UFS Overview

Aug. 21, 2015
Mobile AE
What is UFS?
What is UFS?

UFS ???

• Universal Flash Storage (UFS) is a common flash storage specification for digital cameras, mobile phones and consumer electronic devices. This could bring higher data transfer speed and increased reliability in flash memory storage.

• High performance
• Serial interface
• Command queuing

• Mobile usages friendly
• Flash memory oriented
• Low power

SCSI Standardization
SATA High Speed
eMMC Low Power

UFS
What is UFS?

JEDEC defines UFS as the next generation mobile storage spec.
What is UFS?

UFS vs eMMC

- Multiple Partitions
- Boot / RPMB
- Enhanced /Extended
- Parallel Interface
- Good Features, But e.MMC today command-response based architecture is limiting
  - Designed for Single-thread computing model
  - Increasing bus frequency will not further improve BW

- Erase (Purge)
- Reliable Write
- PON
- SDR/DDR

- Write Protection
- Cache
- BKOPs
- HS200

- Context ID & Data Tag
- Sleep &Reset
- Dynamic Capacity
- Sync

- Low-Power Serial Interface
  - Multi-Task, Async & Queueing
  - Command Prioritization, Out-of-order execution

- Stateless

Universal Flash Storage [UFS]
What is UFS?

UFS vs eMMC

- Multiple Partitions
- Boot / RPMB
- Enhanced / Extended
- Erase (Purge)
- Reliable Write
- PON
- Write Protection
- Cache
- BKOPs
- Context ID & Data Tag
- Sleep & Reset
- Dynamic Capacity

Value-added Point

- Low-Power Serial Interface
  Multi-Task, Async & Queuing
  Command Prioritization, Out-of-order execution

Universal Flash Storage[UFS]

eMMC functions

SCSI SAM-5 & UPIU Format

Layered Protocol(UniPro) + Serial interface(M-PHY)

주: SCSI : layered communication architecture, Client-server model
What is UFS?

UFS Merits

- Serial Interface
- Stateless Transition
- Asynch operation
- Command Queuing
- Low Power

* CMD queuing is supported from eMMC5.1
What is UFS?

Serial Interface

• the demand for speed and cost has led to parallel communication links becoming deprecated in favor of serial links.

• Speed:
  – Clock skew reduces the speed of every link.
  – Crosstalk creates interference between the parallel lines,

• Cost: The decreasing cost of integrated circuits
What is UFS?

Async Command & Command Queuing

- Set it and forget it
- Optimal for multi processing
- Support multiple CMD queuing Maximize parallel programming
- Better throughput through better NAND utilization
What is UFS?

Async Command & Command Queuing

Re-Ordering.
- Command queue execution sequence can be changed by considering NAND interleaving
- Number of active NAND chips per time is improved.
- Suitable for UFS because UFS protocol is a multi-thread, asynchronous protocol
**What is UFS?**

## eMMC vs UFS

- **UFS has better features than eMMC**

<table>
<thead>
<tr>
<th>Items</th>
<th>eMMC</th>
<th>UFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer scheme</td>
<td>Sync</td>
<td>Async</td>
</tr>
<tr>
<td>Command Queue</td>
<td>No (Yes for eMMC5.1)</td>
<td>Yes 32-Queue Depths</td>
</tr>
<tr>
<td></td>
<td>Packed Commands</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>State Transition</td>
<td>Stateless</td>
</tr>
<tr>
<td>Interface</td>
<td>Parallel/Half-Duplex</td>
<td>Serial / Full-Duplex</td>
</tr>
<tr>
<td>Priority</td>
<td>No</td>
<td>SCSI Command Priority</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Higher-Priority LU</td>
</tr>
<tr>
<td>Abort Scheme</td>
<td>HPI</td>
<td>Task Management Scheme</td>
</tr>
<tr>
<td>Features</td>
<td>Legacy eMMC Functions</td>
<td>Same with eMMC</td>
</tr>
<tr>
<td>Command Set</td>
<td>Legacy eMMC Commands</td>
<td>SCSI Commands</td>
</tr>
<tr>
<td>Partitions</td>
<td>Boot/RPMB/User Area</td>
<td>8-LU (including Boot)/ RPMB</td>
</tr>
</tbody>
</table>
UFS Specification
UFS Specification

UFS Architecture

[Diagram showing UFS architecture with various components and connections labeled.]
UFS Specification

UFS Protocol Layer

Device Manager (QueryRequest)
- Device Manager
- UDM_SAP
- UIO_SAP

Application Layer
- UFS CommandSet Layer (UCS)
  - UFS Native Command Set
  - Simplified SCSI Command Set
  - Future Extension ...
- Task Manager
  - UTP_CMD_SAP
  - UTP_TM_SAP

UFS Transport Protocol Layer (UTP)
  - UIC_SAP

UFS InterConnect Layer (UIC)
  - MIPI UniPro
  - MIPI M-PHY
UFS Specification

UFS Architecture

Figure 2-4 — UFS System Model
M-PHY

Each 1LANE

PWM-G0 ~ PWM-G7 (Normal Speed) HS-G1 ~ HS-G3 (High Speed)
PWM-G1: 3 ~ 9Mbps
HS-G1: (1.25Gbps or 1.45Gbps)
optional: HS-G2(2.5Gbps or 2.9Gbps) HG-G3(5Gbps or 5.8Gbps)
HS-Mode: A/B → for EMI Mitigation

