UFS Logical Driver API

1 Introduction

The Document describes the Generic APIs for the logical driver of UFS host controller. This abstracted APIs will hides the datatype complexity defined at Physical level device driver(PDD) and helps user to build UFS application with reduced data type complexity and also aid in adopting the APIs in test automation framework like Open Hardware Software Interface standard(OpenHSI) .

1.1 References

[1] JEDEC 220C, UFS 2.1 Specification document.

2 API reference: UFS driver

2.1 ufs_init

Synopsis:
```
status_t ufs_init(struct ufs_struct *ufshci);
```

Purpose: Initialization of UFS host controller at logical layer.

Arguments:
ufshci - Pointer to ufs_struct

Pre-condition: `ufs_preinit(struct ufs_struct *ufshci, unsigned int vendor_id, unsigned int device_id).`

Return values: UFS_SUCCESS or FAILURE.

2.2 ufs_preinit

Synopsis:
```
int ufs_preinit(struct ufs_struct *ufshci,unsigned int vendor_id, unsigned int device_id);
```

Purpose: allocating and initializing ufs_struct.
Arguments:
ufshci - Pointer to ufs_struct
vendor_id - ufs vendor id
device_id – ufs device id

Return values: 0.

2.3 ufs_deinit

Synopsis:
```
status_t ufs_deinit(struct ufs_struct *ufshci);
```

Purpose: de-initialization of UFS host controller at logical layer deallocates the memory of private data structure**.**

Arguments:
ufshci - Pointer to ufs_struct

Return values: void.

2.4 ufs_upiu_create_nop_out

Synopsis:

```
    int ufs_upiu_create_nop_out(struct ufs_struct *ufshci, struct
utp_transfer_cmd_desc *cmd);
```

Purpose: Creates NOP out UPIU that needs to be sent to device

Arguments:

ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information

Return values: UFS_SUCCESS or FAILURE.

2.5 ufs_upiu_receive_nop_in

Synopsis:

```
    int ufs_upiu_receive_nop_in(struct ufs_struct *ufshci, struct
utp_transfer_cmd_desc *cmd);
```

Purpose: Sent from Device as a response to NOP out

Arguments:

ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information

Return values: UFS_SUCCESS or FAILURE.

2.6 ufs_create_scsi_cmd

Synopsis:

```
     int ufs_create_scsi_cmd(struct ufs_struct *ufshci, unsigned
int flag, unsigned int lba,unsigned int size, unsigned char
pagecode, unsigned char subpage_code, unsigned char
power_flush_loej_start_bit);
```

Purpose: Creates SCSI command UPIU that needs to be sent to device

Arguments:

ufshci - Pointer to ufs_struct
flag - Flag value that needs to be configure
in the 1st bit of the SCSI command
lba - logical block addr
size - size value
pagecode - Specifies which mode page to return
buffer id: Specifies a buffer within the
logical unit
select report : Specifies the type of
logical unit addresses that shall be reported
subpage_code - Specifies which subpage mode page to return

power_flush_loej_start_bit - value of power_flush, no_flush, loej and start bits

Return values: **UFS_SUCCESS or FAILURE.**

2.7 ufs_upiu_create_cmd

Synopsis:

int ufs_upiu_create_cmd(struct ufs_struct *ufshci, struct utp_transfer_cmd_desc *cmd, unsigned char *data_buff_id, unsigned int expected_data_len, unsigned char lun, unsigned char flag);

Purpose: Creates UPIU command to send to the device

Arguments:

ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information
data_buff_id - Unique_id that holds the data to be sent
expected_data_len - expected data length
lun - logical Unit Number
(0 <= LUN <= 127)
flag - flag value
(The content of the flags field may vary based on transaction type opcode. Reference Table 10-12)

Return values: UFS_SUCCESS or FAILURE.

2.8 ufs_upiu_receive_response

Synopsis:

int ufs_upiu_receive_response(struct ufs_struct *ufshci, struct utp_transfer_cmd_desc *cmd);

Purpose: decodes the response received from device to host on UPIU command sent.

Arguments:

ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information

Return values: UFS_SUCCESS or FAILURE.

