Capture on J7 through CSI2RX Controller – Deep Dive

Camera
Serial
Interface

13th March 2019

Agenda

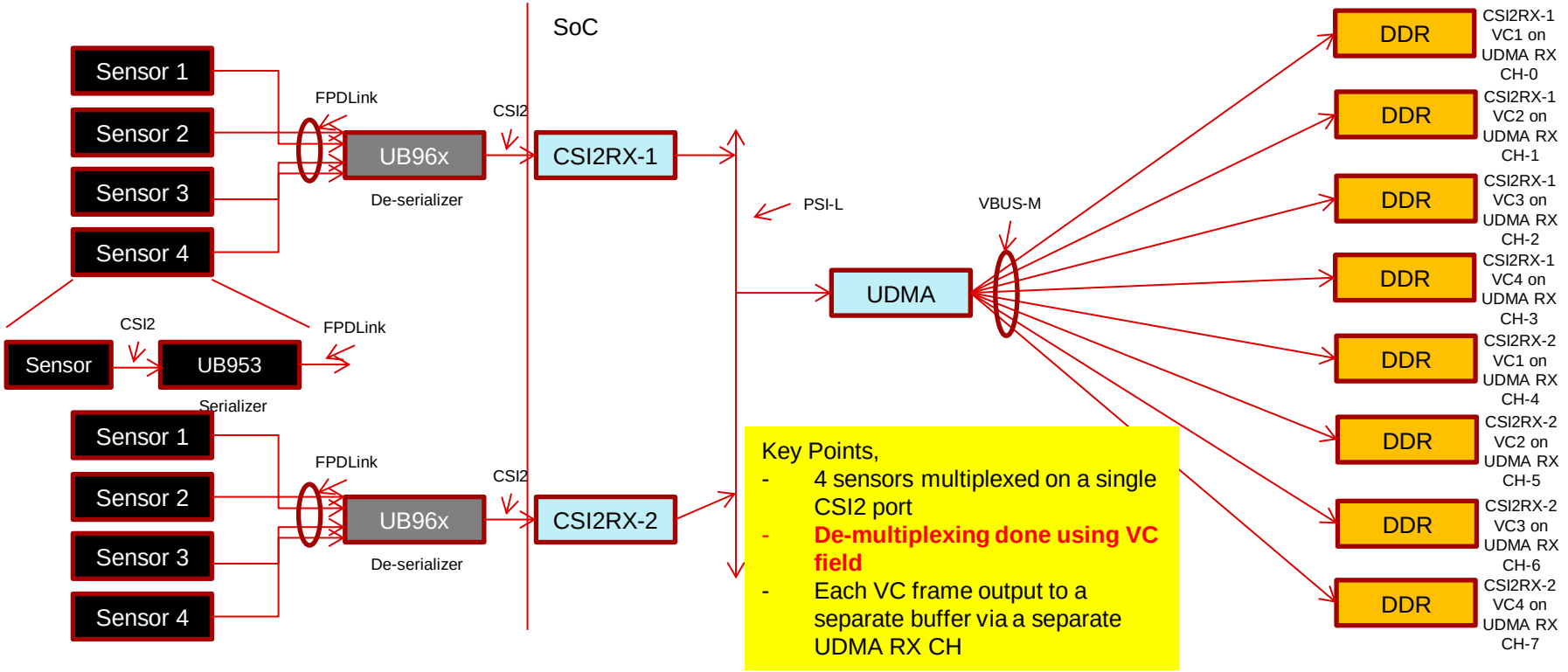
- Capture Use-cases on J7
- What’s CSI2?
- CSI2RX Controller
- CSI2RX FVID2 Driver