TYPE I: PWM Signaling → in UFS 1.0
TYPE II: system Clock Reference(NRZ Signaling)

2.9Gbps(HS-G2) & 1.45Gbps(HS-G1) In UFS 1.0(currently)

- Extensible by lane increase in pairs
- Each lane's speed is 2.9Gbps(HS-G2). So, ideal bi-directional speed is double of this.
- Interface of card type UFS is same as M-PHY of embedded UFS. So, host side implementations (HW, SW) have no difference.
UNIPRO

• UniPro = Unified Protocol

• Optimized
  ❖ For mobile use cases & multiple applications
  ❖ Low power & small battery-powered systems

• Enables minimized/extendable implementations

• Reliability with error detection and correction via retransmission simplifies protocol design

• Optimally uses MIPI's PHY technologies
  ❖ Allows aggressive power optimization
  ❖ Allows for bandwidth scaling options

• Formal UniPro SDL model available
• UniPro testing specification available
UTP

- highest HW Layer
- SLOT : HW Resource for Command Acceptance
- 32 CMD Slots, 8 Task Slot, 1 Query Slot ➔ Each Slot has it’s own DESC. (Description of operation)
Hardware vs. Software

Host

Device

SW

HW

“Packets”

“TCP”

“UDP”

8B

10B
Standardization

Similar functional features as eMMC

Storage unit

Protocol Layer & FTL

Device Manager

Task Router

UFS Port (Cport, UPIU)

DME

Transport (L4)

Network (L3)

Data Link (L2)

PHY Adapter (L1.5)

PHY

Tx

Rx

Device Manager

Task Queue

Device Server

Task Manager

LU #

ISO/IEC

JTC-1

American National Standards Institute (ANSI)

InterNational Committee for Information Technology Standards (INCITS)

Information Technology Industry Council (ITI)

IEEE

T10 Technical Committee on SCSI Storage Interfaces

Fibre Channel, HIPPI, and IPI

ATA, ATAPI, and Serial ATA

T11 Technical Committee on Fibre Channel Interfaces

T13 Technical Committee on ATA Storage Interfaces

JEDEC

mipi alliance

SK hynix
Normative Reference

- UFS HCI Version 2.0

[MIPI-M-PHY], *MIPI Alliance Specification for M-PHY℠, Version 3.0*
[MIPI-UniPro], *MIPI Alliance Specification for Unified Protocol (UniPro℠), Version 1.6*
[SAM], *INCITS T10 draft standard: SCSI Architecture Model – 5 (SAM–5), Revision 05, 19 May 2010*
[SPC], *INCITS T10 draft standard: SCSI Primary Commands – 4 (SPC–4), Revision 27, 11 October 2010*
[SBC], *INCITS T10 draft standard: SCSI Block Commands – 3 (SBC–3), Revision 24, 05 August 2010*
[UFS], *JEDEC JESD220B, Universal Flash Storage (UFS), Version 2.0*
[JEP], *JEDEC JEP106, Standard Manufacturer’s Identification Code*

- UFS Device Version 2.0

[MIPI-M-PHY], *MIPI Alliance Specification for M-PHY℠, Version 3.0*
[MIPI-UniPro], *MIPI Alliance Specification for Unified Protocol (UniPro℠), Version 1.6*
[MIPI-DDB], *MIPI Alliance Specification for Device Descriptor Block (DDB), Version*
[SAM], *INCITS T10 draft standard: SCSI Architecture Model – 5 (SAM–5), Revision 05, 19 May 2010*
[SPC], *INCITS T10 draft standard: SCSI Primary Commands – 4 (SPC–4), Revision 27, 11 October 2010*
[SBC], *INCITS T10 draft standard: SCSI Block Commands – 3 (SBC–3), Revision 24, 05 August 2010*
## UFS UPIU

### General UPIU Format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Transaction Type</td>
</tr>
<tr>
<td>1</td>
<td>Flags</td>
</tr>
<tr>
<td>2</td>
<td>LUN</td>
</tr>
<tr>
<td>3</td>
<td>Task Tag</td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>Command Set Type</td>
</tr>
<tr>
<td>6</td>
<td>Query Function / Task Manag. Function</td>
</tr>
<tr>
<td>7</td>
<td>Response</td>
</tr>
<tr>
<td>8</td>
<td>Status</td>
</tr>
<tr>
<td>9</td>
<td>Total EHS Length</td>
</tr>
<tr>
<td>10</td>
<td>Device Information</td>
</tr>
<tr>
<td>11</td>
<td>Data Segment Length</td>
</tr>
</tbody>
</table>

### Transaction Specific Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>Extra Header Segment (EHS) 1</td>
</tr>
<tr>
<td>k+1</td>
<td></td>
</tr>
<tr>
<td>k+2</td>
<td></td>
</tr>
<tr>
<td>k+3</td>
<td></td>
</tr>
<tr>
<td>j</td>
<td>Extra Header Segment (EHS) N</td>
</tr>
<tr>
<td>j+1</td>
<td></td>
</tr>
<tr>
<td>j+2</td>
<td></td>
</tr>
<tr>
<td>j+3</td>
<td></td>
</tr>
</tbody>
</table>

