SCF FAPI Support

Table of contents

List of Figures

Overview

Aerial cuBB supports the 5G FAPI 222.10.02 defined by the Small Cell Forum. This release supports most of the control interface (P5) and data path interface (P7) SCF messages.

SCF FAPI Messages Supported

The table below summarizes the status of the SCF FAPI messages supported.

SCF Messages	PDU Types	SCF L2 Adapter	SCF TestMAC	E2E with SCF TestMAC and RU Emulator
DL_TTI.request	PDCCH	Y	Y	Y
	PDSCH[7]	Y	Y	Y
	CSI-RS	Y	Y	Y[3]
	SSB	Y	Y	Y
UL_TTI.request	PRACH	Y	Y	Y
	PUSCH[7]	Y	Y	Y
	PUCCH	Y	Y	Y
	SRS[5][6]	Y	Y	Y
UL_DCI.request	PDCCH	Y	Y	Y
SLOT errors		N	N	N
TX_Data.request[1]	PDSCH	Y	Y	Y
Rx_Data.indication[1]	PUSCH (also contains RNTI, HARQ Id, UL_CQI, Timing adv, RSSI)	Y	Y	Y
CRC.indication	CRC	Y	Y	Y
UCI.indiaction	PUSCH[8]	Y	Y	Y
	PUCCH format 0,1	Y	Y	Y

	PUCCH format 2,3,4	Y	Y	PF2 and PF3 only
	SR for format 0,1	Y	Y	Y
	SR for format 2,3,4	Y	Y	Y[4]
	HARQ for format 0,1	Y	Y	Y
	HARQ for format 2,3,4	Y	Y	PF2 and PF3 only
	CSI part 1	Y	Y	Y
	CSI part 2	Y	Y	PUSCH only
	RSSI and UL SINR metrics	Y	Y	PUSCH, UCI on PUSCH and PF0,1,2,3
SRS.indication[5][6]	SRS	Y	Y	Y
RACH.indication	PRACH	Y	Y	Y
Config.request[2]		Y	Y	
Config.response		Y	Y	
Start.request		Y	Y	
Stop.request		Y	Y	
Stop.indication		Y	Y	
Error.indication		Y	Y	
Param.request		N	N	
Param.response		N	N	

Note[1]: The SCF implementation is based on SCF_222.10.02, but with the following exceptions:

- PDU Length of TX_DATA.request and RX_DATA.indication are changed to 32-bits. This is defined in SCF_222.10.03.

- The implementation supports multiple UE per TTI when the TLV tag is 2 in each PDU. However, the offset value in the TLV is ignored and L1 assumes all TBs in that slot placed in a flat buffer one after the other.

- The `RX_DATA.indication` FAPI message contains the MAC PDU (TB data) in the `data_buf` of the NVIPC message.

Field	Type	Description
TX_DATA.request PDU Length	uint16_t	The total length (in bytes) of the PDU description and PDU data, without the padding bytes. Value: 0 65535 Change type to uint32_t, value range is: 0 ~ 2^32 -1 **[NVIDIA change]**: Use it as the PDU data (TB data) size without the PDU description.
RX_DATA.Indication PDU Length	uint16_t	The length of PDU in bytes. A length of 0 indicates a CRC or decoding error. Value: 0 65535 Change type to uint32_t, value range is: 0 ~ 2^32 -1
RX_DATA.Indication PDU	Variable	The contents of PDU. This will be a MAC PDU. [NVIDIA workaround]: Removed this field, do not parse it in wireshark dissector. For SCF_222.10.04, although the tag value is set to 1, the MAC PDU is still delivered in a separate NVIPC buffer.
UL_TTI.request SRFlag	uint8_t	Indicates SR. Only valid for format 0 and 1. **[NVIDIA workaround]**: Enhance to use it as BitLenSr for format 2, 3, 4.

Note[2]: Precoding Matrix (Table 3-33) with vendor tag 0xA011 is supported. Digital beam table (Table 3-32) is not supported.

Note[3]: For NZP CSI-RS, only 4 antennas and single CSI-RS PDU.

Note[4]: The current implementation supports multi-bit SR over PUCCH format 2, 3, and 1. Because SCF FAPI 10.02 doesn't provide any field explicitly suggesting the bit length of the SR in the `PUCCH_PDU` of `UL_TTI.request`, use the `SRFlag` field to provide the SR bit length. For example, if the desired SR bit length is 3, set `SRFlag = 3`.