Acronyms

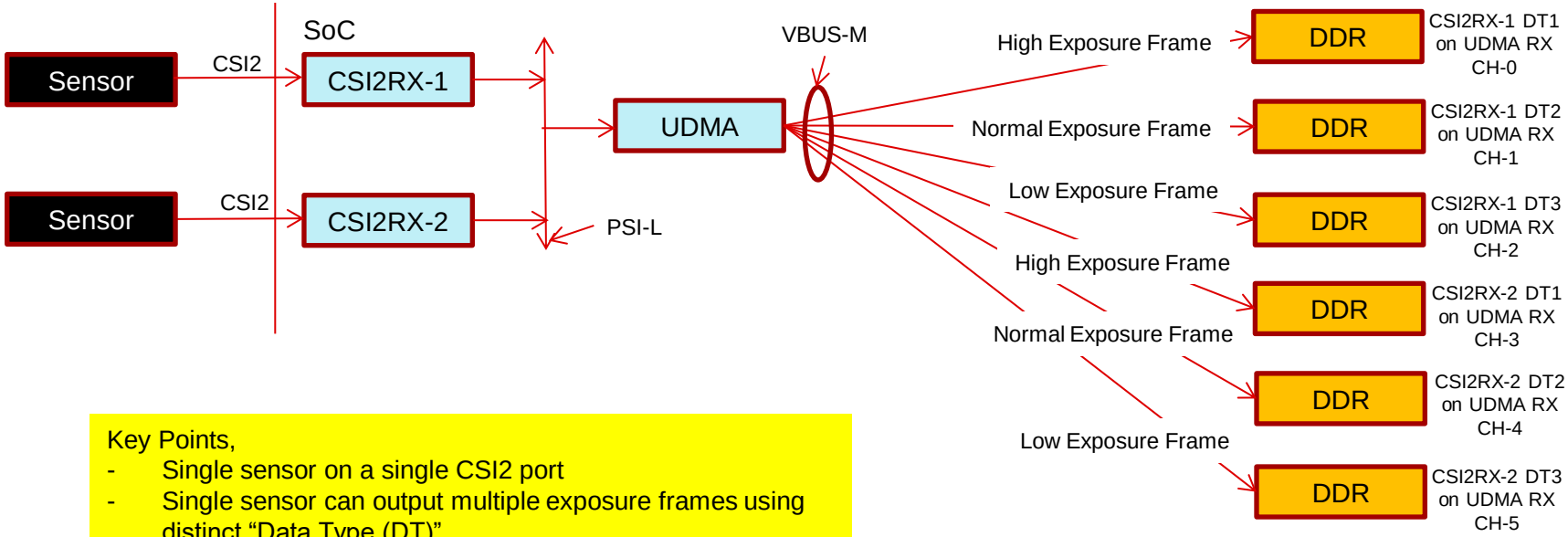
- CMS – Camera Mirror Systems
- OTF – On the fly
- WDR – Wide Dynamic Range
- VC – CSI2 Virtual Channel
- DT – CSI2 Data Type
- VISS – Vision Imaging Sub System
- ISP – Image Signal Processing
- bpp – bits per pixel
- ECC – Error Correction Code
- I2C – Inter-Integrated Circuit
- UDMA – Unified DMA Controller
- PSI – Packet Streaming Interface
- EMI – Electro Magnetic Interference
- CRC – Cyclic Redundancy Check
- DDR – Dual Data Rate
- SoC – System on Chip

Capture Use-case on J7

Capture Use-case(1/3): Multi-Sensor (e.g. Surround View, CMS, Radar)



Capture Use-case(2/3): Single-Sensor (e.g. Front Camera, CMS, Radar)