### Table 10-9: UPIU associated to a single task

<table>
<thead>
<tr>
<th>Initiator UPIU</th>
<th>Target UPIU</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP Out</td>
<td>NOP In</td>
</tr>
<tr>
<td>Command, Data Out</td>
<td>Ready to Transfer, Response</td>
</tr>
<tr>
<td>Command</td>
<td>Data In, Response</td>
</tr>
<tr>
<td>Task Management Request</td>
<td>Task Management Response</td>
</tr>
<tr>
<td>Query Request</td>
<td>Query Response</td>
</tr>
</tbody>
</table>

Table 10-3: General format of the UFS Protocol Information Unit

Basic Head format

Different information included depend on “Transaction Type”
UFS Specification

Transaction Type | Flags | LUN | Task Tag
---|---|---|---
4 | Reserved | Command Set Type |
5 | Query Function / Task Mgmt. Function | Response | Status
8 | Total ELS Length | Device Information | Data Segment Length
12 | 13 | 14 | 15
16
20
24
28
k
k+1
k+2
k+3

Extra Header Segment (EHS) 1

Extra Header Segment (EHS) N

Device Server

Task Manager

Device Manager

Initiator To Target | Transaction Code | Target to Initiator | Transaction Code
---|---|---|---
NOP OUT | 00 0000b | NOP IN | 10 0000b
COMMAND | 00 0001b | RESPONSE | 10 0001b
DATA OUT | 00 0010b | DATA IN | 10 0010b
TASK MANAGEMENT REQUEST | 00 0100b | TASK MANAGEMENT RESPONSE | 10 0100b
Reserved | 01 0001b | READY TO TRANSFER | 11 0001b
QUERY REQUEST | 01 0110b | QUERY RESPONSE | 11 0110b
Reserved | 01 1111b | REJECT UPIU | 11 1111b
Reserved | Others | Reserved | Others
# UFS UPIU Transaction Type

<table>
<thead>
<tr>
<th>Initiator To Target</th>
<th>Transaction Code</th>
<th>Target to Initiator</th>
<th>Transaction Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP OUT</td>
<td>00 0000b</td>
<td>NOP IN</td>
<td>10 0000b</td>
</tr>
<tr>
<td>COMMAND</td>
<td>00 0001b</td>
<td>RESPONSE</td>
<td>10 0001b</td>
</tr>
<tr>
<td>DATA OUT</td>
<td>00 0010b</td>
<td>DATA IN</td>
<td>10 0010b</td>
</tr>
<tr>
<td>TASK MANAGEMENT REQUEST</td>
<td>00 0100b</td>
<td>TASK MANAGEMENT RESPONSE</td>
<td>10 0100b</td>
</tr>
<tr>
<td>Reserved</td>
<td>01 0001b</td>
<td>READY TO TRANSFER</td>
<td>11 0001b</td>
</tr>
<tr>
<td>QUERY REQUEST</td>
<td>01 0110b</td>
<td>QUERY RESPONSE</td>
<td>11 0110b</td>
</tr>
<tr>
<td>Reserved</td>
<td>01 1111b</td>
<td>REJECT UPIU</td>
<td>11 1111b</td>
</tr>
</tbody>
</table>
UFS Specification

Command Flow

1. Host Application
   Ex) Write 6 (0 ~ 4 sector)

2. Driver
   - Memory allocation
   - Make UPIU
   - Make data
   - Make descriptor

3. UFS Host
   - UPIU Transfer
   - Data Transfer

4. UFS Device
   - UPIU receive
   - Data receive

5. UFS Device LU
   - Task Queue -> Device Server
   - Device Manager
   - R/W Buffer

6. Device FTL
   - Using Mailbox & ipc Buffer

7. Device Nand
   - Using NFC
UFS Specification

Command Flow

Figure 2-4 — UFS System Model

Driver Stack
Host Controller
Device Controller
Flash
UFS Specification

Command Flow

Table 11-10: READ (6) UFS Command

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **100** ~ **104** Sector Read
- Driver
- UFS Host
- Application
- Driver Start
- Flash
- Device Controller
- UFS Device
- (LU : Logical Unit)
- Device Level Managing
- Descriptors
- Device Manager
- UFS Command
- UFS Init
- UFS Data
- LU-1
- Storage
Read Operation Flow

App Client → UTP → UniPort → UTP → UFS Unit

1. Command UPIU (READ CDB)
2. Data in UPIU (READ DATA)
3. Data in UPIU (READ DATA)
4. Response UPIU (STATUS INFO)

TIME

Loop

UTP copies Data in to App Client Buffer

Send Command (L_T_L_Q, CDB (Read CMD),...)

T_CO_DATA.req(Command UPIU, EOM)

T_CO_DATA.cnf_L(L4CPortResultCode)

T_CO_DATA.ind(Data in UPIU, MsgStatus)

T_CO_DATA.cnf_L(L4CPortResultCode)

Send Data-In (L_T_L_Q, Device Server Buffer, Application Client Buffer Offset, Request Byte Count)

Data-In Delivered (L_T_L_Q, Delivery Result)

Command Received (L_T_L_Q, CDB (Read CMD),...)

T_CO_DATA.ind(Data in UPIU, MsgStatus)

T_CO_DATA.rsp_L()
Write Operation Flow

App Client → UTP → UniPort → UTP → UFS Unit DevServer

COMMAND UPIU (WRITE CDB)

RTT UPIU (DMA CONTEXT)

DATA OUT UPIU (WRITE DATA)

RESPONSE UPIU (STATUS INFO)

TIME

Loop

UTP fetches Data Out from App Client Buffer

Command Received (I_T_L_Q, CDB (Write CMD), ...)