2.9 ufs_upiu_create_query_request

Synopsis:

int ufs_upiu_create_query_request(struct ufs_struct *ufshci, struct utp_transfer_cmd_desc *cmd, unsigned char *data_xfer_buff_id, unsigned char flag, unsigned int data_segment_len, ufs_query_type query_op, ufs_query_function query_fun, unsigned char idn_value, unsigned char index_val, unsigned char select_val, unsigned short length, unsigned int value_data, unsigned char *min_um_area_size_id);

Purpose: creates query request UPIU to be sent from host to device

Arguments:

ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information
flag - flag value
('0' or '1'/ 'TRUE' or 'FALSE' for clear and reset respectively)

data_buff_id- data segment length
query_op - Query op-code specifies query request operation read/write.
(Reference Table 10-31.)

query_fun - unique value to each query function
(Reference Table 10-29.)
idn_value - index value
(0-255)
select_val - selector value
(depends on descriptor length.)
length - query data length
(less than, or equal to or greater than actual descriptor size)
value_data - write attribute value
(32 bit, right justified big endian value, unused upper bit shall be set to zero.)

min_um_area_size_id - Unique id that holds Minimum Unified memory area size

Return values: UFS_SUCCESS or FAILURE.

2.10 ufs_upiu_receive_query_response

Synopsis:

int ufs_upiu_receive_query_response(struct ufs_struct *ufshci, struct utp_transfer_cmd_desc *cmd, unsigned char *data_read_buff_id, unsigned char *min_um_area_size_id);

Purpose: checks for the success of the response received from device

Arguments:

ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information
data_read_buff_id - Unique id that holds the read data
min_um_area_size_id - Unique id that holds Minimum Unified memory area size

Return values: UFS_SUCCESS or FAILURE.

2.11 **ufs_upiu_create_task_management_request**

Synopsis:
int ufs_upiu_create_task_management_request(struct ufs_struct *ufshci, struct utp_task_mgmt_req_desc *utmrd, unsigned char flag, unsigned char lun, unsigned char tm_function);

Purpose: creates task management request UPIU that needs to be sent to device

Arguments:

ufshci	-	Pointer to ufs_struct
utmrd	-	unique value that holds UTMRD data
flag	-	flag value
lun	-	logical unit number (0-127)
tm_function	-	task management function value (Reference Table 10-23)

Return values: UFS_SUCCESS or FAILURE.

2.12 **ufs_upiu_receive_task_management_response**

Synopsis:
int ufs_upiu_receive_task_management_response(struct ufs_struct *ufshci, struct utp_task_mgmt_req_desc *utmrd);

Purpose: decodes and check the response received from device

Arguments:

ufshci	-	Pointer to ufs_struct
utmrd	-	Unique id that holds UTMRD related value

Return values: UFS_SUCCESS or FAILURE.

2.13 **ufs_destroy_command**

Synopsis:
int ufs_destroy_command(struct utp_transfer_cmd_desc *cmd);

Purpose: releases the memory assigned to the input command buffer id

Arguments:

cmd	-	Unique id that hold command information

Return values:

2.14 **ufs_create_replay_protected_memory_block_frame**

Synopsis:

void ufs_pro_create_replay_protected_memory_block_frame (unsigned char *rpmb_buffer_id, unsigned char *data_buffer_id, unsigned int counter, unsigned short addr, unsigned short block_count, unsigned short req_code);

Purpose: created RPMB data frame

Arguments:

rpmb_buff_id	-	Unique id that holds RPMB related info
data_buff_id	-	Unique Id that holds the data that needs to be transferred
counter	-	Write Counter value (0 – 2^20 -1.)
addr	-	Address location to read from or to write to (0-255)
block_count	-	Number of blocks to read or write (0-255)
req_code	-	RPMB message request (Reference Table 12-5.)

Return values: void

3 LDD WRAPPER API

3.1 ufshci_interrupt_routine

Synopsis:
int ufshci_interrupt_routine(struct ufs_struct *ufshci);

Purpose: It handles the interrupts of UFSHCI. It checks the interrupt status register and sets respective events.

Arguments:
ufshci - Pointer to ufs_struct

Return values: 0

3.2 ufshci_check_device_bus_master_support

Synopsis:
int ufshci_check_device_bus_master_support(struct ufs_struct *ufshci);

Purpose: Checks the bus master support capabilities.

Arguments:
ufshci - Pointer to ufs_struct

Return values: 0

3.3 ufshci_check_device_management_entity_endpoint_reset

Synopsis:
int ufshci_check_device_management_entity_endpoint_reset(struct ufs_struct *ufshci);

Purpose: check whether endpoint reset is supported by device or not.

