ns.
Tx_AttrID[7:0]	Input	This signal indicates the configuration attributes for read or write operations, or M-TX Capability attribute or OMCS Status Attributes for read operation.
Tx_AttrRdVal[7:0]	Output	Attribute Read data from M-TX.
Tx_AttrWrVal[7:0]	Input	Attribute Write data to M-TX.
	Input	Attribute write enable to M-TX. '1' - Write operation. '0' - Read operation.
Tx_CfgEnbl	Input	Configuration Enable to M-TX. This signal indicates an enable for attribute read or write operation.
Tx_CfgUpdt	Input	Protocol shall not issue TX_CfgUpdt between the protocol setting TX_Burst to "1" and the M-TX setting TX_SaveState_Status_N to "1".
Tx_CfgRdyN	Output	This signal indicates whether M-TX is ready to accept write data. '0' - M-TX is ready to accept write data. '1' - M-TX is not ready to accept write data.
Tx_LineReset	Input	This signal indicates M-TX to issue LINERESET.

Table 8: M-TX Data Transfer Signals

Tx_SaveState_Status_N	Output	tx_savestate_status_n indicates the MTX is entering or exiting a SAVE state. The Protocol Layer can use this signal to understand when the M-TX is not transmitting PREPARE, SYNC, HOB, PAYLOAD, TOB, BURST Extension or LINE-CFG information. The M-TX sets tx_savestate_status_n to "0" when it enters into a SAVE state. The M-TX sets tx_savestate_status_n to "1" when it exits a SAVE state.
Tx_SymbolClk	Input	M-TX DATA Symbol Clock
Tx_PhyDIRDY	Input	PHY Data Input Ready. This signal indicates that M-TX is ready to accept new data.
Tx_Symbol[79:0]/[39:0]/ [19:0]/[9:0]	Input	tx_symbol is used for BURST data transfer to the M-TX. The contents of this bus depend on the whether the 8b10b encoding function in the M-TX is bypassed. When the M-TX 8b10b encoding function is bypassed, tx_symbol carries the raw data to send on the LINEs, parallelized according to the implemented width. The LSb of tx_symbol shall correspond to the earliest transmitted bit. When the M-TX 8b10b encoding function is enabled, 64/32/16/8 LSbs of tx_symbol are used to carry the

		unencoded DATA or control symbol. The M-TX shall ignore the unused MSbs of tx_symbol.
Tx_ProtDORDY[7:0]	Input	PHY Data Output Ready This signal indicates that data is available in the corresponding tx_symbol.
Tx_DataNCtrl[7:0]	Input	tx_datanctrl indicates the type of symbol on tx_symbol. The bits of tx_datanctrl are mapped to the same as the bits of tx_protdordy. '0' - data symbol. '1' - control symbol.
Tx_Burst	Input	This active high signal indicates that burst is in progress.
Tx_Adapt_Active	Output	TX_ADAPT_ACTIVE = 1 indicates to the Protocol that MTX is in ADAPT sub-state. The signal is asserted after PREPARE. The signal is de-asserted with the end of the ADAPT sequence.
Tx_Adapt_Req	Input	The Protocol Layer shall set TX_ADAPT_REQ to "1" to initiate an ADAPT sequence. Once set to "1", the M-TX shall send the PREPARE sequence followed by the ADAPT sequence. The Protocol shall de-assert TX_ADAPT_REQ with TX_ADAPT_ACTIVE = 1 earliest, but latest with the end of the ADAPT sequence when the MTX returns to STALL. The Protocol Layer shall ensure RX_ADAPT_Control set to ADAPT prior to assertion of this signal.

Table 9: M-RX CTRL Interface Signals

Signal	Direction	Description
Rx_CfgClk	Input	Receive Control Interface Clock.
Rx_Reset	Input	Active-high asynchronous reset to reset M-RX. Should be asserted for at least 100 ns.
Rx_LineReset	Output	This signal indicates the status of LINERESET. '1' - LINERESET is detected. '0' - LINERESET is in exit state
Rx_AttrID[7:0]	Input	This signal indicates the configuration attributes for read or write operations, or M-RX Capability attribute or OMCS Status Attributes for read operation.
Rx_AttrRdVal[7:0	Output	Attribute read value from M-RX.
Rx_AttrWrVal[7:0]	Input	Attribute write value to M-RX.
Rx_AttrWrn	Input	Attribute write enable to M-RX. '1' - Write operation.