T_CO_DATA.ind(Ready to Transfer UPIU, MsgStatus)

T_CO_DATA.cfg.L4CPortResultCode)

T_CO_DATA.rsp.L()

Receive Data-Out (I_T_L_Q, App Client Buffer Offset Request Byte Count, Device Server Buffer)

UTP copies Data Out to Device Server Buffer

Data-Out Received (I_T_L_Q, Delivery Result)

Execute Write CMD
## SCSI Command

- **Scsi Command Set supported by UFS** is based on UFS native commands and Scsi primary commands spec(SPC-4/SBC-3).

<table>
<thead>
<tr>
<th>Commands (CDB Size)</th>
<th>Description</th>
<th>DATA IN UPIU</th>
<th>DATA OUT UPIU</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST UNIT READ(6)</td>
<td>Test if device is ready (not for a self-test)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Inquiry(6)</td>
<td>Report device information - type, manufacture, etc</td>
<td>a single Data In UPIU(36byte)</td>
<td>X</td>
</tr>
<tr>
<td>Request Sense(6)</td>
<td>Report sense data - current status of device</td>
<td>a single Data In UPIU(18byte)</td>
<td>X</td>
</tr>
<tr>
<td>Read Capacity(10)</td>
<td>Report medium capacity and block size can be issued per Logical Unit</td>
<td>a single Data In UPIU(8byte)</td>
<td>X</td>
</tr>
<tr>
<td>Start Stop Unit(6)</td>
<td>Change power condition or load or eject medium</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Read(10)</td>
<td>Transfer data from medium to host</td>
<td>a series of Data In UPIU's</td>
<td>X</td>
</tr>
<tr>
<td>Write(10)</td>
<td>Transfer data from host to medium</td>
<td>Ready To Transfer UPIU</td>
<td>a Data OUT UPIU per RTT</td>
</tr>
<tr>
<td>Read Buffer(10)</td>
<td>Read microcode and other data and tunneling</td>
<td>a series of Data In UPIU's</td>
<td>X</td>
</tr>
<tr>
<td>Write Buffer(10)</td>
<td>Transfer microcode and other data and tunneling</td>
<td>Ready To Transfer UPIU</td>
<td>a Data OUT UPIU per RTT</td>
</tr>
<tr>
<td>Mode Select(10)</td>
<td>Set parameter, modes, etc</td>
<td>Ready To Transfer UPIU</td>
<td>a Data OUT UPIU per RTT</td>
</tr>
<tr>
<td>Mode Sense(10)</td>
<td>Report parameters and other device Information - geometry, other</td>
<td>a series of Data In UPIU's</td>
<td>X</td>
</tr>
<tr>
<td>Report LUNS(12)</td>
<td>Report the accessible logical unit inventory</td>
<td>one or more Data IN UPIU's (Most likely one data in UPIU) (8byte+8*n) (n = LUN)</td>
<td>X</td>
</tr>
<tr>
<td>Verify(10)</td>
<td>Verify medium data is same as transferred data To determine if specific LBA's are accessible</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Format Unit(6)</td>
<td>Format medium into logical blocks, manage medium and defects</td>
<td>Ready To Transfer UPIU</td>
<td>a Data Out UPIU containing the number of bytes</td>
</tr>
<tr>
<td>Send Diagnostic(6)</td>
<td>Perform diagnostic operations on LU or device</td>
<td>Ready To Transfer UPIU</td>
<td>a Data Out UPIU containing the number of bytes</td>
</tr>
<tr>
<td>Synchronize Cache(10)</td>
<td>Recording most recent device data to medium</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
SCSI Command & Operation

✓ Scsi commands operation by Command Flag Type

• No Data (Test Unit Ready, Start Stop Unit, Verify)
• Data from Device (Inquiry, Read6, Request Sense, Read Capacity, Mode Sense, Report LUN)
• Data to Device (Write6, Mode Select, Unmap, Format Unit)

✓ Response : target success, fail, vendor specific
✓ SCSI Status : GOOD, CHECK CONDITION BUSY Etc...
SCSI Command & Operation (TUR)

1. Indicating Interrupt
2. Read Command UPIU From Descriptor Buffer
   1) Transaction Type : Identify UPIU
   2) LUN : Routing to Task Queue in LU
   3) Operation Code : Identify Command Type
   4. Response (Make Response UPIU & Descriptor)
SCSI Command & Operation

✓ Scsi commands operation by Command Flag Type
  • No Data (Test Unit Ready, Start Stop Unit, Verify)
  • Data from Device (Inquiry, Read6, Request Sense, Read Capacity, Mode Sense, Report LUN)
    • Data to Device (Write6, Mode Select, Unmap, Format Unit)

✓ FUA : 1 – cache on , 0 – cache off
  ✓ LBA : Start block
  ✓ Transfer Length : Size

HOST

Data IN UPIU

Device

Status Response
SCSI Command & Operation

✓ Scsi commands operation by Command Flag Type
  • No Data (Test Unit Ready, Start Stop Unit, Verify)
  • Data from Device (Inquiry, Read6, Request Sense, Read Capacity, Mode Sense, Report LUN)