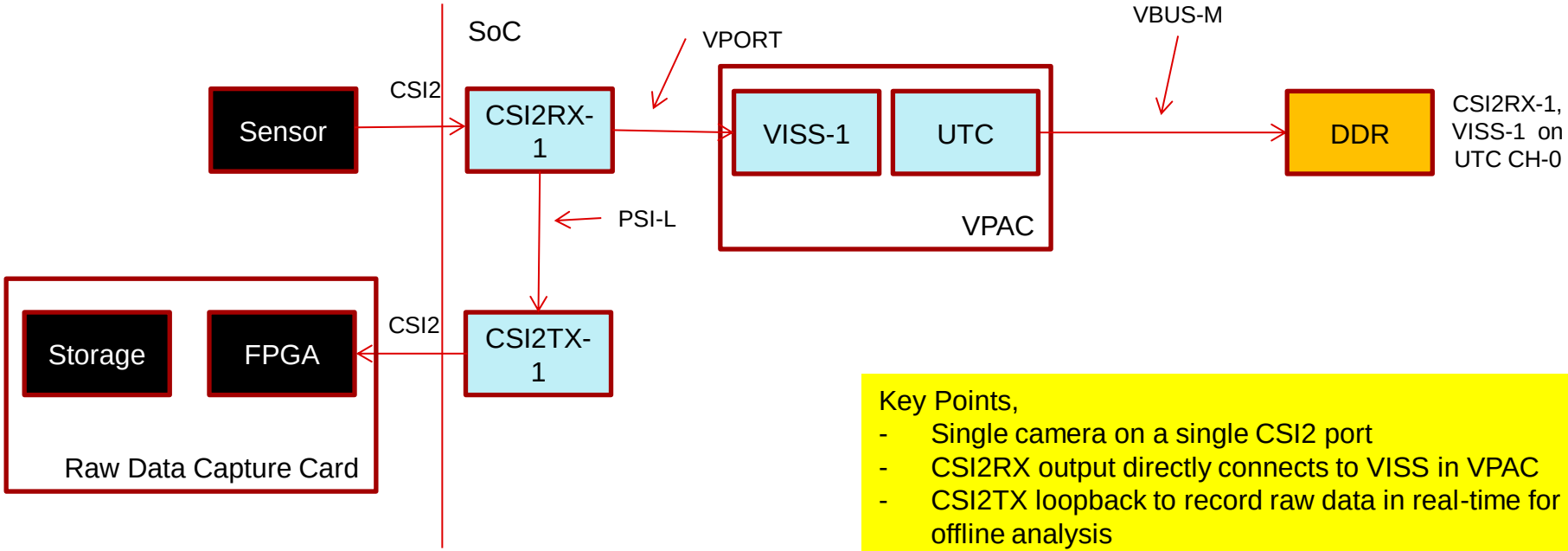


Key Points,

- Single sensor on a single CSI2 port
- Single sensor can output multiple exposure frames using distinct “Data Type (DT)”
- **De-multiplexing done using DT field**
- Each DT frame output to a separate buffer via a separate UDMA RX CH

DT1 = RAW10
DT2 = RAW12
DT3 = RAW14

Capture Use-case(3/3): Single-Sensor OTF (e.g. Front Camera, CMS)



Key Points,

- Single camera on a single CSI2 port
- CSI2RX output directly connects to VISS in VPAC
- CSI2TX loopback to record raw data in real-time for offline analysis
- Allows lower latency, lower DDR BW (some features like multi-exposure WDR not possible)

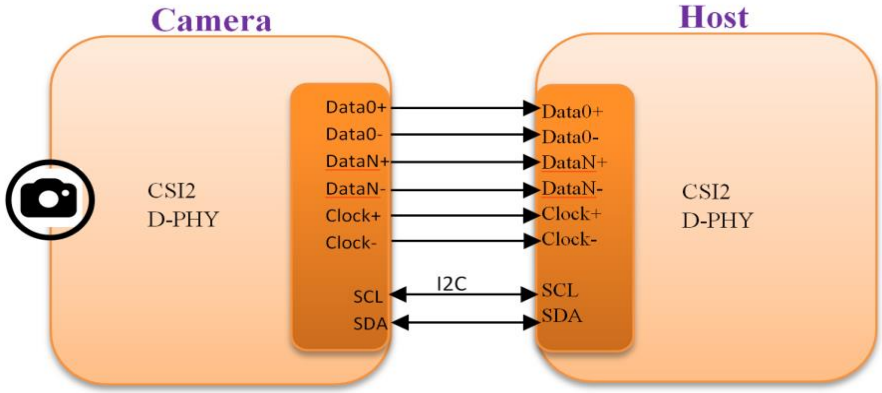
What’s CSI2?