Note[5]: `SRS.indication` and SRS PDU in `UL_TTI.request` are supported according to SCF FAPI 222.10.02. SRS can be enabled when flag `enable_srs` is set in the cuphycontroller_xxx.yaml file i.e. `enable_srs: 1`.

Note[6]: `SRS.indication` and SRS PDU in `UL_TTI.request` are also supported according to SCF FAPI 222.10.04, which needs to be enabled with the "-DSCF_FAPI_10_04=ON" build option and flag `enable_srs` is set in the cuphycontroller_xxx.yaml file i.e. `enable_srs: 1`, as described in Running cuBB End-to-End.

- The format of the `SRS.indication` message is given in SCF FAPI 222.10.04 Table 3-129; the report TLV is defined in Table 3-130.
- The supported report type is Normalized Channel I/Q Matrix defined in Table 3.132 for codebook or nonCodebook SRS usage.
- The SRS Report TLV tag is 1 (customized value), the length is the actual report size in bytes without padding, the value field has the offset (in bytes) into the data_buf portion of NVIPC message for each SRS PDU. The report data is placed in the `data_buf` portion of the NVIPC message for all SRS PDUs.
- In case of wideband SRS, it is possible that the data_buf portion of NVIPC message carrying SRS.indication does not have enough space to accomodate SRS channel vectors for all the SRS PDUs. In this case, Aerial supports splitting of SRS.indication into multiple message. This feature can be enabled using CONFIG TLV 0x102B / indicationInstancesPerSlot as defined in 5G FAPI 222.10.04 specification table 3-36 for PHY configuration. If this TLV is not enabled by L2 and SRS.indication cannot accomodate all the SRS channel vectors, the SRS.indication will carry partial SRS information. On processing such a SRS PDU, an error indication with error code 0x35 is sent to L2 indicating partial SRS indication.
- Additionally Table 3.131 FAPIv3 Beamforming report, with PRG-level resolution for beamManagement SRS usage is also supported. The SRS Report TLV tag is 2 (customized value), is defined for encoding the SINR reports in the `msg_buf` at an offset of 32 bit from the value field, the length is the actual report size in bytes without padding. Also, currently PRG size of 2 is only supported.

Note[7]: If flag `mMIMO_enable` is set in the cuphycontroller_xxx.yaml file i.e. `mMIMO_enable: 1` to enable Dynamic Beamforming, indicates that the L2 shall encode the TX Precoding and Beamforming PDU & RX Beamforming PDU to include fields for numPRGs, prgSize and digBFInterface but L2 shall not encode the beamIdx because

when Dynamic Beamforming is used, L2 does not have information available for beamIds but L2 needs to provide the remaining information in the PDU to L1.

Note[8]: To get HARQ values in `UCI.indication` for UCI on PUSCH, before complete PUSCH slot processing, L2 should include PHY configurationTLV 0x102B (indicationInstancesPerSlot) with UCI.indication set to 2, according to Table 3–36 in SCF FAPI 222.10.04. If UCI.indication set to 2 in config.request for any cell the early HARQ feature will get activated for all cells.

Vendor Specific Message

A new vendor specific message `SLOT.response` was added after the 22-4 release. Before the 22-4 release, L2 has to set an event using the nvIPC notify function to inform L1 about “EOM” after sending the last FAPI message. This works well for single cell and when all FAPI messages are on time. L1 also uses the nvIPC notify function to set an event after sending each message.

The new `SLOT.response` FAPI message is used by L2 as the last FAPI message for each cell in each slot. It has the following advantages:

- It works as “EOM” for each cell in each slot.
- Each cell sends a `SLOT.response` as the last FAPI message of each slot.
- L2 should send `SLOT.response` even in empty slots (i.e. slots that have no scheduling).
- A “Dummy” or empty DL/UL TTI are optional/not-required.
- The notify event from L2 is optional/not-required.