• Data to Device (Write6, Mode Select, Unmap, Format Unit)

HOST

Data OUT UPIU

✓ LBA : Start block
✓ Transfer Length: Size

Device

Status Response
Descriptor & Attribute

- **Descriptor**: General Configuration for Device-level
- **Attribute**: Frequently Configurable Range Value
- **Flag**: Frequently Configurable BOOL Value
## Descriptor

### Table 12.3: Read descriptor opcode

<table>
<thead>
<tr>
<th>Device Description</th>
<th>Configuration Descriptor</th>
<th>Unit Descriptor</th>
<th>Geometry Descriptor</th>
<th>NAND Unit Descriptor</th>
<th>Power Descriptor</th>
<th>Interconnect Descriptor</th>
<th>Manufacturer Name String</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
</tr>
<tr>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
</tr>
<tr>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Table 12.4: Write descriptor opcode

<table>
<thead>
<tr>
<th>Device Description</th>
<th>Configuration Descriptor</th>
<th>Unit Descriptor</th>
<th>Geometry Descriptor</th>
<th>NAND Unit Descriptor</th>
<th>Power Descriptor</th>
<th>Interconnect Descriptor</th>
<th>Manufacturer Name String</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
</tr>
<tr>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
</tr>
<tr>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
<td>02h</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Confidential**

15.08.21 ~ 18.08.20
HOST Controller Interface

Figure 1 — UFS Architecture Overview

Figure 2 — General architecture of UFS Host Controller Interface.
HCI Register

Figure 3 — A conceptual block diagram of UFS host system

Table: UFS Command Set Layer

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20h</td>
<td>23h</td>
<td>IS</td>
<td>Interrupt Status</td>
</tr>
<tr>
<td>24h</td>
<td>27h</td>
<td>IE</td>
<td>Interrupt Enable</td>
</tr>
<tr>
<td>28h</td>
<td>2Bh</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>30h</td>
<td>33h</td>
<td>HCS</td>
<td>Host Controller Status</td>
</tr>
<tr>
<td>34h</td>
<td>37h</td>
<td>HCE</td>
<td>Host Controller Enable</td>
</tr>
<tr>
<td>38h</td>
<td>3Ah</td>
<td>UCMA</td>
<td>Host UI Error Code PHY Adapter Layer</td>
</tr>
<tr>
<td>3Bh</td>
<td>3Dh</td>
<td>UCMA</td>
<td>Host UI Error Code Data Link Layer</td>
</tr>
<tr>
<td>3Eh</td>
<td>4Eh</td>
<td>UCNI</td>
<td>Host UI Error Code Network Layer</td>
</tr>
<tr>
<td>4Fh</td>
<td>5Fh</td>
<td>UCTRL</td>
<td>Host UI Error Code Transport Layer</td>
</tr>
<tr>
<td>60h</td>
<td>64h</td>
<td>UEXTME</td>
<td>Host UI Error Code DME</td>
</tr>
<tr>
<td>6Ch</td>
<td>6Fh</td>
<td>UCTRL</td>
<td>UTP Transfer Request Interrupt Aggregation Control Register</td>
</tr>
</tbody>
</table>

Table: UFS Transfer Request List Base Address

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50h</td>
<td>53h</td>
<td>UTRLBA</td>
<td>UTP Transfer Request List Base Address</td>
</tr>
<tr>
<td>54h</td>
<td>57h</td>
<td>UTRLAUU</td>
<td>UTP Transfer Request List Base Address Upper 32-Bits</td>
</tr>
<tr>
<td>58h</td>
<td>5Ah</td>
<td>UTRLDO</td>
<td>UTP Transfer Request List Door Bell Register</td>
</tr>
<tr>
<td>5Ch</td>
<td>5Fh</td>
<td>UTRLCL</td>
<td>UTP Transfer Request List Clear Register</td>
</tr>
<tr>
<td>60h</td>
<td>64h</td>
<td>UTRRLR</td>
<td>UTP Transfer Request Run-Stop Register</td>
</tr>
<tr>
<td>68h</td>
<td>6Fh</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Table: UFS Command Control

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>90h</td>
<td>93h</td>
<td>UCMS</td>
<td>UIC Command Register</td>
</tr>
<tr>
<td>94h</td>
<td>97h</td>
<td>UCMDAG0</td>
<td>UIC Command Argument 1</td>
</tr>
<tr>
<td>98h</td>
<td>9Bh</td>
<td>UCMDAG1</td>
<td>UIC Command Argument 2</td>
</tr>
<tr>
<td>9Ch</td>
<td>9Fh</td>
<td>UCMDAG2</td>
<td>UIC Command Argument 3</td>
</tr>
<tr>
<td>A0h</td>
<td>A7h</td>
<td>AFH</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
UTP Transfer Request

1. SetCommand Type
2. Data Direction
3. Interrupt
4. Initialize OCS with 0xF
5. Set UTP Command Descriptor Base Address
6. Set Response Offset & Length
7. Set PRDT Offset & Length
8. Check UTP Transfer Request List Run Stop Register
9. Set UTP Transfer Request Interrupt Aggregation Control Register to 1
10. Set Counter and Timer Reset bit to 1
11. Interrupt aggregation counter threshold
12. Interrupt aggregation timer threshold
13. Set Doorbell register(UTRLDBR) to 1