		'0' - Read operation.
Rx_CfgEnbl	Input	Configuration Enable to M-RX. This signal indicates an enable for attribute read or write operation.
Rx_CfgUpdt	Input	RX_CfgUpdt transfers the contents of the INLINE-CR registry to the effective configuration bank during a SAVE state.
Rx_CfgRdyn	Output	This signal indicates whether M-RX is ready to accept write data. '0' - M-RX is ready to accept write data. '1' - M-RX is not ready to accept write data.
Rx_Hibern8Exit	Output	Rx_hibern8Exit indicates the M-RX is exiting HIBERN8. The MRX sets RX_Hibern8Exit_Type-I to "1" when it detects a DIF-Z to DIFN transition on the LINE. The M-RX sets Rx_hibern8exit_type-i to "0" when the MRX is in either HIBERN8 or DISABLED state.
Rx_LccRdDet	Output	Rx_lccrddet indicates that M-RX received an LCC-READ sequence, which results in the update of corresponding attributes in the M-RX.

Table 10: M-RX Data Interface Signals

Signal	Direction	Description
Rx_SymbolClk	Input	M-RX DATA Symbol Clock
Rx_Symbol[79:0]/[39:0]/ [19:0]/[9:0]	Output	Rx_symbol is used for BURST data transfer from the M-RX. When the 10b8b decoding function is bypassed, Rx_symbol carries the raw data as received on the LINEs, parallelized according to the implemented width. The LSb of Rx_symbol shall correspond to the earliest received bit. When the 10b8b decoding function is enabled; only the 64/32/16/8 LSbs of Rx_symbol are used to carry the decoded DATA or control symbol. The MRX shall set the remaining MSbs to "0".
Rx_PhyDORDY[7:0]	Input	PHY Data Output Ready This signal indicates that data is available in the corresponding Rx_symbol.
Rx_DataNCtrl[7:0]	Input	tx_datanctrl indicates the type of symbol on Rx_symbol. The bits of Rx_datanctrl are mapped to the same as the bits of Rx_protdordy. '0' - data symbol. '1' - control symbol.
Rx_Burst	Input	This active high signal indicates that burst is in progress.

5.5 Power Pads

Table 11: Power Pads

Pin	Type	Direction	Description
DVDD	POWER	INOUT	Core supply for Digital logic of M-PHY. It can be either Analog Interface or SoC interface
DVSS	POWER	INOUT	
VDDIO_BG	POWER	INOUT	IO supply for Bandgap and PLL
VSSIO_BG	POWER	INOUT	
VDDIO_TX	POWER	INOUT	IO supply for transmitter
VSSIO_TX	POWER	INOUT	
VDDIO_RX	POWER	INOUT	IO supply for receiver
VSSIO_RX	POWER	INOUT	
VDDC	POWER	INOUT	Core supply for all Analog front end. It is necessary if DVDD/DVSS from SoC.
VSSC	POWER	INOUT	

5.6 Hard Macro Deliverables

The full IP package set of deliverables are defined below:

- User guide and Integration Guides
- GDS-II
- CDL netlist for LVS
- LVS reports
- DRC and Antenna reports
- LIB files
- LEF
- Scan-inserted netlist for DFT
- Verification Environment with behavioral models

6 UFS 3.0 Software Stack & Driver

6.1 Overview

The UFS Host Stack is a stack developed for UFS Host Controllers communicating with UFS Device controllers via UniPro and M-PHY. The stack can also be used for validating a UFS device during its development and integration life cycles thereby helping designers to reduce the time to market for their product.

The modular UFS Host stack is architected to be OS and platform independent which eases porting efforts. It has a thin OS and hardware abstraction layers making it highly portable.

The UFS host stack has a low level hardware layer, which is OS independent, and users can employ this layer alone for UFS host/device validation with no driver complexity. The UFS stack provides a generic API to access, control and configure the bus driver, host controller driver and the underlying hardware. The stack include functions for UFS initialization, UniPro attribute configuration, sending/Receiving of commands/tasks in the form of UPIUs, data transfer, UFS interrupt handling, UFS device configuration and UFS host controller hardware configuration. The UFS host stack supports a single UFS Host controller with a single UFS Device.

6.2 Features

- Compliant with JEDEC UFS HCI 3.0 and MIPI UniPro Specification version 1.8
- Portability in choice of OS, processors and hardware
- Easy-to-use interface for applications
- Fully documented generic device operation API

6.3 Architecture

The Arasan UFS host stack consists of the following layers:

- Application Interface Layer (API Layer)
- Protocol Layer
- Host Controller Driver Layer
- Low Level Hardware Abstraction Layer
- OS Abstraction Layer

The layered architecture allows for easy porting to various operating systems and various platforms. Client applications such as the function drivers interface with the API layer to use the UFS device. The low level details of the protocol are abstracted for the end-user and are handled in the software stack. A set of well-defined APIs are provided at this layer.