Arguments:
ufshci - Pointer to ufs_struct

Return values: 0

3.4 ufshci_config_auto_hibernation

Synopsis:
int ufshci_config_auto_hibernation(struct ufs_struct *ufshci, unsigned char timer_scale, unsigned short timer_value, unsigned int hclk_div_value);

Purpose: configures auto hibernation mode after idle state.

Arguments:
ufshci - Pointer to ufs_struct
timer_scale - timer scale value
timer_value - timer value
hclk_div_value - HCLK divider value

Return values: 0

3.5 ufshci_verify_hibernation

Synopsis:
int ufshci_verify_hibernation(struct ufs_struct *ufshci);

Purpose: checks whether the UNIPRO stack has entered or exited successful in hibernation mode and prints the status of hibernation request .

Arguments:
ufshci - Pointer to ufs_struct

Return values: 0

3.6 ufshci_config_intr_aggregation

Synopsis:
int ufshci_config_intr_aggregation(struct ufs_struct *ufshci, unsigned char count, unsigned char timer);

Purpose: enables interrupt aggregation value and configures counter and timer values.

Arguments:
ufshci - Pointer to ufs_struct
count - counter value
timer - timer value

Return values: void

3.7 ufshci_reset_intr_aggregation

Synopsis:
int ufshci_reset_intr_aggregation(struct ufs_struct *ufshci);

Purpose: resets interrupr aggregation and resets counter/timer values.

Arguments:

ufshci - Pointer to ufs_struct
count - counter value
timer - timer value

Return values: void

3.8 ufshci_configure_buffer_size

Synopsis:

int ufshci_configure_buffer_size(struct ufs_struct *ufshci, unsigned char buffer_size);

Purpose: configures buffer size.

Arguments:

buffer_size - Max Burst length value

Return values: void

3.9 ufshci_configure_unified_memory_extention_mode

Synopsis:

int ufshci_configure_unified_memory_extention_mode(struct ufs_struct *ufshci, unsigned long long ume_base_addr, unsigned int umaomax);

Purpose: configures the UME Base address and upper boundary of the Unified Memory area.

Arguments:

ufshci - Pointer to ufs_struct
ume_base_addr - UME base address
umaomax - Upper boundary of the Unified Memory area

Return values: void

3.10 ufshci_enables_unified_memory_extention

Synopsis:

int ufshci_enable_unifies_memory_extention(struct ufs_struct *ufshci);

Purpose: enables unified memory extension feature.

Arguments:

ufshci - Pointer to ufs_struct

Return values: void

3.11 ufshci_read_purge_attribute

Synopsis:
int ufshci_read_purge_attribute(struct ufs_struct *ufshci,struct utp_transfer_cmd_desc *cmd);

Purpose: reads the value present in simplified register offset.

Arguments:
ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information

Return values: 0

3.12 ufshci_register_write

Synopsis:
int ufshci_register_write(struct ufs_struct *ufshci, unsigned int register_offset, unsigned int value);

Purpose: writes the value to specified register offset.

Arguments:
ufshci - Pointer to ufs_struct
value - value that needs to be stored
register_offset - register offset

Return values: 0

3.13 ufshci_register_read

Synopsis:
int ufshci_register_read(struct ufs_struct *ufshci, unsigned int *buffer, unsigned int register_offset);

Purpose: reads the value present in simplified register offset.

Arguments:
ufshci - Pointer to ufs_struct
buffer - buffer to store read register
register_offset - register offset

Return values: 0

3.14 ufshci_unipro_config_dme

Synopsis:
int ufshci_unipro_config_dme(struct ufs_struct *ufshci, unsigned char dme_opcode, unsigned short mib_attr, unsigned short gen_select_idx, unsigned char reset_level, unsigned char attr_set_type, unsigned int mib_write_val);

Purpose: configures UNIPRO DME primitives through UIC command registers.

Arguments:

ufshci - Pointer to ufs_struct
dme_opcode - Op-code of a UIC Command to be dispatched to local UIC layer
mib_attr - the ID of the attribute of the requested
gen_select_idx - targeted M-PHY data lane or CPort or Test Feature when relevant
reset_level - reset type
attr_set_type - indicates whether attribute value is normal or static
mib_write_value - indicates value of attribute to be set

Return values: 0

3.15 ufshci_unipro_check_cmd_completion

Synopsis:

int ufshci_unipro_check_cmd_completion(struct ufs_struct *ufshci, unsigned char dme_opcode);

Purpose: Checks the command completion status.