Figure 4 — UTP Transfer Request Descriptor

Figure 6 — Data structure for Physical Region Description Table

Figure 5 — UTP Command Descriptor (UCD)
UTP Transfer Request Completion

If Regular Command(UTRD.I == 0)
1) Check IA interrupt counter
2) If Counter is 0, IA Timer is running

Four conditions which make IS.UTRCS is set
1) UTRD.I == 1
2) When Counter reach Interrupt threshold(IACTH) value
3) Timeout occurs (IATOVAL)
4) When OCS value is not SUCCESS

If IE.UTRCE is set, Interrupt is disabled
Host Controller link startup

Figure 8 — Host controller link startup sequence
Host Power Mode Change

**UIC does not define high-level primitive for setting these power mode change parameters. DME_SET of UIC attributes must be used.**

- DME_SET (PA_ActiveTxDataLanes, ...)
- DME_SET (PA_ActiveRxDataLanes, ...)
- DME_SET (PA_TXGear, ...)
- DME_SET (PA_RXGear, ...)
- DME_SET (PA_TXTermination, ...)
- DME_SET (PA_RXTermination, ...)
- DME_SET (PA_HS, ...)

**Preparing the remote L2 timeout values for the next power mode.**
This is equivalent to preparing the RemoteL2TimerData parameter for UIC’s DME_POWER_MODE req. No change is performed on UIC.

- DME_SET (PA_PWRModeUserData[0], ...)
- DME_SET (PA_PWRModeUserData[11], ...)

**Preparing the local L2 timeout values for the next power mode.**
This is equivalent to preparing the LocalL2TimerData parameter for UIC’s DME_POWER_MODE req. No change is performed on UIC.

- DME_SET (DME_Local_FC0ProtectionTimeout, ...)
- DME_SET (DME_Local_TC0ReplayTimeout, ...)
- DME_SET (DME_Local_AF0ReqTimeout, ...)
- DME_SET (DME_Local_FC1ProtectionTimeout, ...)
- DME_SET (DME_Local_TC1ReplayTimeout, ...)
- DME_SET (DME_Local_AF1ReqTimeout, ...)

**Setting PA_PWRMode triggers the power mode change.**
PA_PWRMode[3:0] and PA_PWRMode[7:4] correspond to TxPowerMode and RxPowerMode DME_POWER_MODE req parameters, respectively.

- DME_SET (PA_PWRMode, ...)

**The above PA or DME_LocalL2TimerData attributes must not be set during the power mode change.**

---

**UIC power mode change procedure**

The power mode change procedure is completed when IS_UPMS is set to ‘1’. This is equivalent to DME_POWER_MODE.ind.

---

**Figure 9 — UIC Power mode change**
UFS Host & Device System

UFS HOST - PF100 (Application Processor)

- AUTO TEST APP (For TG 64.5)
- CHAR DRIVER
- LLD

- BLOCK DRIVER with File System

UFS PROTOCOL FIRMWARE

- UFS FTL FIRMWARE

- UFS PATHFINDER
TEST Implementation

- Device driver and AUTO TEST application have been developed on PC based HOST environment
- HOST(AP) / device firmware implemented
- Technical understanding of AP side will be helpful for co-work with AP vendors
- Simplicity and easy debugging are most important factor of AUTO TEST application

7.12.1 Test Case Id: UFS_Read6_01

If this test case is used to BOOT well-known logical unit, WRITE command may not be issued from host because WRITE command is not supported on BOOT well-known logical unit. Data comparison may not be processed.

<table>
<thead>
<tr>
<th>Ref. specs Section</th>
<th>UFS: Section 11.3.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Purpose</td>
<td>To Verify the READ (6) command reads 256 blocks of data when TRANSFER LENGTH = 00h.</td>
</tr>
<tr>
<td>Test Procedure</td>
<td>[Precondition]: 1. Issue READ CAPACITY (10) command and verify LU has more than or equal to 256 Logical blocks [Main]: 1. Issue WRITE (6) command. 2. Issue READ (6) command. 3. Verify the Expected Output. [Cleanup]: None</td>
</tr>
</tbody>
</table>

Input parameter values

| WRITE (6) | LBA = 00h, TRANSFER LENGTH = 00h, CONTROL = 00h. Any Data can be written. |
| READ (6)  | LBA = 00h, TRANSFER LENGTH = 00h, CONTROL = 00h. |

Expected Output

1. The READ (6) command response shall be:
<table>
<thead>
<tr>
<th>Response</th>
<th>Status</th>
<th>Sense Key</th>
<th>Additional Sense Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target  Success</td>
<td>GOOD</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

2. Compare data buffers of WRITE (6) and READ (6) commands. Data shall be same.

Command #2 : Final Result...OK!

[root@ufs auto_test_1]# find . -name "*.chS" | xargs wc -l
74 . /Source/Pathfinder_Linux_Drv_V01/inc/common/common.h
296 . /Source/Pathfinder_Linux_Drv_V01/inc/common/ufs_AdaptationLayer.h
62956 total
UFS Device
HOST & Device E2E Interface

HOST DEVICE (Client)

SCSI Initiator

Application Client

1 to M Application Clients per Host

Unipro Port

UFS DEVICE (Server)

SCSI Target

Device Server

Task Manager

Logical Unit

LUN

Resources under the control of Logical Unit (memory area, etc.)