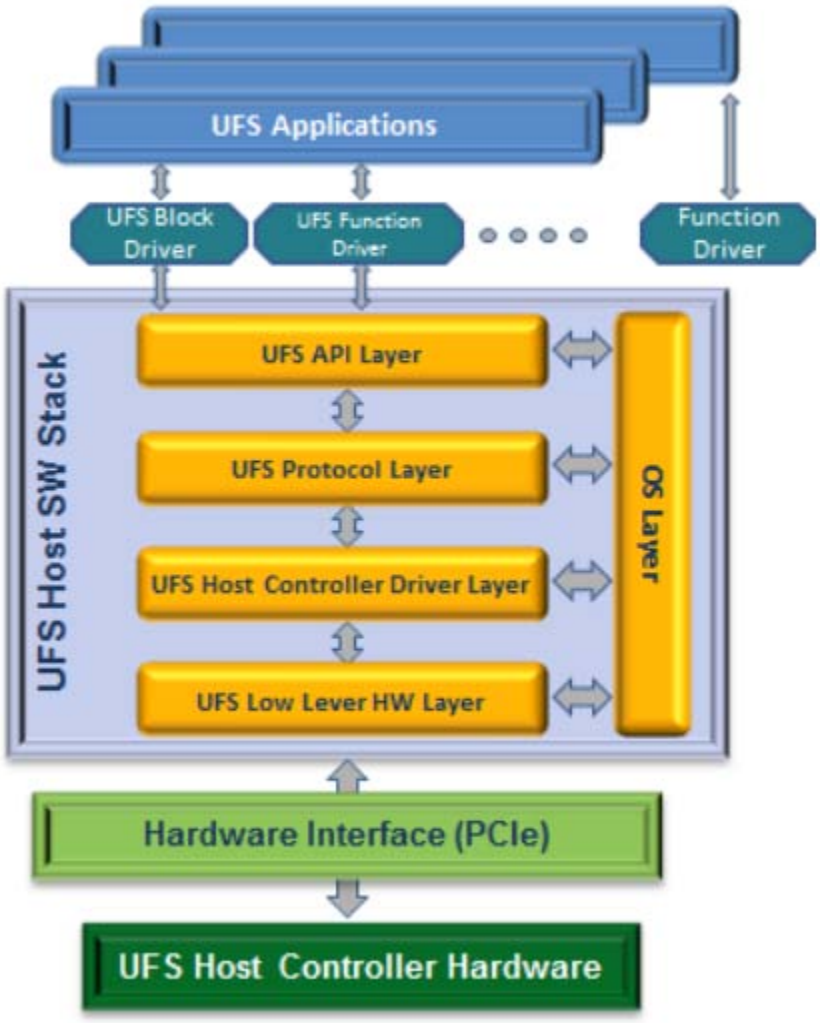

Figure 12: UFS Host Stack Architecture

6.4 Deliverbles

- Source code in C and/or binaries for application processor UFS stack.
- Software Developers Guide

7 UFS 3.0 Hardware Validation Platform

7.1 Overview

Arasan's Hardware Validation Platform (HVP) can be used for development and/or validation of UFS 3.0 devices and hosts, complying with the UFS standard specification version 3.0. The HVP can be purchased in either a Host or Device configuration. The HVP contains everything you need to launch your products in the shortest possible time frame including Host or Device IP, M-PHY signaling interface, software drivers and documentations. A monitor, standard USB keyboard and mouse are required.

Depending on configuration, the Arasan HVP handles UFS 3.0 Host or Device Protocol at transmission level, packing data and checking for transaction format correctness

7.2 Features

- UFS 3.0 Host and Device configurations available
- Complete UFS 3.0 hardware implementation
- Interop-proven UniPro 1.8 link layer
- MIPI M-PHY 4.1 Interface
- High speed mode Gear 1, Gear 2 , Gear 3 and Gear 4.
- Supports 2 lanes for 23.3 Gbps max bandwidth
- Task management operations
- Supports multiple partitions (LUNs) (to dummy memory) with partition management
- Definable write-protect group size
- Boot mode operation
- Device enumeration and discovery
- Background operations
- Secure Erase and Trim operations enhance security
- Supports Write-protect options

7.3 Description

The UFS 3.0 Hardware Validation Platform (HVP) is a complete Linux-based system for validation of UFS 3.0 Host or Device 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 UFS 3.0 Host IP or Device IP and software drivers running on a Linux OS which enables user-written programs to fully utilize the controller functions.