The `SLOT.response` message format is shown below:

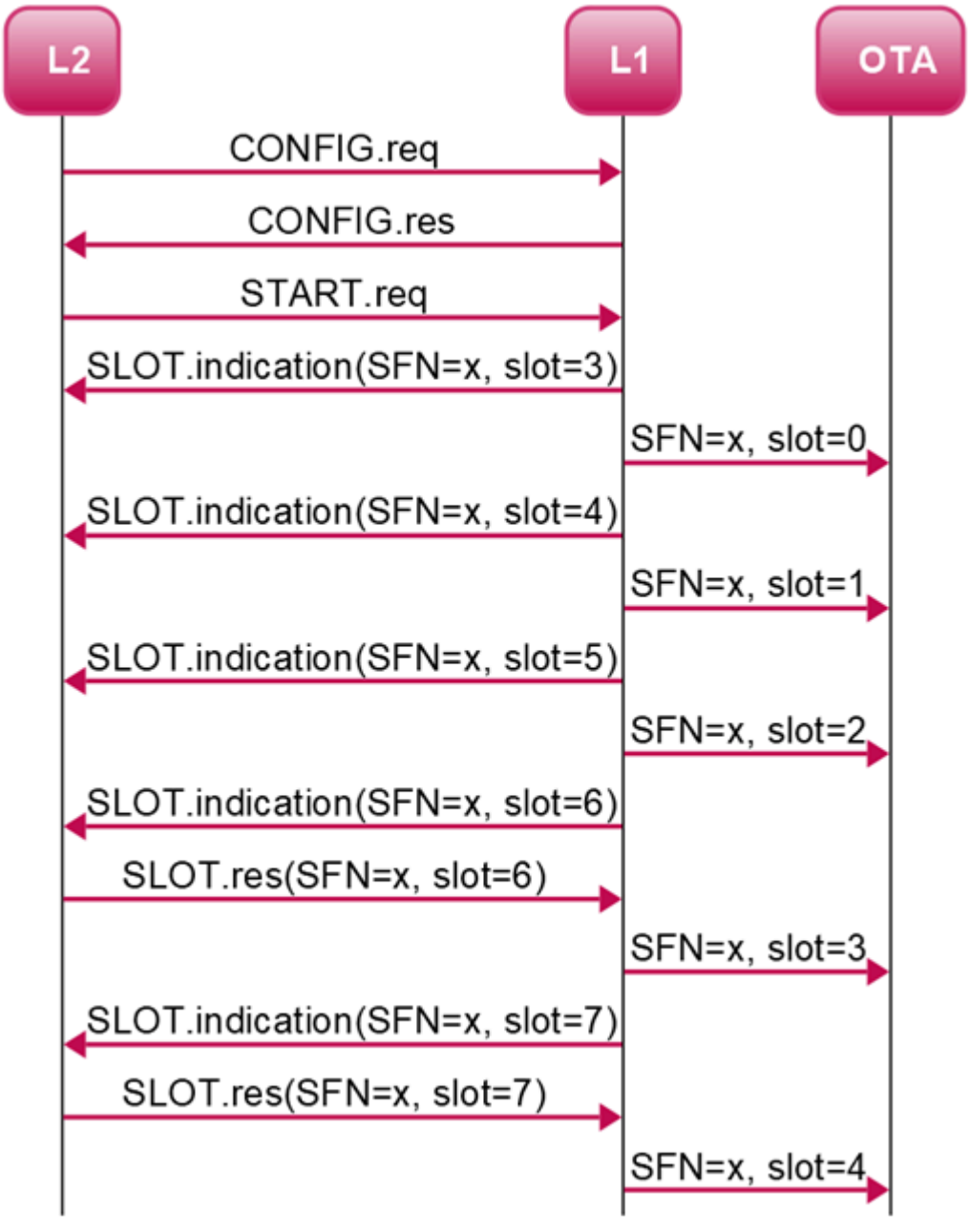
> /********** * Slot.response **********/ typedef struct { scf_fapi_body_header_t msg_hdr; uint16_t sfn; uint16_t slot; } __attribute__ ((__packed__)) scf_fapi_slot_rsp_t; Message-id 0x8F is used for this message { ...

SCF_FAPI_RX_PRACH_INTEFERNCE_INDICATION = 0x8E, SCF_FAPI_SLOT_RESPONSE = 0x8F, SCF_FAPI_RESV_2_END = 0xFF, } scf_fapi_message_id_e;

L1 continues to send a notify event after all FAPI messages to L2 to minimize impact on L2.

Message Sequence

An example message sequence is shown below:



> **Note**
>
> On receiving the first SLOT.indication, L2 is unable to send SLOT.response for 2-3 slots because it has a slot advance of 3.