1 to N Logical Units per Device

Figure 7-1 UFS SCSI Domain
Logical Unit

- LU: Externally addressable, independent, processing entity.
- Device contains 1 or Max 8 Normal logical units & 4 Well-known LU
- A Logical unit contains
  - DEVICE SERVER: Processing SCSI Commands
  - TASK MANAGER: Performs Task management Functions
  - TASK SET: A Conceptual Group of Commands

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Logical Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>bLUEnable</td>
<td>Logical Unit Enable</td>
<td>LU 0, ..., LU 7</td>
</tr>
<tr>
<td>bBootLunID</td>
<td>Boot LUN ID</td>
<td>LU 0, ..., LU 7</td>
</tr>
<tr>
<td>bLUWriteProtect</td>
<td>Logical Unit Write Protect</td>
<td>LU 0, ..., LU 7</td>
</tr>
<tr>
<td>bMemoryType</td>
<td>Memory Type</td>
<td>LU 0, ..., LU 7</td>
</tr>
<tr>
<td>dNumAllocUnits</td>
<td>Number of allocation units assigned to the logical unit. The value shall be calculated considering the capacity adjustment factor of the selected memory type.</td>
<td>LU 0, ..., LU 7</td>
</tr>
<tr>
<td>bDataReliability</td>
<td>Data Reliability</td>
<td>LU 0, ..., LU 7</td>
</tr>
<tr>
<td>bLogicalBlockSize</td>
<td>Logical Block Size</td>
<td>LU 0, ..., LU 7</td>
</tr>
<tr>
<td>bProvisioningType</td>
<td>Provisioning Type</td>
<td>LU 0, ..., LU 7</td>
</tr>
</tbody>
</table>
```
LU Configuration

Protocol Address

<table>
<thead>
<tr>
<th>Enhanced Area</th>
<th>Normal Area</th>
<th>Hidden Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBA=0</td>
<td>LBA=1</td>
<td>LBA=2</td>
</tr>
<tr>
<td>LBA=3</td>
<td>LBA=4</td>
<td>LBA=5</td>
</tr>
<tr>
<td>LBA=6</td>
<td>LBA=7</td>
<td>LBA=8</td>
</tr>
<tr>
<td>LBA=9</td>
<td>LBA=10</td>
<td>LBA=11</td>
</tr>
</tbody>
</table>

NAND Address

- Logical Unit 0
- Logical Unit 1
- Logical Unit 2
- Logical Unit 3
- Logical Unit 4
- Logical Unit 5
- Logical Unit 6
- Logical Unit 7
- RPMB W-LU (16MB)
- Admin Area
LU Configuration

8.4 UFS Descriptor

TABLE 1 Configuration Descriptor Header and Device Descriptor Configurable parameters

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>1</td>
<td>bLength</td>
<td>90h</td>
</tr>
<tr>
<td>01h</td>
<td>1</td>
<td>bDescriptorType</td>
<td>01h</td>
</tr>
<tr>
<td>02h</td>
<td>1</td>
<td>bNumberLU</td>
<td>00h</td>
</tr>
<tr>
<td>03h</td>
<td>1</td>
<td>bBootEnable</td>
<td>01h</td>
</tr>
<tr>
<td>04h</td>
<td>1</td>
<td>bDescrAccessEn</td>
<td>01h</td>
</tr>
<tr>
<td>05h</td>
<td>1</td>
<td>bInitPowerMode</td>
<td>01h</td>
</tr>
<tr>
<td>06h</td>
<td>1</td>
<td>bHighPriorityLU</td>
<td>00h</td>
</tr>
<tr>
<td>07h</td>
<td>1</td>
<td>bSecureRemoval</td>
<td>00h</td>
</tr>
<tr>
<td>08h</td>
<td>1</td>
<td>bInitAstralCCLevel</td>
<td>00h</td>
</tr>
<tr>
<td>09h</td>
<td>2</td>
<td>wPeriodicRTCUpdate</td>
<td>0000h</td>
</tr>
<tr>
<td>08h:0Fh</td>
<td>5</td>
<td>Reserved</td>
<td>00h</td>
</tr>
</tbody>
</table>

TABLE 2 Unit Descriptor Configurable parameters

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Name</th>
<th>LUN=0</th>
<th>LUN=1</th>
<th>LUN=2</th>
<th>LUN=3</th>
<th>LUN=4</th>
<th>LUN=5</th>
<th>LUN=6</th>
<th>LUN=7</th>
</tr>
</thead>
<tbody>
<tr>
<td>10h+16LUN</td>
<td>1</td>
<td>blUEnable</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
<td>01h</td>
</tr>
<tr>
<td>11h+16LUN</td>
<td>1</td>
<td>bBootUnitID</td>
<td>00h</td>
<td>00h</td>
<td>01h</td>
<td>02h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
</tr>
<tr>
<td>12h+16LUN</td>
<td>1</td>
<td>blUEWriteProtect</td>
<td>00h</td>
<td>01h</td>
<td>02h</td>
<td>00h</td>
<td>00h</td>
<td>01h</td>
<td>02h</td>
<td>00h</td>
</tr>
<tr>
<td>13h+16LUN</td>
<td>1</td>
<td>bMemoryType</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
</tr>
<tr>
<td>14h+16LUN</td>
<td>4</td>
<td>dDataAllocUnits</td>
<td>1000h</td>
<td>1000h</td>
<td>1000h</td>
<td>1000h</td>
<td>1000h</td>
<td>1000h</td>
<td>1000h</td>
<td>1000h</td>
</tr>
<tr>
<td>15h+16LUN</td>
<td>1</td>
<td>bDataReliability</td>
<td>00h</td>
<td>01h</td>
<td>00h</td>
<td>01h</td>
<td>00h</td>
<td>01h</td>
<td>00h</td>
<td>01h</td>
</tr>
<tr>
<td>16h+16LUN</td>
<td>1</td>