Arguments:

ufshci - Pointer to ufs_struct
dme_opcode - OPCODE of UIC command

Return values: 0

3.16 ufshci_utrd_doorbell_complete

Synopsis:

int ufshci_utrd_doorbell_complete(struct ufs_struct *ufshci);

Purpose: indicates the host device that the command is configured in the slot by setting respective slot in doorbell register and waits until command completion interrupts occurs.

Arguments:

ufshci - Pointer to ufs_struct

pre-condition: ufshci_utrd_descriptor()

Return values: 0

3.17 ufshci_utmrd_check_response

Synopsis:

int ufshci_utmrd_check_response(struct ufs_struct *ufshci);

Purpose: It checks the service response fields of task management response UPI.

Arguments:

ufshci - Pointer to ufs_struct

Return values: 0

3.18 ufshci_utmrd_descriptor

Synopsis:

int ufshci_utmrd_descriptor(struct ufs_struct *ufshci, struct utp_task_mgmt_req_desc *utmrd);

Purpose: fills up UTMRD descriptor.

Arguments:

ufshci - Pointer to ufs_struct
utmrd - Unique id pointing to UTMRD related info

Return values: 0

3.19 ufshci_utrd_descriptor

Synopsis:

int ufshci_utrd_descriptor(struct ufs_struct *ufshci,struct utp_transfer_cmd_desc *cmd, unsigned short prdt_len, unsigned char dir, unsigned char intr_bit);

Purpose: fills up UTRD descriptor.

Arguments:

ufshci - Pointer to ufs_struct
cmd - Unique id that hold command information
prdt_len - PRDT length
dir - data direction
intr_bit - interrupt bit

Return values: 0

3.20 ufshci_utmrd_doorbell_complete

Synopsis:

int ufshci_utmrd_doorbell_complete(struct ufs_struct *ufshci);

Purpose: It indicates that command is configured in the slot by setting respective slot in doorbell register and waits until command completion interrupts occurs, up which it checks OCS field.

Arguments:

ufshci - Pointer to ufs_struct

pre-condition: ufshci_utrmd_descriptor()

Return values: 0

3.21 ufshci_handle_hibernate_error

Synopsis:
int ufshci_handle_hibernate_error(struct ufs_struct *ufshci);

Purpose: To handle this error, it performs re-initialization of the UFS host controller.

Arguments:
ufshci - Pointer to ufs_struct
Return values: 0

3.22 ufshci_handle_device_fatal_error

Synopsis:
int ufshci_handle_device_fatal_error(struct ufs_struct *ufshci);

Purpose: It waits for device fatal error interrupt to occur. To handle this error it performs DME reset and re-initialization of the UFS host controller. .

Arguments:
ufshci - Pointer to ufs_struct

Return values: 0

3.23 ufshci_handle_host_controller_fatal_error

Synopsis:
int ufshci_handle_host_controller_fatal_error(struct ufs_struct *ufshci);

Purpose: To handle this error, it performs re-initialization of the UFS host controller.

Arguments:
ufshci - Pointer to ufs_struct
Return values: 0

3.24 ufshci_handle_system_bus_error

Synopsis:
int ufshci_handle_system_bus_error(struct ufs_struct *ufshci);

Purpose: This task handles a system bus error which occurs due to bad memory pointers, bad operation and protection violation. It waits for system bus interrupt to occur.

Arguments:

ufshci - Pointer to ufs_struct

Return values: 0

3.25 ufshci_handle_uic_error

Synopsis:
int ufshci_handle_uic_error(struct ufs_struct *ufshci);

Purpose: waits for an UFS interconnect error interrupt to occur. Upon interrupt it checks for UIC layer error registers to know which layer caused an error. To handle this error, it performs re-initialization of the UFS host controller.

Arguments:
ufshci - Pointer to ufs_struct
Return values: 0

3.26 ufshci_erase_test

Synopsis:
int ufshci_erase_test(struct ufs_struct *ufshci, struct utp_transfer_cmd_desc *cmd, unsigned char *buf_id, unsigned char lun, unsigned short num_of_desc);

Purpose: provides type descriptor with value 0x3 and sends the UNMAP command to the device to perform erase operation

Arguments:
ufshci - Pointer to ufs_struct
cmd - pointer to transfer cmd
buf_id - buffer_id
lun - logical unit number
num_of_desc - number of descriptors

Return values: 0

3.27 ufshci_release_memory

Synopsis:
int ufs_release_memory(char *buffer_id);

Purpose: free-up the acquired memory indicated by a unique id.

Arguments:
buf_id - buffer_id