Impact of Late Messages

- All messages are late for a cell (DL_TTI+TX_DATA+UL_DCI or UL_TTI)
    - All messages are dropped for the said cell. No impact on other cells.
- DL_TTI arrived on time but TX_DATA.request is late for a cell
    - This is considered as a partial slot. Due to cell grouping, PDSCH & DL-PDCCH is dropped for all cells.
- UL_TTI is late for a cell
    - ULSCH is not processed for the said cell. No impact on other cells.
- UL_DCI is late for a cell
    - UL-PDCCH is not processed for the said cell. No impact on other cells.
- `SLOT.response` is late for a cell
    - All FAPI messages received in time will be processed for the cell.

How to Enable or Disable SLOT.response

This feature is enabled by default in L1 after the 23-1 release. When integrating with L2, L2 is required to send this vendor-specific message in the manner described above.

Option ENABLE_L2_SLT_RSP should be configured with the same value in L1, L2 and libnvipc.so standalone build for L2. Refer to cuBB Quickstart Guide for details.

If L2 doesn't support the `SLOT.response` message, disable this feature by setting the "-ENABLE_L2_SLT_RSP=OFF" flag in the cmake command:

```
cmake .. -DENABLE_L2_SLT_RSP=OFF
```

Once the feature is enabled, the following is true:

- L2 has to send a vendor-specific `SLOT.response` message as the last FAPI message for each cell.
    - L2 to send this message even in empty slot (where nothing is scheduled).
- `allowed_fapi_latency` is deprecated and presumed to be 0.
    - L2 to complete sending all FAPI messages within the 500 us time-budget marked by `SLOT.indication` from L1.
    - Late FAPI messages will be dropped.
- A “Dummy” DL/UL TTI messages in empty slots is optional.
- A notify event after sending all FAPI messages is optional.
    - `ipc_sync_mode` in the L2 Adapter config file is deprecated.
- L1 will continue to send a Notify event after all FAPI messages to minimize impact on L2.

Dynamic Beamforming for 32T32R

- To enable this feature in Aerial software, flag `mMIMO_enable` should be set/introduced in the cuphycontroller_xxx.yaml file i.e. `mMIMO_enable: 1`.
- Two additional TLVs are required in CONFIG.req:
    - TLV 0xA016 denoting NUM_TX_PORT (uint8_t)
        - This field specifies the number of DL BB ports for PHY. 5G FAPI 222.10.04 described the field numTxAnt and numRxAnt in Table 3-37 as - ‘numTxAnt cannot exceed the number of DL BB ports for the PHY’. Hence the fields in table 3-37 represent the logical antenna ports.

        - 5G FAPI 223 describes baseband ports as a mapping between layers to RU TX/RX ports. PHY needs to know the BB ports from L2 (see Fig 3-3 in SCF-223.2.0.4).
        - This field will be used by PHY to read the number of DL BB ports.
        - If the TLV is not received from L2 and flag `mMIMO_enable` is set in the cuphycontroller_xxx.yaml file i.e. `mMIMO_enable: 1`, the default value for number of DL BB ports is set to 8.
    - TLV 0xA017 denoting NUM_RX_PORT (uint8_t)
        - This field specifies the number of UL BB ports for PHY. 5G FAPI 222.10.04 described the field numTxAnt and numRxAnt in Table 3-37 as - 'numRxAnt cannot exceed the number of UL BB ports for the PHY'. Hence the fields in table 3-37 represent the logical antenna ports.
        - 5G FAPI 223 describes baseband ports as a mapping between layers to RU TX/RX ports. PHY needs to know the BB ports from L2 (see Fig 3-3 in SCF-223.2.0.4).
        - This field will be used by PHY to read the number of UL BB ports
        - If the TLV is not received from L2 and flag `mMIMO_enable` is set in the cuphycontroller_xxx.yaml file i.e. `mMIMO_enable: 1`, the default value for number of UL BB ports is set to 4.

- DL & UL TTI have an additional field added for TRP scheme. See Note-6 in SCF FAPI Messages supported section
- Dynamic Beamforming is supported for PDSCH (RA type-1 and without CSI-RS) and PUSCH only
- A UE that is scheduled for SRS on S-slot should not be scheduled for dynamic beamforming of PDSCH and PUSCH in subsequent D & U slots until SRS indication for the UE is received. This prevents a race condition between L1 and L2 where the SRS channel vectors have been updated in the GPU hosted memory, but the latest SRS channel vectors are yet to be sent to L2. In this case, L2 might make a scheduling decision based on stale SRS channel vectors and the BFW calculation might happen with refreshed SRS channel vectors.