Quick Facts

- Widely used in mobile and automotive applications
- High performance with low power and low EMI
- Compatible with *MIPI D-PHY* and/or *MIPI C-PHY*
- Unidirectional differential serial interface with high speed data and clock lanes
- Packet based protocol for data transmission
- Uses bi-directional control interface compatible with I2C standard

CSI2 Standard

- Up to 16 Virtual Channels(with v2.0)
- Data formats supported:
  - RAW6/7/8/9/10/12/14/(with v2.0)16/20
  - RGB444/555/565/666/888
  - YUV420 8/10 bit
  - YUV422 8/10 bit
- MIPI D-PHY: 2.5 Gbps per lane and 4 lanes in total



CSI2RX Controller

CSI2RX Controller: Features

- Compliant with MIPI CSI-2 v1.3
- Virtual channel extension and RAW16/20 for MIPI CSI2 v2.0 support
- Support of the MIPI CSI-2 protocol over DPHY PPI interface up to maximum 4*2.5 Gbps
- Configurable Data Lane positions
- Programmable Interrupt Events
- Protocol Error Detection
- Virtual Channel / Data type de-interleaving
- Processing of data on 4 independently configurable Streams
- Payload FIFO operation

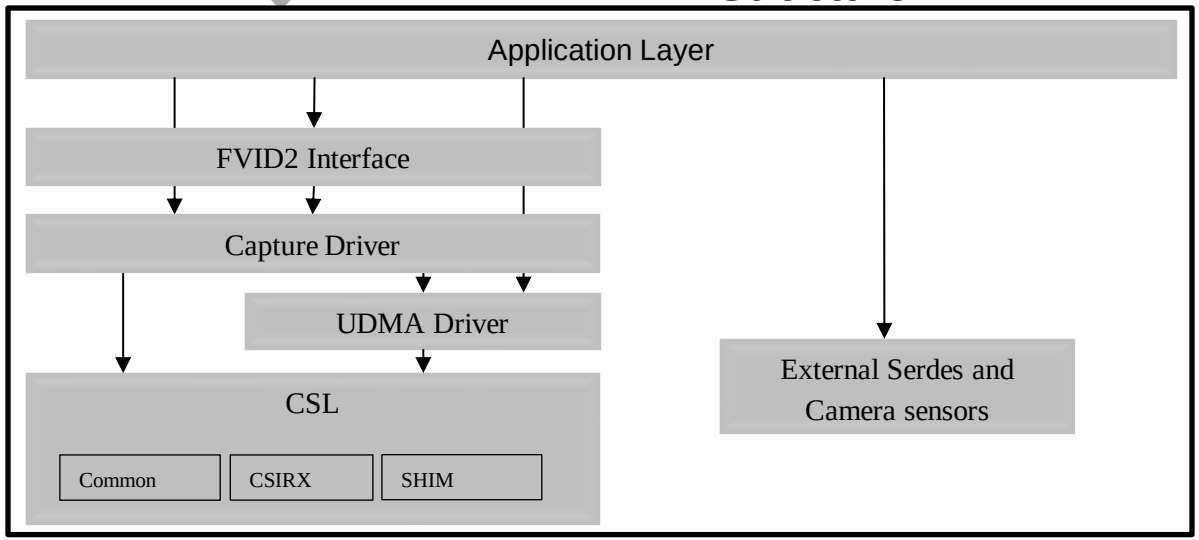