<td>blLogicalBlockSize</td>
<td>0Ch</td>
<td>0Ch</td>
<td>0Ch</td>
<td>0Ch</td>
<td>0Ch</td>
<td>0Ch</td>
<td>0Ch</td>
<td>0Ch</td>
</tr>
<tr>
<td>17h+16LUN</td>
<td>1</td>
<td>bProvisioningType</td>
<td>00h</td>
<td>02h</td>
<td>03h</td>
<td>00h</td>
<td>02h</td>
<td>03h</td>
<td>00h</td>
<td>00h</td>
</tr>
<tr>
<td>18h+16LUN</td>
<td>2</td>
<td>wContextCapabilities</td>
<td>00h</td>
<td>01h</td>
<td>02h</td>
<td>03h</td>
<td>04h</td>
<td>05h</td>
<td>00h</td>
<td>01h</td>
</tr>
<tr>
<td>19h+16LUN</td>
<td>3</td>
<td>Reserved</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
<td>00h</td>
</tr>
</tbody>
</table>
Higher Priority LU

Figure 13-11: Host-Device interaction for LU with same the priority

Figure 13-12: Host-Device interaction for LU with different priorities
Boot Sequence

- UFS CMDs and Modes required by boot sequence

- UIC Layer Initialization Phase
  After Link Startup, By Default, M-PHY Speed is PWM G1 Mode

- UTP Layer Initialization Phase
  We expect that Host send only one NOP OUT UPIU, but, multi-NOPs are OK. The Host can change bRefClkFreq value of descriptor after UTP layer initialization phase.

- BOOT W-LU Ready Phase
  UNIT ATTENTION CONDITION will be cleared after sending RESP

- Boot W-LU Transfer Phase

- Application Layer Initialization Phase

- Device Initialization Done
SCSI Command & Operation (Unmap)

Erase operation Type

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Discard</td>
<td>(unmap=any)</td>
</tr>
<tr>
<td>Erase</td>
<td>(unmap=0)</td>
</tr>
<tr>
<td>Purge</td>
<td>(emmc = sanitize)</td>
</tr>
<tr>
<td>Format</td>
<td>erase + purge</td>
</tr>
</tbody>
</table>

Descriptor of regarding Erase operation

<table>
<thead>
<tr>
<th>bProvisioningType (OTP)</th>
<th>ReadCapacity16 (TPE)</th>
<th>ReadCapacity16 (TPRZ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Don't care</td>
</tr>
<tr>
<td>1</td>
<td>Don't care</td>
<td>Don't care</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Erase in the multiple of dEraseBlockSize unit (bigger size than logical blocks)

Erase or Discard

One time programmable value during LU Configuration
Erase or Discard is determined by ReadCapacity16 for TPRZ field value

bProvisioningType = 2
Unmap Command = Discard
Any value read

bProvisioningType = 3
Unmap Command = Erase
0 value read
# Write Protection

## UNIT DESCRIPTOR

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Name</th>
<th>MDV (1)</th>
<th>User Conf. (2)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>05h</td>
<td>1</td>
<td>bLUWriteProtect</td>
<td>00h</td>
<td>Yes</td>
<td>Logical Unit Write Protect&lt;br&gt; 00h: LU not write protected&lt;br&gt; 01h: LU write protected when fPowerOnWPEn = 1&lt;br&gt; 02h: LU permanently write protected when fPermanentWPEn = 1&lt;br&gt; 03h: Reserved (for UFS Security Extension specification)&lt;br&gt; Others: Reserved</td>
</tr>
</tbody>
</table>

## FLAGS

<table>
<thead>
<tr>
<th>IDN</th>
<th>Name</th>
<th>Type</th>
<th>Type (1)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>02h</td>
<td>fPermanentWPEn</td>
<td>Read / Write once</td>
<td>D</td>
<td>0</td>
<td>Permanent Write Protection Enable&lt;br&gt; fPermanentWPEn enables permanent write protection on all logical units configured as permanent protected; it cannot be toggled or cleared once it is set.&lt;br&gt; 00h: Permanent write protection disabled</td>
</tr>
<tr>
<td>03h</td>
<td>fPowerOnWPEn</td>
<td>Read / Power on reset</td>
<td>D</td>
<td>0</td>
<td>Power On Write Protection Enable&lt;br&gt; fPowerOnWPEn enables the write protection on all logical units configured as power on write protected.&lt;br&gt; If fPowerOnWPEn is equal to one and the device receive a Query Request to clear or toggle this flag, the Query Request shall fail and Response field shall be set to &quot;F3h&quot; (Parameter already written).&lt;br&gt; The device shall set fPowerOnWPEn to zero in the event of power cycle or hardware reset.&lt;br&gt; 0b: Power on write protection disabled.&lt;br&gt; 1b: Power on write protection enabled.</td>
</tr>
</tbody>
</table>

![Diagram](image-url)
Q & A
Thanks!!!