The Arasan HVP comprises a PC platform running the Ubuntu / Fedora Linux operating system, an FPGA IP Board which is pre-programmed with either UFS 3.0 Host or Device IP and a connectivity board which includes SMP connectors which may be used to connect the system to second M-PHY

based UFS 3.0 system (DUT). The DUT can either be another HVP with a complementary IP configuration or a SoC containing a UFS 3.0 Host or Device and an M-PHY. The UFS 3.0 HVP also includes a binary version of Arasan's UFS 3.0 Host Stack or Device Stack. The stack can also be used for validating a UFS 3.0 Host or Device during its development and integration life cycles thereby helping designers to reduce the time to market for their product.

The UFS 3.0 stack provides a generic API set to access, control and configure the bus driver, Host or Device controller driver and the underlying hardware. The stack include functions for UFS initialization, UniPro attributes configuration, sending and receiving commands and tasks in the form of UPIUs, data transfer, UFS interrupt handling, UFS device configuration and UFS host controller hardware configuration. The UFS 3.0 Host stack can support a single UFS 3.0 Host controller with a single UFS 3.0 Device.

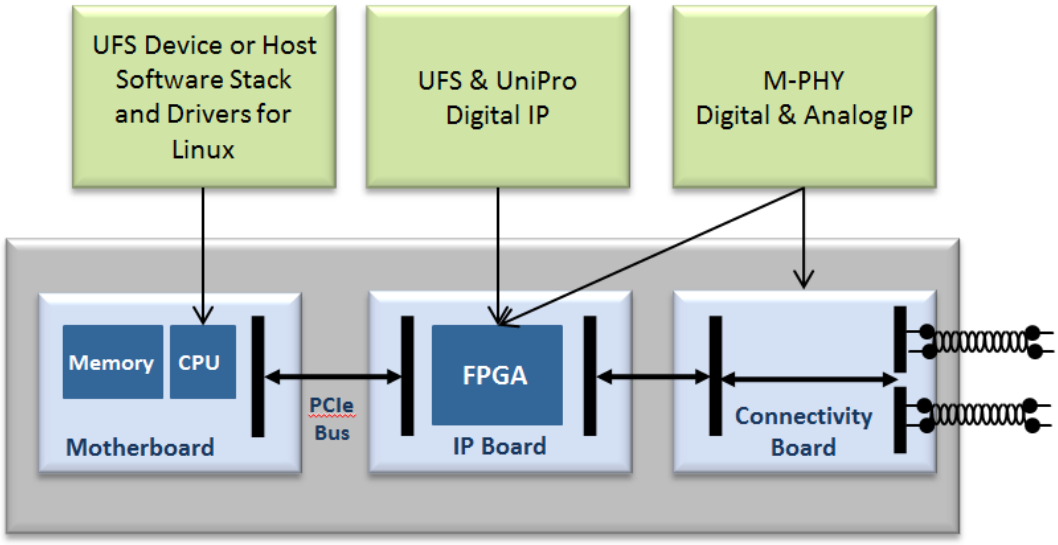

Figure 13: UFS HVP Architecture

7.4 Deliverables

- Intel-based PC running Fedora Linux with:
  - Arasan UFS 3.0 FPGA card
  - UFS 3.0 software stack (in binary / compiled form)
- SMA cables

8 UFS Verification IP (VIP)

8.1 UVM Verification Environment

The Arasan UFS Host Verification IP, bundled and delivered with the UFS Host silicon IP, is a comprehensive test environment for verification, validation, debugging, and testing of UFS Host application for functionality, compliance, and interoperability. The UFS host verification allows the complete and extensive testing of a UFS host application; it also ensures full functionality of the System on Chip (SoC) or Application Specific Integrated Circuit (ASIC) before the design is finalized.

The UFS Host Verification IP is delivered with UVM testbench which includes UniPro functional model, M-PHY function model, UFS Host UVM model, UFS device UVM model, AXI master model, and AXI slave model. The Arasan UFS Host UVM Verification test environment consists of UVM agents for UFS Host and UFS Device, UVM Scoreboard, UVM Monitor for checking UFS Host and UFS Device Functionality, UVM Functional Coverage Model for UFS Host, System Verilog Assertion Checkers and Design Under Test (DUT) components.