CSI2RX FVID2 Driver

CSI2RX FVID2 Driver: Features

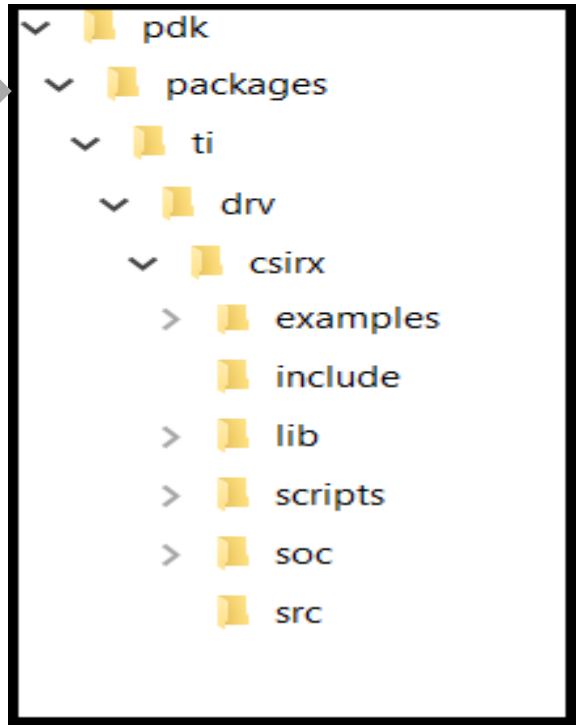
- FVID2 Interface for applications
- Capture of following formats:
  - RAW8/10/12/14/16/20
  - YUV422-10 bit
  - RGB888
- Up to 32 capture channels per CSI2RX controller instance
- OTF mode and loop-back mode to re-transmission pads of CSI2TX controller
- Error Handling and Recovery
- FIFO handling
- D-PHY configuration
- Configurable number of Data Lanes to use and their ordering
- Frame drop buffer programming in case of Queue did not happen in time

CSI2RX FVID2 Driver: Overview(1/2)

Capture Application Layers



CSIRX Driver Directory Structure



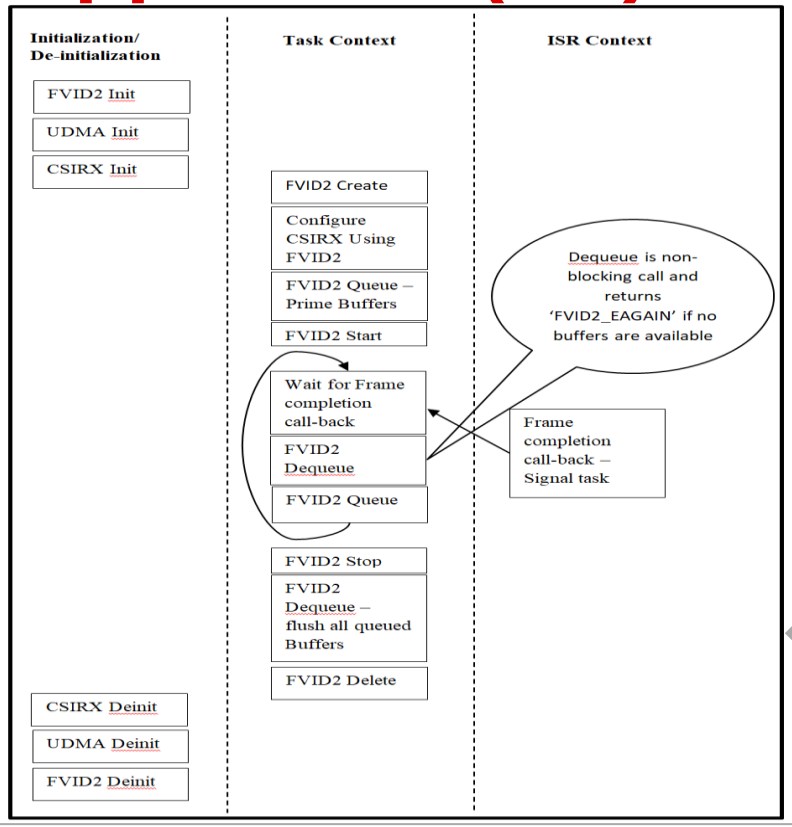
CSI2RX FVID2 Driver: Overview(2/2)

- Interface files:
  - *pdk/packages/ti/drv/csirx/csirx.h*
    - It is a capture driver interface file. Application should only include this file.
  - *pdk/packages/ti/drv/csirx/include/csirx_cfg.h*
    - It is a capture drivers configuration file.
- Implementation files:
  - *pdk/packages/ti/drv/csirx/src*
- SoC files:
  - *pdk/packages/ti/drv/csirx/soc*