Two new FAPI messages have been defined from L2 to L1 to implement beamforming weight calculation in L1 as follows:

- SCF_FAPI_DL_BFW_CVI_REQUEST = 0x90
- SCF_FAPI_UL_BFW_CVI_REQUEST = 0x91

Structure of the FAPI message from L2 to L1 for beamforming weight calculation are as below. The same message structure is used for DL(PDSCH) and UL(PUSCH). When used for DL(PDSCH), it is referred to as DLBFW_CVI.request and when used for UL(PUSCH), it is referred to as ULBFW_CVI.request.

Table 3-1001 DLBFW_CVI.request message body

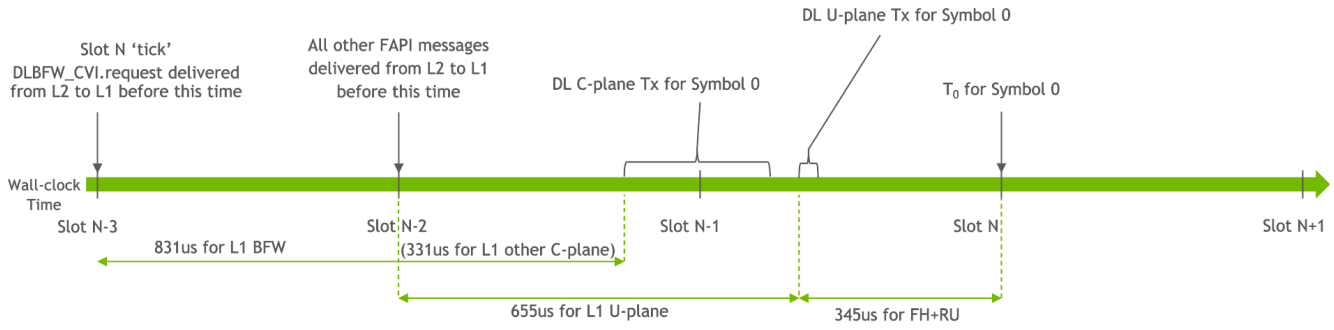
Field	Type	Description
SFN	uint16_t	SFN Value: 0 -> 1023
Slot	uint16_t	Slot Value: 0 ->159
nPDUs	uint8_t	Number of PDUs that are included in this message. All PDUs in the message are numbered in order. Each PDU is corresponding to a UE Group. Value: 0 -> 255
For Number of PDUs {		
PDUSize	uint16_t	Size of the PDU control information (in bytes). This length value includes the 4 bytes required for the PDU type and PDU size parameters. Value 0 -> 65535
DLBFW CVI Configuration	structure	See Table 3-1002 DLBFW CVI PDU
}		