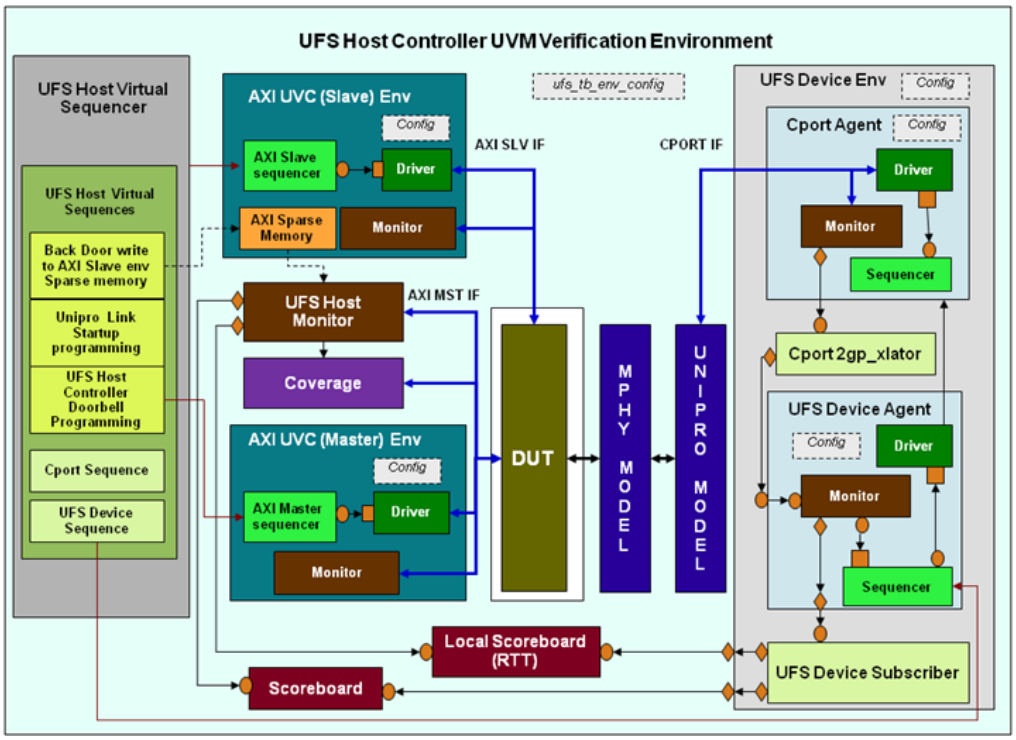

Figure 14: UFS Verification Environment Diagram

8.1.1 UVM Verification Environment Components

8.1.1.1 UFS Host Virtual Sequencer

This contains the “axi_master_seqr”, “axi_slave_seqr” and “ufs_dev_seqr” sequencers, “cport_seqr”. UFS virtual sequences (e.g. ufs_utp_base_seq) generates the UTP descriptors and UFS UPIU structures with valid random configurable test parameters. After generating UTP descriptor packets and UPIU Packets, UFS virtual sequence fills these generated packets in AXI Sparse Memory (AXI Slave Memory) of AXI Slave env, this slave env is AXI UVC which is configured as Active slave, Passive Master. Memory write and read operation to AXI Sparse Memory are performed using AXI UVC’s backdoor access mechanism.

8.1.1.2 AXI UVC Master Env

AXI UVC Master is AXI UVC configured as Active Master and Passive Slave. All the register (UFS host and UniPro attribute) programming operations are performed using AXI UVC Master Sequencer.

8.1.1.3 AXI UVC Slave Env

AXI UVC Slave is AXI UVC configured as Active Slave and Passive Master. Sparse memory is part of the AXI UVC Slave environment.

8.1.1.4 UFS Host TB Monitors

There are four monitors used verification environment:

- UFS Host Monitor
- UFS Device Monitor
- UFS Host AXI Monitor
- Cport Monitor

UFS Host and Device monitors collects and checks the upius for any protocol errors. These monitors also send the Upius to ufs scoreboard through UVM analysis port. UFS Host AXI user monitor is added for debugging at AXI interface level. Cport monitor is part of UFS Device env’s Cport Agent and can be used to see and debug transfers that happens between Unipro Model and UFS Device.

8.1.1.5 UFS Device Env

UFS Device env contains UFS Device Agent and a Cport Agent, both of these agents can be configured through their respective handles in ufs_device_config object. User can provide a new implementation of configuration tasks (defined in base test) in different tests to change the configuration of UFS Device env.