CSI2RX FVID2 Driver: Understanding FVID2 Interface

- FVID2 APIs:
  - *FVID2_init*
    - Initializes the drivers and the hardware. Should be called before calling any of the FVID2 functions
  - *FVID2_deInit*
    - Un-initializes the drivers and the hardware
  - *FVID2_create*
    - Opens a instance/channel video driver
  - *FVID2_delete*
    - Closes a instance/channel of a video driver
  - *FVID2_control*
    - To send standard (set/get format, alloc/free buffers etc..) or device/driver specific control commands to video driver
  - *FVID2_queue*
    - Submit a video buffer to video driver. Used in display/capture drivers
  - *FVID2_dequeue*
    - Get back a video buffer from the video driver. Used in display/capture drivers
  - *FVID2_start*
    - Start video capture or display operation.
  - *FVID2_stop*
    - Stop video capture or display operation.

CSI2RX FVID2 Driver: Usage - Application(1/3)



Building CSI2RX Driver:

```
gmake.exe -s -j csirx BOARD=j721e_sim
CORE=mcu2_0 BUILD_PROFILE=release
```

Building CSI2RX Driver Sample Application:

```
gmake.exe -s -j csirx_capture_testapp
BOARD=j721e_sim CORE=mcu2_0
BUILD_PROFILE=release
```

Application Calling Sequence

CSI2RX FVID2 Driver: Usage-Configurations(2/3)

- Create parameters(Csirx_CreateParams):
  - numCh
    - Number of channels to be configured/processed
  - chCfg
    - Channel configuration
  - instCfg
    - Instance configuration
  - frameDropBuf
    - Address of Frame Drop buffer
  - frameDropBufLen
    - Frame Drop buffer length in bytes
- Channel configuration(Csirx_ChCfg):
  - chId
    - Unique channel ID
  - chType
    - Channel type: Capture, OTF, loop-back
  - vcNum
    - Virtual channel number
  - inCsiDataType
    - CSI2 data format for capturing
  - outFmt
    - Frame attributes like dimension, storage format specifier
- Instance configuration(Csirx_InstCfg):
  - enableCsiv2p0Support
    - Optional CSI2 v2.0 support enable control
  - numDataLanes
    - Number of data lanes used for capturing
  - dataLanesMap
    - Position of data lanes
  - enableErrbypass
    - Error bypass mode control

CSI2RX FVID2 Driver: Usage-Example(3/3)

- Two channels RGB888 (1920 x 1080) capture configuration:

Code Snippet:

```
createPrms.numCh = 2U;
createPrms.chCfg[0U].chId = 0U;
createPrms.chCfg[0U].chType = CSIRX_CH_TYPE_CAPT;
createPrms.chCfg[0U].vcNum = 0U;
createPrms.chCfg[0U].inCsiDataType = FVID2_CSI2_DF_RGB888;
createPrms.chCfg[0U].outFmt.width = 1920U;
createPrms.chCfg[0U].outFmt.height = 1080U;
createPrms.chCfg[0U].outFmt.pitch[0U] = (1920U * 4U);
createPrms.chCfg[0U].outFmt.dataFormat = FVID2_DF_BGRX32_8888;
createPrms.chCfg[1U].chId = 1U;
createPrms.chCfg[1U].chType = CSIRX_CH_TYPE_CAPT;
createPrms.chCfg[1U].vcNum = 1U;
createPrms.chCfg[1U].inCsiDataType = FVID2_CSI2_DF_RGB888;
createPrms.chCfg[1U].outFmt.width = 1920U;
createPrms.chCfg[1U].outFmt.height = 1080U;
createPrms.chCfg[1U].outFmt.pitch[0U] = (1920U * 4U);
createPrms.chCfg[1U].outFmt.dataFormat = FVID2_DF_BGRX32_8888;
```

CSI2RX FVID2 Driver: Un-supported Features

- Capture of following formats:
  - RAW6/7
  - YUV420-8 bit
  - RGB565/RGB666
- Dynamic Stream configuration
- Lane Polarity Position change
- Clock Lane position configuration

Questions?
Thank You