PCIe Logical Driver API

1. Introduction

The Document describes the Generic APIs for the logical driver of PCIe RC and EP controller. These abstracted APIs will hide the datatype complexity defined at Physical level device driver(PDD) and helps users to build PCIe applications with reduced data type complexity and also aid in adopting the APIs in test automation framework like Open Hardware Software Interface standard(OpenHSI) .

2. RC LDD WRAPPER API

2.1 pcieRcInit

Synopsis:
status_t pcieRcInit(pcieRootcompDataHandle_t *pdata);

Purpose: Initialize RC

Arguments:
*pdata Pointer to RC private data structure of PCIe stack

Return Value:
0 on success and -1 on failure

2.2 pcieRcCfg

Synopsis:
```
status_t pcieRcCfg(pcieRootcompDataHandle_t *pdata,
        unsigned long pcieAddr,
        unsigned long cpuAddr,
        unsigned long epCfgAddr,
        unsigned long rcCfgAddr,
        unsigned long addrSpaceSize,
        unsigned char numRdDmaChnls,
        unsigned char numWrDmaChnls,
        unsigned char numObRegions,
        unsigned char numIbRegions);
```

Purpose: Configure RC

Arguments:

pcieAddr	PCIe Address
cpuAddr	CPU Address
epCfgAddr	EP configuration Address
rcCfgAddr	RC configuration Address
addrSpaceSize	Address space size
numRdDmaChnls	Number of read DMA channels
numWrDmaChnls	Number of write DMA channels
numObRegions	Number of outbound regions
numIbRegions	Number of inbound regions

Return Value:
0 on success -1 on failure

2.3 pcieRcConfigAtuIbRegion

Synopsis:

```
status_t pcieRcConfigAtuIbRegion(pcieRootcompDataHandle_t *pdata,
        unsigned long baseAddr,
        unsigned long trgtAddr,
        unsigned long len,
        unsigned int type,
        unsigned int ibRgn,
        unsigned char barNum);
```

Purpose: Configure RC ATU inbound region

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
baseAddr	Base address
trgtAddr	Target address
len	Length
type	Type
ibRgn	Inbound region
barNum	BAR number

Return Value:

0 on success and -1 on failure

2.4 pcieRcConfigAtuObRegion

Synopsis:

```
status_t pcieRcConfigAtuObRegion(pcieRootcompDataHandle_t *pdata,
        unsigned long baseAddr,
        unsigned long trgtAddr,
        unsigned long len,
        unsigned int type,
        unsigned int obRgn);
```

Purpose: Configure RC ATU outbound region

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
baseAddr	Base address
trgtAddr	Target address
len	Length
type	Type
obRgn	Outbound region

barNum BAR number

Return Value:
0 on success and -1 on failure

2.5 pcieRcConfigReadWrite

Synopsis:
status_t pcieRcConfigReadWrite(pcieRootcompDataHandle_t *pdata,
unsigned int regOffset,
unsigned int *buffer ,
unsigned int capId,
unsigned char configSpaceSelect,
unsigned char accessSize,
readWriteFlag rw);

Purpose: read or write RC configuration area

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
regOffset	Offset to the register
*buffer	Output buffer into which data is read or whose content is used for write
capId	Capability ID
configSpaceSelect	Selection of configuration space
accessSize	Access size
rw	Read or write action specifier

Return Value:
0 success and -1 on failure

2.6 pcieRcDeinit

Synopsis:
void pcieRcDeinit(pcieRootcompDataHandle_t *pdata);

Purpose: Deinitialize RC and release relevant memory

Arguments:
*pdata Pointer to RC private data structure of PCIe stack

Return Value:

2.7 pcieRcDeinitprint

Synopsis:
void pcieRcDeinitprint(pcieRootcompDataHandle_t *pdata);

Purpose: Deinitialize print API link

Arguments:
*pdata Pointer to RC private data structure of PCIe stack

Return Value:

2.8 pcieRcEnumerate

Synopsis:
status_t pcieRcEnumerate(pcieRootcompDataHandle_t *pdata);

Purpose:
PCIe enumeration of entire fabric under RC

Arguments:
*pdata Pointer to RC private data structure of PCIe stack

Return Value:
0 on success and -1 on failure

2.9 pcieRcGetEpMemDetails

Synopsis:
```
status_t pcieRcGetEpMemDetails(pcieRootcompDataHandle_t *pdata,
        unsigned short vendorId,
        unsigned short pcieDeviceId,
        unsigned char barNum,
        unsigned char bus,
        unsigned char dev,
        unsigned char fun,
        unsigned int *addr,
        unsigned int *size);
```

Purpose:

Get EP memory details. This will provide the base address and size of the memory relative to the asked BAR number of the device.