8.1.1.6 Scoreboard

UFS UVM testbench scoreboard functionality is to compare the transmitted and received UFS Protocol UPIUs.

Upiu's are received from UFS Host Monitor and UFS Device Monitor. There are two implementation ports inside the scoreboard for recieved UPIUs. These ports are as below:

- host_item_export
- device_item_export

The "host_item_export" provides the implementation of the write function of analysis port of UFS Host monitor. The "device_item_export" provides the implementation of the write function of analysis port of UFS device monitor.

8.1.1.7 Local Scoreboard

Local scoreboard is used to compare READY TO TRANSFER upiu packets of UFS Host and UFS Device. User can enable or disable the use of local scoreboard by configuring "has_rtt_scoreboard" field of ufs_tb_env_config as 1. By default the local scoreboard will be active.

8.1.1.8 Coverage Model

UFS UVM testbench coverage is implemented to record functional coverage bins which are coded based on UFS protocol. There are two main functional coverage models:

- UFS Host controller Register coverage model
- UFS Protocol coverage model

The above Coverage model bins are connected with host monitor interface through UVM analysis port.

8.1.2 UFS Host UVM Model

UFS Host model consists of two interfaces to connect the UFS host UVM model with DUT using AXI UVC:

- AXI master UVC interface with DUT AXI slave interface (IF0)
- DUT master interface with the AXI UVC slave (IF1)

All the UFS host controller DUT register programming are performed using "IF0". All the memory access for UTP and UPIU transmission and reception with DUT are performed using "IF1".