Table 3-1002 DLBFW CVI PDU

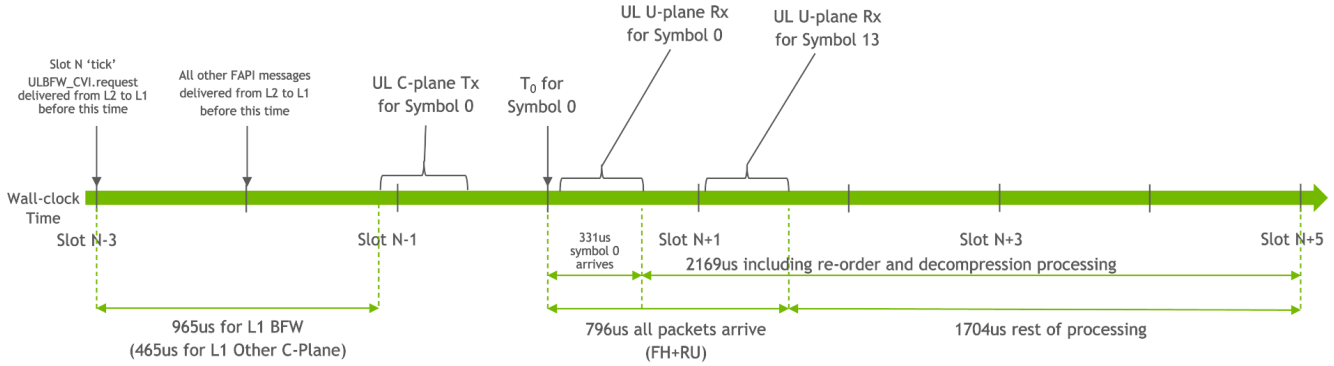
Field	Type	Description
rbStart	uint16_t	For resource allocation type 1. [TS38.214, sec 5.1.2.2.2] The starting resource block within the BWP for this PDSCH. Value: 0->274
rbSize	uint16_t	For resource allocation type 1. [TS38.214, sec 5.1.2.2.2] The number of resource block within for this PDSCH. Value: 1->275
numPRGs	uint16_t	Number of PRGs spanning this allocation. Value : 1->275
prgSize	uint16_t	Size in RBs of a precoding resource block group (PRG) – to which same precoding and digital beamforming gets applied. Value: 1->275
nUe	uint8_t	Number of UE in this group Value 1 -> 12
For Number of UEs {		
RNTI	uint16_t	The RNTI used for identifying the UE when receiving the PDU Value: 1 -> 65535.
pduIndex	uint16_t	PDU index associated to pduIndex in PDSCH PDU of DL_TTI.request Value: 0 -> 65535
gnbAntIdxStart	uint8_t	Corresponding to "gI" indicated by "Array representing channel matrix H" in SRS.indication. L1 can regard antenna indices from "gnbAntIdxStart" to "gnbAntIdxEnd" are used. Value: 0
gnbAntIdxEnd	uint8_t	Corresponding to "gI" indicated by "Array representing channel matrix H" in SRS.indication. L1 can regard antenna indices from "gnbAntIdxStart" to "gnbAntIdxEnd" are used. Value: 31
numOfUeAnt	uint8_t	Value: 1->3
For number of UE Ants {		
ueAntIdx	uint8_t	Corresponding to "uI" indiacated by "Array representing channel matrix H" in SRS.indication. Value: 0,1,2,3
}		
}		

Timeline for receiving DLBFW_CVI.request and ULBFW_CVI.request is as shown below:

Downlink timeline for slot N (32T32R)



Uplink timeline for slot N (32T32R) - PUSCH/PUCCH/PRACH)



Additional Aerial Specific Error Codes Reported in ERROR.indication from L1 to L2

Additional Aerial specific error codes have been added, starting from value 0x33, and L2 may receive these error codes in ERROR.indication message from L1 to L2. For example:

SCF_ERROR_CODE_FAPI_END = 0x32,

//Vendor specific error codes —- begin

SCF_ERROR_CODE_L1_PROC_OBJ_UNAVAILABLE_ERR = 0x33,

SCF_ERROR_CODE_MSG_LATE_SLOT_ERR = 0x34, //Indicates that L1's timer thread did not wake up on the slot boundary and slot indication for the indicated SFN,slot is late and will not be sent from L1 to L2

SCF_ERROR_CODE_PARTIAL_SRS_IND_ERR = 0x35, //Indicates partial SRS indication

SCF_ERROR_CODE_L1_DL_CPLANE_TX_ERROR = 0x36, //Indicates a DL C-plane trasmission error (Timing/Functional)

SCF_ERROR_CODE_L1_UL_CPLANE_TX_ERROR = 0x37, //Indicates a UL C-plane trasmission error (Timing/Functional)

SCF_ERROR_CODE_L1_DL_GPU_ERROR = 0x38, //Indicates a DL GPU pipeline processing error

SCF_ERROR_CODE_L1_DL_CPU_TASK_ERROR = 0x39, //Indicates a DL CPU Task incompletion error

SCF_ERROR_CODE_L1_UL_CPU_TASK_ERROR = 0x3A, //Indicates a UL CPU Task incompletion error

SCF_ERROR_CODE_L1_P1_EXIT_ERROR = 0x3B, //Indicates Part 1 of the error indication during L1 app exit process

SCF_ERROR_CODE_L1_P2_EXIT_ERROR = 0x3C, //Indicates Part 2 of the error indication during L1 app exit process post cudaDeviceSynchronize if CUDA coredump env variables are set

SCF_ERROR_CODE_L1_DL_CH_ERROR = 0x3D, //Indicates DL channel run (CPU/GPU) error

SCF_ERROR_CODE_L1_UL_CH_ERROR = 0x3E //Indicates UL channel run (CPU/GPU) error