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
vendorId	EP Vendor ID
pcieDeviceId	EP Device ID
barNum	BAR number
bus	Bus number
dev	Device number
fun	Function number
*addr	Address pointer to EP memory
*size	Pointer to size value

Return Value:
0 on success and -1 on failure

2.10 pcieRcPreInit

Synopsis:
status_t pcieRcPreInit(pcieRootcompDataHandle_t **tmp_pdata);

Purpose:
Allocate private data structures and get driver API pointers

Arguments:

**tmp_pdata	Pointer to RC private data structure of PCIe stack pointer

Return Value:
0 on success and -1 on failure

2.11 pcieRcProgConfigRegion

Synopsis:
status_t pcieRcProgConfigRegion(pcieRootcompDataHandle_t *pdata);

Purpose:
Program RC configuration region

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack

Return Value:
0 on success and -1 on failure

2.12 pcieRcProgPrefetchIoLimit

Synopsis:

```
status_t pcieRcProgPrefetchIoLimit(pcieRootcompDataHandle_t *pdata,
        unsigned int rcBarConfigReg ,
        unsigned int type1BaseLmtCntrl,
        unsigned int enableOrDisable);
```

Purpose:

Program prefetch IO limit

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
rcBarConfigReg	RC BAR configuration register
type1BaseLmtCntrl	type 1 base lmt cntrol
enableOrDisable	Enable or Disable

Return Value:

Value on success and ERRORINVAL on failure

2.13 pcieRcReadEpConfig

Synopsis:

```
status_t pcieRcReadEpConfig(pcieRootcompDataHandle_t *pdata,
        unsigned int *buffer,
        unsigned short offset,
        unsigned char size,
        unsigned char bus,
        unsigned char pcieDevice,
        unsigned char function);
```

Purpose:

Read EP configuration area

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
*buffer	Pointer to output buffer into which data will be read
offset	Register offset
size	size
bus	Bus number
pcieDevice	Device number
function)	Function number

Return Value:

0 on successful read and EINVAL on failure

2.14 pcieRcReadEpMem

Synopsis:

```
status_t pcieRcReadEpMem(pcieRootcompDataHandle_t *pdata,
        unsigned int offset,
        unsigned char *dataPtr,
        unsigned int *addrPtr,
        unsigned int numOfBytes);
```

Purpose: Read EP memory

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
offset	Offset
*dataPtr	Pointer to output data buffer
*addrPtr	Pointer to memory address
numOfBytes	Number of bytes to be read

Return Value:

0 on success and -1 on failure

2.15 pcieRcWriteEpConfig

Synopsis:

```
status_t pcieRcWriteEpConfig(pcieRootcompDataHandle_t *pdata,
        unsigned short offset,
        unsigned char size,
        unsigned char bus,
        unsigned char pcieDevice,
        unsigned char function,
        unsigned int value);
```

Purpose:

Write EP configuration area

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
*buffer	Pointer to input buffer from which data will written
offset	Register offset
size	size
bus	Bus number
pcieDevice	Device number

function Function number

Return Value:
0 on successful read and EINVAL on failure

2.16 pcieRcWriteEpMem

Synopsis:
status_t pcieRcWriteEpMem(pcieRootcompDataHandle_t *pdata,
unsigned int offset,
unsigned char *dataPtr,
unsigned int *addrPtr,
unsigned int numOfBytes);

Purpose:
Write EP memory

Arguments:
*pdata Pointer to RC private data structure of PCIe stack
offset Offset
*dataPtr Pointer to input data buffer
*addrPtr Pointer to memory address
numOfBytes Number of bytes to be written

Return Value:
0 on success and -1 on failure

2.17 pcieRcAtuObCfgTlpInfo

Synopsis:
status_t pcieRcAtuObCfgTlpInfo(pcieRootcompDataHandle_t *pdata,
unsigned int ob_rgn,unsigned int format,
unsigned int tlp_type, unsigned int traffic_class,
unsigned int is_id_order, unsigned int tlp_hints, unsigned int tlp_digest,
unsigned int poisoned_data, unsigned int attr, unsigned int address_type,
unsigned int length, unsigned int tag, unsigned int processing_hints);

Purpose:
Configure ATU outbound TLP information

Arguments:
*pdata Pointer to RC private data structure of PCIe stack
ob_rgn Outbound region

format	Format
tlp_type	TLP type
traffic_class	Traffic class
is_id_order	Indicates if the ID is ordered ID
tlp_hints	TLP hints
tlp_digest	TLP digest
poisoned_data	Indicates if data is poisoned data
attr	Attribute
address_type	Address type
length	Length
tag	Tag
processing_hints	Processing hints

Return Value:
0 on success and -1 on failure

2.18 pcieRcRegWrite

Synopsis:
status_t pcieRcRegWrite (pcieRootcompDataHandle_t *pdata, uint32_t size, uint32_t offset, uint32_t data);

Purpose: Register write

Arguments:
*pdata Pointer to RC private data structure of PCIe stack
size Size
offset Offset address
data Data to be written

Return Value:
0

2.19 pcieRcRegRead

Synopsis:
status_t pcieRcRegRead (pcieRootcompDataHandle_t *pdata, uint32_t size, uint32_t offset, void *data);

Purpose:
Register write

Arguments:
*pdata Pointer to RC private data structure of PCIe stack

size	Size
offset	Offset address
data	Data to be written

Return Value:
0

2.20 pcieRcResizeLink

Synopsis:
status_t pcieRcResizeLink(pcieRootcompDataHandle_t *pdata, int link_width);