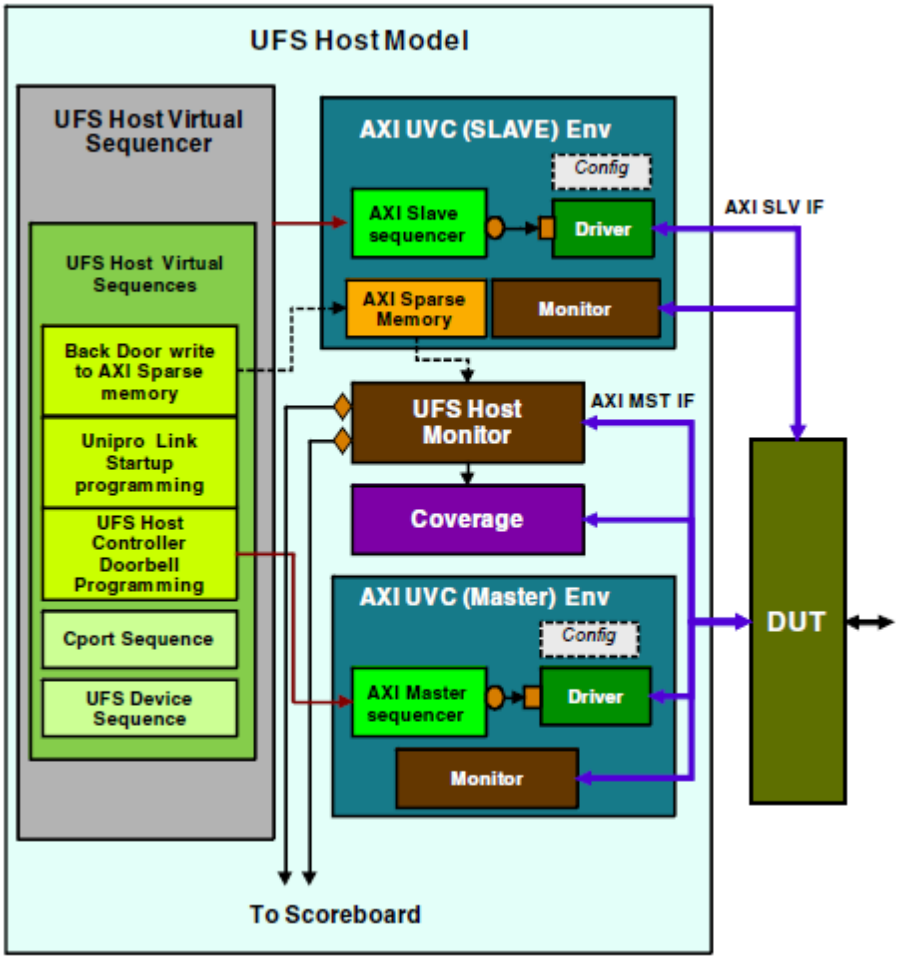

Figure 15: UFS Host UVM Environment

8.1.3 UFS Device UVM Model

UFS device model is an important component of UFS host controller UVM verification testbench. It provides the complete functionality of an UFS device. UFS device is implemented as an UVM Environment. It consists of an UFS Device Agent and a Cport Agent. UFS Device Agent communicates with Cport Agent which communicates to Unipro BFM to fetch incoming UPIU's (Command, Data Out, Query Request, Nop-Out and Task Management Request) or to drive device UPIU's (Ready To Transfer, Response, Datain, Query Response, Nop-In, Reject and Task Management Response). UFS Device environment can be configured using the configuartion class "ufs_device_config" object.

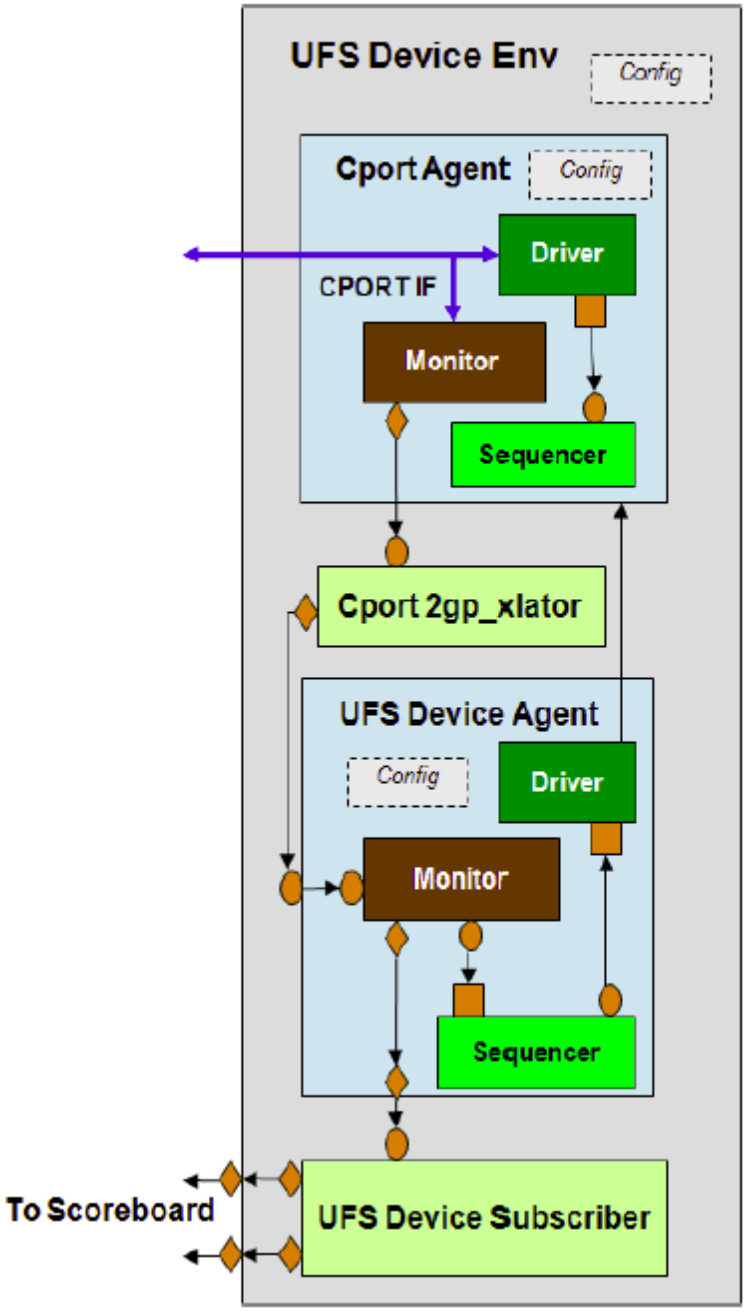

Figure 16: UFS Host UVM Environment

8.1.4 Verification Deliverables

- Comprehensive suite with simulation tests for ease of SoC integration
- Verification components and test files provided
- Verification environment well documented
- Verification test plan well document
- Functional and Code Coverage reports

8.2 Verilog Verification Environment

The Arasan UFS Host Verification IP is a comprehensive test environment for verification, validation, debugging, and testing of UFS Host application for functionality, compliance, and inter-operability. The UFS host Verification allows the complete and extensive testing of a UFS host application, it also ensures full functionality of the SoC or ASIC before the design is finalized.