Purpose: Alter the PCIe link size

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
link_width	Target link width

Return Value:
0 on success and -1 on failure

2.21 pcieRcUnusedTieOffLanes

Synopsis:
status_t pcieRcUnusedTieOffLanes(pcieRootcompDataHandle_t *pdata, int lanes);

Purpose: Tie off unused lanes

Arguments:

*pdata	Pointer to RC private data structure of PCIe stack
lanes	Lanes

Return Value:
0 on success and -1 on failure

2.22 pcieRcSpeedChangeRequest

Synopsis:
status_t pcieRcSpeedChangeRequest(pcieRootcompDataHandle_t *pdata, unsigned int target_speed_req);

Purpose:
Change PCIe link speed

Arguments:
*pdata Pointer to RC private data structure of PCIe stack
target_speed_req Target link speed to be requested

Return Value:
0 on success and -1 on failure

3. EP LDD WRAPPER API

3.1 pcieEpPreInitDev

Synopsis:
status_t pcieEpPreInitDev(struct pcie_dev_ep_prv_data **tmp_drv_data);

Purpose:
Allocate memory for device private data structure

Arguments:
**tmp_drv_data Pointer to RC private data structure pointer of device

Return Value:
0 on success, -1 on failure

3.2 pcieEpCfgDev

Synopsis:
status_t pcieEpCfgDev(struct pcie_dev_ep_prv_data *drv_data,
unsigned long ep_base,unsigned long pcie_axi_addr,
unsigned int num_of_funcs,unsigned char num_ob_regions,
unsigned char num_ib_regions,
unsigned char num_rd_dma_channels,unsigned char num_wr_dma_channels);

Purpose:
configure EP device

Arguments:
*drv_data Pointer to RC private data structure of device
ep_base EP base address
pcie_axi_addr AXI address
num_of_funcs Number of functions

num_ob_regions	Number of outbound regions
num_ib_regions	Number of inbound regions
num_rd_dma_channels	Number of read dma channels
Num_wr_dma_channels	Number of write dma channels

Return Value:
0 on success, -1 on failure

3.3 pcieEpConfigObDev

Synopsis:
```
status_t pcieEpConfigObDev(struct pcie_dev_ep_prv_data *drv_data,
        unsigned long base_addr,
        unsigned long trgt_addr,unsigned long len,unsigned int type,
        unsigned int ob_rgn);
```

Purpose:
configure device outbound region

Arguments:

*drv_data	Pointer to RC private data structure of device
base_addr	EP base address
trgt_addr	Target address
len	Length
type	Type
ob_rgn	Outbound region

Return Value:
0 on success, -1 on failure

3.4 pcieEpInitDev

Synopsis:
```
status_t pcieEpInitDev(struct pcie_dev_ep_prv_data *drv_data);
```

Purpose:
EP device initialize

Arguments:

*drv_data	Pointer to RC private data structure of device

Return Value:
0 on success, -1 on failure

3.5 pcieEpInitSetUpLdd

Synopsis:

```
int pcieEpInitSetUpLdd(struct ep_prv_data **pdata_pro,
                       struct pcie_dev_ep_prv_data **drv_data,
                       unsigned long ep_base,
                       unsigned long pcie_axi_addr,
                       unsigned int num_of_funcs,
                       unsigned char num_rd_dma_channels,
                       unsigned char num_wr_dma_channels,
                       unsigned long base_addr,
                       unsigned long trgt_addr,
                       unsigned long len,
                       unsigned int type,
                       unsigned long atu_ob_base_addr,
                       unsigned long atu_ob_trgt_addr,
                       unsigned long atu_ob_len,
                       unsigned int atu_ob_type);
```

Purpose:

Configure PCIe stack ldd layer values

Arguments:

**pdata_pro	Pointer to EP private data structure of device
**drv_data	Pointer to RC private data structure of device
ep_base	EP base address
pcie_axi_addr	AXI address
num_of_funcs	Number of functions
num_rd_dma_channels	Number of read dma channels
num_wr_dma_channels	Number of write dma channels
base_addr	Base address
trgt_addr	Target address
len	Length
type	Type
atu_ob_base_addr	ATU outbound base address
atu_ob_trgt_addr	ATU outbound target address
atu_ob_len	ATU outbound length
atu_ob_type	ATU outbound type

Return Value:

0 on success, -1 on failure

3.6 pcieEpDeinitDev

Synopsis:

status_t pcieEpDeinitDev(struct pcie_dev_ep_prv_data *drv_data);

Purpose:
Deinitialize device

Arguments:
*drv_data Pointer to RC private data structure of device

Return Value:
0 on success, -1 on failure

3.7 pcieEpDeinitialize

Synopsis:
status_t pcieEpDeinitialize(struct pcie_dev_ep_prv_data *drv_data,struct ep_prv_data *pdata);

Purpose:
Deinitialize memory allocated for device private data structures

Arguments:
*drv_data Pointer to RC private data structure of device
*pdata Pointer to the PCIe EP stack

Return Value:
0 on success, -1 on failure