The UFS Host Verification IP is delivered with a Unipro functional model, UFS host functional model, UFS device functional model, ahb master model and axi slave model.

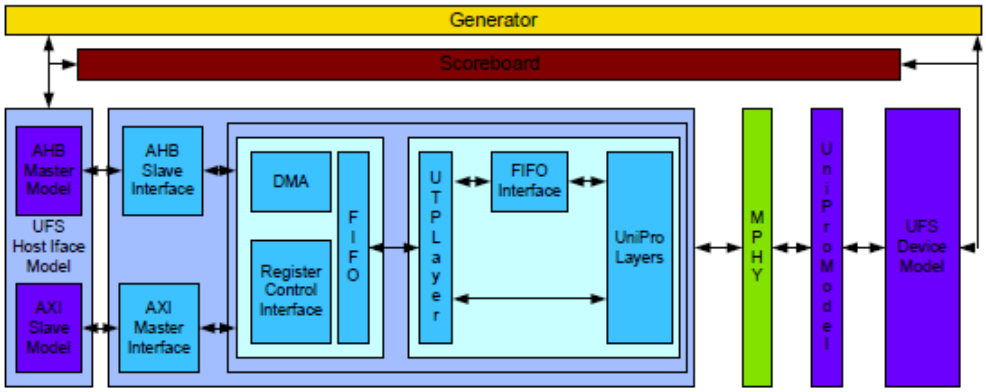

Figure 17: UFS Host Verilog Environment

The Arasan UFS Host Verification test environment consists of:

8.2.1 Generator

The Function of Generator is to generate valid UPIU commands for write and read transactions and it is also responsible for driving generated commands to AXI Interface at host side or to Unipro model at Device side.

8.2.2 Scoreboard

The Function of scoreboard is basically to check the correctness of data transfer. In this verification environment, scoreboard is responsible for checking all types of UPIU packets transmitted from UFS host to device and UFS device to UFS host.

8.2.3 AHB Master Model

The AHB Master Model interface performs register programming to the AHB slave interface of the UFS host controller. This Register programming done at the very beginning of each transaction performed between host and device. Status update read from unipro layers and UFS controller are conveyed through AHB master Interface. AHB master model supports single DWORD interface, it does not support BURST transfer.

8.2.4 AXI Slave Model

The AXI slave Model interfaces to the AXI Master interface of the UFS host controller. This model receives the transactions generated from the Generator model and responds to those transactions. The Host model has Memory model which maintains the actual memory for the Descriptors and the Buffers. The AXI slave Model performs backdoor access to this memory model when responding to read/write transactions from the UFS. The AXI slave Model supports both single DWORD as well as BURST accesses on the AXI Master Interface.

8.2.5 UniPro Function Model

This model works in both Master and Slave mode. Configurable as either an upstream or downstream mode. Unipro model functionality is same as Unipro layers.

8.2.6 MPHY Function Model

Behavioral model which supports differential signaling technique for communication. It supports transfer in both HS-MODE and PWM-MODE with different GEAR settings. It controls line termination and drive strength. It uses 8b10b symbol encoding.ufs_host_unipro_axi_top.

8.2.7 DUT

8.2.7.1 DMA Controller

The DMA Controller is a powerful scatter-gather based direct memory access (DMA). It can operate either in 32-bit data, 64bit data or 128-bit data transfer mode. The transfer size on the AXI bus is sized accordingly.

The DMA transfers messages to the Tx fifo interface or receives from the UTP Rx which is present in DUT and the data buffers in the Host Memory. Descriptors that reside in the Host memory and DMA fetches these descriptors based on register programming completed at the initial stage of command transfer via AHB Master model.

The DMA supports Transmit and Receive operations independently.

8.2.7.2 FIFO Interface

The FIFO interface module has internal RAM buffers on the transmit direction for each traffic class. This RAM stores the data and control information for each traffic class and send the data to the unipro layers.

8.2.7.3 Unipro Layers

UniPro is organized as a stack of protocol layers that roughly follow the OSI Reference Model (OSI/RM) for networking. This means that any given layer in the protocol stack is independent of higher layers but depends on the layer directly below it.

Although a layer can be described conceptually as communication between that layers and its peer entity at the other end of the link or network.

Unipro Stack consists of:

- Transport Layer (L4)
- Network Layer (L3)
- Data Link Layer (L2)
- PHY adapter (L1.5)

9 Services & Support

9.1 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.

9.2 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.

9.3 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.

9.4 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”.

9.5 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.